

KOS.content

02 | 2014

Ergebnisse der Untersuchungen des
Kompetenzzentrum Open Source der DHBW-Stuttgart

Sommer 2014
band.1

INHALT BAND.1

Inhalt __

Vergleichsstudie zu Open Source Produkten für das Anforderungsmanagement __ 1

Vergleich und Prototyp von Open-Source-ETL-Tools __ 93

Marktanalyse clientseitiger Open Source Frameworks zum Bau browserbasierter Anwendungen __ 179

Open Source Workflow-Management-Systeme __ 263

Konzeption und Implementierung eines Performance-Test für Key-Value-Datenbanken __ 309

Methodik/Systematik einer Beratungsdienstleistung für ein OS Beratungszentrum __ 383

Textanalyse am Beispiel von Beschwerdeschreiben (PostgreSQL) __ 437

Textanalyse am Beispiel von Beschwerdeschreiben (MongoDB) __ 461

Das Kompetenzzentrum Open Source (KOS)

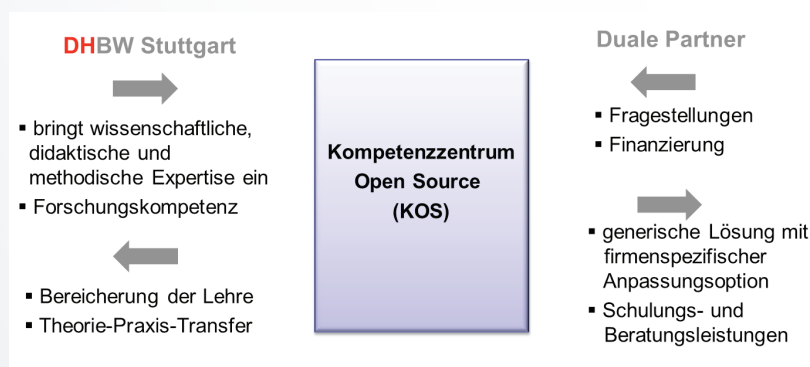
Ziel des Projektes

Das Projekt Kompetenzzentrum Open Source der DHBW Stuttgart wurde mit der Zielsetzung ins Leben gerufen, die Einsatzfelder für Open Source Software in Unternehmen zu identifizieren und durch den Einsatz quelloffener Produkte und deren kostengünstigen Einsatzmöglichkeiten Optimierungen in ausgewählten Geschäftsbereichen zu erzielen.

Dies bedeutet konkret, dass z.B. Open Source Software evaluiert wird, um Lizenzkosten zu reduzieren, bewertet wird, ob sie diverse Qualitätskriterien erfüllt und erfolgreich(er) und effizient(er) in Unternehmen genutzt werden kann. Das Ziel des Projektes ist es hierbei, allgemeingültige Lösungskonzepte für Problemstellungen zu erarbeiten, welche von den am Projekt beteiligten Unternehmen zu firmenspezifischen Lösungen weiterentwickelt werden können. Die beteiligten Unternehmen partizipieren so an den Ergebnissen des Projekts.

Zusammenarbeit mit den Dualen Partnern

Die Zusammenarbeit mit den Dualen Partnern gestaltet sich entlang deren Anforderungen und Bedürfnissen. Sie sind die Themengeber für betriebliche Fragestellungen, die im Rahmen des Projekts untersucht werden. Die DHBW steuert die wissenschaftliche, didaktische und methodische Expertise und Forschungskompetenz bei und untersucht die identifizierten Themenfelder.



Im Rahmen des Projektes steuert die DHBW Stuttgart die wissenschaftliche Expertise und Forschungskompetenz bei zur Bearbeitung der betrieblichen Fragestellungen der Dualen Partner. Es entstehen generische Lösungen, welche von den Partnern an Ihre Situation angepasst werden kann.

Im Rahmen der Arbeit entstehen (generische) Lösungen, an denen die Partner teilhaben können indem sie diese auf ihre spezifische Unternehmenssituation anpassen. Zudem fließen die Ergebnisse in die Arbeit der DHBW ein, sodass hier dem Anspruch an eine hohe Anwendungs- und Transferorientierung ganz im Sinne einer kooperativen Forschung Rechnung getragen wird.

An den Ergebnissen des Projekts partizipieren die Dualen Partner Allianz Deutschland AG, die Deutsche Rentenversicherung Baden-Württemberg, die HALLESCHKE Krankenversicherung a.G. und die WGV-Informatik und Media GmbH.

Vergleichsstudie zu Open Source Produkten für das Anforderungsmanagement

Schriftliche Ausarbeitung
im Rahmen der Lehrveranstaltung „Integrationsseminar“

Vorgelegt von

Cathrin Kahre
Isabel Shen
Annika Kunde
Philipp Richter
Tobias Tröndle

am 11.07.2014

Fakultät Wirtschaft
Studiengang Wirtschaftsinformatik
Kurs WI2011I

Inhaltsverzeichnis

Abkürzungsverzeichnis	III
Abbildungsverzeichnis.....	IV
1 Einleitung	1
2 Anforderungsmanagement	3
2.1 Was ist Anforderungsmanagement	3
2.1.1 <i>Definition</i>	3
2.1.2 <i>Ziele von Anforderungsmanagement</i>	4
2.1.3 <i>Funktionen</i>	6
2.2 Was ist Anforderungsmanagement-Software	6
2.2.1 <i>Definition und Ziele</i>	7
2.2.2 <i>Abgrenzung</i>	8
3 Markt der Anforderungsmanagement-Software	9
3.1 Volumen des Marktes	9
3.2 Verfügbare Produkte	11
3.2.1 <i>Kommerzielle Produkte</i>	11
3.2.2 <i>Open Source Produkte</i>	17
4 Aufbau eines Kriterienkataloges	21
4.1 Konsolidierung des Kriterienkataloges	21
4.1.1 <i>Vorgehensweise</i>	21
4.1.2 <i>Ergebnis: Der Kriterienkatalog</i>	23
4.2 Zuordnung der Funktionalitäten nach Unternehmensgröße.....	32
4.3 Einordnung der verfügbaren Produkte in die Zielgruppen.....	36
4.3.1 <i>Kommerzielle Software Produkte</i>	36
4.3.2 <i>Open Source Software</i>	49
5 Vergleichsstudie der Open Source Produkte	54
5.1 Große Unternehmen	54
5.1.1 <i>Aktueller Marktführer: IBM Rational DOORS</i>	54
5.1.2 <i>Betrachtung der Eignung der Open Source Produkte</i>	57
5.1.3 <i>Mögliche Substitutionsprodukte für IBM Rational DOORS</i>	58
5.2 Mittlere Unternehmen.....	61
5.2.1 <i>Aktueller Marktführer: OneDesk</i>	61
5.2.2 <i>Betrachtung der Eignung der Open Source Produkte</i>	62
5.2.3 <i>Mögliche Substitutionsprodukte für OneDesk</i>	64
5.3 Kleine Unternehmen	68
5.3.1 <i>Aktueller Marktführer: CASE Spec</i>	68
5.3.2 <i>Analyse einer möglichen Alternative aus dem Open Source Segment</i>	70
5.3.3 <i>Vergleich des kommerziellen Produktes und der Open Source Lösung</i>	75
5.3.4 <i>Bewertung des Open Source Potenzials</i>	77
6 Fazit	79
Anhang.....	81
Quellenverzeichnisse	84

Abkürzungsverzeichnis

ALM	Application Life Cycle Management
API	Application Programming Interface
CAGR	Jährliche Wachstumsrate
CLM	Collaborative LifeCycle Management
CRM	Customer Relationship Management
CSV	Comma-seperated Values
ERP	Enteprise Resource Planning
HTML	Hypertext Markup Language
IEEE	Institute of Electrical and Electronics Engineers
IT	Informationstechnologie
LDAP	Lightweight Directory Access Protocol
MRD	Marketing Requirements Document
PDF	Portable Document Format
PRD	Product Requirements Document
QC	Quality Center
ReqIF	Requirements Interchange Format (früher: RIF)
RM	Requirements Management
RMF	Requirements Modeling Framework
RTF	Rich Text Format
SaaS	Software-as-a-Service
SAML	Security Assertion Markup Language
SHORE	Semantic Hypertext Object Repository
SRS	Software Requirements Specification
SWOT	Strengths-Weaknesses-Opportunities-Threats
TFS	Team Foundation Server
UML	Unified Modeling Language

Abbildungsverzeichnis

Abb. 1: Entstehungszeitpunkt von Fehlern in Softwareentwicklungsprojekten	5
Abb. 2: Weltweiter Umsatz Anforderungsmanagement-Software 2012 nach Anbieter	10
Abb. 3: Schematische Vorgehensweise bei der Erstellung des Kriterienkataloges	21
Abb. 4: Ermittelte Kategorien des Kriterienkataloges	24
Abb. 5: Benutzeroberfläche von ProR	76

Tabellenverzeichnis

Tabelle 1: Übersicht kommerziell verfügbarer Anforderungsmanagement Software	16
Tabelle 2: Übersicht der Open Source Lösungen des Anforderungsmanagements	20
Tabelle 3: Kategorie „Allgemeines“ des Kriterienkataloges	25
Tabelle 4: Kategorie „Anforderungsmanagement“ des Kriterienkataloges	26
Tabelle 5: Kategorie „Änderungsmanagement“ des Kriterienkataloges	27
Tabelle 6: Kategorie „Collaboration“ des Kriterienkataloges	27
Tabelle 7: Kategorie „Aufbau und Navigation“ des Kriterienkataloges	28
Tabelle 8: Kategorie „Administration“ des Kriterienkataloges	28
Tabelle 9: Kategorie „Zugang und Integration“ des Kriterienkataloges	29
Tabelle 10: Kategorie „Daten Import & Export“ des Kriterienkataloges	30
Tabelle 11: Kategorie „Analyse“ des Kriterienkataloges	31
Tabelle 12: Zuordnungsmatrix der Produkte nach Unternehmensgröße	32
Tabelle 13: Übersicht der zugeordneten Basic-, Advanced- und Optional-Funktionalitäten ..	34
Tabelle 14 Zuteilung der kommerziellen Produkte zu den Zielgruppen	37
Tabelle 15 Zuteilung der Open Source Produkte zu den Zielgruppen	49
Tabelle 16: SWOT-Analyse des Open Source Produktes ProR	72

1 Einleitung

“Die IT, die an sie formulierten Anforderungen und erfolgreich abgeschlossene Projekte – diese drei Punkte scheinen nur schwer zusammenzufinden.”

Thomas Niebisch¹

Was Thomas Niebisch in seinem Buch “Anforderungsmanagement in sieben Tagen” unter der Überschrift “Das IT-Paradoxon” beschreibt, spiegelt eine Problematik wider, welche in vielen Projekten der Informationstechnologie (IT) aktuell zu Tage tritt. Das Anforderungsmanagement als Schnittstelle zwischen den Wünschen der Fachabteilung und der Umsetzung der IT-Abteilung wird zu häufig unterschätzt und ohne die nötige Aufmerksamkeit schnell abgehandelt. Hierdurch entstehen jedoch schwerwiegende Fehler, für deren Korrektur im Nachhinein sowohl zusätzliche zeitliche als auch finanzielle Ressourcen aufgewendet werden müssen.

Um dieses Risiko zu umgehen, ist es zum einen nötig, für die notwendige Sensibilisierung zu dem Thema zu sorgen und zum anderen den Mitarbeitern im Unternehmen Werkzeuge und Hilfestellungen anzubieten, mit welchen ein erfolgreiches und effizientes Anforderungsmanagement betrieben werden kann. Hierzu zählt in erster Linie ein IT-gestütztes Anforderungsmanagementsystem, mit welchem Anforderungen aufgenommen, kategorisiert, zugeordnet und verwaltet werden können.

Problemstellung

Der Markt für Anforderungsmanagement-Software ist jedoch sehr umfangreich und es fällt schwer, inmitten der Vielzahl an Produkten den Überblick zu behalten und die optimale Lösung für das eigene Unternehmen zu finden. Auch die im Internet verfügbaren Vergleichsstudien helfen nur bedingt weiter, da diese meist direkt oder indirekt von einem der Anbieter in Auftrag gegeben wurden und dementsprechend die gewünschten Ergebnisse präsentieren. Zudem finden sich im Internet keinerlei Studien, welche Open Source Produkte in den Vergleich miteinbeziehen.

Zielsetzung der Arbeit

Um diese Lücke ein Stück weit zu schließen, soll im Rahmen dieser Arbeit ein Vergleich verschiedener Anforderungsmanagement-Tools erfolgen. Nachdem im ersten Schritt Grundlagen zum Anforderungsmanagement erläutert werden und ein Marktüberblick erfolgt, soll der anschließende Vergleich sowohl die Unterschiede verschiedener kommerzieller Produkte zeigen, als auch Open Source Produkte miteinbeziehen. Bei der Untersuchung der Open

¹ Niebisch, T. (2013), S. 11

Source Lösungen sollen insbesondere auch Open Source Spezifika, wie die Größe der Entwicklercommunity oder der Zeitpunkt des letzten Updates in die Betrachtung miteinfließen.

Neben dem Vergleich verschiedener Anforderungsmanagement-Lösungen soll ein weiterer Schwerpunkt auf der Einordnung der untersuchten Produkte nach Eignung für bestimmte Unternehmensgrößen liegen. So sollen jeweils sowohl kommerzielle als auch Open Source Tools identifiziert werden, welche sich besonders für den Einsatz in kleinen, mittelgroßen und großen Unternehmen eignen. Durch diesen Schritt soll es zusätzlich erleichtert werden, im konkreten Bedarfsfall schnell ein geeignetes Produkt identifizieren zu können, welches zum einen über den benötigten Funktionsumfang verfügt und zum anderen auch in einem bezahlbaren Rahmen liegt.

Zusammenfassend ist es somit das Ziel dieser Arbeit, einen **unabhängigen und objektiven Überblick** über die verschiedenen Produkte im Bereich Anforderungsmanagement zu geben und einige dieser Produkte näher zu untersuchen und miteinander zu vergleichen. Ebenso sollen Open Source Lösungen analysiert werden. Die **Zuordnung geeigneter Produkte für bestimmte Unternehmensgrößen** soll helfen, im Individualfall schnell geeignete Produkte identifizieren zu können. Auch in diesem Schritt sollen sowohl kommerzielle, als auch Open Source Produkte berücksichtigt werden, um das volle Spektrum der aktuell auf dem Markt verfügbaren Anforderungsmanagement-Lösungen abzubilden.

2 Anforderungsmanagement

Kapitel 2 dient als Einführung in das Anforderungsmanagement. In diesem Sinne soll hier auf das Anforderungsmanagement an sich sowie die technische Unterstützung in Form von Anforderungsmanagement-Software eingegangen werden.

2.1 Was ist Anforderungsmanagement

Dieses Unterkapitel geht auf das Anforderungsmanagement im Allgemeinen ein und liefert entsprechend zunächst eine Definition, bevor die Ziele des Anforderungsmanagements betrachtet werden. Abschließend wird auf die Funktionen dieses Unternehmensgebiets eingegangen.

2.1.1 Definition

Anforderungsmanagement ist ein Anwendungsgebiet des Business Engineering. Business Engineering „ist die methodische und ganzheitliche Gestaltung und Optimierung der Geschäftsprozesse einer Organisation“². Hierbei werden Geschäftsmodelle herangezogen und aus Sicht der Kunden umstrukturiert, um das Unternehmen effizienter zu führen. Im Rahmen der Umstrukturierung werden Anforderungen aufgenommen, welche beschreiben, wie die zum Unternehmen zugehörigen Prozesse in Zukunft ablaufen werden.³ Im Laufe dieses Business Engineering werden Anforderungen immer wieder verändert, das heißt sie werden angepasst und weiterentwickelt oder es kommen neue Anforderungen hinzu. In dieser Arbeit wird eine sehr allgemein gehaltene Definition des Anforderungsmanagements gewählt, da hier lediglich eine Einführung ins Thema gegeben werden soll. So sind die Ermittlung, Dokumentation, Validierung und Verwaltung der Anforderungen sind Gegenstände des Anforderungsmanagements.⁴

Weitere Anwendungsgebiete des Anforderungsmanagements sind die Produkt- und Softwareentwicklung, wobei sich diese Arbeit auf Softwareentwicklungsprojekte bezieht. Die dokumentierten Anforderungen stellen die Basis für die Kommunikation zwischen Auftraggeber und Auftragnehmer dar. In Meetings werden die Anforderungen des Kunden gesammelt und besprochen⁵, wobei das Verständnis der Fachsprache des Kunden eine Schwierigkeit für den Auftragnehmer darstellt. Jede Anforderung muss deswegen besprochen und dem Auf-

² Schienmann, B. (2002), S. 305

³ Vgl. ebenda, S. 305

⁴ Vgl. Rupp, C. (2008), S. 18

⁵ Vgl. Sommerville, I. (2012), S. 137

tragnehmer in einer für ihn zugänglichen Sprache verständlich kommuniziert werden. Nach der Aufnahme der Anforderungen ist es wichtig, dass diese vom Auftraggeber noch einmal überprüft werden, um sicherzustellen, dass im Entwicklungsprozess die korrekten Anforderungen umgesetzt werden. Zudem bildet die Dokumentation und Verwaltung der Anforderungen die rechtliche Grundlage für den Erfolg des Projektes.⁶ Beide Parteien können sich bei Übergabe des Projektes auf die dokumentierten Anforderungen berufen, so dass Streitigkeiten über die Umsetzung von Anforderungen ausgeschlossen werden können. Aus diesem Grund ist die Verwaltung der Anforderungen über die Aufnahme der zu Beginn des Projektes ermittelten Anforderungen hinaus sehr wichtig. Jede Änderung muss im Katalog der Anforderungen festgehalten werden, um eine rechtlich korrekte Form zu gewährleisten.

Zusammenfassend lassen sich die folgenden Aufgabenbereiche des Anforderungsmanagements festhalten:

- Ermittlung von Anforderungen
- Dokumentation von Anforderungen
- Prüfung und Validierung von Anforderungen
- Verwaltung von Anforderungen
- Pflege des Anforderungskatalogs
- Kommunikation mit dem Kunden
- Arbeit mit Werkzeugen für das Anforderungsmanagement⁷

2.1.2 Ziele von Anforderungsmanagement

In erster Linie ist das Ziel des Anforderungsmanagements die Schaffung eines gemeinsamen Verständnisses der zu entwickelnden Lösung.⁸ Normalerweise kommen Auftraggeber und Auftragnehmer aus verschiedenen Branchen, in denen die Unternehmen unterschiedliche Fachsprachen gebrauchen. Im Rahmen des Anforderungsmanagements muss eine gemeinsame Kommunikationsbasis geschaffen werden, damit beiden Parteien den Inhalt des Projektes verstehen.

Außerdem ist das Vermeiden von Fehlern in der Umsetzung ein sehr wichtiges Ziel, da so ein schneller Projekterfolg erreicht werden kann.⁹ Hierbei ist es wichtig, dass aufgenommene Anforderungen dem Kunden zur Kontrolle vorgelegt werden, da nur so Fehler, Ungenauigkeiten oder Unstimmigkeiten vor der Umsetzung des Projektes entdeckt werden. Falls dieser

⁶ Vgl. Rupp, C. (2009), S. 16

⁷ Vgl. Grande, M. (2011), S. 11

⁸ Vgl. ebenda, S. 15

⁹ Vgl. ebenda, S. 11

Schritt nicht eingehalten wird, werden falsche Anforderungen in die Entwicklungsphase eines Projektes gegeben und oft werden die Fehler dann erst bei Übergabe des Projektes sichtbar. Da das Ergebnis in diesem Fall überarbeitet werden muss, investiert der Auftragnehmer zusätzliche Arbeitszeit in das Projekt, wodurch die Kosten für den Auftraggeber steigen.¹⁰

Die folgende Grafik zeigt, dass die meisten Fehler in Softwareentwicklungsprojekten in der Phase der Anforderungsaufnahme ihren Ursprung haben. Deutlich weniger entstehen in den Phasen Design, Programmierung und Test, und Wartung.

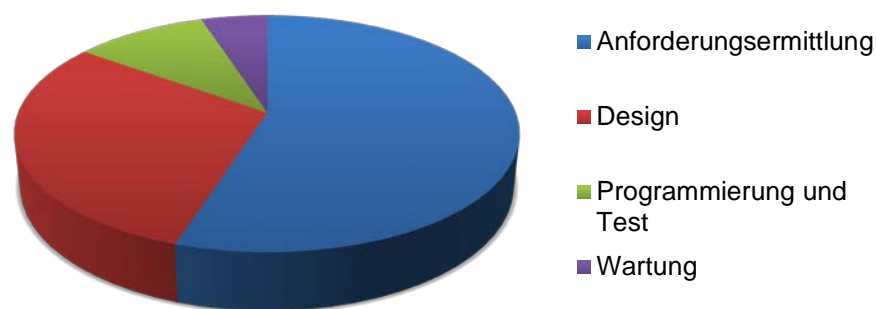


Abb. 1: Entstehungszeitpunkt von Fehlern in Softwareentwicklungsprojekten¹¹

Ein weiteres Ziel des Anforderungsmanagements ist die genaue Dokumentation der Anforderungen als rechtliche Grundlage. Es passiert, dass erst bei Übergabe des Projektes falsch oder ungenau umgesetzte Anforderungen entdeckt werden, oder der Auftraggeber sich sicher ist, eine Anforderung an die Software oder das Produkt gestellt zu haben, diese jedoch im Ergebnis nicht existiert. Der Auftragnehmer behauptet natürlich, von diesen fehlenden Anforderungen nichts gewusst zu haben. Um in diesem Fall eine rechtliche Grundlage zu haben, auf die sich beide Seiten berufen können, ist es wichtig, dass der Anforderungskatalog mit der Dokumentation aller Anforderungen über den gesamten Projektverlauf hinweg gepflegt und verwaltet wird. So müssen Anforderungen, die erst später im Projekt hinzukommen, hier aufgenommen werden und Anforderungen, die mit der ersten Implementierung der Lösung überflüssig werden, auch aus dem Anforderungskatalog entfernt werden.

¹⁰ Vgl. Rupp, C. (2008), S. 9

¹¹ Vgl. Grande, M. (2011), S. 15

2.1.3 Funktionen

Aus der Definition des Anforderungsmanagements, welches sich mit der Ermittlung, Dokumentation, Überprüfung und Verwaltung von Anforderungen beschäftigt, lassen sich die folgenden Funktionen ableiten. Sie sind Bestandteil der Werkzeuge, mit denen die Verwaltung der Anforderungen in Softwareprogrammen betrieben wird.¹²

Die Benennung einer Anforderung ist hier ein wesentlicher Bestandteil. Zudem sollte jede Anforderung in kurzen Worten beschrieben werden, um sicherzustellen, dass die Bedeutung einer Anforderung für Auftraggeber und –nehmer gleichermaßen geklärt ist. Es empfiehlt sich, die Anforderungen nach einem bestimmten Kriterium aus Gründen der Übersichtlichkeit zu gruppieren. Beispiel hierfür ist die Einordnung zu Projektabschnitten. Die Funktion Priorität sollte ebenfalls zur Verfügung stehen, um für die kommenden Projektphasen die Dringlichkeit des Designs oder der Implementierung festzuhalten. Zudem ist es ratsam, den Status einer Anforderung zu erfassen. Auf diese Weise lässt sich erkennen, ob die Anforderung sich in Bearbeitung befindet, bereits designt oder umgesetzt worden ist oder sich derzeit zur Validierung beim Kunden befindet. Weitere wichtige Funktionen sind die Darstellung von Abhängigkeiten zu anderen Anforderungen und die Abbildung von Strukturen des zu entwickelnden Systems. Im Falle eines Softwareentwicklungsprojektes bedingen sich viele der Anforderungen, sie haben eine Verbindung zueinander. Für den Fall, dass Änderungen an einer solchen Anforderung vorzunehmen sind, muss auf einen Blick klar sein, welche weiteren Anforderungen es auf eventuelle Änderungen zu überprüfen gilt. Die Abbildung der Struktur ist wichtig, um stets den Überblick über die Abhängigkeiten zu behalten und das Gesamtbild der zu entwickelten Lösung abrufbar zu haben. Ein mögliches Werkzeug hierfür ist der Einsatz von UML (Unified Modelling Language) -Diagrammen.¹³ Beispiele sind das Anwendungsfall-Diagramm¹⁴ oder das Aktivitätsdiagramm¹⁵.

2.2 Was ist Anforderungsmanagement-Software

Bevor einzelne Produkte und Funktionen genauer untersucht werden, muss zunächst der Begriff der Anforderungsmanagement-Software genauer betrachtet werden. In diesem Kapitel wird zunächst, basierend auf bestehenden Studien und Publikationen, eine Definition für Anforderungsmanagement-Software erarbeitet. Im Anschluss erfolgt eine Abgrenzung spezi-

¹² Vgl. o. A. (2014)

¹³ Vgl. o. A. (2014)

¹⁴ Ein Anwendungsfall-Diagramm stellt die Beziehung zwischen Akteuren und Anwendungsfällen dar.

¹⁵ Ein Aktivitätsdiagramm beschreibt Abläufe, welche mit Knoten und Kanten dargestellt werden.

alisierter Anforderungsmanagement-Tools, die in dieser Arbeit betrachtet werden, von allgemeinen Anwendungsprogrammen.

2.2.1 Definition und Ziele

Im Gegensatz zu anderen betrieblichen Anwendungsprogrammen, wie z.B. Enterprise Resource Planning (ERP) oder Customer Relationship Management (CRM)-Systemen, finden sich nur wenige einheitliche Definitionen für Anforderungsmanagement-Software. Während Anforderungsmanagement an sich Gegenstand unzähliger Publikationen ist, wird der Bereich spezialisierter, unterstützender Software nur selten genauer definiert und beleuchtet. Zusätzlich wird in vielen Fällen nicht konkret zwischen Anforderungsmanagement und Anforderungsmanagement-Software unterschieden. Im Folgenden wird daher zunächst eine allgemeingültige Definition von Anforderungsmanagement-Software erarbeitet.

Bei Anforderungsmanagement Software handelt es sich generell um Anwendungssoftware, d.h. sie dient der Unterstützung und Bearbeitung nützlicher oder gewünschter, nicht systemtechnischer Funktionalitäten.¹⁶ Die „gewünschte Funktionalität“ orientiert sich dabei an den im vorhergehenden Kapitel vorgestellten Bereichen des Anforderungsmanagements.¹⁷ Konkret handelt es sich also um die systematische Planung, Erstellung und Verwaltung von Anforderungen und die Unterstützung der damit zusammenhängenden Prozesse und Ziele.

Manche Autoren, wie z.B. Heller, liefern detailliertere Ziele und Funktionalitäten von Anforderungsmanagement -Software in ihren Definitionen. Dabei werden insbesondere genannt: Definition, Strukturierung, Diskussion, Priorisierung, Änderung und Kommunikation von Anforderungen.^{18,19} In neueren Veröffentlichungen werden zusätzlich die Unterstützung von Kollaboration und verteilter Entwicklung sowie die Visualisierung von Anforderungen als Ziele genannt.^{20,21}

Zusammenfassend ergibt sich folgende Definition für Anforderungsmanagement-Software:

Bei Anforderungsmanagement-Software handelt es sich um spezialisierte Anwendungsprogramme, die der systematischen Planung, Erstellung und Verwaltung von Anforderungen sowie der Unterstützung der damit zusammenhängenden Prozesse dienen. Konkret unter-

¹⁶ Engesser, H. et al (1993)

¹⁷ Vgl. dazu auch: Bittner, M. et al (2004)

¹⁸ Vgl. Bittner, M. et al (2004)

¹⁹ Vgl. Heller, G. (2013)

²⁰ Vgl. Bittner, M. et al (2004)

²¹ Vgl. Heller, G. (2013)

stützt Anforderungsmanagement-Software die kollaborative Definition, Strukturierung, Diskussion, Priorisierung, Änderung, Kommunikation und Visualisierung von Anforderungen.

2.2.2 Abgrenzung

Neben spezialisierter Anforderungsmanagement-Software, wie im vorhergehenden Kapitel definiert, werden oftmals auch andere Arten von Anwendungen im Rahmen des Anforderungsmanagements genutzt. In diesem Kapitel werden verschiedene nicht-spezialisierte Anwendungen, die für das Anforderungsmanagement genutzt werden können, vorgestellt und von spezialisierter Anforderungsmanagement-Software abgegrenzt.

Bei den nicht spezialisierten Anwendungen handelt es sich um Software, die nicht speziell zum Zwecke des Anforderungsmanagements entwickelt wurde. Zu den am meisten genutzten, nicht spezialisierten Anwendungen zählen: Office Anwendungen und Pakete (z.B. *Microsoft Office, Open Office, IBM Lotus*), Textverarbeitungs-Programme (z.B. *Microsoft Word, Open Office Writer, Google Docs*), Tabellenkalkulations-Programme (z.B. *Microsoft Excel, Libre Office Calc, Google Spreadsheets*), Präsentationssoftware (z.B. *Microsoft PowerPoint, Apple Keynote, Prezi*), Visual Modeling Software und Issue-Tracking-Tools (z.B. *HP OpenView, Bugzilla, VMware Service Manager*).²²

Obwohl alle diese Tools grundsätzlich für (Teile des) Anforderungsmanagement verwendet werden können, weisen sie gravierende Nachteile gegenüber spezialisierten Anwendungen auf, die sich vor allem bei größeren Projekten zeigen. Die größten Nachteile sind das Fehlen Anforderungsmanagement-spezifischer Funktionalitäten, die generell eingeschränkte Eignung einzelner Produkte, die nur durch die Kombination verschiedener Anwendungen erhöht werden kann, das Fehlen einer zentralen Speicherlösung sowie der Mangel an vordefinierten Templates und Prozessabfolgen. Durch diese Einschränkungen sind die oben genannten Anwendungen allenfalls als Einstiegslösung für das Anforderungsmanagement zu verstehen und sollten bei fortgeschrittenen Projekten durch spezialisierte Anforderungsmanagement-Software ersetzt werden.²³

Aufgrund ihrer Relevanz im Unternehmensumfeld werden in dieser Arbeit lediglich Anforderungsmanagement-spezifische Anwendungen analysiert und hinsichtlich ihrer Eignung für unterschiedliche Unternehmensgrößen bewertet.

²² Vgl. Birk, A. (2013)

²³ Vgl. Ebenda

3 Markt der Anforderungsmanagement-Software

In diesem Kapitel wird auf den Markt der Anforderungsmanagement-Software eingegangen. Nach einer Beschreibung der allgemeinen Situation am Markt unter Abschnitt 3.1 erfolgt die Betrachtung einzelner Produkte. Dem Ziel des Projektes entsprechend wird dabei zwischen kommerziellen Lösungen und Open Source Produkten unterschieden.

3.1 Volumen des Marktes

Mangelhaftes Anforderungsmanagement ist eine der häufigsten Ursachen für fehlerhafte Software. Die Visualisierung, Definition und das Management von Software-Anforderungen kann die Qualität von Software-Projekten und deren Ergebnissen deutlich verbessern. Die steigende Komplexität und Kritikalität von Software-Applikationen und -Systemen sowie der anhaltende wirtschaftliche Druck für die Relevanz, Anpassungsfähigkeit, Compliance und schnellere Markteinführungszeit von Software fordern zunehmend interaktive Anforderungsmanagement-Lösungen. Gefragt sind daher vor allem intuitiv bedienbare und benutzerfreundliche sowie anpassungsfähige und kollaborative Anforderungsmanagement-Softwarelösungen.²⁴

Zudem rückt der rasante Wachstum mobiler Applikationen und sozialer Netzwerke sowie die zunehmende Personalisierung von IT Anforderungsmanagement-Software immer stärker in den Fokus von Softwareentwicklung und Softwareeinsatz als kritischen Erfolgsfaktor. Im Jahre 2012 schätzte IDC den weltweiten Markt für Anforderungsmanagement-Software auf 314,6 Millionen USD; 13 % mehr im Vergleich zum Vorjahr. In Zeiten komplexer Beschaffungsstrukturen und permanenter Volatilität von IT und Entwicklung besteht für Unternehmen ein großer Bedarf an Orientierungshilfen für deren IT-Initiativen sowie unterstützende Tools, um Anforderungen zu visualisieren, zu definieren und automatisiert zu managen. Daher prognostiziert IDC auch für die folgenden Jahre bis 2016 eine jährliche Wachstumsrate (CAGR) von 13.5 % für das Marktvolumen von Anforderungsmanagement-Software, mit einem Gesamtumsatzpotenzial von bis zu 591 Millionen USD in 2017.²⁵

²⁴ Vgl. IDC (2013), p. 1 f.

²⁵ Vgl. IDC (2013), p. 1 f.

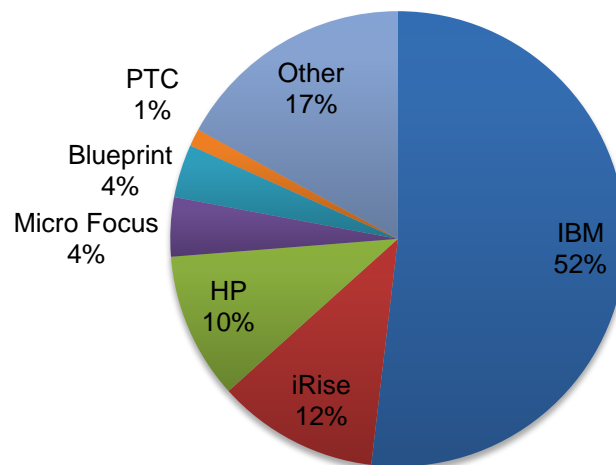


Abb. 2: Weltweiter Umsatz Anforderungsmanagement-Software 2012 nach Anbieter²⁶

Das kontinuierlich starke Wachstum ist vor allem auf die hohe Nachfrage und innovative Neuanbieter im Anforderungsmanagement-Softwaremarkt sowie neue Veröffentlichungen von bestehenden Anbietern zurückzuführen. Wie in Abbildung 2 dargestellt, ist IBM mit einem Marktanteil von 51,9 % zurzeit der dominanteste etablierte Anbieter. Der Fokus in IBMs Strategie zur Verteidigung seiner Marktführerschaft liegt vor allem auf anpassungsfähiger Anforderungsmanagement-Software zur Integration in eingebettete Systeme und agilen, IT-orientierten Umgebungen mit hoher Compliance.²⁷

Neue Marktteilnehmer im Anforderungsmanagement-Softwaremarkt, die unter anderem für das starke Marktwachstum verantwortlich sind, sind zum Beispiel iRise, HP und andere innovative kleinere Anbieter. iRise ist ein schnell wachsender Anbieter, der effektiv die Geschäftsfelder Anforderungsvisualisierung und Simulationsfähigkeiten bedient, welche aufgrund der Explosion an mobilen Applikationen eine starke Nachfrage genießen. iRise ist nach IBM der zweitgrößte Anbieter im Anforderungsmanagement-Softwaremarkt mit einem Marktanteil von 11,5 %, dicht gefolgt von HP mit einem Marktanteil von 10,4 %. HP begann 2009 Anforderungsmanagement-Software als eigenständiges, separates Produkt anzubieten und verzeichnet seitdem starke Wachstumszahlen. Des Weiteren positioniert sich HP vor allem durch sein Angebot an agilen Herangehensweisen für Anforderungsmanagement sowie Lösungen zur Qualitätsverbesserung von Anforderungen.²⁸

²⁶ Enthalten in: IDC (2013), p. 4

²⁷ Vgl. IDC (2013), p. 2 ff.

²⁸ Vgl. IDC (2013), p. 5 ff.

Dadurch, dass Business Analysten, Stakeholder und IT Teams zunehmend versuchen, gemeinsam Anforderungen an Software zu definieren, scheitern Projekte oftmals mangels effektiver Kommunikationsmittel und an prozess- und organisatorisch-bedingten Problemen. Schlechte Prozess- und Organisationsstrategien, sowie schwer bedienbare Tools stellen grundlegende Herausforderungen für Unternehmen bei der Verbesserung ihres Anforderungsmanagement dar. Daher ist zu erwarten, dass etablierte Anbieter wie IBM in Zukunft Partnerschaften mit Anbietern intuitiver Anforderungsmanagement-Softwarelösungen wie iRise, Blueprint, und Micro Focus eingehen werden, um deren Lösungen als Teil in umfassende Application-Management-Suiten zu integrieren. In diesem Sinne sind sowohl IBM als auch HP bereits in Partnerschaften mit iRise involviert.²⁹

Aus verschiedenen unabhängigen Marktstudien ist nicht zu erkennen, welche Marktteilnehmer auf welche Abnehmergruppe, wie z.B. kleine, mittlere oder große Unternehmen, abzielen; oder umgekehrt, welche Marktteilnehmer besonders beliebt bei bestimmten Abnehmergruppen sind. Das nächste Kapitel liefert neben der Vorstellung verfügbarer Produkte eine erste Abschätzung, in welche Zielgruppe diese einzuordnen sind.

3.2 Verfügbare Produkte

Dieses Kapitel dient der Übersicht der verfügbaren Produkte des Anforderungsmanagements. Es wird dabei zwischen kommerziell erwerblichen Produkten sowie Open Source Lösungen unterschieden. In beiden Abschnitten erfolgt neben der tabellarischen Benennung der Produkte auch eine Einführung zu den Funktionalitäten der Programme.

3.2.1 Kommerzielle Produkte

Im folgenden Abschnitt sollen die kommerziell verfügbaren Produkte des Anforderungsmanagements und ihre Funktionalitäten dargestellt werden. Häufig wird für das Anforderungsmanagement auch auf Office Lösungen wie MS Word oder Excel zurückgegriffen. Wie im ersten Abschnitt dargestellt, wird der Markt inzwischen jedoch von einer Reihe an professionellen Lösungen dominiert, die erweiterte Möglichkeiten zur Verfügung stellen³⁰. Die nachfolgende Übersicht dient als Grundlage für den anschließenden Kriterienkatalog, anhand dessen die Produkte in Zielgruppen kategorisiert werden sollen. Gleichzeitig sollen die derzeitigen Marktführer des jeweiligen Kundensegments im späteren Teil der Analyse näher

²⁹ Vgl. IDC (2013), p. 9 f.

³⁰ Vgl. Mühlbauer, S. / Versteegen, G. / Keller, F. (2006)

betrachtet werden, um einen Vergleich mit den erhältlichen Open-Source Produkten zu ermöglichen.

Ergebnisse der Literatur- und Internetrecherche

Die Literaturrecherche hat ergeben, dass eine Vielzahl verschiedener Werkzeuge für das Anforderungsmanagement existiert, die sich in ihren Funktionalitäten teilweise stark unterscheiden. Die Atlantic System Guild hat im Jahre 1995 ihre Sammlung „Volere“ vorgestellt, die Hilfsmittel und Materialien zu dem Thema Anforderungsanalyse umfasst. In diesem Rahmen wird seither auch eine Liste verfügbarer Werkzeuge zum Anforderungsmanagement gepflegt, die bis dato rund 80 Einträge enthält³¹. Andere Quellen benennen bis zu 50 verschiedene Softwareangebote³² oder spezialisieren sich auf einzelne Lösungen. Da die „Volere“ Sammlung als bewährter Standard in der Softwareentwicklung gilt, soll die Liste an dieser Stelle zu Rate gezogen werden. Aufgrund des Umfangs kann jedoch nicht auf alle dort genannten Vertreter kommerzieller Anforderungsmanagement-Software eingegangen werden. Aus diesem Grund wird unter Betrachtung anderer Quellen eine erste Auswahl getroffen, die Tabelle 1 zu entnehmen ist.

Die Auflistung liefert dabei Informationen zu den Anbietern sowie eine erste Übersicht zu den besonderen Fähigkeiten der jeweiligen Lösung in Bezug auf das Anforderungsmanagement sowie. Um die weiteren Ausführungen der Arbeit zu unterstützen, findet zudem eine Gruppierung anhand der ersten Einschätzung zur potenziellen Unternehmensgröße statt.^{33,34}

³¹ Vgl. Atlantic Systems Guild Ltd. (2014)

³² Vgl. Alexander, I. (2014)

³³ Vgl. Atlantic Systems Guild Ltd. (2014)

³⁴ Vgl. Alexander, I. (2014)

Name / Anbieter	Fähigkeiten	Tendenz Unternehmensgröße
Quality Center 9.2 <i>Hewlett-Packard</i>	<ul style="list-style-type: none"> - Anforderungsmanagement als Modul innerhalb des vollständigen Anwendungs-Lebenszyklus - Import / Export Funktionen aus MS Excel und Word - Kategorisierung der Anforderungen nach Bedeutung und Unternehmensrisiko - Fähigkeit, Auswirkungen von Änderungen an den Anforderungen zu analysieren 	Große Unternehmen
Rational DOORS <i>IBM</i>	<ul style="list-style-type: none"> - Kollaboratives System für die Einbindung aller Stakeholder eines Projektes - Integrierte Funktion für die Antragsstellung von Änderungen an einer Anforderung - Diskussionsforen - Skalierbare Lösung zur flexiblen Anpassung an die Unternehmensanforderungen - Keine Software-Installation nötig, da Nutzung über Internet-Browser 	Große Unternehmen
Accept 360° <i>Accept Software, Inc.</i>	<ul style="list-style-type: none"> - Erstellung von Use Cases - Möglichkeit, eine Datenbank mit Projektmodellen zu erstellen - Online Support Center für den Wissensaustausch mit Kunden und Partnern - Bildet kompletten Lebenszyklus eines Projektes ab und erlaubt die Verknüpfung mit der Finanz-, Strategie- und Ressourcenplanung 	Große – mittlere Unternehmen
Accompa <i>Accompa, Inc.</i>	<ul style="list-style-type: none"> - Bedarfsabhängige Nutzung (SaaS) - Online Kollaborationsmöglichkeiten durch Foren - Individualisierbare Formulare zur Aufnahme neuer Anforderungen von extern oder intern - Nachverfolgen der Projekthistorie 	Große – mittlere Unternehmen

Integrity <i>PTC</i>	<ul style="list-style-type: none"> - Automatische Kommunikation an die Stakeholder bei Anforderungsänderungen - Übertragung bestehender Anforderungen in andere Projekte - Verlinkung zwischen parallelen Projekten 	Große – mittlere Unternehmen
ReMa <i>Accord Software and Systems Pvt. Ltd.</i>	<ul style="list-style-type: none"> - Zugrifflevel auf das System sind individuell anpassbar - Kommentarfunktion für einzelne Anforderungen - Vergleichsfunktion für verschiedene Statuslevel - Kontrolle über Änderungen durch Antragsstellung - Import / Export von MS Word, Excel und CSV Dokumenten - Automatische Berichterstattung 	Große – mittlere Unternehmen
Avenqo PEP <i>Avenqo GmbH</i>	<ul style="list-style-type: none"> - Abbilden des kompletten Anforderungsmanagements-Zyklus von der Bestandsaufnahme, über die Historie bis hin zu den Projektfortschritten - Flexible Einstellungen, was Statusberichte, Prioritäten und Benutzerzugänge angeht 	Mittlere Unternehmen
Blueprint <i>Blueprint Software Systems Inc.</i>	<ul style="list-style-type: none"> - Cloud-basierte Lösungen - Kollaborationsmöglichkeiten über Komponenten sozialer Netzwerke - Nachverfolgen der Historie 	Mittlere Unternehmen
Caliber <i>Borland</i>	<ul style="list-style-type: none"> - Zentrale Datenbank - Anpassungsmöglichkeiten über die GUI oder Wizards - Nachvollziehen des kompletten Lebenszyklus - Zugang zu Glossaren zum Standardisieren und Definieren von Terminologien 	Mittlere Unternehmen
CASE Spec <i>Goda Software, Inc.</i>	<ul style="list-style-type: none"> - Schnittstellen zu anderen Werkzeugen und automatisierte Import / Export Funktionen - Diagrammfunktionen zur Erstellung von Use Cases, Aktivitäten, Klassen, Objekten und Komponenten - Erstellung von automatischen Dokumenten und Statusberichten - Skalierbare Client-Server-Architektur 	Mittlere Unternehmen

<p>Cognition Cockpit <i>Cognition Corporation</i></p>	<ul style="list-style-type: none"> - Nachvollziehen und Vergleichen der Historie - Umfassende Import / Export Funktionen von Anforderungen - Möglichkeit der automatischen Generierung von Statusberichten 	<p>Mittlere Unternehmen</p>
<p>Foresight <i>Foresight Systems Inc.</i></p>	<ul style="list-style-type: none"> - Grafischer Editor zur Erstellung von Datenflussdiagrammen und Mini-Specs - Schnittstelle zu Simulationswerkzeugen - Einbettung des Anforderungsmanagements durch eine Verlinkung zu anderen Lösungen wie IBM DOORS 	<p>Mittlere Unternehmen</p>
<p>OneDesk <i>OneDesk Inc.</i></p>	<ul style="list-style-type: none"> - Bedarfsabhängige Nutzung für kleine Unternehmen - Integrierte Feedbackschleife für den Kunden - Komponenten für den sozialen Austausch in Form von Foren, Blogs, Chats und Abstimmungen - Integrierte Finanzplanung und Analyse - Analyse der Kundenzufriedenheit in Bezug auf die Erfüllung der Anforderungen 	<p>Mittlere Unternehmen</p>
<p>RaQuest <i>SparxSystems Japan</i></p>	<ul style="list-style-type: none"> - Erlaubt den Export und Druck der Anforderungslisten - Visuelle Darstellung der Verknüpfung von Anforderungen in Matrizen - Erstellung von Use Cases anhand der Anforderungen - Verantwortliche können bestimmten Anforderungen zugeordnet werden 	<p>Mittlere Unternehmen</p>
<p>RM <i>Polarion Software</i></p>	<ul style="list-style-type: none"> - Ermöglicht die Kommentierung von Anforderungen - Datenimport/-export möglich - Enthält eine Ranking-Funktion zur Bewertung von aufgenommenen Anforderungen - Bietet die Möglichkeit zur Erstellung zusammenfassender Reports - Erstellung von To-Do-Listen 	<p>Mittlere Unternehmen</p>

Visure Requi- rements <i>Visure Soluti- ons</i>	<ul style="list-style-type: none"> - Visuelle Darstellungsmöglichkeiten für die Zusammenhänge zwischen den Anforderungen - Konfigurationsmöglichkeiten für die verschiedenen Benutzerrollen - Wiederverwendbarkeit von Anforderungen - Integrierte Kollaborationsplattform 	Mittlere Unternehmen
Enterprise Architect <i>Sparx Systems Ltd.</i>	<ul style="list-style-type: none"> - Einbettung der Anforderungen in Modelle der Projektplanung - Umfassende grafische Darstellungsmöglichkeiten 	Mittlere – kleine Unternehmen
Jama Contour <i>Jama Software, Inc.</i>	<ul style="list-style-type: none"> - Tracken der Anforderungshistorie - Wiederverwendbarkeit definierter Anforderungen - Import / Export Möglichkeiten für MS Word und Excel - Webbasierte Anwendungen für die schnelle und weltweite Nutzung der Lösung 	Mittlere – kleine Unternehmen
Work-space.com <i>Work-space.com</i>	<ul style="list-style-type: none"> - Zentrale Datenbank für den kompletten Anwendungszyklus - Echtzeitanalyse und -berichterstattung - Kollaborationsmöglichkeiten über Foren und Kommentarfunktion 	Mittlere – kleine Unternehmen
LiteRM <i>ClearSpecs Enterprises</i>	<ul style="list-style-type: none"> - Einfache Bedienbarkeit - Große Datenbank an individualisierbaren Vorlagen - Kategorisierung nach verschiedenen Typen, Benutzerrollen und Status - Import / Export Funktionen 	Kleine Unternehmen

Tabelle 1: Übersicht kommerziell verfügbarer Anforderungsmanagement Software

Eine Auswahl der aufgelisteten Software Lösungen wird die Grundlage für die Entwicklung des Kriterienkataloges in Kapitel 4 bilden. Zunächst sollen im nachfolgenden Abschnitt gleichermaßen die Open Source Produkte vorgestellt werden.

3.2.2 Open Source Produkte

Nachdem im vorangegangenen Schritt kommerzielle Produkte zur effektiven Durchführung von Anforderungsmanagement vorgestellt wurden, sollen im nächsten Teil Open Source Produkte identifiziert und vorgestellt werden, welche ebenfalls für das Management von Anforderungen genutzt werden können.

Insgesamt ist das Angebot an Open Source-Lösungen viel geringer als das an kommerziellen Produkten. Belegen lässt sich dies beispielsweise durch eine nähere Betrachtung der, im vorangegangenen Abschnitt vorgestellten, Volere-Sammlung. Unter den insgesamt 79 Tools, welche in dieser Sammlung beschrieben werden, finden sich lediglich 3 Open-Source Lösungen.³⁵ Dies zeigt bereits, dass der Fokus der Hersteller klar auf kommerziellen Produkten liegt, deren Code nicht offen zur Verfügung gestellt wird.

Zudem muss bei der Suche nach Open Source Lösungen zunächst jeweils geprüft werden, wie gut die jeweilige Anwendung wirklich ist. Zwar bieten einige Entwickler ihre Anforderungsmanagement-Tools und deren Code im Internet an, jedoch sind nicht alle dieser Tools ausreichend entwickelt, dass diese ohne weiteres eingesetzt werden könnten. Daher muss zunächst eine Vorselektion erfolgen, um nur diejenigen Produkte auszuwählen, welche ohne weiteren Entwicklungsaufwand sofort für ihre eigentliche Aufgabe genutzt werden können.

Vor einer näheren Betrachtung der hier ausgewählten Produkte sollte man sich jedoch der Vor- und Nachteile von Open Source Lösungen bewusst sein.

So gibt es zunächst zwei Argumente, welche stark für einen Einsatz von Open Source Lösungen sprechen. Deren größter Vorteil liegt darin, dass diese Produkte kostenfrei eingesetzt werden können und keine Lizenzgebühren fällig werden, welche je nach Hersteller und Produkt erheblich sein können. Der zweite große Vorteil von Open Source Lösungen liegt in der Zugänglichkeit des Quellcodes. Dadurch kann die Software ohne größeren Aufwand an die Anforderungen des eigenen Unternehmens angepasst werden.

Dennoch sollte man sich auch der Nachteile bewusst sein, welche ein Einsatz von Open Source Lösungen mit sich bringen kann. Da diese nicht kommerziell entwickelt werden, können oft nur geringere Aufwände aufgebracht werden, um diese Tools zu verbessern. Die Ressourcen- und Mittelknappheit schlägt sich in der Folge häufig in den Lösungen nieder, welche daher nur über einen eingeschränkten Funktionsumfang verfügen. Zudem sollte beim Einsatz von Open Source Lösungen darauf geachtet werden, dass es eine aktive Entwicklercommunity wie zum Beispiel Firefox gibt, welche sich kontinuierlich mit dem Produkt befassen. Wird das Tool nicht mehr weiter entwickelt, was oft bei Nischenprodukten der Fall ist,

³⁵ <http://volere.co.uk/tools.htm>

kann dies dazu führen, dass Sicherheitslücken nicht geschlossen werden und auch keinerlei Funktionsverbesserungen vorgenommen werden. Nur wenn diese Nachteile von den oben genannten Vorteilen überwogen werden, sollte ein Einsatz von Open Source Lösungen in Betracht gezogen werden.³⁶

Im Folgenden werden Open-Source Lösungen vorgestellt, welche über einen ausreichenden Funktionsumfang verfügen, sodass damit die grundlegenden Aufgaben des Anforderungsmanagements erfüllt werden können. Zudem enthalten einige der Produkte interessante Zusatzfunktionen, welche hier ebenfalls aufgeführt werden.

Name / Anbieter	Fähigkeiten	Tendenz Unternehmensgröße
aNimble <i>ideaStub</i>	<ul style="list-style-type: none"> - Nachfolger von OSRMT - Unterstützt verschiedene Entwicklungsmodelle, wie Wasserfall, Agil oder Scrum - Bietet personalisierte Dashboards - Enthält zusätzliche Funktionen für Projekt Management, Portfolio und Test Management - Ermöglicht die Implementierung eigener Erweiterungen - Unterstützt sowohl Microsoft SQL Server als auch MySQL <p>Verfügt über eine web-basierte GUI</p>	Mittlere Unternehmen
ProR <i>Formal Mind GmbH</i>	<ul style="list-style-type: none"> - Ist auf Erweiterbarkeit ausgelegt - Teil des Eclipse Requirements Modeling Frameworks - Verfügt über eine anpassbare und mächtige Benutzeroberfläche - Kann mit anderen Eclipse-Tools verknüpft werden - Kommerzieller Support ist verfügbar - Website mit Dokumentation und ausführlichen Tutorials verfügbar 	Mittlere Unternehmen

³⁶ <http://rmblog.accompa.com/2012/04/free-open-source-requirements-management-tool/>

<p>SHORE Version 2.0 SD&M AG</p>	<ul style="list-style-type: none"> - Ist ein Hypertext-Projektarchiv - Fasst Inhalte aus verschiedenen Tools, Dokumenten, etc. zusammen und bietet einen Überblick - Zusammenfassung kann mithilfe von Web-Browsern oder Queries durchsucht werden - Dokumente, Objekte und Beziehungen können mithilfe einer Tool-eigenen Sprache spezifiziert werden 	<p>Mittlere bis kleine Unternehmen</p>
<p>OSRMT <i>ideaStub</i></p>	<ul style="list-style-type: none"> - Unabhängiges Tool für Entwickler, Tester und Analysten - Funktioniert im Rahmen eines "Client-Server"-Modells - Enthält Funktionen zum Management von Anforderungen, Design und Tests - Sehr gute User-Bewertungen auf Sourceforge.net <p>Projekt wird nicht mehr fortgeführt, letzte Version wurde 2007 veröffentlicht</p>	<p>Kleine Unternehmen</p>
<p>rmtoo <i>flonatel GmbH</i></p>	<ul style="list-style-type: none"> - Konzipiert für Software-Entwickler - Keine GUI vorhanden - Geschrieben in Python - Abhängigkeiten können dargestellt werden - Output kann in Form von LaTeX, PDF oder HTML Dokumenten erfolgen - Unterstützt Aufwandsschätzungen für agile Software-Entwicklungsprojekte - Letztes Update wurde 2012 durchgeführt 	<p>Kleine Unternehmen</p>
<p>ReqT <i>Lund University</i></p>	<ul style="list-style-type: none"> - Basiert auf einer eigenen Sprache (DSL) innerhalb der plattformunabhängigen Sprache Scala - Kann aus der Kommandozeile heraus bedient werden - Ausführung ebenfalls in IDEs mit Scala-Unterstützung (z.B. Eclipse) möglich - Anforderungs-Dokumente zur Veröffentlichung im Internet können automatisch generiert werden 	<p>Kleine Unternehmen</p>

Unicase <i>TU München</i>	<ul style="list-style-type: none"> - Ermöglicht eine vertikale und horizontale Nachvollziehbarkeit sämtlicher Änderungen - Beinhaltet ein integriertes Software Configuration Management System - Bietet Projekt-Management-Funktionen - Basiert auf der Eclipse-Plattform - Ermöglicht eine automatische Nachverfolgung von Änderungen 	Kleine Unternehmen
------------------------------	--	--------------------

Tabelle 2: Übersicht der Open Source Lösungen des Anforderungsmanagements

Auf Grundlage der in Kapitel 3.2 und 3.3 aufgelisteten Software Produkte wird im nachfolgenden Kapitel der Kriterienkatalog entwickelt.

4 Aufbau eines Kriterienkataloges

Dieses Kapitel dient der Entwicklung des Fragenkataloges und Beurteilung der einzelnen Produkte. Zu diesem Zweck wird zunächst auf die Vorgehensweise bei der Erstellung des Fragenkataloges eingegangen. Dementsprechend werden in dem Abschnitt 4.1 die Ergebnisse in Form der einzelnen Kriterien beschrieben. Im Anschluss erfolgt in Kapitel 4.2 die Zuordnung der Relevanz der einzelnen Fähigkeiten in Bezug auf die Unternehmensgröße. Dieses Unterkapitel dient als Basis für die letztliche Einordnung der Produkte. Eine Reihe ausgewählter Software-Lösungen wird dem Kriterienkatalog unterzogen und anhand der Resultate aus 4.2 schließlich auf die Eignung für die Unternehmensgrößen klein, mittel beziehungsweise groß hin geprüft.

4.1 Konsolidierung des Kriterienkataloges

Im ersten Abschnitt des Kapitels soll das Vorgehen bei der Definition der relevanten Funktionalitäten sowie der Entwicklung des zugehörigen Fragenkataloges dargestellt werden. Im zweiten Abschnitt wird das Ergebnis, der Kriterienkatalog, präsentiert.

4.1.1 Vorgehensweise

Der Kriterienkatalog ist das Resultat eingehender Produktanalysen. Das Vorgehen bei der Erstellung des Fragenkatalogs ist dabei zunächst in Abbildung 3 zusammenfassend dargestellt. Die Schritte 1 bis 3 werden nachfolgend näher erläutert, Schritt 4 ist Teil der Erläuterungen aus Kapitel 4.2.



Abb. 3: Schematische Vorgehensweise bei der Erstellung des Kriterienkataloges

Schritt 1: Auswahl relevanter Anwendungen

Die Basis der Matrix bildet dabei die Betrachtung einzelner Anforderungsmanagement-Software Produkte, die zum einen kommerziell und zum anderen als Open Source Lösungen verfügbar sind. Die entsprechenden Produkte wurden dabei unter 3.2 bereits einführend vorgestellt. Aufgrund des Umfangs der kommerziellen Software-Lösungen wurde die Anzahl untersuchter Produkte jedoch auf zehn beschränkt: Accompa, HP Quality Center, IBM Rational DOORS, LiteRM, ReMa, OneDesk, Case Spec, RM, PTC Integrity und Caliber. Dieser Umfang ist zum einen groß genug, als dass eine annähernd objektive Analyse möglich ist. Dementsprechend wurde bei der Auswahl auch darauf geachtet, möglichst unterschiedliche Produkte auszuwählen und gleichzeitig die in Kapitel 3.1 vorgestellten Marktführer mit einzuschließen. Gleichzeitig beschränkt sich die Zahl an Software-Lösungen allerdings auch auf einen überschaubaren Rahmen, der es zulässt, die Produkte nicht nur oberflächlich, sondern tiefergehend zu betrachten. Aufgrund des geringeren Umfangs konnten die Open Source Produkte in ihrer Gesamtheit analysiert werden. Hier hat sich zum Teil aber bereits schnell herausgestellt, dass die angebotenen Lösungen nicht mehr weiter entwickelt werden und dementsprechend keine ernstzunehmenden Alternativen für kommerzielle Software darstellen.

Schritt 2: Analyse der Funktionen

Die aufgelisteten Produkte wurden zunächst in Hinblick auf ihre Funktionalitäten geprüft. Als Quelle dienten vornehmlich die im Internet vorhandenen „Data Sheets“ und Präsentationen der Unternehmen, es wurde jedoch auch auf das Mittel der persönlichen Kundenberatung zurückgegriffen. Es wurde jedoch schnell deutlich, dass Informationen, die nicht öffentlich im Internet zugänglich gemacht werden, nur schwer im persönlichen Gespräch ermittelt werden können. Insbesondere der Preis konnte daher nicht für alle kommerziellen Produkte ausgewiesen werden.

Nichtsdestotrotz ergab sich bei der Recherche eine Vielzahl verschiedener Funktionalitäten, die je nach Umfang des Produktes stark variierten. Gleichzeitig zeigte sich, dass sich ein Großteil der Funktionalitäten über die Produkte der verschiedenen Anbieter hin ähnelt und scheinbar eine Grundvoraussetzung für Anforderungsmanagement-Software darstellt. So zählt beispielsweise die Möglichkeit, Anforderungen anzulegen, zu bearbeiten und gegebenenfalls zu löschen zu den Fähigkeiten, die ein Produkt in jedem Falle aufweisen muss, um den Zweck des Anforderungsmanagements zu erfüllen.

Schritt 3: Konsolidierung des Kriterienkataloges

Anhand der herausgearbeiteten Fähigkeiten konnte ein Kriterienkatalog entwickelt werden. Um diesen möglichst übersichtlich zu gestalten, wurden die einzelnen Aspekte in neun Kategorien eingeordnet, die unter anderem die allgemeinen Fähigkeiten, das Anforderungsma-

nagement sowie die Integrationsmöglichkeiten abbilden. Es wurde davon abgesehen, spezifische Open Source Funktionalitäten in einer eigenen Kategorie abzugrenzen. Die entsprechenden Features, die gegebenenfalls bei der Betrachtung der sieben Open Source Lösungen erarbeitet werden konnten, wurden gleichermaßen diesen Kategorien zugeordnet, um eine Vergleichbarkeit der Produkte sicherzustellen. Es gibt lediglich einen Unterpunkt für Open Source Spezifika, der in die Kategorie „Allgemeines“ integriert wurde.

Die Anordnung der Kategorien ergibt sich aus ihren thematischen Schwerpunkten und der Wichtigkeit für den Zweck des Anforderungsmanagements. Ausgehend von den allgemeinen Attributen wird im Anschluss der Kern der Software, die Funktionen des Anforderungsmanagements, untersucht. In engem Zusammenhang damit steht das Änderungsmanagement. Sowohl Collaboration als auch Aufbau und Navigation und Administration beschreiben Bereiche, die essentiell sind für den Einsatz der Software. Der Zugang und Integration sowie der Datenaustausch befassen sich weniger mit den Fähigkeiten für ein effektives Anforderungsmanagement als den Umgang mit der Software. Daher folgen diese Kategorien nach. Den Abschluss bildet der Aspekt „Analyse“, da dieser nicht zwangsläufig zum Anforderungsmanagement, sondern eher zum Projektmanagement zählt.

Die Reihenfolge der einzelnen Features innerhalb der Kriterien ergibt sich schließlich aus ihrer Gewichtung nach Basic, Advanced und Optional, wie es in Kapitel 4.2 erläutert wird. Das bedeutet, die immer stärker differenzierenden Merkmale erscheinen am Ende der Liste. Gibt es in einer Kategorie Unterpunkte, so werden diese separat nach dem gleichen Schema aufgebaut.

4.1.2 Ergebnis: Der Kriterienkatalog

Im Folgenden werden die einzelnen Kategorien des so erarbeiteten Kriterienkataloges näher betrachtet. Abbildung 4 liefert dazu zunächst eine Übersicht über alle ermittelten Kategorien.



Abb. 4: Ermittelte Kategorien des Kriterienkataloges

In der folgenden näheren Betrachtung wird davon abgesehen, auf jedes Feature der Kategorien im Detail einzugehen, stattdessen sollen die übergeordneten Fähigkeiten näher beschrieben werden. Dementsprechend wird die Kategorie zunächst kurz vorgestellt, bevor im Anschluss der zugehörige Ausschnitt der Matrix wiedergegeben wird.

Kategorie 1: Allgemeines

An dieser Stelle werden alle Aspekte zusammengefasst, die zwar die Software als solche, jedoch nicht die Fähigkeiten für das Anforderungsmanagement beschreiben. Es finden sich Punkte, die die Wettbewerbsfähigkeit der Lösung bedingen, insbesondere der Preis sowie die gewährleisteten Support-Funktionen. Dieser Faktor ist besonders bei den Open Source Produkten ausschlaggebend, da kostenlos angebotene Produkte im Allgemeinen nicht die gleichen Mittel der Anwendungshilfe für den Endverbraucher zur Verfügung stellen können wie kommerzielle. Gleichmaßen finden sich in dieser Kategorie die Open Source Spezifika in Form der Entwickler-Community oder des Vorhanden-Seins eines Bugtrackers. Die Angabe zum letzten Update verdeutlicht, inwiefern die Software noch weiter entwickelt wird.

Außerdem wird auf die Skalierbarkeit der Lösung eingegangen. Hier wird betrachtet, ob der Anbieter dem Kunden die Möglichkeit gibt, die Nutzung der Software dem persönlichen Bedarf preislich anzupassen. Auch wird untersucht, ob für das Produkt eine Testversion verfügbar ist beziehungsweise ob das Produkt lediglich in einer übergeordneten Software-Lösung erhältlich ist oder auch als eigenständiges Produkt.

Allgemeines
Version
Preis
Plattform
Open-Source Spezifika
Entwickler-Community
Letztes Update
Bugtracker
Support
Web Hilfe
Telefonischer Support
Verfügbarkeit von Trainings (Web-Trainings)
Verfügbarkeit von Trainings (onsite)
Skalierbarkeit
Testversion verfügbar
Verfügbarkeit als eigenständiges Produkt

Tabelle 3: Kategorie „Allgemeines“ des Kriterienkataloges

Kategorie 2: Anforderungsmanagement

Diese Kategorie erfüllt den Zweck, auf die Spezifika von Anforderungsmanagement-Software einzugehen. In diesem Sinne werden die Fähigkeiten behandelt, die eine Lösung aufweisen muss beziehungsweise kann, um das Anforderungsmanagement eines Unternehmens zu ermöglichen. Da das Anlegen von Anforderungen als Grundvoraussetzung dient, damit eine Software das Anforderungsmanagement unterstützen kann, wurde darauf verzichtet, diese Fähigkeit gesondert im Kriterienkatalog aufzuführen. Stattdessen wurde auf darüberhinausgehende Differenzierungsmerkmale eingegangen, die das Anlegen von Anforderungen betreffen. Zunächst wird daher abgefragt, in welcher Form Anforderungen angelegt oder bearbeitet werden können. Dazu zählt insbesondere, ob bereits angelegt Anforderungsprofile wiederverwendet werden können und es definierte Richtlinien gibt, anhand derer die Anforderungen angelegt werden müssen.

Gleichzeitig stellt der Umfang der Anforderungshistorie einen wichtigen Punkt dar: Können Änderungen an Anforderungen über den vollständigen Projektzyklus verfolgt werden? Besteht die Möglichkeit, verschiedene Versionen einer Anforderung miteinander zu vergleichen?

Die Kategorie beinhaltet zudem, inwieweit Anforderungen verschiedener Projekte parallel bearbeitet werden können und ob es die Möglichkeit des „Baselinings“ gibt, das heißt, ob der Status Quo bei Projektstart abgebildet werden kann. Außerdem stellt sich die Frage nach

verschiedenen Projektmanagement-Vorgehensweisen, kann das Anforderungsmanagement beispielsweise auch ein agiles Vorgehen abbilden.

Darüber hinaus sollten Anforderungen transparent in den Projektverlauf mit eingebunden werden. Daher spielt die Nachverfolgbarkeit der Attribute eine wichtige Rolle. Es stellt sich die Frage, ob Anforderungen projektintern beziehungsweise auch projektübergreifend miteinander verlinkt werden können. Auch wird untersucht, ob die Software-Lösung eigenständig Fehler in Form inkonsistenter Anforderungen aufdeckt, zum Beispiel, wenn eine Anforderung nicht mit dem Projekt verlinkt ist.

Zudem wird dargestellt, welche Funktionen es für die Ideensammlung beim Anforderungsmanagement gibt, sprich, ob Vorschläge online oder per Mail eingereicht werden können sowie die Einbindung eines Glossars.

Anforderungsmanagement
Anlegen, Ändern und Löschen von Anforderungen
Definierbare Richtlinien für Anforderungs-Attribute
Bewertung und Klassifikation von Anforderungen
Wiederverwendbarkeit angelegter Anforderungsprofile (Templates)
Anhang unterstützender Dokumente
Anforderungshistorie
Integriertes System zur Versionskontrolle
Vollständige Abbildung der Versionshistorie
Gleichzeitige Verwaltung von Anforderungen verschiedener Projekte
Agiles Projektmanagement ist abbildbar
Aufnahme des Status Quo („Baselining“)
Nachverfolgbarkeit von Anforderungen
Projektübergreifende Verlinkungen
Projektinterne Verlinkungen
Verlinkung von Programmcode mit korrespondierenden Anforderungen
Kennzeichnung inkonsistenter Anforderungen (Orphans)
Integrierte Funktionen für das Sammeln von Anforderungen und Daten
Sammeln von Anforderungen via E-Mail
Sammeln von Anforderungen via Web
Einbindung eines Glossars

Tabelle 4: Kategorie „Anforderungsmanagement“ des Kriterienkataloges

Kategorie 3: Änderungsmanagement

Das Change-Management spielt bei Projektverläufen eine entscheidende Einflussgröße, da sich an dieser Stelle entscheidet, in welcher Form Änderungen bewilligt und durchgeführt werden. Das Änderungsmanagement ist eng mit dem Anforderungsmanagement verknüpft und wird daher an dieser Stelle ausgewiesen.

Kern der Kategorie ist die Frage, ob ein Änderungsmanagement für Anforderungen integriert ist. Können entsprechende Anfragen zur Änderung einer Anforderung erstellt und bewertet werden? Außerdem ist es wichtig, zu erfragen, wie die Freigabe solcher Änderungen erfolgt.

Änderungsmanagement
Integriertes Änderungsmanagement
Erstellen, Bewerten und Freigeben von Änderungsanfragen

Tabelle 5: Kategorie „Änderungsmanagement“ des Kriterienkataloges

Kategorie 4: Collaboration

Die Möglichkeiten der internen Kollaboration sind sehr entscheidend für Unternehmen, in denen mehrere Mitarbeiter am Anforderungskatalog arbeiten. Es soll daher dargestellt werden, inwiefern die Mitarbeiter in einen Austausch über die Anforderungen treten können. Es stellt sich die Frage, ob es Möglichkeiten gibt, Anforderungen zu kommentieren oder untereinander zu teilen. Besitzt die Softwarelösung eine integrierte Community, sprich, gibt es Social Media ähnliche Funktionen wie Foren oder Diskussionsrunden? Außerdem soll betrachtet werden, inwiefern Neuerungen an die beteiligten Mitarbeiter und Stakeholder übermittelt werden, zum Beispiel in Form benutzerdefinierter E-Mail Nachrichten.

Collaboration
Kommentier Funktion
Integrierte Community
Teilen von Anforderungen
(Teil-)Automatisierte Kommunikation mit Kunden und Shareholdern
Benutzerdefinierte E-Mail Benachrichtigungen

Tabelle 6: Kategorie „Collaboration“ des Kriterienkataloges

Kategorie 5: Aufbau und Navigation

In der fünften Kategorie soll die Anforderungsmanagement-Software in Hinblick auf ihre Benutzerfreundlichkeit beziehungsweise -führung beleuchtet werden. Es wird untersucht, ob es einfache Such- und Filterfunktionen gibt und welche Ansichten von der Software unterstützt

werden, beispielsweise in tabellarischer Form oder als Dokument. Gleichzeitig wird ermittelt, in welcher Art die Anforderungen visualisiert werden können.

Aufbau und Navigation
Suchfunktion
Filterfunktion
Ansichten
Tabellen Ansicht
Dokumenten Ansicht
Baum Ansicht
Roadmap Ansicht
Visualisierung mittels Use Cases

Tabelle 7: Kategorie „Aufbau und Navigation“ des Kriterienkataloges

Kategorie 6: Administration

Dieser Unterabschnitt stellt die Fähigkeiten in Bezug auf die Administration dar. Es wird erfragt, inwiefern die Benutzer durch einen Administrator verwaltet werden können und welche Möglichkeiten dieser Administrator besitzt. So stellt sich die Frage nach der Rechteverwaltung und der Individualisierbarkeit der Ansichten.

Administration
Administrator Ansicht mit Informationen zu Systemstatus, Projekten und Nutzern
Benutzerverwaltung
Integrierte Benutzer- und Rechteverwaltung
Individuelle Anpassung von Nutzeransichten
Rollenbasierte Rechteverwaltung

Tabelle 8: Kategorie „Administration“ des Kriterienkataloges

Kategorie 7: Zugang und Integration

Kategorie 7 beschäftigt sich erneut weniger mit den Fähigkeiten für das Anforderungsmanagement, sondern eher mit den Software Spezifika. So stellt sich die Frage nach den Zugriffsmöglichkeiten auf das Programm. Als Zugriffsmöglichkeiten wurden als Standard der Web Zugang identifiziert, während Desktop und mobiler Zugang erweiterte Funktionen darstellen.

Auch spielt die Integration eines Test- und Defect-Managements eine Rolle. Dieses gehört zwar nicht zum Anforderungsmanagement im eigentlichen Sinne, ist aber eng mit diesem

verknüpft. Durch die Bereitstellung einer öffentlichen API ergeben sich Möglichkeiten in der Erweiterung der Software.

Gleichzeitig wurden die unterstützten Anmeldeverfahren betrachtet, wie beispielsweise die Security Assertion Markup Language (SAML)³⁷ als Framework für den Austausch von sicherheitsbezogenen Informationen zur Authentifizierung. Als weitere Anmeldeverfahren wurden OAUTH2 für die standardisierte, sichere API-Autorisierung³⁸ sowie Google Sign-On, das heißt der Zugriff über einen Google-Nutzeraccount, identifiziert. Darüber hinaus wird die Möglichkeit des Single-Sign Ons betrachtet, das die einmalige Autorisierung eines Nutzers für alle Rechner und Dienste ermöglicht.

Schließlich ist es zudem wichtig, die Anbindungsmöglichkeiten an andere Produkte zu betrachten wie die Cloud Computing Anbieter Salesforce und Zendesk, das Werkzeug zur Fehlerverwaltung Jira oder den Team Foundation Server (TFS) von Microsoft für die kollaborative Softwareentwicklung.

Zugang und Integration
Zugriffsmöglichkeiten
Web Zugang
Desktop Zugang
Mobiler Zugang
Integriertes Test- und Defect Management
Bereitstellung einer öffentlichen API
Unterstützte Anmeldeverfahren
Single-Sign On
SAML
OAUTH2
Google Sign-On
Integrierte Anbindung anderer Produkte
Salesforce
Zendesk
Jira
TFS

Tabelle 9: Kategorie „Zugang und Integration“ des Kriterienkataloges

³⁷ Vgl. dazu: OASIS (2014)

³⁸ Vgl. dazu: Hofman, B. (2012)

Kategorie 8: Daten Import & Export

Bei der Aufnahme von Anforderungen ist es wichtig, mit dem Kunden in Interaktion zu stehen. Häufig werden Anforderungen zunächst mit anderen Werkzeugen erstellt und erst später in die Software Lösung übertragen. Daher ist es entscheidend, dass Anforderungen von anderen Programmen in das Anforderungsmanagement-Tool übernommen werden können. Gleichermaßen müssen Anforderungen jederzeit ausgegeben werden können. Daher beschäftigt sich Kategorie 8 mit dem Import und Export von Daten und den dabei unterstützten Formaten. Gängige Formate sind beispielsweise Microsoft Word und Excel, die Hypertext Markup Language (HTML) oder auch das Rich Text Format (RTF). Weitere mögliche Formate sind CSV, Comma-separated values, wie häufig für den Austausch von Datenbanktabellen genutzt, der Klartext (engl. Plain text) sowie das Portable Document Format (PDF).

Für die Anforderungen an sich bestehen auch verschiedene Formate, unter anderem das Requirements Interchange Format (ReqIF, beziehungsweise RIF) oder der Standard des Institute of Electrical and Electronics Engineers (IEEE) in Form von Software Requirements Specification (SRS). Des Weiteren existiert das „Product Requirements Document“ (PRD) als Darstellungsformat für Anforderungen, das häufig auf dem „Marketing Requirements Document“ (MRD) basiert³⁹.

Daten Import & Export
Unterstützte Formate für Import / Export
Microsoft Excel
HTML
Microsoft Word
RTF
CSV
Plain Text
PDF
Unterstützte Anforderungs-Formate
RIF
PRD
SRS
MRD

Tabelle 10: Kategorie „Daten Import & Export“ des Kriterienkataloges

³⁹ Vgl. dazu: Shrivathsan, M. (2012)

Kategorie 9: Analyse

Die letzte Kategorie des Fragenkataloges beschäftigt sich schließlich mit den Analyse Möglichkeiten, die von der Software geboten werden. Dies ist besonders entscheidend für die Unterstützung des Entscheidungsfindungsprozesses auf Management-Ebene, da hier die Anforderungen beziehungsweise der Fortschritt in Bezug auf das Anforderungsmanagement strukturiert und übersichtlich dargestellt werden.

Es wird daher untersucht, ob die Software es erlaubt, automatisch Berichte beziehungsweise graphische Analysen in Form von Diagrammen auszugeben. Gleichzeitig stellt sich die Frage, ob benutzerdefinierte Vorlagen integriert werden können und ob Test-Cases anhand der Anforderungen automatisch generiert werden können.

Analyse
Automatisierte Erstellung von Berichten
Automatisierte Erstellung grafischer Analysen
Einbindung benutzerdefinierter Vorlagen möglich
Automatische Erstellung von Test-Cases basierend auf Anforderungen

Tabelle 11: Kategorie „Analyse“ des Kriterienkataloges

Dieser entwickelte Fragenkatalog bildet das Kernstück unserer nachfolgenden Analyse. Er wurde auf die genannten Produkte angewandt, um auf diese Art und Weise in übersichtlicher Form die Fähigkeiten der einzelnen Lösungen darzustellen. Den ausgewählten Produkten wird jeweils zugeordnet, ob sie das Feature voll erfüllen (+), teilweise erfüllen (0) oder gar nicht erfüllen (-). Aufgrund der eingangs bereits erwähnten Schwierigkeiten der Informationsbeschaffung, existiert darüber hinaus die Antwort „keine Angabe“. Die dabei entstandene Matrix kann Anhang 1 entnommen werden.

Es sind teilweise große Abweichungen aber auch Überschneidungen zwischen den Produkten festzustellen. Um es zu ermöglichen, die Anforderungsmanagement-Software dahingehend zu beurteilen, ob sie tendenziell eher für kleine, mittelständige oder große Unternehmen geeignet wäre, wird im anschließenden Abschnitt herausgearbeitet, welche Features für welche Unternehmensgröße besonders entscheidend sind und inwiefern diese Gewichtung in dem Kriterienkatalog abgebildet ist.

4.2 Zuordnung der Funktionalitäten nach Unternehmensgröße

In einem zweiten Schritt werden nun die vorgestellten Software-Produkte für Anforderungsmanagement auf ihre Relevanz und Eignung für bestimmte Unternehmensgrößen untersucht. Dabei kann zwischen drei potenziellen Zielgruppen unterschieden werden: große, mittlere und kleine Unternehmen. Zur Bestimmung der Größe eines Unternehmens wird in dieser Arbeit die Kennzahl der Arbeitnehmerzahl herangezogen. Nach europäischer Rechtslegung zählen Unternehmen mit weniger als 50 Mitarbeitern zu kleinen Unternehmen, zwischen 50 und 249 Mitarbeitern zu mittleren Unternehmen und ab 250 Mitarbeitern zu großen Unternehmen.⁴⁰

Die Zuordnung, welches Produkt für welche Unternehmensgröße am besten geeignet ist, bestimmt sich durch das Vorhandensein gewisser Funktionalitäten. Zur präziseren und konsistenten Zuteilung werden die identifizierten Funktionalitäten des Fragenkataloges aus Kapitel 4.1 in die drei Kategorien *Basic*, *Advanced* und *Optional* eingeteilt. Erfüllt die untersuchte Software Lösung vor allem Basic-Funktionalitäten, wird das Produkt zur Nutzung in kleinen Unternehmen empfohlen. Erfüllt das Produkt zusätzlich Advanced-Funktionalitäten, kommt es zudem für mittlere Unternehmen in Betracht. Für große Unternehmen muss das untersuchte Produkt ergänzend noch Optional-Funktionalitäten vorweisen können. Entscheidend für die Zuordnung ist dabei eine festgelegte Mindestanforderung von 50 %. Erfüllt ein Produkt demnach mehr als die Hälfte der Funktionalitäten einer Kategorie, wie in Tabelle 12 dargestellt, kann eine Empfehlung für die entsprechende Unternehmensgröße ausgesprochen werden.

Unternehmensgröße	Vorhandene Funktionalitäten der Kategorie		
	Basic	Advanced	Optional
Klein	> 50 %		
Mittel	> 50 %	> 50 %	
Groß	> 50 %	> 50 %	> 50 %

Tabelle 12: Zuordnungsmatrix der Produkte nach Unternehmensgröße

Der Kriterienkatalog wird in 12 Basic-Funktionalitäten, 27 Advanced-Funktionalitäten und 30 Optional-Funktionalitäten eingeteilt. Besitzt also ein Produkt z.B. 7 von 12 Basic-Funktionalitäten, erfüllt es 58,3 % der Funktionalitäten der Basic-Kategorie. Somit kann das untersuchte Produkt für kleine Unternehmen in Betracht gezogen werden. Eine Übersicht der

⁴⁰ Vgl. EU-Kommission (2003)

eingordneten Funktionalitäten nach der kategorischen Einteilung aus Kapitel 4.1 liegt in Tabelle 13 vor.

	Basic	Advanced	Optional
Allgemeines	<i>Support</i>		
	<ul style="list-style-type: none"> • Webhilfe • Telefonischer Support 	<ul style="list-style-type: none"> • Web-Trainings 	<ul style="list-style-type: none"> • Trainings onsite
Anforderungsmanagement	<i>Anlegen, Ändern und Löschen von Anforderungen</i>		
	<ul style="list-style-type: none"> • Definierbare Richtlinien für Anforderungs-Attribute • Bewertung und Klassifikation von Anforderungen 	<ul style="list-style-type: none"> • Wiederverwendbarkeit angelegter Anforderungsprofile (Templates) • Anhang unterstützender Dokumente 	
	<i>Anforderungshistorie</i>		
		<ul style="list-style-type: none"> • Integriertes System zur Versionskontrolle • Vollständige Abbildung der Versionshistorie 	
		<ul style="list-style-type: none"> • Gleichzeitige Verwaltung von Anforderungen verschiedener Projekte • Agiles Projektmanagement ist abbildbar • Aufnahme des Status Quo ("Baselining") 	<ul style="list-style-type: none"> • Einbindung eines Glossars
	<i>Nachverfolgbarkeit von Anforderungen</i>		
	<ul style="list-style-type: none"> • Projektinterne Verlinkungen 	<ul style="list-style-type: none"> • Projektübergreifenden Verlinkungen • Kennzeichnung inkonsistenter Anforderungen (Orphans) 	<ul style="list-style-type: none"> • Verlinkung von Programmcode mit korrespondierenden Anforderungen
<i>Integrierte Funktionen für das Sammeln von Anforderungen & Daten</i>			
			<ul style="list-style-type: none"> • via Email • via Web
Änderungsmanagement		<ul style="list-style-type: none"> • Integriertes Änderungsmanagement • Erstellen, Bewerten und Freigeben von Änderungsanfragen 	
Collaboration		<ul style="list-style-type: none"> • Kommentierfunktion • Integrierte Community 	<ul style="list-style-type: none"> • Teilen von Anforderungen • (Teil-) Automatisierte Kommunikation mit Kunden und Shareholdern • Benutzerdefinierte Email Benachrichtigungen
Aufbau und Navigation	<ul style="list-style-type: none"> • Suchfunktion • Filterfunktion 		<ul style="list-style-type: none"> • Visualisierung mittels Use Cases
	<i>Ansichten</i>		
	<ul style="list-style-type: none"> • Tabellen Ansicht 	<ul style="list-style-type: none"> • Dokumenten Ansicht • Baum Ansicht 	<ul style="list-style-type: none"> • Roadmap Ansicht

Adminis- tration	<ul style="list-style-type: none"> Administrator Ansicht mit Informationen zu Systemstatus, Projekten und Nutzern 		
	<i>Benutzerverwaltung</i>	<ul style="list-style-type: none"> Integrierte Benutzer- und Rechteverwaltung Individuelle Anpassung von Nutzeransichten 	<ul style="list-style-type: none"> Rollenbasierte Rechteverwaltung
Zugang und Integration	<i>Zugriffsmöglichkeiten</i>		
	<ul style="list-style-type: none"> Web Zugang 	<ul style="list-style-type: none"> Desktop Zugang 	<ul style="list-style-type: none"> Mobiler Zugang
		<ul style="list-style-type: none"> Integriertes Test- und Defect-Management Bereitstellung einer öffentlichen API 	
	<i>Unterstützte Anmeldeverfahren</i>		
		<ul style="list-style-type: none"> SSO 	<ul style="list-style-type: none"> SAML OAUTH2 Google Sign-On
	<i>Integrierte Anbindung anderer Produkte</i>		
			<ul style="list-style-type: none"> Salesforce Zendesk Jira TFS
Daten- austausch	<i>Unterstützte Formate für Import/Export</i>		
	<ul style="list-style-type: none"> Microsoft Excel HTML 	<ul style="list-style-type: none"> Microsoft Word RTF CSV 	<ul style="list-style-type: none"> Plain text PDF
	<i>Unterstützte Anforderungs-Formate</i>		
		<ul style="list-style-type: none"> RIF 	<ul style="list-style-type: none"> PRD SRS MRD
Analyse		<ul style="list-style-type: none"> Automatisierte Erstellung von Berichten 	<ul style="list-style-type: none"> Automatisierte Erstellung grafischer Analysen Benutzerdefinierte Vorlagen Automatische Erstellung von Test-Cases
TOTAL	12	27	30

Tabelle 13: Übersicht der zugeordneten Basic-, Advanced- und Optional-Funktionalitäten

Basic

Die Basic-Kategorie beinhaltet alle grundlegenden Funktionalitäten, die wesentlich und kennzeichnend für Anforderungsmanagement-Software sind, wie z.B. die Möglichkeit zum Anlegen und Klassifizieren von Anforderungen, eine Filter- und Suchfunktion und die Unterstützung der meistbenutzten Datenformaten Microsoft Excel und HTML. Für Anwendungsmanagement-Software kann auch vorausgesetzt werden, dass es eine Möglichkeit zur Administration des Projektes und der verschiedenen Nutzer gibt. Die einfachste Zugriffsmöglichkeit auf das Anforderungsmanagement-System sollte via Web gewährleistet sein. Auch klassische Support-Möglichkeiten wie eine Telefon-Hotline oder eine FAQ-Seite im Web sind in der Basic-Kategorie aufgelistet.

All diese Funktionalitäten sollte eine Anforderungsmanagement-Softwarelösung unabhängig von der Unternehmensgröße vorweisen können. Produkte, die vorwiegend ausschließlich die

Basic-Funktionalitäten abdecken, sind vor allem für kleinere Unternehmen geeignet, da diese häufig nur die elementaren Funktionen von Anforderungsmanagement-Software benötigen. Zusätzliche Funktionen könnten als überflüssig empfunden werden und die Handhabung unübersichtlich und benutzerunfreundlich gestalten.

Advanced

Erweiterte Funktionen, die über die obengenannten grundlegenden Funktionalitäten von Anforderungsmanagement-Software hinausgehen und der Funktionalität des Produktes einen deutlichen Mehrwert bringen, werden in der Advanced-Kategorie gesammelt. Mittlere Unternehmen stehen im Vergleich zu kleineren Unternehmen einer Vielzahl von weiteren Herausforderungen gegenüber.

So kann sich Anforderungsmanagement in mittleren Unternehmen mit größeren Projekten zunehmend als unübersichtlich und aufwendig erweisen. Benötigt werden daher fortschrittliche Funktionalitäten, die die Übersichtlichkeit gewährleisten und die Benutzerfreundlichkeit steigern, wie z.B. die Wiederverwendbarkeit angelegter Anforderungsprofile, die Dokumentation der Anforderungshistorie, die Aufnahme des Status-Quos mittels Baselineing oder eine integrierte Änderungsmanagement-Funktion. Mit einer größeren Anzahl an Mitarbeitern im Unternehmen steigt die Wahrscheinlichkeit, dass an verschiedenen Projekten gleichzeitig gearbeitet wird. Daher sind Funktionen, die paralleles Arbeiten und projektübergreifende Verlinkungen erlauben, von hohem Wert. Gleichermaßen erhöht sich die Wahrscheinlichkeit, dass im Durchschnitt in größeren Teams zusammen an Projekten gearbeitet wird. Daher sind Collaboration-Funktionalitäten gefordert, die den Kommunikationsaustausch erleichtern, wie z.B. eine integrierte Community oder Kommentierfunktionen. Zudem kann die Administration bei einer erhöhten Mitarbeiterzahl pro Projekt durch Funktionen wie die Personalisierung von Nutzeransichten und eine integrierte Benutzer- und Rechteverwaltung vereinfacht werden.

Weitere Advanced-Funktionalitäten, die für mittlere Unternehmen von Bedeutung sein können und Anforderungsmanagement benutzerfreundlicher gestalten, sind unter anderem die Möglichkeit zur Abbildung von agilem Projektmanagement, ein integriertes Test- und Defect-Management, die Unterstützung weiterer Zugangsmöglichkeiten auf die Software oder die Unterstützung zusätzlicher Datenformate, wie z.B. Microsoft Word und CSV. Auch eine Analyse-Funktionalität zur automatischen Erstellung von Berichten ist eine wertvolle und zeitsparende Funktionalität, die dieser Kategorie zugeordnet wird.

Optional

Funktionalitäten, die „nice to have“, aber nicht essentiell für Anforderungsmanagement-Software sind, fallen unter die Kategorie Optional. In dieser Kategorie befinden sich spezielle

Funktionalitäten, die in bestimmten Fällen vor allem für große Unternehmen bedeutsam sein könnten. So ist z.B. die Skalierbarkeit auf eine hohe Anzahl von Nutzern von entscheidender Bedeutung für die Einsatzfähigkeit der Produkte in großen Unternehmen. Mit einem großen Mitarbeiterkreis steigt auch der Ruf nach erweiterten Collaboration-Funktionalitäten, die z.B. das Teilen von Anforderungen erlauben oder die Kommunikation mit Kunden und Shareholdern automatisieren. Weitere spezielle Optional-Funktionalitäten sind unter anderem die Unterstützung zusätzlicher Zugangsmöglichkeiten auf die Anforderungsmanagement-Software, die Unterstützung weiterer Datenformate, die integrierte Anbindung anderer Software-Produkte und fortgeschrittene Visualisierungs- und Analysefunktionen.

Wie eingehend erwähnt, wurde diese Einordnung der Funktionalitäten für die Bewertung der Produkte genutzt. Die ausgewählten Produkte wurden wie in Kapitel 4.1 geschildert der Analyse unterzogen. Für jede Lösung wurde im Kriterienkatalog errechnet, wie viele Basic-, Advanced beziehungsweise Optional Features sie erfüllt. Ein „+“ entspricht dabei einer Erfüllung zu 100 %, das heißt, das Produkt hat in dieser Kategorie einen Punkt erhalten. Für die Wertung „0“ wurden 0,5 Punkte vergeben, während „k.A.“ sowie ein „-“ keine Punkte in der entsprechenden Fähigkeit ergaben. Auf diese Art und Weise konnte für jede Software die Gesamtpunktzahl in der jeweiligen Spalte „Basic“, „Advanced“ und „Optional“ errechnet werden und anhand dessen der Prozentsatz erfüllter Fähigkeiten bestimmt werden.

Die so ermittelten Ergebnisse sind im nachfolgenden Kapitel für jede untersuchte Software individuell dargestellt.

4.3 Einordnung der verfügbaren Produkte in die Zielgruppen

In diesem Kapitel sollen die einzelnen Produkte, die bereits im Kriterienkatalog untersucht wurden, den entsprechenden Zielgruppen zugeordnet werden. Dabei wird die in Kapitel 4.2 dargestellte Methode als Grundlage für die die Einordnung genutzt. Es wird zwischen den kommerziellen Software Produkten und den Open Source Lösungen unterschieden.

4.3.1 Kommerzielle Software Produkte

Nachfolgend werden die zehn bereits vorgestellten kommerziellen Software Produkte unter Beachtung der Ergebnisse des Kriterienkatalogs betrachtet und den Zielgruppen zugeordnet. Tabelle 14 zeigt die Zuordnung der Produkte zu den empfohlenen Zielgruppen. Die jeweiligen Marktführer sind mit einem Stern gekennzeichnet.

Produktname	Hersteller	Zielgruppe
LiteRM	Clear Specs	Keine Empfehlung
CASE Spec ★	Goda	Kleine Unternehmen
OneDesk ★	OneDesk Inc.	
RM	Polarion	Mittlere Unternehmen
Accompa	Accompa Inc.	
ReMa	Accord Software & Systems	
Rational DOORS ★	IBM	
Quality Center	Hewlett-Packard	Große Unternehmen
Integrity	PTC	
Caliber	Borland	

Tabelle 14 Zuteilung der kommerziellen Produkte zu den Zielgruppen

Im Folgenden werden die Produkte gemäß der Reihenfolge in Tabelle 14 genauer betrachtet und anhand ihrer Eigenschaften den unterschiedlichen Unternehmensgrößen zugeteilt.

LiteRM – Clear Specs

Basic: 25%, Advanced: 30%, Optional: 10%, Preis: \$ 59,95 pro Lizenz

Das Tool Lite RM der Firma Clear Specs ist ein schlankes Anforderungsmanagement-Tool, welches auf einen kleinen Funktionsumfang zum günstigen Preis setzt.

Preis-Struktur

Mit dem Tool LiteRM versucht die Firma Clear Specs einen anderen Weg zu gehen als andere Anbieter, indem sie auf eine Niedrigpreisstrategie setzt. Mit gerade einmal \$ 59,95 pro Lizenz ist dieses Tool mit Abstand das günstigste, welches im Rahmen dieser Analyse untersucht wurde.

Funktionsumfang

Der günstige Preis spiegelt sich stark im Funktionsumfang wieder: LiteRM erfüllt gerade einmal 25 % der Basis-Fähigkeiten, 30 % der weiterführenden Fähigkeiten und 10 % der optionalen Fähigkeiten.

Clear Specs richtet einen zu geringen Fokus auf die elementaren Funktionen eines Anforderungsmanagement-Tools. Grundlegende Funktionalitäten wie zum Beispiel eine Tabellen-Ansicht oder eine Filterfunktion fehlen, bzw sind nur eingeschränkt verfügbar. Zudem ist kein

dedizierter Support verfügbar, sondern lediglich eine allgemeine E-Mail-Adresse der Firma für Fragen aller Art. Außerdem ist weder der Web-Zugriff auf das Programm noch ein Export nach Microsoft Excel möglich.

Aufgrund des stark eingeschränkten Funktionsumfangs kann LiteRM für **keine Unternehmenskategorie** empfohlen werden. Auch kleinen Unternehmen, welche sicherlich die Zielgruppe für das Produkt bilden, kann LiteRM nicht vorbehaltlos empfohlen werden. Grundlegende Funktionen sind nicht vorhanden, sodass der Kauf dieser Software auch nicht mit dem günstigen Preis gerechtfertigt werden kann. Unternehmen sollten hier abwägen, entweder mehr Geld in ein Produkt mit solidem Funktionsumfang zu investieren oder Open Source Lösungen als Alternative in Betracht ziehen.

CASE Spec – Goda

Basic: 91,7 %, Advanced: 77,8 %, Optional: 38,3 %, Preis: ab \$ 129 / Monat

CASE Spec von Goda Software ist eine umfassende Anforderungsmanagementtool zur software-basierten Unterstützung von Systementwicklungsprojekten. CASE Spec ist sowohl als Lizenz wie auch als SaaS verfügbar.

Funktionsumfang

Die Softwarelösung CASE Spec von Goda erfüllt im Kriterienkatalog 91,7 % der Basic-Funktionalitäten, 77,8 % der Advanced-Funktionalitäten und 38,3 % der Optional-Funktionalitäten. CASE Spec kann daher zunächst für mittlere Unternehmen empfohlen werden.

Allgemein ist für dieses Produkt festzuhalten, dass es über eine kostenlose Testversion verfügt, als eigenständiges Produkt zu erwerben ist, die Basic-Supportmöglichkeiten Telefon und Web zur Verfügung stehen und Webtrainings angeboten werden. CASE Spec erfüllt alle Basic- und Advanced-Funktionalitäten für die Kategorie Anforderungsmanagement, wie das Anlegen, Ändern und Löschen von Anforderungen, die Kontrolle der Anforderungshistorie, die Nachverfolgung von Anforderungen über projektinterne sowie -übergreifende Verlinkungen, die gleichzeitige Verwaltung von Anforderungen verschiedener Projekte, Baselining, und die Abbildung von agilem Projektmanagement. Auch für die Kategorie Aufbau und Navigation werden nahezu alle Funktionalitäten erfüllt: So werden verschiedene Ansichten von der Tabellen- bis zur Roadmap-Ansicht angeboten sowie eine Filter- und Suchfunktion. Auch die Advanced-Funktionalitäten für ein integriertes Änderungsmanagement sowie Test- und Defect-Management sind bei CASE Spec integriert. Das Programm ist über einen Web- und Desktop-Zugang aufrufbar; die optionale Funktionalität eines mobilen Zugangs ist nicht inte-

griert. Ob die Softwarelösung über Advanced- und Optional-Funktionalitäten wie spezielle Anmeldeverfahren, eine Integration anderer Produkte, oder die Bereitstellung einer öffentlichen API unterstützt, ist nicht aus dem Webauftritt des Produktes zu entnehmen. CASE Spec unterstützt alle Basic- und Advanced-Datenformate für den Import und Export von Anforderungen, wie Microsoft Word, Excel, HTML, RTF etc. Zum Teil werden sogar auch Optional-Datenformate wie Plain Text und PDF unterstützt. CASE Spec ermöglicht auch weitgehend alle im Kriterienkatalog vorgestellten Advanced- und Optional-Funktionalitäten für die Erstellung von Analyseberichten.⁴¹

Nachteile

Defizite weist CASE Spec vor allem in der Kategorie Collaboration auf. Außer einer Kommentierfunktion und der Möglichkeit der benutzerdefinierten Email-Benachrichtigungen werden Advanced- und Optional-Funktionalitäten für paralleles Arbeiten nur zum Teil oder gar nicht erfüllt. Dies ist ein entscheidender Nachteil für die Verwendung des Produktes in größeren Unternehmen mit größeren Teams. Auch bei der Benutzerverwaltung erfüllt CASE Spec nicht alle Basic- und Advanced-Funktionalitäten. So ist z.B. eine Administratoransicht oder die individuelle Anpassung weiterer Nutzeransichten nur zum Teil gegeben.

Preis-Struktur

Für \$ 129 im Monat wird das Produkt als SAS-Software angeboten. Darin wird Zugang für drei Nutzer gewährleistet, jeder weitere Nutzer kostet einen Aufpreis von \$ 20 im Monat. Voraussetzung ist lediglich eine Mindestvertragslaufzeit von 3 Monaten. Die relativ unverbindlichen Konditionen sind optimal für kleinere Unternehmen, die mit solch einer Lösungsmanagement-Software flexibel nach ihren Bedürfnissen einsetzen können. Auch die Möglichkeit, CASE Spec als eigenständiges Produkt zu erwerben, ist eine Option, die kleinere Unternehmen in Betracht ziehen können. In diesem Lizenz-Modell mit sogenannten *floating licenses* wird eine fortwährende Nutzung des Produktes gekauft. Pro Benutzer, auch genannt *floating client*, muss eine Lizenz erworben werden. Der Lizenzpreis für einen floating client beträgt \$ 1,000, für fünf floating clients \$ 2,500, für 60 floating clients \$ 17,000. CASE Spec Server und Unternehmensdatenbank residieren auf der Unternehmensseite. Die Software kann auf beliebig vielen Computern installiert werden und deren Lizenznutzer wird nur über die Anzahl der gleichzeitigen Nutzer gemessen.⁴² Insbesondere aufgrund des Preismodells kann für CASE Spec neben mittleren Unternehmen auch eine Empfehlung für kleinere Unternehmen ausgesprochen werden.

⁴¹ Vgl. Goda Software Company (o.J.)

⁴² Vgl. Goda Software Company (o.J.)

Nach der festgelegten 50 %-Hürde des Kriterienkatalogs kann für dieses Produkt eine Empfehlung für **mittlere Unternehmen** ausgesprochen werden. Es weisen allerdings auch einige Faktoren für den Einsatz von CASE Spec in **kleineren Unternehmen** hin. Vor allem das skalierbare Pricing-Modell der Softwarelösung ist attraktiv und bezahlbar für kleinere Unternehmensgrößen.

OneDesk – OneDesk Inc.

Basic: 100%, Advanced: 70%, Optional: 58%, Preis: 10\$/Monat – 250\$/Monat

OneDesk Requirements Management (im weiteren OneDesk) von OneDesk Inc. ist eine vollständig Cloud-basierte Anwendung zum Anforderungsmanagement.

Funktionsumfang

Der Funktionsumfang von OneDesk geht zum Teil weit über die Funktionalitäten klassischer Anforderungsmanagement-Software hinaus. Deutlich wird dies bereits im Aufbau der Anwendung: OneDesk präsentiert sich als allumfassende Lösung für Produkt- und LifeCycle Management mit integriertem Anforderungsmanagement. Eine eigenständige Anforderungsmanagement Lösung ist nicht verfügbar. OneDesk zeichnet sich weiterhin vor allem durch sein hohes Maß an Kompatibilität mit anderen Produkten, wie z.B. Salesforce.com, und seine integrierten Community-Funktionen aus.

In der Nutzwertanalyse erreicht OneDesk 100 % der Basic-Features, 70 % der Advanced-Features und 58 % der Optional-Features. Da alle Kategorien zu mehr als 50 % erfüllt sind, liegt zunächst eine Einordnung in die Sparte der großen Unternehmen nahe. Aufgrund der lediglich knappen Erfüllung der optionalen Funktionalitäten, die die Eignung für große Unternehmen beschreiben, werden bei der Einordnung jedoch noch weitere Aspekte mit einbezogen.

Preis-Struktur

Erwähnenswert ist zunächst das Preis- und Lizenzmodell von OneDesk. Der Hersteller zeigt bereits auf seiner Homepage eine leicht nachvollziehbare Preisstruktur, die sich wie folgt gliedert: \$ 10 pro Monat für 5 Nutzer mit 3 Projekten, \$ 29 je Monat für 10 Nutzer mit bis zu 5 Projekten, \$ 99 pro Monat für 30 Nutzer und 15 Projekte und \$ 250 je Monat für 50 Nutzer mit bis zu 30 Projekten.⁴³ OneDesk ist damit zum Teil erheblich günstiger als andere kommerzielle Produkte für große Unternehmen. Eine Einordnung in die Kategorie der mittleren Unternehmen wäre dadurch prinzipiell vertretbar.

⁴³ Vgl. OneDesk, Inc. (2014)

Skalierbarkeit & Flexibilität

Wie bereits bei der Preis-Struktur angedeutet, lässt sich OneDesk sehr leicht an veränderte Rahmenbedingungen anpassen. Das flexible Lizenzmodell sowie der Aufbau der Lösung lassen schnelle und einfache Skalierungen zu. Dies ist besonders relevant für kleine bis mittlere Unternehmen, die zunächst mit wenigen Projekten bzw. Nutzern beginnen, im Laufe der Zeit aber unter Umständen großen Wachstum verzeichnen. OneDesk bietet für solche Situationen alle nötigen Voraussetzungen.

Zusammenfassend wird OneDesk in dieser Arbeit in die Kategorie der **mittleren Unternehmen** eingeordnet. Dies begründet sich jedoch nicht in der mangelnden Eignung für große Unternehmen, sondern viel mehr in der speziellen Eignung für mittlere Unternehmen. Entscheiden hierfür ist der große Funktionsumfang in Verbindung mit den o.g. Punkten.

RM – Polarion

Basic: 100%, Advanced: 85%, Optional: 43%, Preis: ab \$1.490

Das Tool RM der Firma Polarion bietet zu einem relativ günstigen Preis ein Tool mit großem Funktionsumfang. Die Software kann zusätzlich durch Add-Ons sehr einfach um weitere Funktionen ergänzt werden.

Funktionsumfang

Die Software RM der Firma Polarion schneidet zunächst bei der Bewertung der Basic-Features ausgezeichnet ab, die Funktionen werden zu 100 % erfüllt. Ebenso werden die weitergehenden Funktionen zu 85 % abgedeckt, die optionalen Funktionen zu 43 %. Setzt man hier nun wieder die 50 % Hürde zur Einordnung in die Unternehmenskategorien an, lässt sich Polarion sowohl für kleine Unternehmen als auch für Mittelständler empfehlen.

Preis-Struktur

Ein großer Vorteil von Polarion liegt in seinem Preis, so ist eine zeitlich unbegrenzt gültige Lizenz bereits ab \$1.490 zu erwerben. Im Bereich der professionellen Anforderungsmanagement-Tools ist dies somit eines der günstigsten Produkte. Ebenso bietet RM eine Vielzahl verschiedener Lizenzmodelle und kann sowohl on premise als auch als SaaS erworben werden. Dadurch wird das Produkt insbesondere für Mittelständler attraktiv, die für einen günstigen Preis ein solides Produkt erhalten.

Erweiterbarkeit

Neben dem Preis spricht auch die Anpassbarkeit des Produkts für Polarion. Neben der Grundversion stellt die Firma eine Vielzahl von Erweiterungen zur Verfügung, mit deren Hilfe

auch spezielle Kundenwünsche abgedeckt werden können. So bietet Polarion beispielsweise Extensions für MS Office, Subversion oder agile Software-Entwicklung.

Support

Ebenso bietet Polarion die gesamte Palette an Support-Funktionen, wie telefonischen Support und Web Hilfe. Zusätzlich werden aber auch Web-Trainings und Onsite-Schulungen angeboten, was beispielsweise die Einführung von RM erleichtern kann und somit eine höhere Akzeptanz bei den Mitarbeitern generiert.

Somit lässt sich Polarion als ideales Produkt für **mittelständische Unternehmen** empfehlen, da diese eine Vielzahl an Funktionen zu einem günstigen Preis zur Verfügung gestellt bekommen. Kleine Unternehmen müssen abwägen, ob die breite Funktionspalette benötigt wird oder ob es vielleicht auch ein günstigeres Produkt mit etwas reduzierterem Funktionsumfang gibt, welches ausreicht. Auch für Großunternehmen ist RM grundsätzlich attraktiv, jedoch muss hier genau überprüft werden, ob die bereitgestellten optionalen Funktionen ausreichend sind, bzw. ob die fehlenden Funktionen über Extensions nachgerüstet werden können.

Accompa – Accompa Inc.

Basic: 83,3%, Advanced: 55,6%, Optional: 46,7%, Preis: \$199/Monat - \$799/Monat

Accompa ist eine Softwarelösung im Segment des Anforderungsmanagements, welche Testversionen und persönliche Demoversionen anbietet. Die Software hat viele Funktionen. Diese werden in drei verschiedene Preisstrukturen unterteilt, womit die Software für Unternehmen unterschiedlicher Größe attraktiv wird. Im Folgenden werden das Preismodell und die dazugehörigen Funktionen vorgestellt.

Preis-Struktur

Accompa ist in drei verschiedenen Umfangsgrößen erhältlich. Diese nennen sich Standard für \$199 im Monat, Corporate für \$399 im Monat und Enterprise für \$799 im Monat. Die Preise gelten für jeweils fünf Nutzerlizenzen. Im Standardpaket kostet jede weitere Lizenz pro Monat \$29, für Corporate-Nutzer \$39 und für die Enterprise-Nutzer \$79.⁴⁴

Funktionsumfang

Generell beinhalten die umfangreicheren Produkte auch die Funktionen der kleineren Lösungen. Alle Lösungen sind ein Software-as-a-Service (SaaS). Für Kunden steht lediglich ein

⁴⁴ Vgl. Accompa (o.J.)

Websupport bereit, über Trainings oder den Garantiumfang werden auf der Website keine Angaben gemacht.

Standard - Die Standardlösung vom Accompa ist ein klassisches Tool zum Anforderungsmanagement mit den üblichen Werkzeugen. Die Anforderungen können für verschiedene Projekte gleichzeitig verwaltet werden. Eine Visualisierung der Anforderungen erfolgt mittels Use Cases. Ein integriertes Diskussionsboard zum Austausch mit dem Kunden und anderen Projektbeteiligten steht den Anwendern zur Verfügung. Abhängigkeiten und Beziehungen zwischen Anforderungen können im Standardtool abgebildet werden. Die Software verfügt über eine Funktion mittels derer Excel-Dateien sowohl importiert als auch exportiert werden können. Zudem lassen sich Word-Dateien, PDF-Dokumente und HTML-Dateien exportieren. Ein weiteres Merkmal ist die Verfügbarkeit einer Änderungshistorie.⁴⁵

Corporate - In der Corporate-Lösung steht zusätzlich die Funktion bereit, mittels Webformularen und Email Anforderungen zu sammeln und direkt in die Projekte zu übertragen. Außerdem werden hier die verschiedenen Rollen im Projekt aufgegriffen und den Nutzer entsprechende Nutzungsrechte eingeräumt. Diese Funktion kann von bis zu zehn Anwendern genutzt werden.⁴⁶

Enterprise - In der Enterprise Lösung steht den Anwendern eine öffentliche API zur Verfügung. Außerdem ist ein Single-Sign-on Verfahren integriert, welches mittels einer Active Directory oder LDAP realisiert wird. Auch hier können den verschiedenen Nutzerrollen individuelle Rechte eingeräumt werden. In diesem Produkt können beliebig viele Nutzer aufgesetzt werden.⁴⁷

Insgesamt erfüllt Accompa 83,3% Basic-Features, 55,6% der Advanced-Features und 46,7% der Optional-Features. Da die Software somit in den Kategorien Basic und Advanced mehr als die Hälfte der Features erfüllt, wird die Software der **mittleren Unternehmensgröße** zugeschrieben. Diese Einschätzung ergibt sich des Weiteren aus dem Preis, zu dem die Software verfügbar ist, sowie aus ihren Funktionen und Features.

⁴⁵ Vgl. Accompa (o.J.)

⁴⁶ Vgl. ebenda

⁴⁷ Vgl. ebenda

ReMa – Accord Software & Systems

Basic: 79%, Advanced: 59%, Optional: 20%, Preis: keine Angabe

ReMa – Requirements Manager (im Weiteren ReMa) von Accord Software & Systems ist eine Desktop-basierte Anwendung für das Anforderungsmanagement. ReMa ist als eigenständige Anforderungsmanagement-Software verfügbar.

Funktionsumfang

Bei einer Bewertung mithilfe des Kriterienkataloges erreicht ReMa 79 % der Basic-, 59 % der Advanced- und 20 % der Optional-Kriterien. Obwohl bzw. gerade weil diese Zahlen vor allem im Bereich der optionalen Kriterien in hohem Maße von fehlenden Informationen beeinflusst werden, erscheint das Produkt generell für mittlere Unternehmen geeignet. Dieser Eindruck ergibt sich aus der durchaus breiten Abdeckung aller grundlegenden Funktionen sowie dem Überschreiten der 50 %-Hürde bei den Advanced-Features. Laut Informationen der Hersteller-Webseite ist eine Testversion auf Anfrage verfügbar. Support bzw. Training erscheinen nur eingeschränkt verfügbar

Preis-Struktur

Zu der preislichen Eignung von ReMa kann aufgrund fehlender Informationen leider keine Aussage getroffen werden.

Eine endgültige Zuteilung zu einer der drei Unternehmensgrößen gestaltet sich aufgrund der fehlenden Informationen als schwierig. Dank der erzielten Ergebnisse in der Bewertung durch den Kriterienkatalog wird ReMa jedoch in den Bereich der **mittleren Unternehmen** eingeteilt.

Rational DOORS – IBM

Basic: 100%, Advanced: 83%, Optional: 65%, Preis: \$ 4.924 / Jahr und Nutzer

Wie bereits in einem der vorangegangenen Kapitel erwähnt, stellt IBM mit Rational DOORS den aktuellen Marktführer. Aufgrund des breiten Angebots von IBM für alle möglichen Bedürfnisse des Kunden kann Rational DOORS zudem sehr gut mit anderen IBM Produkten verknüpft und integriert werden.

Funktionsumfang

Ein weiterer großer Vorteil von IBM Rational DOORS liegt in seinem sehr breiten Funktionsumfang. So erfüllt das Tool 100 % der Basis-Features, 83 % der erweiterten und 65 % der optionalen Funktionen, was in dieser Kategorie den besten Wert darstellt. Zudem ist das

Tool vom Grundaufbau her derart gestaltet, dass jederzeit auch individuelle Wünsche des Kunden mit aufgenommen und in das Tool implementiert werden können. Diesen Service zur Anpassung des Tools an individuelle Kundenbedürfnisse bietet IBM durch die hauseigene Beratungssparte selbst an.

IBM Rational DOORS bietet fast alle Funktionen, die im Bereich des Anforderungsmanagements denkbar sind. Dies beginnt beispielsweise bei individueller Skalierbarkeit, sodass das Produkt exakt an die Anzahl der Nutzer angepasst werden kann und bei Bedarf die Lizenzen auch erweitert oder verringert werden können. Zudem bietet IBM sämtliche Support-Optionen – sowohl klassischen Telefonsupport und Web Hilfe als auch Online- und Onsite-Trainings.

Die grundlegenden Funktionen des Anforderungsmanagements, welche im hier zugrunde gelegten Kriterienkatalog die nächste Kategorie bilden, werden alle erfüllt, ebenso wie Collaboration-Funktionen, welche gerade bei großen Unternehmen immer mehr an Bedeutung gewinnen.

Ebenso bietet Rational DOORS umfangreiche Funktionen zur Administration, sodass auch bei großen Projekten die Projektleitung stets den Überblick behalten und das Projekt entsprechend gesteuert werden kann. Darüber hinaus lässt sich auch das Änderungsmanagement mithilfe des Tools durchführen.

Zugänglich ist die Software sowohl via Desktop als auch über das Internet, wobei das Tool auch die geläufigen Anmeldeverfahren unterstützt. Auch beim Thema Datenaustausch bietet das Tool vielfältige Möglichkeiten, so werden sämtliche im Kriterienkatalog definierten Dateiaustauschformate unterstützt. Gleichmaßen gut schneidet Rational Doors in der letzten Kategorie des Kriterienkatalogs, beim Thema Analysefunktionen, ab und stellt dem Benutzer eine Reihe verschiedener Funktionalitäten zur Verfügung.

Preis-Struktur

Der breite Funktionsumfang des Tools hat allerdings auch seinen Preis, so werden pro Jahr und Nutzer \$ 4.924 fällig, wodurch sich das Produkt klar am oberen Ende der Preisspanne einordnet. Somit stellt sich für jedes Unternehmen die Frage, in welchem Umfang ein Anforderungsmanagement-Tool genutzt werden soll und ob dieser eine derartige Investition rechtfertigt.

Insgesamt stellt sich IBM Rational DOORS somit in erster Linie für **Großunternehmen** als attraktiv dar, da das Tool umfangreiche Funktionen bietet, allerdings auch zu einem entsprechenden Preis. Somit müssen bereits mittelgroße Unternehmen sehr genau abwägen, ob dieser Funktionsumfang wirklich benötigt wird oder ob eine schlankere und dafür kostengünstigere Lösung eine bessere Alternative darstellt.

Quality Center (QC) – Hewlett-Packard GmbH

Basic: 83,3%, Advanced: 87 %, Optional: 45%, Preis: keine Angabe

Das Quality Center der Hewlett-Packard GmbH ist ein Anforderungsmanagementtool, das über umfangreiche Funktionen verfügt. Die Software zeichnet sich durch ein integriertes Fehlermanagement aus.

Funktionsumfang

Die Softwarelösung erfüllt im hier vorgestellten Kriterienkatalog 83,3 % der Basic-Features, 87 % der Advanced-Features und 45 % der Optional-Features. Aus diesem Grund wird das HP QC in dieser Arbeit den mittleren Unternehmensgrößen empfohlen.

Die nähere Betrachtung zeigt, dass dem Anwender ein Telefonsupport, Websupport und Trainings zum Produkt angeboten werden.⁴⁸ Garantieleistungen sind vorhanden, werden jedoch im Webauftritt nicht spezifiziert. Alle Basisfeatures wie die Verwaltung von Anforderungen in mehreren Projekten separat oder auch übergreifend, die Verwendung von Templates, eine Versionshistorie, Abbildung von Abhängigkeiten und Verlinkungen sind in der Software enthalten. Eine Kommunikationsbasis innerhalb des Programms ist vorhanden.⁴⁹ Das HP QC verfügt über verschiedene Ansichtsmöglichkeiten wie eine Roadmap-Ansicht oder eine Baumansicht. Rollenbasierte Nutzereinstellungen können in der Software vorgenommen werden. Die Lösung beinhaltet zudem ein eigenes Änderungsmanagement sowie ein integriertes Fehlermanagement. Dieses zeichnet die Lösung von HP aus. Es bestehen für den Nutzer verschiedene Zugangsmöglichkeiten, über den Desktop oder das Web. Das Programm nutzt eine öffentliche API und ist an das Produkt Salesforce angebunden, welches es für große Unternehmen ebenfalls attraktiv erscheinen lässt.⁵⁰

Leider liegen keine Preisinformationen zu dem Produkt vor. Anhand der Fähigkeiten zeigt sich jedoch, dass das HP Quality Center für **große Unternehmen** geeignet erscheint.

⁴⁸ Vgl. Hewlett-Packard GmbH (o.J.)

⁴⁹ Vgl. ebenda

⁵⁰ Vgl. Hewlett-Packard GmbH (o.J.)

Integrity – PTC

Basic: 75%, Advanced: 78%, Optional: 32%, Preis: keine Angabe

Die Software-Lösung Integrity der Firma PTC dient der Verwaltung sämtlicher Prozesse der System- und Softwareentwicklung über den gesamten Lebenszyklus hinweg. Entsprechend umfasst es auch Werkzeuge des Anforderungsmanagement.

Funktionsumfang

Integrity weist viele Fähigkeiten des Anforderungsmanagements auf. Aus dem Kriterienkatalog ergibt sich, dass 75 % der Basis-Fähigkeiten, 78 % der weiterführenden und 32 % der optionalen Features erfüllt werden.

Legt man die 50 % Hürde bei der Einordnung in die Zielgruppen zugrunde, lässt sich schließen, dass Integrity hauptsächlich für mittelständige Unternehmen interessant ist mit einer Tendenz in Richtung großer Unternehmen. Insbesondere die weitreichenden Kollaborationsmöglichkeiten in Form von Diskussions Foren, automatisierten E-Mail Benachrichtigungen und Kommentier Funktionen sowie die Benutzerverwaltung durch einen Administrator machen das Werkzeug zu einem geeigneten Kandidaten für das Anforderungsmanagement in Unternehmen, die mehrere Standorte aufweisen. Hinzu kommt ein hohes Maß an Skalierbarkeit, wodurch die Anforderungsaufnahme für Projekte mit einer Vielzahl an Aspekten möglich wird. Die Support Funktionalitäten reichen von Telefon-Support bis hin zu onsite Trainings. Die fehlenden Punkte, die das Produkt bei den optionalen Fähigkeiten aufweist, liegen vornehmlich in den unterstützten Datei- und Anforderungsformat. Gleichermäßen gibt es Abzüge, was die Integration anderer Produkte angeht, wobei hier angemerkt werden muss, dass Anforderungsmanagement an sich bereits in ein Application-LifeCycle-Management-Tool eingebunden ist. So umfasst Integrity nicht nur die Lösung für das Anforderungsmanagement, sondern auch das Test Management, das Änderungs- und Konfigurationsmanagement sowie das System Modellierung Management.⁵¹

Preis-Struktur

Dieser Aspekt kann zum einen als Vorteil gesehen werden, gleichzeitig bedingt es allerdings auch sehr hohe Kosten, die mit dem Erwerb des Produktes verbunden sind. Die Beratungsfirma Info-Tech Research Group hat in einer Analyse von Anforderungsmanagement Software aus dem Jahr 2012 der Lösung von PTC ein „Poor“ in der Kostspieligkeit gegeben und damit eines der schlechtesten Rankings, was den Preis betrifft⁵². Darüber hinaus urteilt Info-Tech, dass das Produkt eher für Unternehmen, mit mindestens 25 Software-Entwicklern

⁵¹ Vgl. PTC Inc. (2014)

⁵² Vgl. Info-Tech Research Group (2012)

ausgelegt ist und sich somit eher für große Unternehmen eignet. Aus dem Kriterienkatalog ergibt sich allerdings, dass die Fähigkeiten in diesem Falle nicht so umfassend sind, wie bei Vergleichsprodukten.

Insgesamt lässt sich folgern, dass PTC Integrity von den Fähigkeiten für Mittelständler geeignet wäre, der Preis das Segment aber tendenziell auf **große Unternehmen** beschränkt.

Caliber – Borland

Basic: 87,5 % %, Advanced: 68,5 %, Optional: 48,3 %, Preis: 7.440 € (Multi)/2.380 € (Single)

Im Januar 2014 wurde Version 11.3 der Software-Lösung Caliber von Borland veröffentlicht. Das Tool dient speziell dem Anforderungsmanagement und zeichnet sich vor allem durch seine umfassenden Visualisierungsmöglichkeiten aus.

Funktionsumfang

Das Produkt schneidet in den Kriterien Basic und Advanced sehr gut ab, hier verzeichnet es 87,5 % beziehungsweise 68,5 % der erreichbaren Punkte. Die optionalen Fähigkeiten sind zu 48,3 % vertreten und somit erreicht Caliber in dieser Kategorie auch fast die 50 % Hürde.

Das Produkt in Bezug auf das Anforderungsmanagement stark aufgestellt, es gibt lediglich Abzüge in den Aspekten des parallelen Arbeitens und der Kennzeichnung inkonsistenter Verlinkungen. Gleichmaßen ausgeprägt sind die Möglichkeiten für die interne Kollaboration sowie die Darstellungsmöglichkeiten und Benutzerverwaltung. Die unterstützten Anmeldeverfahren konnten nicht ermittelt werden, doch wurde ausdrücklich auf die Option des Single-Sign Ons und auf die Bereitstellung einer öffentlichen Programmierschnittstelle (engl.: Application Programming Interface, API), die eine beliebige Integration weiterer Programme erlaubt, hingewiesen. Weitere Abstriche müssen in dem Bereich Daten Import & Export gemacht werden – insbesondere für den Import werden nur die Microsoft Lösungen Word und Excel unterstützt, für den Export sind weiterhin CSV und Plain Text möglich. Weitere Stärken des Programmes liegen in den umfassenden Analysemöglichkeiten, die sowohl grafische als auch textbasierte Berichte erstellen können.

Vor- und Nachteile

Eine Besonderheit, die auch die Beratungsfirma Seilevel in ihrer Evaluation von Anforderungsmanagement-Software betont, ist die Möglichkeit, agiles Projektmanagement in dem Werkzeug abzubilden. Darüber hinaus werden die Verlinkungs- und grafischen Darstellungsmöglichkeiten der Zusammenhänge zwischen den Anforderungen positiv hervorgehoben⁵³. Einer der aufgeführten Nachteile ist die teils unübersichtlich gestaltete Benutzeroberfläche sowie die beschränkte Möglichkeit, mehrere Anforderungen parallel zu bearbeiten.

Aufgrund der umfangreichen Funktionalitäten und des Preises ist Caliber eher für mittelständische als kleine Unternehmen geeignet. Dank der vielseitigen Features, die die interne Kollaboration ermöglichen, ist es daher sogar eher denkbar, Caliber als Anforderungsmanagement-Software in **großen Unternehmen** einzuführen.

4.3.2 Open Source Software

Analog zum vorhergegangenen Kapitel werden im Nachfolgenden die Open Source Lösungen vorgestellt. Tabelle 15 zeigt die Zuordnung der Produkte zu den empfohlenen Zielgruppen

Produktname	Entwickler	Zielgruppe
OSRMT	ideaStub	Keine Empfehlung
SHORE v. 2.0	SD & M AG	Keine Empfehlung
Unicase	TU München	Keine Empfehlung
ReqT	Lund University	Keine Empfehlung
aNimble	ideaStub	Kleine Unternehmen
rmtoo	flonatel	Kleine Unternehmen
ProR	Formal Mind GmbH	Kleine Unternehmen

Tabelle 15 Zuteilung der Open Source Produkte zu den Zielgruppen

Im Folgenden werden die Produkte gemäß der Reihenfolge in Tabelle 15 genauer betrachtet und anhand ihrer Eigenschaften den unterschiedlichen Unternehmensgrößen zugeteilt.

⁵³ Vgl. Beatty, J. u.a. (o. J.), S. 7

OSRMT – ideaStub

Basic: 58,3%, Advanced: 31,5%, Optional: 11,7%

Das Tool OSRMT der Firma ideaStub wurde 2007 entwickelt und auf der Plattform Sourceforge.net bereitgestellt. Seitdem wurden allerdings keine Änderungen vorgenommen, weshalb die Software in dieser Arbeit **nicht näher betrachtet** wird. OSRMT wurde von ideaStub jedoch als Grundlage für das Tool aNimble verwendet, welches im Folgenden ebenfalls vorgestellt wird.

SHORE v. 2.0 – SD & M AG

Basic: 66,7%, Advanced: 22,2%, Optional: 0%

SHORE, kurz für Semantic Hypertext Object Repository, ist ein XML-basiertes Repository für Projektdokumente beliebigen Typs, die inhaltlich in einem Zusammenhang stehen wie z.B. die Ergebnisdokumente und Programme, die in einem Software-Entwicklungsprojekt anfallen.

Funktionsumfang

Die Palette von Dokumenttypen, die sich nach SHORE importieren lassen, reicht von schwach strukturierten Textdokumenten, wie Anforderungen, Testfällen und Fehler- und Änderungsprotokollen, bis hin zu streng formalisierten Dokumenten wie Programm-Quelltexte mit definierter Syntax. Die Idee ist, mit SHORE alle Phasen eines Entwicklungsprojektes abzudecken.⁵⁴

Die letzte Aktualisierung dieses Open Source Projektes wurde zuletzt 2009 vorgenommen und wurde seither nicht weiter verfolgt.⁵⁵ Aus diesem Grund wird die Software in dieser Arbeit **nicht weiter berücksichtigt**.

Unicase – TU München

Basic: 50%, Advanced: 22,2%, Optional: 15%

Das Anforderungsmanagementtool der TU München wurde 2009 entwickelt und seither nicht weiter verfolgt.⁵⁶ Aus diesem Grund wird die Software in dieser Arbeit **nicht berücksichtigt**.

⁵⁴ Vgl. sd&m AG (2001)

⁵⁵ Vgl. SourceForge (2014)

ReqT – Lund University

Basic: 66,7%, Advanced: 37%, Optional: 16%

ReqT ist eine frei verfügbare, Kommandozeilen-basierte Anwendung für Anforderungsmanagement. Erstmals entwickelt wurde es 2011 an der Lund Universität in Schweden.

Funktionsumfang

ReqT erreicht bei einer Bewertung mithilfe des Kriterienkatalogs derzeit 66,6 % der Basic-, 37 % der Advanced- sowie 16 % der Optional-Features. Obwohl diese Bewertung generell zumindest eine Eignung für kleine Unternehmen nahe legt, muss dieses Ergebnis, nicht zuletzt aufgrund der nur knappen Überschreitung der 50 % Hürde, kritisch betrachtet werden.

Nachteile

Bereits eine Analyse der allgemeinen Charakteristika offenbart erste Schwächen, die einer Eignung für den unternehmerischen Einsatz entgegenstehen. Obwohl in regelmäßigen Abständen an ReqT gearbeitet wird und auch entsprechende Commits verzeichnet sind, gibt es offenkundig nur einen Entwickler, der derzeit aktiv an ReqT arbeitet. Dies bedeutet für Unternehmen zum einen eine hohe Abhängigkeit sowie eine zu erwartende langsame Weiterentwicklung und Fehlerbehebung. Eine fehlende GUI erschwert zudem die Verwendung der Applikation für Nutzer ohne entsprechende Einarbeitung und steht damit einer breiten Nutzung im Weg. Positiv zu vermerken ist jedoch die hohe Skalierbarkeit und Flexibilität der Applikation. Diese wird vor allem durch einen semi-formalen Aufbau sowie die Einbindung von Scala als Programmiersprache erreicht.⁵⁷

Vorteile

Bei den weiteren Funktionen kann ReqT vor allem bei der Definition, Erstellung, und Verarbeitung von Anforderungen und den dazugehörigen Datenmodellen punkten. Des Weiteren werden Filter- und Suchfunktion, Microsoft Word und Excel, sowie verschiedene Berichte und Analysen unterstützt.

Alles in allem handelt es sich bei ReqT um ein Tool, dessen Stärken vor allem in seinem sehr flexiblen Aufbau liegen. Um diese Stärken nutzen zu können bedarf es allerdings aufgrund der fehlenden Benutzeroberfläche fortgeschrittener Programmier-Kenntnissen sowie einer intensiven Einarbeitung. ReqT ist daher prädestiniert für eine Nutzung in technischen Forschungsabteilungen sowie in der Lehre, kann jedoch aus den oben genannten Gründen nicht für Unternehmen empfohlen werden und wird daher in dieser Arbeit **nicht weiter berücksichtigt**.

⁵⁶ Vgl. Brügge, B./Creighton, O./Helming, J./Kögel, M. (2009), S. 1

⁵⁷ Vgl. Regnell, B. (2013), S.5

aNimble – ideaStub

Basic: 58,3%, Advanced: 33,3%, Optional: 10%

Wie zuvor angesprochen, ist das Tool aNimble die Weiterentwicklung von OSRMT; beide Lösungen werden von der Firma ideaStub angeboten. Im Gegensatz zu OSRMT ist aNimble deutlich aktueller, jedoch muss auch hier kritisch angemerkt werden, dass das letzte Release aus dem Jahr 2012 stammt und somit bereits über 2 Jahre zurückliegt. Da das Tool jedoch allein im Jahr 2013 über 6.000-mal heruntergeladen wurde, soll dieses hier kurz vorgestellt werden.

Funktionsumfang

Betrachtet man den Funktionsumfang, welchen aNimble bietet, so erfüllt das Tool 50 % der Basis-Funktionen, 30 % der weitergehenden Funktionen und 7 % der optionalen Funktionen. Bewertet man wieder mithilfe der 50 % Schwelle, kommt aNimble somit für kleine Unternehmen in Frage, da das Tool genau die Hälfte der Basis-Funktionen erfüllt.

Die Besonderheit von aNimble ist, dass es komplett web-basiert ist, wodurch sich das Tool von den meisten anderen Open Source Tools unterscheidet. Zudem unterstützt aNimble die agile Software-Entwicklung und bietet in Form einer Kommentier-Funktion bereits erste Collaboration-Ansätze. Zudem bietet aNimble Analyse-Optionen, sodass Berichte und grafische Analysen automatisch generiert werden können.

Nachteile

Kritisch zu sehen ist, dass es keine wirkliche Entwickler-Community gibt, sondern das Tool ausschließlich von ideaStub entwickelt wird. Das Unternehmen bietet keine Funktionen wie beispielsweise einen Bugtracker an. Laut bereitgestellten Wiki liegt die letzte Änderung bereits über ein Jahr zurück.

Alles in allem kann aNimble mit Fokus auf den Funktionsumfang für **kleine Unternehmen** empfohlen werden, da das Tool hier ein interessantes Spektrum an Möglichkeiten bietet. Allerdings sollte man auch bedenken, dass das Tool bereits heute veraltet ist und auch nicht sichergestellt ist, dass Fehler in Zukunft ausgebessert werden. Nur wer bereit ist, dieses Manko in Kauf zu nehmen, sollte über einen Einsatz von aNimble nachdenken.

rmtoo – flonatel

Basic: 33,3%, Advanced: 13%, Optional: 10%

Das Tool rmtoo der Firma flonatel bringt zunächst denselben Nachteil mit sich wie das zuvor erläuterte Produkt aNimble. Auch hier liegt die letzte Aktualisierung im Jahr 2012 während es seither keinerlei Neuerungen mehr gegeben hat. Ebenso zeigen die Bewertungen, dass das Tool in keiner Kategorie mehr als 20 % erreicht. Dies lässt sich vor allem auf die fehlende GUI zurückführen, was eine Vielzahl möglicher Funktionen ausschließt. Aus diesem Gründen wird das Tool für die weiteren Untersuchungen **nicht berücksichtigt**.

ProR – Formal Mind GmbH

Basic: 54%, Advanced: 33%, Optional: 20%

Das Open Source Produkt ProR ist seit Mitte April 2014 als Version 0.10 kostenlos bei der Formal Mind GmbH verfügbar.

Funktionsumfang

Was die Fähigkeiten des Kriterienkataloges betrifft, so erreicht ProR derzeit 54 % der Basic-, 33 % der Advanced- und 20 % der Optional- Fähigkeiten. Somit kann nur im Bereich der Basis Features die 50 % Hürde erreicht werden, was das Produkt vornehmlich für kleine Unternehmen interessant macht.

Die genauere Betrachtung zeigt, dass das Produkt in Bezug auf die Kriterien des Anforderungsmanagements gut abschneidet – zumindest sind hier alle erforderlichen Basic Features erfüllt. Auch gibt es Ansätze für die Abbildung der Versionshistorie sowie des Baselinings. Wenig fortgeschritten sind allerdings die Administration sowie die Analyse-Möglichkeiten, sodass das Werkzeug bisher für mittelständige Unternehmen wenig Unterstützung bietet.

Vorteile

Positiv anzumerken ist, dass das letzte Release erst vor kurzem veröffentlicht wurde und somit aktiv an der Entwicklung der Lösung gearbeitet wird.

Von seinen Fähigkeiten her ist ProR demnach zum jetzigen Zeitpunkt vor allem für **kleine Unternehmen** geeignet, die darüber hinaus davon profitieren würden, das Produkt kostenlos nutzen zu können.

5 Vergleichsstudie der Open Source Produkte

Dieses Kapitel soll dazu dienen, die verfügbaren kommerziellen Produkte für die verschiedenen Unternehmensgrößen mit den möglichen Alternativen aus dem Open Source Segment zu vergleichen. Es wird dabei zwischen großen, mittelständigen und kleinen Unternehmen unterschieden und jeweils eines der führenden kommerziellen Produkte vorgestellt. Es ist dabei zu beachten, dass es sich nicht zwangsläufig um den Marktführer handelt, sondern um das Produkt, das nach Betrachtung der Matrix am geeignetsten für die jeweilige Zielgruppe erscheint. Idealerweise wird nach der Vorstellung dieses Produktes die jeweils denkbare Open Source Alternative vorgestellt, die sich gleichermaßen aus dem Kriterienkatalog ergeben hat. Im Anschluss sollen die beiden Produkte miteinander verglichen werden, um schließlich das Potenzial der Open Source Lösung beurteilen zu können. Da sich nicht für alle Unternehmensgrößen eine Open Source Alternative anbietet, wird stattdessen auf denkbare kommerzielle Substitutionsprodukte eingegangen.

5.1 Große Unternehmen

In Kapitel 4 wurde dargestellt, welche Funktionalitäten für den Einsatz von Anwendungsmanagement-Software in großen Unternehmen nötig und attraktiv sind. Aufgrund der hohen Mitarbeiterzahl in dieser Zielgruppe ist zum einem die Skalierbarkeit der Softwarelösung auf eine hohe Anzahl von Nutzern von entscheidender Bedeutung sowie zum anderen das Vorhandensein von Funktionalitäten, die die unternehmensinterne Zusammenarbeit ermöglichen und erleichtern. Im Folgenden wird eines der kommerziellen Produkte, das sich entsprechend aus dem Kriterienkatalog für große Unternehmen ergeben hat, näher betrachtet und danach das Open-Source Potenzial in dieser Unternehmensgröße analysiert. Im Anschluss darauf werden zudem mögliche kommerzielle Substitutionsprodukte zum identifizierten Marktführer bewertet.

5.1.1 Aktueller Marktführer: IBM Rational DOORS

Der aktuelle globale Marktführer für Anforderungsmanagement mit einem Marktanteil von 51,9 % im Jahr 2012 ist IBM mit der Anforderungsmanagement-Lösung IBM Rational DOORS Next Generation, das vor allem größere Unternehmen verwenden.⁵⁸ Auch in dieser Arbeit, nach der Betrachtung der Matrix, wurde der etablierte Marktführer mit einer Erfüllung von 100 % in der Basic-Kategorie, 83,3 % in der Advanced-Kategorie und 65 % in der Optio-

⁵⁸ Vgl. IDC (2013), p. 4

nal-Kategorie aus allen vorgestellten Anforderungsmanagement-Softwarelösungen als das Produkt identifiziert, das sich am besten für die Zielgruppe der großen Unternehmen eignet. Weitere kommerzielle Produkte, die für den Bereich der großen Unternehmen empfohlen werden können, sind Quality Center von HP und Software Caliber von Borland, die in Kapitel 5.1.3 untersucht werden.

Rational DOORS stellt Projektteams Funktionen für das Definieren und Verwalten von Anforderungen, ein Arbeitselementsystem für die Aufgabenverwaltung und -planung und ein von einem Data Warehouse unterstütztes Berichterstellungssystem bereit. Dabei kann die Anwendung *Requirements Management (RM)* dazu verwendet werden, Anforderungen in einem Lebenszyklusentwicklungsprojekt zu definieren, zu verwalten und Berichte dazu zu erstellen.

Collaboration

Hervorzuheben ist der Punkt, dass es möglich ist, die oben erwähnten Funktionen im Web-Client mit der gesamten Stakeholder-Community durchzuführen. So können beispielsweise ganze Teams Anforderungen mithilfe von Attributen, Schlagwörtern, gefilterten Ansichten, angepassten Artefakt- und Linktypen, Änderungsindikatoren und Projekt-Dashboards verwalten.⁵⁹ Weitere Funktionalitäten, die die Zusammenarbeit in einem großen Mitarbeiterkreis erleichtern, sind Features wie zum Beispiel die Optionen:⁶⁰

- angepasste Dokumenttypen, Attribute, Datentypen, Linktypen und Projektschablonen zu erstellen
- Hintergrundinformationen, Stakeholder-Bedarf, Projektziele und andere Informationen in Rich-Text-Artefakten festzuhalten
- Links von Elementen in einem Artefakt zu anderen Projektartefakten zu erstellen oder auf externe Websites zu verlinken
- Anforderungssammlungen und -module mit Entwicklungs- und Testplänen zu verknüpfen
- Glossare zu definieren und Begriffe zu erstellen, die in Projektartefakte integriert werden

⁵⁹ Vgl. IBM Knowledge Center (o.J.a)

⁶⁰ Vgl. IBM Knowledge Center (o.J.a)

Besonderheiten

Weitere Optional-Features aus dem Kriterienkatalog, sind ebenfalls in der RM-Anwendung von IBM Rational DOORS integriert, wie zum Beispiel die Möglichkeit:⁶¹

- Anforderungsdateien aus CSV-Dateien und ReqIF-Dateien zu importieren
- automatische Berichte zu generieren
- Benutzeroberflächen mit Skizzen, Storyboards und Anzeigenflüssen zu simulieren

Integration

Neben der RM-Anwendung bietet IBM Rational DOORS zusätzliche Lösungen für Collaborative Lifecycle Management (CLM) an und kann mit Anwendungen wie Change and Configuration Management oder Quality Management kombiniert und eingesetzt werden, um die Arbeit von Analysten mit der von Entwicklungs- und Testteams zu verbinden. Eine CLM-Integration stellt eine einheitliche Lösung für Artefakt Verknüpfung, Dashboards, Sicherheit und Benutzerschnittstellen-Frameworks bereit.⁶²

Bewertung des Funktionsumfangs

Bei der skalierbaren, interaktiven Softwarelösung von IBM handelt es um ein umfassendes Angebot im Segment des Anforderungsmanagements. Elementare Funktionen neben den eigentlichen Anforderungsmanagement-Funktionalitäten, wie die Verfügbarkeit einer Testversion und verschiedenen Supportmöglichkeiten sind gegeben. Rational DOORS ist als eigenständiges Produkt zu erwerben zu einem relative hohen Preis von \$ 4.924 pro Jahr und Nutzer.

In Bezug auf die Nachteile von Rational Doors ist lediglich zu erwähnen, das keine mobile Zugriffsmöglichkeit integriert ist. Es ist derzeit auch keine Anbindung an die vorgestellten Produkte Salesforce, Zendesk, Jira, TFS möglich. Ansonsten werden hier alle Basic- und Advanced- Features abgedeckt. Zur Unterstützung von bestimmten Anmeldeverfahren, wie z.B. Google Sign-On, und gewissenen Anforderungsformaten, wie z.B. PRD und SRS liegen keine Informationen vor. Ansonsten werden alle relevanten Funktionalitäten in den Bereichen Anforderungsmanagement, Änderungsmanagement, Collaboration, Aufbau und Navigation, Administration, Zugang und Integration, Datenaustausch sowie Analyse erfüllt. Wie bereits hervorgehoben, sticht Rational DOORS vor allem im Collaboration heraus.

⁶¹ Vgl. IBM Knowledge Center (o.J.a)

⁶² Vgl. IBM Knowledge Center (o. J. b)

Zusammenfassend kann festgehalten werden, dass IBM Rational DOORS nahezu in allen Kategorien des Kriterienkatalogs alle Funktionalitäten – Basic, Advanced, als auch Optional – vorweisen kann. Neben den umfassenden Anforderungsmanagement-Features ist zudem die Möglichkeit zur individuellen Anpassung des Tools je nach Kundenwünschen hervorzuheben. Aus diesem Grund ist die Empfehlung für den Einsatz in großen Unternehmen, als auch der relative hohe Preis gerechtfertigt.

5.1.2 Betrachtung der Eignung der Open Source Produkte

Für große Unternehmen sind, neben den Basis- und Advanced-Funktionalitäten aus dem vorgestellten Kriterienkatalog, zusätzlich Funktionalitäten aus der Optional-Kategorie bedeutsam, die vor allem die Koordination von und die Zusammenarbeit innerhalb einer großen Anzahl von Mitarbeitern ermöglichen und erleichtern. Es zeigt sich jedoch, dass keine der vorgestellten Open Source-Softwarelösungen die 50 % Hürde der Optional-Kategorie erreicht, die in dieser Arbeit als Schwelle für die Eignung für größere Unternehmen festgelegt wurde. Bereits in der Advanced-Kategorie verfügt keines der Produkte die Hälfte aller Funktionalitäten; erst in der Basic-Kategorie finden sich Open Source-Lösungen, die die 50 % Hürde schaffen.

Neben allgemeinen klassischen Anforderungsmanagement-Funktionalitäten, die die Open Source-Softwarelösungen bereits nicht vorweisen können, mangelt es zudem an Funktionalitäten in den Kategorien „Collaboration“ und „Administration“. Diese zwei Kategorien sind von hoher Wichtigkeit für größere Unternehmen, da mit einer großen Anzahl an Mitarbeitern die Größe der Projektteams und die Anzahl der Projekte steigt. Wie bereits in Kapitel 4.2 erläutert, kann ein effektives und effizientes Anforderungsmanagement in dieser Größenordnung nur durch eine klare Administration und kollaborative Funktionen gewährleistet werden. Da keines der Open-Source Produkte die aufgelisteten Fähigkeiten in diesen Kategorien aufweist, können diese nicht für den Einsatz in großen Unternehmen empfohlen werden.

Ein Großteil der vorgestellten Produkte kann zudem nicht berücksichtigt werden, weil ihre Entwicklung nicht weiter vorangetrieben wird. Es kann nicht davon ausgegangen werden, dass in naher Zukunft eine der Lösungen so weit entwickelt ist, dass es für den Einsatz in großen Unternehmen geeignet ist. Aus diesem Grund ergeben sich keine nennenswerten Open-Source Alternativen zu dem vorgestellten kommerziellen Marktführer für die Zielgruppe der großen Unternehmen.

Nichtsdestotrotz bewertet IDC in einem Analysebericht zum globalen Anforderungsmanagement-Softwaremarkt von 2012 das Open Source Potenzial sehr hoch, mit der Begründung,

dass kommerzielle Anbieter zunehmend Open Source Komponenten in ihre Lösung integrieren, um die Softwareentwicklung zu beschleunigen, und Open Source Plattformen benutzen, wie z.B. SCM und andere ALM-Produkte. Demnach bleiben die Entwicklungen auf dem Markt von Open Source Lösungen zu verfolgen; zum jetzigen Zeitpunkt kann jedoch keine geeignete Alternative zu den kommerziellen Produkten aufgezeigt werden.⁶³

5.1.3 Mögliche Substitutionsprodukte für IBM Rational DOORS

Da sich im Bereich der großen Unternehmen keine mögliche Open Source Alternative findet, weil diese entweder nicht weiter entwickelt wurden oder schlicht nicht über die Funktionalitäten verfügen, die für große Unternehmen wichtig erscheinen, stellt dieses Kapitel mögliche kommerzielle Alternativen vor.

Die Produkte, welche neben dem Marktführer DOORS von IBM in Frage kommen, sind das Quality Center von HP und die Software Caliber von Borland. Diese Einschätzung ergibt sich aus den ermittelten Prozenten für beide Tools aus dem Kriterienkatalog. Das HP QC erreicht 83,3 % der Basic-Features, 87 % der Advanced-Features und 45 % der Optional-Features. Caliber verzeichnet 87,5 % der Basic-Features, 68,5 % der Advanced-Features und 48,3 % der Optional-Features. Beide Alternativen erreichen bei den Optional-Features die 50 % Hürde nicht, da sie diese Prozentzahl aber am besten erreichen, werden sie zu diesem Vergleich herangezogen. Wenn man die Prozentangaben für die drei Feature-Kategorien vergleicht, liegt Caliber in zwei der drei Bereiche vor dem HP QC. Das HP QC überholt Caliber nur in einer Kategorie. Um hier einen detaillierteren Vergleich zu ziehen, wird im Folgenden auf alle Bereiche des Kriterienkatalogs Rücksicht genommen.

HP QC und Caliber

Preisstruktur und allgemeine Kriterien

Preisinformationen liegen für das HP QC nicht vor. Für Caliber kostet eine Multi-User Lizenz 7.440€ im Monat. Eine Single-User Lizenz beträgt 2.380€ im Monat. In den sonstigen Kriterien im allgemeinen Bereich sind beide Produkte ähnlich gut aufgestellt. Caliber läuft auf dem Betriebssystem Windows, während das HP QC auf Windows und UNIX nutzbar ist. Für beide Produkte sind Testversionen verfügbar und es stehen Trainings in Form von Handbüchern, Web-Trainings und einem Telefonsupport zur Verfügung. In der Kategorie Anforderungsmanagement erfüllen beide Tools nahezu alle Kriterien vollständig. Abweichungen beim HP QC gibt es hier bei der Einbindung eines Glossars und der Abbildung eines agilen Anforderungsmanagements, da hierzu keine Informationen vorliegen. Caliber stellt keine Informatio-

⁶³ Vgl. IDC (2012)

nen zu integrierten Funktionen zur Anforderungssammlung bereit. Zwei weitere Kriterien werden hier mit „0“ bewertet. Im Hinblick auf die Gewichtung der Optional-Features lässt sich feststellen, dass Caliber für zwei Kriterien dieser Features keine Angaben macht („Integrierte Funktion für das Sammeln von Anforderungen“) und das HP QC für eine der Optional-Features Kategorie keine Informationen bereit hält („Einbindung eines Glossars“). Somit schneidet das HP QC hier besser ab.

Collaboration

In der Kategorie Collaboration haben beide Tools in je einem Bereich eine „0“ bekommen, während die übrigen Kriterien voll erfüllt werden. Beide Bereiche, die die Produkte nur teilweise erfüllen, werden den Optional-Features zugeschrieben. Beim HP QC muss der Anwender kleine Abstriche bei der automatisierten Kommunikation zu Kunden oder Stakeholdern machen. Caliber erfüllt das Teilen von Anforderungen nur teilweise, da hierfür erst eine URL generiert werden muss.

Aufbau und Navigation

Für die Kategorie Aufbau und Navigation erhalten beide Produkte ebenfalls in den meisten Bereichen ein „+“. Währenddessen steht dem Anwender im HP QC keine Tabellenansicht zur Verfügung. Ob eine Dokumentenansicht verfügbar ist, lässt sich für Caliber nicht ermitteln. Außerdem hat Caliber keine integrierte Roadmap-Ansicht, sondern nur eine Diagramm-Ansicht, daher erhält es hier nur ein „teilweise erfüllt“. Dieses Kriterium wird in der Gewichtung den Optional-Features zugeschrieben, wodurch das HP QC auch in dieser Kategorie besser abschneidet, da es hier alle Funktionen erfüllt, die den Optional-Features zugeschrieben werden.

Weitere Funktionen

In der Kategorie Administration erfüllen beide Produkte alle ermittelten Kriterien. Ein stark beworbenes Feature für das HP QC ist das integrierte Änderungsmanagement. Für diese Kategorie liegen für Caliber keine Informationen vor. Die nächste Kategorie ist Zugang und Integration. Hier erfüllen beide Tools einige Kriterien nicht. So sind kein mobiler Zugriff und kein Google Sign-on für das HP QC möglich. Außerdem gibt es ein paar Produkte mit denen sich die Software nicht verknüpfen lässt. Auch Nutzer von Caliber können nicht mobil auf die Software zugreifen. Für die unterstützten Anmeldeverfahren sowie zu einigen Produkten, mit denen sich andere Anforderungsmanagement-Tools verknüpfen lassen, liegen keine Informationen vor. Des Weiteren erlaubt Caliber kein integriertes Test- und Defect-Management, was im HP QC eine mögliche Option ist. Die Erfüllung der Optional-Features verläuft auch in dieser Kategorie zu Gunsten des HP QCs, da beide Produkte zwar keinen mobilen Zugriff erlauben und wenig Angaben zur Verknüpfung mit anderen Produkten vorliegen, doch liegen

für das HP QC Informationen zu den unterstützten Anmeldeverfahren vor, was im Falle von Caliber nicht gegeben ist.

Datenaustausch

In der Kategorie Datenaustausch müssen die Anwender ebenfalls einige Abstriche bei beiden Produkten machen. Die Datenformate HTML, RTF, CSV und PDF unterstützt das HP QC nicht. Caliber unterstützt die Formate HTML, RTF und PDF nicht. Die Formate CSV und Plain Text können nur exportiert werden. Über die unterstützten Änderungsformate liegen für beide Produkte keine Informationen vor. In Hinblick auf die Gewichtung der Optional-Features hat das HP QC im Kriterium des Plain Text Datenaustauschs einen Vorteil gegenüber Caliber, da dieses vom HP QC erfüllt wird und von Caliber nicht. Die restlichen Kriterien der Optional-Features sind bei beiden gleich bewertet worden.

Analyse

Die Kategorie Analyse wird von Caliber vollständig erfüllt. Das HP QC erfüllt das Kriterium automatisierte Erstellung grafischer Analysen nur teilweise. Die übrigen Kriterien werden auch vom HP QC vollständig erfüllt. Hier lässt sich festhalten, dass Caliber bei den Optional-Features besser bewertet wird, da die Erstellung grafischer Analysen zu den Optional-Features gezählt wird.

Zusammenfassend lässt sich feststellen, dass beiden Produkte in den einzelnen Kategorien und Kriterien ziemlich vergleichbar sind. Vereinzelt gibt es Kriterien, die nicht oder nur teilweise erfüllen werden, doch in der Gesamtheit werden die meisten Bereiche von beiden Tools abgedeckt. In Bezug auf die Verfügbarkeit von Optional-Features hat das HP QC dennoch etwas besser abgeschnitten als Caliber, da es in vier Kategorien mehr Optional-Features erfüllt als Caliber. Caliber hingegen wird in einer Kategorie hinsichtlich der Optional-Features besser bewertet. Bezüglich der Kosten lässt sich festhalten, dass für das HP QC keine Informationen vorliegen und das Produkt von Borland für große Unternehmen mit 7.440€ im Monat für eine Multi-User Lizenz erschwinglich erscheint. Da somit kein Preisvergleich erfolgen kann, gerade dieses Kriterium für Unternehmen jeglicher Größe jedoch eine wichtige und oft ausschlaggebende Information ist, wird die Software Caliber im Hinblick auf den Preis den großen Unternehmen empfohlen.

Zusätzlich sollte an dieser Stelle noch einmal das Ergebnis der Gewichtung im Kriterienkatalog beachtet werden. Caliber hat in der Auswertung der Optional-Features 48,3% erhalten, knapp vor HP QC, welches 45% erlangt hat. Da somit insgesamt mehr Optional-Features von Caliber erfüllt werden, kann auch hier eine Empfehlung für dieses Produkt ausgesprochen werden. Es lässt sich festhalten, dass Caliber eine gute Alternative zu DOORS von IBM im kommerziellen Bereich darzustellen scheint.

5.2 Mittlere Unternehmen

Die Fähigkeiten, die Kapitel 4.3 zufolge insbesondere für mittlere Unternehmen ausschlaggebend sind, liegen unter anderem in erweiterten Collaboration und Administrations-Features, da Mitarbeiterzahlen zwischen 50 und 250 bereits größere Projektgruppen bedingen. Gegenüber den Werkzeugen für große Unternehmen sind weniger Analysemöglichkeiten, Zugangsmöglichkeiten und Dateiformate gefragt, da diese zu einer Unübersichtlichkeit und weniger intuitiven Benutzeroberfläche führen können. Des Weiteren spielt insbesondere auch der Preis eine große Rolle, da jede Kostenaufwendung einen belastenden Faktor für die Unternehmen darstellt.

Im folgenden Kapitel sollen daher die identifizierten Software-Lösungen für mittelständige Unternehmen näher betrachtet werden und auf ihr Potenzial geprüft werden. Dazu wird im ersten Schritt begründet, nach welchen Kriterien ein Marktführer identifiziert wurde. Analog zu Kapitel 5.1 erfolgt im Anschluss die Bewertung des Open Source Potenzials sowie die Betrachtung kommerzieller Alternativen zum ermittelten Marktführer.

5.2.1 Aktueller Marktführer: OneDesk

Als Marktführer im Bereich der mittleren Unternehmen wurde in dieser Arbeit OneDesk Requirements Management von OneDesk Inc. ausgewählt. Andere kommerzielle Produkte für den Bereich der mittleren Unternehmen sind Accompa und Polarion, die in Kapitel 5.2.3 analysiert werden.

OneDesk erfüllt bei der Bewertung 100 % der Basic-Features, 70 % der Advanced-Features und 58 % der Optional-Features. Trotz der durch diese Bewertung möglichen Einteilung in den Bereich der großen Unternehmen, wurde OneDesk aufgrund verschiedener Punkte den mittleren Unternehmen zugeteilt. Diese Entscheidung wurde bereits in Kapitel 4.3 begründet. Im Folgenden wird nun erläutert, welche Gründe zu der Identifikation von OneDesk als Marktführer geführt haben.

Skalierbarkeit & Flexibilität

OneDesk hebt sich bereits in seinen allgemeinen Charakteristika von anderen Produkten ab. Besonders erwähnenswert sind hierbei der SaaS-basierte Aufbau, das einfach skalierbare und kostengünstige Preismodell sowie die Integration des Anforderungsmanagement in einer ganzheitlichen Lösung zum Produkt- und LifeCycle-Management. Mittelgroße Unternehmen haben dadurch nicht nur die Möglichkeit, das Lizenzmodell schnell und einfach an sich verändernde Rahmenbedingungen anzupassen, sondern können gleichzeitig das gesamte Entwicklungs- und Produktmanagement in einer Lösung verwalten.

Integration

Um den Umstieg und die Integration eventuell bereits bestehender Applikationen zu vereinfachen, stellt OneDesk standardmäßig eine große Anzahl von Schnittstellen bereit. So können z.B. Anwendungen wie Salesforce.com, Jira, Zendesk und TFS einfach über bestehende Konnektoren integriert werden. Für die maßgeschneiderte Anbindung an weitere Produkte ist darüber hinaus eine öffentliche API verfügbar.

Bewertung des Funktionsumfangs

Im Bereich des eigentlichen Anforderungsmanagement muss OneDesk lediglich Abstriche beim sogenannten Baselineing machen. Ansonsten werden hier alle Basic- und Advanced-Features abgedeckt. Weiterhin werden alle relevanten Punkte im Bereich Collaboration, Administration sowie Änderungsmanagement erfüllt. Wie bereits zum Teil erwähnt, grenzt sich OneDesk zudem im Bereich Zugang und Integration durch die hohe Kompatibilität mit anderen Applikationen sowie die breite Unterstützung gängiger Authentifizierungsverfahren wie SAML, OAUTH2, SSO und Google Sign-On von anderen Produkten ab. Leichte Schwächen in den für mittlere Unternehmen relevanten Kriterien zeichnen sich lediglich in den Bereichen Aufbau und Navigation sowie Datenaustausch ab. Während beim Aufbau fehlende Baum- und Dokumenten-Ansichten zu bemängeln sind, fällt im Bereich Datenaustausch die mangelnde Unterstützung von Microsoft Word und RTF sowie aller gängigen Anforderungs-Formate negativ auf.

Zusammenfassend kann festgehalten werden, dass sich OneDesk vor allem durch sein hohes Maß an Skalierbarkeit, die breite Integration anderer Applikationen sowie sein Lizenzmodell auszeichnet. Bei der Betrachtung der weiteren Faktoren führt lediglich die fehlende Unterstützung gängiger Datenformate zu Abzügen bei den ansonsten durchgängig hohen Punktzahlen bei der Bewertung. Die Gesamtheit der oben genannten Faktoren führt dazu, dass OneDesk in dieser Arbeit als Marktführer den mittleren Unternehmen empfohlen wird.

In den folgenden Kapiteln werden nun Alternativen aus dem Open Source Segment sowie dem kommerziellen Bereich vorgestellt und analysiert.

5.2.2 Betrachtung der Eignung der Open Source Produkte

Um geeignete Open Source Produkte zu ermitteln, welche sich für den Einsatz in Unternehmen mittlerer Größe eignen, erfolgt zunächst eine nähere Betrachtung der zuvor identifizierten und im Kriterienkatalog aufgelisteten Produkte. Dabei wird untersucht, welche Produkte neben den Basic-Features auch mindestens 50 % der erweiterten Features erfüllen. Wie bereits erläutert, liegt diesem Verfahren die Annahme zugrunde, dass Unternehmen mittlerer

Größe höhere Ansprüche an ein Anforderungsmanagement-Tool haben, als dies bei kleinen Unternehmen der Fall ist.

Zu den erweiterten Features zählen beispielsweise Funktionen wie die Abbildbarkeit agiler Softwareentwicklung oder auch Grundfunktionen im Bereich Collaboration, wie eine Kommentarfunktion oder auch eine integrierte Community. Zudem wird davon ausgegangen, dass Unternehmen mittlerer Größe im Gegensatz zu kleinen Unternehmen auch Funktionen wie eine integrierte Benutzer- und Rechteverwaltung und ein integriertes Änderungsmanagement benötigen.

Untersucht man den Kriterienkatalog nun unter diesen Gesichtspunkten, muss festgestellt werden, dass keines der Open Source Produkte über mindestens 50% der erweiterten Funktionen verfügt und somit den Ansprüchen gerecht wird. Die im Rahmen dieser Arbeit untersuchten Open Source Produkte erfüllen im Durchschnitt gerade einmal 26 % der erweiterten Funktionen.

Die erfolgreichsten Produkte im Bereich der erweiterten Funktionen sind mit jeweils 33 % aNimble der Firma ideaStub, sowie ProR der Firma Formal Minds. Da aNimble im Bereich der Basis-Funktionen immerhin 58 % und ProR 54 % erzielen konnten, soll untersucht werden, ob diese zumindest mit Einschränkung für Unternehmen mittlerer Größe empfohlen werden können.

ProR – Formal Mind GmbH

Das Produkt ProR der Firma Formal Mind GmbH zeigt jedoch auch bei wichtigen Funktionen klare Defizite, da beispielsweise keine Benutzer- und Rechteverwaltung möglich ist. Da jedoch davon ausgegangen werden kann, dass in mittelgroßen Unternehmen auch größere Entwicklungsteams mit der Software arbeiten sollen, ist diese Funktion unverzichtbar. Ebenso werden weder Kommentarfunktion noch automatische Berichte Erstellung unterstützt, um weitere Beispiele für wichtige Advanced-Features zu nennen. Aufgrund der Defizite im Funktionsumfang kann ProR zum jetzigen Zeitpunkt nicht für mittelgroße Unternehmen empfohlen werden und soll stattdessen im späteren Verlauf der Arbeit für kleine Unternehmen in Betracht gezogen werden.

aNimble - ideaStub

Das Tool aNimble von ideaStub scheint zunächst über die wichtigsten der erweiterten Features zu verfügen. So sind beispielsweise eine Kommentarfunktion, eine eingeschränkte Rechteverwaltung und auch die Abbildung agiler Softwareentwicklung möglich. Allerdings wurde aNimble zuletzt 2012 aktualisiert und es gibt auch keine Entwickler-Community, welche dieses Produkt weiterentwickelt, sondern lediglich eine Firma, die ihre Arbeit an dem Tool jedoch eingestellt zu haben scheint. Aus diesem Grund kann auch aNimble nicht für

den Einsatz in mittelgroßen Unternehmen empfohlen werden, da seitens des Anbieters keinerlei Support besteht und beispielsweise auch sämtliche im Programmcode enthaltenen Fehler selbst ausgebessert werden müssten. Insbesondere das Bugfixing würde jedoch einen erheblichen zeitlichen Aufwand bedeuten, wodurch sich auch der Umstand nicht rechnet, dass die Software zunächst kostenfrei verfügbar ist.

Ausgehend von diesen Erkenntnissen muss festgestellt werden, dass die Open Source Lösungen aktuell unzureichend entwickelt sind, um für Unternehmen mittlerer Größe eine ernsthafte Alternative zu kommerzieller Software darzustellen. Bedingt durch Defizite im Funktionsumfang und veraltete Versionen, die nicht mehr aktualisiert werden, kann mittelgroßen Unternehmen derzeit nicht zum Einsatz von Open Source Software geraten werden.

5.2.3 Mögliche Substitutionsprodukte für OneDesk

Da, wie im vorangegangenen Kapitel erläutert, keine der am Markt verfügbaren Open Source Lösungen für Unternehmen mittlerer Größe geeignet ist, sollen im Folgenden zwei kommerzielle Alternativen zu OneDesk vorgestellt werden, welche sowohl vom Funktionsumfang als auch preislich für mittelgroße Unternehmen attraktiv sind. Zur Auswahl der Alternativen wurden verschiedene Kriterien berücksichtigt, insbesondere das Abschneiden der Produkte bei den erweiterten Features des Kriterienkataloges, sowie der Preis, weil davon ausgegangen wurde, dass mittelgroße Unternehmen keine allzu großen Investitionen in ein Anforderungsmanagement-Tool tätigen möchten.

RM - Polarion

Die Auswahl des ersten Substitutionsproduktes für OneDesk gestaltet sich mit Verweis auf den Kriterienkatalog sehr einfach. Setzt man bei der Auswahl den Fokus zunächst ausschließlich auf das Abschneiden der einzelnen Produkte im Bereich der erweiterten Features, so erkennt man, dass das Tool RM der Firma Polarion hier mit 88,9 % der Spitzenreiter ist. Da RM ebenfalls 100 % der Basis-Funktionen erfüllt, scheint dieses Produkt eine sehr geeignete Alternative darzustellen.

Preis-Struktur

Nachdem das Produkt allein vom Funktionsumfang attraktiv für mittelgroße Unternehmen ist, muss in einem nächsten Schritt der Preis geprüft werden, da viele der Produkte mit umfangreichen Funktionen auch einen entsprechenden Preis haben, welcher für Unternehmen mittlerer Größe nicht finanzierbar wäre. Polarion gibt den Preis für das Tool RM auf der Unter-

nehmenswebsite offen an: \$ 1.490 für eine personalisierte Lifetime Licence.⁶⁴ Vergleicht man dieses Angebot mit denen anderer Hersteller, so zeigt sich sehr schnell, dass Polarion verhältnismäßig günstig ist. Somit sprechen schon einmal die beiden Hauptfaktoren für einen möglichen Einsatz von RM in mittelgroßen Unternehmen und das Tool kann im Folgenden genauer auf seine Eignung überprüft werden.

Flexibilität & Skalierbarkeit

Zunächst zeichnet sich Polarion RM durch seine Flexibilität und Skalierbarkeit aus. Polarion bietet eine ganze Reihe möglicher Lizenzformen und bietet RM vor allem auch als On Premise Lösung oder auch über ein Cloud Hosting an. Somit kann Polarion RM in jedem Unternehmen unabhängig von deren individueller IT-Strategie eingesetzt werden. Zudem bietet Polarion sowohl personenbezogene als auch mehrfach nutzbare Lizenzen an. Somit kann die Software ideal an die Bedürfnisse des jeweiligen Unternehmens angepasst werden.

Support

Ebenso weist Polarion im Supportbereich ein breites Angebot auf. Neben dem üblichen Telefon- und Websupport werden sowohl Online- als auch Onsite-Trainings angeboten, wodurch beispielsweise bei der Einführung des Produkts im Unternehmen auf den Einsatz einer zusätzlichen Beratungsfirma, welche den Schulungsteil übernimmt, verzichtet werden kann und alles aus einer Hand kommt.

Erweiterte Funktionen

Neben den Grundfunktionen besitzt RM auch eine Vielzahl an erweiterten Funktionen wie beispielsweise die Abbildbarkeit agiler Softwareentwicklung, projektübergreifende Verlinkungen oder ein integriertes System zur Versionskontrolle. Ebenso bietet Polarion RM Collaboration-Funktionen, sodass auch größere Entwickler-Teams, wovon in mittelgroßen Unternehmen auszugehen ist, effektiv zusammenarbeiten können.

Aufbau & Administration

Ebenso enthält die Software umfangreiche Werkzeuge zur Suche und Filterung von Anforderungen und kann die Anforderungen in den meisten Ansichten darstellen. Gleichmaßen sind umfangreiche Administrationsfunktionen vorhanden, sodass klar geregelt werden kann, welcher Benutzer welche Rechte erhält und welche Inhalte er dadurch sehen kann. Diese Flexibilität ist gerade für mittelgroße Unternehmen wichtig, die planen, in Zukunft zu expandieren. Derlei Funktionen können demnach sehr schnell relevant werden, sobald zusätzliche Entwickler dem Team beitreten.

⁶⁴ Vgl. Polarion (o.J.a)

Zugang

Zusätzlich kann sowohl per Web- als auch per Desktop-Zugang auf die Software zugegriffen werden, was ebenfalls wieder einen Zugewinn an Flexibilität bedeutet. Auch was den Datenaustausch betrifft, unterstützt Polarion RM alle gängigen Formate. Darüber hinaus bietet das Tool eine Reihe an automatischen Analyse-Möglichkeiten, was eine sehr große Zeitersparnis bedeuten kann, wenn beispielsweise Berichte nicht manuell erstellt werden müssen.

Erweiterungen

Ein weiterer großer Vorteil ist das Modell von Polarion, zusätzliche Funktionen durch Extensions anzubieten.⁶⁵ Im Rahmen dieser Extensions bietet Polarion erweiterte Funktionen an, beispielsweise in den Bereichen Workflow-Management, Reporting, Test Automation oder agile Softwareentwicklung, sodass die Software einfach an die individuellen Bedürfnisse jedes Unternehmens angepasst werden kann. Dies spart dem jeweiligen Betrieb sowohl zeitliche als auch finanzielle Aufwände, die fehlenden Funktionen gegebenenfalls selbst zu entwickeln und diese dann behelfsmäßig an eine Software anzubinden. Im Gegensatz dazu rechnet sich der Einsatz vorgefertigter Extensions sehr schnell. Hinzu kommt der Faktor, dass die Erweiterungen sofort verfügbar sind und nicht zuerst entwickelt werden müssen.

Somit kann Polarion RM in vollem Maße für mittelgroße Unternehmen empfohlen werden und stellt eine gute Alternative zu OneDesk dar. Für Unternehmen gilt es lediglich, die jeweiligen Vor- und Nachteile der Tools gegeneinander abzuwägen, um dadurch für das eigene Unternehmen das Beste zu bestimmen. Obwohl bereits zwei gute Produkte für den Einsatz in mittelgroßen Unternehmen gefunden wurden, soll im Folgenden noch ein drittes Produkt vorgestellt werden, um die Vielfalt der am Markt verfügbaren Alternativen zu verdeutlichen.

Accompa – Accompa Inc.

Eine weitere Lösung, die im Umfeld mittlerer Unternehmen eingesetzt werden kann und damit gleichzeitig als Substitutionsprodukt für OneDesk in Frage kommt, ist Accompa von Accompa Inc. Betrachtet man zunächst das Ergebnis der Nutzwertanalyse sowie allgemeine Charakteristika von Accompa, zeichnet sich bereits hier eine potentielle Eignung der Anwendung als Alternative zu OneDesk und Polarion RM ab.

Accompa erreicht bei der Bewertung mithilfe des Kriterienkataloges rund 83 % der Basic-, 55 % der Advanced- und 46 % der Optional-Features. Obwohl Accompa damit hinter den beiden Produkten OneDesk und Polarion RM zurück bleibt, erscheint eine prinzipielle Eignung durchaus als gegeben.

⁶⁵ Vgl. Polarion (o.J.b)

Preis - Struktur

Ein Blick auf das Preis- und Lizenzmodell zeigt eine drei-stufige Struktur, die sich wie folgt gliedert: Standard für \$ 199 im Monat, Corporate zu \$ 399 je Monat und Enterprise für \$ 799 pro Monat. Alle Versionen enthalten fünf Nutzer-Lizenzen. Wie erwähnt können weitere Nutzer für \$ 29 pro Nutzer und Monat in der Standard Version, bzw. \$ 39 und \$ 79 pro Nutzer und Monat in der Corporate beziehungsweise Enterprise Version hinzugefügt werden. Aufgrund der Einschränkungen der Standard Version ist für Unternehmen mittlerer Größe mindestens die Corporate Version zu empfehlen. Accompa ist damit von den drei in diesem Abschnitt behandelten Produkten das teuerste. Vor allem die hohen Kosten für zusätzliche Nutzer können bei wachsenden Teams schnell zu einer Multiplikation der Kosten führen. Trotz dieser Gefahr bietet dieses Modell insbesondere für kleine Gruppen eine ideale Basis und erlaubt ein hohes Maß an Flexibilität im Falle sich ändernder Umstände. Obwohl Accompa in diesem Vergleich das teuerste Produkt ist, bleibt es doch mit seinen Preisen hinter Produkten für große Unternehmen zurück und ist von diesem Gesichtspunkt her durchaus für Unternehmen mittlerer Größe geeignet.

Funktionsumfang

Der Funktionsumfang von Accompa im Bereich Anforderungsmanagement ist vergleichbar mit dem der beiden Konkurrenzprodukte. In diesem Punkt lassen sich daher weder Vor- noch Nachteile finden, die für beziehungsweise gegen eine Nutzung von Accompa sprechen. Punkten kann Accompa im Bereich Collaboration. Als einziges aller analysierten Tools erreicht Accompa in diesem Bereich 100 % in der Bewertung. Im Bereich Aufbau und Navigation werden ähnliche Punktzahlen wie bei Polarion RM erzielt. Accompa hat dadurch einen leichten Vorteil gegenüber OneDesk, das in diesem Bereich kleinere Abstriche hinnehmen muss. Im Bereich Administration ergeben sich keine entscheidenden Unterschiede im Vergleich mit den anderen Produkten. Alle wesentlichen Kriterien werden erfüllt. Im Abschnitt Änderungsmanagement fällt Accompa hinter die beiden Vergleichsprodukte zurück. Während sowohl OneDesk als auch Polarion RM ein integriertes Änderungsmanagement anbieten, fehlt diese Komponente bei Accompa gänzlich. Im Bereich Zugang und Integration zeigen sich sowohl Vorteile als auch Nachteile von Accompa. Während Accompa als einzige Anwendung alle drei untersuchten Zugangsmöglichkeiten (Desktop, Web, Mobile) bereitstellt und sich außerdem durch eine breite Unterstützung gängiger Anmeldeverfahren sowie die Verfügbarkeit einer öffentlichen API auszeichnet, sind nativ keine Schnittstellen mit anderen Applikationen verfügbar. Im Bereich Datenaustausch kann sich Accompa durch die Unterstützung von drei der vier Anforderungs-Formate von den anderen beiden Produkten abheben. Der Bereich Analyse wird aufgrund fehlender Daten sowie niedriger Relevanz für mittlere Unternehmen an dieser Stelle nicht weiter betrachtet.

Zusammenfassend lässt sich sagen, dass Accompa in vielen Bereichen einen ähnlichen Funktionsumfang wie OneDesk und Polarion RM bietet. Die Entscheidung für oder gegen eines der drei Produkte hängt daher von den individuellen Anforderungen des jeweiligen Unternehmens ab. Generell kann Accompa für mittlere Unternehmen durchaus als Alternative zu OneDesk und PolarionRM empfohlen werden. Die hohen Kosten für zusätzliche Nutzer führen allerdings zu einer Einschränkung dieser Empfehlung und lassen Accompa bei größeren Nutzerzahlen zunehmend hinter OneDesk und Polarion RM zurückfallen.

Nachdem die Bereiche der großen sowie der mittleren Unternehmen nun eingehend untersucht wurden, sollen im Folgenden nun noch Produkte, die v.a. für kleine Unternehmen geeignet sind, analysiert werden.

5.3 Kleine Unternehmen

Unter Kapitel 4 wurde bereits dargestellt, welche Fähigkeiten in Bezug auf Anwendungsmanagement-Software besonders entscheidend für kleine Unternehmen sind. Es zeigt sich, dass hier weniger Features für die unternehmensinterne Zusammenarbeit nötig sind, da in der Regel kein großer Mitarbeiterkreis an dem Anforderungskatalog beteiligt ist. Stattdessen sollte sich die Software durch einen günstigen Preis bei Abdeckung der Mindestanforderungen für Anforderungsmanagement auszeichnen. Im Folgenden soll eines der kommerziellen Produkte, das sich entsprechend aus dem Kriterienkatalog ergeben hat, analysiert werden und im Anschluss mit einer denkbaren Alternative aus dem Open-Source Segment verglichen werden. Auf diese Art und Weise ist am Ende die Bewertung des Open-Source Potentials möglich.

5.3.1 Aktueller Marktführer: CASE Spec

Der in dieser Arbeit ausgewählte Marktführer der kleinen Unternehmen ist das Produkt CASE Spec von Goda. Andere kommerzielle Produkte, die für kleine Unternehmen in Frage kämen sind ReMa oder Integrity. Allerdings liegen zu beiden Produkten auch auf Nachfrage keine Preisangaben vor, weshalb ihre breite Basic-Feature-Abdeckung hier nicht berücksichtigt werden kann.

Die Software erfüllt 11 der 12 Basic-Features, was einer Übereinstimmung von 91,7 % entspricht. Weiterhin sind 21 der 27 Advanced-Features gegeben. Somit käme eine Einordnung in die mittlere Unternehmensgröße in Betracht.

Preis- & Lizenzmodell

Mit Hinblick auf das Preismodell lässt sich die Software jedoch eher den kleinen Unternehmen zuordnen. Eine on-premise Lizenz kostet im Monat 1.000\$.⁶⁶ Dies bedeutet, dass die Software auf den Servern des Kunden installiert wird und der Kunde die Software eigenständig anwendet. Der Anbieter stellt dem Kunden Upgrades zu Verfügung, damit die Software stets auf dem neuesten Stand bleibt. Ein anderes Modell der Softwarenutzung ist das off-premise Modell. Hier kauft der Kunde eine oder mehrere Lizenzen und die Software ist aus der Cloud abrufbar. Es ist hier somit nicht nötig, eigene Server oder Infrastruktur für Hardware zur Verfügung zu haben, um dieses Tool zu nutzen. Auch dieses Modell bietet Goda seinen Kunden bei dem Vertrieb von CASE Spec an. Drei Lizenzen kosten in diesem Fall 129\$ im Monat.⁶⁷ Diese Alternative ist sehr kostengünstig, da eine Lizenz im Monat für 43\$ zu haben ist. Dieser Preis ist für kleine Unternehmen erschwinglich und stellt eine gute Alternative zu üblichen Methoden in kleinen Unternehmen dar: So ist es in kleinen Unternehmen weit verbreitet, ein einfaches Excel-Sheet anzulegen, Spalten nach Belieben zu benennen und auf diese Weise die Anforderungen zumindest zu dokumentieren. Eine richtige Lösung des Anforderungsmanagement bietet diese Methode jedoch nicht. Betrachtet man nun die verfügbaren Features in CASE Spec, wird klar, dass es sich hier um ein solides Angebot im Segment des Anforderungsmanagements handelt, das viele wichtige und elementare Funktionen beherrscht.

Funktionsumfang von CASE Spec

So ist eine Testversion verfügbar und Goda bietet seinen Kunden verschiedene Supportmöglichkeiten. In der Kategorie „Anforderungsmanagement“ erfüllt das Produkt alle Basic-Features und darüber hinaus noch viele Advanced-Features, wie zum Beispiel das Anhängen von unterstützenden Dokumenten. Ähnlich gut erscheint die Einschätzung der Kriterien in den Kategorien „Collaboration“ und „Aufbau und Navigation“. In dem Bereich „Administration“ wird eine Schwachstelle des Produktes ersichtlich, da CASE Spec über keine Administratorsicht mit Informationen zu Systemstatus, Projekten oder Nutzern verfügt. In der Kategorie „Änderungsmanagement“ werden die Kriterien den Advanced-Features zugeschrieben und diese durchweg von CASE Spec erfüllt, was zusammen mit dem insgesamt hohen Anteil an Advanced-Features darauf hindeutet, dass die Software teilweise auch für mittlere Unternehmen geeignet wäre. Auch in den Kategorien „Zugang und Integration“ und „Datenaustausch“ werden alle Basic-Features erfüllt. Im Bereich „Analyse“ konnte kein Basic-Feature identifiziert werden. Dennoch sind hier das automatische Erstellen von Berichten und grafischen Analysen, sowie eine automatische Erstellung von Test-Cases basierend auf den Anforderungen möglich.

⁶⁶ Vgl. Goda Software Company (o.J.)

⁶⁷ Vgl. Goda Software Company (o.J.)

Auf Grund des preiswerten Angebots und den verfügbaren Features, welche in hohem Maße den Basic-Features und den Advanced-Features zugeordnet werden können, wird das Produkt CASE Spec **kleinen Unternehmen** als Marktführer empfohlen.

5.3.2 Analyse einer möglichen Alternative aus dem Open Source Segment

Aufgrund der Tatsache, dass viele der Open-Source Produkte ausscheiden, da ihre Entwicklung nicht weiter vorangetrieben wird, ergeben sich nur eine geringe Anzahl möglicher Alternativen zu den kommerziellen Produkten. Es zeigt sich, dass gerade die Produkte OSRMT beziehungsweise aNimble von ideaStub viele der Grundfähigkeiten einer Anforderungsmanagement-Software aufweisen, allerdings sind beide Lösungen nicht weiter in der Entwicklung. Gleichermaßen verhält es sich mit dem Ansatz Unicase der TU München, der allerdings auch seit 2009 nicht weiter verfolgt wird. Daher ergibt sich für kleine Unternehmen nur eine Möglichkeit aus der Software ProR, die von Formal Minds aufgebaut wird. Am letzten Update, das Mitte April 2014 veröffentlicht wurde, zeigt sich, dass aktuell an der Lösung gearbeitet wird und diese entsprechend weiterentwickelt wird. Im Folgenden soll daher diese Software Lösung näher betrachtet werden, um einen Vergleich zu CASE Spec zu ermöglichen.

ProR – Formal Mind GmbH

ProR wurde erstmals im Rahmen einer Doktorarbeit von Michael Jastram entwickelt. Genau genommen stellt es lediglich die graphische Benutzeroberfläche für das zugrundeliegende „Requirements Modeling Framework“ (RMF) dar, das als Open Source Projekt von der Eclipse Community entwickelt wurde⁶⁸.

Entwicklungshintergründe der Software

RMF bildet die Grundlage, die es ermöglicht, mit Anforderungen zu arbeiten, die auf dem Requirements Interchange Format (ReqIF, früher: RIF) Modell beruhen, einem XML-Dateiformat⁶⁹. ReqIF hat sich dabei in den vergangenen Jahren vor allem durch die deutsche Automobilbranche zu einem Standard für den Austausch von Anforderungen entwickelt⁷⁰. Zusammenfassend lässt sich also sagen, dass die Software ProR die Benutzeroberfläche für das Anforderungsmanagement Framework RMF darstellt, dessen Logik auf dem ReqIF Modell beruht. ProR wurde 2009 im Rahmen des EU FP7 Projektes „Deploy“ initiiert, dessen Ziel es war, die Interoperabilität industrieller Werkzeuge im Anforderungsmanagement zu

⁶⁸ Vgl. Eclipse Foundation (2014 a)

⁶⁹ Vgl. HIS (2009)

⁷⁰ Vgl. Jastram, M. (2010), S. 3

ermöglichen⁷¹. Treibende Kraft für die stetige Weiterentwicklung der Lösung ist dabei die Formal Mind GmbH aus Düsseldorf, die eine Ausgründung der Heinrich-Heine Universität Düsseldorf darstellt⁷². Die Geschäftsleitung hat der Urheber der ProR Plattform, Michael Jastram, inne, der weiterhin aktiv in dem Projekt involviert ist.

Besonderheiten der Software

Hintergrund des Projektes war dabei das Ziel, ein Werkzeug bereitzustellen, das sich durch Interoperabilität und Erweiterbarkeit auszeichnet. Entwickler Jastram war sich dabei bewusst, dass die Systembeschreibung eine große Herausforderung im Bereich des Anforderungsmanagements darstellt. Aus diesem Grund hat er nach einem Ansatz gesucht, der eine konsistente Systembeschreibung erlaubt, „die aus einem Initialen Satz von Anforderungen entwickelt wird“⁷³. Die Zielgruppe der entstandenen Lösung liegt dabei sowohl im akademischen als auch industriellen Umfeld und soll der eigenen Aussage nach all denen dienen, für die Word oder Excel nicht mehr ausreicht. Die Software präsentiert sich mit den Grundvoraussetzungen eines Anforderungsmanagements-Werkzeuges – so können Anforderungen angelegt werden, mit Attributen wie „implementiert“, „in Arbeit“ beziehungsweise „nicht implementiert“ versehen werden und auch Verlinkungen zwischen einzelnen Anforderungen hergestellt werden⁷⁴. Eine Besonderheit des Produktes liegt in der Erweiterbarkeit durch „Presentations“: Eclipse Plug-Ins, die es ermöglichen ProR beliebig zu erweitern. Die Lösung beinhaltet standardmäßig bereits erst Presentations, die zum Beispiel einen Zeilenumbruch erlauben⁷⁵. Da ProR auf diese Art und Weise viele Anpassungsmöglichkeiten bietet, könnten Firmen sich die Software mit dem nötigen Wissen in Java Programmierung ihren Bedürfnissen anpassen.

Funktionsumfang der angebotenen Software-Pakete

Die Software ist über die Website von Formal Mind für die Betriebssysteme Linux, Mac und Windows abrufbar; die Installation ist in 7 einfachen Schritten beschrieben⁷⁶. Darüber hinaus ist ein Video verfügbar, das zunächst die Installation darstellt⁷⁷. Es handelt sich dabei um ein Paket aus der Basis Software ProR und der Erweiterung ProR Essentials. Diese Erweiterungen umfassen zum einen eine Rich Text-Unterstützung, sodass Texte im XHTML Format formatiert angezeigt werden können und gleichzeitig das Editieren mit einem grafischen Editor ermöglicht wird. Außerdem bietet die Erweiterung das Feature „ReqIF Diff“, das einen grafischen Vergleich zweier Dateien möglich macht und damit die Historie der Anforderun-

⁷¹ Vgl. Deploy-Projct (o. J.)

⁷² Vgl. Jastram, M. (o. J.)

⁷³ Jastram, M. (2012), S. 7

⁷⁴ Vgl. Eclipse Foundation (2014 b)

⁷⁵ Vgl. Stelzer, J. u.a. (2014)

⁷⁶ Vgl. Formal Mind (2014 a)

⁷⁷ Vgl. Jastram, M. (2012)

gen übersichtlich abbildet. Weitere Features sind „Suspekte Links“ und „ReqIFz Transfer“. Ersteres verbessert die Verlinkungsmöglichkeiten von ProR, indem dadurch automatisch Links durch ein Flag hervorgehoben werden, sobald sich Änderungen in Quelle oder Ziel eines Linkes ergeben. So kann der Autor überprüfen, ob Links nach Änderungen weiterhin korrekt sind. Das zweite Feature, ReqIFz Transfer, erlaubt das Einbetten von Objekten wie beispielsweise Bildern oder Dateien, die dann als separate Dateien gespeichert werden. Zuletzt bietet das Essentials Paket noch eine Team-Unterstützung, die die interne Kollaboration mit ProR verbessern soll. Es stellt eine Ergänzung zu dem grafischen Vergleich ReqIF Diff dar, da somit sichergestellt wird, dass bei parallelem Bearbeiten Konflikte gelöst werden können⁷⁸.

SWOT-Analyse zu ProR

Nach der Vorstellung der Software an sich und der zugehörigen Fähigkeiten sollen im Folgenden die entscheidenden Aspekte in einer Strengths-Weaknesses-Opportunities-Threats (SWOT) Matrix zusammengefasst werden, um so den Vergleich mit dem kommerziellen Produkt zu erleichtern.

<p>Strengths</p> <ul style="list-style-type: none"> - Unterstützt alle gängigen Betriebssysteme - Einfache Installation - Übersichtliche Benutzeroberfläche - Standard Anforderungsformat ReqIF als Basis - Kürzlich veröffentlichtes Update - Verfügbarkeit von Trainings - Einfaches Bug Reporting - Erste Ansätze der internen Kollaboration - Ausgereifte Logik für die Verlinkung von Anforderungen 	<p>Weaknesses</p> <ul style="list-style-type: none"> - Web Hilfe lediglich in Form eines Handbuch - Bisher keine Unterstützung der Benutzerverwaltung durch einen Administrator - Gängige Import- / Exportformate wie Word und Excel werden nicht unterstützt
<p>Opportunities</p> <ul style="list-style-type: none"> - Ständige Erweiterung - Nachvollziehbarkeit der Entwicklungsfortschritte in Form eines Blogs - Einziges Open-Source Programm, das den ReqIF Standard implementiert - Industrielle Bedeutung und Unterstützung - Entwicklung eines Prototyps für das Reporting 	<p>Threats</p> <ul style="list-style-type: none"> - Mangelnde finanzielle Mittel - Kleine Gruppe von Entwicklern - Geringe Aktivität der ProR User Community - Support teilweise kostenpflichtig

Tabelle 16: SWOT-Analyse des Open Source Produktes ProR

⁷⁸ Vgl. Formal Mind (2014 b)

Stärken und Chancen

In der Matrix zeigt sich, dass die Stärken und Chancen deutlich gegenüber den Schwächen und Risiken überwiegen. Zunächst ist positiv hervor zu heben, dass die Software plattformübergreifend eingesetzt werden kann und die gängigen Betriebssysteme Linux, Mac und Windows unterstützt. Die Installation ist einfach und gut beschrieben. Gleichermaßen übersichtlich ist die tabellarische aufgebaute Benutzeroberfläche, die gerade für Nutzer von Eclipse leicht zu verstehen ist. Wie bereits vorher angemerkt, basiert ProR auf dem Austauschformat ReqIF, das als Standard für das Anforderungsmanagement gilt. Die geringe Zeitspanne seit dem letzten Update verdeutlicht die Aktualität der Entwicklung. Außerdem sind Videos verfügbar, die eine erste Einführung in das Programm geben. Darüber hinaus gehende Trainings sind zudem auf Anfrage erhältlich. Eine weitere Stärke liegt in dem einfachen Bug Tracking, da Nutzer eventuelle Lücken des Systems in der formalmind Studio Community aufzeigen können⁷⁹. Durch die Erweiterung ProR Essentials gibt es erste Ansätze für die interne Kollaboration sowie eine fortschrittlich ausgereifte Logik für die Nachvollziehbarkeit der Anforderungen. Diese Stärken der Software an sich werden durch verschiedene Chancen unterstützt: Zunächst zeigen die Updates, dass an einer ständigen Erweiterung des Programms gearbeitet wird. Die Software ist verhältnismäßig jung in seinem Entwicklungsstatus und hat daher enormes Potenzial, weitere Fähigkeiten zu implementieren. Es kommt hinzu, dass die aktuellen Arbeiten und Fortschritte für alle Nutzer des Programms in einem Blog nachvollziehbar kommuniziert werden. Hier werden neue Fähigkeiten und deren Mehrwert außerdem eingehend erklärt⁸⁰. ProR stellt derzeit das einzige Open-Source Programm dar, das auf dem ReqIF Standard basiert. Dadurch kommt der Lösung eine hohe industrielle Bedeutung zu und wird von verschiedenen Unternehmen wie Airbus, Atos oder MKS unterstützt⁸¹. Das deutliche Verbesserungspotenzial wird derzeit bereits aktiv vorangetrieben. So wurde beispielsweise im Rahmen einer Masterarbeit die Möglichkeit des Reportings untersucht und ein erster Prototyp für die übersichtliche Ausgabe des Anforderungsstatus entwickelt⁸².

Schwächen und Risiken

Natürlich steht diesen Stärken und Chancen auch eine Reihe an Schwächen gegenüber. So können einige der Basis Funktionalitäten für das Anforderungsmanagement bisher nicht abgedeckt werden. Zum einen gibt es keine Unterstützung der Benutzer- und Rechteverwaltung durch einen Administrator. Dieser Aspekt schränkt die internen Kollaborationsmöglichkeiten deutlich ein. Zudem ist bisher keine Anbindung von Microsoft Word oder Excel für den

⁷⁹ Vgl. Formal Mind (o. J.)

⁸⁰ Vgl. Formal Mind (2014 c)

⁸¹ Vgl. Eclipse Foundation (2014 a)

⁸² Vgl. Salem, S. (2013)

Import oder Export von Daten gegeben. Außerdem beschränkt sich die Web Hilfe derzeit auf ein Handbuch beziehungsweise Tutorial, das tendenziell weniger für Nutzer ohne Programmierkenntnisse geeignet ist. Mit Open Source Software sind darüber hinaus stets deutliche Risiken verbunden, die in der Analyse beachtet werden müssen. Der offensichtlichste und schwerwiegendste Aspekt sind dabei zunächst die fehlenden finanziellen Mittel. Ohne die Einnahmen von dem Verkauf der Software sind Erweiterungen nur schwer finanzierbar, da die Entwicklung zeit- und kostenintensiv ist. Des Weiteren beschränkt sich die aktive Entwickler Community derzeit auf drei Entwickler und auch die Nutzer Community weist eine eher geringe Aktivität auf. Das letztgenannte Risiko, der kostenpflichtige Support, muss unter Abwägung betrachtet werden. Die Formal Mind GmbH bietet seinen Kunden die Möglichkeit, die Open Source Software den individuellen Bedürfnissen anzupassen und spezielle Trainings durchzuführen⁸³. Auf diese Art und Weise macht das Unternehmen Einnahmen, die es in die Weiterentwicklung des Programmes investieren kann. Gleichzeitig verliert die Lösung dadurch natürlich seinen entscheidenden Vorteil der Kosteneinsparung, da bestimmte Services kostenpflichtig angeboten werden. Dennoch ist festzuhalten, dass die Lösung weiterhin frei verfügbar heruntergeladen werden kann und auch das Handbuch zugänglich ist – lediglich weiterführende Anpassungen müssen entgeltlich erworben werden. Dadurch lässt sich schließen, dass dieses Geschäftsmodell nicht unbedingt ein Risiko darstellt, da der Open Source Gedanke weiterhin vertreten wird.

Insgesamt lässt sich sagen, dass die Stärken der Software die Schwächen und Risiken deutlich überdecken. Die Lösung weist bereits einen ausgereiften Entwicklungsstand auf und lässt weitere Verbesserungen in der kommenden Zeit erwarten. Es ist jedoch einschränkend anzumerken, dass der Status insbesondere in Bezug auf die interne Kollaboration bisher nicht voll ausgereift ist. Daher ist das Produkt zum jetzigen Zeitpunkt vor allem für **kleine Unternehmen** interessant, die keiner umfassenden Kollaborationsmöglichkeiten bedürfen.

Im Folgenden sollen die gewonnenen Ergebnisse zurate gezogen werden, um ProR mit der kommerziellen Software CASE Spec zu vergleichen.

⁸³ Vgl. Formal Mind (2014 d)

5.3.3 Vergleich des kommerziellen Produktes und der Open Source Lösung

Anhand der vorangegangenen Analysen der Produkte CASE Spec und ProR soll im Folgenden ein Vergleich der kommerziellen Software und der Open Source Lösung erfolgen.

Betrachtung der Unterschiede innerhalb der Kategorien

Wie aufgezeigt, weist CASE Spec schon in der Kategorie „Allgemeines“ deutliche Vorteile gegenüber ProR auf, da für die kommerzielle Lösung ein besserer Support zur Verfügung steht. Dies ist angesichts der Finanzierungsmöglichkeiten des Open Source Produktes nicht verwunderlich, wobei auch bereits darauf hingewiesen wurde, dass für ProR kostenpflichtige Support-Wege bestehen. In Bezug auf die Fähigkeiten des Anforderungsmanagements ähneln sich die beiden Produkte weitestgehend, insbesondere die Basic Features werden von beiden Software Lösungen gleichermaßen abgedeckt. Die Lücken von ProR gegenüber CASE Spec liegen im Bereich des parallelen Verarbeitens mehrerer Projekte sowie der Möglichkeit, agiles Projektmanagement abzubilden. Diese beiden Fähigkeiten erscheinen jedoch für den Einsatz in kleinen Unternehmen weniger ausschlaggebend.

In Kapitel 5.3.2 wurden die Ansätze für Kollaborationsmöglichkeiten bei ProR bereits vorgestellt. Während ProR in diesem Bereich derzeit noch wenig fortschrittlich ist, schneidet die kommerzielle Lösung hier deutlich besser ab, da Anforderungen kommentiert und per E-Mail kommuniziert werden können. Allerdings wurde auch bereits diskutiert, dass gerade kleine Unternehmen nicht zwingend umfassende Kollaborationsmöglichkeiten benötigen, sodass ProR im Vergleich weiterhin mit der kommerziellen Lösung konkurrieren kann.

In Hinblick auf den Aufbau und die Navigation erfüllt ProR allerdings auffallend wenige Features. Die nachfolgende Abbildung verdeutlicht, dass der Aufbau stark von Eclipse geprägt ist und gerade für ungeübte Nutzer eine Herausforderung darstellen könnte.

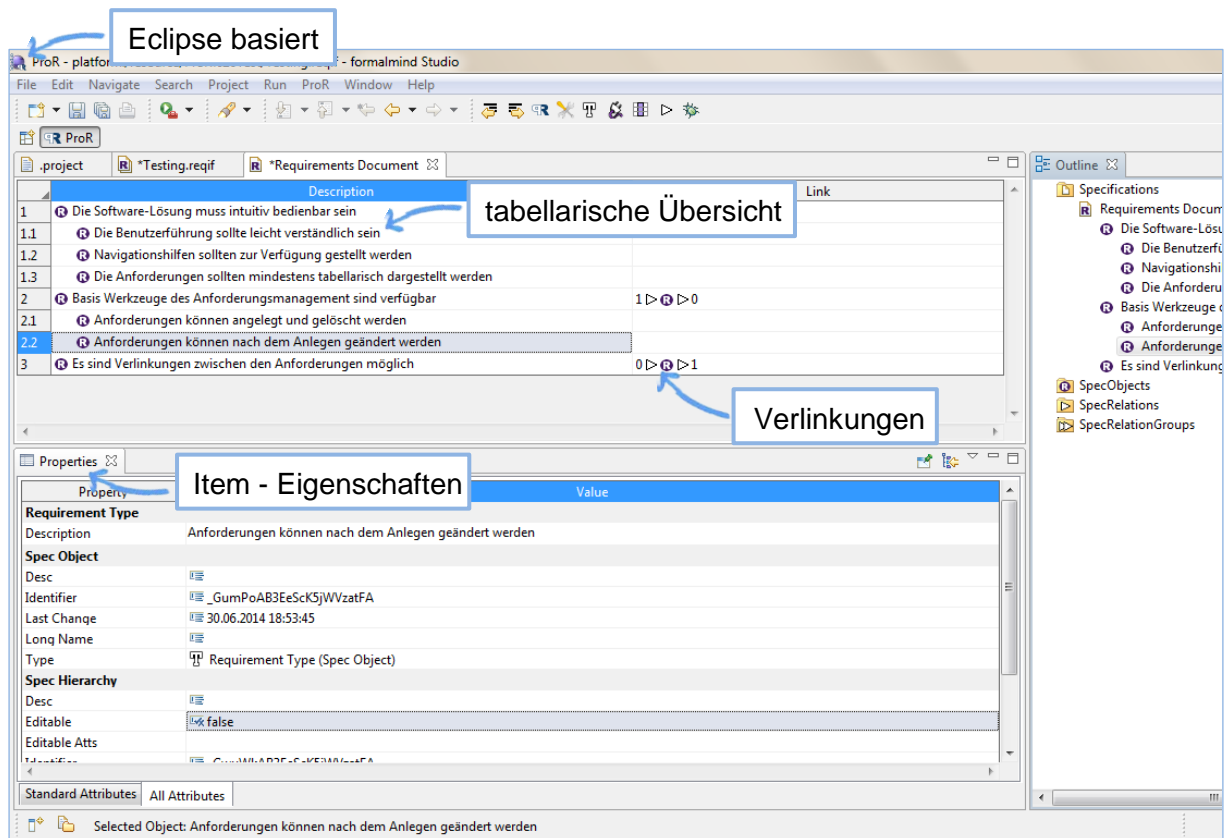


Abb. 5: Benutzeroberfläche von ProR

Auch in Hinblick auf die Administration ist ProR weniger ausgereift als die proprietäre Lösung; was das Änderungsmanagement betrifft, sind die Lösungen aber wieder vergleichbar.

Die Zugangs- und Integrationsaspekte können nur bedingt verglichen werden, da an dieser Stelle auf beiden Seiten Angaben fehlen. Allerdings scheint CASE Spec in Bezug auf den Zugang über den Webbrowser sowie das integrierte Test- und Defect Management der Open Source Lösung überlegen. Dabei ist aber erneut zu bedenken, dass gerade für kleine Firmen ein Web Zugang nicht zwingend erforderlich ist, solange die Software auf den betreffenden PCs zur Verfügung steht.

Für die letzten beiden Kategorien ergibt sich ein ähnliches Bild: CASE Spec hat umfassende Import / Export Formate, während ProR lediglich HTML unterstützt. Betrachtet man die unterstützten Anforderungs-Formate muss jedoch hervorgehoben werden, dass ProR wie erläutert auf dem Austauschformat RIF beziehungsweise ReqIF beruht, sodass ein standardisierter Austausch ermöglicht wird. Analysewerkzeuge werden lediglich von CASE Spec geboten, jedoch wurde ja bereits darauf hingewiesen, dass ProR in diesem Bereich weitere Entwicklungen plant.

Zusammenfassende Abwägung der Unterschiede

Die aufgezeigten Unterschiede sind auch in den Prozentzahlen erkennbar. CASE Spec deckt eindeutig einen größeren Bereich ab als ProR. Unter Kapitel 5.3.1 wurde ja bereits aufgezeigt, dass die kommerzielle Lösung von den Fähigkeiten her auch für mittelständige Unternehmen einsetzbar wäre, gerade da es umfangreiche Analyse- und Kollaborationsfeatures stellt. ProR erreicht die 50 % Hürde der Basisfunktionalitäten nur knapp und muss erst deutliche Entwicklungsschritte machen, bevor ein Einsatz in größeren Unternehmen denkbar wird. Das Hauptargument für CASE Spec für den Gebrauch in kleinen Unternehmen war der Preis von 1.000\$ als on-premise Lizenz, beziehungsweise 129\$ für drei off-premise Lizenzen. Obwohl im Vergleich zu anderer kommerzieller Software sehr kostengünstig, kann die Open Source Lösung an dieser Stelle natürlich noch stärker überzeugen. Da üblicherweise jede Kosteneinsparung einen Vorteil bringt, muss dieser Aspekt sehr hoch gewertet werden.

In Abwägung der einzelnen Fähigkeiten zeigt sich, dass ProR nicht so viele Basis Features abdeckt wie CASE Spec, aber nichtsdestotrotz alle wichtigen Punkte aufweist, um erfolgreich das Anforderungsmanagement abzudecken. Somit lässt sich sagen, dass ProR zwar von den Fähigkeiten nicht so weit ausgereift ist wie CASE Spec, durch die kostenfreie Nutzung aber dennoch mit der kommerziellen Lösung konkurrieren kann und schon zum jetzigen Entwicklungsstand für kleine Unternehmen interessant ist.

5.3.4 Bewertung des Open Source Potenzials

In der Analyse unter Abschnitt 5.3.2 wurde deutlich, dass ProR bereits viele Fähigkeiten des Anforderungsmanagements aufweist. Wie am Ende des Kapitels dargestellt, ist es zum jetzigen Zeitpunkt aufgrund des Projektstatus eher für kleine Unternehmen geeignet, doch gerade die ständigen Erweiterungen wie in Form von ProR Essentials machen die Lösung interessant.

Industrieller Einsatz von ProR

Tatsächlich findet das Projekt bereits breite Unterstützung in der Industrie, so haben neben der Heinrich-Heine Universität Düsseldorf unter anderem auch Airbus, Atos und MKS ihr Interesse an dem Projekt geäußert⁸⁴. Dies zeigt, dass die Nachfrage nach Open Source Lösungen für das Anforderungsmanagement vorhanden ist und durchaus Potenzial aufweist. Dank dieser Unterstützung stellt ProR einen ernstzunehmenden Kandidaten für das Anforderungsmanagement dar. Auch in dem Vergleich unter Kapitel 5.3.3 hat sich gezeigt, dass ProR von den Fähigkeiten zwar keiner kommerziellen Lösung entspricht, aber dennoch seinen Zweck für das Anforderungsmanagement zufriedenstellend erfüllt. Gerade Firmen, die

⁸⁴ Vgl. Eclipse Foundation (2014 a)

für ihr Anforderungsmanagement derzeit auf Microsoft Word oder Excel zurückgreifen, könnten von ProR profitieren. Dieses bietet durch die Nachvollziehbarkeit der Anforderungen in Form von Verlinkungen deutliche Vorteile gegenüber den Standard-Textverarbeitungs-Werkzeugen. Hinzu kommt, dass die Installation und Benutzeroberfläche sich in einem Test als einfach handhabbar erwiesen haben und damit auch für Mitarbeiter mit geringen bis gar keinen Programmierkenntnissen geeignet sind.

Im Vergleich zu den übrigen identifizierten Open Source Produkten scheint ProR das einzige geeignete Produkt für das Anforderungsmanagement in kleinen Unternehmen zu sein. Selbstverständlich bleibt abzuwägen, ob die Vorteile gegenüber einer kostengünstigen Lösung wie CASE Spec überwiegen, da die individuelle Anpassung von ProR auch Kosten mit sich bringen würde. Nichtsdestotrotz könnten Firmen die Software auch kostenfrei testen und prüfen, inwiefern der Einsatz für das eigene Anforderungsmanagement geeignet ist, ohne Investitionen zu tätigen. Daher ist der ProR Ansatz eine ernstzunehmende Alternative, die sich insbesondere durch den ReqIF Standard auszeichnet. Vorausgesetzt, die Entwicklung wird stetig aufrechterhalten, kann die Lösung zukünftig auch für mittelständige Unternehmen an Bedeutung gewinnen, die auf diesem Standard beruhen.

Alles in allem lässt sich demnach festhalten, dass ProR **deutliches Potenzial** besitzt, als Open Source Software kommerzielle Produkte des Anforderungsmanagements innerhalb kleiner Unternehmen abzulösen. Mit fortschreitenden Entwicklungen in Bezug auf die Kollaborationsmöglichkeiten wird sich die Zielgruppe weiter ausdehnen können, doch zum jetzigen Zeitpunkt liegt das Hauptaugenmerk auf **kleinen Unternehmen**.

6 Fazit

Ziel dieser Arbeit war es, einen unabhängigen und objektiven Überblick über verschiedene Produkte im Bereich Anforderungsmanagement zu geben und ausgewählte Produkte näher zu untersuchen und zu vergleichen. Um das volle Spektrum an aktuell auf dem Markt verfügbaren Anforderungsmanagement-Lösungen abzubilden, sollten zudem Open Source Lösungen in diese Untersuchungen miteinbezogen werden. Dabei bestand die Absicht, neben den aktuellen kommerziellen Marktführern mögliche Substitutionsprodukte aus dem Bereich Open Source zu identifizieren und zu bewerten.

Insgesamt wurden 17 Produkte, davon zehn kommerzielle sowie sieben aus dem Bereich Open Source, in den Kategorien Allgemeines, Anforderungsmanagement, Collaboration, Aufbau und Navigation, Administration, Änderungsmanagement, Zugang und Integration, Datenaustausch, und Analyse mithilfe des eigens erstellten Kriterienkataloges analysiert und anhand eines Gewichtungssystems den entsprechenden Unternehmensgrößen zugeordnet.

Als Marktführer für große Unternehmen wurde IBM Rational DOORS identifiziert. Die Betrachtung der Eignung von Open Source Produkten zeigte, dass für diese Unternehmensgröße derzeit keine nennenswerten Alternativen verfügbar sind. Die Gründe hierfür liegen hauptsächlich in den kleinen beziehungsweise inaktiven Entwickler-Communities sowie in mangelnden Funktionalitäten in den Bereichen Administration und Kollaboration. Aus dem Bereich der kommerziellen Anwendungen konnten das HP Quality Center sowie Caliber von Borland als mögliche Substitutionsprodukte für IBM Rational DOORS ermittelt werden.

Der Bewertung nach ist OneDesk Requirements Management von OneDesk Inc. aktueller Marktführer für mittlere Unternehmen. Ebenso wie im Segment großer Unternehmen zeigten sich bei der Betrachtung des Open Source Potenzials keine nennenswerten Alternativen. Entscheidend war auch hier der mangelnde Funktionsumfang. Stattdessen wurden die möglichen kommerziellen Alternativen Accompa und Polarion RM näher betrachtet.

Für kleine Unternehmen wurde CASE Spec von Goda als Marktführer ausgewiesen. An dieser Stelle konnte zudem eine mögliche Open Source Alternative in Form von ProR von Formal Minds identifiziert werden. Obwohl ProR im Verhältnis zu CASE Spec einen deutlich geringeren Funktionsumfang aufweist, kann es aufgrund der geringeren Kosten sowie zu erwartender zukünftiger Entwicklungen einem Vergleich durchaus standhalten.

Insgesamt ist das Open Source Potenzial zum jetzigen Zeitpunkt allerdings beschränkt: Außer im Bereich der kleinen Unternehmen gibt es momentan keine ernstzunehmenden Open Source Alternativen zu der proprietären Software. Mit fortschreitender Entwicklungsarbeit und einer wachsenden Community könnte sich dieses Bild allerdings in den nächsten Jahren

verändern und Open Source Produkte damit auch für mittlere bis große Unternehmen interessant werden lassen. Es bleibt abzuwarten, inwiefern sich die Open Source Anwendungen in Bezug auf ihre Eignung für den industriellen Einsatz in den kommenden Jahren weiterentwickeln.

Die Analyse kommerzieller Software ergibt ein anderes Bild: Schon bei der Betrachtung von nur zehn Produkten konnten für jede Unternehmensgröße neben den aktuellen Marktführern weitere kommerzielle Alternativen mit individuellen Stärken und Schwächen gegenüber den Marktführern identifiziert werden.

Zusammenfassend kann daher festgehalten werden, dass im Markt für Anforderungsmanagement-Software eine breite Abdeckung durch kommerzielle Anwendungen vorherrscht, während die Open Source Produkte zum jetzigen Zeitpunkt ein geringes Potenzial aufweisen.

Anhang

Anhangverzeichnis

Anhang 1: Erstellter Kriterienkatalog	82
---	----

Anhang 1: Erstellter Kriterienkatalog

Kriterienkatalog für Anforderungsmanagement Software

Gewichtung	Kriterien	Proprietär									Open Source						
		ReMa (AccordSoft)	OneDesk (OneDesk Inc)	Accompia	Quality Center (HP)	Rational DOORS (IBM)	LiteRM (ClearSpec)	CASE Spec (Goda)	Integrity (PTC)	Caliber (Borland)	RM (Polarion)	OSRMT (ideaStub)	a Nimble (ideaStub)	mttoo (fbonate)	ProR (Formal Minds)	SHORE (S&M AG)	ReqT (Lund University)
Basic	Algemeines																
Advanced	Version	2.0	2.0	2.0	2.0	5.0	10.0	2010	2010	Caliber 11.3	1.5	0.4	23	0.10	2.0.1	3.0X	
Optional	Preis	k.A.	Projekte - \$250/Monat	\$399/Monat inkl. 5 Saas	>59000 per seat Saas	124€ pro Jahr und User Saas	559,95	nur OR-Off-premises License OR Saas	Unix, Windows	Ult-User / 2.380 €	\$1.900	entfällt	entfällt	entfällt	entfällt	entfällt	
	Plattform	Windows	SaaS	SaaS	SaaS	SaaS	Windows	SaaS	Windows	Windows/Linux/SaaS	Windows	Windows	Windows	Linux, Mac, Windows	Java	Windows (Kala)	
	Open-Source Spezifika																
	Entwickler-Community										3 developers	1 company	1 company	3 developers	2 project admins	1 contributor	
	Letztes Update										28.03.2007	23.12.2012	26.03.2012	22.04.2014	28.03.2009	19.05.2014	
	Bugtracker										+	-	+	+	+	+	
	Support																
✓	Web Hilfe	-	+	+	+	+	0	+	+	0 (Handbuch)	+	0	0	0 (Handbuch)	0	0	
✓	Telefonischer Support	0	+	+	+	+	0	+	+	+	-	-	-	-	-	-	
✓	Verfügbarkeit von Trainings (Web-Trainings)	-	+	+	+	+	-	+	+	+	-	-	-	-	-	-	
✓	Verfügbarkeit von Trainings (on-site)	k.A.	k.A.	k.A.	+	+	-	-	+	k.A.	+	-	-	+	-	-	
✓	Stabilität	k.A.	+	k.A.	k.A.	+	k.A.	+	+	-	+	-	-	0 (verbessert)	-	+	
✓	Testversion verfügbar	+	+	+	+	+	+	+	+	0 (Demo)	+	entfällt	entfällt	entfällt	entfällt	entfällt	
✓	Verfügbarkeit als eigenständiges Produkt	+	-	-	-	+	+	+	+	-	+	+	+	+	-	+	
	Anforderungsmanagement																
✓	Anlegen, Ändern und Löschen von Anforderungen	+	+	0	+	+	-	+	-	+	+	-	k.A.	k.A.	+	+	
✓	Definierbare Richtlinien für Anforderungs-Attribute	+	+	+	+	+	0	+	+	+	+	+	+	+	+	+	
✓	Bewertung und Klassifikation von Anforderungen	k.A.	+	0	+	+	0	+	+	+	+	k.A.	-	k.A.	+	+	
✓	Wiederverwendbarkeit angelegter Anforderungsprofile (Templates)	k.A.	+	+	+	+	+	+	k.A.	+	+	+	-	-	-	-	
✓	Anhang unterstützen der Dokumente	k.A.	+	+	+	+	+	+	+	+	+	+	+	-	-	-	
	Anforderungshistorie																
✓	Integriertes System zur Versionskontrolle	+	+	+	+	+	0	+	+	+	+	+	+	+	+	+	
✓	Vollständige Abbildung der Versionshistorie	+	+	+	+	+	-	+	+	+	+	+	-	+	+	+	
✓	Gleichzeitige Verwaltung von Anforderungen verschiedener Projekte	+	+	+	+	+	+	+	+	0	+	0	k.A.	k.A.	-	+	
✓	Agiles Projektmanagement ist abbildbar	k.A.	0	k.A.	0	k.A.	+	+	-	+	+	+	+	-	-	k.A.	
✓	Aufnahme des Status Quo ("Baselineing")	k.A.	-	+	+	k.A.	-	+	+	+	+	-	-	+	-	0	
	Nachverfolgbarkeit von Anforderungen																
✓	Projektspezifische Verlinkungen	+	+	+	+	+	+	+	+	+	+	+	-	+	+	+	
✓	Projekübergreifende Verlinkungen	k.A.	+	+	+	+	+	+	+	+	+	+	-	+	+	+	
✓	Kennzeichnung inkonsistenter Anforderungen (Orphans)	+	0	+	+	+	0	0	+	0	0	-	-	0	+	0	
✓	Verlinkung von Programmpunkten mit korrespondierenden Anforderungen	+	-	-	+	+	0	+	+	+	+	-	-	+	-	-	
	Integrierte Funktionen für das Sammeln von Anforderungen und Daten																
✓	Sammeln von Anforderungen via Email	k.A.	+	+	+	+	-	-	-	k.A.	0	-	-	-	-	-	
✓	Sammeln von Anforderungen via Web	k.A.	+	+	+	+	-	-	-	k.A.	-	-	-	-	-	-	
✓	Einbindung eines Glassars	k.A.	-	k.A.	k.A.	+	-	k.A.	0	+	-	-	-	+	-	-	
	Änderungsmanagement																
✓	Integriertes Änderungsmanagement	+	+	-	+	+	-	+	+	k.A.	+	0	k.A.	k.A.	+	-	
✓	Erstellen, Bewerten und Freigeben von Änderungsanfragen	+	+	-	+	+	-	+	+	k.A.	+	0	k.A.	k.A.	0	-	
	Kollaboration																
✓	Kommentar Funktion	+	+	+	+	+	-	+	+	+	+	-	+	-	-	+	
✓	Integrierte Community	k.A.	+	+	+	0	-	+	+	+	-	-	-	0	-	+	
✓	Tellen von Anforderungen	k.A.	0	+	+	+	+	0	0	0 (URL generieren)	-	-	-	-	-	0	
✓	(Teil-) Automatische Kommunikation mit Kunden und Stakeholdern	k.A.	+	+	0	+	-	0	+	0	-	-	-	-	-	-	
✓	Benutzerdefinierte Email Benachrichtigungen	k.A.	+	+	+	+	-	+	+	+	+	-	-	-	-	-	
	Aufbau und Navigation																
✓	Suchfunktion	+	+	+	+	+	+	+	+	+	+	k.A.	+	0	-	+	
✓	Filterfunktion	+	+	+	+	+	0	+	k.A.	+	+	+	k.A.	0	+	+	
	Ansichten																
✓	Tabellen Ansicht	+	+	+	-	+	-	+	+	+	+	+	+	-	+	0	
✓	Dokumenten Ansicht	+	-	+	+	k.A.	+	+	+	k.A.	0	-	-	-	+	0	
✓	Baum Ansicht	+	-	0	+	+	+	+	+	+	+	+	+	-	0	+	
✓	Roadmap Ansicht	-	+	-	+	+	-	+	-	0 (Diagramm)	+	-	-	-	-	-	
✓	Visualisierung mittels Use Cases	k.A.	-	+	+	k.A.	-	k.A.	k.A.	+	0	+	k.A.	-	0	+	
	Administration																
✓	Administrator Ansicht mit Informationen zu Systemstatus, Projekten und Nutzern	k.A.	+	0	+	+	-	-	+	+	+	0	0	-	-	0	
	Benutzerverwaltung																
✓	Integrierte Benutzer- und Rechteverwaltung	+	+	+	+	+	-	+	+	+	+	0	0	-	-	-	
✓	Individuelle Anpassung von Nutzeransichten	+	+	+	+	+	-	0	+	+	+	-	-	-	0	-	
✓	Rollenbasierte Rechteverwaltung	+	+	+	+	+	-	+	+	+	+	0	0	-	-	0	

Gewichtung
 Basic Advanced Optional

Kriterien
 Voll erfüllt (+), zum Teil erfüllt (0), nicht erfüllt (-), keine Angabe (i.A.)

Proprietär										Open Source						
ReMa (Acord Soft)	OneDesk (OneDesk Inc.)	Accorpa	Quality Center (HP)	Rational DOORS (IBM)	LiteRM (ClearSpec)	CASE Spec (Goda)	Integrity (PTC)	Caliber (Borland)	RM (Polarion)	OSRMT (ideaStub)	a Nimble (ideaStub)	rmoo (flinate)	ProR (Formal Minds)	SHORE (S&M AG)	ReqT (Lund University)	Unicase (TU München)

Quellenverzeichnis

Literaturverzeichnis

- Beatty, J. et al (o. J.): Seilevel's Evaluation of Requirements Management Tools: Summaries and Scores, (Hrsg.: Seilevel), o. V. : Austin
- Bittner, M. et al (2004): Requirements for Requirements Management Tools, in: Proceedings of the 12th IEEE International Requirements Engineering Conference (RE'04)
- Engesser, H. et al (1993): Duden Informatik. Ein Sachlexikon für Studium und Praxis. 2., vollständig überarbeitete und erw. Aufl., Dudenverlag: Mannheim
- Grande, M. (2011): 100 Minuten Anforderungsmanagement, 1. Auflage, Vieweg + Teubner Verlag: Wiesbaden
- IDC (2013): Worldwide Requirements Visualization, Definition, and Management 2013–2017 Forecast and 2012 Vendor Shares, IDC: w. p.
- Jastram, M. (2010): ProR, an Open Source Platform for Requirements Engineering based on RIF, o.V.: Düsseldorf
- Jastram, M. (2012): The ProR Approach: Traceability of Requirements and System Descriptions, Inaugural-Dissertation, o. V. : Düsseldorf
- Niebisch, T. (2013): Anforderungsmanagement in sieben Tagen, Springer Verlag: Heidelberg
- Rupp, C. (2008): Systemanalyse kompakt, 2. Auflage, Springer Verlag: Heidelberg
- Rupp, C. (2009): Requirements- Engineering und –management, 5. Auflage, Carl Hanser Verlag: München
- Salem, S. (2013): Reporting für ReqIF: Von der Analyse bis zur Auswertung, Masterarbeit, o. V.: Düsseldorf

- Schienmann, B. (2002): Kontinuierliches Anforderungsmanagement, Prozesse – Techniken – Werkzeuge, Addison-Wesley: München
- Sommerville, I. (2012): Software Engineering, 9. Auflage, Pearson: München

Sonstige Literatur

- Brügge, B. et al (2009): Unibase – an Ecosystem for Unified Software Engineering Research Tools, München, S. 1

Verzeichnis der Internet- und Intranet-Quellen

- Accompa (o.J.): Enterprise Quality. Affordable
ing, <http://www.accompa.com/purchase.html>, Abruf: 30.06.2014
- Alexander, I. (2014): Requirements Tools, Free, Shareware, Free Trial & Commercial, <http://www.scenarioplus.org.uk/vendors.htm#Accept>, Abruf: 19.05.2014
- Atlantic Systems Guild Ltd. (2014): Requirements Tools, <http://www.volere.co.uk/tools.htm>, Abruf: 19.05.2014
- Birk, Andreas (2013): What Software Tools Support Requirements Management?, <http://makingofsoftware.com/2013/what-sw-tools-support-rm>, Abruf: 02.07.2014
- Deploy-Project (o. J.): Deploy is an European Commission Information and Communication Technologies FP7 project, <http://www.deploy-project.eu/>, Abruf: 20.06.2014
- Eclipse Foundation (2014 a): Requirements Modeling work, <http://www.eclipse.org/proposals/modeling.mdt.rmf/>, Abruf: 20.06.2014
- Eclipse Foundation (2014 b): ProR, Requirements Engineering Platform – An editor powered by the Eclipse RMF Project, <http://www.eclipse.org/rmf/pror/>, Abruf: 20.06.2014
- EU-Kommission (2003): Empfehlung der Kommission vom 6. Mai 2003 betreffend die Definition der Kleinstunternehmen sowie der kleinen und mittleren Unternehmen, <http://eur-lex.europa.eu/legal->

- [con-](#)
[tent/DE/TXT/PDF/?uri=uriserv:OJ.L_.2003.124.01.0036.01.DE](#)
[U](#), Abruf: 25.05.2014
- Formal Mind (o. J.): Formal Mind Studio, Submit a
quest, https://formalmind.zendesk.com/anonymous_requests/new,
[w](#), Abruf: 21.06.2014
- Formal Mind (2014 a): Formal Mind Studio – Leichtes Anforderungsmanagement und
ReqIF-Editor, <http://www.formalmind.com/de/studio>, Abruf:
20.06.2014
- Formal Mind (2014 b): ProR Essentials, <http://www.formalmind.com/de/pror-essentials>,
Abruf: 20.06.2014
- Formal Mind (2014 c): Newsfeed/Blog, <http://www.formalmind.com/de/blog>, Abruf:
21.06.2014
- Formal Mind (2014 d): Services, <http://www.formalmind.com/de/services>, Abruf:
21.06.2014
- Goda Software Company (o.J.): Pricing/Order details, <http://www.analysttool.com/Pricing/>,
Abruf: 30.06.2014
- Heller, G. (2013): RM Tools – what are they anyway?
, <http://makingofsoftware.com/2013/rm-tools-what-are-they-anyway>, Abruf: 02.07.2014
- Hewlett-Packard GmbH (o.J.): Quality Center Enterprise,
<http://www8.hp.com/de/de/software-solutions/quality-center-quality-management/>, Abruf: 30.06.2014
- HIS (2009): Requirements Interchange Format, <http://www.automotive-his.de/rif/doku.php?id=welcome>, Abruf: 20.06.2014
- Hofmann, B. (2012): Wie funktioniert OAuth2, Der Standard für Clientautorisierung im Überblick,
<http://phpmagazin.de/So-funktioniert-OAuth2-5081.html>,
Abruf: 07.07.2014
- Infot-Tech Research Group (2012): Vendor Landscape: Agile ALM, ALMs for the poor, and
for the rich, <http://www.slideshare.net/venj123/agile-alm-tool-comparison>, Abruf: 21.06.2014

- IBM Knowledge Center (o.J.a): Rational DOORS Next Generation, http://www-01.ibm.com/support/knowledgecenter/SSJJ9R_5.0.0/com.ibm.rational.rmm.help.doc/topics/c_rm_intro.html, Abruf: 04.07.2014
- IBM Knowledge Center (o.J.b): Rational-Lösung für Collaborative Lifecycle Management, http://www-01.ibm.com/support/knowledgecenter/SSJJ9R_5.0.0/com.ibm.help.common.jazz.calm.doc/topics/c_calm_common.html, Abruf: 04.07.2014
- Jastram, M. (o. J.): Anforderungen integrieren mit dem Eclipse Requirement Modeling Framework, http://fg-re.gi.de/fileadmin/gliederungen/fg-re/Treffen_2012/jastram.pdf, Abruf: 20.06.2014
- Jastram, M. (2012): Comparing ReqIF Files with ProR Essentials Diff, <http://www.youtube.com/watch?v=-XSc4sqi0gg&feature=youtu.be>, Abruf: 20.06.2014
- Mühlbauer, S. / Versteegen, G. / Keller, F. (2006): Werkzeuge für das Anforderungsmanagement, Komplexe Abläufe, <http://www.elektronikpraxis.vogel.de/themen/embeddedsoftwareengineering/planung/articles/50996/>, Abruf: 19.05.2014
- OneDesk, Inc. (2014): Pricing, <http://www.onedesk.com/pricing/>, Abruf 02.07.2014
- Polarion (o.J.a): How to buy, <https://www.polarion.com/products/requirements/licensing.php>, Abruf: 03.07.2014
- Polarion (o.J.b): Extensions, <https://www.polarion.com/products/requirements/extensions>, Abruf: 03.07.2014
- PTC Inc. (2014): PTC Integrity – Accelerating Innovation in Software Intensive Products, <http://www.mks.com/platform/our-product>, Abruf: 21.06.2014
- Regnell, B. (2013): Flexible Requirements Modeling with reqT, <http://reqt.org/reqT-intro.pdf>, Abruf 02.07.2014

- sd&m AG (2001): SHORE 1.1.0 Administrator-Handbuch, <http://www.openshore.org/Download/Docs/shore-administratorhandbuch.pdf>, Abruf: 21.06.2014
- Shrivathsan, M. (2012): What is MRD? Market Requirements document, <http://pmblog.accompa.com/2012/05/24/what-is-mrd-market-requirements-document/>, Abruf: 07.07.2014
- SourceForge (2014): OpenSHORE, <http://sourceforge.net/projects/openshore/files/>, Abruf: 21.06.2014
- Stelzer, J. u.a. (2014): RMF/User Guide /Tutorial, https://wiki.eclipse.org/RMF/User_Guide/Tutorial, Abruf: 20.06.2014
- o.A. (2014): Anforderungsanalyse/Requirements ing, <http://www.infforum.de/themen/anforderungsanalyse/re-methoden.htm>, Abruf: 22.05.2014
- OASIS (2014): OASIS Security Services (SAML) TC, https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security, Abruf: 07.07.2014

Vergleich von Open Source ETL-Tools und Erstellung eines Prototyps

Schriftliche Ausarbeitung
im Rahmen der Lehrveranstaltung „Integrationsseminar“

Vorgelegt von

Ines Schlenker
Janine Jakob
Lara Hochhalter
Muazzez Algac
Sarah Ahmed

am 11.07.2014

Fakultät Wirtschaft
Studiengang Wirtschaftsinformatik
Kurs WWI2011I

Inhaltsverzeichnis

Abkürzungsverzeichnis	IV
Abbildungsverzeichnis.....	IV
Tabellenverzeichnis.....	V
1 Einleitung	1
1.1. Problemstellung	1
1.2. Zielsetzung.....	3
1.3. Vorgehensweise.....	4
2 Extraktion-Transformation-Laden (ETL)	5
2.1 ETL-Prozess	5
2.1.1 Extraktion.....	6
2.1.2 Transformation.....	7
2.1.3 Laden.....	8
2.2 ELT-Prozess	8
2.3 Open Source.....	10
2.4 Open Source Lizenzen.....	10
3 Methodik	12
3.1 Kriterienkatalog	12
3.2 Prototyp	15
4 Beschreibung der ETL Anwendungen	18
4.1 Kettle.....	18
4.2 Adeptia.....	21
4.3 Jitterbit	27
4.4 Xineo XIL	29
4.5 CloverETL.....	31
5 Kriterienkatalog und Gewichtung.....	36
5.1 Grobkriterien	36
5.2 Feinkriterienkatalog.....	38
5.2.1 Beschreibung Produktprofil	39
5.2.2 Beschreibung Anbieterprofil	42
5.3 Auswertung der ETL Anwendungen.....	45
5.3.1 Kettle – Auswertung des Feinkriterienkatalogs.....	45
5.3.2 CloverETL- Auswertung des Feinkriterienkatalogs.....	48
6 Prototypische Umsetzung mit Kettle.....	51
6.1 Konzept zur Umsetzung des Prototyps	51
6.2 Erstellung einer großen Datenbasis	52
6.3 Datenbank einrichten und Verbindungsaufbau zu Kettle	53
6.4 ETL-Prozess mit Kettle.....	55
6.4.1 Input: Datenintegration.....	56
6.4.2 Output-Generierung	57
6.4.3 Transform-Prozess.....	57
6.4.4 Weitere Funktionen	59
7 Zusammenfassung und Ausblick.....	60

7.1	Zusammenfassung und Bewertung	60
7.2	Ausblick	61
8	Anhang.....	62
	Anhangverzeichnis	62
	Quellenverzeichnisse	72

Abkürzungsverzeichnis

B2B	Business-to-Business
BI	Business Intelligence
BPM	Business Process Management
CE	Community Edition
CRM	Customer Relationship Management
CSV	Comma Separated Value
DOM	Document Object Model
EAI	Enterprise Application Integration
EE	Enterprise Edition
ELT	Extract Transform Load bzw. Extraktion Laden Transformation
ER	Entity Relationship
ERP	Enterprise-Resource-Planning
ETL	Extraktion, Transformation, Laden
JDBC	Java Database Connectivity
JNDI	Java Naming and Directory Interface
ODBC	Open Database Connectivit
OSBI	Open Source Business Intelligence
PDI	Pentaho Data Integration
SaaS	Software as a Service
SOA	Serviceorientierte Architektur
SQL	Structured Query Language
XIL	XML Import Language

Abbildungsverzeichnis

Abb. 1: Gartner Hype Cycle Special Report 2011.....	1
Abb. 2: Allgemeiner ETL-Prozess.....	5
Abb. 3: ELT-Prozess	9
Abb. 4: Auswahlsschritte.....	12
Abb. 5: DIN ISO 9126.....	14
Abb. 6: Kettle Logo.....	18
Abb. 7: Adeptia Logo.....	21
Abb. 8: Die drei Komponenten der Adeptia ETL	23
Abb. 9: Schema Parser	25
Abb. 10: Data Mapper	25
Abb. 11: Jitterbit Logo	27
Abb. 12: Jitterbit - 3 Schritte Durchführung.....	28
Abb. 13: ETL-Prozess in Jitterbit	29
Abb. 14: Xineo Logo.....	29
Abb. 15: Definition von Datenquellen	30
Abb. 16: CloverETL Logo	31
Abb. 17: CloverETL Sprachen und Code.....	32
Abb. 18: CloverETL Features	33
Abb. 19: CloverETL Product Suite.....	33
Abb. 20: Überblick CloverETL	34
Abb. 21: Konzept zur Prototypen Umsetzung.....	51
Abb. 22: CSV Beispieldatei	53
Abb. 23: ER Modellierung der unterschiedlichen Datenbanktabellen.....	54
Abb. 24: Verbindungsaufbau zwischen MySQL und Kettle.....	55
Abb. 25: Ablaufprozess ETL mit unterschiedlichen Komponenten.....	56

Tabellenverzeichnis

Tab. 1: Transformationsfunktionalitäten.....	34
Tab. 2: Architektur.....	35
Tab. 3: Archivfunktion.....	35
Tab. 4: Bewertung der Grobkriterien für die fünf ausgewählten ETL-Tools.....	38
Tab. 5: Bewertung der Feinkriterien	38

1 Einleitung

(Sarah Ahmed, Muazzez Algac, Lara Hochhalter, Janine Jakob, Ines Schlenker)

1.1. Problemstellung

Heutzutage werden nahezu überall riesige Datenmengen erzeugt – Social- Media- Platt- formen, Call Centers, Digitalisierung von Patientenakten etc. Diese heterogenen Daten- ströme wachsen exponentiell und es kommen immer mehr Datenquellen hinzu. Das schnelle Anwachsen von Datenmengen wird auch als ‚Big Data‘ bezeichnet.

Der Begriff Big Data wurde zum ersten Mal im Jahre 2011 in dem Gartner Hype Cycle Special Report erwähnt. Der Gartner Hype Cycle Special Report bewertet jährlich die Ein- führung der neuesten Technologien und wie viel Aufmerksamkeit diese in Zukunft be- kommen werden.¹ Dieser Report ist eine große Hilfe, um zu sehen, wie viel Potential eine neue Technologie hat und ob sich die Einführung lohnt. In der unten abgebildeten Kurve in Abbildung 1 kann man sehen, dass der Zyklus in fünf Bereiche eingeteilt wird. Die erste Phase wird als ‚Technology Trigger‘ bezeichnet. Der technologische Auslöser stellt das anfängliche Interesse der Fachwelt dar.² In dieser Phase befindet sich auch Big Data. In der Abbildung ist des Weiteren zu erkennen, dass Big Data durch ‚Extreme Information Processing and Management‘ ergänzt wird.

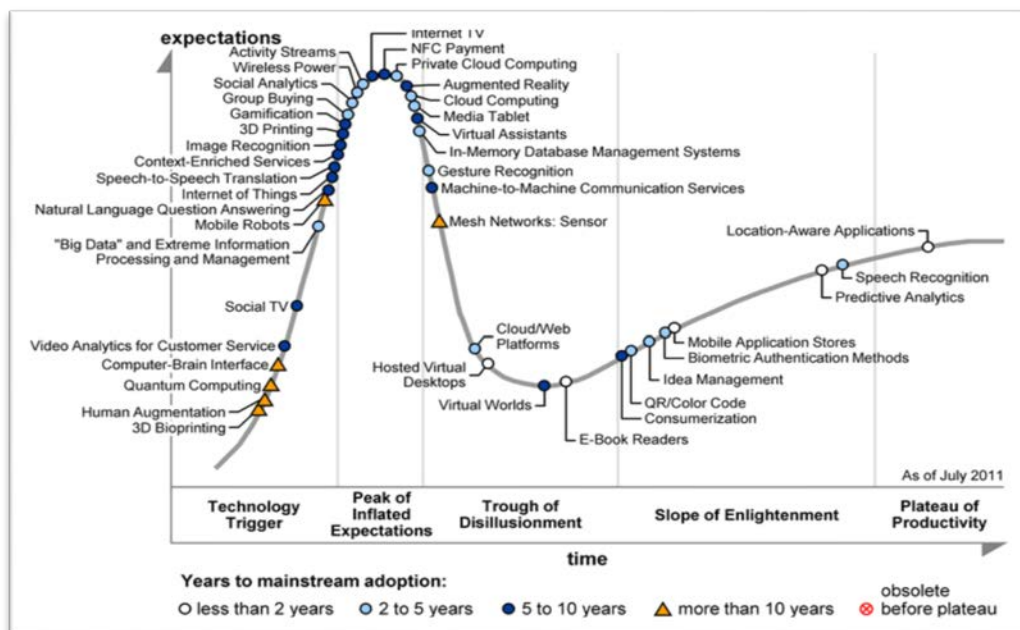


Abb. 1: Gartner Hype Cycle Special Report 2011³

¹ Vgl. Bienert, J. (2011)

² Vgl. Floor (o.J.)

³ Enthalten in: Big Data (o.J.)

Dies zeigt, dass die Schwierigkeit nicht nur in der reinen Datenspeicherung liegt, sondern darin, aus den gespeicherten Daten nutzvolle Informationen zu generieren. Denn Daten sind für die Informationswirtschaft eines Unternehmens unentbehrlich und deswegen eine wichtige Ressource.

In der Betrachtung der Lösungsansätze zu Big Data Fällen werden diese häufig mit folgenden Begriffen dimensioniert: ‚Volume‘, ‚Variety‘ und ‚Velocity‘. Das sind die drei berühmten V. Im Folgenden wird erklärt, worum es bei jedem einzelnen V geht. Volume (dt. Volumen) ist die erste Dimension der Big Data, es beschreibt das weltweit wachsende Datenvolumen.

Die zweite Dimension Variety (dt. Vielfalt) steht für die unzähligen Datenquellen. Heutzutage wird man sowohl von strukturierten Daten als auch von unstrukturierten Daten überflutet.

Die letzte Dimension Velocity (dt. Geschwindigkeit) beschreibt die schnelle Analyse und die sofortige Bereitstellung der riesigen Datenmengen.⁴

Eine weitere wichtige Komponente bezüglich Big Data ist die Notwendigkeit der Datenanalyse im Unternehmen.⁵ Denn das Ziel jedes Unternehmens ist es, die richtigen Informationen zu erlangen, um die notwendigen Verbesserungen im Unternehmen zu erbringen. Das Unternehmen möchte eine schnelle und sofortige Verfügbarkeit der Informationen für Analyse- und Kontrollzwecke, damit es wettbewerbsfähig bleiben kann. Denn ohne eine schnelle Reaktionszeit auf die sich ständig verändernden Marktbedingungen, besteht die große Gefahr, die Wettbewerbsfähigkeit zu verlieren.

Das Unternehmen braucht eine Technologie, die sowohl mit der riesigen Datenmenge umgehen kann als auch mit Datenbeständen, die über verschiedene Datenquellen und Systeme verteilt sind. Denn das Unternehmen möchte sofortige Ergebnisse. In Zukunft ist dies durch traditionelle Datenbanksysteme nicht mehr möglich, da diese durch die Massendaten extrem belastet werden und so die Verarbeitung der Daten extrem langsam vorgeht. Aussagekräftige Berichte sind die Grundlage für wichtige unternehmerische Entscheidungen.

Ein Paradigmenwechsel ist erforderlich. Dieser Paradigmenwechsel wird durch Agilität/Flexibilität, Massendaten und ‚Total Cost of Ownership‘ hervorgerufen.⁶

⁴ Vgl. IBM (o.J.)

⁵ Vgl. pmOne (o.J.)

⁶ Vgl. IBSolution (2012)

Ein Unternehmen kann als agil bezeichnet werden, wenn es in der Lage ist, schnell und flexibel mit Veränderungen umzugehen, z.B. in kürzester Zeit auf verschiedene Kundenwünsche einzugehen.⁷

Neue Technologien sollen es schaffen, die Effizienz im Unternehmen zu steigern. Man möchte mit einem gleichbleibenden Budget einen vergleichsmäßig höheren Output erzielen. Dabei sollten Unternehmen Open Source Produkte nicht außer Acht lassen, da diese eine ernstzunehmende Konkurrenz zu den kommerziellen Anbietern darstellen.

Um Geschäftsdaten zu verwalten, verwenden viele Unternehmen heutzutage verschiedene Business Intelligence (BI) Tools. Die BI Werkzeuge erleichtern die Aufbereitung und strukturierte Auswertung von Unternehmensdaten. Dies wird ermöglicht durch die Extraktion von Daten von verschiedenen Anwendungssystemen und der darauffolgenden Harmonisierung der Daten. Die verfügbaren Daten werden so für verschiedene Analysemöglichkeiten bereitgestellt und dienen der schnellen Entscheidungsfindung.⁸

Dieser Prozess der Datenintegration ist auch bekannt als ‚Extract-Transform-Load‘ (ETL) Prozess. Bevor Daten in das Datawarehouse befördert werden können, müssen sie in das richtige Format gebracht werden, da sie oft aus verschiedenen Datenquellen stammen. Für dieses Verfahren wird ein ETL-Werkzeug benötigt. Das ETL-Werkzeug hat folgende drei Aufgaben: Die Extraktion von Daten aus verschiedenen Datenquellen, die Transformation dieser Daten und das Laden der aggregierten Daten in das zentrale System.

Unternehmen, die sich entschließen ein ETL-Tool anzuschaffen, haben eine große Auswahl an kommerziellen ETL-Tools und Open Source Produkten. Im Rahmen dieser Seminararbeit werden ausgewählte Open Source ETL-Tools in Abstimmung mit dem Auftraggeber miteinander verglichen und analysiert.

1.2. Zielsetzung

Die vorliegende Seminararbeit dient als Hilfestellung für den Auftraggeber, um das richtige und passende Open Source ETL-Tool zu finden. Das Ziel der Arbeit ist es zum einen, einen Überblick über die ausgewählten Open Source ETL-Tools zu geben und zum anderen einen Prototyp zu erstellen, um so eine Empfehlung für ein Open Source ETL-Werkzeug auszusprechen.

Der Fokus der Arbeit liegt dabei auf dem Vergleich und der Evaluation der verschiedenen ETL-Tools. Dabei werden Eigenschaften der unterschiedlichen Open Source Produkte beleuchtet. Die Arbeit soll als Grundlage für die Auswahl des richtigen ETL-Tools für den

⁷ Vgl. Heines, A. (2012)

⁸ Vgl. Haneke, U. (2012)

Auftraggeber dienen, indem die Produkte basierend auf Funktionsumfang und Einsatzbereich kategorisiert werden. Bei der Evaluation werden Kriterien wie Performance, Benutzerfreundlichkeit, Anzahl der Schnittstellen zur Anbindung unterschiedlicher Quell- und Zielsysteme, Werkzeuge für die regelbasierte Transformation der Daten, Marktrelevanz und Zukunftssicherheit berücksichtigt.

1.3. Vorgehensweise

Diese Seminararbeit behandelt zunächst theoretisch das Thema ETL-Tool. Basierend darauf wird auch der Begriff Open Source erläutert. Es werden neben der Technologie auch die Einsatzbereiche der ETL Werkzeuge beschrieben. Nach dieser Erklärung wird auf die Methodik des Kriterienkatalogs und des Prototyps eingegangen. Im Praxisteil werden die ausgewählten Open Source ETL-Tools untersucht und beschrieben. Dazu wird ein Kriterienkatalog erstellt, welcher zur Bewertung der verschiedenen Tools dient. Neben der Identifizierung der Kriterien muss auch ihre jeweilige Gewichtung festgelegt werden. Nachdem jedes Tool mithilfe des Kriterienkatalogs ausgewertet wurde, kann eine Vorauswahl für den Feinkriterienkatalog getroffen werden. Darauffolgend wird ein Prototyp erstellt. Abschließend folgen eine Empfehlung und ein analysierender Rückblick bezüglich der verglichenen und getesteten ETL-Tools.

2 Extraktion-Transformation-Laden (ETL)

(Sarah Ahmed, Muazzez Algac, Lara Hochhalter, Janine Jakob, Ines Schlenker)

Die Abkürzung ETL steht für den klassischen Prozess der Extraktion, der Transformation und dem Laden von Daten. In der Fachliteratur ist der moderne Extraktion, Laden, Transformation (ELT) Ansatz ebenfalls beschrieben, welcher in Kapitel 2.2 kurz erläutert ist und vom klassischen ETL-Prozess abgegrenzt wird.

2.1 ETL-Prozess

Aufgrund der in Unternehmen stark zunehmenden Menge an Daten, welche oft dezentral gehalten werden, wie auch der Vielzahl an unterschiedlichen und verteilten Informationssystemen werden die Systemlandschaften immer komplexer. Um trotzdem genaue Informationen systemübergreifend erzeugen und speichern zu können, müssen mithilfe des ETL-Prozesses die Daten aus den unterschiedlichen Quellsystemen entsprechend umgewandelt werden, sodass das Data Warehouse angemessene Abfragen durchführen kann.⁹ Dieser Prozess ist einerseits sehr aufwendig und bestimmt signifikant die Qualität des Data Warehouses.

Abbildung 2 zeigt den *klassischen* ETL-Prozess auf, welcher in den nachfolgenden drei Unterkapiteln erläutert wird. Für die Durchführung der Teilschritte des ETL-Prozesses sind unterschiedliche Vorgehensweisen und somit verschiedene Architekturen denkbar. Je nach Anwendungsfall ist die eine Prozessvariante zielführender als die andere. Dennoch verfolgen beide Varianten dasselbe übergeordnete Ziel.¹⁰ Der moderne ELT-Ansatz aus Kapitel 2.2 ist eine Variante dafür.

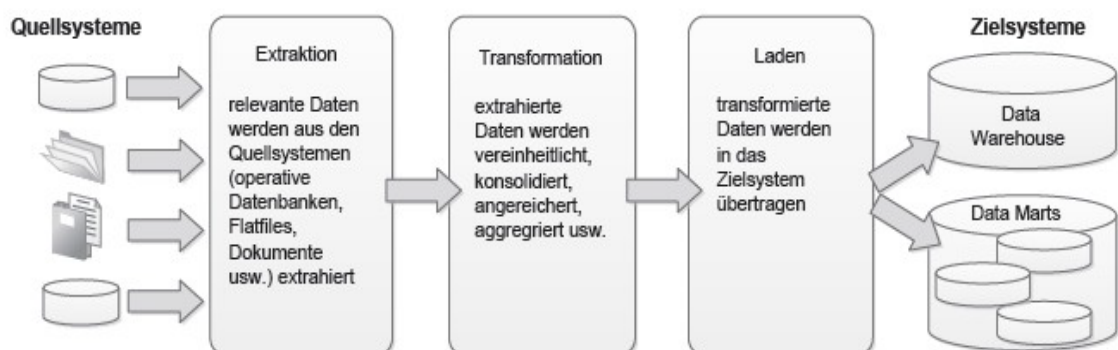


Abb. 2: Allgemeiner ETL-Prozess¹¹

⁹ Vgl. Rossak, I. (2013), S.37

¹⁰ Vgl. ebenda, S. 38

¹¹ Vgl. Rossak, I. (2013), S. 38

Durch solche Überlegungen wie die Folgende entstanden unterschiedliche ETL-Systeme für verschiedene Kundenwünsche, wie sie in dieser Arbeit auch präsentiert werden, bei denen alle oder lediglich einige ausgewählte Schritte des ETL-Prozesses in ein separates System verlagert werden:¹²

Die Daten aus den Quellsystemen können in einen gesonderten Arbeitsbereich innerhalb des operativen Systems extrahiert werden, aufbereitet und anschließend von dort aus in das Zielsystem geladen werden. Das erspart die Kosten und den Aufwand für ein zusätzliches System, vermindert aber die Performance.¹³

2.1.1 Extraktion

Im ersten Schritt des klassischen ETL-Prozesses findet eine Extraktion aller für relevant befundenen Daten statt.¹⁴ Dies kann im ‚push‘- oder auch im ‚pull‘-Verfahren vollzogen werden. Das heißt, dass entweder das Quellsystem die Extraktion der Daten anstößt (push) oder das ETL-System die Daten anfordert (pull).

Wie Abbildung 2 es verdeutlicht, werden im ersten Schritt des ETL-Prozesses Daten aus den Quellsystemen ausgelesen. Ein Quellsystem kann zum Beispiel eine operative Datenbank sein, aus der die entsprechenden Daten gezogen werden. Die Daten liegen zumeist in verschiedenen Formaten vor. Sie können wie einfache Tabellen in strukturierter Form existieren, wie Daten aus Comma-separated Values (CSV)- oder Extensible Markup Language (XML)-Dateien in semistrukturierter Form oder auch wie Daten aus E-Mails unstrukturiert sein. Je nach Quellsystem werden für diesen Vorgang verschiedene Techniken eingesetzt. Bei der Extraktion der Daten aus Datenbanken kommen zumeist die Bordmittel der Datenbanken zum Einsatz, wie ‚Bulk Loader‘¹⁵- oder Export-Funktionalitäten. Für semistrukturierte Daten werden hierfür andere spezielle Extraktoren verwendet. Unabhängig davon ist jederzeit sicherzustellen, dass die Quellsysteme erreichbar sind und dass auf die benötigten Daten zugegriffen werden kann. Dies beinhaltet sowohl die technische wie auch die rechtliche Absicherung des Extraktionsprozesses. Bereits die unterschiedlichen Datenformate können für das ETL-System eine Herausforderung darstellen.

Der jeweilige Zeitpunkt und Umfang der Datenextraktion legt ein Unternehmen anhand der verfolgten Ziele innerhalb seiner Strategie fest. Es kann entweder Snapshot-basiert der gesamte Datenbestand zu dem ausgewählten Zeitpunkt extrahiert werden oder Delta-

¹² Vgl. Rossak, I. (2013), S. 38

¹³ Vgl. ebenda

¹⁴ Vgl. ebenda, S. 39

¹⁵ Vgl., dazu auch Stackoverflow (2014)

basiert, was die Herausnahme der Daten mit angefallenen Änderungen seit der letzten Extraktion bedeutet. Mögliche zeitliche Varianten dafür sind zum Beispiel:¹⁶

- Periodische Extraktion:
Die Daten werden zu einem festgelegten Zeitpunkt in regelmäßigen Zeitabständen extrahiert, z.B. am Ende der Woche.
- Anfragegesteuerte Extraktion:
Die Daten werden je nach Bedarf für weitere Analysen extrahiert, z.B. bei Aufnahme neuer Daten in die Quellsysteme.
- Ereignisgesteuerte Extraktion:
Die Datenextraktion beginnt durch ein eingetretenes Ereignis, wie z.B. das Übersteigen des Messwertes von einem Schwellenwert.
- Permanente Extraktion:
Die Daten werden kontinuierlich zu jedem Zeitpunkt extrahiert, wie z.B. bei Verfolgung von Währungskursen oder Börsenwerten.

2.1.2 Transformation

Die Transformation der Daten gilt als der wichtigste Schritt im gesamten ETL-Prozess, jedoch gleichzeitig auch als die zeit- und arbeitsintensivste Aufgabe. Dieser Teilschritt beschäftigt sich damit, dass die extrahierten Daten vereinheitlicht, konsolidiert, angereichert und aggregiert werden. Die syntaktische, strukturelle und semantische Heterogenität der einzelnen Quellsysteme sind der Grund für den hohen Aufwand. Folgende drei Arten von Transformationen werden benötigt:¹⁷

- Syntaktische Transformation:
Diese Art der Transformation beinhaltet eine Homogenisierung und Konsolidierung in Bezug auf Zeichensätze, Datenformate, Datentypen etc., wie beispielsweise die Konvertierung aller Daten in Unicode, die Umwandlung von ‚bit‘ in ‚boolean‘ usw.
- Strukturelle Transformation:
Es setzt auf eine Homogenisierung und Konsolidierung in Bezug auf Modellierungsebene, Schlüsselwerte, Standards usw. Beispiele dafür sind die Vereinheitlichung von Vorschriften oder die Auflösung von Wertelisten in eigene Tabellen.

¹⁶ Vgl. Rossak, I. (2013), S. 39

¹⁷ Vgl. Rossak, I. (2013), S. 39f.

- Semantische Transformation:

Dieser Typ der Transformation beinhaltet die Homogenisierung und Konsolidierung bezüglich Benennungen, Bemaßungen, verwendete Skalen etc., wie z.B. das Längenmaß in Meter, Währungswerte in Euro oder alle Varianten einer Anrede in das Geschlecht auf ‚w‘ und ‚m‘ vereinheitlicht werden.

Viele Transformationen können sogar während der Extraktion der Daten mithilfe von Datenbankmitteln, wie z.B. Structured Query Language (SQL) Abfragen durchgeführt werden. Zum Beispiel können mithilfe der Funktion *REPLACE* die im Quellsystem hinterlegten Werte ‚Frau‘ und ‚Herr‘ (für die Anrede) in die Werte ‚w‘ und ‚m‘ (für das Geschlecht) im Arbeitsbereich konvertiert werden. Etwas aufwendiger, aber dennoch unter Verwendung von Datenbankmitteln machbar, ist die Abbildung von Regeln, welche sich in Quell- und Zielsystemen differenzieren. Der Vorgang der Übertragung von bereits transformierten Daten in den neuen Arbeitsbereich ist oftmals effizienter. ETL-Werkzeuge unterstützen den Vorgang der Verarbeitung der extrahierten Daten mit vorgefertigten Transformationskomponenten. Der Benutzer des ETL-Systems kann die gewünschte Funktion anhand eines Katalogs im ETL-Tool auswählen, welche er auf einer grafischen Benutzeroberfläche den vorher entworfenen Transformationsschritten anordnet und diese miteinander verbindet. Anschließend werden im Hintergrund die entsprechenden Anfragen an die Datenbank generiert.¹⁸

2.1.3 Laden

Der letzte Teil des klassischen ETL-Prozesses beschäftigt sich mit dem Laden der extrahierten und transformierten Daten in eine Zieldatenbank (siehe Abbildung 2), welche dort anschließend dauerhaft gespeichert werden. Für diesen Vorgang stehen Bordmittel der Zieldatenbank bereit, wie z.B. spezielle Bulk-Load-Verfahren, welche die Befüllung der Zieltabellen durch dementsprechend formatierte Textdateien unterstützen. Für die Befüllung der Zieldatenbank bieten ETL-Systeme entsprechende Komponenten zur Unterstützung bei der Generierung des erforderlichen Programmcodes an. Abhängig von der Menge und Komplexität der zu ladenden Daten, kann der letzte Teilschritt des ETL-Prozesses ein langwieriges Vorgehen bedeuten.¹⁹

2.2 ELT-Prozess

Zur Vervollständigung des Themas folgt noch eine Erläuterung des modernen ELT-Prozesses in Abgrenzung zum ETL-Prozess. Trotz einiger Vorteile, die der moderne ELT-

¹⁸ Vgl. Rossak, I. (2013), S. 39f.

¹⁹ Vgl. Ebenda, S. 40f.

Ansatz bietet, liegt der Fokus dieser Seminararbeit auf den ausgewählten ETL-Systemen und dem klassischen ETL-Prozess. Deshalb wird der ELT-Ansatz lediglich kurz angesprochen. Abbildung 3 illustriert den ELT-Prozess.

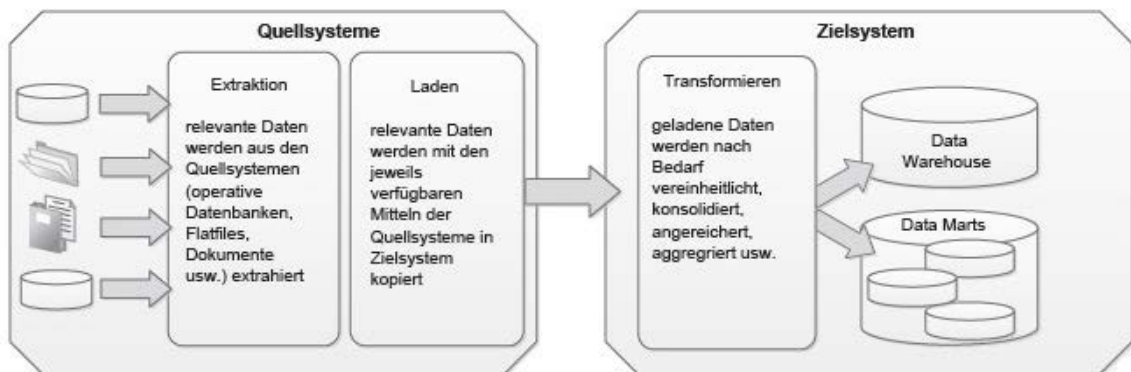


Abb. 3: ELT-Prozess²⁰

Mit der Zunahme an neuen Funktionen in den verwendeten Datenbankmanagementsystemen ergibt sich die Möglichkeit mehr Tätigkeiten der klassischen ETL-Systeme in Datenbanken zu verlegen. Im Vergleich zum klassischen ETL-Prozess sieht der moderne Ansatz eine Entkoppelung der Teilschritte vor und beinhaltet zusätzlich eine Verlagerung der Transformation in die Zieldatenbank. Genauso wie beim ETL-Ansatz beginnt der Prozess mit der Extraktion der entsprechenden Daten aus den Quellsystemen. Der Unterschied liegt jedoch darin, dass die überführten Daten direkt in das Zielsystem kopiert werden. Die benötigten Transformationen erfolgen erst auf Anfrage. Die Tätigkeit der Datentransformation erledigt die Infrastruktur des Zielsystems, zum Beispiel mithilfe von SQL-Abfragen.²¹

Dadurch dass der Transformationsprozess nun im Zielsystem erfolgt, wird dieses stärker belastet. Der größere Vorteil ergibt sich aber daraus, dass sich die Verarbeitungszeit verkürzt, da die Anfragen eher mengen- statt satzorientiert sind und sich die Möglichkeit einer Parallelverarbeitung ergibt. Ein weiteres Argument für den ELT-Prozess ist die Verringerung der Netzwerklast, da die extrahierten Daten nicht durch ein zusätzliches System geführt werden müssen, wie es Abbildung 3 bildlich darstellt. Das ELT-Vorgehen bietet eine hohe Flexibilität bei gleichzeitig verringertem Aufwand, da lediglich die für die Fragestellung benötigten Daten gewählt und transformiert werden. Ein Nachteil dieses Verfahrens ist der Verlust des Überblicks über die Herkunft der Daten bzw. des Datenflusses.²²

²⁰ Vgl. Rossak, I. (2013), S. 42

²¹ Vgl. ebenda, S. 41f.

²² Vgl. ebenda

2.3 Open Source

Das Konzept, nach dem Computerprogramme mit ihrem Quellcode ausgeliefert werden, nennt sich Open Source. Diesen Quellcode kann jeder einsehen und verändern. Open Source Software wird auch oft *kostenlose* Software genannt, was aber nicht immer der Fall sein muss. Die Schnittmenge von sogenannter freier Software und Open Source Software ist sehr groß, aber nicht identisch. Anhand folgender Kriterien kann Open Source Software genauer definiert und von anderer Software, wie auch proprietärer Software, unterschieden werden.²³ Erst wenn ein Programm alle zehn Kriterien der Open Source Initiative erfüllt hat, handelt es sich um Open Source Software.²⁴

1. Die freie Weitergabe der Software darf nicht durch die Lizenz verboten werden.
2. Das Programm muss den Quellcode beinhalten.
3. Die Lizenz muss Veränderungen und eine abgeleitete Software zulassen.
4. Die Integrität des Quellcodes des Autors muss gegeben sein.
5. Die Lizenz darf keine Personen oder Gruppen diskriminieren.
6. Die Lizenz darf keine Einschränkung bezüglich des Einsatzfeldes geben.
7. Die Rechte an einer Software müssen auf alle Personen übergehen, an die diese weitergegeben wurde.
8. Die Lizenz darf nicht auf ein bestimmtes Produkt beschränkt werden.
9. Die Lizenz darf andere Software nicht einschränken.
10. Die Lizenz muss technologieneutral sein.

Ob ein Unternehmen sich für proprietäre oder Open Source Software entscheidet, hängt von einzelnen Faktoren ab, die eine Firma für sich individuell analysieren und gewichten muss. Zu berücksichtigende Kriterien sind beispielsweise Anpassbarkeit, Sicherheit, Support, Kosten oder auch Zukunftssicherheit der Software.

2.4 Open Source Lizenzen

Genauso wie bei proprietärer Software, werden bei Open Source Software die Lizenzbedingungen festgelegt, die beschreiben, was der Nutzer mit der Software machen darf. Diese Lizenzen werden nach Copyleft-Prinzip in Copyleft und Non-Copyleft unterteilt.

²³ Vgl. Gabler Wirtschaftslexikon (o.J.); Vgl. MacFarlane, F. (2013), S. 6

²⁴ Vgl. Open Source Initiative (2014); Vgl. MacFarlane, F. (2013), S. 7 ff.

Copyleft wird vom Wort Copyright abgeleitet. Dieses Prinzip stellt sicher, dass Open Source Software auch in weiterentwickelten Versionen von jedem vervielfältigt, verändert, verbreitet und öffentlich zugänglich gemacht werden kann.²⁵

Die verschiedenen Typen von Copyleft sind folgende:²⁶

- Strenges Copyleft:
Die modifizierte Software darf wie die ursprüngliche Software nur unter der gleichen Lizenz veröffentlicht und weitergegeben werden.
- Beschränktes Copyleft:
Nur das Ursprungswerk verliert die Lizenz nicht. Die Modifikationen, die in einer eigenen Datei gespeichert werden, können unter einer anderen Lizenz als die restliche Software weitergegeben werden.
- Non-Copyleft:
Die modifizierte Software muss nicht unter der Lizenz des Ursprungswerks stehen.

Zu den berühmtesten und wichtigsten Open Source Lizenzen zählen GNU General Public License (GPL) Version 2 und Version 3, GNU Lesser General Public License (LGPL) und Apache License.²⁷

- GNU General Public License:
Das ist eine Lizenz mit strengem Copyleft. Etwa 60% der Open Source Software untersteht dieser Lizenz, darunter auch so ein bekanntes Programm wie Linux. Sie ist die verbreitetste und wichtigste Open Source Lizenz.
- Lesser General Public License:
Diese Lizenz ist speziell für Programmbibliotheken gedacht und hat ein beschränktes Copyleft. Sie erlaubt die Einbindung von Software nicht nur mit anderer Open Source Software, sondern auch mit proprietären Programmen.
- Apache License:
Sie gehört zu den Non-Copyleft Lizenzen und wurde von der Apache Software Foundation für ihre eigene OSS entwickelt. Der Apache Webserver ist das wichtigste Programm, das dieser Lizenz untersteht.

²⁵ Vgl. Bandel, G. (2010), S. 10; Vgl. Schaaf, A. (2013), S. 17

²⁶ Vgl. Schaaf, A. (2013), S. 17

²⁷ Vgl. ifrOSS (2014)

3 Methodik

(Sarah Ahmed, Muazzez Algac, Lara Hochhalter, Janine Jakob, Ines Schlenker)

3.1 Kriterienkatalog

Heutzutage gibt es für die meisten Anwendungsbeispiele eine Unzahl an unterschiedlichen Software Anbietern. Um eine Übersicht zu erlangen, wird meist ein Kriterienkatalog angewendet, um die unterschiedlichen Softwareprogramme zu bewerten und zu vergleichen.

Ein Kriterienkatalog ist grundsätzlich zusammengesetzt aus Kennzahlen und Indikatoren. Nur durch das Zusammenspiel dieser lässt sich eine ganzheitliche Bewertung festlegen. „Kennzahlen alleine können nur unzureichend alle Prozesse, die ein System ausmachen, abbilden und einer Bewertung zugänglich machen. Indikatoren sollen es ermöglichen, nicht exakt abbildbare, nur indirekt messbare Merkmale eines Systems (immaterielle Größen wie Benutzerzufriedenheit, Bedienbarkeit, etc.) zu beschreiben und zu bewerten.“²⁸

Ein einheitlicher Kriterienkatalog bietet die Möglichkeit die vorhandene Software nach den gleichen Gesichtspunkten einordnen zu können und macht eine spätere Auswahl leichter.

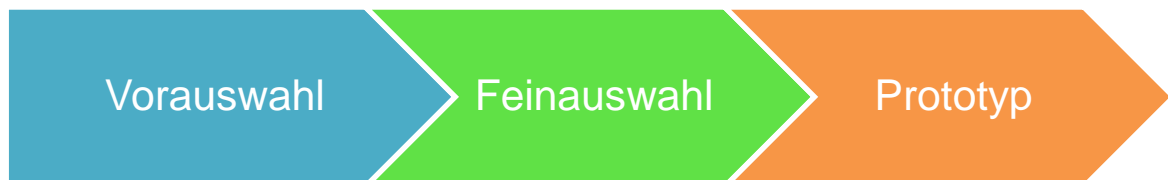


Abb. 4: Auswahlverfahren²⁹

Um einen Kriterienkatalog anzuwenden, muss zuvor eine Vorauswahl der vorhandenen Softwareanbieter getroffen werden, da die Anzahl für eine Feinbewertung mithilfe eines Kriterienkataloges meist zu groß ist. Im nächsten Schritt werden die ausgewählten Softwareprogramme einen Kriterienkatalog durchlaufen. Im letzten und endgültigen Schritt werden die nach den Kriterien bestgewichteten Top zwei Anbieter selektiert und durch einen Praxistest mithilfe eines Prototyps noch einmal gewichtet.³⁰ Zur Bewertung mithilfe eines Prototyps siehe Kapitel 3.2.

Ein Kriterienkatalog besteht aus unterschiedlichen Indikatoren und Kennzahlen, sowie einer Gewichtung und Skala für jede Kategorie.

²⁸ Vgl. Stöckl, C. (2006)

²⁹ Vgl. Gustke, S. (2011)

³⁰ Vgl. Klüpfel, S. / Mayer, T. (2007)

Nach Meier sollte ein Kriterienkatalog folgendermaßen aussehen und angewendet werden:

1. „Die Beurteilung erfolgt durch einen - im Idealfall geschulten - Fachmann
2. Die Kriterien sind vollständig, valide und verlässlich.
3. Die Kriterien liegen in strukturierter Form vor und sind in Kriteriums Kategorien (mit Überschriften) unterteilt.
4. Diese Kriteriums Kategorien sind einzeln als Prüfinstrumente anwendbar.
5. Im Idealfall ist ein umfangreicher Katalog zu einer "Kurzprüfliste" extrahiert.
6. Die Kriterien sind sachlich korrekt, verständlich, treffend und knapp formuliert.
7. Der Katalog ist einfach und im Idealfall elektronisch auswertbar. Voraussetzung hierfür ist u.a., dass die Kriterien in Sachzusammenhänge vorstrukturiert und die Bewertungsskalen einfach konstruiert sind (z.B. Ja/Nein-Antworten).
8. Der Katalog verfügt über sogenannte Filter, d.h. Mechanismen, die eingangs den betreffenden Anwendungsbereich Unterrichtszusammenhang, die Zielgruppe u.a. erfragen und so den Anwender von vornherein zu den für ihn individuell relevanten Kriterien führt. Diese Filter beeinflussen eine Qualitätsprüfung außerordentlich.
9. Der Kriterienkatalog ist sprachlich und grafisch einwandfrei und übersichtlich zu bearbeiten.“³¹

Eine genaue Bewertung der Software erfolgt mithilfe von Metriken und Beurteilungskriterien. Hierbei kann die ISO 9126 als Grundlage genommen werden, sie beschreibt die wesentlichen Qualitätsmerkmale von Softwareprogrammen.³²

Nach der DIN ISO/IEC 9126 kann die externe und interne Softwarequalität in sechs unterschiedliche Kategorien eingeteilt werden. Diese werden wiederum in Unterkategorien aufgeteilt.³³ Die Norm ISO/IEC 9126 weist vor allem im deutschen Raum einen sehr hohen Nutzungsgrad auf und bietet sich deshalb perfekt für die Bewertung von Softwareprogrammen an. Es muss bei der späteren Auswahl der Kriterien für die ETL Software aber darauf geachtet werden, dass jede der folgenden Kriterien auch mit einer Open Source Software gültig sein kann.³⁴ Jede Kategorie wird je nach Anforderungen mit einer gewissen Gewichtung versehen, um die Erwartungen des Kunden bestmöglich zu treffen und eine realitätsnahe Bewertung abgeben zu können. Folgende Grafik soll eine Übersicht der unterschiedlichen Qualitätsmerkmale bieten:

³¹ Vgl. Meier, A. (1995), S. 189ff.

³² Vgl. Dialog Data (o.J.)

³³ Herzwurm, G. / Mikusz, M.,(2013)

³⁴ Vgl. O.A. (o.J.) c

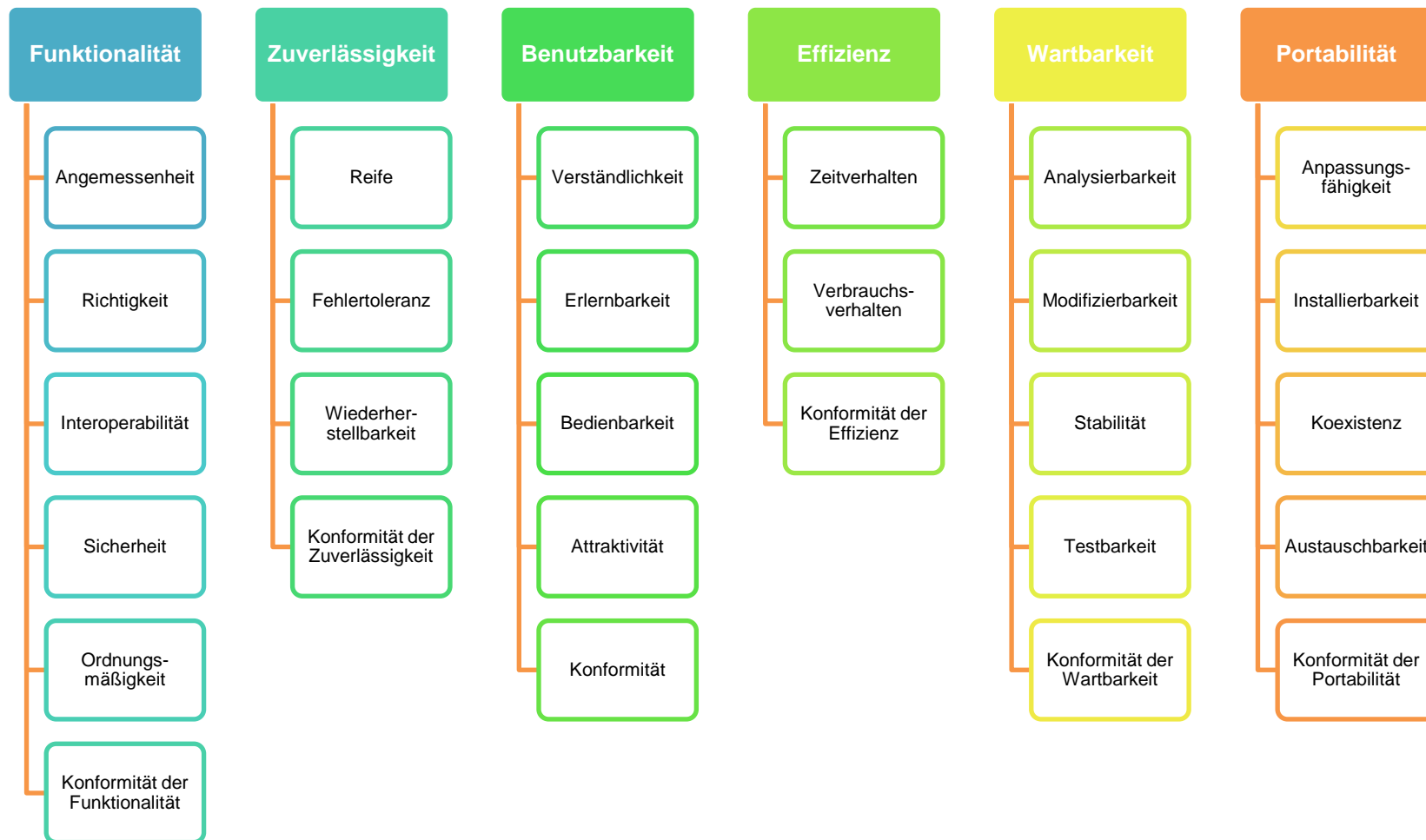


Abb. 5: DIN ISO 9126³⁵

³⁵ Mit Änderungen entnommen aus: Herzwurm, G. / Mikusz, M.,(2013)

3.2 Prototyp

Data Warehouses sind teure, komplexe Projekte, weshalb die Implementierung eine solide Planung voraussetzt. Um Gesamtkosten des Projekts zu reduzieren, gibt es eine einfache Möglichkeit in der Vorbereitungsphase.

Die Hauptkomponente eines Data Warehouse-Projekts ist die ETL Entwicklung. In der Praxis unterscheiden sich die Kosten zwar von Projekt zu Projekt, aber im Durchschnitt macht der ETL Prozess etwa 70% der Entwicklungskosten aus.³⁶ Einer der Kostenfaktoren ist der relativ hohe Preis der ETL- Entwicklungsressourcen. Zum Beispiel verlangt ein ETL-Entwickler oft einen 30-40% höheren Stundensatz als ein Business Intelligence Report Writer'.³⁷ Die Herstellung eines ETL Prototyps gibt einen Einblick und verringert die Kosten durch eine Verkürzung des ETL-Entwicklungsprozesses. Mithilfe dieser Methode kann eine optimale Nutzung der teuren ETL-Ressourcen ermöglicht werden.

Es gibt verschiedene Kriterien, die die Kosten und die Komplexität eines ETL-Prozesses beeinflussen. Die folgenden Faktoren haben dabei einen großen Einfluss auf die Anzahl und die Komplexität von ETL und damit auf ihre Kosten:

- Die Anzahl der verschiedenen Datenquellen, die am Prozess beteiligt sind
- Die Konsistenz in Bezug auf die Stammdatendefinitionen zwischen den Systemen
- Das Niveau der Datenqualität in den Systemen

Idealerweise sollte man einen guten Einfluss auf diese drei Faktoren bekommen, bevor die ETL-Entwickler miteinbezogen und ein hohes Kapital eingesetzt wird, um sicherzustellen, dass die Bedürfnisse der Nutzer befriedigt werden können.

Ein Teil der Vorbereitung ist die Erstellung eines ‚Proof of Concepts‘ oder eines Prototypen, der die wichtigsten Bestandteile darstellt und die Kriterien prüft. Der effektivste Weg ist dabei ein Prototyp,

- ... der nur so tief auf die Daten eingeht, sodass
 - ... alle Datenquellen, die benötigt werden, integriert werden.
 - ... eine hohe Prüfung der Stammdaten und der Datenqualität durchgeführt werden kann.
- ... dessen Output nur so hoch ist, dass
 - ... Endbenutzern beispielsweise Berichte geliefert werden können, um einen Eindruck zu bekommen.
 - ... der Funktionsumfang eingegrenzt werden kann.

³⁶ Vgl. Standen, J.(2009)

³⁷ Vgl. ebenda

Das Ziel eines Data Warehouse Prototyps ist es, die zugrunde liegenden Daten kennenzulernen und dazu in der Lage zu sein verschiedene Datentransformationstechniken auszuprobieren, indem man diese auf reale Daten anwendet.

Das Ziel ist weder ein fertiges Produkt zu liefern, noch tatsächlich nutzbare Berichte für Endnutzer herzustellen. Sondern das Ziel ist ein Versuchsmodell eines geplanten Produkts. Es dient der Vorbereitung des Endprodukts und hilft dabei die Tauglichkeit und Akzeptanz zu prüfen.

Ein Prototyp sollte sich auf die Dimensionen und Datenquellen, die als größte Herausforderung gesehen werden, fokussieren. Der springende Punkt bei der Erstellung eines Prototyps ist es so schnell wie möglich voranzukommen, möglichst viel Feedback von Nutzern zu bekommen und in der Lage zu sein, unerwünschte Entdeckungen bei der eigentlichen Data Warehouse Entwicklung zu vermeiden. Falls man eine Problematik bei der Datenqualität entdecken sollte, kommt es beim Prototyp nicht darauf an, die Probleme gleich zu lösen, sondern sie erst zu identifizieren. Man sollte bezüglich der Problematiken offen gegenüber Nutzern sein, damit diese sich entsprechend darauf einstellen können.

Wenn man keinen Prototypen herstellen würde, und dann bei der Produktion Datenqualitätsfragen aufkommen würden, wird dies zu einer definitiven Verzögerung des Projekts führen und Kosten in einem unbekanntem Ausmaß steigern.

Zusammengefasst kann man davon ausgehen, dass Prototypen...

- das Risiko minimieren, dass das Projekt den zeitlichen und finanziellen Umfang sprengt.
- den Vorteil haben, Endnutzern eine Probe des tatsächlichen Produkts zu geben.
- die Menge der Nacharbeit im ETL-Code präventiv minimieren, da unterschiedliche Datentransformationssätze frühzeitig getestet werden.
- das Risiko einer teuren ETL Entwicklungsphase verringern, da unbekannte Datenqualitätsprobleme vorab identifiziert werden können.

Oft werden Prototypen in einer Kombination von Excel mit MS Access oder anderen Datenbanken aufgebaut. In der Theorie kann dies funktionieren, allerdings ergibt sich oftmals ein Problem bei Excel, da es ernsthafte Probleme im Umgang mit größeren Datensätzen hat. Es darf nicht vergessen werden, dass es darum geht, einen Prototyp zu generieren und nicht darum den SQL-Code schon zu programmieren. Verschiedene Datentypen, Feldformate, Spaltennamensregelungen etc. verhindern oftmals ein schnelles Vorgehen und daher muss die Zeit miteinberechnet werden.

Für einen ETL Prototypen benötigt man in der Regel ein einfaches ETL-Tool, das die grundlegenden Arten der Funktionalität und die grafische Benutzeroberfläche von High-End-ETL-Tools bietet. Es sollte auch eine flexible Handhabung von unterschiedlichen Datentypen bieten, die aus unterschiedlichen Quellen, darunter auch Excel, extrahiert werden können. Insgesamt kann man davon ausgehen, dass man ein klares, fokussiertes und einfach zu bedienendes ETL-Tool nutzen sollte, ohne hohe Kosten zu produzieren.

4 Beschreibung der ETL Anwendungen

(Sarah Ahmed, Muazzez Algac, Lara Hochhalter, Janine Jakob, Ines Schlenker)

Der Praxisteil dieser Seminararbeit beschäftigt sich zunächst mit der Beschreibung der, nach Absprache mit dem Auftraggeber, ausgewählten Open Source ETL Anwendungen Kettle, Adeptia, Jitterbit, Xineo und CloverETL. Dem folgt der speziell erstellte Kriterienkatalog und dessen Gewichtung, sowie abschließend die Bewertung der favorisierten ETL Tools anhand des Feinkriterienkataloges.

4.1 Kettle



Abb. 6: Kettle Logo³⁸

Kettle, auch bekannt unter dem Namen Pentaho Data Integration (PDI), ist eines der führenden Open Source ETL-Tools. Es ist seit 2006 Teil der Business Intelligence Lösung von Pentaho. Pentaho gehört zu den weltweit erfolgreichsten Open Source Business Intelligence (OSBI) Produktanbietern. Das Open Source Unternehmen hat bereits verschiedene Auszeichnungen bekommen, unter anderem den 'Best of Open Source Award' für die BI Suite des Unternehmens. Pentaho-Produkte werden auch von großen Unternehmen wie der OTTO Group und ProSiebenSat1 benutzt. Zudem arbeitet Pentaho mit den bekannten IT-Unternehmen wie Sun, HP und RedHat. Die Community wächst von Tag zu Tag. Momentan arbeiten mehr als 30.000 Mitglieder an der Weiterentwicklung von Pentaho. Laut Sourceforge.net ist Pentaho die Nummer eins im Bereich der OSBI Produkten. Die BI Suite wurde bereits mehr als zwei Millionen Mal heruntergeladen.³⁹

In diesem Kapitel wird näher auf das Unternehmen Pentaho eingegangen und wie sich deren ETL-Tool Kettle/PDI von anderen Open Source ETL-Tools unterscheidet.

Pentaho

Neben JasperSoft ist Pentaho einer der bekanntesten Open Source Software Anbieter. Das Unternehmen wurde im Jahre 2004 von Richard Daley und James Dixon gegründet. Die Firma, mit Sitz in Orlando, verfügt über zwei komplette Business Intelligence Suiten:

³⁸ Enthalten in: Passioned Group (o.J.)

³⁹ Vgl. Haneke, U. et al. (2010), S.177

Pentaho Community Edition (CE) und Pentaho Enterprise Edition (EE). CE ist die Open Source Version und EE ist die zertifizierte kommerzielle Lösung. Die Business Intelligence Suite beinhaltet eine ETL Komponente, sowie mehrere Analysekomponenten, Reportingkomponenten und Workflowkomponenten.

Die Community Edition steht unter folgenden Lizenzarten: GNU Public License v2, Lesser General Public License 2v, Common Public License und Mozilla Public License. Die Community Edition stellt ihre verschiedenen Komponenten zum Einzeldownload bereit. Sie werden nicht als Gesamtpaket angeboten. Pentaho erlaubt so die persönliche Zusammenstellung der einzelnen Komponenten nach individuellen Bedürfnissen. Komponenten wie Mondrian, Kettle und Weka sind von anderen Open Source Projekten erworben.

Seit 2006 sind Mondrian und Kettle Bestandteile der OSBI Suite von Pentaho. Die Pentaho Enterprise Edition verfügt über dieselben Software-Bestandteile wie die Community Edition. Jedoch sind die Komponenten der EE zertifiziert, da sie stabiler laufen. Durch den professionellen Support und Zusatztools, ist die EE im Vergleich zu CE kostenpflichtig.⁴⁰

Die Pentaho Business Intelligence Plattform besteht aus unterschiedlichen Komponenten. Die Seminararbeit fokussiert sich dabei auf das Business Intelligence Framework welches aus ETL, Webdiensten und dem Sicherheitszentrum besteht.

Die Architektur der Pentaho BI Plattform ist eine Zusammenführung von eigenem Source Code und bereits getestetem Open Source Code. Die Kombination von Pentaho BI Framework und Pentaho BI Komponenten (Reporterstellung, Analyse etc.) ergeben den Pentaho Server. Der Server läuft auf einem J2EE-fähigen Webserver wie Apache.

ETL mit Pentaho Data Integration

Wie bereits beschrieben, verfügt die Pentaho BI Suite auch über eine ETL Komponente. Die Komponente Pentaho Data Integration (PDI) ist für die Implementierung des ETL-Prozesses zuständig. PDI ist aus dem sehr bekannten Open Source Projekt Kettle hervorgegangen und bietet vier verschiedene Werkzeuge für die ETL Prozesse. Die einzelnen Werkzeuge können nach dem Download ohne Installation einzeln entpackt und verwendet werden. Somit erlaubt Pentaho den Anwendern den Einsatz von PDI mit Pentaho oder als eigenständiges Tool.

⁴⁰ Vgl. Haneke, U. et al. (2010), S.178

Aufbau von PDI

Das ETL-Tool besteht aus folgenden vier Einzelkomponenten:

- **Spoon:** grafische Benutzerschnittstelle
- **Kitchen:** Stapelverarbeitung von Transformationen
- **Pan:** Stapelverarbeitung von Job-Ketten
- **Carte:** Aktivierung von ‚Remote-Server‘ für die Spoon-Transformationen

Dabei stellt Spoon die grafische Oberfläche dar und wird für das Design genutzt. Es wird eingesetzt, um ETL-Vorgänge zu erstellen. Die restlichen Tools sind für die Prozessausführungen zuständig.⁴¹

Das wesentliche Gestaltungselement des ETL-Werkzeugs ist die Transformation von Daten. Vereinfacht gesagt, umfassen Transformationen den Datenimport und -export. Eine Transformation besteht aus einer Abfolge von Sprüngen (engl. hops) und Prozessschritten (engl. steps) und beschreibt den Datenstrom, der in eine bestimmte Richtung fließt. Ein Step ist z.B. das Laden von Daten einer CSV-Datei oder einer Datenbank. Die Steps werden durch ‚drag-and-drop‘ in die Transformation gefördert. Dabei stehen die Steps in einer Baumstruktur zu den ausgewählten Daten. In der grafischen Entwicklungsumgebung Spoon werden derartige Steps als vordefinierte Bausteine zur Verfügung gestellt, um den Anwender bei der Step-Definition zu unterstützen. Die Verbindung der Steps erfolgt durch den sogenannten Hops. Einzelne Steps werden in eine bestimmte Richtung für einen Abfluss des Transformationsprozesses zusammengebündelt. Zusätzlich zu den Transformationen werden auch ‚Jobs‘ benötigt, um ETL-Vorgänge darzustellen. Jobs sind Ablaufpläne bzw. eine Zusammenfassung mehrerer Transformationen. Dabei können Transformationsdateien direkt oder durch einen Job-Eintrag ausgeführt werden. Jobs bestehen aus Job-Einträgen, die eine sequentielle Ausführung von Transformationsdateien ermöglichen. Jobs sind sozusagen für die Umsetzung von mehreren Transformationen und verschiedenen Prozessen zuständig. Nach der Transformation werden die Daten in einer Datei im XML-Format mit der Endung *.ktr* gespeichert oder man legt die Transformationen und Jobs in einem Repository ab. Verschiedene Datenbankformate wie MySQL oder sogar Formate von anderen Open Source Anbietern wie Palo MOLAP Server können für die Erstellung eines Repository verwendet werden. Je nach Datenbankformat, kann der Zugriff durch Treiber wie Java Database Connectivity (JDBC) oder Java Naming and Directory Interface (JNDI) erfolgen.⁴²

⁴¹ Vgl. Haneke, U. et al. (2010), S.182

⁴² Vgl. ebenda

PDI basiert auf der Entwicklungsumgebung Eclipse und ist erweiterbar mit Plug-Ins. Hierzu gibt es kostenfreie und kostenpflichtige Plug-Ins, wie das SAP-Plug-In ‚ProERPconn‘. Mit diesem Plug-In wird der Datenaustausch mit SAP-Systemen ermöglicht.⁴³

Nutzung

Die Bedienung der Entwicklungsumgebung ist einfach und verständlich aufgrund des übersichtlichen Aufbaus. Der Anwender kann durch einen Mausklick Steps von Transformationen erstellen. Die Hops zeigen dabei die Verbindungspfeile zwischen den Steps an. Für den Import und Export der Daten wird eine Datenbankbindung benötigt. Die Anbindung für den Datentransfer entsteht einmalig im View-Fenster und ist für den mehrfachen Gebrauch verfügbar.⁴⁴

Spoon ist in der Version 5.0.1 in englischer Sprache erhältlich. Der Anwender kann sofort auf der Startseite eine Verbindung zu einem existierenden Repository herstellen oder darauf verzichten. Es ist jederzeit möglich Repositories zu erstellen, zu bearbeiten und zu löschen.⁴⁵

4.2 Adeptia



Abb. 7: Adeptia Logo⁴⁶

Adeptia wurde im Juni 2000 gegründet und bietet seitdem Software für Datenintegration, Enterprise Application Integration (EAI), Business-to-Business (B2B) Integration, sowie Business Process Management (BPM) und Enterprise Business Integration mit dem Ziel an, einen breiteren Markt anzusprechen als die laut Fortune 1000 umsatzstärksten amerikanischen Unternehmen⁴⁷. Dies will Adeptia ermöglichen, indem sie eine leistungsfähige und kostengünstige Lösung anbieten, welche einfach zu bedienen und ebenfalls schnell zu implementieren ist. Des Weiteren gibt Adeptia auf seiner Homepage an, der einzige Softwareanbieter zu sein, dessen Produkte alle diese Eigenschaften erfüllen und in einem ganzheitlichen Angebot für den Kunden zur Verfügung stehen. Adeptias Technologien sind zu 100% webbasiert und somit auf jeder Plattform verwendbar.⁴⁸

⁴³ Vgl. IT-Novum GmbH (2009), S. 22

⁴⁴ Vgl. ebenda

⁴⁵ Vgl. Haneke, U. et al. (2010), S.182

⁴⁶ Enthalten in: Frevvo (o.J.)

⁴⁷ Vgl. Adeptia (o.J)

⁴⁸ Vgl. ebenda

Zusammengefasst rühmt sich Adeptia damit, mit einem einzigen Produkt all das anbieten zu können, was Mega-Hersteller mit mehreren Produkten, die durch Unternehmenserwerbe angesammelt wurden, bereitstellen.⁴⁹

Als Referenzen für eine erfolgreiche Implementierung ihrer Software, listet Adeptia Kunden wie die Lincoln Financial Group, AT&T/SBC, Pepsi, General Motors, Meadowbrook Insurance, Lawrence Livermore National Lab, Mercedes Benz, sowie die Tower Group und Chrysler auf.⁵⁰

ETL Suite – Allgemeines und Bestandteile

Adeptias ETL Suite kombiniert den Datentransport mit einem leistungsstarken Metadaten-Management und einer Datentransformationsfähigkeit, wobei eine serviceorientierte Architektur (SOA), sowie ein prozessbasiertes Modell für die Software verwendet wird. Dadurch können flexible und lose verbundene Datenflüsse entwickelt werden, welche jeder beliebigen Aggregation oder Data Warehousing Lösung zugutekommen.⁵¹ Die ETL Software von Adeptia ist somit ideal geeignet für ein ETL-Szenario.

Die hier beschriebene Software von Adeptia zur Datenintegration zeichnet sich durch eine grafische und prozessbasierte Integration und Datenabbildung aus, mit der Datenquellen in kürzester Zeit mit Stakeholdern verknüpft werden können und somit auch schwierige Herausforderungen bei der unternehmensweiten Datenintegration gelöst werden können.⁵²

Die Adeptia ETL Suite ist also eine umfassende Lösung, die eine leistungsstarke Datenkonvertierung anbietet. Das bedeutet weiterhin, dass jede beliebige Konvertierung unterstützt wird. Dazu gehört zum Beispiel die Konvertierung von⁵³

- Flat file zu XML
- XML zu CSV
- Jede beliebige Datenbank zu XML
- XML zu EDI
- XML zu EDI
- XML zu EDI
- HIPAA zu XML
- HL7 zu XML
- ... und jede weitere Kombination.

Adeptia ETL besteht aus insgesamt drei unterschiedlichen Komponenten, welche in Abbildung 8 grafisch illustriert werden.

⁴⁹ Vgl. Adeptia (o.J)

⁵⁰ Vgl. ebenda

⁵¹ Vgl. Solutions Review (2014)

⁵² Vgl. Adeptia (o.J)

⁵³ Vgl. ebenda



Abb. 8: Die drei Komponenten der Adeptia ETL⁵⁴

Die erste der drei Komponenten ist das webbasierte ‚Design Studio‘. Dieses ermöglicht mithilfe eines Wizards, Datenregeln so zu dokumentieren, dass sie einen Zusammenhang zwischen Überprüfungen, Überarbeitungen und Zuordnungen (auch Mapping genannt) herstellen. Des Weiteren enthält das ‚Design Studio‘ eine Bibliothek mit Funktionen, welche im Voraus erstellt werden können und so immer wieder erneut verwendet werden können. Der ‚Data Mapper‘ verfügt außerdem über die Fähigkeit eine Vorschau der tatsächlichen Quell- und Zieldaten anzuzeigen, während die Regeln spezifiziert werden. Dies ist allerdings nur möglich, solange die Datei mit den Quelldaten verfügbar ist.⁵⁵ Zusammengefasst ist das Design Studio eine Web-Browser-basierte Umgebung für IT-Mitarbeiter und Geschäftsanalysten, welche dazu genutzt werden kann, gemeinsam Datenflüsse zu entwerfen, zu konfigurieren und zu verwalten.

Die zweite Komponente der Adeptia ETL ist der sogenannte ‚Central Repository‘ (dt. Zentralspeicher).⁵⁶ Hier werden alle wiederverwendbaren Objekte wie Datenflüsse, Schemata, Zuordnungsregeln, Workflow-Aufgaben und Web-Services-Komponenten gespeichert, die mithilfe des ‚Design Studios‘ erstellt wurden.

Die dritte und damit letzte Komponente ist die ‚Run-time Execution Engine‘, also eine Laufzeitkomponente.⁵⁷ Hier werden die Regeln zum Mapping und die Datenflustransaktionen auf die eingehenden Daten und Nachrichten angewendet und ausgeführt. Diese dritte Komponente ist somit zuständig für das Einleiten, Verwalten, Steuern und Verfolgen der Ausführung der Datenflüsse.

⁵⁴ Vgl. Adeptia (o.J)

⁵⁵ Vgl. ebenda

⁵⁶ Vgl. ebenda

⁵⁷ Vgl. ebenda

ETL Suite - Vorgangweise

Die ETL Suite von Adeptia verfolgt bei seiner Vorgehensweise drei grundsätzliche Schritte:

- Data Mapping
- Data Transport
- Data Flows

Diese stimmen in ihren Grundzügen mit dem ETL-Prozess überein. Im Folgenden soll näher auf diese drei Schritte und deren Software-Komponenten eingegangen werden.

Data Mapping

Der ETL Prozess ist entscheidend für die Automatisierung von Prozessintegration. Der Data Mapping-Engine der Adeptia ETL Suite ermöglicht, wie bereits gesagt, jede beliebige Transformation zwischen verschiedenen Datenformaten. Dabei können Mapping-Regeln aus mehreren Quellen auf mehrere Ziele, unabhängig von ihrem Datenformat, angewendet werden. Daraus ergibt sich, dass Daten aus mehreren Quellen aggregiert werden können und der Inhalt automatisch an mehrere Ziele verteilt werden kann. Mithilfe der Simulationsfähigkeit können Abläufe außerdem verifiziert werden, bevor sie tatsächlich durchgeführt werden. Die folgenden zwei Komponenten unterstützen die Arbeit des Data Mappings:

- Schema Parser:

Ein Schema ist die Struktur eines Dateiformats und spezifiziert somit Informationen über die verschiedenen Datenfelder und Datensätze, welche in einer Nachricht oder Datei enthalten sind. Das Entwerfen eines Schemas ist also der Prozess der Bereitstellung von Metadaten-Informationen. Es ist eine Web-Schnittstelle, die es IT-Anwendern ermöglicht Datenformate zu gestalten und zu verwalten, sowie Metadaten für alle gängigen Dateiformate (z.B. Flat Files, EDI, XML, Microsoft Excel etc.) und relationale Datenbanken (z.B. Oracle, SQL Suite, IBM DB2, MySQL etc.) anzugeben. Abbildung 9 illustriert diese Vorgehensweise und zeigt die Benutzeroberfläche des ‚Schema Parser‘.⁵⁸

⁵⁸ Vgl. Adeptia (o.J.)

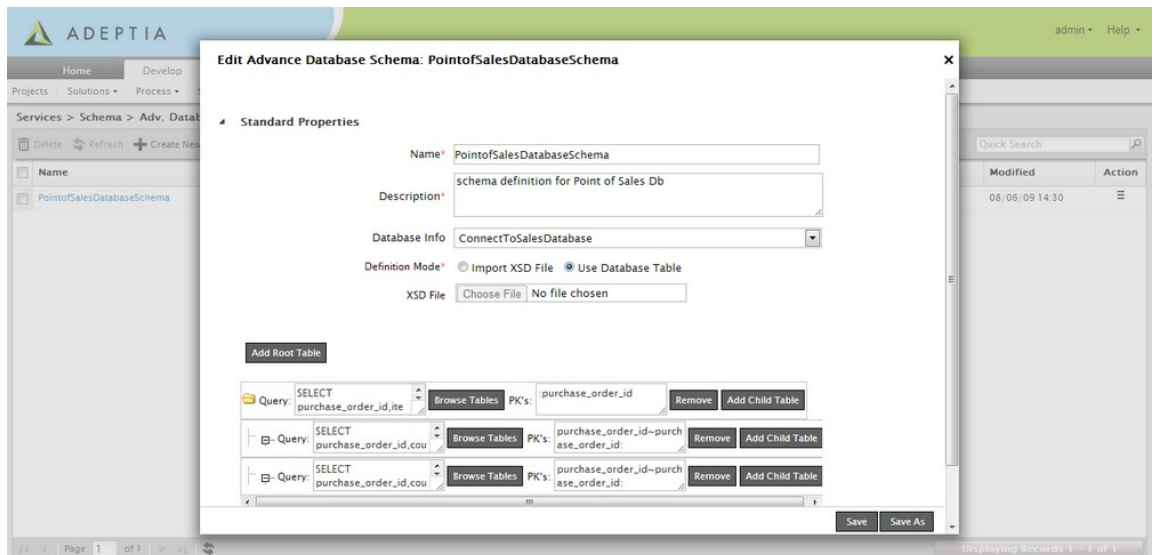


Abb. 9: Schema Parser⁵⁹

- Data Mapper:

Der Data Mapper ist eine webbasierte Schnittstelle für die Verwaltung von ETL-Regeln. Er vereinfacht die Verwaltung von Mapping-Regeln, die die Regeln...

- mithilfe einer visuellen, drag-and-drop Schnittstelle definieren.
- in einfachem Englisch angeben und nicht als Code.
- vom Code direkt in PDF dokumentieren.
- für spätere Bearbeitungen archivieren.
- direkt in einen ausführbaren Programmcode konvertieren.

Abbildung 10 zeigt einen Ausschnitt des 'Data Mappers'.

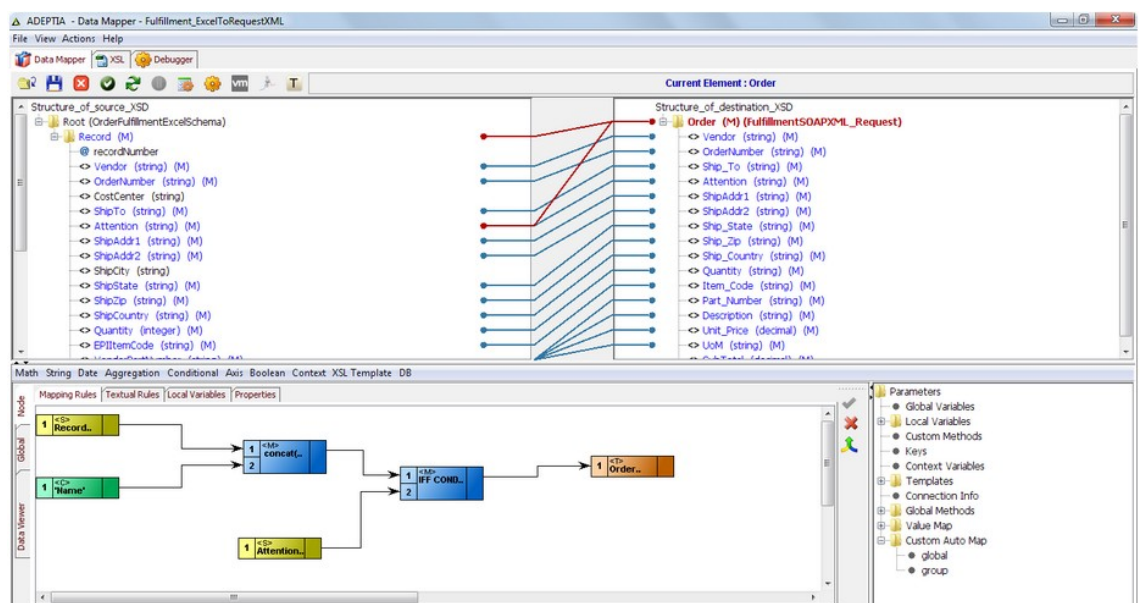


Abb. 10: Data Mapper⁶⁰

⁵⁹ Enthalten in: Adeptia (o.J.)

Data Transport

Die Adeptia ETL Suite bietet einen sicheren Zweiweg-Transport, welcher es ermöglicht verschiedene Quell- und Zielsysteme anzugeben. Dieser Datentransport kann mithilfe eines Wizards konfiguriert werden, ohne dass man eine einzige Zeile Code schreiben muss. Dadurch, und mithilfe der lose verbundenen Architektur der Software, kann man ebenfalls wiederverwendbare Services erstellen, die sowohl für das Empfangen als auch für das Senden einer Datennachricht gebraucht werden können. Unterstützte Transportprotokolle sind dabei HTTP/S, FTP, SMTP/POP3, JMS, JCA, JDBC und SOAP.

Data Flows

Die Adeptia ETL Suite enthält eine Komponente zur einfachen Erstellung von Datenflüssen. Diese ermöglicht es, ...

- Datenflüsse visuell zu gestalten, zu entwickeln und aufrechtzuerhalten, um bestehende Unternehmenssysteme und –prozesse zu automatisieren und zu verbessern.
- mithilfe einer einfach zu bedienenden Schnittstelle intuitiv und schnell Prozesse zu erstellen, Regeln zu dokumentieren und die Ausführung zu optimieren.
- die Integration mithilfe von einfacher Programmierung zu verbessern.

Die Adeptia ETL Suite ist sehr gut für jede Art von Benutzer geeignet, da man keine übermäßigen technischen Kenntnisse für die Verwendung des ETL-Tools benötigt. Daneben ist die Anwendung der Adeptia ETL Suite sehr intuitiv, das heißt man benötigt keine lange Einarbeitungsphase, sondern kann schon nach wenigen einführenden Klicks damit arbeiten. Auch die Mappings sind sehr benutzerfreundlich gestaltet, da die resultierende XML-Datei automatisch mit den generierten Daten getestet werden kann. Positiv anzumerken ist außerdem, dass die Produktivität des Tools mithilfe von minimalen Trainings schnell verbessert werden kann und zum Speichern der Objekte das XML-Format verwendet wird.⁶¹

Die Adeptia ETL Suite enttäuscht jedoch mit der Ausführungsumgebung, die sehr statisch aufgebaut ist und es nicht ermöglicht, die Datenflüsse zwischen den Aktivitäten mit zu verfolgen. Außerdem können keine dynamischen Metadaten verwendet werden und der DataMapper läuft nicht mit Flash 7.⁶²

⁶⁰ Enthalten in: Adeptia (o.J.)

⁶¹ Vgl. Gallardo, E. (2013)

⁶² Vgl. ebenda

4.3 Jitterbit



Abb. 11: Jitterbit Logo⁶³

Jitterbit ist ein Open Source Integrationssystem für Daten und Applikationen, das eine schnelle und einfache Art und Weise liefert, um eine Integrationslösung zu entwerfen, zu konfigurieren, zu testen und bereitzustellen.⁶⁴ Die Lösung zielt darauf ab, die Herausforderungen in Bezug auf Kosten und Komplexität zugehörig zu der Verbindung von Applikationen, Daten und Systemen zu überwinden. Die Jitterbit Open Source Integrationslösung ist eine grafische Plattform für den Anschluss von Legacy-, Unternehmens- und On-Demand-Anwendungen, einschließlich Business Process Fusion, ETL, SaaS und SOA.⁶⁵

Der Hauptsitz des Unternehmens Jitterbit befindet sich im kalifornischen Oakland, USA und wurde 2004 gegründet. Jitterbit hat außerdem einen Sitz in Washington, Florida, London, Indien und Malaysia. Das Unternehmen hat es sich zum Ziel gemacht, den heutigen modernen Unternehmen agile Integration anzubieten. Es gibt bereits über 35.000 Jitterbit Anwender in 125 Ländern mit einer 99-prozentigen Kundenzufriedenheit. Dazu zählen zum Beispiel Bayer, SunKist oder auch UNITED. Zu den 300 Partnern auf sechs Kontinenten gehören beispielsweise Microsoft, Amazon, SAP oder auch Salesforce.com.⁶⁶

Unabhängig vom offiziellen Erwerb der Open Source Software über Jitterbit oder einen Partner, kann sie auch über Sourceforge.net heruntergeladen werden. Von 1. Januar 2014 bis 1. Juli 2014 wurde das Jitterbit ETL-Tool bereits 971-mal von Sourceforge.net heruntergeladen. Der Großteil der Anwender (77%) benutzt das Betriebssystem Windows. Ein Drittel der gesamten Anwender stammen aus den USA. Die weiteren Benutzer stammen aus 70 anderen Ländern, worunter Deutschland auf Platz 2 der häufigsten Downloads mit 100 Downloads liegt. Seit Mitte 2013 haben keine Anwender eine persönliche Meinung hinterlassen. Das Forum der Webseite wurde zuletzt 2010 aktiv benutzt. Daher wurden seitdem keine Neuigkeiten, neue Dokumente, Informationen zu Fehlern oder Ähnlichem verzeichnet. Laut der Statistik nimmt die Anzahl der Downloads von Monat zu Monat ab.⁶⁷

Details zur Software

⁶³ Enthalten in: Jitterbit (2014b)

⁶⁴ Vgl. SourceForge (2014a)

⁶⁵ Vgl. Solutions Review (2014)

⁶⁶ Vgl. Jitterbit (2014a)

⁶⁷ Vgl. SourceForge (2014b)

Jitterbit verwendet auch gemeinsam nutzbare Integrationsvorlagen, genannt Jitterpak, die es der Community erlauben verbreitete Integrationen mit den beliebtesten Web-Services, Anwendungs-APIs und Datenbanken wiederzuverwenden. Jitterbit Version 5 bietet jedoch kein Jitterpak für IBM an, was einige Anwender trotzdem nicht davon abhält.⁶⁸ Ob in der Cloud oder im Haus erhältlich, Jitterbit erkennt Systemkonfigurationen automatisch und erlaubt nichttechnischen Anwendern durch das Zeigen und Klicken, Quell- und Zielsysteme zu definieren. Mithilfe von drag-and-drop können Datentransformationen abgebildet werden und Integrationsoperationen je nach Zeitplan ausgeführt werden.⁶⁹ 80 bis 90 Prozent der Jitterbit Installationen werden dafür verwendet, um Daten zwischen Front-Office-Anwendungen wie CRM und Back-Office-Anwendungen wie ERP zu bewegen.⁷⁰ Jitterbit unterstützt viele verschiedene Dokumententypen und Protokolle, wie XML, Web Services, Datenbanken, LDAP, Text, FTP, HTTP(S) usw.⁷¹

Wie in Abbildung 12 sichtbar, erfolgt die Datenintegration anhand von drei Schritten, welche laut Jitterbit maximal fünf Tage betragen.



Abb. 12: Jitterbit - 3 Schritte Durchführung⁷²

In Schritt 1 ‚Design‘ folgt die Erstellung des Designs für die Integration. Im Jitterbit Studio wird die Konfiguration lediglich durch Klicken statt durch Programmcode durchgeführt. Dabei kann eine schnelle Verbindung zu hunderten Applikationen aufgebaut werden. Ein kompletter Integrationslebenszyklus kann darin absolviert werden. Mithilfe des Jitterbit Automapper können Felder miteinander verbunden und abgebildet werden. Durch den Workflow Designer können hierbei Integrationsprozesse aufgebaut werden. Zusätzlich kann in dieser Komponente, mit der bereits vorhandenen Formelbibliothek, die Geschäftslogik einfach hinzugefügt werden.

Im zweiten Schritt ‚Deploy‘ erfolgt das Einsetzen der Integration in einer Cloud oder im Unternehmen. Dasselbe Design kann überall verwendet werden, ob in der Cloud, im Unternehmen oder in einer Hybridlösung. Um Echtzeitanforderungen zu erfüllen, lässt sich

⁶⁸ Vgl. Woodie, A. (2013)

⁶⁹ Vgl. Solutions Review (2014)

⁷⁰ Vgl. Woodie, A. (2013)

⁷¹ Vgl. SourceForge (2014a)

⁷² Enthalten in: Jitterbit (2014d)

mit einem elastischen Cloud Agent Netzwerk die Anwendung leicht skalieren. Das genutzte Smart Clustering führt zu einer hohen Performance und hohen Verfügbarkeit.

Der dritte Schritt ‚Manage‘ beinhaltet und beschreibt die Integration der Daten, welche standort- und gerätunabhängig durch ein Cloud Management ausgeführt werden kann. Es bleibt jedoch zu beachten, dass die Zugriffsrechte nur an teaminterne Teilnehmer weitergeleitet werden dürfen, um die Datensicherheit zu gewährleisten. Echtzeit Fehlermeldungen und ein vollständiger Auditprozess sind in die Software integriert und lassen sich einfach abrufen.⁷³

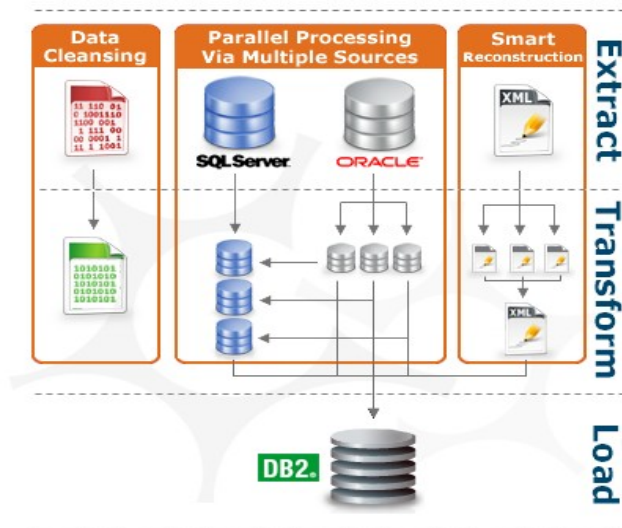


Abb. 13: ETL-Prozess in Jitterbit⁷⁴

Wie in Abbildung 13 aufgeführt, beinhaltet Jitterbit den klassischen ETL-Prozess als Integrationswerkzeug, welches jedoch nur eins von vielen darstellt.

4.4 Xineo XIL



Abb. 14: Xineo Logo⁷⁵

Xineo (XML Import Language) XIL beschreibt eine XML-Sprache zur Transformation verschiedener datensatzbasierter Quellen in XML Dokumente und bietet eine voll funktions-

⁷³ Vgl. Jitterbit (2014c)

⁷⁴ Enthalten in: Jitterbit (2014e)

⁷⁵ Enthalten in: Xineo (o. J.)

fähige XIL Verarbeitungsimplementierung.⁷⁶ Diese Implementierungskomponente beinhaltet eine Unterstützung für relationale und strukturierte Textquellen und ist erweiterbar durch seine öffentlichen (Application Programming Interface) APIs. Dadurch ist eine dynamische Integration von neuen Datenquellen-Implementierungen gewährleistet. Die Implementierung relationaler Textquellen wird durch JDBC ermöglicht und die der strukturierten Texte durch CSV. Es bietet auch eine Abstraktion durch das Ausgabeformat, so dass Xineo Stream oder DOM-Dokumente ausgibt. Das Document Object Model (DOM) ist eine Spezifikation einer Schnittstelle für den Zugriff auf HTML- oder XML Dokumente.⁷⁷

Wie bereits angedeutet, beinhalten XIL Dokumente zwei Arten von Elementen:

1. **XIL- definierte Elemente**, die von der XML Maschine interpretiert werden und die beschreiben wie Daten aus einer oder mehreren Quellen importiert werden.
2. **Andere Elemente** definieren Vorlagen, die instanziiert werden, um das generierte Dokument zu erzeugen.

Datenquellen definieren

Der erste Schritt beim Datenimport ist es die Quellen zu definieren. Datenquellen wurden so definiert, dass sie das ‚xil:source‘ Element benutzen. Jeder Datenquelle werden ein definierender Typ und ein benutzerdefinierter Name, der zukünftig verwendet werden kann, zugeordnet. Je nach Quellentyp kann jede Datenquelle eine gewisse Anzahl an Eigenschaften besitzen.

Das untere Beispiel in Abbildung 15 zeigt eine JDBC Datenquelle mit dem Namen ‚myDataSource‘⁷⁸. Es werden Eigenschaften verwendet, um Datenbank-Verbindungsparameter zu definieren:

```
<?xml version="1.0" ?>
<xil:xil xmlns:xil="http://www.xineo.net/XIL-1.0">
<xil:source type="sql" name="myDataSource">
<xil:property name="url"> jdbc:mysql://myServer/myBase </xil:property>
<xil:property name="userId"> myUserId </xil:property>
</xil:source>
[...]
```

Abb. 15: Definition von Datenquellen⁷⁹

⁷⁶ Vgl. Xineo.net(o.J.)

⁷⁷ Vgl. Xineo.net(o.J.)

⁷⁸ Vgl. ebenda

Verwendung von CSV Dateien

CSV Dateien sind strukturierte Textdateien, in denen jede Zeile eine Reihe von Feldern enthält. Diese werden in der Regel durch Kommas oder andere Zeichen, wie z.B. Spalten, getrennt. In den meisten Fällen sind diese Dateien kompatibel mit XIL, da sie mit einem ‚regex‘ Datenquellentyp behandelt werden.⁸⁰

Aber in komplexeren Fällen, sind reguläre Ausdrücke nicht ausreichend, zum Beispiel wenn man Abfragen aus CSV-Einträgen durchführen möchte. In diesen Fällen kann man den CSV-JDBC-Treiber verwenden. Dieser ist ein einfacher read-only JDBC-Treiber, der CSV-Dateien als Datenbanktabellen verwendet. Zusammenfassend gibt es zu Xineo XIL wenige Informationen über die Handhabung und über die Funktionen.

4.5 CloverETL



Abb. 16: CloverETL Logo⁸¹

CloverETL ist eine kommerzielle Open Source Software, die von der Firma Javalin entwickelt und unterstützt wird. Die 2002 von David Pavlis veröffentlichte Software beinhaltet seit dem mehr als 1 Million Zeilen Code. Das zum Großteil (70%) Java-basierte, nach J2EE Standard, ETL Framework wird zur systematischen Strukturierung und Integration von Datensätzen verwendet (siehe Abbildung 17). Als weitere Programmiersprachen werden zu 28% XML und zu 2% andere Sprachen verwendet. Durch die Anwendung von Java besitzt CloverETL den Vorteil, plattformunabhängig angewendet werden zu können. Unterschiedliche Betriebssysteme wie Linux, Windows, HP-UX, AIX, IBM As/400, Solaris oder Mac OS X können verwendet werden.⁸² CloverETL erlaubt es Daten in Applikationen, Datenbanken und Data Warehouses zu verteilen, zu transformieren sowie unterschiedliche Filter anzuwenden. Kunden von CloverETL sind unter anderem Oracle, IBM, Salesforce, GoodData, Comcast, Active Network und Sunny. Insgesamt nutzen mehr als

⁷⁹ Enthalten in: Xineo.net (o.J.)

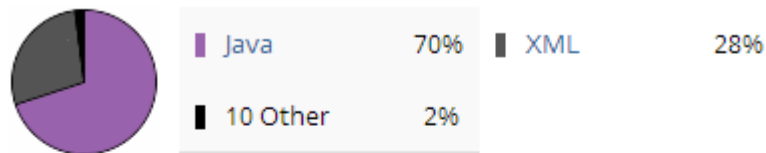
⁸⁰ Vgl. Xineo.net(o.J.):

⁸¹ Enthalten in: CloverETL (o.J.)

⁸² Vgl. Ohloh Black Duck Open HUB (2014)

2000 direkte Kunden die Software.⁸³ Im Jahr 2012 wurde neben dem aktuellen amerikanischen Markt auch nach West Europa expandiert. Die Software findet global Anwendung.⁸⁴

Languages



Lines of Code

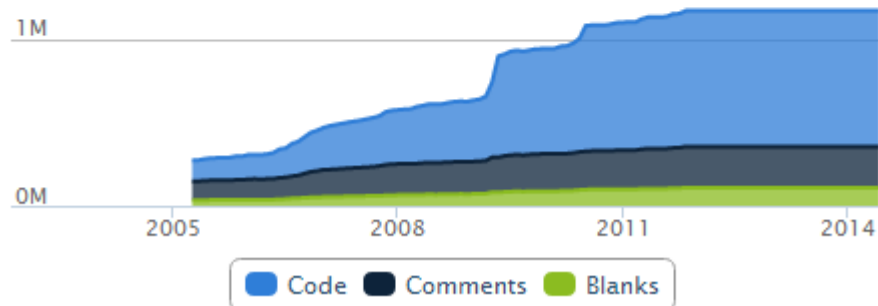


Abb. 17: CloverETL Sprachen und Code⁸⁵

Des Weiteren kann CloverETL als Einzellösung, d.h. als Server Applikation oder aber eingebettet als Java Library in anderen Applikationen, verwendet werden.⁸⁶ Ressourcen wie Verbindungen, Grafiken und Metadaten werden im XML Format gespeichert. Ein weiterer Vorteil ist die Skalierbarkeit und Ausbaufähigkeit der Software. So können Unternehmen CloverETL exakt nach ihren Wünschen und Anforderungen nutzen. Durch die Open Source Lizenz belaufen sich die Kosten für CloverETL auf null. Die folgende Abbildung 18 soll die genannten und einige weitere Funktionalitäten von CloverETL veranschaulichen.

Heutzutage arbeiten mehr als 60 Spezialisten und Experten an der Weiterentwicklung und Instandhaltung von CloverETL.⁸⁷

⁸³ Vgl. CloverETL (o.J.)

⁸⁴ Vgl. CloverETL (o.J.)

⁸⁵ Enthalten in: Ohloh Black Duck Open HUB (2014)

⁸⁶ Vgl. Ohloh Black Duck Open HUB (2014)

⁸⁷ Vgl. CloverETL (o.J.)

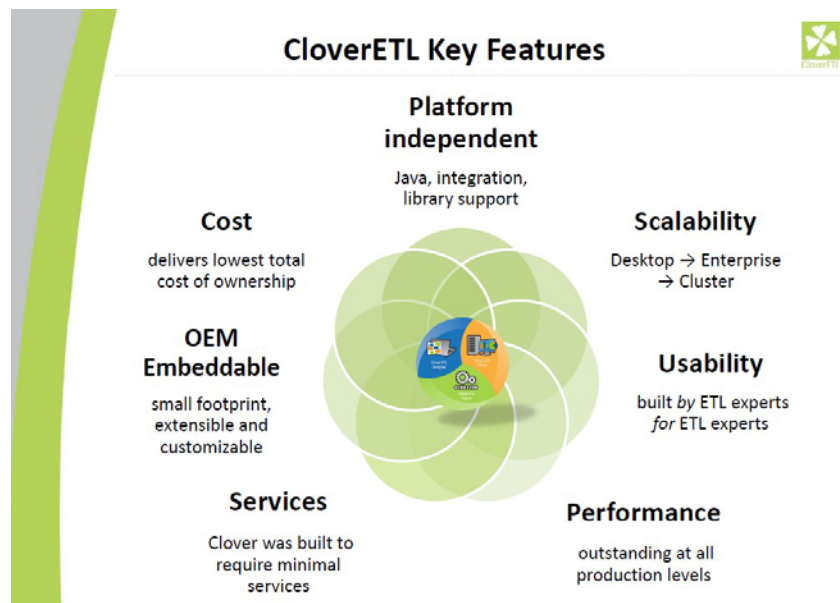


Abb. 18: CloverETL Features⁸⁸

Die CloverETL Product Suite besteht aus drei Teilen, dem CloverETL Server, dem Clover ETL Designer und der CloverETL Engine (siehe Abbildung 19).⁸⁹

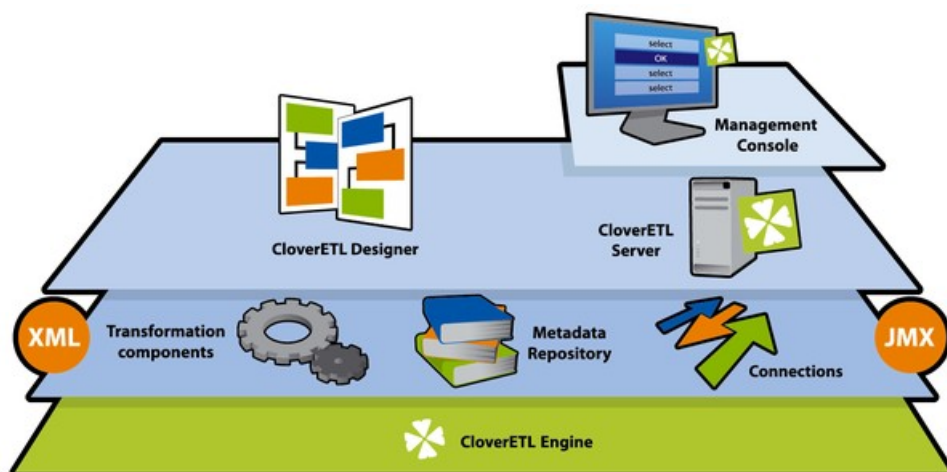


Abb. 19: CloverETL Product Suite⁹⁰

Der CloverETL Server stellt die Produktionsplattform dar. Hier können Webapplikationen wie Tomcat, WebSphere oder Glassfish für die Integration von Unternehmen genutzt werden. Der CloverETL Designer bietet die Möglichkeit die Software visuell zu designen, sowie eine Transformation-Entwicklungsumgebung mithilfe von Eclipse aufzubauen. Es besitzt eine sehr einfache, intuitive grafische Benutzeroberfläche. Durch einfaches drag-and-drop lassen sich Arbeitsschritte ausführen. Eine Components-Library, sowie eine Debug Funktion stehen zur Verfügung.

⁸⁸ Enthalten in: Ulrych, J. (2011)

⁸⁹ Vgl. o.A. (O.J)

⁹⁰ Enthalten in: CloverETL (o.J.)

Die CloverETL Engine ist das Herzstück der Software. Wie oben erwähnt, wird zum Großteil Java als Programmiersprache genutzt. Zudem ist die CloverETL Engine optimal für Original Equipment Manufacturer (OEM) – Zwecke geeignet.⁹¹ Folgende Abbildung 20 soll die Bestandteile von CloverETL genauer erläutern.

- Components
 - › Prebuilt algorithms
- Graph
 - › Processing algorithm in visual form
- Data Flow
 - › Edges between components
- Process
 - › Build > Connect > Configure
 - › Data processed as structured records with named fields
 - › Components operate on record fields

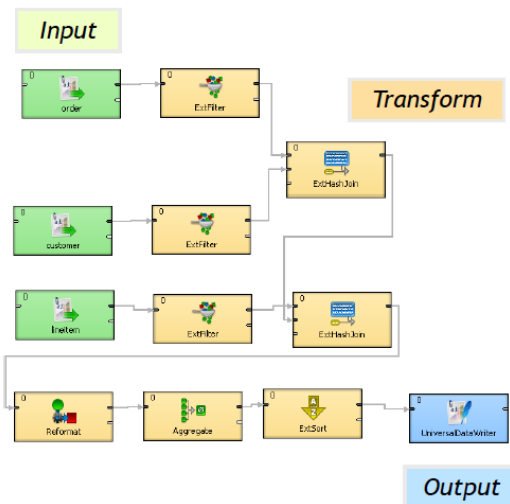


Abb. 20: Überblick CloverETL⁹²

Weitere Informationen:

Transformationsfunktionalitäten:⁹³

Lesen und schreiben	Text oder binäre Dateien, CSV, XML
	Archivieren via ZIP oder GZIP
	Remote Transfer über HTTP/FTP(S)
	Meldungen über JMS
Verbindung zur Datenbank	JDBC Verbindung, Möglichkeit zum Massensladen von Daten
	Unterstützt jeden SQL Standard
Transformatoren und Aggregatoren	Vielzahl an Datenmanipulationen, Sortier- Debugging, Joining Algorithmen, sowie Statistik und weiteren mathematischen Funktionen
	Anpassbar durch benutzerdefinierten Code
	Alternative Implementierungen zur höheren Effizienz

Tab. 1: Transformationsfunktionalitäten

⁹¹ Vgl. Ulrych, J. (2011)

⁹² Enthalten in: CloverETL (o.J.)

⁹³ Vgl. Ulrych, J. (2011)

Architektur:⁹⁴

CloverETL Designer ist eine unabhängige Applikation	Entwicklung und Ausführung ohne zentralen Server möglich
	Metadaten, sowie Transformationen werden lokal gespeichert
	Verfügbar für alle gängigen Plattformen (Linux, Windows, Mac)

Tab. 2: Architektur

Archivfunktion⁹⁵

XML Format Dateien und Verzeichnisse	Transformationen und Metadaten sind im leserlichen XML Format gespeichert
	Lesbar / verfügbar für Management Systeme, wie CVS, SVN, IBM Rational ClearCase, Git
	Integration mit Eclipse VMS Clients, z.B. Superclipse, EGit, ClearCase

Tab. 3: Archivfunktion

⁹⁴ Vgl. Ulrych, J. (2011)

⁹⁵ Vgl. ebenda

5 Kriterienkatalog und Gewichtung

(Sarah Ahmed, Muazzez Algac, Lara Hochhalter, Janine Jakob, Ines Schlenker)

Die Wichtigkeit und Bedeutung eines Kriterienkataloges für die Bewertung von Software wurde in Kapitel 3.1 beschrieben. Nun soll speziell für die Anforderungen des Auftraggebers ein Kriterienkatalog erstellt werden, um ETL Open Source Software zu bewerten. Um den höchsten Mehrwert für das Unternehmen erzielen zu können, wurde beschlossen, einen einheitlichen Kriterienkatalog in Absprache mit einem weiteren DHBW Projektteam zu erarbeiten. Dies ermöglicht es Synergieeffekte zu erzielen und die Ergebnisse beider Teams besser vergleichen zu können, um das optimale Open Source ETL-Tool zu empfehlen.

Die Grob- und Feinkriterien sind hierbei an die oben erläuterten ISO Standards zur Softwarequalität angelehnt, sowie an weitere eigene Ergänzungen aus Recherchen und in Absprache mit dem Vertreter des Auftraggebers. Innerhalb der Grobkriterien werden grundlegende Anforderungen überprüft. Übersteht die Software diese Auswahl, wird im weiteren Schritt eine Feinbewertung durchgeführt. Diese ist eingeteilt in Produkt- und Anbieterprofil der jeweiligen Software. Die am besten bewertete Software durchläuft schließlich den Prototyp-Test.

5.1 Grobkriterien

Die unten aufgeführten Grobkriterien wurden zur Vorauswahl der Software erarbeitet. Sie entsprechen den gegebenen Anforderungen des Kunden und ermöglichen einen groben Überblick über die Funktionalität und Anwendbarkeit der Software. Auf die Grobkriterien wird im Folgenden nicht näher eingegangen, da in dieser Arbeit der Feinkriterienkatalog und die damit verbundene Bewertung im Mittelpunkt stehen.

Nach Vorgaben des Unternehmens gibt es einige Kriterien, die als grundlegende Voraussetzung gelten. So sollte das Tool beispielsweise sowohl auf einer Windows- als auch auf einer Linux-Plattform lauffähig sein. Zudem sollten bestimmte Datenformate, wie CSV-Dateien, XML-Dateien, Excel-Dateien, relationale Datenbanken, hierarchische Datenbanken und großrechnerspezifische Dateiformate (z. B. VSAM-Dateien) unterstützt werden. Des Weiteren soll die Angemessenheit des Produktes gegeben sein, worunter die Eignung für die Nutzung und Ausführung spezieller Aufgaben verstanden wird.

Grobkriterium 1 – Dokumentation und Informationsvielfalt:

Die Verfügbarkeit von Informationen in Form von Dokumentationen, Web-Demos, Online-Tutorials, Wikis oder auch von Einträgen in Foren der Community ist ein entscheidender Faktor dafür, wie gut ein Anwender sich in eine neue Software einarbeiten kann oder wie schnell er bei Problemen eine Lösung findet.

Grobkriterium 2 – Funktionalitätsumfang:

Für den Funktionalitätsumfang wird zum einen analysiert, in wie weit die gebotenen Funktionalitäten den Anforderungen entsprechen und zum anderen auch, ob das Tool standardmäßig über zusätzliche Funktionalitäten verfügt oder bei Bedarf um diese erweitert werden kann.

Grobkriterium 3 – Downloadzahlen:

Eine Analyse der Downloadzahlen gibt Aufschluss über die Marktakzeptanz und deren Entwicklung. Sourceforge.net, ein Hosting-Provider für Open Source Projekte, bietet eine Übersicht über die Downloadzahlen zahlreicher Open Source Projekte⁹⁶ und dient im Rahmen dieser Arbeit als Informationsquelle für das Grobkriterium „Downloadzahlen“.

Grobkriterium 4 – Nennenswerte Partner:

Das Kriterium „Nennenswerte Partner“ beschreibt, ob ein Produkthanbieter in Kooperation mit einem großen Anbieter, beispielsweise von einem Softwareanbieter, steht. Diese Information gibt Aufschluss darüber, inwieweit aufgrund der Zusammenarbeit, eine Kompatibilität zwischen weiteren Software- oder Hardwareprodukten besteht.

Grobkriterium 5 – Ergonomie und Benutzerfreundlichkeit:

Die Ergonomie und Benutzerfreundlichkeit einer Software sind Messgrößen für die Schnittstelle zwischen Tool und Anwender und stehen dafür, mit welchem Einarbeitungsaufwand ein Tool genutzt werden kann, wie intuitiv die Bedienung ist und inwieweit dem Anwender Hilfestellung durch das Tool geboten wird.

Resultat

Die unten abgebildete Übersichtstabelle mit einzelnen Bewertungspunkten pro ETL Tool hilft die Auswahl der 2 besten Tools nachzuvollziehen.

⁹⁶ Vgl. SourceForge (2014c)

	Kettle	Adeptia	Jitterbit	Xineo XIL	CloverETL
<i>Grobkriterium 1 –</i> Dokumentation und Informationsvielfalt	5	4	4	1	4
<i>Grobkriterium 2 –</i> Funktionalitätsumfang:	5	3	3	3	5
<i>Grobkriterium 3 –</i> Downloadzahlen:	5	3	2	k.A.	3
<i>Grobkriterium 4 –</i> Nennenswerte Partner:	5	4	3	k.A.	5
<i>Grobkriterium 5 –</i> Ergonomie und Benut- zerfreundlichkeit:	4	4	4	2	4
Gesamtergebnis:	24	18	16	6	21

Tab. 4: Bewertung der Grobkriterien für die fünf ausgewählten ETL-Tools

Für den aktuellen Fall zeigt sich, dass die Open Source ETL-Lösungen CloverETL und Pentaho Data Integration (Kettle) das größtmögliche Potenzial haben, beim Auftraggeber eingesetzt werden zu können. Im weiteren Verlauf der Arbeit werden diese beiden Tools den Feinkriterienkatalog durchlaufen.

5.2 Feinkriterienkatalog

Der Feinkriterienkatalog beinhaltet Schritt zwei der Softwareauswahl. Die Gewichtung der unterschiedlichen Kategorie ist ausschlaggebend für die spätere Auswahl der Favoriten. Hierbei wurde eine Skala von 0 bis 5 ausgewählt:

0 sehr schlecht

1 gering

2 mittelmäßig

3 gut

4 sehr gut

5 außerordentlich gut

k.A. Keine Angabe möglich

Tab. 5: Bewertung der Feinkriterien

Der Kriterienkatalog, welcher in *Anhang 1* einzusehen ist, ist unterteilt in zwei Hauptkategorien, das Produkt- sowie das Anbieterprofil. Jede Kategorie ist wiederum unterteilt in Subkategorien, die im Folgenden beschrieben werden sollen.

5.2.1 Beschreibung Produktprofil

Die ausgewählten Kriterien im ersten Teil des Feinkriterienkatalogs sollen helfen das Produktprofil besser einzuschätzen und zu bewerten. Darunter fallen Kriterien wie Funktionalität, Leistung und Bedienbarkeit. Im Folgenden werden diese und weitere wichtige Kriterien erläutert.

Kategorie Ökosystem des Herstellers

Nach Bosch besteht „ein Software Ökosystem aus seiner Anzahl an Softwarelösungen, welche die Aktivitäten und Transaktionen der Akteure, des zugehörigen sozialen oder wirtschaftlichen Ökosystems und den Organisationen, welche die Software bereitstellen, ermöglicht, unterstützt und automatisiert.“⁹⁷

Software Ökosysteme entstehen dadurch, dass die heutigen Anforderungen an eine Software meist nicht mehr von einer Firma alleine getragen werden können. Unter ihnen wird demnach das Zusammenspiel zwischen Hersteller und weiteren Akteuren beschrieben. Hierunter fallen Drittanbieter, Partner und Stakeholder in Form von Kunden. Die Akteure innerhalb des Ökosystems agieren zwar unabhängig voneinander, stehen aber dennoch stets im Zusammenhang teilweise auch im Wettbewerb mit dem Hersteller.⁹⁸ Die Kategorie beschreibt also, in welcher Intensität alle Akteure zusammenarbeiten und inwieweit nennenswerte Partner in Kooperation mit dem Hersteller stehen.

Kategorie Funktionalität

Die Kategorie Funktionalität beschreibt die ganzheitliche Funktionsweise der Software. Hierbei wird die Möglichkeit der regelbasierten Transformation der Daten, die Richtigkeit der Aufgabenausführung, die Interoperabilität und die Modularität des Tools betrachtet.

Unter der *regelbasierten Transformation der Daten* versteht man, welche Funktionalitäten das Tool bereitstellt, um die Transformationsregeln zu definieren. Dies kann beispielsweise über eine grafische Lösung erfolgen oder durch Codierung mithilfe einer Programmiersprache wie beispielsweise Java. Die Richtigkeit des Tools beschreibt die Korrektheit der

⁹⁷ Bosch, Jan (2009) S. 111

⁹⁸ Vgl. Wolff-Marting, V. / Rothmann, P.(2011)

gelieferten Ergebnisse, beispielsweise das Ergebnis aus einer Zusammenführung von zwei unterschiedlichen Dateitypen. Im dritten Kriterium dieser Kategorie, der Interoperabilität, wird die Funktionalität beim Zusammenwirken der Software mit weiteren Schnittstellen und Systemen bewertet. Dazu gehören auch die Anbindung von Quell- und Zielsystemen sowie Quell- und Zieldaten.

Zuletzt spielt die Modularität der Softwarebestandteile eine entscheidende Rolle. Eine Software besitzt ein gewisses Maß an Modularität, wenn sie aus verschiedenen Modulen aufgebaut ist, die durch den Kunden beliebig zusammengestellt werden können.⁹⁹

Kategorie Leistung

Die Leistung einer Software wird durch die Komponenten Zuverlässigkeit, Verfügbarkeit und Skalierbarkeit beschrieben und bietet im Gegensatz zur Funktionalität nicht einen Überblick über die Bestandteile, sondern Informationen über die konkrete Ausführung.

Eines der wichtigsten Merkmale von Softwarequalität ist die Zuverlässigkeit einer Software. Sie muss ein bestimmtes Leistungsniveau zu vorgeschriebenen Bedingungen innerhalb eines bestimmten Zeitrahmens einhalten können. Zu diesem Kriterium gehört beispielsweise die Wahrscheinlichkeit für das Auftreten von unerwarteten Fehlern, welche sich folglich negativ auf die Zuverlässigkeit eines Systems auswirken.¹⁰⁰

Die Verfügbarkeit einer Software kann sich auf die gespeicherten Daten und Informationen sowie auf die Funktionalitäten beziehen. Sie spielt deshalb eine enorme Rolle, da bei einem Systemausfall oder einer Nichtverfügbarkeit von Daten ein hoher wirtschaftlicher Schaden entstehen kann. Ein weiterer wichtiger Indikator für die Leistung einer Software ist deren Skalierbarkeit. Sie beschreibt die Möglichkeit der Anpassungsfähigkeit an verschiedenen Umgebungen und Situationen, ohne größere Änderungen an der Software vornehmen zu müssen. Eine gut skalierbare Software kommt bei zehnfacher Leistung in etwa mit dem zehnfachen Ressourcenverbrauch (beispielsweise CPU oder Arbeitsspeicher) aus.¹⁰¹

Kategorie Bedienbarkeit / Ergonomie / Handhabung

Diese Kategorie bezieht sich auf die Schnittstelle zwischen ETL-Tool und Anwender und lässt sich durch die Kriterien Verständlichkeit/ Erlernbarkeit, grafische Benutzeroberfläche, Fehlertoleranz und Komplexität der Administration beschreiben.

⁹⁹ Vgl. Lackes, R. (o.J)

¹⁰⁰ Vgl. Siepermann, M. / Lackes, R. (o.J.)

¹⁰¹ Vgl. O.A. (o.J. b)

Das erste Kriterium, Verständlichkeit/ Erlernbarkeit, bedeutet, dass der Nutzer die Funktionen des Tools schnellstmöglich verstehen und nachvollziehen können sollte, um mit möglichst geringem Einarbeitungsaufwand das Tool nutzen zu können. Hierunter fällt auch die Frage nach der Installierbarkeit eines Tools. Durch Online-Hilfen, User-Guides, Tutorials oder Demonstrationsvideos kann dieses Kriterium positiv beeinflusst werden. Auch die grafische Benutzeroberfläche spielt für das Handling der Anwendung eine große Rolle, welche sich durch folgende Punkte auszeichnen sollte: systematische Anordnung der Oberfläche, intuitive Kategorisierung der Funktionen und Einhaltung üblicher Prozeduren. Zur Bedienbarkeit gehört auch die Fehlertoleranz des Tools. Vertippt sich beispielsweise der Benutzer bei einer Dateiendung oder ähnlichem, sollte das System entweder den Fehler aufzeigen oder ihn automatisch korrigieren, damit keine größeren Probleme daraus resultieren. Auch das Kriterium Komplexität der Administration ist Teil einer guten Handhabung. Es zeichnet sich dadurch aus, wie gut sich Daten, Projekte oder auch Zugriffsrechte innerhalb des Tools verwalten lassen.

Kategorie Performance / Effizienz

Die Performance eines Tools bedeutet im weitesten Sinne, wie gut ein Tool funktioniert und ob ein Tool als gut oder eher schlecht wahrgenommen wird. Dazu gehört die Tatsache, wie gut eine Anpassung an die Hardware vorgenommen werden kann, wie hoch die Geschwindigkeit und der Datendurchsatz sind, wie hoch der Ressourcenverbrauch bei der Anwendung des Tools ist und mit welchen Reaktionszeiten auf Benutzereingaben beantwortet wird. Unter der Anpassung an die Hardware versteht man, wie individuell diese durchgeführt werden kann. Dies kann beispielsweise über Konfigurations- oder Initialisierungsdateien des Tools vorgenommen werden, welche eine genaue Zuteilung der Hardware ermöglichen. Die Geschwindigkeit und der Datendurchsatz des ETL-Tools beschreiben, wie schnell Datensätze geladen oder transformiert werden, wie schnell Dateien generiert werden. Dabei ist der Datendurchsatz „eine Messgröße für die Leistungsfähigkeit eines Übertragungssystems.“¹⁰² Für dieses Kriterium kann aber auch die Zeit betrachtet werden, welche der Startvorgang des Tools in Anspruch nimmt.

Für die Performance spielt auch der Ressourcenverbrauch eine Rolle, da dieser im Regelfall möglichst gering gehalten werden soll. Dazu gehören beispielsweise der Speicherplatz, welcher durch das ETL-Tool belegt wird und die Beanspruchung der CPU oder des Arbeitsspeichers bei der Ausführung.

¹⁰² o. V. (2014), S. 1

Kategorie: Daten- und Zukunftssicherheit

Zur Kategorie Sicherheit gehören die Bereiche Datensicherheit mit den Kriterien Backup sowie Manipulierbarkeit und Zugriffsrechte und der Bereich Zukunftssicherheit mit dem Kriterium Releases. In Bezug auf die Datensicherheit ist die Sicherung in Form von Backups ein großes Thema. Die Datensicherung dient zur Sicherstellung von vorhandenen Datenbeständen im Falle eines Datenverlustes. Zudem umfasst die Datensicherung auch die Sicherung der Verfügbarkeit, Integrität und Konsistenz. Sicherheitsrelevante Fehler und Sicherheitslücken in einem Programm verringern dessen Sicherheit und somit auch dessen Stabilität. Diese können verschiedene Ursachen haben, wie beispielsweise die Programmkonzeption oder -konfiguration. Je sicherer ein Programm, desto geringer ist dessen Manipulierbarkeit. Hierfür ist es auch wichtig, Zugriffsrechte für das Tool zu vergeben um dadurch unerlaubte Zugriffe zu verhindern.¹⁰³

Für den Sicherheitsaspekt „Zukunftssicherheit“ sind insbesondere Releases von Bedeutung, da diese zeigen, dass eine Weiterentwicklung des Tools stattfindet. Von einem Release im Softwarebereich spricht man, wenn eine neue Version in den Markt eingeführt wird. Es gibt drei Arten von Releases: Major Release, Minor Release und Emergency Fix. Major Releases beinhalten viele neue Funktionen, die die vorherigen Probleme ganz aufheben. Bei den Minor Releases werden lediglich kleinere Erweiterungen hinzugefügt. Die letzte Art, der Emergency Fix, ist für die Behebung von kleinen Problemen zuständig. Releases sind unter anderem auch wichtig für Sicherheitsaktualisierungen und Wartungsaktualisierungen für schwerwiegende Fehler.¹⁰⁴

5.2.2 Beschreibung Anbieterprofil

In diesem Bereich wird das Anbieterprofil detailliert untersucht. Dies ermöglicht eine verlässliche Beschreibung der Kompetenzen und Referenzen der Anbieter. Zudem werden die Angebotsbreite und die Service-Möglichkeiten bewertet. Im Folgenden werden die einzelnen Anbieterprofil-Kriterien erklärt.

Kategorie Marktakzeptanz

Die *Marktakzeptanz* der ausgewählten Open Source Software stellt ein zentrales Auswahlkriterium dar, welches den allgemeinen Produktstatus innerhalb des fokussierten ETL-Softwarebereichs repräsentieren soll. Inhaltlich werden hierfür die Punkte ‚Google

¹⁰³ Vgl. Bundesamt für Sicherheit in der Informationstechnik (o.J.)

¹⁰⁴ Vgl. Schmid, O. (w.y.)

Trends‘, ‚Auszeichnungen/Zertifizierungen‘ und ‚Verbreitung (Entwickler, Anwender, Downloads)‘ untersucht und gewichtet.

Mithilfe von ‚Google Trends‘ soll geprüft werden, inwiefern das Open Source Tool von den Internetnutzern nachgefragt wird – basierend auf historischen Statistiken entsprechender Suchanfragen – und, in welchen Zeithorizont diese Nachfragen einzuordnen sind. Aus technischer Sicht zeigt Google einen Graph, welcher das Interesse am über Google gesuchten Begriff im zeitlichen Verlauf aufzeigt. Auch das regionale Interesse, anhand einer Weltkarte, sowie weitere verwandte Suchanfragen können eingesehen werden.¹⁰⁵

Unter dem Punkt ‚Auszeichnungen/Zertifizierung‘ wird in diesem Kontext die Preisvergabe bzw. die bestehenden Testergebnisse externer Prüfungen, wie beispielhaft jene von Stiftung Warentest oder eine mögliche TÜV-Zertifizierung verstanden. Die Vergabe der Auszeichnungen und Zertifizierungen werden entsprechend evaluiert, um einen Einblick in die durchgeführte Art der Untersuchung zu erhalten und diese als aussagekräftig deklarieren zu können.

Unter dem letzten Punkt dieses Auswahlkriteriums wird die ‚Verbreitung (Entwickler, Anwender, Downloads)‘ geprüft. Das Prüfergebnis wird hierfür einheitlich über die Quelle von ‚Sourceforge‘ bezogen und kann damit als Vergleichswert verwendet werden. Diese Seite ist im Entwicklerumfeld sehr stark fragmentiert. Benutzt wird sie sowohl für Downloads, welche aktiv als Beschaffung der Software für den praxisbezogenen Einsatz genutzt wird, als auch als Austausch von Entwicklern, welche den aktuellen Quellcode beziehen. Aufgrund dessen kann dieser Punkt als Repräsentation für den Fortbestand und die Weiterentwicklung des Produkts gesehen werden.

Kategorie Aktivität der Community

Unter dem Auswahlkriterium *Aktivität der Community* werden die Punkte ‚Forenaktivität‘, ‚direkte Ansprechpartner‘ und ‚FAQ‘ (Frequently Asked Questions) geprüft.

Unter der ‚Forenaktivität‘ wird die Möglichkeit der Softwarenutzer verstanden, sich bei Problemen gegenseitig austauschen und helfen zu können. Zentrale Basis bilden dafür eine oder mehrere entsprechende Austauschplattformen – zumeist Foren – welche den Problemsteller und die unterstützenden Helfer zusammenführen. Durch die Aktivität innerhalb der Foren lassen sich gute Rückschlüsse sowohl auf die Verbreitung als auch auf die aktive Nutzung der ETL-Software schließen.

¹⁰⁵ Vgl. Google Trends (2014)

Der Punkt ‚direkte Ansprechpartner‘ umschreibt den angebotenen Support des Anbieters für sein Produkt. Hier sind neben einer Supporthotline, Email-Ansprechpartnern o.ä. auch die Beschäftigung eigener Entwickler, mit dem zu erwartenden Know-how, von Vorteil. Dadurch lassen sich in extremen Fällen – aufgrund zeitlicher oder technischer Hintergründe – schnelle Lösungen finden.

Als ‚FAQ‘ werden die bestehenden Antworten auf die gängigsten Fragen zum Produkt verstanden.¹⁰⁶ Diese können eine schnelle Hilfestellung, oftmals vor allem bei der Installation und im Anfangsstadium, bieten. Innerhalb des angefertigten Kriterienkatalogs werden sowohl deren Quantität als auch die Qualität für die jeweiligen ETL-Tools untersucht.

Kategorie Dokumentation

Innerhalb des Auswahlkriteriums der *Dokumentation* wird untersucht, inwiefern eine umfangreiche, stringent strukturierte und qualitativ hochwertige Dokumentation vorhanden ist. Dies bezieht sich sowohl auf den Anwender- als auch auf den Entwicklerbereich.

Unter dem Punkt ‚Code Dokumentation‘ wird die Dokumentierung des Quellcodes untersucht. Im Rahmen dieser Arbeit kann hier allerdings keine tiefgreifende Untersuchung durchgeführt werden.

Bei der Überprüfung des ‚Benutzerhandbuchs‘ wird zunächst die generelle Verfügbarkeit dieser Art der Hilfestellung ermittelt. Besteht diese Form der Dokumentation, wird sie auf inhaltliche Faktoren untersucht – fokussierend auf Aufbau, konsequenter Strukturierung, Schwierigkeitsgrad und Umfang. Einen weiteren, eng verwandten Punkt stellt dabei die ‚Sprache der Dokumentation‘ dar. Diese beinhaltet die Bewertung für die angebotenen Sprachen der Dokumentation, mit Fokus auf Englisch und Deutsch.

Kategorie Dienstleistungsangebot

Das Auswahlkriterium des Dienstleistungsangebots umfasst die vom Anbieter selbst und von sogenannten Drittanbietern offerierten Unterstützungsdienstleistungen. Zentrale Rolle spielen das ‚Training‘ und die ‚Beratung‘ für das Open Source ETL-Tool. Ein Drittanbieter kann in diesem Kontext als „[...] ein Unternehmen, das Netzwerkprodukte oder Ersatzteile für Produkte und Produktfamilien anbietet, ohne dem Anbieter des Produkts („Erstanbieter“) anzugehören oder von ihm beauftragt zu sein („Zweitanbieter“).“¹⁰⁷ verstanden werden.

¹⁰⁶ Vgl. Duden (2013)

¹⁰⁷ Vgl. Enzyklo (o.J.)

Unter dem Punkt ‚Training‘ werden angebotene Kurse und Weiterbildungsmaßnahmen verstanden, um die eigenen Mitarbeiter auf einen möglichst effizienten Umgang mit dem letztlich eingesetzten Softwareprodukt zu schulen. Für die ‚Beratung‘ werden die angebotenen technischen Unterstützungen untersucht. Diese beinhalten individuelle Erweiterungen um neue Funktionalitäten oder Anpassungen an spezielle Gegebenheiten und Wünsche.

Kategorie Finanzierbarkeit

Das letzte Auswahlkriterium beinhaltet die allgemeine Finanzierbarkeit.

Der Punkt ‚Lizenz‘ beinhaltet die Bewertung der Lizenz, unter welcher die Open Source Software steht. Die Relevanz ergibt sich vor allem im Hinblick auf die dauerhaft anfallenden Lizenzierungskosten und die zur Verfügung stehenden Möglichkeiten, wie einer eigenen Anpassung.

Die ‚Version‘ untersucht zum einen die Vielfalt der angebotenen „Toolpakete“ mit deren unterschiedlichen Ausprägungen, in Richtung des Funktionsumfangs und der Servicegarantien. Zum anderen wird untersucht, inwiefern trotz der Fokussierung auf den Open Source Bereich eine direkte Vertragsvereinbarung – in Verbindung mit fixen Kosten – einzugehen ist, um den gewünschten Funktionsumfang auch tatsächlich zu erhalten.

5.3 Auswertung der ETL Anwendungen

5.3.1 Kettle – Auswertung des Feinkriterienkatalogs

Im Folgenden wird das ETL Tool Kettle anhand des Feinkriterienkatalogs ausgewertet.

Ökosystem des Herstellers

In der Kategorie „Ökosystem des Herstellers“ schneidet Kettle sehr gut ab. Viele Firmen, unter anderem, das Schönberg-solutions GmbH¹⁰⁸ bieten Beratung und Hilfeleistung für das Pentaho ETL Tool. Auch an Partnern und Schnittstellen mangelt es Pentaho nicht. Ein gutes Beispiel ist der Konnektor ProERPconn, ein SAP-Plugin, das eine SAP-Anbindung ermöglicht. Er ist ohne Probleme in die Umgebung implementierbar, jedoch ist es ein kostenpflichtiges Plugin.¹⁰⁹

¹⁰⁸ Vgl. Schönberg-solutions GmbH (2014)

¹⁰⁹ Vgl. IT-Novum GmbH (2009), S. 22

So ist es nicht wunderlich, dass Pentaho viele bekannte Kunden hat. Bereits im Kapitel 4.1 wurde erwähnt, dass Unternehmen wie die OTTO Group und ProSiebenSat1 Pentaho-Produkte einsetzen.

Funktionalität

Kettle schneidet, was seine Funktionalität betrifft, in den Tests gut bis sehr gut ab. Das liegt vor allem an der Interoperabilität und Modularität des Tools, da es eine quell offene, einfach zu bedienende, erweiterbare und leistungsfähige Software ist, die neben relationalen Datenbanken auch Excel- bzw. XML-Dateien oder NoSQL Datenbanken einbinden kann. Allerdings verfügt Kettle über keinen eigenen Scheduler – für eine regelbasierte Transformation der Daten muss ein Zusatzprogramm namens ‚Kitchen‘ verwendet werden.

Leistung

Nach erfolgreichem Testen von mehreren Datensätzen, kann man davon ausgehen, dass Kettle einen zuverlässigen Betrieb ermöglicht und damit die Anforderungen an Zuverlässigkeit erfüllt. Auch die Verfügbarkeit ist zufriedenstellend, da die gespeicherten Daten und Informationen ohne Probleme zugänglich waren. Zudem bietet Kettle ein paralleles Verarbeitungsmodell für enorme Leistung und Skalierbarkeit. Somit kommt das Tool mit der gegebenen Leistung und dem entsprechenden Ressourcen sehr gut aus.

Bedienbarkeit

Auch was die Bedienbarkeit und Handhabung des Tools betrifft, kann Kettle mit seinen Eigenschaften auftrumpfen. Das Tool ist sehr intuitiv gestaltet und lässt sich schon nach wenigen einführenden Klicks problemlos bedienen. Auch die grafische Benutzeroberfläche Spoon lässt sich intuitiv erlernen und auch von Benutzern mit wenig technischem Know-how gut verwenden. Die Fehlersuche gestaltet sich recht komfortabel, wobei Kettle es auch ermöglicht große Datenmengen zu untersuchen.

Performance/ Effizienz

Auch bei Performance überzeugt Kettle mit der Hardware-unabhängigen Eigenschaft. Man braucht lediglich das Java Runtime Environment (JRE) oder das Java Development Kit (JDK). Bei Erstellen des Prototyps waren die Geschwindigkeit und der Datendurchsatz sehr zufriedenstellend. Die Reaktionszeit auf die Benutzereingaben war recht kurz.

Datensicherheit/ Zukunftssicherheit

Backups lassen sich mithilfe des Repository einfach machen und geben dem Anwender so die gewünschte Sicherheit.

Das ETL Tool Kettle bietet eine großartige Verwaltung von Sicherheitsberechtigungen für Benutzer und Rollen. Sogar eine Integration mit bestehenden Sicherheitsdefinitionen in Lightweight Directory Access Protocol (LDAP) und Active Directory sind gegeben. Auch einzelne Berechtigungen können festgelegt werden wie z.B. für die Steuerung von Anwenderaktionen: Lesen, Ausführen oder Erstellen.

Marktakzeptanz

Kettle genießt eine große Marktakzeptanz. Dies wird auch die Informationen von ‚Google Trends‘ ersichtlich. Seit 2009 hat sich das Interesse an Kettle verdreifacht. Dabei gehört Deutschland zu den Top 3 Regionen, die online nach Pentaho Kettle gesucht haben.

Unter dem Punkt ‚Auszeichnungen/Zertifizierung‘ glänzt Pentaho Business Analytics mit fünf „Open Source Software Award“. ¹¹⁰

Wie auch schon im Kapitel 4.1. erwähnt, arbeiten mehr als 30.000 Mitglieder an der Weiterentwicklung von Pentaho. Diese Zahl zeigt das große Entwicklerumfeld für Pentaho und seine Komponenten an. Die Entwicklerzahl ist ein guter Anhaltspunkt um den Fortbestand und die Weiterentwicklung des Produkts zu bewerten.

Aktivität der Community

Kettle verfügt über einen großen Community Support, das zeigt allein schon, dass Pentaho ein eigenes Forum unterstützt, wo sich Benutzer austauschen und gegenseitig beraten können. Es stehen auch direkte Ansprechpartner aus der Organisation zur Verfügung, wobei man auf eine Antwort mehrere Tage warten muss. Trotzdem kann man die Kategorie ‚Aktivität der Community‘ in Hinsicht auf Kettle als sehr gut beschreiben, da man viele und auch sehr detaillierte FAQs zu so gut wie jedem Thema finden kann – so zum Beispiel einen ‚Beginner’s FAQ‘ in dem die Installation und die ersten Schritte genau beschrieben werden.

Dokumentation

Sowohl das Benutzerhandbuch als auch die Dokumentation des Quellcodes ist sehr benutzerfreundlich und hilfreich. Das Benutzerhandbuch / die Dokumentation gibt es auf folgenden Sprachen: Koreanisch, Spanisch, Russisch und Englisch. Die Dokumentation

¹¹⁰ Vgl. Pentaho Corporation (2014)

ist gut strukturiert mit viel weiteren Referenzen wie Büchern und Links. Leider wird die Dokumentation nicht auf Deutsch angeboten.

Dienstleistungsangebot

Der ETL-Anbieter Pentaho bietet mehrere online Trainings für Kettle an. Diese können je nach Bedarf auch vor Ort zur Verfügung gestellt werden.

Finanzierbarkeit

Kettle unterliegt der GNU LGPL-Lizenz welches im Kapitel 2.4 erläutert wurde. Diese Lizenz ist dafür bekannt, dass es die Verbreitung von freier Software fördert und die rechtlichen Rahmenbedingungen dafür schafft.¹¹¹

Im Kapitel 4.1 wurde bereits erwähnt, dass die Pentaho Suite in zwei Editionen angeboten wird. Die Community Edition beinhaltet weniger Support und Wartung als die Enterprise Edition, da die EE kostenpflichtig ist. Jedoch genügen die Basis Tools im Community Edition, da diese als Basis Kettle beinhaltet und den Zugriff über JDBC, Open Database Connectivity (ODBC) oder JNDI ermöglicht.

5.3.2 CloverETL- Auswertung des Feinkriterienkatalogs

Im Folgenden wird das ETL Tool CloverETL anhand des Feinkriterienkatalogs ausgewertet.

Ökosystem des Herstellers

In der Kategorie „Ökosystem des Herstellers“ schneidet CloverETL, ähnlich wie Kettle, gut ab. Der Hersteller weist nennenswerte Partner und Kunden wie Salesforce, Oracle oder IBM auf.

Funktionalität

Ähnlich wie Kettle, bietet CloverETL ein hohes Maß an Funktionalität. Das liegt vor allem an den regelmäßigen Transformationen, durch die eine hohe Anzahl an Datenmanipulationsalgorithmen möglich ist. Diese sind auch anpassbar an einen vom Anwender definierten Code. Durch die hohe Transformationsperformance erhöht sich auch die Skalierbarkeit.

¹¹¹ Vgl. Haneke, U. et al. (2010), S.178

Leistung

Die hohe Leistung von CloverETL ermöglichte das Testen von mehreren Datensätzen ohne weitere Probleme. Die Verfügbarkeit betreffend kann CloverETL durchaus mit Kettle mithalten, da die gespeicherten Daten und Informationen ohne Probleme zugänglich waren. Die Verfügbarkeit und Zuverlässigkeit werden durch Redundanzen und Cluster ermöglicht.

Bedienbarkeit

CloverETL ist ein benutzerfreundliches Tool. Es ist sehr intuitiv gestaltet und lässt sich durch ‚drag-and-drop‘-Optionen problemlos bedienen. Die Komplexität des Tools ist relativ gering.

Performance / Effizienz

CloverETL überzeugt durch seine Plattformunabhängigkeit mithilfe von JAVA. Die Verarbeitungsgeschwindigkeit war zufriedenstellend, aber nicht überragend. Der Ressourcenverbrauch wurde gering gehalten. Die Reaktionszeit der Benutzereingaben war zufriedenstellend.

Datensicherheit / Zukunftssicherheit

Was die Datensicherheit betrifft, gibt es keine Angaben zum Backup. Es ist aber möglich Daten zu exportieren. Daneben bietet CloverETL Nutzern die Möglichkeit einer Data Map und einer schnellen Datentransformation, die auch die Analyse, Validierung, Bearbeitung und Formatierung von Daten enthält. Diese visuell intuitive Data Map ermöglicht eine einfache Manipulation von Daten und einer Interaktion mit ihnen.

Marktakzeptanz

Zur Marktakzeptanz fehlen teilweise schlagkräftige Informationen, z.B. zu Zertifizierungen. Google Trends gibt an, dass das Suchvolumen zu gering sei, um Ergebnisse anzeigen zu können. SourceForge.net weist innerhalb von einer Woche 110 Downloads von CloverETL auf, darunter sind besonders viele in Nordamerika.¹¹²

Aktivität der Community

Die Software sticht besonders positiv in der Kategorie ‚Aktivität der Community‘ hervor. Ein eigenes Forum über CloverETL erhält ständige Pflege und ist viel verwendet. Die FAQs sind ausführlich und übersichtlich.¹¹³

¹¹² Vgl. SourceForge (2014a)

¹¹³ Vgl. CloverETL (o.J.)

Dokumentation

Die Kategorie ‚Dokumentation‘ erhält ebenfalls eine sehr gute Bewertung. Im Vergleich zu einem sehr ausführlichen und gut strukturierten Benutzerhandbuch, einem Quick Start Guide mit Bildern oder auch den White Paper, könnte die Dokumentation zum Source Code mehr Inhalt benötigen. Alle Dokumentationen sind auf Englisch verfügbar.

Dienstleistungsangebot

Das Dienstleistungsangebot von Seiten der Anbieter fällt gut aus. Der Kunde kann zwischen einem persönlichen Standardtraining oder einem erweiterten Training wählen.¹¹⁴ Das Angebot von Drittanbietern im Bereich Training ist rar, doch zum Thema Beratung waren keine Informationen auffindbar.

Finanzierbarkeit

CloverETL bietet die Lizenz LGPL an, weshalb die Open Source Software in der Kategorie Finanzierbarkeit exzellent abschneidet.¹¹⁵ Dank verschiedener Versionen wird dem Anwender viel Entscheidungsfreiheit gegeben.

Nachdem alle Punkte des Feinkriterienkatalogs sorgfältig ausgewertet wurden, ist es ersichtlich, dass das Open Source ETL-Tool Kettle in mehreren Kategorien besser abschneidet als CloverETL. Daraus ergibt sich, dass für das Testen mithilfe des Prototyps das ETL-Tool Kettle verwendet und analysiert wird.

Der Vergleich beider Tools mit der jeweiligen Bewertung kann in *Anhang 1* nachgesehen werden. Die Excel Tabelle bietet eine übersichtliche Bewertung der Feinkriterien zwischen Kettle und CloverETL.

¹¹⁴ Vgl. CloverETL (o.J.)

¹¹⁵ Vgl. Ebenda

6 Prototypische Umsetzung mit Kettle

(Sarah Ahmed, Muazzez Algac, Lara Hochhalter, Janine Jakob, Ines Schlenker)

Die prototypische Umsetzung ermöglicht die Bewertung des Open Source Tools Kettle in der Praxis. Im Folgenden wird der Ablaufprozess von der Erstellung einer Datenbank bis hin zur Erzeugung eines Output Files prototypisch aufgezeigt. Dazu wird ein Konzept für die Umsetzung eines Prototyps beschrieben. Es soll auf die einzelnen Schritte näher eingegangen, damit der Leser dies gegebenenfalls einfach nachmachen kann. Da der Auftraggeber von einem einheitlichen Prototyp der zwei DHBW Projektgruppen profitieren wird, wurde der Prototyp und die Datenbasis in Zusammenarbeit mit der zweiten DHBW Projektgruppe erstellt. Das erleichtert dem Auftraggeber den Vergleich und die eigene Bewertung der getesteten ETL-Tools der zwei Projektgruppen, anstatt wenn bereits zu Beginn gravierende Unterschiede in den Grundlagen vorhanden sind.

6.1 Konzept zur Umsetzung des Prototyps

Als Grundlage und Anhaltspunkt für die spätere prototypische Umsetzung wird ein Konzept zur Umsetzung des Prototyps benötigt, bis der anschließende Transformationsprozess von Kettle durchgeführt werden kann. Hierbei soll eine Reduktion der Komplexität durch das Aufteilen des Gesamtprozesses in einzelne aufeinander aufbauende Schritte erreicht werden. Das Konzept befasst sich mit dem kompletten Ablaufprozess von der Erstellung einer Datenbank und der Datenbasis, der Verbindung mit Kettle und der Überführung der Daten in ein beliebiges Format. Hierbei werden die vielseitigen Möglichkeiten, die mit dem Tool umgesetzt werden können, aufgezeigt.

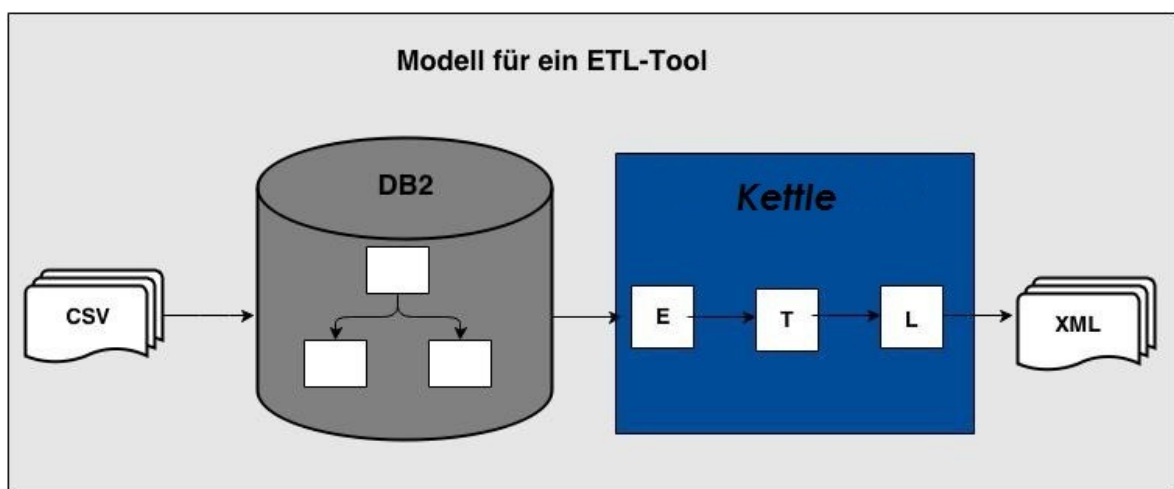


Abb. 21: Konzept zur Prototypen Umsetzung

Das grundsätzliche Konzept zur Erstellung des Prototyps ist in der Abbildung 21 dargestellt. Die Abbildung dient als Gesamtübersicht über das Konzept und die Einzelschritte. In den darauffolgenden Kapiteln wird immer wieder darauf Bezug genommen. Der Ablaufprozess kann in drei verschiedene Schritte mit unterschiedlichen Aufgaben unterteilt werden:

- **Der erste Schritt** befasst sich mit der **Erstellung einer großen Datenbasis**, damit die Situation in dem Unternehmen simuliert werden kann. Der große Datenbestand ermöglicht das Testen des Tools Kettle. Dabei können beliebige Dateien wie bspw. CSV, XML oder JSON erzeugt werden.
- **Der zweite Schritt** befasst sich mit der **Einrichtung der Datenbank** und der anschließenden **Verbindung zu Kettle**. Es stehen verschiedene Datenbanksysteme zur Auswahl relationale DB (DB2 von IBM) und hierarchische DB (DL1). In dieser Arbeit liegt der Fokus auf den relationalen Datenbanken. Anschließend müssen hier die automatisch generierten Tabellen und Daten aus Schritt eins eingebunden werden.
- **Der dritte Schritt** befasst sich mit dem gesamten **ETL-Prozess mit Kettle**, welcher zum **Testen von Kettle** führt. Er befasst sich mit der Datenintegration in Kettle, der Transformation der ausgewählten Daten und der Erstellung eines beliebigen Ausgabeformats und Überführung der Daten in das Zielsystem. In Abbildung 21 wird als Ausgabeformat eine XML-Datei erstellt. Nach dem Prozess Extract-Transform-Load kann jedoch jedes beliebige Dateiformat erzeugt werden.

Die spätere prototypische Umsetzung wird diesen Ablaufprozess im Detail beschreiben, damit eine abschließende Bewertung des Tools Kettle durchgeführt werden kann.

6.2 Erstellung einer großen Datenbasis

Dieses Kapitel stützt sich auf den ersten Schritt der Abbildung 21. Die Struktur und der Inhalt der Daten sind von großer Bedeutung, um die verschiedenen Daten mit dem ETL-Tool zu transformieren.

In dieser Arbeit wird der Fokus auf das Eingabeformat CSV gelegt. Dieses Format dient der Speicherung bzw. dem Austausch von einfach strukturierten Daten. Das CSV-Format wird dabei oft von Unternehmen eingesetzt, um Daten zwischen unterschiedlichen Computersystemen auszutauschen. Ein Merkmal dieses Formates ist, dass die Werte durch Kommas/Semikolons voneinander getrennt sind. Ein kleiner Ausschnitt aus einer der erzeugten CSV Dateien ist in Abbildung 22 dargestellt.

	A	B	C	D	E	F
1	KundenNr	Vorname	Nachname	Geburtsdatum	VersicherungsNr	AdressNr
2	592639	Shirley	Cole	15.03.1926	264595	742988
3	781474	Laura	Marshall	03.02.1973	562861	125121
4	628536	Jeremy	Burke	03.12.1927	610399	442609
5	793109	Catherine	Torres	28.08.1972	476458	820034

Abb. 22: CSV Beispieldatei

Für eine bessere Lesbarkeit wurde die Datei mit Microsoft Excel geöffnet. Die verwendeten Daten simulieren Versicherungsdaten, die von Unternehmen eingesetzt werden können. Die Abbildung zeigt eine Tabelle mit unterschiedlichen Spalten und den dazugehörigen Werten. Die automatisch generierte CSV-Datei beinhaltet insgesamt 1000 unterschiedliche Datensätze.

6.3 Datenbank einrichten und Verbindungsaufbau zu Kettle

Um die Daten in einer Datenbank zu speichern und diese zugänglich zu machen, muss eine Datenbank erstellt werden, bevor diese mit Kettle verbunden wird. In dieser Arbeit liegt dabei der Fokus auf relationalen Datenbanken, u.a. da diese weite Verwendung bei der HALLSCHE KVAG finden. Die Auswahl einer Datenbank fiel dafür auf die Open Source Software MySQL, da sie von dem Anwender wenige Vorkenntnisse fordert und trotzdem sehr leistungsstark ist. In MySQL lassen sich Daten erstellen, bearbeiten und verwalten.

Nachdem die Art der Datenbank festgelegt wurde, muss diese nun erstellt und mit Daten gefüllt werden. MySQL bietet die einfache Möglichkeit mittels der Entity Relationship (ER) Modellierung die Datenbank mit unterschiedlichen Tabellen generieren zu lassen.

Das in Abbildung 23 gezeigte ER-Modell zeigt die zu generierende Tabellen *Kundendaten*, *Versicherungsdaten* und *Kontaktdaten*. Die Tabelle *Kundendaten* ist die Oberklasse und beinhaltet neben dem Primärschlüssel *Kundennummer* die Fremdschlüssel *VNr* und *AdressNr*. Die Struktur der Tabellen richtet sich nach den einzulesenden CSV-Dateien. In Abbildung 23 ist beispielhaft ein Ausschnitt aus der Tabelle *Kundendaten* dargestellt. Anhand dieser Struktur wurde das ER-Modell erstellt.

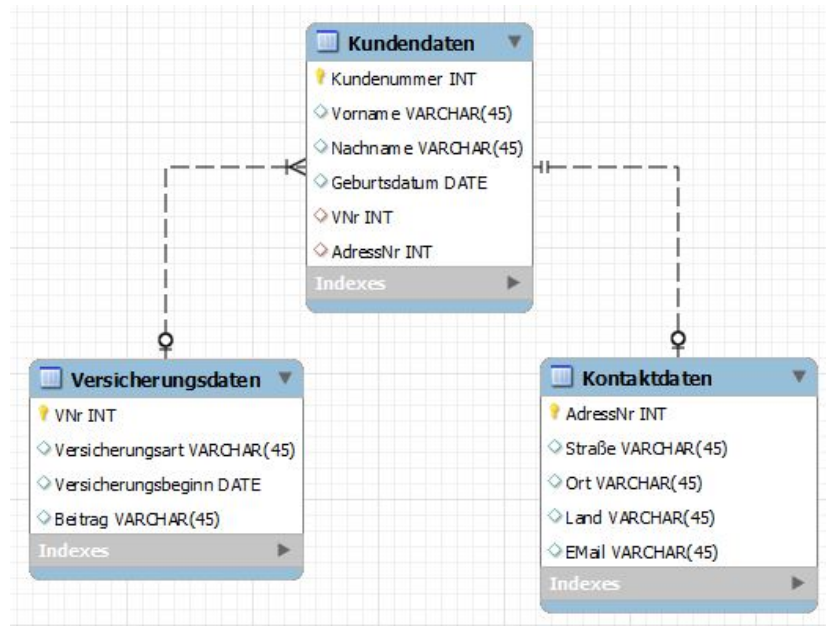


Abb. 23: ER Modellierung der unterschiedlichen Datenbanktabellen

Mithilfe des ER-Modell kann die Datenbank mit den gezeigten Tabellen automatisch erzeugt werden. Die Befüllung der Tabellen mit Datensätzen erfolgt über die SQL Anweisung:

```
LOAD DATA LOCAL INFILE '/work/KundenDB.csv'
INTO TABLE kundendaten
FIELDS TERMINATED BY ';'
LINES TERMINATED BY '\r\n'
IGNORE 1 LINES
(Kundennummer, Vorname, Nachname, Geburtsdatum, VNr, AdressNr);
```

Mit dieser Anweisung werden die Daten aus Abbildung 23 in die Tabelle *Kundendaten* sequentiell geschrieben. Jedem Attribut werden die Werte aus der CSV-Datei *KundenDB.csv* zugewiesen. Insgesamt werden jeweils 1000 Datensätze in die drei Tabellen geschrieben.

Um Daten aus der Datenbank zu beziehen oder in diese schreiben zu können, muss eine Verbindung zwischen der Anwendung Kettle und der MySQL Datenbank bestehen. Abbildung 24 zeigt wie die Verbindung zwischen einer Datenbank und Kettle aufgebaut werden kann und der grafischen Benutzeroberfläche Spoon. Im linken Abschnitt der Anwendung kann die neue Datenbankverbindung über *View>Transformations>Database connections*, danach mit einem Rechtsklick auf *New* und dem anschließenden Ausfüllen des Pop-up Fensters erstellt werden. Die erfolgreiche Verbindung zwischen Kettle und dem MySQL Server kann über den *Test* Button im Pop-up Fenster vorab getestet werden. Bei einem Misserfolg folgt eine Fehlermeldung mit der aufgelisteten Fehlerursache als Hilfestellung

für den Anwender. Nachdem Kettle mit der MySQL Datenbank verbunden wurde, ist ein Zugriff auf die einzelnen Tabellen der Datenbank möglich. Wie die folgende Abbildung zeigt, bietet Kettle eine lange Liste an unterschiedlichen Typen von Quellsystemen, mit welchen eine Verbindung aufgebaut werden kann, z.B. auch mit IBM DB2, Vertica oder Oracle anstatt nur mit MySQL.

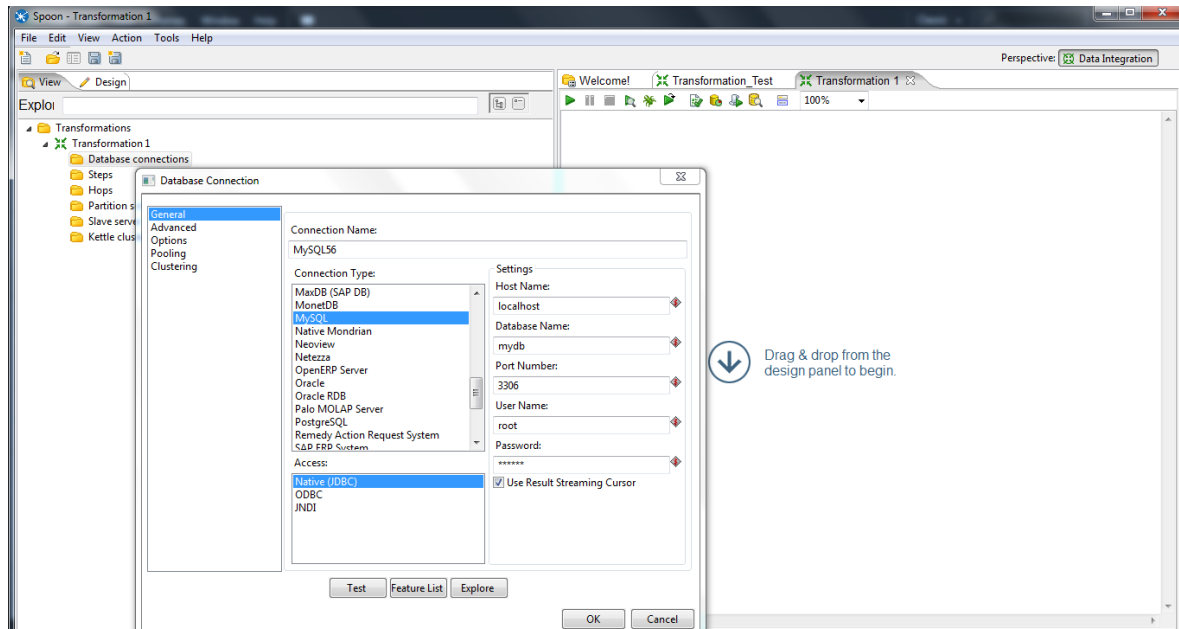


Abb. 24: Verbindungsaufbau zwischen MySQL und Kettle

6.4 ETL-Prozess mit Kettle

Die eigentliche Erstellung der Datenintegration oder sonstiger High Level Aufgaben erfolgt mithilfe der Anlegung eines neuen Jobs in der grafischen Oberfläche Spoon. Ein Job ist eine High Level Komponente, u.a. für die Koordination und Kontrolle des Datenflusses, die Ausführung von Transformationen, dem Senden von E-Mails bei fehlgeschlagenen Ausführungen usw. Die Ausführung der einzelnen Schritte innerhalb des Jobs folgt in der kreierten Reihenfolge. Durch einen Job werden eine oder mehrere Komponenten miteinander verbunden. Der entwickelte Ablaufprozess wird automatisch in Code übersetzt und zur Ausführung gebracht, denn der Benutzer erstellt die Ausführungsschritte lediglich über drag-and-drop im Design Panel.

Anstatt zusätzlich einen Job anzulegen, wurde in dem aufgeführten Beispiel gleich mit dem Anlegen einer Transformation gearbeitet. Für größere und komplexere Ausführungen des ETL-Prozesses empfiehlt sich dennoch eine vorherige Erstellung eines Jobs. Die Erstellung einer Transformation geht bereits tiefer in die Materie. Sie beschäftigt sich konkret mit der Bewegung der Zeilen von Daten zwischen den Quell- und Zielsystemen und der Transformation dieser. Ein weiterer entscheidender Unterschied zu einem Job ist,

dass die ausgewählten Teilschritte alle parallel statt nacheinander ausgeführt werden. Gleich bleibt, dass alle Ablaufprozesse über drag-and-drop der einzelnen Komponenten erstellt werden.

Der Prototyp hat in dem hier aufgeführten Anwendungsbeispiel aus Abbildung 25 das Ziel, die Daten aus der MySQL Tabelle *Kundendaten* und den CSV-Dateien *Kontaktdaten* und *Versicherungsdaten* in die Tabelle *Kunden_all* in der MySQL Datenbank zurück zu schreiben und alle integrierten Daten in eine XML-Datei zu kombinieren. Die aufgeführte Transformation in Abbildung 25 zeigt lediglich einen möglichen Ablaufprozess mit den einzelnen Schritten Extraktion, Transformation und Laden der Daten bzw. Generierung einer XML-Datei, wie es in Kettle gemacht werden könnte.

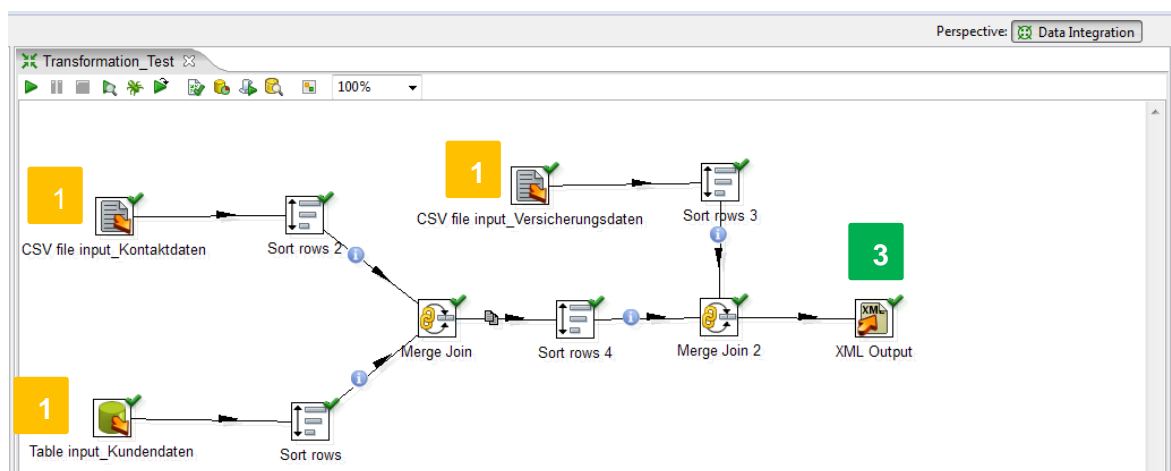


Abb. 25: Ablaufprozess ETL mit unterschiedlichen Komponenten

6.4.1 Input: Datenintegration

Nach der erfolgreichen Verbindung zwischen der MySQL Datenbank und Kettle erfolgt zuerst die Extraktion der benötigten Daten aus den Quellsystemen (MySQL Datenbank und lokaler Rechner) in Kettle. Anhand der Schritte *View>Transformations* und Rechtsklick mit der Maus auf *New* öffnet sich ein neues Fenster für eine Transformation. Über *Design>Input* lassen sich per drag-and-drop unterschiedliche Typen von Daten aus verschiedenen Quellsystemen in das Design Panel integrieren. Durch sogenannte *Hops* kann man Pfeile als Verbindungsglied zwischen einzelnen Transformations- oder Jobschritten einfügen, um einen Datenfluss zu generieren. *Anhang 2* veranschaulicht mögliche Input-Komponenten, die gewählt werden können. Wie bereits erwähnt und in Abbildung 25 anhand der orangenen Kästchen mit der Zahl eins veranschaulicht, beinhaltet das Anwendungsbeispiel zweimal ein *CSV file input* aus dem Quellsystem des lokalen Rechners und einen *Table input* aus der MySQL Datenbank. Durch jeweiliges Klicken auf das Inputsymbol, nach dem Herüberziehen des Symbols in das Design Panel, lässt sich

das Dateiformat in dem Pop-Up Fenster integrieren und wird später bei der Ausführung des Prozesses hochgeladen und eingelesen.

6.4.2 Output-Generierung

Da das gewünschte Ziel bereits bekannt ist, ist es strategisch gesehen sinnvoll als nächsten Schritt die gewünschte Output-Komponente in den Ablaufprozess zu integrieren. Jedoch ist es nicht falsch, wenn man sich zuerst mit dem Transformationsprozess beschäftigt. Für einen neuen Anwender, der sich noch nicht mit allen Transform-Komponenten auskennt, war es jedoch umgekehrt einfacher. Denn durch das Integrieren und Testen verschiedener Transform-Komponenten, kann der Anwender den bestmöglichen Transformationsprozess für sich herausfinden, da es oftmals verschiedene Möglichkeiten gibt oder bei falscher Anwendung bei der Prozessausführung Fehlermeldungen auftreten.

Wie in Abbildung 25 in dem grünen Kästchen mit der Zahl drei gekennzeichnet, besteht die ausgewählte Komponente lediglich aus einer generierten XML-Datei. Durch Anklicken des XML-Symbols, nach Herüberziehen des Symbols in das Design Panel, kann die MySQL Tabelle *Kunde_all* in das Pop-Up Fenster integriert werden. Das bewirkt, dass bei der Ausführung des ETL-Prozesses die MySQL Tabelle mit allen Daten gefüllt wird. Zusätzlich dazu lassen sich die Namen der einzelnen Tabellenspalten überarbeiten. Das heißt, dass Anwender gewisse Bezeichnungen der zukünftigen Tabelle hinzufügen oder löschen können.

Anhang 3 bietet eine Übersicht zu weiteren Output-Komponenten, die der Anwender auswählen kann. Zudem befindet sich darunter jeweils ein Screenshot zu der ausgegeben XML-Datei und der Überführung der veränderten Daten in die Tabelle *Kunde_all* der relationalen Datenbank MySQL. Nach der Ausführung des *execute* Pfeils werden diese zwei Outputs generiert und weitere Informationen zu den Resultaten der Ausführung des ETL-Prozesses erstellt. Diese sind als Screenshots in *Anhang 4 und 5* vorhanden. Vor allem die Zahlen zur Geschwindigkeit der Ausführung, der Performance und sonstige Metriken sind von großem Interesse für den Auftraggeber. Dazu befinden sich weitere Übersichten in *Anhang 8 bis 12*. Diese Metriken sind in die Bewertung von Kettle mit eingeflossen, welche zu einem positiven Ergebnis verholfen haben.

6.4.3 Transform-Prozess

Im Weiteren wird anhand der Abbildung 25 der Prototyp in Bezug auf den Transformationsprozess näher beschrieben. Da der Transformationsschritt für beliebig viele Tabellen und Dateien aus den unterschiedlichsten Quellsystemen durchgeführt werden kann, jedoch zumeist die Daten in unterschiedlichen Formaten und Namen aufzufinden sind,

müssen diese durch Transformationen homogenisiert werden. In diesem Anwendungsbeispiel sind zur Homogenisierung der MySQL Tabelle und der zwei CSV-Dateien nur wenige Schritte notwendig.

Anhand der Transform-Komponente *Sort rows* lassen sich mithilfe von ausgewählten Feldüberschriften die Daten aus den Dateien sortieren. Beispielsweise besitzt die Tabelle *Kundendaten* das Feld *AdressNr*, jedoch heißt das fast identische Feld in der CSV-Datei *Adressnummer*. Durch Klicken auf das ausgewählte *Sort rows* Symbol lassen sich die entsprechenden Feldnamen anzeigen und zum Beispiel eins davon heraussuchen, wonach die Reihenfolge der zwei Datensätze sortiert werden sollen. Die *AdressNr* und *Adressnummer* sollen laut Einstellung sich in aufsteigender Reihenfolge der Nummern sortieren. Das erleichtert den späteren Prozess der Zusammenführung der Daten bei der Ausgabe der XML-Datei, welche lediglich nach der Sortierung noch nicht durchführbar ist. Zumindest zeigt diese Ausführung eine Fehlermeldung in der Übersicht zu den Ausführungsergebnissen in Spoon auf. Deshalb wird ein weiterer Zwischenschritt zur Erstellung einer Verbindung zwischen den verschiedenen Datensätzen benötigt.

Mithilfe der programmspezifischen Funktion *Merge Join* lässt sich festlegen, welche der vorherigen Sortierungsschritte zuerst ausgeführt werden soll, obwohl normalerweise alle Komponenten in einer Transformation in Kontrast zu einem Job parallel ausgeführt werden. Dies wird benötigt, um anschließend wie bei der SQL Programmierung einen *Outer* oder *Inner Join* anzuwenden. Diese Einstellung wird in Kettle lediglich durch das Klicken mit der Maus statt durch die SQL Programmierung verarbeitet, auch wenn es demselben Prinzip folgt. Die etwaigen Programmierungskennnisse des Anwenders sind bei Kettle kein essentieller Bestandteil. Danach kann ein *Key field* für jedes der zwei *Sort rows* eingestellt werden. Diese Hauptfelder sind wie bereits in dem Beispiel zu den *Kontaktdaten* und *Kundendaten* die Felder *AdressNr* und *Adressnummer*, anhand dessen die Datensätze der zwei Dateiformate zusammengeführt werden können. Da sich anschließend ein einheitlicher Datensatz ergibt, welcher mit den Daten der CSV-Datei *Versicherungsdaten* zusammengefügt werden soll, muss derselbe Vorgang noch einmal wiederholt werden. Da sowohl die eben verschmolzene Datei wie auch die Datei *Versicherungsdaten* das Feld *Versicherungsnummer* beinhalten, wird nach dem Feld aufsteigend sortiert. Daraufhin wird ein *Inner Join* angewendet, der die Datensätze verbindet. Im Anschluss an diesen Transformationsprozess kann der vorher beschriebene Output-Vorgang ausgeführt werden bzw. kann durch das Klicken auf den *execute* Pfeil der gesamte aufbereitete ETL-Prozess ausgeführt werden.

Anhang 6 bietet eine Übersicht zu den weiteren Transform-Komponenten in Kettle, welche der Anwender in seinen Prozess einbeziehen kann.

6.4.4 Weitere Funktionen

Weiterführend dazu befindet sich in *Anhang 7* ein Screenshot von Kettles übersichtlich gestalteten grafischen Benutzeroberfläche (Spoon) mit den einzelnen Schritten und Funktionen, die in den Ablaufprozess integriert werden können, den Resultaten der Ausführung des Prozesses und der gesamten Transformation. Da der HALLSCHE KVaG Kettle noch unbekannt ist und ihnen sowohl die Performance, als auch die Übersichtlichkeit des Tools wichtig sind, dienen weitere Screenshots zur Veranschaulichung des Tools. Dabei befinden sich Screenshots zu den Resultaten des ausgeführten Ablaufprozesses wie Metriken zu den einzelnen Prozessschritten, eine grafisch dargestellte Performance zu den Einzelschritten, Fehlermeldungen und eine Übersicht zu den integrierten Daten.

Wie in *Anhang 13* dargestellt, bietet Kettle viele weitere Funktionen und Integrationsmöglichkeiten je nach Anwenderwunsch an, darunter die Join-Funktion, welche in das Anwendungsbeispiel integriert wurde. Die einzelnen Funktionen zu Bulk Loading, Big Data oder auch Kryptografie können aufgrund der Vielzahl und des ausgewählten Fokus auf den ETL-Prozess jedoch nicht weiter ausgeführt werden, da sie sonst den Rahmen der Seminararbeit sprengen würden.

7 Zusammenfassung und Ausblick

(Sarah Ahmed, Muazzez Algac, Lara Hochhalter, Janine Jakob, Ines Schlenker)

In diesem Kapitel werden die Ergebnisse der vorliegenden Seminararbeit zusammengefasst und bewertet. Der Ausblick wird unter anderem eine kritische Betrachtung des Einsatzes eines Open Source ETL-Tools in einem Unternehmen, wie dem Auftraggeber, geben.

7.1 Zusammenfassung und Bewertung

Im Rahmen dieser Seminararbeit wurde die Frage untersucht, welches Open Source ETL-Tool am überzeugendsten in Bezug auf...

- die Anzahl der Schnittstellen zur Anbindung unterschiedlicher Quell- / Zielsysteme,
- die Performance,
- die regelbasierte Transformation der Daten,
- die Benutzerfreundlichkeit und
- die Marktrelevanz und Zukunftssicherheit ist.

ETL-Tools sind die Schnittstelle für Business Intelligence-Anwendungen. Sie steuern den Datenfluss zwischen unzähligen Datenquell-Systemen und BI-Anwendungen. Umso wichtiger ist es, dass das effizienteste und effektivste Tool für die Bearbeitung gewählt wird.

Nachdem die einzelnen ETL-Werkzeuge vorgestellt wurden, erfolgte eine Vorauswahl basierend auf den Ergebnissen des Grobkriterienkatalogs. Die Vorauswahl schloss drei der fünf Tools aus. In die engere Auswahl kamen *CloverETL* und *Kettle*, da die anderen Tools aufgrund fehlender Informationen und der marginalen Benutzer-Community nicht die Anforderungen für eine genauere Untersuchung erfüllen.

Die weitere Evaluation der beiden ausgewählten ETL-Werkzeuge umfasste eine Kombination praktischer Tests, die auf einem detaillierten Feinkriterienkatalog basierten. Durch die Bewertung der Feinkriterien stellte sich heraus, dass das ETL-Tool *Kettle* sowohl beim Anbieterprofil als auch beim Produktprofil eindeutige Stärken aufwies und sich somit gegen *CloverETL* durchsetzte.

Die Vorteile des ETL-Werkzeugs *Kettle* sind:

- Höhere Performance
- Höhere Marktakzeptanz
- Höhere Benutzerfreundlichkeit

Zudem weist Kettle einen überraschend hohen Reifegrad vor, der sich sowohl in guter Bedienbarkeit äußert, als auch in hoher Stabilität.

Aufgrund dieser Analyse wird Kettle als das einzusetzende ETL-Tool empfohlen. Die Gründe für diese Entscheidung liegen primär in der Marktrelevanz und in der Performance des Tools.

Diese Entscheidung wurde zusätzlich durch eine prototypische Entwicklung eines ETL-Prozesses bestätigt. Der ETL-Prozess, der CSV und XML Daten extrahiert, transformiert und danach in einer relationalen Datenbank abspeichert, wurde beispielhaft mit dem evaluierten ETL-Tool umgesetzt.

Insgesamt hat die Evaluation ergeben, dass CloverETL und Kettle die zwei zu favorisierenden ETL-Werkzeuge sind und auch beide die nötigen Funktionalitäten aufweisen. Kettle jedoch überzeugt durch die effizienteren und leistungsstärkeren Funktionen.

7.2 Ausblick

Die im Rahmen dieser Seminararbeit geäußerte Empfehlung ist trotz der detaillierten Bewertung und Evaluation mithilfe der Kriterienkataloge und des Prototyps, eher allgemein zu betrachten. Der Grund dafür ist, dass jedes Unternehmen spezielle Anforderungen hat und somit für jedes Unternehmen ein anderes ETL-Tool am günstigsten ist. Die Evaluation der fünf ausgewählten Tools in dieser Arbeit basieren auf den Anforderungen des Auftraggebers, jedoch konnten die Tests nur mit einer vergleichsweise geringen Datenmenge durchgeführt werden – im alltäglichen Gebrauch des ETL-Tools im Unternehmen kann es sehr gut vorkommen, dass größere Mengen innerhalb einer kürzeren Zeit mit einem höheren Aufwand verarbeitet werden müssen.

Deswegen ist es recht interessant, die zukünftigen Entwicklungen im Open Source Bereich weiterhin zu beobachten und zu verfolgen. Das Themengebiet der ETL-Tools bietet noch sehr viel Potenzial, welches von Unternehmen noch ausgeschöpft werden sollte. Heutzutage spielen auch Cloud-Dienste durch den technologischen Fortschritt eine große Rolle was BI-Lösungen angeht. So gilt es, zu untersuchen, wie diese Dienste für ETL-Tools genutzt werden können.

8 Anhang

Anhangverzeichnis

Anhang 1: Feinkriterienkatalog und Bewertung der zwei ausgewählten ETL-Tools	63
Anhang 2: Mögliche Input-Komponenten für die Datenintegration in Kettle	64
Anhang 3: Weitere mögliche Output-Komponenten in Kettle	65
Anhang 4: Die gefüllte Tabelle <i>Kunde_all</i> in MySQL als Output des ETL-Prozesses	65
Anhang 5: Die XML-Datei als Output des ETL-Prozesses	66
Anhang 6: Weitere mögliche Transform-Komponenten in Kettle	67
Anhang 7: Die grafische Benutzeroberfläche Spoon	68
Anhang 8: Die Ergebnisdaten im Überblick in Kettle	68
Anhang 9: Datensammlung (und evtl. Fehlermeldungen)	69
Anhang 10: Metriken zu den einzelnen Schritten des Ablaufprozesses	69
Anhang 11: Weitere Metriken	70
Anhang 12: Graphen zu der Performance der einzelnen und gesamten Prozessschritte	70
Anhang 13: Weitere mögliche Funktionen in Kettle	71

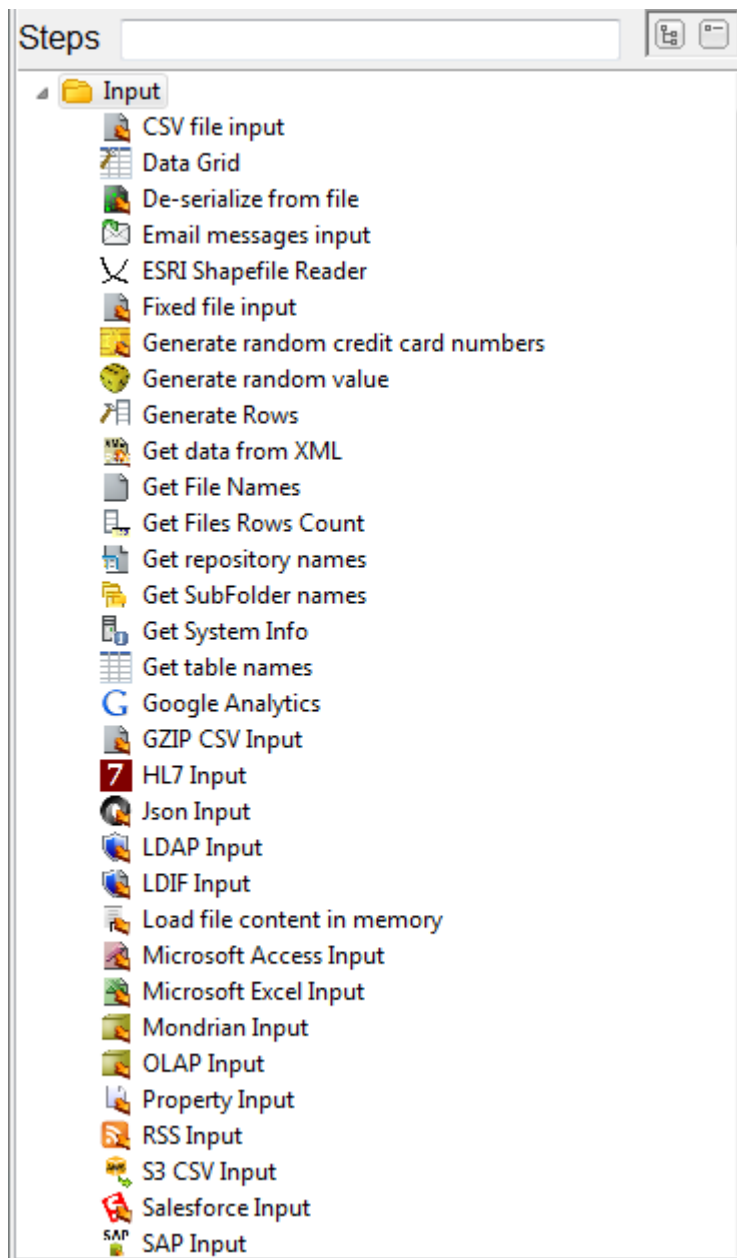
Kriterienkatalog zur Bewertung verschiedener Open Source ETL-Tools

Legende: 0 1 2 3 4 5
(k.A. Keine Angabe, 0= sehr schlecht , 5= sehr gut)

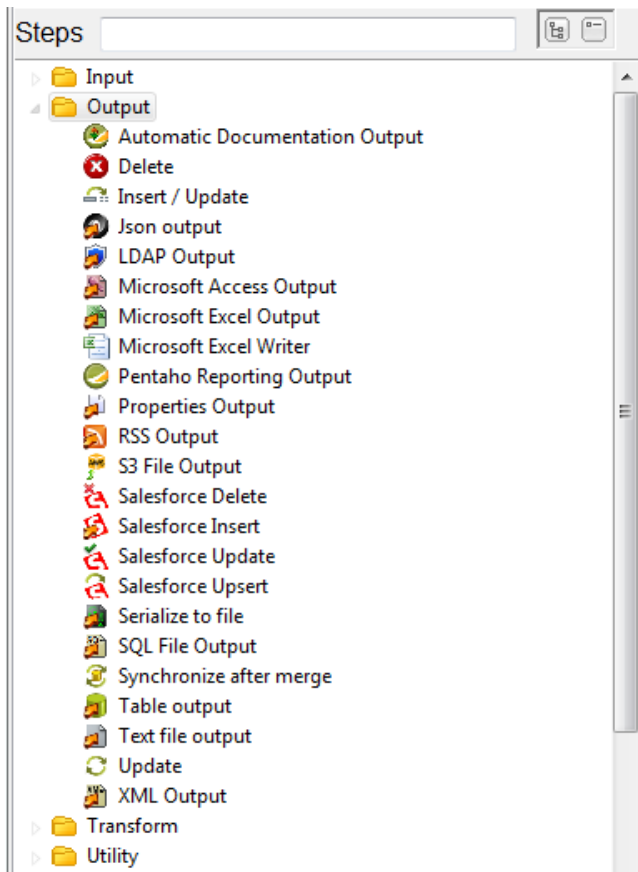
Oberkategorie	Kategorie	Gewichtung Kategorie	Kriterium	Gewichtung Kriterium	Kette	CloverETL
PRODUKTPROFIL	Ökosystem des Herstellers	10%	Drittanbieter für Beratung und externe Dienstleistung	30%	5	k.A.
			Partner zur Bereitstellung von Schnittstellen	30%	4	5
			Stakeholder/ Kunden	40%	5	5
			Zwischenergebnis	100%	4,7	3,5
	Funktionalität	20%	Regelbasierte Transformation der Daten	25%	3	4
			Richtigkeit	25%	3	4
			Interoperabilität	25%	5	3
			Modularität	25%	5	5
	Zwischenergebnis	100%	4,0	4,0		
	Leistung	20%	Zuverlässigkeit	40%	5	4
			Verfügbarkeit	30%	3	4
			Skalierbarkeit	30%	4	3
			Zwischenergebnis	100%	4,1	3,7
	Bedienbarkeit/ Ergonomie/ Handhabung	20%	Verständlichkeit und Erlernbarkeit	50%	4	3
			Graphische Benutzeroberfläche	10%	5	5
			Fehlertoleranz	30%	4	2
			Komplexität der Administration	10%	3	3
	Zwischenergebnis	100%	4,0	2,9		
Performance/ Effizienz	15%	Individuelle Anpassung an die jeweilige Hardware	10%	5	3	
		Geschwindigkeit und Datendurchsatz	30%	5	2	
		Ressourcenverbrauch	30%	4	4	
		Reaktionszeit auf Benutzereingaben	30%	5	2	
Zwischenergebnis	100%	4,7	2,7			
Datensicherheit/ Zukunftssicherheit	15%	Backup	40%	4	k.A.	
		Manipulierbarkeit	30%	5	4	
		Releases	30%	4	3	
		Zwischenergebnis	100%	4,3	2,1	
Gesamtergebnis des Produktprofils		100%			4,2	3,2
ANBIETERPROFIL	Marktakzeptanz	25%	Google Trends	10%	4	1
			Auszeichnungen/ Zertifizierungen	30%	5	k.A.
			Verbreitung (Entwickler, Anwender, Downloads)	60%	5	3
			Zwischenergebnis	100%	4,9	1,9
	Aktivität der Community	25%	Forenaktivität	40%	5	5
			Direkte Ansprechpartner	40%	3	4
			FAQ	20%	5	4
			Zwischenergebnis	100%	4,2	4,4
	Dokumentation	20%	Code Dokumentation	40%	4	3
			Benutzerhandbuch	40%	5	5
			Sprache der Dokumentation	20%	4	4
			Zwischenergebnis	100%	4,4	4,0
Dienstleistungsangebot	15%	Training durch den Anbieter	30%	4	4	
		Beratung durch den Anbieter	20%	5	3	
		Training durch Drittanbieter	30%	k.A.	2	
		Beratung durch Drittanbieter	20%	k.A.	k.A.	
Zwischenergebnis	100%	2,2	2,4			
Finanzierbarkeit	15%	Lizenz	60%	5	5	
		Version	40%	4	4	
		Zwischenergebnis	100%	4,6	4,6	
Gesamtergebnis des Anbieterprofils		100%			4,2	3,4

Ergebnisse in der Übersicht		Gewichtung des Profils	Kette	CloverETL
Produktprofil		60%	4,2	3,2
Anbieterprofil		40%	4,2	3,4
Gesamtergebnis des Kriterienkatalogs		100%	4,2	3,3

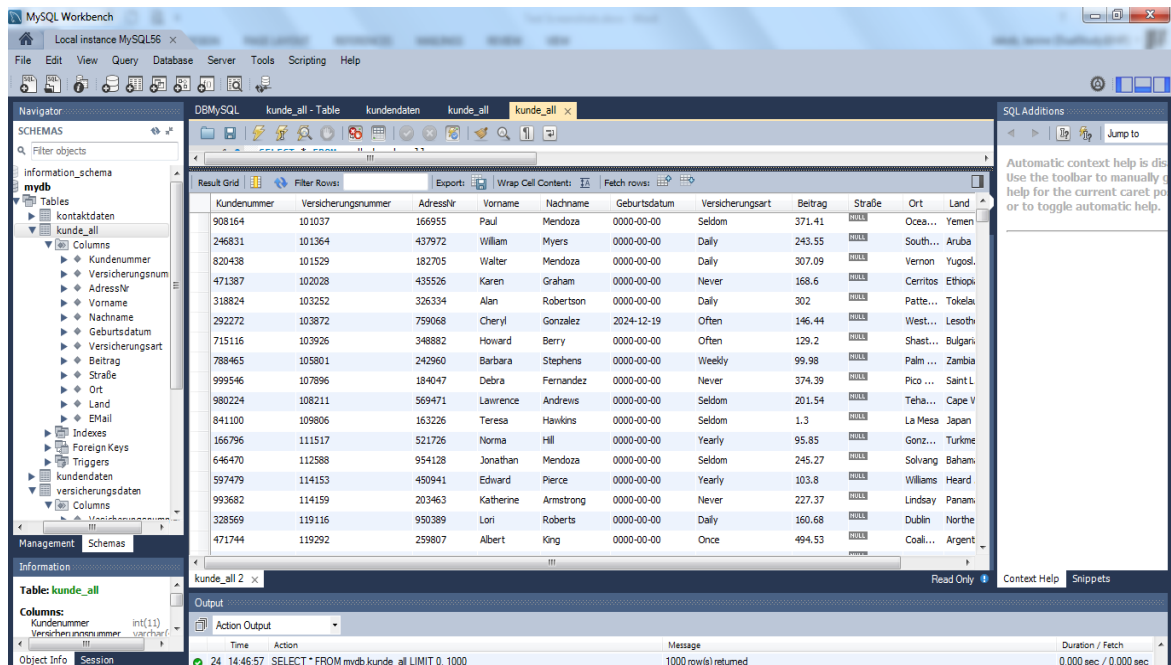
Anhang 1: Feinkriterienkatalog und Bewertung der zwei ausgewählten ETL-Tools



Anhang 2: Mögliche Input-Komponenten für die Datenintegration in Kettle



Anhang 3: Weitere mögliche Output-Komponenten in Kettle



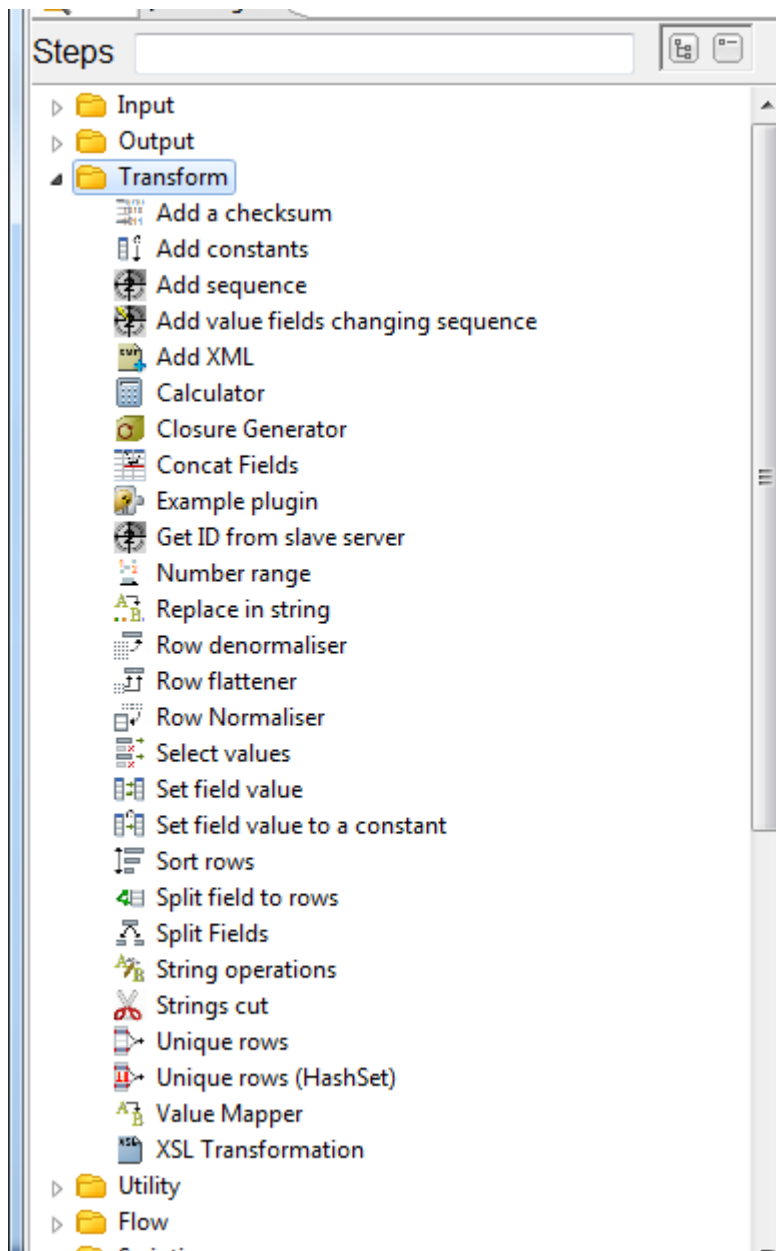
Anhang 4: Die gefüllte Tabelle Kunde_all in MySQL als Output des ETL-Prozesses

```

<Strasse>209</Strasse>
<Ort>Rolling Hills Estates</Ort>
<Land>Thailand</Land>
<Email>jsnyder@quire.org</Email>
</Row>
- <Row>
  <Versicherungsnummer>243150</Versicherungsnummer>
  <Versicherungsart>Oftan</Versicherungsart>
  <Versicherungsbeginn>10.011.997</Versicherungsbeginn>
  <Monatsbeitrag>98,4</Monatsbeitrag>
  <Kundennummer>400562</Kundennummer>
  <Vorname>Clarence</Vorname>
  <Nachname>Rodriguez</Nachname>
  <Geburtsdatum>0000-00-00</Geburtsdatum>
  <Versicherungsnummer_1>243150</Versicherungsnummer_1>
  <AdressNr>252882</AdressNr>
  <Adressnummer>252882</Adressnummer>
  <Strasse>876</Strasse>
  <Ort>Redlands</Ort>
  <Land>Monaco</Land>
  <Email>kcollins@blogxs.com</Email>
</Row>
- <Row>
  <Versicherungsnummer>243536</Versicherungsnummer>
  <Versicherungsart>Monthly</Versicherungsart>
  <Versicherungsbeginn>6.031.998</Versicherungsbeginn>
  <Monatsbeitrag>122,2</Monatsbeitrag>
  <Kundennummer>525798</Kundennummer>
  <Vorname>Ryan</Vorname>
  <Nachname>Wright</Nachname>
  <Geburtsdatum>0000-00-00</Geburtsdatum>
  <Versicherungsnummer_1>243536</Versicherungsnummer_1>
  <AdressNr>639207</AdressNr>
  <Adressnummer>639207</Adressnummer>
  <Strasse>7</Strasse>
  <Ort>Soledad</Ort>
  <Land>Swaziland</Land>
  <Email>fshaw@dynabox.org</Email>
</Row>
- <Row>
  <Versicherungsnummer>244071</Versicherungsnummer>
  <Versicherungsart>Oftan</Versicherungsart>
  <Versicherungsbeginn>19.121.996</Versicherungsbeginn>
  <Monatsbeitrag>427</Monatsbeitrag>
  <Kundennummer>861506</Kundennummer>
  <Vorname>Diana</Vorname>
  <Nachname>Bennett</Nachname>

```

Anhang 5: Die XML-Datei als Output des ETL-Prozesses



Anhang 6: Weitere mögliche Transform-Komponenten in Kettle

The screenshot shows the Spoon Transformation Studio interface. The top part displays a workflow diagram with the following steps: 'Table input_Kundendaten' and 'CSV file input_Kontaktdaten' feed into 'Sort rows'. 'Sort rows' feeds into 'Merge Join'. 'Merge Join' feeds into 'Sort rows 4'. 'Sort rows 4' feeds into 'Merge Join 2'. 'Merge Join 2' feeds into 'Sort rows 3'. 'Sort rows 3' feeds into 'Merge Join 2'. 'Merge Join 2' feeds into 'XML Output'. The bottom part shows the 'Execution Results' table.

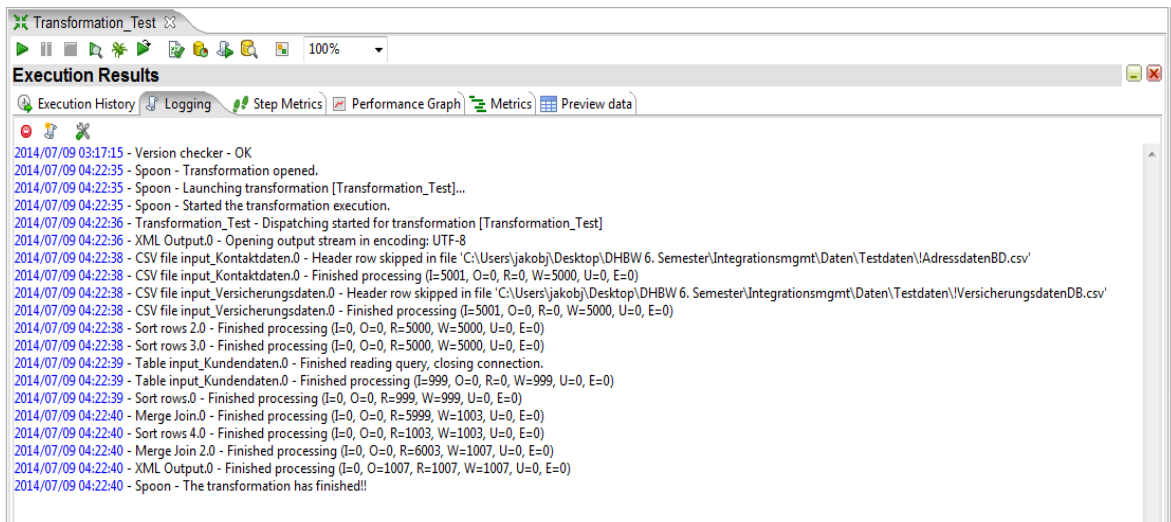
#	Versicherungsnummer	Versicherungsart	Versicherungsbeginn	Monatsbeitrag	Kundennummer	Vorname	Nachname	Geburtsdatum	Versic
406	476458	Oftentimes	25.062.006	79,6	793109	Catherine	Torres	0000-00-00	47645
407	477907	Yearly	9.112.003	144,3	544799	Sean	Stone	0000-00-00	47790
408	478047	Weekly	1.012.002	141,5	901247	Adam	Henderson	0000-00-00	47804
409	478379	Daily	3.032.012	129,2	453169	Frances	Sanchez	0000-00-00	47837
410	479290	Monthly	2.082.006	277,1	472601	Shirley	Welch	2025-05-19	47929
411	480053	Daily	30.052.006	59	181797	Earl	Shaw	0000-00-00	48005
412	480401	Oftentimes	21.081.999	78,1	227782	Sean	Hicks	2017-01-19	48040

Anhang 7: Die grafische Benutzeroberfläche Spoon

The screenshot shows the 'Execution Results' table in Spoon Transformation Studio. The table contains 10 columns: #, Versicherungsnummer, Versicherungsart, Versicherungsbeginn, Monatsbeitrag, Kundennummer, Vorname, Nachname, Geburtsdatum, Versicherungsnummer_1, and AdressNr. The data rows are numbered from 928 to 953.

#	Versicherungsnummer	Versicherungsart	Versicherungsbeginn	Monatsbeitrag	Kundennummer	Vorname	Nachname	Geburtsdatum	Versicherungsnummer_1	AdressNr
928	934868	Weekly	16.112.002	285,1	394181	Daniel	Black	0000-00-00	934868	386144
929	935181	Once	27.092.005	369,4	442193	Justin	Oliver	0000-00-00	935181	397526
930	936302	Oftentimes	18.051.998	200,6	368354	Kathryn	Clark	0000-00-00	936302	607960
931	937430	Once	26.072.007	402,1	892581	Ryan	Nichols	0000-00-00	937430	115759
932	937848	Monthly	28.082.011	102,7	221341	Irene	Payne	0000-00-00	937848	628253
933	939599	Monthly	8.061.996	202,4	637380	Jennifer	Hicks	0000-00-00	939599	747795
934	939840	Never	3.122.004	256,8	448492	Walter	Parker	0000-00-00	939840	337103
935	940358	Weekly	4.112.004	366,3	340249	Andrew	Cox	2016-01-19	940358	727420
936	941606	Once	22.112.003	433,9	879732	Norma	Murray	0000-00-00	941606	236399
937	941691	Daily	5.101.996	362,6	169467	Alice	Berry	2003-05-19	941691	862533
938	942203	Yearly	24.052.004	315,7	295014	Frances	Lynch	0000-00-00	942203	748406
939	942676	Once	24.071.998	11,8	629443	Angela	Powell	0000-00-00	942676	633501
940	942833	Daily	18.121.998	163	698819	Sarah	Martin	0000-00-00	942833	461412
941	943022	Seldom	5.122.003	326,6	392838	Christopher	Cruz	0000-00-00	943022	508581
942	943417	Oftentimes	19.082.011	14,1	784138	Anne	Johnston	0000-00-00	943417	854732
943	943490	Monthly	19.022.002	112,4	416851	Janet	Oliver	0000-00-00	943490	546774
944	944011	Once	23.022.003	466,6	504310	Randy	Simmons	2011-10-19	944011	451735
945	944449	Never	15.062.007	140,8	375928	Richard	Griffin	0000-00-00	944449	285295
946	944822	Oftentimes	16.012.000	471,9	747400	Anne	Hayes	0000-00-00	944822	131626
947	947325	Oftentimes	26.082.006	470,4	427192	Todd	Hughes	0000-00-00	947325	184483
948	947454	Weekly	3.082.002	389,1	144570	Catherine	Day	2019-10-19	947454	101336
949	947512	Weekly	9.091.996	335,8	322905	Sean	Jacobs	2001-10-19	947512	339322
950	948650	Yearly	9.082.000	204,1	616125	Dorothy	Bishop	2021-01-19	948650	205136
951	949188	Yearly	11.032.013	63,2	226429	Carol	Ross	2022-07-19	949188	553309
952	950357	Once	7.011.999	491	926807	Deborah	Price	0000-00-00	950357	634462
953	950702	Weekly	8.062.001	454,9	494159	Kevin	Rivera	0000-00-00	950702	248365

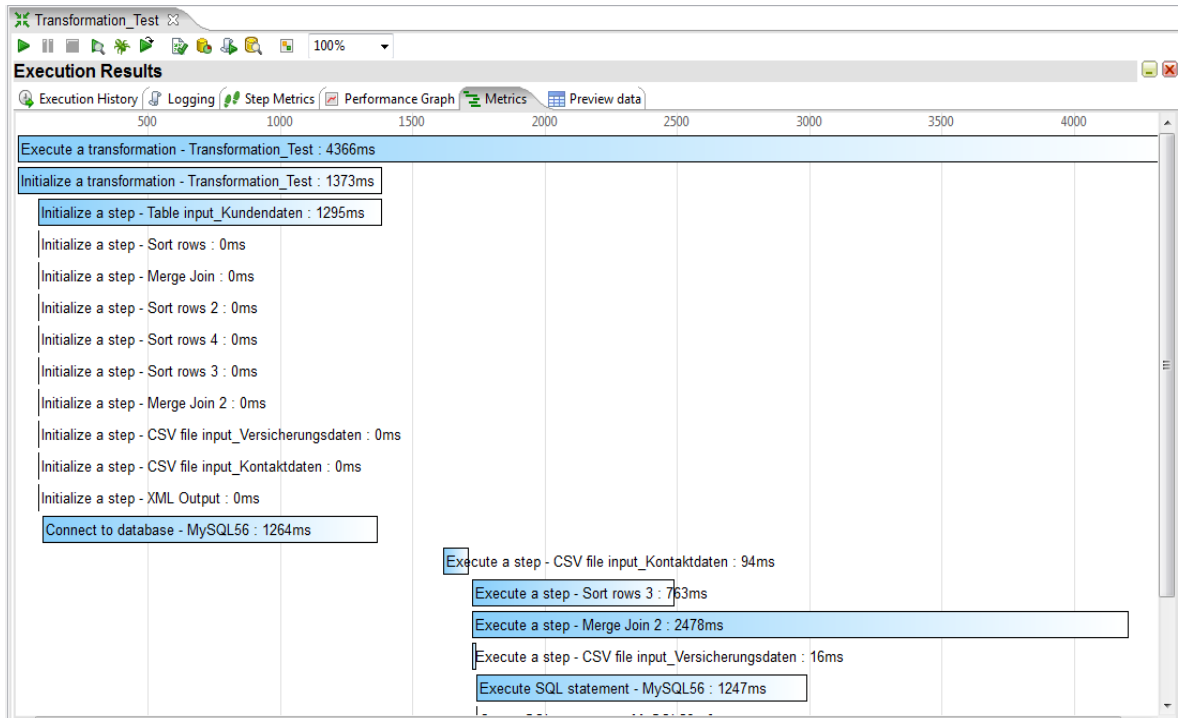
Anhang 8: Die Ergebnisdaten im Überblick in Kettle



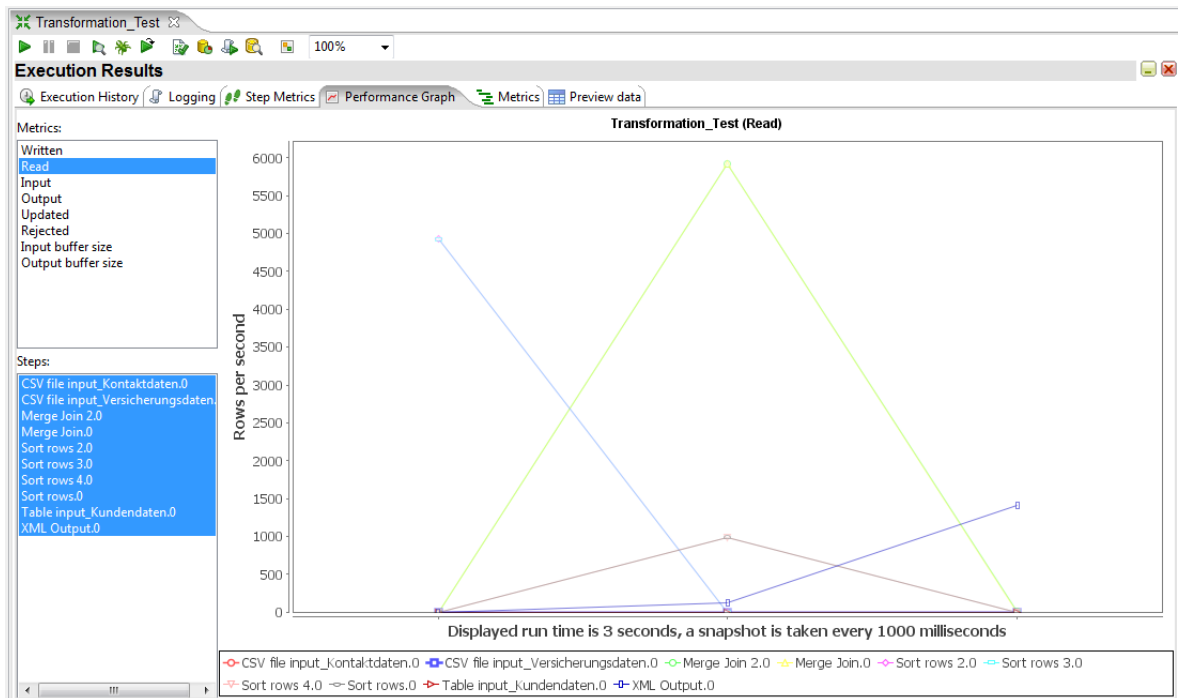
Anhang 9: Datensammlung (und evtl. Fehlermeldungen)

#	Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)
1	Table input_Kundendaten	0	0	999	999	0	0	0	0	Finished	1.5s	667
2	CSV file input_Kontakttdaten	0	0	5000	5001	0	0	0	0	Finished	0.1s	53,202
3	CSV file input_Versicherungsdaten	0	0	5000	5001	0	0	0	0	Finished	0.1s	40,008
4	Sort rows 2	0	5000	5000	0	0	0	0	0	Finished	0.8s	5,945
5	Sort rows 3	0	5000	5000	0	0	0	0	0	Finished	0.9s	5,734
6	Sort rows	0	999	999	0	0	0	0	0	Finished	1.5s	653
7	Merge Join	0	5999	1003	0	0	0	0	0	Finished	2.0s	3,031
8	Sort rows 4	0	1003	1003	0	0	0	0	0	Finished	2.0s	499
9	Merge Join 2	0	6003	1007	0	0	0	0	0	Finished	2.6s	2,320
10	XML Output	0	1007	1007	0	1007	0	0	0	Finished	2.8s	365

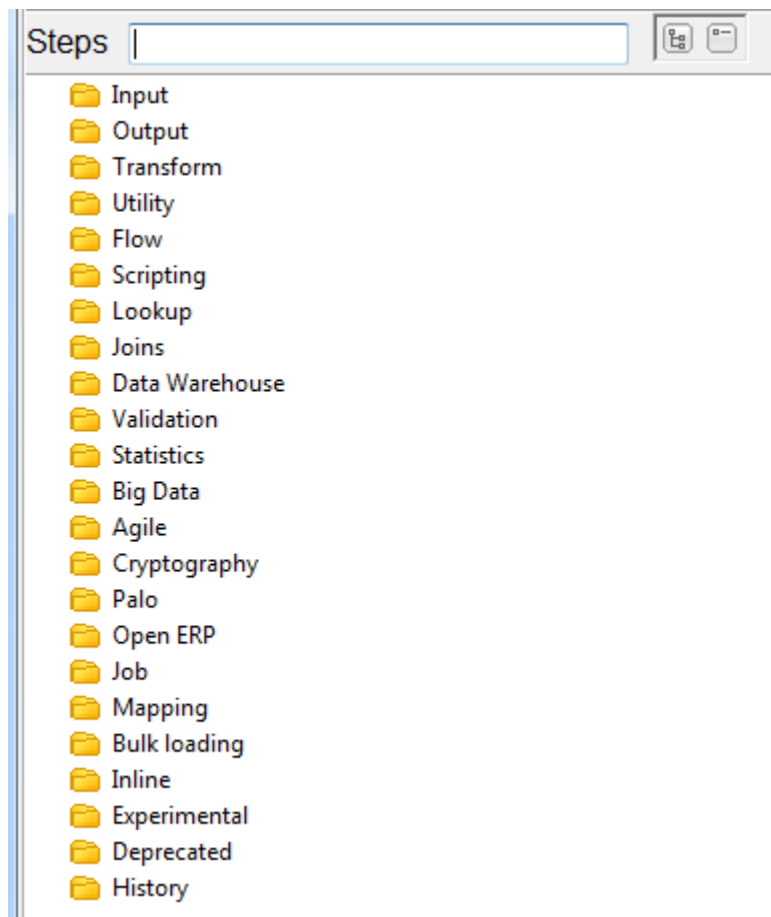
Anhang 10: Metriken zu den einzelnen Schritten des Ablaufprozesses



Anhang 11: Weitere Metriken



Anhang 12: Graphen zu der Performance der einzelnen und gesamten Prozessschritte



Anhang 13: Weitere mögliche Funktionen in Kettle

Quellenverzeichnisse

Literaturverzeichnis

- Bandel, G. (2010): Open Source Software: Fördert oder hemmt diese Art der Softwareentwicklung den Wettbewerb?, Hamburg: Diplomica
- Bosch, J. (2009): From software product lines to software ecosystems', Proceeding SPLC '09 Proceedings of the 13th International Software Product Line Conference Carnegie Mellon University Pittsburgh, 2009, 111–119
- Haneke, U. et al. (2010): Open Source Business Intelligence (OSBI): Möglichkeiten, Chancen und Risiken quelloffener BI-Lösungen, München, Carl Hanser Verlag
- MacFarlane, F. (2013): Open Source Enterprise-Resource-Planning (ERP) Software: Evaluation, Installation und Test – Eine Machbarkeitsstudie, Hamburg, Diplomica Verlag GmbH
- Meier, A. (1995): Wer braucht Kriterienkataloge? In: P. Schenkel und H. Holz [Hrsg.]: Evaluation multimedialer Lernprogramme und Lernkonzepte. Nürnberg, Bildung und Wissen. S. 189ff.
- Rossak, I. (2013): Datenintegration, München, Carl Hanser Verlag
- Schaaf, A. (2013): Open-Source-Lizenzen, Untersuchung der GPL, LGPL, BSD und Artistic License, Hamburg: Diplomica

Internetquellenverzeichnis

- Adeptia (o.J.): About Adeptia,
<http://www.adeptia.com/company/company.html>,
Abruf: 26.06.2014
- Bienert, J. (2011): Big Data in 2-5 Jahren Mainstream - Gartner's 2011 Hype Cycle Special Report,
<http://www.bigdata.de/2011/08/10/bigdata-in-2-5-jahren-%20mainstream/>, Abruf 29.05.2014
- Big Data (o.J.): Big Data ,http://www.bigdata.de/wp-content/uploads/2011/08/215650_0001.gif, Abruf: 05.06.2014
- Bundesamt für Sicherheit in der Informationstechnik (o.J.): Glossar und
nen, https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/Glossar/glossar_node.html, Abruf: 23.06.2014
- CloverETL (o.J.): CloverETL Designer User's
Guide <http://doc.cloveretl.com/documentation/UserGuide/index.jsp?topic=/com.cloveretl.gui.docs/docs/index.html>, Abruf 15.06.2014
- CloverETL (2014a): Frequently Asked Questions (FAQ),
<http://www.cloveretl.com/faq>, Abruf: 03.07.2014
- CloverETL (2014b): CloverETL Training Program,
<http://www.cloveretl.com/services/training>, Abruf: 03.07.2014
- CloverETL (2014c): Open Source & Community Edition,
<http://www.cloveretl.com/products/open-source>,
Abruf: 03.07.2014
- CloverETL (o.J.): CloverETL Designer User's Guide
<http://doc.cloveretl.com/documentation/UserGuide/index.jsp?topic=/com.cloveretl.gui.docs/docs/index.html>,
Abruf 15.06.2014
- Dialog Data (o.J.): Softwarebewertung, <http://www.dialogdata.com/org/qs>

- tung, <http://www.dialogdata.com/org/qs/qs5sw11bew.html>, Abruf 12.06.2014
- Duden (2013): FAQ, <http://www.duden.de/rechtschreibung/FAQ>, Abruf: 24.06.2014
- Enzyklo (o.J.): Drittanbieter, <http://www.enzyklo.de/Begriff/Drittanbieter>, Abruf: 24.06.2014
- Essiscope (o.J.): ISO 9126: The Standard of Reference, <http://www.cse.dcu.ie/essiscope/sm2/9126ref.html>, Abruf 12.06.2014
- ETLTools.net (o.J.): Free OpenSource ETL Tools, <http://www.etltools.net/clover-etl.html>, Abruf 15.06.2014
- Floor (o.J.): Gartner's Hype Cycle, <http://www.floor.nl/ebiz/gartnerhypecycle.htm>, Abruf: 29.05.2014
- Frewo (o.J.): Adeptia, http://www.mansasys.com/.../img/adeptia_logo_transparent.png, Abruf: 26.06.2014
- Gabler Wirtschaftslexikon (o.J.): Open Source, <http://wirtschaftslexikon.gabler.de/Definition/open-source.html>, Abruf: 04.06.2014
- Gallardo, E. (2013): Adeptia Overview, <http://www.bi-lab.com.ar/adeptia/adeptia/overview>, Abruf: 03.07.2014
- Google Trends (2014): Adeptia, <https://www.google.de/trends/explore#q=Adeptia>, Abruf: 25.06.2014
- Gustke, S. (2011): Vorgehensmodell Softwareevaluierung, http://www.skssolutions.de/uploads/media/Vorgehensmodell_Softwareevaluierung_ibbs.pdf, Abruf 12.06.2014
- Haneke, U. (2012): Die Nutzung von Open-Source-Business-Intelligence-Systemen im öffentlichen Sektor, <http://www.opensourcepublicsector.de/?p=96>, Abruf: 05.06.2014
- Heines, A. (2012): Wie werden Unternehmen agil?,

- Herzwurm, G. / Mikusz, M,(2013): <http://www.projektlandschaften.de/allgemein/wie-werden-unternehmen-agil.html>, Abruf: 08.07.2012
- Herzwurm, G. / Mikusz, M,(2013): Qualitätsmerkmale von Software, <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/is-management/Systementwicklung/Management-der-Systementwicklung/Software-Qualitätsmanagement/Qualitätsmerkmale-von-Software/index.html>, Abruf 12.06.2014
- IBM (o.J.): What is big data?, <http://www-01.ibm.com/software/data/bigdata/>, Abruf:05.06.2014
- IBSolution (2012): SAP HANA - Überblick und Möglichkeiten, <http://www.youtube.com/watch?v=zgsClqwNib4&feature=related>, Abruf: 10.06.2014
- ifrOSS (2014): Welches sind die wichtigsten Open Source Lizenzen und welchem Lizenztyp gehören sie an?, <http://www.ifross.org/welches-sind-wichtigsten-open-source-lizenzen-und-welchem-lizenztyp-gehoren-sie>, Abruf: 20.06.2014
- It Wissen (o.J): Availability, www.itwissen.info/definition/lexikon/Verfuegbarkeit-availability.html , Abruf: 15.06.2014
- Jitterbit (2014a): Company, <http://www.jitterbit.com/company/>, Abruf: 27.06.2014
- Jitterbit (2014b): Jitterbit, <http://www.jitterbit.com/>, Abruf: 26.06.2014
- Jitterbit (2014c): Learn: Integration Made Easy, <http://www.jitterbit.com/learn/>, Abruf: 01.07.2014
- Jitterbit (2014d): Data Integration & ETL, <http://www.jitterbit.com/solutions/etl-data-integration/>, Abruf: 01.07.2014
- Klüpfel, S. / Mayer, T. (2007): Checkliste und Kriterienkatalog zur Unterstützung der Softwareauswahl, <http://www.hwk->

- kon-
stanz.de/html/seiten/outputAdbFileNew.php?id=2556,
Abruf 12.06.2014
- Lackes, R. (o.J.): Modularität,
<http://wirtschaftslexikon.gabler.de/Definition/modularitaet.html>, Abruf: 15.06.2014
- Ohloh Black Duck Open HUB Clover ETL, <http://www.ohloh.net/p/4270>, Abruf: 15.06.2014
- (2014):
- Open Source Initiative (2014): The Open Source Definition,
<http://opensource.org/osd>, Abruf: 04.06.2014
- Passioned Group (o.J.): PENTAHO DATA INTEGRATION (KETTLE),
<http://www.etltool.com/etl-vendors/pentaho-data-integration-kettle/>, Abruf: 25.06.2014
- Pentaho Corporation (2014): Pentaho Awarded Fifth Consecutive InfoWorld Bossie Award, <http://www.pentaho.com/press-room/releases/98/pentaho-awarded-fifth-consecutive-infoworld-bossie-award>, Abruf: 28.06.2014
- pmOne (o.J.): Big Data – Marketing-Hype oder echter Mehrwert?,
<http://www.pmone.de/de/im-fokus/bigdata/>,
Abruf: 05.06.2014
- Schoenberg-solutions GmbH Kompetenz in Softwareentwicklung und IT-Beratung, <http://www.schoenberg-solutions.de/schoeso/>, Abruf: 29.06.2014
- (2014):
- Schmid, O. (o.J.): Aktuelle Themen der Informatik - IT Infrastructure Library Release Management, http://webuser.hs-furtwangen.de/~kaspar/seminar0405/ReleaseManagement_ausarbeitung.pdf, Abruf: 23.06.2014
- Sean (2014): Skalierbarkeit, <http://www.fremdwort.de/suchen/bedeutung/Skaliierbarkeit>, Abruf 15.06.2014
- Siepermann, M. / Lackes, R. (o.J.) Zuverlässigkeit, <http://wirtschaftslexikon.gabler.de/Arckheit>, <http://wirtschaftslexikon.gabler.de/Archiv/77319/zuverlaessigkeit-v7.html>, Abruf 15.06.2014
- Solutions Review (2014a): Data Integration Solutions Buyers Guide,
[KOS.content 2.2014 | 174](http://solutions-</p>
</div>
<div data-bbox=)

- re-
view.com//dl/2014_Solutions_Review_Data_Integration_Buyers_Guide_MK63.pdf, Abruf: 25.06.2014
- Solutions Review (2014b): Data Integration Solutions Directory, <http://solutions-review.com/data-integration/data-integration-solutions-directory/>, Abruf: 03.07.2014
- SourceForge (2014a): Jitterbit – Open Source Integration, <http://sourceforge.net/projects/jitterbit/>, Abruf: 01.07.2014
- SourceForge (2014b): Jitterbit – Open Source Integration, <http://sourceforge.net/projects/jitterbit/files/stats/map?dates=2014-01-01%20to%202014-07-01>, Abruf: 01.07.2014
- SourceForge (2014c): TimeLine, <http://sourceforge.net/projects/timeline/files/stats/timeline>, Abruf: 27.06.2014
- Stackoverflow (o.J.): What does "bulk load" mean?, <http://stackoverflow.com/questions/4462074/what-does-bulk-load-mean>, Abruf: 03.07.2014
- Standen, J.(2009): Making rapid prototypes for data warehouse ETL jobs, <http://www.datamartist.com/making-rapid-prototypes-for-data-warehouse-etl-jobs>, Abruf: 05.06.2014
- Stöckl, C. (2006): Konzeption und Evaluation eines Kriterienkataloges für rechnergestützte Befundübermittlung zur kooperativen Versorgung, http://iig.umit.at/dokumente/phd_stoeckl.pdf, Abruf 12.06.2014
- Ulrych, J. (2011) ETL, <http://www.slideshare.net/cloveretl/introduction-to-etl-and-data-integration>, Abruf 15.06.2014
- Wolff-Marting, V. / Rothmann, P.(2011): Software Ökosysteme <http://it-ra->

dar.eu/serendipity/uploads/transkripte/Zusammenfassung_Interview_Jansen.pdf, Abruf 15.06.2014

Woodie, A. (2013):

Jitterbit unveils new version of integration tool,
<http://www.itjungle.com/fhs/fhs011513-story05.html>,
Abruf: 27.06.2014

Xineo.net(o.J.):

Xineo XIL Tutorial,
<http://software.xineo.net/doc/xml/xil/tutorial.html>,
Abruf: 26.06.2014

Ehrenwörtliche Erklärung

Ich, **Ines Schlenker**, erkläre ehrenwörtlich,

1. dass ich meine Seminararbeit ohne fremde Hilfe angefertigt habe;
2. dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe;
3. dass ich meine Seminararbeit bei keiner anderen Prüfung vorgelegt habe.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Stuttgart, 11.07.2014

(Ort, Datum)

(Unterschrift)

Ehrenwörtliche Erklärung

Ich, **Muazzez Algac**, erkläre ehrenwörtlich,

1. dass ich meine Seminararbeit ohne fremde Hilfe angefertigt habe;
2. dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe;
3. dass ich meine Seminararbeit bei keiner anderen Prüfung vorgelegt habe.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Stuttgart, 11.07.2014

(Ort, Datum)

(Unterschrift)

Ehrenwörtliche Erklärung

Ich, **Sarah Ahmed**, erkläre ehrenwörtlich,

1. dass ich meine Seminararbeit ohne fremde Hilfe angefertigt habe;
2. dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe;
3. dass ich meine Seminararbeit bei keiner anderen Prüfung vorgelegt habe.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Stuttgart, 11.07.2014

(Ort, Datum)

(Unterschrift)

Ehrenwörtliche Erklärung

Ich, **Lara Hochhalter**, erkläre ehrenwörtlich,

1. dass ich meine Seminararbeit ohne fremde Hilfe angefertigt habe;
2. dass ich die Übernahme wörtlicher Zitate aus der Literatur sowie die Verwendung der Gedanken anderer Autoren an den entsprechenden Stellen innerhalb der Arbeit gekennzeichnet habe;
3. dass ich meine Seminararbeit bei keiner anderen Prüfung vorgelegt habe.

Ich bin mir bewusst, dass eine falsche Erklärung rechtliche Folgen haben wird.

Stuttgart, 11.07.2014

(Ort, Datum)

(Unterschrift)

Marktanalyse clientseitiger Open Source Frameworks zum Bau browserbasierter Anwendungen für mobile Endgeräte

Schriftliche Ausarbeitung
im Rahmen der Lehrveranstaltung „Integrationsseminar“

Vorgelegt von

Alexander Croll,
Tobias Dietz,
Philipp Koch,
Tim Soetebeer

am 11.07.2014

Fakultät Wirtschaft
Studiengang Wirtschaftsinformatik
Kurs WWI2011I

Inhaltsverzeichnis

Inhaltsverzeichnis.....	I
Abkürzungsverzeichnis	IV
Abbildungsverzeichnis.....	V
Vorwort.....	V
Abstract.....	VI
Einleitung	1
1 Theoretische Grundlagen.....	3
1.1 Einführung in Web-Anwendungen.....	3
1.1.1 Funktionsweise von Web-Anwendungen.....	4
1.1.2 Strukturierung von Webseiten im Browser – HTML.....	5
1.1.3 Übermittlung von Daten – XML.....	6
1.1.4 Design von Webseiten im Browser – CSS.....	6
1.1.5 Das Verhalten von Webseiten erstellen – JavaScript.....	7
1.1.6 Asynchrone Technologien - Ajax.....	7
1.1.7 Dynamische Webseiten.....	8
1.1.8 Das Konzept Model-View-Controller.....	10
1.2 Abgrenzung server-/clientseitiger Web-Anwendungen.....	11
1.3 Abgrenzung mobiler/desktopbasierter Web-Anwendungen sowie nativer mobiler Anwendungen.....	14
1.4 Frameworks zur Anwendungsentwicklung.....	15
1.4.1 Bestandteile eines Entwicklungsframeworks.....	16
1.4.2 Framework-Arten.....	17
1.5 Bewertung marktrelevanter Produkte anhand einer Nutzwertanalyse.....	18
1.5.1 Erarbeitung eines Marktüberblickes.....	18
1.5.2 Die Nutzwertanalyse.....	19
2 Marktüberblick der Open Source Frameworks für Web-Anwendungsentwicklung.....	23
3 Entwurf eines Anforderungskatalogs für die Bewertung von Web-App-Frameworks.....	26
3.1 Aufbau und Nutzung der Nutzwertanalyse.....	26
3.2 Identifikation und Begründung der Anforderungen.....	30
3.2.1 Funktionale Anforderungen.....	30
3.2.1.1 Interface (GUI).....	30
3.2.1.2 Darstellungssteuerung.....	33
3.2.1.3 Geschäftslogik.....	37
3.2.1.4 Datenmodelle.....	38
3.2.1.5 Architektur.....	41

3.2.2	Nichtfunktionale Anforderungen	43
3.2.2.1	Reifegrad	44
3.2.2.2	Verbreitungsgrad.....	45
3.2.2.3	Nutzung	46
4	Exemplarischer Vergleich marktrelevanter Frameworks anhand des Kriterienkatalogs	49
4.1	Bewertung der Frameworks anhand der Kriterien	50
4.1.1	Nutzwertanalyse von Sencha Touch.....	50
4.1.2	Nutzwertanalyse von jQuery und jQuery UI.....	54
4.1.3	Nutzwertanalyse von Backbone.JS und Underscore.JS.....	58
4.2	Auswertung	62
5	Fazit	64
5.1	Zusammenfassung.....	64
5.2	Ausblick	65
	Quellenverzeichnisse	66
	Ehrenwörtliche Erklärung	76

Abkürzungsverzeichnis

Ajax	Asynchronous JavaScript and XML
API	Application Programming Interface
ASP	Active Server Pages
CGI	Common Gateway Interface
CSS	Cascading Style Sheets
DHTML	Dynamic HTML
DOM	Document Object Model
GUI	Graphical User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext-Transfer-Protocol
JS	JavaScript
JSON	JavaScript Object Notation
JSP	Java Server Pages
MVC	Model-View-Controller
MVP	Model-View-Presentator
MVVM	Model-View-ViewModel
PHP	Hypertext Preprocessor
RIA	Rich Internet Application
SSI	Server Side Includes
UI	User Interface
URL	Uniform Resource Locator
XML	Extensible Markup Language

Abbildungsverzeichnis

Abbildung 1: Funktionsweise einer Web-Anwendung	4
Abbildung 2: Ajax-Modell einer Web-Anwendung (asynchrone Datenübertragung)	8
Abbildung 3: Repräsentationen des MVC-Prinzips	11
Abbildung 4: Traditionelle Architektur einer Web-Anwendung.....	12
Abbildung 5: Moderne Architektur einer Web-Anwendung.....	13
Abbildung 6: Kategorien und Kriterien	20
Abbildung 7: Beispielhafte Nutzwertanalyse	21
Abbildung 8: Festlegung zu bewertender Frameworks	26
Abbildung 9: Gewichtung der funktionalen Anforderungen	27
Abbildung 10: Gewichtung der nichtfunktionalen Anforderungen.....	28
Abbildung 11: Fehlermeldung bei inkorrekt vergebener Gewichtung	28
Abbildung 12: Übersicht Bewertungstabelle	29

Vorwort

Die nachfolgenden Ausarbeitungen und Arbeitsergebnisse sind das gemeinschaftliche Werk von Alexander Croll, Tobias Dietz, Philipp Koch und Tim Soetebeer. Die Arbeit ist daher als Gruppenprojekt zu verstehen. Alle Teammitglieder haben zu gleichen Teilen zu den Inhalten beigetragen.

Abstract

Browserbasierte, clientseitige Web-Anwendungen ermöglichen zunehmend einen plattformunabhängigen Zugriff auf komplexe Online-Anwendungen. Grund hierfür ist die steigende Präsenz von innovativen Geschäftsmodellen, welche ihre Dienste den Nutzern online zur Verfügung stellen wollen. Hinzu kommt außerdem die wachsende Notwendigkeit, den Zugriff auf Desktop-Browserapplikationen durch mobile Endgeräte ortsunabhängig zu ermöglichen. Um möglichst vielen Unternehmen die Möglichkeit zu bieten, plattformunabhängige und moderne Clientanwendungen zu schreiben, bieten Frameworks vorgefertigte Gerüste, welche den Entwicklungsaufwand deutlich minimieren sollen. Die Vielfalt an Entwicklungs-Frameworks am Markt bringt eine Funktionsvielfalt mit sich, die in der Regel durch IT-Spezialisten kategorisiert und gefiltert werden muss.

Die vorliegende Arbeit beschäftigt sich daher zunächst mit den Grundlagen clientseitiger, browserbasierter Anwendungen für mobile Endgeräte, um anschließend einen aktuellen Marktüberblick zu liefern. Auf eine Auswahl von über 60 Frameworks werden zunächst KO-Kriterien angewandt, welche sich aus der zugrundeliegenden Projektbeschreibung ergeben. Dies ermöglicht eine detailliertere Betrachtung der verbleibenden Frameworks. Es zeigt sich, dass das betrachtete Feld enorm vielfältig ist und relevante Merkmale häufig nicht deutlich differenzierbar dokumentiert sind. Durch die große Auswahl an Frameworks ist es außerdem notwendig, einen genauen Anforderungsplan der zu entwickelnden Anwendung zu entwerfen, um die Entscheidung für ein entsprechendes Framework zu rechtfertigen. Um mehrere Frameworks und deren Funktionen detailliert zu vergleichen, wird daher eine Nutzwertanalyse entwickelt. Die darin zu vergleichenden Funktionen basieren auf der Ausarbeitung der marktrelevanten Frameworks. Mit Hilfe dieser Analyse können somit funktionale und nichtfunktionale Komponenten von Anwendungsframeworks entsprechend der Anforderungen der Entwickler gewichtet und bewertet werden, um eine Entscheidung zu erleichtern. Anhand eines exemplarischen Vergleichs wird abschließend die Verwendung der Nutzwertanalyse verdeutlicht. Diese Arbeit soll somit Entwicklern dabei helfen, sich im Markt der Frameworks zurecht zu finden sowie eine fundierte Entscheidungsbasis für die Auswahl eines Frameworks ermöglichen.

In diesem Zusammenhang zeigt die vorliegende Ausarbeitung auf, dass der Markt von einer Vielzahl an kleinen, spezialisierten Frameworks geprägt ist, die nicht immer präzise von einer einfachen Bibliothek oder einem Plug-In unterschieden werden können. Dies führt gleichzeitig zu einer enormen Funktionsvielfalt, die bei der Analyse und Bewertung der Frameworks berücksichtigt werden muss. Grundsätzlich ist aber zu sagen, dass durch die offenen Strukturen der meisten Produkte fehlende Funktionen relativ einfach ergänzt werden

können, indem das jeweilige Framework mit einem anderen kombiniert wird. Deutlich wurde im Rahmen der Recherche allerdings auch, dass einige marktdominierende Frameworks wie z.B. jQuery häufig die Grundlage für durch Plug-Ins erweiterbare Frameworks bieten. Diese Kombinationen ermöglichen so eine Spezialisierung und Individualisierung je nach Anwendungsfall.

Die wissenschaftliche Untersuchung wurde im Rahmen der Lehrveranstaltung „Integrationsseminar“ von Studenten im Alter von 21-23 Jahren erstellt, die gemeinsam im 6. Semester das Studienfach „International Management for Business and Information Technology“ an der DHBW Stuttgart studieren. Dies sind die vier Teammitglieder:

Einleitung

Die Entwicklung von leistungsfähigen, drahtlosen Netzwerken führte zu einer Ära der omnipräsenten Zugänge zu Informationen und Services. Durch die zunehmende Unabhängigkeit von Endgerät und Lokation ergibt sich ein wachsendes Interesse daran, innovative Geschäftsmodelle und entsprechende Anwendungen dahingehend auszurichten. Dabei wird schlussendlich ein Paradigmenwechsel in der Software- und Webentwicklung herbeigeführt. Dieser Wandel vollzieht sich vor allem durch die Verschiebung von serverseitigen hin zu clientseitigen Technologien, die Browseranwendungen auf Desktop oder mobilen Endgeräten zur Verfügung stellen.¹ Die entwickelten Anwendungen bieten zusätzlich den Vorteil, dass sie entsprechend der Optik und Haptik klassischer nativer Anwendungen entwickelt werden können, während sie durch die clientseitige Ausführung „teure“ Anfragen über das Netz minimieren.²

Um moderne, plattformunabhängige Clientanwendungen zu schreiben, bieten Frameworks vorgefertigte Gerüste, welche den Entwicklungsaufwand deutlich minimieren sollen. Die enorme Anzahl an Entwicklungs-Frameworks am Markt bringt allerdings auch eine Funktionsvielfalt mit sich, die in der Regel durch IT-Spezialisten kategorisiert und gefiltert werden muss. Da die entsprechenden Frameworks sich zum Teil stark in Ausrichtung, Funktionsumfang, Generalität und Reifegrad unterscheiden, wird eine konsistente Vergleichbarkeit der Frameworks zusätzlich erschwert.

Auf Basis dieser Problemstellung ergibt sich die Zielsetzung der folgenden Ausarbeitung. Ziel ist es, durch eine Studie des Marktes zunächst einen Überblick über die verfügbaren Frameworks zu ermöglichen. Weiterhin wird eine Bewertungsgrundlage in Form einer Excel-basierten Nutzwertanalyse entwickelt. Mit Hilfe der Arbeitsmappe können anschließend funktionale und nichtfunktionale Komponenten von Anwendungs-Frameworks entsprechend der Anforderungen der Entwickler gewichtet und bewertet werden, um eine Entscheidungsfindung zu erleichtern.

Die erläuterte Zielsetzung spiegelt sich im Aufbau dieser Ausarbeitung folgendermaßen wieder: Im Einleitungsteil werden zunächst theoretische Grundlagen der Web-Anwendungsprogrammierung mit besonderem Fokus auf clientseitige Web-Anwendungen erläutert. Unter Anwendung von Ausschluss-Kriterien wird dann eine zuvor erstellte Auswahl an Webframeworks eingegrenzt. Diese Auswahl soll die grundlegenden Eigenschaften zur Entwicklung clientseitiger Anwendungen auf mobilen Endgeräten erfüllen. Weiterhin wird so durch die detaillierte Betrachtung der relevanten Frameworks die Erstellung von Kategorien

¹ Vgl. Kim, Y. / Lee, K. (2009)

² Vgl. Smith, D. / Cearley, D. / Prentice, B. (2013)

mit entsprechenden Features als Basis für eine wiederverwertbare Bewertungsgrundlage ermöglicht. Diese Grundlage wird folglich als Nutzwertanalyse in die Excel-Arbeitsmappe einfließen, deren Bedienung ebenfalls erläutert wird. Im Anschluss wird anhand eines exemplarischen Vergleichs die Verwendung der Arbeitsmappe verdeutlicht, um aufzuzeigen, wie sich die Nutzwertanalyse bei unterschiedlichen Frameworks verhält. Abschließend ist eine Bewertung in Form einer Reflexion der Arbeitsergebnisse vorgesehen. Diese liefert zudem einen Ausblick auf zukünftig zu erwartende Entwicklungen.

1 Theoretische Grundlagen

Die theoretischen Grundlagen sollen sich intensiv mit den für die Arbeit notwendigen Konzepten und Technologien befassen. Es soll vermittelt werden, welche technologischen Aspekte bei Web-Anwendungen eine Rolle spielen. Des Weiteren soll deutlich werden, welche Rolle Frameworks bei der Anwendungsentwicklung spielen und wie sich diese darstellen und funktionieren. Beide Aspekte werden ausführlich behandelt, so dass die theoretische Basis für das Verständnis der weiteren Arbeit gelegt wird.

1.1 Einführung in Web-Anwendungen

Als Einstieg in das Thema ist es entscheidend, zunächst die relevanten Begriffe und Zusammenhänge zu definieren und zu erklären. Das Thema Web-Anwendungen oder auch Web-Applikationen steht dabei im Vordergrund. Daher soll an dieser Stelle eine kurze Definition für das bessere Verständnis dienen.

„Eine Web-Anwendung oder Webapplikation ist ein Computer-Programm, das auf einem Webserver ausgeführt wird, wobei eine Interaktion mit dem Benutzer ausschließlich über einen Webbrowser erfolgt. Hierzu sind der Computer des Benutzers (Client) und der Server über ein Netzwerk, wie das Internet oder über ein Intranet miteinander verbunden, so dass die räumliche Entfernung zwischen Client und Server unerheblich ist.“³

Aus der Definition wird deutlich, dass es sich bei Web-Anwendungen um Applikationen handelt, welche es dem Benutzer ermöglichen, über das Web einen gewissen Dienst aufzurufen. Dabei „dient ein Browser [als Client] und der Server wird durch einen Webserver bereitgestellt.“⁴

Damit wären die grundlegenden Merkmale von Web-Anwendungen definiert. Im nächsten Schritt geht es darum, dass der Aufbau und weitere Konzepte, sowie Merkmale verdeutlicht werden.

³ Aktiv Software GmbH (o.J.), S.1

⁴ Fern-Uni Hagen (2014)

1.1.1 Funktionsweise von Web-Anwendungen

Nachfolgende Grafik soll aufzeigen, wie das grundlegende Konzept von Web-Anwendungen aufgebaut ist und funktioniert.

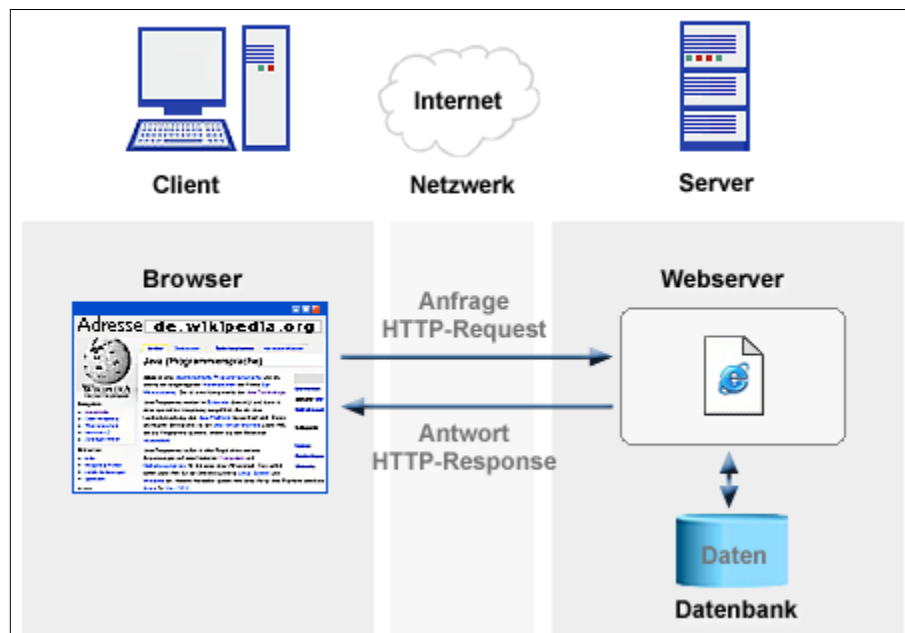


Abbildung 1: Funktionsweise einer Web-Anwendung⁵

Die Grafik verdeutlicht die grundlegende Funktionsweise einer Web-Anwendung. Auf der Client-Seite kommt ein Browser zum Einsatz. Ein Browser wird dabei als ein Computerprogramm bezeichnet, welches als eine Art Benutzeroberfläche oder Graphical User Interface (GUI) dient und so für die Darstellung der Web-Inhalte sorgt.⁶ Der Server auf der anderen Seite wird durch einen Webserver repräsentiert. Dabei handelt es sich um eine Software, bzw. eine Software in Kombination mit der passenden Hardware, welche Dienste, Daten und Inhalte für das Web zur Verfügung stellen.^{7,8} Die Kommunikation zwischen den beiden beteiligten Parteien erfolgt über ein Netzwerk. Dieses muss nicht zwangsläufig das Internet sein, dabei kann es sich ebenso um ein Intra- oder Extranet handeln.⁹ Aus Gründen der Übersichtlichkeit wird im Folgenden aber grundsätzlich vom Internet als verwendetem Netzwerk ausgegangen, da es sich dabei ebenso um die häufigste Variante handelt. Bei der Eingabe einer Internet-Adresse im Browser durch den Nutzer wird diese Uniform Resource Locator (URL) auf dem entsprechenden Web-Server angefragt. Dies erfolgt mittels eines Hypertext-Transfer-Protocol (HTTP)-Request. Für die genaue Funktionsweise wird ein konkretes Beispiel betrachtet. Die verwendete Quelle nutzt als URL

⁵ Fern-Uni Hagen (2014)

⁶ Vgl. Springer Gabler (2011)

⁷ Vgl. Blog-Botfrei (2013)

⁸ Vgl. Wikipedia (2014a)

⁹ Vgl. XI-Interactive

„http://www.howstuffworks.com/web-server.htm“, dabei dient http als Übermittlungsprotokoll an den Webserver, welcher den Namen „www.howstuffworks.com“ trägt. Über den Namen sowie eine unmissverständliche IP-Adresse erfolgt eine Adressierung des Servers. Bei diesem Beispiel ist der angeforderte Filename „web-server.htm“.¹⁰ Diese Anfrage wird von dem Web-Server bearbeitet, dieser erzeugt eine Antwortseite, welche ebenfalls mittels des HTTP-Protokolls übermittelt wird. Die Antwortseite wird im Browser graphisch dargestellt und entspricht dem über die URL adressierten File.¹¹

Im Folgenden soll gezielt auf einige Komponenten eingegangen werden, welche in dem eben beschriebenen Zusammenhang auftauchen. Dabei sollen die für das Verständnis dieser Arbeit relevanten Aspekte berücksichtigt werden. Grundsätzlich können Web-Anwendungen in client- und serverseitige unterschieden werden. Diese Kategorisierung soll allerdings im weiteren Verlauf der Arbeit etwas genauer beschrieben werden. An dieser Stelle ist es lediglich relevant zu wissen, dass es eine solche Unterscheidung gibt und dass der Fokus dieser Ausarbeitung auf den client-seitigen Technologien liegt. Aus diesem Grund wurden in diesem Kapitel vor allem auf Client-Seite relevante Aspekte berücksichtigt.

1.1.2 Strukturierung von Webseiten im Browser – HTML

Wie bereits aus der Funktionsweise von Web-Anwendungen hervorgegangen ist, wird die Anwendung auf der Client-Seite im Browser dargestellt. Diese Darstellung erfolgt über die sogenannte Hypertext Markup Language (HTML). Es handelt sich bei HTML nicht um eine Programmiersprache, sondern lediglich um eine Strukturierungssprache, die dem Browser angibt, wie die Inhalte der Webseite dargestellt werden müssen. Darüber hinaus ist HTML plattformunabhängig, wodurch es nicht relevant ist, welche Hardware bzw. welches Betriebssystem zum Einsatz kommt.¹² Mit Hilfe von HTML-Tags wird die Struktur der Seite vorgegeben. Diese gelten dabei als die zentrale Komponente von HTML, da über die Tags die komplette Struktur der Seite festgelegt wird.¹³

Dabei ist entscheidend zu verstehen, dass seit der Einführung der 4. Version von HTML eine klare Trennung von Struktur und Design einer Webseite vorgenommen wird. Mit Hilfe der Cascading Style Sheets (CSS) kann das Design einer Webseite angepasst werden – ohne, dass der HTML-Code berührt wird. Eine detailliertere Betrachtung von CSS folgt im Anschluss an dieses Unterkapitel. Die nahezu komplette Trennung von Struktur und Design wurde erst durch die Einführung von HTML 5 vollzogen. Das beschreibt gleichzeitig die wichtigste Neuerung durch die Einführung von HTML 5, diese spiegelt sich maßgeblich in der

¹⁰ Vgl. How-Stuff-Works (2012)

¹¹ Vgl. Fern-Uni-Hagen (2014)

¹² Vgl. HTML-Seminar (o.J.)

¹³ Vgl. HTML-Seminar (o.J.)

Einführung einiger neuer Tags und Attribute wider, sowie in der Abschaffung eines Großteils der Styling-Elemente, die nun ausschließlich durch CSS abgedeckt sind.¹⁴

1.1.3 Übermittlung von Daten – XML

An dieser Stelle gilt es noch kurz den oft missverstandenen Unterschied zwischen HTML und der Extensible Markup Language (XML) zu verdeutlichen. Oftmals wird davon ausgegangen, dass XML und HTML ähnliche Teilbereiche behandeln. Dies ist allerdings nicht der Fall, denn XML ist für die Übermittlung von Daten und nicht deren Darstellung zuständig. Des Weiteren gibt es keine vorgeschriebenen Tags, diese können stattdessen selbst definiert werden. Der Hauptunterschied zu HTML liegt allerdings darin, dass XML dazu dient, um Daten zu transportieren und zu speichern.¹⁵ Die wesentliche Information mit Bezug zu XML ist, dass XML selbst nichts tut, denn erst in Verbindung mit einer Software können die in XML geschriebenen Informationen übermittelt werden.

1.1.4 Design von Webseiten im Browser – CSS

Die sogenannten Cascading Style Sheets oder auch CSS ist ebenso wie HTML eine Auszeichnungssprache.¹⁶ Während mit Hilfe von HTML Webseiten-Inhalte strukturiert werden, wird CSS dafür verwendet, diese Inhalte zu formatieren. Es zeigt sich also, dass CSS für das Design einer Webseite verantwortlich ist. Dabei kann CSS bspw. für das Festlegen der Schriftarten, Farben, Rändern, Linien, Höhen, Breiten, Hintergrundbildern und für eine fortgeschrittene Positionierung benutzt werden.¹⁷ CSS ist ebenso plattformunabhängig und kann daher von jedem Browser verwendet werden. Außerdem wird durch CSS die Pflege einer Seite enorm erleichtert, da es nun eine klare Trennung zwischen Struktur und Design gibt. Aus diesem Grund ist es auch möglich, dass bspw. verschiedene Layouts für verschiedene Medientypen erstellt werden. Des Weiteren bieten sich zusätzliche, fortgeschrittene und anspruchsvolle Techniken zur Erstellung eines ansprechenden Designs.¹⁸

¹⁴ Vgl. Züger, D. (2010), S.7 ff.

¹⁵ Vgl. W3-Schools (2014a)

¹⁶ Vgl. Webmasterpro (2014)

¹⁷ Vgl. HTML.net (o.J.)

¹⁸ Vgl. HTML.net (o.J.)

1.1.5 Das Verhalten von Webseiten erstellen – JavaScript

Neben HTML und CSS stellt JavaScript die wichtigste Komponente oder Sprache für Web-Entwickler dar.¹⁹ JavaScript (JS) an sich, welche ebenfalls plattformunabhängig ist, erweitert HTML um zusätzliche Komponenten. Des Weiteren müssen JavaScript-Befehle immer direkt in ein HTML Dokument eingebunden werden. Das bedeutet auch, dass keine besonderen Entwicklungsumgebungen notwendig sind. Wie bei HTML ist ein herkömmlicher Texteditor ausreichend.²⁰ Außer einige Syntaxelemente und der Namensgebung besteht keine direkte Verbindung zwischen Java und JavaScript. Im Gegensatz zu Java kann mit JS keine eigenständige Applikation erstellt werden, da JavaScript immer in Kombination mit HTML stehen muss. Allerdings stellt JS auch eine Möglichkeit für nicht erfahrene Programmierer dar, um grundlegende objektorientierte Komponenten in Webseiten einzubauen.²¹ Der wichtigste Aspekt von JavaScript ist, dass es die beschränkten Möglichkeiten von HTML vor allem mit Bezug auf Animation, Projektsteuerung, dynamische Webseiten, Textgenerierung und Anwendungen erweitert und ergänzt.²² Wie aus verschiedenen Quellen hervorgeht, existieren eine Vielzahl konkreter Anwendungsgebiete für JS²³, welche allerdings aus Gründen der Komplexität an dieser Stelle nicht detaillierter betrachtet werden.

1.1.6 Asynchrone Technologien - Ajax

Basierend auf den vorherigen Unterkapiteln und vorausgreifend auf das nachfolgende Kapitel zu dynamischen Webseiten soll an dieser Stelle auf das Thema Asynchronous JavaScript und XML (Ajax) eingegangen werden. Mit Hilfe von Ajax können grundsätzlich schnelle und dynamische Webseiten erstellt werden. Konkret erlaubt Ajax, dass Webseiten durch das Austauschen lediglich kleiner Datenpakete mit dem Server asynchron aktualisiert werden. Das bedeutet, dass es möglich ist, ein Update auf gewissen Bereichen der Seite durchzuführen, ohne die gesamte Seite neuzuladen.²⁴ Ajax selbst basiert auf grundlegenden Internetstandards wie JavaScript und dem Document Object Model (DOM), ebenso wie CSS und XML. Über Ajax werden Daten direkt im Browser manipuliert und aktualisiert, ohne dass die Seite bei jedem Update neu geladen werden muss. Technisch bedeutet das, dass auf der Client-Seite eine sogenannte Ajax Engine, eine API implementiert ist, welche für die direkte Kommunikation von Daten zur Verfügung steht. Neben der ursprünglich angezeigten Seite ist daher ein weiterer Einstiegspunkt zur Datenübermittlung notwendig. Serverseitig ist die

¹⁹ Vgl. W3-Schools (2014b)

²⁰ Vgl. Grammiweb (2014)

²¹ Vgl. Grammiweb (2014)

²² Vgl. Siegert, Dr. P. F. (o.J.)

²³ Vgl. Wikipedia (2014b)

²⁴ Vgl. W3-Schools (2014c)

eigentliche Programmlogik oder der Prozessfluss implementiert.²⁵ Die nachfolgende Grafik verdeutlicht diesen Zusammenhang. Abschließend sind einige sehr bekannte Beispiele, die Ajax verwenden, aufgeführt. Dazu zählen bspw. Google Maps, Gmail, Youtube und Facebook.

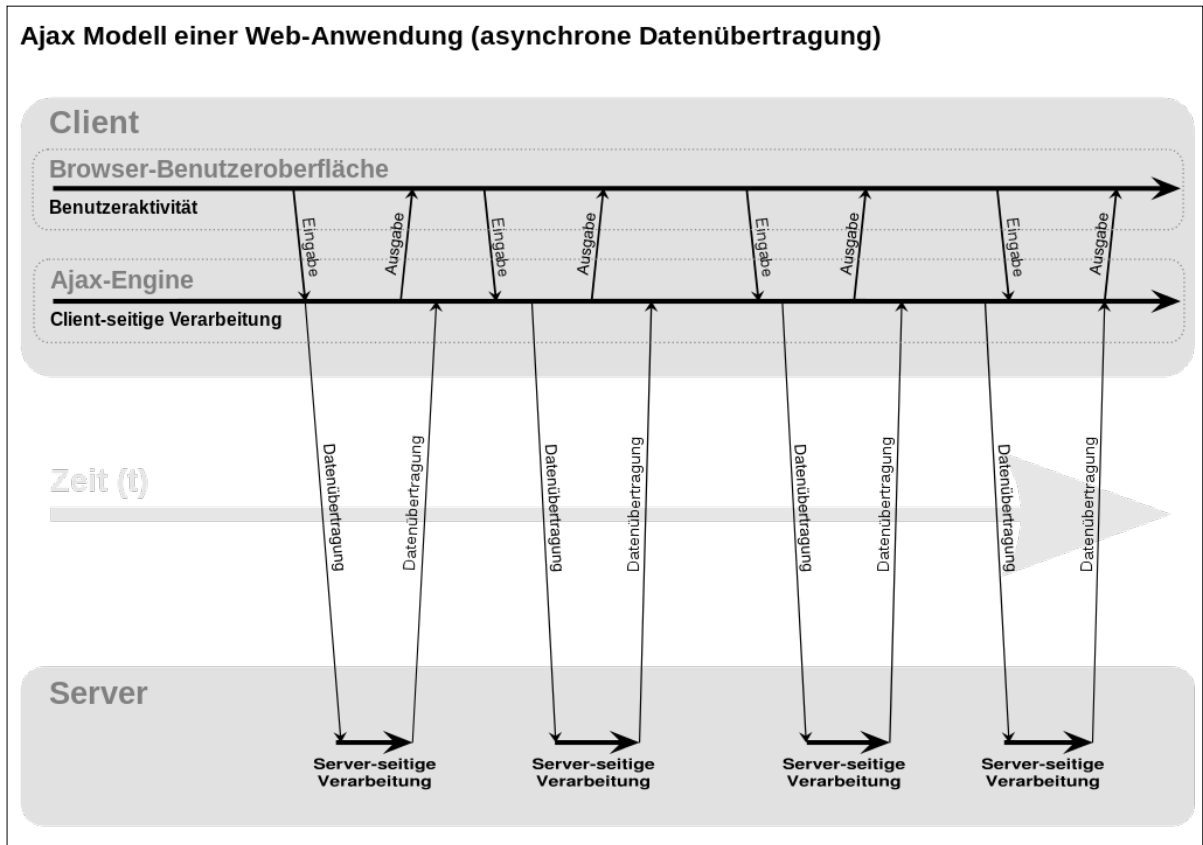


Abbildung 2: Ajax-Modell einer Web-Anwendung (asynchrone Datenübertragung)²⁶

1.1.7 Dynamische Webseiten

Dynamische Webseiten spielen sowohl client- als auch serverseitig eine wichtige Rolle. Daher wurde es an dieser Stelle als eine weitere grundlegende Komponente von Web-Anwendungen aufgenommen.

Grundsätzlich ist der Interaktionsprozess zwischen Client und Server, wie er zuvor beschrieben wurde, ein statischer. Das angeforderte Dokument wird lediglich übermittelt und anschließend wird die Verbindung zum Server geschlossen. Es handelt sich also um eine statische Verbindung, da der Benutzer nicht eingebunden ist und keine wirkliche Interaktion ermöglicht wird.²⁷

²⁵ Wikipedia (2014c)

²⁶ Wikipedia (2014c)

²⁷ Vgl. Metz, C. (2003), S.4

An dieser Problematik schließt sich der Grundgedanke von dynamischen Webseiten an. Dem Client werden vom Server nach wie vor statische HTML-Seiten übermittelt, allerdings erfolgt eine Erstellung der Webseite zur Zeit des Aufrufs dynamisch auf dem Server. Diese Erstellung basiert auf Anforderungen von Seiten des Clients, sowie anderer externer Informationen, bspw. aus einer Datenbank.²⁸ Das bedeutet, dass im Idealfall auf dem Server keine statische Seite, sondern nur noch eine Seitenbeschreibung ohne Inhalt liegt. Demnach wird der Inhalt dynamisch, abhängig von je nach Anwendung verschiedenen Datenquellen, generiert.²⁹

Es existiert eine Vielzahl von Techniken und Ansätzen zur Generierung dynamischer Webseiten. Generell unterscheidet man dabei zwischen dem Common Gateway Interface (CGI) und den Serverpages.³⁰

Über eine CGI-Schnittstelle kann ein beliebiges Programm auf dem Server aufgerufen werden. Dieses muss allerdings ein gültiges HTML-Dokument mit HTTP-Header zurückliefern. Sehr verbreitet sind an dieser Stelle die Skriptsprachen PERL und Python, welche allerdings auch in Form von Serverpages oder als spezielle Webfunktionen verwendet werden können, wodurch die Handhabung wesentlich vereinfacht wird.³¹

Der neben CGI häufiger verwendete Ansatz ist der Einsatz von Serverpages. Dazu zählen vor allem Server Side Includes (SSI), Hypertext Preprocessor (PHP), Active Server Pages (ASP) und Java Server Pages (JSP).³²

Beispielhaft soll hier auf die Verwendung von PHP eingegangen werden. Stand 2013 wurde PHP auf 81% aller Webseiten als serverseitige Programmiersprache verwendet.³³ Damit ist es die am häufigsten verbreitete Sprache zum Erstellen von Webseiten. PHP ist ebenso eine plattformunabhängige Sprache und ist nahezu mit allen Servern und Datenbanken kompatibel.³⁴ Wenn eine Anfrage des Clients auf dem Server eingeht, lädt der Webserver die Datei von der Festplatte und leitet diese an den PHP-Interpreter weiter, welcher wiederum den Code abhängig von der Anfrage dynamisch generiert und an den Client/Browser zurückgesendet. Meist wird ein HTML-Dokument übermittelt – allerdings ist es mit PHP ebenfalls möglich, Bilder, PDFs, Videos etc. zu übermitteln.³⁵ Um eine PHP-Datei ausführen zu können, wird ein System benötigt, welches mit den in der Datei enthaltenen Anweisungen umgehen kann. Dieses System besteht aus einer Schnittstelle mit einem Server-Daemon

²⁸ Vgl. Metz, C. (2003), S.5

²⁹ Vgl. ebenda, S.6

³⁰ Vgl. ebenda, S.9

³¹ Vgl. ebenda, S.10

³² Vgl. ebenda, S.11

³³ Vgl. W3-Techs (2013)

³⁴ Vgl. W3-Schools (2014d)

³⁵ Vgl. Wikipedia (2014d)

oder Server-Dienst und in Kombination mit einem Betriebssystem und einer Datenbank (meist MySQL).³⁶

Oftmals kommt es zur Verwechslung von dynamischen Webseiten und einer dynamischen Seitenerzeugung auf der Client-Seite. Dynamische Webseiten werden durch die bereits beschriebenen Konzepte erklärt. Der Ansatz, Inhalte dynamisch auf Client-Seite zu generieren, erfolgt hauptsächlich über Dynamic HTML (DHTML), wobei HTML um CSS und JavaScript Komponenten erweitert wird. Dabei handelt es sich allerdings nicht um dynamische Webseiten, da ein statischer, immer gleicher Programmcode vom Server an den Client übertragen und dort ausgeführt wird.³⁷ Die Vorteile, die dieser Ansatz mit sich bringt, werden im folgenden Verlauf dieser Arbeit noch näher erläutert. Allerdings können an dieser Stelle bereits die geringeren Raten an Client-Server Kommunikation und somit eine geringere Server-Auslastung und dadurch eine schnellere Ausführung erwähnt werden.³⁸

1.1.8 Das Konzept Model-View-Controller

Das Konzept Model-View-Controller (MVC) wurde ursprünglich für die Entwicklung von Desktopanwendungen ins Leben gerufen, spielt mittlerweile aber auch für die Entwicklung von Web-Anwendungen eine große Rolle.

Bei MVC werden die grundlegenden Bestandteile einer Anwendung voneinander getrennt – sie arbeiten jedoch im Verbund zusammen. Allgemein spricht man hier von den folgenden drei Komponenten:

- Model – die Essenz der Anwendung, bestehend aus Anwendungsdaten, Logik und Geschäftsregeln
- View – die Visualisierungsebene der Anwendung, welche die Informationen entsprechend darstellt
- Controller – Bindeglied zwischen Model und View, welches Input aufnimmt und in Form von Befehlen an den jeweiligen Counterpart weiterleitet³⁹

Für Entwickler macht diese Trennung vor allem dadurch Sinn, dass der gesamte Code nicht nur übersichtlicher wird, sondern auch getrennt entwickelt werden kann. Außerdem kann der Code auch separat getestet werden. Zentraler Begriff ist hier die sogenannte *Separation of Concern*. Die einzelnen Bestandteile können überdies wiederverwendet werden. Während

³⁶ Vgl. Wikipedia (2014d)

³⁷ Vgl. Metz, C. (2003), S.7

³⁸ Vgl. ebenda, S.8

³⁹ Vgl. Osmani, A. (2012)

diese Strukturierung für umfangreiche Anwendungen eine große Hilfe sein kann, ist es bei kleinen Anwendungen unter Umständen ein geringer Mehraufwand.⁴⁰

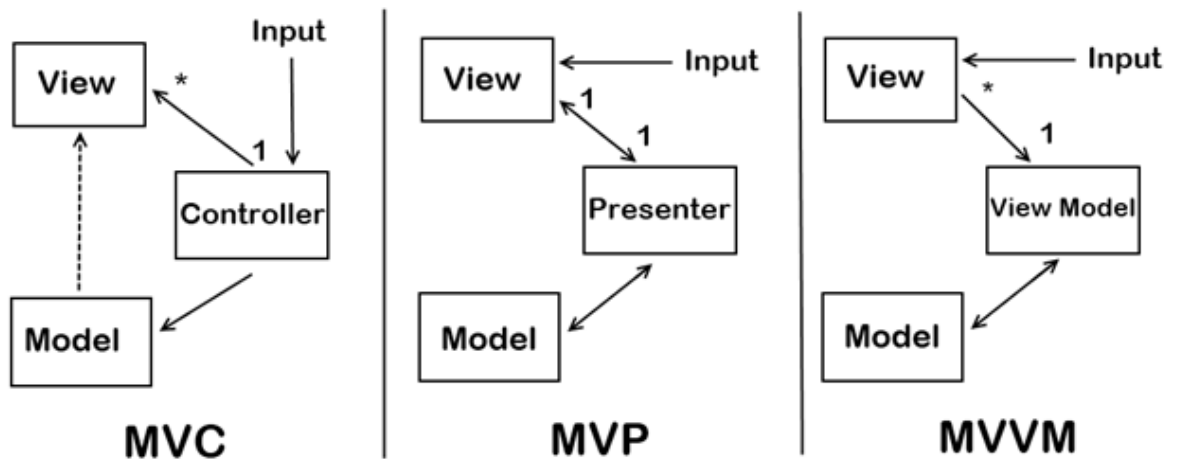


Abbildung 3: Repräsentationen des MVC-Prinzips⁴¹

Es existieren einige Variationen des MVC-Prinzips, von denen sowohl Model-View-Presenter (MVP) als auch Model-View-ViewModel (MVVM) in der obigen Grafik abgebildet sind. Sie unterscheiden sich nur geringfügig im Aufbau und ihrer Logik, die Hauptbestandteile sind dieselben.⁴² Diese Unterschiede ausführlich zu erläutern ist für den weiteren Verlauf der Arbeit nicht notwendig, lediglich ein Gesamtverständnis für den Gebrauch eines MV*-Prinzips ist wichtig. Die Verwendung eines dieser Prinzipien hängt auch vom genutzten Framework für die Entwicklung ab.

1.2 Abgrenzung server-/clientseitiger Web-Anwendungen

Im vorigen Abschnitt wurde das Modell einer Web-Anwendung bereits erläutert und es wurde deutlich, dass bei Web-Anwendungen das Zusammenspiel aus den Komponenten Client (Anwenderseite) bzw. Server (Systemseite) wichtig ist, sich über die letzten Jahre jedoch enorm verändert hat: Der Fokus liegt nun vermehrt auf der Anwenderseite, welche auch in dieser Arbeit in den Mittelpunkt gestellt wird. Um die Hintergründe dieser Neuerung zu verstehen, werden die beiden Anwendungsarchitekturen im Folgenden abgegrenzt.

Die traditionelle Herangehensweise von Web-Applikationen geht davon aus, dass die Serverseite für den Großteil der Operationen zuständig ist und der Client, also der Browser,

⁴⁰ Vgl. Mosers, C. (2011)

⁴¹ Enthalten in: D'Arcy from Winnipeg (2009)

⁴² Vgl. Mosers, C. (2011)

lediglich die Funktion der Visualisierung zu erfüllen hat. Webseiten wurden vom Anwendungsserver an den Client geschickt, wobei HTML den Aufbau, das Aussehen und die Logik in Form von CSS und JavaScript der Seite übermittelt. Bei Interaktion des Anwenders mit der Webseite werden bspw. eingegebene Daten an den Webserver weitergeleitet, welcher diese verarbeitet und darauf basierend JavaScript Object Notation (JSON) Objekte an den Client schickt. Diese werden dazu verwendet, eine neue Seite aufzubauen. Somit ist Dynamik nicht zwingend gegeben, da eine ständige Kommunikation zwischen Client und Server stattfinden muss. Abbildung 4 zeigt die Aufgabenverteilung in einem solchen serverseitigen System:⁴³

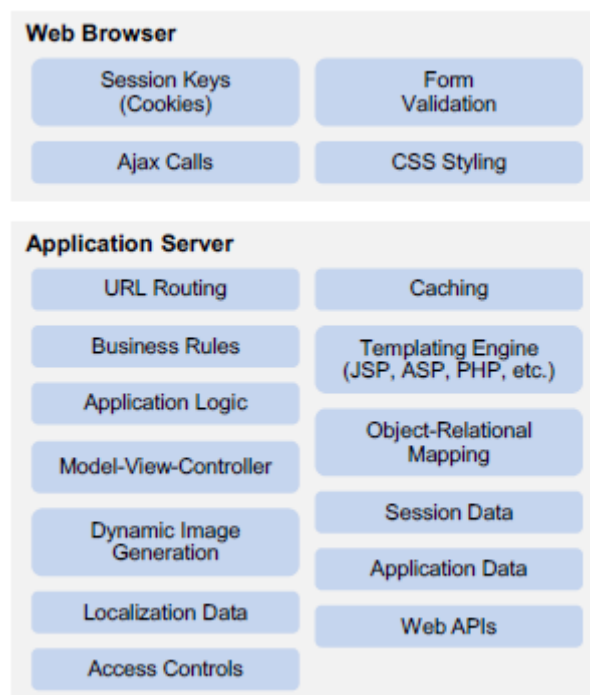


Abbildung 4: Traditionelle Architektur einer Web-Anwendung⁴⁴

Aus dieser Abbildung geht hervor, dass die gesamte Logik einer Web-Anwendung beim serverseitigen Modell auf dem Server liegt. Dort werden das Datenmodell, die Seitentemplates, die Geschäfts- sowie Anwendungslogik und weitere Funktionen gehalten. Auch die Model-View-Controller-Komponente, welche bekanntlich die Teile Datenmodell und Visualisierung logisch trennen soll, befindet sich hier vollständig auf Serverseite.

Diese Herangehensweise hat sich in letzter Zeit gewandelt. Grund hierfür sind unter anderem der rasante Anstieg der Nutzung von modernen mobilen Endgeräten sowie die gesteigerte Leistung von Clients im Vergleich zu früher.

⁴³ Vgl. Wähler, K. (2011)

⁴⁴ Enthalten in: Brian, D./Knoernschild, K. (2013), S.13

Weiterhin ist das Bedürfnis nach höherer Verfügbarkeit ebenfalls ein Grund für die Entwicklung hin zu Web-Anwendungen, die dem Client mehr Aufgaben übergeben. Anwender, insbesondere solche auf mobilen Endgeräten, müssen die Web-Anwendungen auch nutzen können, sollte ihre mobile Datenverbindung einmal kurz unterbrochen werden. Dieses Caching kann durch HTML5-Funktionen erreicht werden. Auch ständiges Nachladen von neuen Seitentemplates bei jeder Benutzer-Interaktion ist auf mobilen Endgeräten besonders lästig, denn es nimmt nicht nur Zeit in Anspruch, sondern geht zudem zu Lasten des beschränkten Datenvolumens.⁴⁵

Die Abbildung 5 zeigt im Vergleich zur Abbildung im vorigen Abschnitt die Verschiebung der Funktionalitäten hin zum Browser und verdeutlicht die moderne Architektur von Web-Anwendungen.

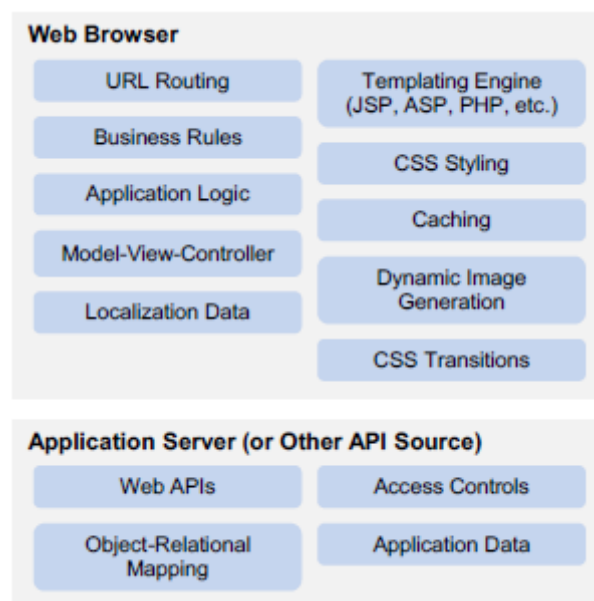


Abbildung 5: Moderne Architektur einer Web-Anwendung⁴⁶

Es ist deutlich zu erkennen, dass die Anwendungsserver schlanker sind und in den meisten Fällen keine Engines zur PHP oder JSP Template-Erstellung mehr besitzen. Auf dem Server liegt dann lediglich noch eine Web Application Programming Interface (API) vor, welche den notwendigen Code an den Client liefert. Die gesamte Logik, Prozesse und Darstellungen werden in diesem Fall vom Browser übernommen.

Die Entwicklung neuer, moderner Web-Applikationen bringt einige Vorteile mit sich, welche u.a. bereits aus dem obigen Abschnitt hervorgingen, hier jedoch noch einmal übersichtlich aufgeführt werden:

⁴⁵ Vgl. Brian, D./Knoernschild, K. (2013), S.24

⁴⁶ Enthalten in: Brian, D./Knoernschild, K. (2013), S.24 ff.

- Bessere und ansprechendere User Interface (UI) Designs
- Umfangreichere Anwendungsfälle für die entsprechenden Web-Anwendungen
- Getrennte und fokussierte Entwicklung durch Trennung von Server und Client
- Schnelligkeit und Verfügbarkeit

Auf der anderen Seite sollten jedoch auch negative Aspekte nicht außer Acht gelassen, wie beispielweise der Bedarf eines aktuellen und unterstützten Browsers.

1.3 Abgrenzung mobiler/desktopbasierter Web-Anwendungen sowie nativer mobiler Anwendungen

Zusätzlich zum beschriebenen Fokus auf Web-Anwendungen, welche clientseitig ausgeführt werden, soll in dieser Arbeit außerdem ein besonderes Augenmerk auf solche Web-Anwendungen gelegt werden, die in den Browsern mobiler Endgeräte laufen. Hierzu ist es notwendig, Unterschiede zwischen mobilen sowie desktopbasierten Anwendungen herauszustellen.

Da in diesem Zusammenhang häufig die Begriffe mobile sowie native Anwendung fallen, ist es wichtig zu verstehen, wie diese zusammenspielen bzw. sich voneinander abgrenzen.

Bei nativen, mobilen Anwendungen spricht man von den allseits bekannten, sogenannten Apps, die sich als Anwendung aus einem App-Store auf ein mobiles Endgerät herunterladen lassen. Es wird von nativen Anwendungen gesprochen, da sie oft in der für das Gerät/Betriebssystem optimierten Programmiersprache geschrieben sind und außerdem auf native Funktionen des Geräts zugreifen können (z.B. Kamerafunktion, GPS, etc.).

Im Unterschied zu webbasierten Anwendungen, welche plattformübergreifend aufgerufen werden können, sind native mobile Anwendungen also nur für ein Gerät optimiert. Web-Anwendungen können aus jedem Browser heraus aufgerufen werden. Bei solchen Web-Anwendungen, die auf mobilen Endgeräten aufgerufen werden, spricht man u.a. auch davon, dass sie ein natives „Look and Feel“ haben. Dies bedeutet, dass die Anwendung auf die eben beschriebenen nativen Funktionen durchaus zugreifen kann und die Anwendung an sich ebenfalls Eigenschaften aufweist, die typisch für ein Gerät/Betriebssystem sind. Dies können zum Beispiel Design- oder Bedienelemente sein. In diesem Zusammenhang werden die Web-Anwendungen auch als hybrid bezeichnet, da sie eine Kombination aus Web-Anwendungen und einigen Spezifikationen von nativen mobilen Anwendungen darstellen. Diese hybriden Web-Anwendungen sind natürlich vorwiegend auf mobile Geräte ausgelegt, jedoch gibt es noch weitere Unterschiede zwischen Web-Anwendungen, die für

Desktopgeräte oder mobile Geräte entwickelt wurden. Diese sollen nun noch einmal genauer betrachtet werden.⁴⁷

Zunächst bestehen Unterschiede im Design der Web-Apps, denn während Anwendungen für den Desktop einen großen Bildschirm voraussetzen, sind mobile Web-Anwendungen in ihrem Platz beschränkt, müssen sich jedoch an die verschiedenen Displaygrößen der einzelnen mobilen Endgeräte anpassen können. Außerdem gibt es weiterhin noch andere Komponenten wie beispielweise den Inhalt oder auch das Verhalten einer Web-Anwendung, welche von der Art des Endgeräts abhängen. Es gibt verschiedene Herangehensweisen, um diese Probleme zu lösen – sowohl serverseitig als auch clientseitig. Aufgrund des Fokus dieser Arbeit werden insbesondere die clientseitigen Möglichkeiten betrachtet. Der wichtigste Begriff in diesem Zusammenhang ist das *responsive Design*. Dies bedeutet, dass eine Web-Anwendung auf äußere, hardware-/softwarebedingte Umstände reagieren kann und sich dementsprechend strukturiert und aufbaut. Für clientseitige Anwendungen heißt das, dass alle Clients vom Server dasselbe initiale Dokument mit CSS, JavaScript und HTML Informationen erhalten und es für unterschiedliche Bildschirme verschieden verwenden. Möglich wird dies mit Hilfe sogenannter *media queries*, welche die Geräteart bzw. die Größe des Displays abfragen, wodurch dann der Code clientseitig im Browser entsprechend generiert wird. Dadurch können Umstände wie Inhalte (Text und Bilder), Layout oder die Ausrichtung (Bildschirmorientierung) angepasst werden.⁴⁸

Weiterhin ist es wichtig, dass nicht nur das Design angepasst wird, sondern die Anwendung ebenfalls weiß, auf welche Funktionen sie zurückgreifen kann. Hierzu zählen die bereits angesprochenen nativen Funktionen eines Betriebssystems bzw. Geräts. Die Web-Anwendung muss also genauso bemerken, ob sie auf einem Blackberry (ohne Touchscreen) oder einem iPhone (mit Touchscreen) ausgeführt wird, damit dem Blackberry nicht eine Touch-Navigation zur Verfügung gestellt wird, welche der Anwender letztlich nicht nutzen kann.⁴⁹

1.4 Frameworks zur Anwendungsentwicklung

Wie eingangs bereits erwähnt, ermöglichen browserbasierte, clientseitige Web-Anwendungen einen weitreichenden plattformunabhängigen Zugriff auf Web-Anwendungen. Diese Software-Systeme können mit der Hilfe von Frameworks entwickelt werden.

⁴⁷ Vgl. Gagern, S. von (2013)

⁴⁸ Vgl. Brian, D./Knoernschild, K. (2013), S.17 ff.

⁴⁹ Vgl. Microsoft.com (2012d)

Bei einem Framework handelt es sich grundlegend um ein „Gerüst“ an vorgefertigten Elementen und Funktionen. Dabei zielen Frameworks für browserbasierte Anwendungen darauf ab, Entwicklern das Erstellen solcher Anwendungen zu vereinfachen und Zeit zu sparen.⁵⁰ Vorgefertigte und teilweise modular verwendbare Komponenten eines Frameworks führen letztendlich zu einem wiederverwertbaren Design.⁵¹

Unter Berücksichtigung der hohen Anzahl an unterschiedlichen Endgeräten, die auf eine Web-Anwendung zugreifen, muss bei der Auswahl besonders großer Wert auf Frameworks gelegt werden, welche alle relevanten Teilnehmer bei gleichzeitiger Beachtung der Leistungsausschöpfung der Technologien bedienen. Dies gilt nicht nur für den privaten, sondern auch und vor allem für den geschäftlichen und öffentlichen Sektor.⁵²

Durch die wachsende Präsenz von neuen Geschäftsmodellen in Verbindung mit Informationstechnologien kann ein Framework ein sinnvolles Gerüst für Unternehmen darstellen, um Anwendungen schnell, plattformunabhängig und in Form von mobilen Endgeräten auch ortsunabhängig zugänglich zu machen.⁵³

1.4.1 Bestandteile eines Entwicklungsframeworks

Für die Replikation und Synchronisation von Web-Anwendungen wird besondere Flexibilität von Frameworks und einzelnen Komponenten verlangt.⁵⁴ Moderne Frameworks für Web-Anwendungen implementieren daher normalerweise einen Datenbankzugriff, Vorlage-Mechanismen, Model-View-Controllers und Möglichkeiten zum Rapid Prototyping durch Scaffolding.⁵⁵ Um in kurzer Zeit lauffähige Anwendungen zu erarbeiten, enthalten Frameworks hierbei unterschiedliche Basisfunktionen, welche die Grundlage für Web-Anwendungen bilden. Hierzu gehören: Bibliotheken, die Laufzeitumgebung und Funktionen wie Session-Verwaltung, Daten-Persistenz und Vorlagen.⁵⁶

An dieser Stelle ist allerdings nochmal die Unterscheidung zwischen client- und serverseitigen Webapplikationen in Erinnerung zu rufen. Die oben aufgeführten Inhalte eines Frameworks unterscheiden sich bei der Betrachtung von Frameworks für clientseitige Webapplikationen insofern, als hierbei lediglich die Geschäftslogik durch das Framework serverseitig zur Verfügung gestellt wird und die Datenhaltung, Darstellung und Verarbeitung clientseitig durchgeführt wird. Dies führt dazu, dass clientseitige Frameworks, die auf das

⁵⁰ Vgl. docforge.com (o.J)

⁵¹ Vgl. Hansen, R. Neumann, G. (2002), S. 157

⁵² Vgl. Stavroulaki, V. u.a. (2006), S.889 ff.

⁵³ Vgl. Lin, F. / Shaw, M. J. / Chuang M. Y. (2005), S. 299 ff.

⁵⁴ Vgl. Kouki, J. / Chainbi, W. / Ghedira, K. (2014), S. 1 ff.

⁵⁵ Vgl. Brian, D. / Knoernschild, K. (2013), S. 21 ff.

⁵⁶ Vgl. docforge.com (o.J)

Erstellen von Anwendungen ausgerichtet sind, die Dateien in die jeweiligen Vorlagen laden, die nicht auf serverseitigen Technologien und Skriptsprachen (wie z.B. PHP) basieren. Bei der Umsetzung von clientseitigen Web-Anwendungen werden somit ausschließlich die Inhalte, die vom Server kommen, angezeigt und manipuliert. Eine dauerhafte Interaktion mit einem Server für Datenhaltung und -austausch ist nicht vorgesehen oder zwingend, da dies dem Vorteil von reduzierter Bandbreite bei clientseitigen Anwendungen entgegenwirken würde. Es wird folglich kein Datenbankzugriff implementiert. Die Verwendung von lokalem Speicher bzw. WebSQL Datenbanken ist jedoch weiterhin möglich.⁵⁷

Die Auswahl eines Frameworks unterscheidet sich daher neben den Funktionen auch durch die verwendete Programmiersprache je nach Anwendungsbereich (im Rahmen dieser Arbeit werden lediglich HTML5, JavaScript und Ajax Frameworks betrachtet). Frameworks basierend auf Skriptsprachen wie PHP schließen sich für die Betrachtung von Frameworks für clientseitige Web-Applikationen daher von vornherein aus.

1.4.2 Framework-Arten

Die hohe Anzahl an unterschiedlichen Frameworks führt dazu, dass am Markt unterschiedliche Kombinationen von Funktionen innerhalb der Systeme verfügbar sind. Grob unterscheiden sich Anwendungs- und Komponentenframeworks.

Anwendungsframeworks verwenden Softwareframeworks, um die Standardstrukturen von Anwendungen zu entwickeln. So können beispielsweise grafische Benutzeroberflächen durch Vorlagen aus den Bibliotheken der Software kreiert werden.⁵⁸

Komponentenframeworks implementieren vorgefertigte Funktionen. Dies bedeutet, dass einige Funktionen, die vordefinierte Aufgaben erfüllen können, in Komponenten integriert werden und so direkt zur Verfügung stehen.

Eine detailliertere Kategorisierung der Frameworks findet in Kapitel 3.2 statt und wird auf einer Gruppierung der tatsächlich am Markt verfügbaren Funktionen basieren. Daher sollte an dieser Stelle lediglich eine Einführung in die Grundidee der Frameworks gegeben werden.

⁵⁷ Vgl. Stackoverflow (2013)

⁵⁸ Vgl. docforge.com (o.J)

1.5 Bewertung marktrelevanter Produkte anhand einer Nutzwertanalyse

1.5.1 Erarbeitung eines Marktüberblickes

Ob für die Entwicklung einer Geschäftsidee, bei der Positionierung eines Unternehmens oder zur Auswahl eines bestimmten Produktes aus einer Vielzahl von Angeboten – in jedem Fall ist ein fundiertes Wissen über die sich am jeweiligen Markt befindlichen Akteure von großer Bedeutung. So ist es einerseits für die Platzierung eines eigenen Produktes unausweichlich, die Fähigkeiten und Eigenschaften der von der Konkurrenz angebotenen Güter zu kennen, um sich mit Hilfe von Alleinstellungsmerkmalen von diesen zu differenzieren. Andererseits trägt die tiefgreifende Kenntnis über Marktteilnehmer dazu bei, sich im Rahmen des Einkaufes eines Produktes für ein bestimmtes Angebot zu entscheiden. Speziell im Kontext der zunehmenden Marktsättigung in vielen Bereichen ist es heutzutage von großer Bedeutung, systematisch bei der Auswahl eines Produktes aus der häufig großen Menge an Substitutionsgütern vorzugehen. Gleichmaßen verhält sich dies bei der Auswahl von Software, welche beispielsweise in einem Unternehmen eingesetzt werden soll. Damit aus dem „Dschungel“ an verschiedener Software mit dem gleichen oder einem ähnlichen Nutzen letztlich das für die Unternehmensanforderungen optimale Produkt gewählt werden kann, muss zunächst eine Struktur und Ordnung in den Gesamtmarkt gebracht werden. Der daraus resultierende Marktüberblick stellt die Voraussetzung für die Nutzwertanalyse dar, bei der nur ein Teil der angebotenen Produkte miteinander verglichen wird. Zur Eingrenzung der marktrelevanten Produkte ist es zunächst sinnvoll, die grundlegenden Vorgaben des Unternehmens klar zu definieren. Beispielsweise könnte wie im Falle dieser Ausarbeitung das Ziel sein, mit Hilfe der Software mobile, browserbasierte Anwendungen zu erstellen. Somit wird die ursprüngliche Auswahl von Frameworks zur Anwendungsentwicklung dadurch weiter eingegrenzt, dass die zu entwickelnden Anwendungen von mobiler, browserbasierter Natur sind. Überdies lassen sich zusätzliche „K.O.-Kriterien“ definieren, die die relevanten Produkte weiter reduzieren. In Bezug auf die vorliegende Arbeit müssen die Frameworks folglich einerseits clientseitig sein, wodurch wiederum die serverseitige Software aus der Betrachtung verschwindet – andererseits muss es sich um Open Source Software handeln, bei der der Quellcode frei verfügbar, zugänglich und editierbar ist. Auf diese Weise ist es möglich, sich Schritt für Schritt einer immer kleiner und damit übersichtlicher werdenden Eingrenzung der zu betrachtenden Produkte zu nähern. Für die gewählten Produkte lässt sich zudem der jeweilige Marktanteil berechnen, welcher einen Überblick darüber gibt, wie stark ein Unternehmen auf dem Markt vertreten ist und welchen Anteil des gesamten Umsatzes es mit seinem Produkt generieren konnte. Dieser Marktanteil kann entsprechend berechnet werden, indem das Absatzvolumen, also der tatsächlich erzielte Umsatz eines Unternehmens, durch das Marktvolumen, also den tatsächlichen Umsatz des Gesamtmarktes, geteilt wird.

Zusammenfassend stellt der Marktüberblick ein Werkzeug zur ersten Eingrenzung des Marktes auf mögliche Produkte dar, während die darauf folgende Nutzwertanalyse die ausgewählten Optionen auf ihre Funktionalitäten untersucht und sie bezüglich der Eignung für den Unternehmenseinsatz bewertet.

1.5.2 Die Nutzwertanalyse

Nachdem in Form des Marktüberblickes dem Gesamtmarkt eine gewisse Struktur verliehen wurde und die marktrelevanten Produkte erarbeitet werden konnten, dient die im nächsten Schritt durchzuführende Nutzwertanalyse dazu, die möglichen Alternativen miteinander zu vergleichen. Anhand einer auf den jeweiligen Anwendungsfall abgestimmten Bewertung kann somit die Auswahl für eine der Optionen vereinfacht werden.

Um im Folgenden mit einer einheitlichen Begrifflichkeit arbeiten zu können, soll der Begriff der Nutzwertanalyse zunächst definiert werden.

„Die Nutzwertanalyse ist ein nicht monetäres Bewertungsverfahren aus dem Bereich der Kostenrechnung. Mit ihrer Hilfe sollen nicht-monetäre Teilziele vergleichbar gemacht werden, um so eine Entscheidung zwischen mehreren Alternativen treffen zu können.“⁵⁹

Entscheidend bei der Nutzwertanalyse ist hier vordergründig, dass die Alternativen auch anhand solcher qualitativer Bewertungskriterien gemessen werden können, die nicht in Geldeinheiten ausdrückbar sind.⁶⁰ Dadurch können auch z.B. technische, psychologische und soziale Faktoren berücksichtigt werden.⁶¹ Anwendbar ist dieses Verfahren somit nicht nur auf die Auswahl verschiedener Produkte und Lieferanten, sondern auch auf andere Investitionsvorhaben wie die Entscheidung für einen neuen Unternehmensstandort.⁶² Als Ergebnis der Nutzwertanalyse wird für alle in Frage kommenden Optionen ein Punktwert ermittelt, der eine direkte Vergleichbarkeit der Alternativen ermöglicht.⁶³

Für die Erstellung einer aussagekräftigen und objektiven Nutzwertanalyse bedarf es eines Vorgehens, welches wie folgt aufgebaut ist:⁶⁴

1. **Formulierung geeigneter Bewertungskriterien:** Sämtliche zielorientierte Auswahlkriterien werden üblicherweise in einem Kriterienkatalog oder Lastenheft zusammengefasst, bei dem es sich um ein Werkzeug zur Beurteilung verschiedener Systeme handelt. Eine Auswahl an möglichen Kriterien ist in Abbildung 6 enthalten.

⁵⁹ Controlling-Portal (2014)

⁶⁰ Vgl. Gabler Wirtschaftslexikon (o.J.)

⁶¹ Vgl. ebenda

⁶² Vgl. Welt der BWL (o.J.)

⁶³ Vgl. manager wiki (2013)

⁶⁴ Vgl. manager wiki (2013)

Diese gliedern sich hauptsächlich in die Kategorien Kommerzielle Aspekte, Qualität, Logistik, Technologie, Risiko sowie Arbeitssicherheit und Umweltschutz. Je nach betrieblichen Anforderungen und Kontext der Produktauswahl werden einige dieser oder andere Aspekte in den Kriterienkatalog übernommen und gegebenenfalls weiter spezifiziert oder angeglichen:



Abbildung 6: Kategorien und Kriterien⁶⁵

2. **Bestimmung der Bedeutung eines Auswahlkriteriums:** Dieser Schritt umfasst die Gewichtung der Kriterien anhand der Ziele und Anforderungen der Analyse
3. **Bewertung der Optionen mit Hilfe von Punktwerten:** Einschätzung der Erfüllung des jeweiligen Kriteriums durch ein zuvor festgelegtes Bewertungsschema, z.B. eine 1 für „schlecht“, eine 10 für „gut“
4. **Ermittlung des einzelnen Nutzwertes:** Multiplikation der Gewichtung des Kriteriums mit dem vergebenen Punktwert

⁶⁵ Enthalten in: RWE Einkauf (o.J.)

5. **Aufsummierung der einzelnen Nutzwerte:** Die Summe aller Multiplikationen ergeben das Endergebnis je Alternative – die Option mit dem höchsten Nutzwert stellt die sinnvollste Alternative dar
6. **Durchführung einer Sensitivitätsanalyse:** Überprüfung der Robustheit mit Hilfe der sinnvollen Veränderung der Gewichtung der Kriterien

Beispielhaft wird in der folgenden Abbildung 7 eine Nutzwertanalyse für ein Unternehmen aufgezeigt, welches eine weitere Filiale in einem anderen Bundesland eröffnen will. Es kommen dabei die drei Bundesländer A, B und C in Frage, wobei diese auf die Oberkriterien Standort, Kosten und Gewinnpotenziale bewertet werden. In diesem Fall erreicht das Bundesland A mit einer Punktzahl von 579 den höchsten Nutzwert. Grund dafür ist, dass dieses Bundesland vor allem bei den Gewinnpotenzialen, also den Kriterien mit der höchsten Gewichtung, am besten bewertet wurde:

Kriterien	Gewichtung	A	A*G	B	B*G	C	C*G
Standort							
Hauptstraße	7	5	35	7	49	3	21
Nähe zu anderen Geschäften	9	5	45	3	27	10	90
Lagernähe	8	10	80	8	64	6	48
Kosten							
steuerliche Belastung	6	4	24	5	30	5	30
Bau- /Mietkosten	15	6	90	7	105	4	60
Werbung	10	7	70	4	40	8	80
Gewinnpotenziale							
wenig Konkurrenz	15	3	45	4	60	3	45
Kunden sind offen für Neues	10	5	50	5	50	6	60
Arbeitslosenquote gering	20	7	140	5	100	6	120
Gesamt	100		579		525		554

Abbildung 7: Beispielhafte Nutzwertanalyse⁶⁶

Die Nutzwertanalyse bietet vor allem aufgrund ihrer Flexibilität entscheidende Vorteile. Die Methodik lässt sich an eine große Anzahl an speziellen Erfordernissen anpassen.⁶⁷ Konkret lassen sich verschiedene Anwendungsszenarien und Einsatzbereiche modellieren. Überdies berücksichtigt diese Form der Analyse eine Vielzahl von möglichen Kriterien, die beliebig durch den Anwender festgelegt werden können. Ebenso handelt es sich bei der Nutzwertanalyse häufig um das einzig anwendbare Hilfsmittel zur Analyse einer Entscheidungssituation, wenn eine gewisse Zielvielfalt zu beachten ist und bzw. oder ein monetärer Wert nicht primär bestimmt werden kann.⁶⁸ Letztendlich stellt auch die Vergleichbarkeit einen deutlichen Vorteil dar – besonders relevant wird dies, weil durch die

⁶⁶ Enthalten in: Controlling-Portal (2014)

⁶⁷ Vgl. 4managers (o.J.)

⁶⁸ Vgl. Gabler Wirtschaftslexikon (o.J.)

Auswahl gemeinsamer Kriterien Alternativen vergleichbar gemacht werden, die zunächst unvergleichbar erscheinen.⁶⁹

Gleichzeitig handelt es sich bei der Vergleichbarkeit jedoch auch um einen möglichen Nachteil, da nicht immer gewährleistet werden kann, dass zwei Alternativen in derselben Hinsicht miteinander verglichen werden können.⁷⁰ Auch wenn die Nutzwertanalyse bereits dem Ziel der Objektivität näher kommt, handelt es sich bei der Bewertung und Gewichtung der einzelnen Kriterien noch immer um eine subjektive Einschätzung. Zudem muss die Eignung dieser Methode vor allem dann in Frage gestellt werden, wenn mehrere Entscheidungsträger mit unterschiedlichen Präferenzen an der Durchführung der Beurteilung beteiligt sind.⁷¹ Zudem muss hinterfragt werden, ob die Wertsynthese der Teilnutzwerte zu einem Gesamtnutzwert praktikabel ist, da vorausgesetzt wird, dass die Teilnutzen kardinal messbar und die Zielkriterien im Hinblick auf ihren Nutzen voneinander unabhängig sind.⁷²

Alles in allem stellt die Nutzwertanalyse ein hilfreiches Werkzeug zur Bewertung verschiedener Handlungsalternativen dar, welches jedoch im Rahmen der Durchführung hinsichtlich ihrer Objektivität und der Vergleichbarkeit immer auch kritisch hinterfragt und überprüft werden muss.

⁶⁹ Vgl. Gabler Wirtschaftslexikon (o.J.)

⁷⁰ Vgl. ebenda

⁷¹ Vgl. 4managers (o.J.)

⁷² Vgl. Gabler Wirtschaftslexikon (o.J.)

2 Marktüberblick der Open Source Frameworks für Web-Anwendungsentwicklung

Das Ziel dieses Kapitels ist es, die zuvor in Kapitel 0 gewonnenen Grundlagen zur Erarbeitung eines Marktüberblickes auf die im Rahmen dieser Ausarbeitung relevanten clientseitigen Open Source Frameworks zum Bau mobiler, browserbasierter Web-Anwendungen anzuwenden. Dieser Schritt stellt eine wichtige Voraussetzung zur Erarbeitung des Kriterienkataloges für die Bewertung dieser Frameworks dar.

In Hinblick auf das Vorgehen in dieser Arbeit wird ein iterativer Ansatz verwendet, welcher die Reihenfolge des Marktüberblickes und des Kriterienkataloges miteinander kombiniert und entsprechend vermischt. Um sich einen Überblick über die Vielzahl der vorhandenen Produkte zu verschaffen, werden zunächst einige K.O.-Kriterien festgelegt, anhand derer sämtliche Produkte untersucht und eingegrenzt werden. Diese Auswahl an Produkten wird dann in Bezug auf ihre Features analysiert, um aus den relevantesten den Kriterienkatalog zu erstellen. Dadurch lässt sich vor allem die benötigte Objektivität sicherstellen.

Im ersten Schritt wird zunächst eine Auswahl der in Frage kommenden Frameworks zum Bau von Web-Anwendungen gesammelt. Dies findet vor allem auf Basis der bekannten Internet-Portale wie www.ohloh.net als Suchmaschine für Open Source Software und www.todomvc.com als Sammlung von Model-View-Frameworks. Ziel ist es, zunächst eine ungefilterte Erfassung der möglicherweise für eine konkretere Betrachtung relevanten Frameworks durchzuführen. Im Abgleich dieser mit weiteren Portalen ergibt sich somit eine Liste von 67 Frameworks, mit Hilfe derer sich mobile Web-Anwendungen entwickeln lassen. Als erste Eingrenzung muss nun überprüft werden, ob diese 67 Frameworks tatsächlich die Mindestanforderungen erfüllen, welche sich wie folgt aus der Aufgabenstellung ergeben:

- Open Source Lizenzen
- Clientseitige Skriptsprache
- Mobile Anwendungen

Anhand dieser Kriterien können bereits 14 Produkte aus der Betrachtung ausgeschlossen werden, da diese mindestens eins der K.O.-Kriterien nicht erfüllen. Somit bleiben noch 53 Frameworks übrig, welche in der folgenden Tabelle aufgelistet werden. Zur weiteren Eingrenzung der marktrelevanten Produkte wird nun als weiteres, sehr wichtiges Kriterium der Reifegrad festgelegt. Dieser gibt Aufschluss darüber, inwiefern das Framework laufend weiterentwickelt und verbessert wird und wie viele Entwickler an dieser Entwicklung beteiligt sind. Somit stellt dieses Kriterium eine wichtige Voraussetzung für die Einführung eines

Frameworks in einem Unternehmen dar. Als Maß werden zur Bewertung einerseits die Angaben auf GitHub verwendet, welches ein webbasierter Hosting-Dienst für die Durchführung von Software-Entwicklungsprojekten ist. Dieses zeigt auf, wie viele „Contributors“, also Beteiligte an der Weiterentwicklung des Frameworks teilhaben und wie viele „Commits“, also Freischaltungen von Änderungen getätigt wurden. Andererseits ist es für die Eingrenzung von entscheidender Bedeutung, in welchen Abständen neue Updates zur Verfügung gestellt haben und in welchem Status in Form der Version sich das Produkt derzeit befindet.

Auf Basis dieser Informationen werden folglich aus der nachfolgenden Tabelle die Frameworks ausgewählt und farbig markiert, die für eine detailliertere Analyse in Bezug auf die Features und Eigenschaften in Betracht gezogen werden:

<u>Relevante Frameworks</u>	<u>Irrelevante Frameworks</u>
Sencha Touch	Dojo
Qooxdoo	MooTools
AngularJS	Wink
YUI	Jo
SproutCore	jQ Touch
jQuery Mobile	DHTMLX Touch
Enyo	SAP5UI
KendoUI mobile	JQmobi
M-Project	KnockbackJS
KnockoutJS	LungoJS
BatmanJS	Titanium
DeftJS	CanJS
ZeptoJS	Junior
Montage	PureMVC
eMobic	CujoJS
Spine	Google Closure JS
Bootstrap	Raphael
BackboneJS	SammyJS
UnderscoreJS	Bikini
EmberJS	SomaJS

HandlebarsJS	jQuery UI
Google Web Kit	Olives
Vanilla JS	PlastronJS
	rAppidJS
	Exoskeleton
	AtmaJS
	ComponentJS
	VueJS
	SerenadeJS
	Derby

Tabelle 1: Überblick über Frameworks

Im nächsten Schritt gilt es nun, diese 23 marktrelevanten Frameworks hinsichtlich ihrer Features und Eigenschaften zu untersuchen und daraus einen Kriterienkatalog abzuleiten, mit dem sich eine auf die jeweiligen Anforderungen abgestimmte Bewertung von ausgewählten Produkten durchführen lässt.

3 Entwurf eines Anforderungskatalogs für die Bewertung von Web-App-Frameworks

3.1 Aufbau und Nutzung der Nutzwertanalyse

Auf Basis der theoretischen Einführung in die für die vorliegende Ausarbeitung relevante Thematik wird im nachfolgenden Kapitel der Kriterienkatalog in Form einer Nutzwertanalyse entwickelt, welcher durch die Autoren der Arbeit erarbeitet wird und entsprechend den eigenen Wertbeitrag darstellt. Die Nutzwertanalyse ist in einer Excel-Datei enthalten und wird zusammen mit der Seminararbeit abgegeben. Dadurch ist diese beliebig anpass- und erweiterbar und kann je nach Szenario und Präferenzen direkt im Unternehmen eingesetzt werden, um verschiedene Alternativen zu bewerten. Es wird zunächst der Aufbau sowie die Nutzung der Nutzwertanalyse beschrieben, um den Anwender mit dem Werkzeug vertraut zu machen.

Neben der Excel-Tabelle findet sich zunächst eine kurze Einführung in das Vorgehen, mit welchem die Tabelle zu bearbeiten ist. Dabei ist es beispielsweise wichtig, dass vor der eigentlichen Nutzung die Makros aktiviert werden, um sämtliche Funktionalitäten der Datei nutzen zu können. Die Anwendung kann nun über den "Start"-Button gestartet werden. Durch den Klick auf diesen Knopf öffnet sich ein Fenster, welches aus verschiedenen Reitern besteht. Der "Start"-Reiter enthält zunächst eine Beschreibung über das Vorgehen bei der Gewichtung und Bewertung der einzelnen Kriterien. Indem man auf "Weiter" klickt, gelangt man in den nächsten Reiter mit der Bezeichnung "Frameworks".

The screenshot shows a dialog box titled "Nutzwertanalyse - Einstellungen" with a close button (X) in the top right corner. The dialog has four tabs: "Start", "Frameworks", "Allgemeine Gewichtung", and "Ende". The "Start" tab is active. Below the tabs, there is a text instruction: "Bitte geben Sie hier die Anzahl zu vergleichender Frameworks an und füllen dann die Textfelder mit den entsprechenden Namen dieser Frameworks. Ihre Eingabe wird automatisch in die Bewertungstabelle übernommen." Below this text is a dropdown menu with the number "3" selected. Underneath the dropdown is a section titled "Frameworks zur Bewertung" containing three empty text input fields. At the bottom of the dialog, there are three buttons: "Abbrechen", "Zurück", and "Weiter".

Abbildung 8: Festlegung zu bewertender Frameworks

In diesem lässt sich mit Hilfe einer Drop Down-Liste die Anzahl der zu vergleichenden Frameworks festlegen, welche auf maximal drei Produkte begrenzt ist. Je nach Auswahl erscheinen dann die entsprechenden Eingabefelder, in die jeweils der Name des zu bewertenden Frameworks einzugeben ist. Diese Eingabe wird automatisch in die Excel-Tabelle übernommen. Durch den Klick auf den "Weiter"-Knopf wird man in den nächsten Reiter navigiert, bei welchem es sich um die allgemeine Gewichtung handelt. Hier geht es darum, die Wichtigkeit der einzelnen Anforderungskategorien zu definieren. Diese sind in funktionale und nichtfunktionale Anforderungen unterteilt, wobei für beide Bereiche insgesamt 100% zu vergeben sind. Zu den funktionalen Anforderungen zählen die Kategorien Benutzeroberfläche, Darstellungssteuerung, Geschäftslogik, Datenmodell und Architektur.

Nutzwertanalyse - Einstellungen

Start | Frameworks | Allgemeine Gewichtung | Ende

Funktionale Anforderungen | Nichtfunktionale Anforderungen

Bitte gewichten Sie in diesem Schritt die unten stehenden Kriterienbereiche entsprechend Ihren Anforderungen. Es sind insgesamt 100% zu vergeben.

Benutzeroberfläche	Zusatzkategorie 1
<input type="text" value="20"/> %	<input type="text" value="0"/> %
Darstellungssteuerung	Zusatzkategorie 2
<input type="text" value="20"/> %	<input type="text" value="0"/> %
Geschäftslogik	
<input type="text" value="20"/> %	
Datenmodell	
<input type="text" value="20"/> %	
Architektur	
<input type="text" value="20"/> %	

Abbrechen Zurück Weiter

Abbildung 9: Gewichtung der funktionalen Anforderungen

Die nichtfunktionalen Anforderungen bestehen aus den Kategorien Reifegrad, Verbreitungsgrad und Nutzung.

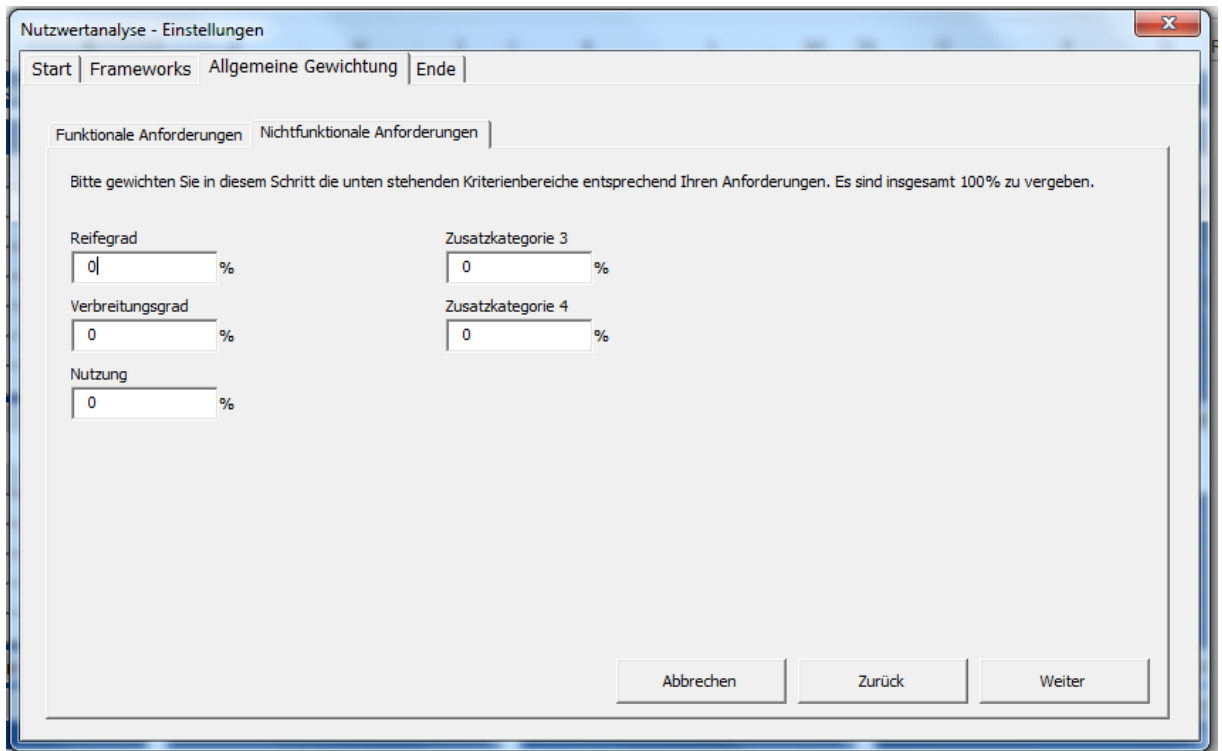


Abbildung 10: Gewichtung der nichtfunktionalen Anforderungen

Wurden keine 100% vergeben, erhält der Anwender die folgende Fehlermeldung:

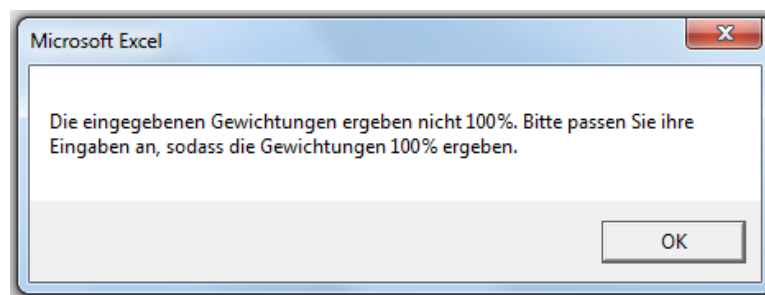


Abbildung 11: Fehlermeldung bei inkorrekt vergebener Gewichtung

Im Laufe dieses Kapitels wird noch beschrieben, aus welchen konkreten Kriterien sich die einzelnen Kategorien zusammensetzen und wie diese Auswahl zustande kommt.

Indem nun erneut auf "Weiter" geklickt wird, öffnet sich der letzte Reiter mit dem Namen "Ende". Hier sorgt der "Fortfahren"-Knopf dafür, dass sämtliche Eingaben aus den Felder direkt an die entsprechende Zelle in der Excel-Tabelle übergeben werden.

Wie die folgende Abbildung zeigt, besteht die Tabelle grundsätzlich aus zwei Dimensionen: Einerseits werden auf der linken Seite die Anforderungskategorien sowie die einzelnen

Kriterien aufgelistet, welche sich über das "+" ausklappen lassen, andererseits werden oben die zu bewertenden Frameworks dargestellt:

Nutzwertanalyse von clientseitigen Open Source Frameworks zum Bau mobiler Web-Anwendungen										
Kriterien										
Funktionale Anforderungen	Gewichtung	Begründung	Bewertung	Score	Begründung	Bewertung	Score	Begründung	Bewertung	Score
Benutzeroberfläche	20,00%			3,42857			4,71429			0,93878
✓ Einfache Widgets	5		2	10		3	15		10	50
✓ Zusammengesetzte Widgets	5		4	20		4	20		8	40
✓ Containers	5		5	25		5	25		3	15
✓ Visualisierung	5		2	10		6	30		4	20
✓ Themes	5		3	15		4	20		8	40
✓ Styles	5		4	20		2	10		7	35
✓ Event Handling	5		4	20		9	45		6	30
Darstellungssteuerung	20,00%			0			0			0
Geschäftslogik	20,00%			0			0			0
Datenmodell	20,00%			0			0			0
Architektur	20,00%			0			0			0
Zusatzkategorie 1				0			0			0
Zusatzkategorie 2				0			0			0
Total Funktionale Anforderungen	100,00%			3,42857			4,71429			0,93878
Nichtfunktionale Anforderungen										
Reifegrad	30,00%			0			0			0
Verbreitungsgrad	40,00%			0			0			0
Nutzung	30,00%			0			0			0
Zusatzkategorie 3				0			0			0
Zusatzkategorie 4				0			0			0
Total Nichtfunktionale Anforderungen	100,00%			0			0			0
Gesamtscore				3,42857			4,71429			0,93878
Platzierung				2			1			3

Abbildung 12: Übersicht Bewertungstabelle

Die Gewichtung je Kategorie wurde, wie bereits erwähnt, aus dem Eingabefenster übernommen. Im nächsten Schritt muss nun die Wichtigkeit der einzelnen Kriterien innerhalb einer Kategorie eingeschätzt werden. Dazu kann eine Zahl auf der Skala von 0-10 vergeben werden, wobei 0 für sehr unwichtig und 10 für sehr wichtig steht. Für jedes Framework muss pro Kriterium zudem eine Bewertung abgegeben werden, welche den Grad der Erfüllung des Kriteriums durch das jeweilige Framework angibt. Auch hier kann auf einer Skala von 0 bis 10 bewertet werden, wobei 0 für sehr schlecht und 10 für sehr gut steht. Aus der Gewichtung pro Kategorie, der Gewichtung pro Kriterium innerhalb einer Kategorie sowie der Bewertung des Kriteriums wird automatisch der Score berechnet, den ein bestimmtes Frameworks in Bezug auf ein Kriterium erreicht. Diese Scores lassen sich addieren, so dass sich ein Score je Kategorie, für die funktionalen und nichtfunktionalen Anforderungen und für das gesamte Framework ergibt. Indem abschließend die Gesamtscores der einzelnen Frameworks miteinander verglichen werden, lässt sich eine Platzierung ermitteln, welche die Produkte hinsichtlich ihrer Fähigkeit für den jeweiligen Anwendungsfall einstuft.

3.2 Identifikation und Begründung der Anforderungen

Die Auswahl der zuvor genannten Anforderungskategorien und der konkreten Kriterien erfolgt, indem die Vielzahl an möglichen clientseitigen Open Source-Frameworks zum Bau von mobilen Web-Anwendungen zuvor auf eine Auswahl von 23 Produkten eingegrenzt wurde. Diese Frameworks können nun auf ihre primären Features untersucht werden. Dabei werden die identifizierten Kriterien aus Gründen der Struktur und Übersichtlichkeit in Kategorien unterteilt und geordnet. Grundsätzlich lässt sich unterscheiden in funktionale Anforderungen, welche aus den Kategorien Benutzeroberfläche, Darstellungssteuerung, Geschäftslogik, Datenmodell und Architektur bestehen, und nichtfunktionale Anforderungen, die sich aus den Bereichen Reifegrad, Verbreitungsgrad und Nutzung zusammensetzen. Die Aufteilung der funktionalen Anforderungen erfolgt dabei in Anlehnung an das zuvor vorgestellte Model-View-Controller-Modell, wobei diese bei der abstrakten Ebene der Benutzeroberfläche beginnt und bis hin zum Datenmodell und der Architektur immer technischer wird.

3.2.1 Funktionale Anforderungen

3.2.1.1 Interface (GUI)

Das Graphical User Interface, die Benutzeroberfläche einer Web-Anwendung, ist die Schnittstelle zwischen dem Anwender (Mensch) sowie dem System (Maschine) und somit hauptverantwortlich für eine gute Kommunikation und Interaktion zwischen diesen beiden Parteien. Die Elemente des GUI haben die Funktion, Befehle vom Anwender entgegenzunehmen und diesem wiederum zum Beispiel Text als Information anzuzeigen.⁷³

Basic Widgets

Widgets dienen im Allgemeinen dazu, eine Anwendung sowohl intuitiv, als auch angenehm bedienbar zu machen. Daher sollte für einen Anwender eine entsprechende Oberfläche (UI) entworfen werden, welche einfach zu nutzen ist.⁷⁴ Dieses User Interface wird mit sogenannten Widgets gefüllt. Anders ausgedrückt handelt es sich bei einem Widget lediglich um die einfachste grafische Einheit.⁷⁵ Viele dieser sind generell bekannt: Es handelt sich hierbei um die grundlegenden Bedienelemente wie Buttons, Labels, Textfelder, Check- oder Radioboxes u.v.m.⁷⁶

⁷³ Vgl. IT Wissen (o.J.)

⁷⁴ Vgl. Kobes, R. (2003)

⁷⁵ Vgl. proggen.org (o.J.)

⁷⁶ Vgl. Kobes, R. (2003)

Compound Widgets

Im Gegensatz zu Basic Widgets, welche nur ein einzelnes Bedienelement darstellen, setzen sich Compound Widgets – wie bereits der Name aus dem englischen „zusammengesetzte Widgets“ schließen lässt – aus mehreren Basic Widgets zusammen.⁷⁷ Ein Compound Widget besteht und beinhaltet also andere Widgets. Compound Widgets werden genutzt, um nicht vollständig neue Widgets entwickeln zu müssen, jedoch dennoch eine neue zusammenhängende und wiederverwendbare logische Einheit zu erstellen. Als Beispiel könnte hier eine Combo Box dienen, welche heruntergebrochen aus den Teilen eines Textfeldes, eines Buttons sowie einer Popup-List besteht.⁷⁸

Containers

Container helfen dem Programmierer bei der Erstellung und Verwaltung des Layouts einer Anwendung. Sie können zunächst einmal als Rahmen beschrieben werden, in dem verschiedene Elemente gruppiert und zusammen gehalten werden können. Hierbei handelt es sich um die gängigen HTML Elemente wie beispielsweise Buttons o.ä.⁷⁹ Die Container stehen in einem Parent-Child Verhältnis, wodurch alle Layout-Eigenschaften des Containers entsprechend für alle Elemente gelten. Dies vereinfacht die Anordnung verschiedener Bestandteile einer Oberfläche und den Umgang mit diesen. Bekannte Container sind z.B. Frames, Forms oder Panels.⁸⁰

Visualisierung

Visualisierungen beziehen sich auf die Möglichkeit, Daten und Informationen ansprechend aufzubereiten und einem Nutzer diese somit auf einfache, übersichtliche und verständliche Weise zu übermitteln. Datenvisualisierungen sind nichts anderes als die weithin bekannten Diagramme, Charts oder Grafiken.⁸¹ Auch Animationen schließt dieses Feature mit ein. Demnach sollte hier bewertet werden, wie ein Framework diese Art von Informationsbereitstellung unterstützt.

⁷⁷ Vgl. Northover, S./MacLeod, C. (2001)

⁷⁸ Vgl. Android Developer (o.J.)

⁷⁹ Vgl. dr.web (2003)

⁸⁰ Vgl. Bada Developer (o.J.)

⁸¹ Vgl. Friedman, V. (2007)

Themes

Themes stellen die allgemeine Idee, welche hinter einer Webseite steckt, dar. Sie sind im Vergleich zu einzelnen Styles als übergeordnet zu betrachten. Ein Theme gibt zum Beispiel die Farbgebung o.ä. einer Web-Anwendung vor. Mit Hilfe solcher Themes können also Web Apps mit Wiedererkennungswert geschaffen werden – in anderen Worten wird eine Art Corporate Design umgesetzt (z.B. Webseitenfarbgebung durch Firmenlogo beeinflusst).⁸²

Themes werden von Frameworks in unterschiedlichem Umfang unterstützt, was in diesem Fall zur Bewertung führt. So kann abgewogen werden, in welchem Umfang und mit welcher Qualität Themes zur Verfügung gestellt werden, oder ob beispielsweise Tools bereitstehen, die das eigenständige Erstellen von Themes ermöglichen.⁸³

Styles

Das Feature Styles beschreibt grundsätzlich den Aufbau und das Aussehen der durch die Frameworks entwickelten Web-Anwendungen. Im Gegensatz zu Themes, welche generell für eine Webseite – egal auf welchem Endgerät – entworfen werden (ähnlich Corporate Design), werden die Styles an die jeweiligen Darstellungsgegebenheiten angepasst und können sich entsprechend häufiger ändern. In der Welt der Webentwicklung wird dies durch ein Zusammenspiel von HTML und CSS ermöglicht, wobei HTML die Struktur und CSS den „Style“ vorgeben.⁸⁴ Für dieses Feature sind demnach sowohl der Umfang an verfügbaren CSS Libraries und Styles, als auch der angedachte Einsatz dieser (möglichst externes Stylesheet, aber auch interner bzw. Inlinestyle möglich) und die Anpassungsfähigkeit dieser Komponenten von Bedeutung.⁸⁵

Event Handling

Dieses Feature beschreibt das Event Handling für Elemente des User Interfaces. Jedes Element des UI kann Events haben, das bekannteste Beispiel sei hier das Klicken eines Buttons und das damit verbundene Ausführen eines spezifischen Codes.⁸⁶

Frameworks können Events auf unterschiedliche Arten verarbeiten: Diese Arten gegeneinander abzugrenzen und zu bewerten ist die Aufgabe dieses Schrittes. Unter anderem sollte ein Auge auf die Unterschiede zwischen der vor allem bei Web Apps

⁸² Vgl. Yii Framework (o.J.)

⁸³ Vgl. Wills, A. (2014)

⁸⁴ Vgl. ASP.NET Resources (2004)

⁸⁵ Vgl. Expression Web Tutorial (2014)

⁸⁶ Vgl. IBM (o.J.)

gängigen deklarativen Programmierung und der imperativen Programmierung⁸⁷ sowie zwischen clientseitigem (JavaScript) und serverseitigem (PHP) Verarbeiten der Events geworfen werden.⁸⁸

3.2.1.2 Darstellungssteuerung

Ein wichtiger Bestandteil einer jeden Web-Anwendung ist die Darstellungssteuerung, welche auch oft als Benutzerschnittstellensteuerung oder im englischen als View-System bezeichnet wird.⁸⁹ Sie dient hauptsächlich dazu, Daten zu visualisieren und Benutzereingaben zu verarbeiten.⁹⁰ Teilweise handelt es sich bei der Darstellungssteuerung auch um die oberste Schicht einer mehrschichtigen Anwendung. Die Aufgaben umfassen größtenteils das Erzeugen von Ausgaben, das Umsetzen von Methodenaufrufen in der Geschäftslogik, das Umsetzen von Antworten der Geschäftslogik in Ausgaben, das Prüfen von Eingaben und einen ereignisgetriebenen (nicht-linearen) Ablauf.⁹¹

Ähnlich wie bei den anderen Kategorien beinhaltet der Begriff der Darstellungssteuerung eine Vielzahl an funktionalen Eigenschaften (Features), welche die Umsetzung der einzelnen Aufgaben ermöglichen, die ihr zukommen. Diese Komponenten sollen im Folgenden näher erläutert werden, damit ersichtlich wird, warum diese für eine Analyse verschiedener Entwicklungsframeworks relevant sind.

Layout Manager

Ein Layout Manager beschreibt Elemente in der Programmierung, welche sich mit der visuellen Anordnung von Komponenten in einem übergeordneten Container befassen. Bei einem solchen Container handelt es sich meist um ein Fenster oder Window. Die Besonderheit eines Layout-Managers ist, dass nicht die Größe und Position einer jeden Komponente mittels Pixelkoordinaten bestimmt wird. Stattdessen besitzt ein Layout-Manager ein Regelwerk, wie die Positionierung stattzufinden hat. Dabei ist diese Vorgehensweise im Unterschied zu einer absoluten Positionierung unabhängig von Zeichensatz und Fenstergröße.⁹²

⁸⁷ Vgl. Heilmann, C. (2007)

⁸⁸ Vgl. Embarcadero (2012)

⁸⁹ Vgl. Sencha (2014), S.5

⁹⁰ Vgl. Schwichtenberg, Dr. H. (o.J.a)

⁹¹ Vgl. Schwichtenberg, Dr. H. (o.J.b)

⁹² Vgl. Exploring Java (o.J.)

Interaktionen

Im Zusammenhang mit Web-Anwendungen insbesondere im Hinblick auf die Darstellungssteuerung beziehen sich Interaktionen größtenteils darauf, wie die Aktionen eines Benutzers im System umgesetzt werden. Dabei sollen Bewegungen oder Gesten in system-relevante Aktionen umgesetzt werden.⁹³ Im Zusammenhang mit mobilen Web-Anwendungen, welche bei dieser Arbeit im Vordergrund stehen, geht es hierbei primär um sogenannte „Touch-Events“. Diese unterscheiden sich von den „Mouse-Events“, welche auf Desktop-Web-Anwendungen eingesetzt werden. Fingerbewegungen auf dem Touchscreen werden demnach in eine Klick- (Auswahl) oder Scroll-Aktion umgesetzt. Aber auch Drag & Drop Funktionen sind möglich.⁹⁴

Drawing

Hierbei handelt es sich um eine zusätzliche Funktionalität, welche über die Verwendung eines zusätzlichen Application Programming Interface in Frameworks zur Entwicklung von Web-Applikationen eingebaut wird. Die Drawing API liefert einen Parser, welcher dynamisch gezeichnete Linien oder Grafiken auf dem Back-End umsetzen kann, ohne das Wissen und die Funktionalitäten des Front-End Toolkits zu kennen, welches zum Zeichnen der Grafik verwendet wurde.⁹⁵ Dieses Konzept wird zum Beispiel bei Anwendungen wie Google Maps eingesetzt.

Theming

Unter Theming fällt die grundlegende Anwendung von Templates und Design-Themes auf Web-Anwendungen. Grundsätzlich handelt es sich dabei um ein systematisches Verfahren das Aussehen einer Web-Anwendung anzupassen. Je nach Framework läuft dieser Prozess über das Theme-Attribut. Hier wird einfach auf das gewünschte Template verwiesen, so dass eine Anpassung stattfindet. In den meisten Frameworks sind diese Themes als eine Anzahl an verschiedenen Dateien, wie beispielsweise View, Layout und relevante Ressourcen Files hinterlegt.⁹⁶

⁹³ Vgl. Sencha (2014), S.5

⁹⁴ Vgl. Lauke, P. H. (2012)

⁹⁵ Vgl. Drupal (2007)

⁹⁶ Vgl. Yii Framework (2011)

Templating

Um das sogenannte Templating durchführen zu können, benötigt das Framework Templating Engines. Mit Hilfe derer ist es möglich, dass die Programm-Logik und Präsentation in zwei voneinander unabhängige Teile aufgeteilt werden. Dadurch wird die Entwicklung beider Teile vereinfacht, außerdem wird das gesamte System flexibler und Modifikationen sowie die Wartung können besser durchgeführt werden.⁹⁷ Durch den Templating-Prozess werden anschließend Platzhalter durch die Anwendung gewisser Vorschriften in finalen Inhalt umgewandelt.⁹⁸ Meist handelt es sich beim Templating um HTML Templates.

Visuelle Effekte

Unter der Bezeichnung visuelle Effekte werden übergreifend verschiedene Teilaspekte zusammengefasst. Dazu zählt unter anderem die Animation verschiedener Eigenschaften. Darunter versteht man, dass eine bestimmte Eigenschaft, wie zum Beispiel die Größe oder Position eines Objektes, für eine bestimmte Zeit geändert wird. Ein weiterer visueller Effekt ist die visuelle Transformation verschiedener Objekte auf Webseiten. Dazu gehören bspw. verschiedene Inhalte verschwimmen oder verblassen zu lassen, ebenso wie bestimmte Objekte für eine bestimmte Dauer zu verfärben oder hervorheben.⁹⁹

Lokalisierung

Die Lokalisierung ist eine wichtige und oft verwendete Komponente bei Web-Anwendungen. Hierbei wird, abhängig von dem Ort an dem die Applikation ausgeführt wird, eine Anpassung der Anwendung vorgenommen. Dabei kann es sich um die Anpassung von Zeit und Datum, sowie deren Format handeln, ebenso wie eine Anpassung von Sortierfunktionen, Widget-Layouts und Struktur und Design der Website in Abhängigkeit von Sprache und Sprachstruktur.¹⁰⁰ Meist wird Lokalisierung allerdings nicht als ein einzelnes Kriterium betrachtet, sondern in Kombination mit weiteren Komponenten der Darstellungssteuerung eingesetzt.

Bedienmöglichkeiten

In Bezug auf Bedienmöglichkeiten gilt es grundsätzlich zwei verschiedene Komponenten zu betrachten. Auf der einen Seite spielen hier allgemeine Funktionalitäten, die sich auf die

⁹⁷ Vgl. Koebler, R. (2013)

⁹⁸ Vgl. Sencha (2014)

⁹⁹ Vgl. Sencha (2014), S.5

¹⁰⁰ Vgl. Sencha (2014), S.5

Steuerung der Anwendung beziehen, eine wichtige Rolle. Bei modernen Rich-Internet-Applications (RIA) zählen hierzu vor allem Drag&Drop Funktionen, ebenso wie die Steuerung durch Tastaturkürzel.¹⁰¹ Andererseits spielen hier Funktionalitäten eine wichtige Rolle die von Menschen mit einer gewissen Beeinträchtigung genutzt werden können. Hierzu zählt z.B. eine Vorlesefunktion, die Möglichkeit Texte zu vergrößern oder eine vereinfachte Tastatur-Navigation.¹⁰² Wie sich also zeigt, hängen die Teilbereiche Bedienmöglichkeiten und Interaktionen eng miteinander zusammen.

Browserunterstützung

Der Begriff Browserunterstützung bezieht sich darauf, für welche Internet-Browser die Entwicklung mit dem in Frage kommenden Framework möglich ist. Das bedeutet, dass die entwickelte Anwendung auf einer Auswahl der zur Verfügung stehenden Internet-Browser lauffähig ist. Je nach Framework und der zu Grunde liegenden Technologie werden nur ausgewählte Konzepte und Formate unterstützt oder die entwickelte Anwendung ist plattform- und browserunabhängig.

View Binding

Die Aufgabe von Bindings ist es generell, eine Applikation oder Anwendung mit dem Document Object Model (DOM) zu verbinden. Dabei werden meist Daten-Attribute zu den HTML-Templates hinzugefügt. Bei Änderungen sind sowohl die DOM-Node, als auch das Java-Script Objekt betroffen. Wird also eine Änderung von dem Benutzer vorgenommen, muss diese Änderung umgehend an das DOM weitergegeben werden, so dass das Model angepasst werden kann. Dies geschieht über sogenannte Controls, welche meist durch UI-Widgets dargestellt werden. Dies ist mit einem gewissen „Recordset“ im DOM verbunden, welches nach einer Validierung der Eingabe angepasst wird. Dieses Update findet dann statt, wenn das „Exit-Event“ der Control ausgelöst wurde. Aus diesem Grund wird das View Binding auch als UI-Data Binding beschrieben.¹⁰³

DOM Scripting

Zunächst soll zum Verständnis das Document Object Model kurz erläutert werden. Dabei handelt es sich um eine browserseitige Anordnung des vom Server übergebenen HTML-

¹⁰¹ Vgl. Innopact (2014)

¹⁰² Vgl. Sencha (2014), S.6

¹⁰³ Vgl. Wikipedia (2014e)

Dokuments. Dieses wird in kleinste Einheiten (=Objekte) unterteilt und in einem DOM-Baum angeordnet. Die einzelnen Elemente nennen sich Knoten oder Nodes.¹⁰⁴

„DOM-Scripting benennt die Technik, mit der man dynamisch im Browser eine Webseite verändert. Mittels JavaScript greift man dabei auf das DOM (Document Object Model) zu.“¹⁰⁵ Eine DOM-Manipulation erfolgt meist in drei Schritten. Zunächst wird der aktuelle DOM-Baum analysiert und die Stelle herausgesucht, die angepasst werden soll. Danach wird die Manipulation intern vorbereitet und abschließend durchgeführt. Hierbei handelt es sich um einen Vorgang, der nicht technologisch, sondern von der Herangehensweise neu ist.¹⁰⁶

3.2.1.3 Geschäftslogik

Wie aus der Aufteilung der einzelnen Kategorien bereits hervorgeht, stellt die Ebene der Geschäftslogik grundsätzlich die Schnittstelle und die Schicht zur Kommunikation zwischen der Benutzeroberfläche und der Datenbank dar. In Bezug auf die behandelten Frameworks zum Bau mobiler Web-Anwendungen besteht diese Kategorie aus den Kriterien Zustandsmanager, Modularität, Testing und multimediale Verarbeitungsmöglichkeiten, welche im Folgenden genauer erläutert werden.

Zustandsmanager

Der Zustandsmanager ermöglicht es dem Entwickler, die Zustandskonsistenz der Anwendung sicherzustellen. Das Zustandsmanagement spielt zudem immer dann eine Rolle, wenn die Verwaltung von Elementen der Benutzeroberfläche von großer Bedeutung ist. Dazu zählen beispielsweise Textfelder, Knöpfe oder Optionsfelder. Konkret hängt der Zustand von einem Objekt immer vom Zustand anderer Objekte ab. Beispielsweise ist ein Knopf nur dann aktiviert und kann angeklickt werden, wenn bestimmte Eingabefelder mit gültigen Werten gefüllt wurden.¹⁰⁷

Modularität

Die Modularität beschreibt die Möglichkeiten, den Code zu strukturieren, um auf diese Weise die benötigte Übersichtlichkeit sicher zu stellen. Dazu zählen beispielsweise Methoden wie das Object Namespacing, bei dem die Namen für Objekte in einer Baumstruktur angeordnet und über entsprechende Pfadnamen eindeutig angesprochen werden. Konkret bezeichnet in

¹⁰⁴ Vgl. Aaken, G. van (2006)

¹⁰⁵ Aaken, G. van (2006)

¹⁰⁶ Vgl. Aaken, G. van (2006)

¹⁰⁷ Vgl. Wigmore, I. (2013)

jedem Namensraum ein Name ein eindeutiges Objekt, während der gleiche Name in einem anderen Namensraum ein anderes Objekt referenzieren kann.

Testing

Dieses Kriterium untersucht die Fähigkeit eines Frameworks, die entwickelte mobile Web-Anwendung im Hinblick auf ihre Funktionsweise und die Existenz von Systemfehlern zu untersuchen. Dabei ist zu bewerten, inwieweit das jeweilige Framework die Durchführung von Komponententests zum Test einzelner Objekte oder Methoden, Integrationstests für die Zusammenarbeit der Komponente untereinander, oder gesamter Systemtests unterstützt.

Multimediale Verarbeitungsmöglichkeiten

Multimediale Verarbeitungsmöglichkeiten beschreiben die Möglichkeiten eines Frameworks, multimediale Ressourcen wie beispielsweise Bilder, Videos oder Audioinhalte in die mobile Web-Anwendung zu integrieren. Dazu gehört ebenfalls die Verarbeitung von Metadaten zu den multimedialen Ressourcen. In diesem Kriterium wird folglich bewertet, wie offen ein Framework gegenüber der Integration von Multimedia ist und welche Werkzeuge es für diese Einbettung bietet.

3.2.1.4 Datenmodelle

Die Datenmodelle-Schicht wurde, wie in der Aufteilung der einzelnen Kategorien hervorgeht, als eigene Schicht erfasst, obwohl sie aber prinzipiell eng mit der Geschäftslogik verknüpft ist. Dabei sorgen die Elemente der Datenmodell-Schicht für die Verwaltung und Verarbeitung der Daten entsprechend der Vorgaben des Systems. Bei den relevanten Elementen handelt es sich um: Databinding, Datenobjekte, Datenquellen, Persistente Daten, Datenmodelle und Speichermethoden.

Databinding

Databinding beschreibt die Technik, die im Rahmen von Web-Applikationen verwendet wird, um Datenquellen zu verbinden und zu synchronisieren. Es stellt hierbei die Verbindung zwischen der Benutzeroberfläche und der Geschäftslogik her. Dabei sorgt Databinding dafür, dass Elemente, die mit bestimmten Werten verbunden sind, eventuelle Wertänderungen direkt reflektieren. Ebenso kann eine äußerliche Veränderung der Daten in einem Element automatisch an die zugrundeliegenden Daten weitergeleitet werden. Databinding kann sich

demzufolge entweder uni- oder bidirektional ausprägen. Bei unidirektionaler Bindung ist A mit B, jedoch B nicht mit A verbunden. Dies führt dazu, dass Änderungen von A sich auf B auswirken, aber nicht umgekehrt. Bei bidirektionaler Bindung beeinflussen Manipulationen an Datenquelle sich jeweils gegenseitig.¹⁰⁸

Datenobjekte

Datenobjekte beziehen sich unmittelbar auf Objekte im Sinne der objektorientierten Programmierung, in der die Architektur Objekte der Wirklichkeit widerspiegelt. Diese Objekte können in einer Benutzeroberfläche grafisch dargestellt werden, während im Hintergrund eine Klasse in der jeweiligen Skriptsprache erstellt wird, die allen Charakteristiken der projektorientierten Programmierung folgt, wie zum Beispiel dem Vererben.¹⁰⁹ Clientseitigen Web-Frameworks ermöglicht dieses Feature bei fehlenden Bäumen, Queues oder Graphen in der Bibliothek, diese vom Frontend zu stellen.¹¹⁰

Datenquellen

Datenquellen setzen sich aus den für die jeweilige Anwendung relevanten Daten zusammen. Bei Datenquellen handelt es sich somit in der Regel um Datensätze, die aus Datenbanken in jeglicher Form, Diensten oder Objekten stammen. Datenquellen ermöglichen beispielsweise, mit entsprechenden Werkzeugen zugrundeliegende Daten mit Steuerungselementen zu verbinden.¹¹¹

Persistente Daten

Bei diesem Kriterium handelt es sich um die Fähigkeit, Daten über die Laufzeit eines Programmes zu speichern hinweg, so dass diese auch nach Neustarten der Web-Anwendung weiterhin verfügbar sind. Üblicherweise geschieht dies über die Festplatte des Computers, was im Falle von mobilen Web-Anwendungen jedoch nicht möglich ist.¹¹² Hier kommen daher Möglichkeiten wie das Caching in Frage, bei dem Ressourcen anhand bestimmter Kriterien in einem Cache, also einmal schneller Puffer-Speicher, zwischengespeichert werden. Alternativ kann auch über die Synchronisierung sichergestellt

¹⁰⁸ Vgl. Microsoft.com (2014a)

¹⁰⁹ Vgl. Pimcore.org (2012)

¹¹⁰ Vgl. Sencha (2014), S. 4 ff.

¹¹¹ Vgl. Vgl. Microsoft.com (2014b)

¹¹² Vgl. Der Dualstudent (o.J.)

werden, dass Daten an einem anderen Ort hinterlegt werden. Das beschriebene Kriterium bewertet die Existenz solcher Methoden zur Sicherstellung persistenter Daten.

Datenmodelle

Das Ziel von Datenmodellen ist es, Strukturen für die Erfassung und Verarbeitung der Daten eines definierten Anwendungsbereichs zu erstellen.¹¹³ Bei objektorientierten Datenmodellen werden die Objekte durch ihre Attribute und Beziehungen modelliert. Der Endzustand eines Datenmodells ist die Speicherung der relevanten Daten entsprechend dem Modell. Die Unterbrechung des klassischen Request/Response-Schemas und die Durchführung von Reaktionen durch den Client bei clientseitigen Web-Anwendungen führen dazu, dass Datenmodelle auch zur clientseitigen Datensicherung verwendet werden. Ein entsprechendes Framework schafft daher Hauptspeicher-Datenstrukturen für die laufenden Inhalte einer Anwendung.¹¹⁴

Speichermethoden

Bei traditioneller Webprogrammierung ging es bislang darum, anwendungsrelevante Daten des Nutzers auf einem Server zu speichern. Anders verhält sich dies bei clientseitigen Browseranwendungen – hier können Daten dauerhaft auf dem lokalen Speicher des Clients gehalten oder, um eine Verfügbarkeit der Daten auf mehreren Geräten zu ermöglichen, auf einen Server (eine Cloud) synchronisiert werden.¹¹⁵ Für das clientseitige Speichern von Daten gibt es drei primäre Gründe: 1. Offline Verfügbarkeit 2. Performance-Steigerung 3. Einfache Programmierung ohne Serverinfrastruktur. Hierfür existieren unterschiedliche APIs, die es der browserbasierten Anwendung ermöglichen, Daten auf der Festplatte des Nutzers zu sichern: Web Speicher, Web SQL Datenbanken, indexierte Datenbanken und Dateizugriff. Cookies können auf Grund ihrer Simplizität in der Regel die Speicheranforderungen komplexer clientseitiger Anwendungen nicht bedienen. Neben dem Speichervorgang ermöglichen die APIs teilweise auch Suchen, Batch-Manipulationen und das Zurückholen von Daten. Der Speichervorgang stellt sich wie folgt dar: Die Daten werden an die zur Verfügung stehende Speicher-API des Browsers geleitet, welche es im gleichen definierten Bereich auf dem Endgerät speichert, in dem auch nutzerspezifische Informationen gehalten werden. Es ist wichtig, den lokalen Speicher, den eine Webseite nutzen darf, durch den Browser zu limitieren.¹¹⁶

¹¹³ Vgl. Gabler Wirtschaftslexikon (o.J.)

¹¹⁴ Vgl. Sencha (2014), S. 4 ff.

¹¹⁵ Vgl. Html5rocks.com (o.J.a)

¹¹⁶ Vgl. Html5rocks.com (o.J.b)

3.2.1.5 Architektur

Architektur-Elemente beschreiben an dieser Stelle generelle Merkmale eines Frameworks die in keiner der dargelegten Schichten vollständig erfassbar gewesen wären. Diese generellen oder nur schwer kategorisierbaren Eigenschaften (wie z.B. I/O Komponenten) können jedoch je nach Anwendungsfall starken Einfluss auf den Gesamteindruck des Frameworks haben. Hierzu zählen: Schnittstellen, Server Calls, Serverbenachrichtigungen, Datenaustauschformate, Single-Page Anwendungen, Programmiersprachen und Real-Time Collaboration.

Schnittstellen (APIs)

Application Programming Interfaces (APIs) oder auch Programmierschnittstellen werden von einem Softwaresystem zur Verfügung gestellt, um Programme an das System anzubinden. Clientseitige APIs dienen in diesem Zusammenhang der Erweiterung von Browserfunktionalitäten. Dabei werden die Schnittstellen traditionell über Browser Plug-Ins eingefügt, allerdings werden für browserbasierende Clientanwendungen inzwischen überwiegend standardisierte Ajax oder JavaScript APIs eingebunden.¹¹⁷ Die Web-APIs von Mozilla vereinen eine Vielzahl an Schnittstellen für Browseranwendungen, vor allem um den Wandel von nativen mobilen Applikationen zu ermöglichen.¹¹⁸

Server Calls

Wie bereits beschrieben, ermöglichen clientseitige Applikationen das Unterbrechen des klassischen Request/Respond Schemas. Server Calls spielen daher vor allem eine Rolle für die Synchronisation von Daten und das Zurverfügungstellen auf mehreren Endgeräten. Bei Ajax entstehen Server Calls zwischen clientseitigem Skript und dem Webservice, der die Anfrage an eine Datenbank weiterleitet.¹¹⁹

Serverbenachrichtigung

Eine besondere API ist die EventSource API in JavaScript. Diese ermöglicht Echtzeit Benachrichtigungen durch den Server an die Clientanwendung. Hierbei wird HTTP Streaming verwendet, um einen mit einer Web-Anwendung verbundenen Server mit der Fähigkeit auszustatten, Daten asynchron an die Anwendung zu pushen. Diese Events zeichnen sich durch die Nutzung einer unidirektionalen und persistenten Verbindung

¹¹⁷ Vgl. W3.org (2014)

¹¹⁸ Vgl. Mozilla (o.J.)

¹¹⁹ Vgl. GWTproject.org (o.J.)

zwischen Client und Server aus. Diese Verbindung erweckt den Anschein, als sei sie vom Server initiiert worden. Das sogenannte „long polling“ (langes Abfragen) basiert ebenso auf der beschriebenen Verbindung, die geöffnet bleibt, bis eine Antwort vom Server, ausgelöst durch neue Daten, gesendet wurde. Diese Verbindung eignet sich allerdings nicht für Anwendungen mit geringer Latenz (z.B. Online-Spiele), hierzu kann eine Socket-Verbindung zwischen Client- und Serversocket mit Hilfe der WebSocket API verwendet werden.¹²⁰

Datenaustauschformate

Datenaustauschformate beschreiben die unterstützten Dateiformate des Frameworks. Austauschformate enthalten gerade so viele Informationen, wie für das Betrachten und Weiterverarbeiten notwendig ist.¹²¹ JSON und XML sind im Falle von clientseitigen Browserapplikationen hervorzuheben. Dabei dient JSON dazu, Objekte zu Zeichenfolgen plattformunabhängig und für den Menschen lesbar zu serialisieren. XML wird wiederum genutzt, um hierarchisch strukturierte Daten plattform- und implementationsunabhängig in einer Textdatei zu definieren und auszutauschen. Das Netzwerkprotokoll SOAP nutzt so beispielsweise XML für Web-Dienste. Die meisten Browser können XML konstruieren, senden und analysieren. Bei Verwendung von Ajax als Skriptsprache empfiehlt sich aber JSON als adäquateres Austauschformat. Beide Formate haben unterschiedliche Stärken: XML hat seinen Ursprung beispielsweise in der Strukturierung von Dokumententexten (siehe XHTML), während JSON in den Programmiersprachen verankert ist und daher Vorteile beim Abbilden von auszutauschenden Daten zeigt. Dies zeigt, dass es wichtig ist, das Austauschformat entsprechend den Anwendungsanforderungen zu wählen.¹²²

Single-Page Anwendung

Single-Page Anwendungen ermöglichen, wie der Name sagt, das Darstellen der Anwendung auf nur einer Seite im Browser. Der Code (Ajax, JavaScript und/oder HTML5) wird mit der Seite geladen oder dynamisch eingefügt. Dabei ist wichtig klarzustellen, dass die Seite zu keinem Zeitpunkt neu geladen wird oder zu einer anderen Seite weiterleitet. Unter Verwendung von HTML5 ist es allerdings möglich, das Gefühl einer Navigation über eine Seite hinaus zu erschaffen. Komplexere Bibliotheken können als Teil oder als Add-on eines Frameworks das Erstellen von Single-Page Anwendungen deutlich erleichtern. Eine Single-

¹²⁰ Vgl. Html5rocks.com (o.J.c)

¹²¹ Vgl. MyWebcheck.de (2014)

¹²² Vgl. Microsoft.com (2014c)

Page Anwendungen folgt grundsätzlich auch den Anforderungen an clientseitige Multi-Page Browserapplikationen.¹²³

Programmiersprachen

Frameworks verwenden unterschiedliche Programmier- bzw. Skriptsprachen. Die Sprachen, die in dieser Arbeit betrachtet werden, fallen in den Bereich des clientseitigen Scriptens, das es ermöglicht, Programme clientseitig mit Hilfe des Browsers zu verwenden. JavaScript, ActionScript, Dart und Typescript sind populäre Vertreter clientseitiger Skriptsprachen, die durch Sprachen wie Ajax und HTML5 zusätzliche Möglichkeiten für die Entwicklung clientseitiger Browserapplikationen erhalten können. Die Auswahl der richtigen Sprache muss demnach anhand der Erfahrung der Entwickler, den Anwendungsszenarien, der vorhandenen Infrastruktur und der Ressourcen bewertet werden.¹²⁴

Real-Time Collaboration

Real-Time Collaboration beschreibt die Fähigkeiten eines Frameworks, mit mehreren Entwicklern parallel an einem Projekt zu arbeiten. Dabei umfasst es vor allem Themen wie unmittelbare Sichtbarkeit von Änderungen am Code ohne die Seite zu aktualisieren, wodurch Fehler und Mehrarbeit vermieden werden. Zusätzlich kann die Möglichkeit von Integrationen mit tiefergreifenden Zusammenarbeits-Systemen geliefert werden, um die Arbeitsprodukte zu diskutieren oder zu verbessern. Programm-Bibliotheken zur Konfliktlösung und zur Synchronisation wie z.B. Share.js oder OpenCoWeb adressieren diese Herausforderungen der modernen Web-App Programmierung.¹²⁵

3.2.2 Nichtfunktionale Anforderungen

Neben der bereits erfolgten Bewertung und Beschreibung der funktionalen Anforderungen, bzw. der technischen Features gilt es im folgenden Kapitel eine Auswahl an nichtfunktionalen Anforderungen und Kriterien einzuführen und zu erläutern. Diese spielen ebenfalls eine entscheidende Rolle, wenn es um die Bewertung eines Frameworks zur Entwicklung von Web-Anwendungen geht.

¹²³ Vgl. Flanagan, D. (2006), S.497 ff.

¹²⁴ Vgl. W3.org (1997)

¹²⁵ Vgl. Grüneberger, F. J. (2012), S.12 ff.

Zu diesen nichtfunktionalen Kriterien zählen primär der Reifegrad eines Frameworks, sowie dessen Verbreitungsgrad und einige relevante Attribute der Nutzung. Jedes dieser Teilgebiete wird durch eine Vielzahl von Eigenschaften weiter definiert. Daher soll im Folgenden eine kurze Beschreibung der relevanten Punkte stattfinden.

3.2.2.1 Reifegrad

Der Reifegrad eines Frameworks ist für die Anwender dieses, also die Entwickler von Web-Anwendungen, ein wichtiges Kriterium zur Bewertung eines solchen Frameworks. Der Reifegrad bestimmt sich über eine Vielzahl von Kriterien. Dabei zeigt die nachfolgende Beschreibung eine Auswahl der nach Meinung der Autoren dieser Arbeit relevanten Komponenten des Reifegrades auf.

Release-Datum

Zunächst spielt das Release-Datum des Frameworks eine wichtige Rolle, bei welchem es sich um den Zeitpunkt der Veröffentlichung handelt. Zu den Release-Daten zählt nicht nur der erste Release, sondern auch die Veröffentlichung von weiterentwickelten Versionen und Updates. Mit Hilfe dieser Daten kann der Anwender Rückschlüsse bezüglich des Entwicklungsstandes eines Frameworks treffen.

Anzahl an Releases

Dieser Punkt steht in einem direkten Zusammenhang mit dem Release-Datum. Denn die Anzahl an verschiedenen Releases, also erneuerte Versionen und Updates lassen ebenfalls Schlüsse auf den Reifegrad zu. Eine höhere Anzahl an Releases könnte hier grundsätzlich als positiv bewertet werden, da es ein Indiz für eine konstante Weiterentwicklung sein könnte.

Anzahl an Entwicklern

Die Anzahl der an dem Framework arbeitenden Entwickler kann ebenso ein Ausschluss-Kriterium für gewissen Frameworks darstellen. So gibt diese grundsätzlich Auskunft darüber, wie intensiv an einem Framework gearbeitet wird und wie dieses weiterentwickelt wird. Ebenso sagt es aus, wie stark das gesamte Projekt auf einzelne Entwickler angewiesen ist. Auch hier gilt in etwa die Faustregel, dass eine höhere Anzahl an Entwicklern positiv zu betrachten ist.

Dauer zur Fehlerbehebung

Damit ist die Dauer gemeint, welche benötigt wird, bis die zuständigen Entwickler eines gewissen Frameworks einen aufgetretenen Fehler behoben haben. Nach Auftreten eines Fehlers sollte dieser so schnell wie möglich behoben sein. Daher steht dieses Kriterium in einem engen Verbund mit weiteren Kriterien wie z.B. der Anzahl an Entwicklern und Releases ebenso wie der allgemeinen Fehleranfälligkeit.

Allgemein bekannte Probleme

Sollte ein Framework bereits bekannte Probleme aufweisen, sind diese in dieser Kategorie zu bewerten. Damit sind Probleme gemeint, die bereits vor dem Einsatz des Frameworks bekannt sind. Auch an dieser Stelle gilt erneut, dass es hier eine enge Korrelation mit weiteren Kategorien innerhalb der gesamten Reifegrad-Bewertung gibt.

Supportmöglichkeiten

Die Frage, die sich bei einem Fehler oder einem allgemeinen Benutzungsproblem stellt, ist, welche Supportmöglichkeiten für den Anwender zur Verfügung stehen. Dies ist besonders bei Open-Source Produkten ein wichtiger Gesichtspunkt, da hier eine allgemein-abdeckende Supportfunktion meist nicht gewährleistet ist. Eine Rolle spielen hier auch die Supportkanäle und die Dauer, bis der Support vorhanden ist oder umgesetzt wurde.

Dokumentation

Ein sehr wichtiges Kriterium für den Anwender eines Frameworks ist die Dokumentation dessen. Das bedeutet, dass eine Art Handbuch besteht, welches dazu dienen soll, dem Anwender aufzuführen und zu beschreiben, welche Möglichkeiten sich mit diesem Framework bieten, wie es angewendet wird und was es zu beachten gilt. Hier gilt grundsätzlich, dass eine umfangreichere Dokumentation den Entwickler deutlich besser unterstützt. Des Weiteren ist auch diese Komponente eng mit den anderen Bewertungskomponenten des Reifegrades verknüpft.

3.2.2.2 Verbreitungsgrad

Der Verbreitungsgrad eines Frameworks ist ein wichtiges Indiz dafür, wie beliebt und anerkannt es unter Entwicklern ist. Der Verbreitungsgrad kann anhand von unterschiedlichen

quantitativen und qualitativen Faktoren gemessen werden, von denen einige in den folgenden Features zusammengefasst werden.

Bewertung durch Anwender

Die Bewertung durch Anwender umfasst Meinungen von Web-Anwendungsentwicklern zu dem zu bewertenden Framework. Diese Meinungen können aus den einschlägigen Internetforen wie bspw. Github, Stackoverflow, o.ä. bezogen werden.

Zusätzlich zu diesen Meinungen können außerdem Erfahrungen von Entwicklern aus dem eigenen Team zur Vervollständigung dieses Meinungsbildes beitragen.

Marktanteil

Der Markt für Frameworks zur Erstellung mobiler clientseitiger Web-Anwendungen ist groß – gleichzeitig aber auch zersplittert. Daher ist es schwierig, für einzelne Frameworks einen entsprechenden und relevanten Marktanteil in Prozentangaben zu finden. Dies gilt insbesondere für die vielen kleineren Frameworks.

Aus diesem Grund sollte der Marktanteil nicht nur in relativen Zahlen im Vergleich zur Gesamtmenge gesehen werden, sondern ebenfalls beinhalten, wie viele Downloads eines Frameworks es bereits gegeben hat oder wie viele Webseiten basierend auf einem gewissen Framework entwickelt wurden.

3.2.2.3 Nutzung

Der Kriterienbereich der Nutzung beschreibt einige Features, die sich mit der allgemeinen Benutzung bzw. Benutzbarkeit beschäftigen. Diese Kriterien sind ebenso wichtig, um entscheiden zu können, ob ein Einsatz eines Frameworks aus Lizenzgründen oder der Benutzerfreundlichkeit in Frage kommt.

Lizenzmodell

Die Frage des Lizenzmodells sollte im Rahmen dieser Arbeit als eine Art „K.O.-Kriterium“ angesehen werden, da es ausschließlich um die Auswahl geeigneter Open Source Frameworks geht. Doch auch unter diesen Frameworks, die unter einer Open Source Lizenz stehen, gibt es Unterschiede. Unter den betrachteten Frameworks dieser Arbeit waren die folgenden am häufigsten vertreten:

Lizenz	Zusammengefasste Eigenschaften
Apache License 2.0 ¹²⁶	<ul style="list-style-type: none"> - Software frei nutzbar, verbreitbar und veränderbar - Quellcodeänderungen müssen nicht dem Lizenzgeber zugeschickt werden - Eigene Software muss nicht Apache 2 – lizenziert sein, um sie mit Apache 2 – Software zu nutzen
BSD ¹²⁷	<ul style="list-style-type: none"> - Software frei nutzbar, verbreitbar und veränderbar - Bedingung: ursprüngliches Copyright / Disclaimer darf nicht entfernt werden
GNU GPL ¹²⁸	<ul style="list-style-type: none"> - Die General Public License fördert das „Freie Software“-Prinzip - Software unter der GNU GPL License ist frei nutzbar, veränderbar und verbreitbar
MIT ¹²⁹	<ul style="list-style-type: none"> - Software frei zu nutzen, verändern, verkaufen, verbreiten - Dabei ist es unwichtig, ob der Quellcode einsehbar oder nicht einsehbar ist - Bedingung: Urheberrechtsvermerk und Erlaubniserklärung müssen in allen Kopien der Software beiliegen

Es gilt für die Bewertung, die Unterschiede von diesen Lizenzen herauszuarbeiten, für das jeweilige Projekt abzuwägen und eine entsprechende Bewertung zu vergeben. Dabei sollte auch den Unterschieden zwischen kostenlos sowie kostenpflichtigen Frameworks bzw. Framework-Versionen Beachtung geschenkt werden.

Usability

Der Begriff „Usability“ ist sehr abstrakt und schwer zu greifen, dazu umfasst er ein umfangreiches Feld an Kriterien. Unter Usability sollte in diesem Zusammenhang jedoch vereinfacht verstanden werden, wie „gut“ ein Framework zu benutzen und wie einfach es zu bedienen ist. Dass diese Bewertung rein subjektiv ausfallen muss, ist selbstverständlich, jedoch stellt dies kein Problem dar, denn der Anwender/die Anwendergruppe soll sich mit dem Framework wohl fühlen und nicht einer objektiven Meinung z.B. aller Programmierer weltweit folgen.

Für die Bedienbarkeit von Framework sind sicherlich einerseits Bedienungskriterien wie die Oberfläche, Interaktivität, Look and Feel sowie ein schnelles Zurechtfinden von Bedeutung, während außerdem noch Faktoren wie der zuverlässige Betrieb oder die Ausführung von Befehlen eine Rolle spielen. Kriterien wie Dokumentation und Tutorien sind im Kriterienkatalog separat aufgeführt. Das Verständnis für den Begriff Usability kann sich auch abweichend dieser Richtlinien bewegen.

¹²⁶ Vgl. Apache Software Foundation (2004)

¹²⁷ Vgl. Open Source Initiative (o.J. a)

¹²⁸ Vgl. GNU (2014)

¹²⁹ Vgl. Open Source Initiative (o.J. b)

Erweiterbarkeit

Unter diesem versteht man die Möglichkeit, ein Framework hinsichtlich seines Funktionsumfangs in der Zukunft ausbauen zu können. Für die Entwicklung von Web-Anwendungen könnte ein solches Framework, welches sich neuen Gegebenheiten in der Zukunft anpasst und mit neuen Funktionalitäten aufwartet, von großer Bedeutung sein. Zu bewerten sind somit also mögliche Extensions, Plug-Ins, Add-ons, o.ä., die dem Framework zusätzliche Features verleihen, welche es vorher nicht hatte und welche es somit auch ggf. gegenüber anderen Frameworks abgrenzen.

4 Exemplarischer Vergleich marktrelevanter Frameworks anhand des Kriterienkatalogs

In dem zuvor durchgeführten Marktüberblick wurde deutlich, dass eine Vielzahl an Frameworks zur Entwicklung clientseitiger mobiler Web-Anwendungen existiert. Zunächst wurde diese Liste über die Anwendung der Ausschlusskriterien weiter reduziert. Im Anschluss fand eine Reduktion der ausgewählten Frameworks basierend auf Reife- und Verbreitungsgrad statt. Abschließend wurden aus dieser Liste drei Frameworks oder Kombinationen von Frameworks selektiert, um einen exemplarischen Vergleich einzelner Frameworks durchzuführen, um somit die bereits vorgestellte Nutzwertanalyse veranschaulichen zu können.

Eben dieser exemplarische Vergleich wird im Folgenden durchgeführt. Dabei fiel die Auswahl auf eine Selektion von drei Framework(-kombinationen). Nach der primären Analyse im Marktüberblick wurde deutlich, dass die drei ausgewählten Komponenten vermeintlich drei verschiedene Teilgebiete abdecken, wodurch die Nutzwertanalyse samt Bewertungsgrundlagen und Anforderungsbewertungen möglichst gut veranschaulicht werden können.

Zunächst fiel die Wahl auf das Framework Sencha Touch. Nach einer ersten kurzen Analyse wurde deutlich, dass dieses Framework vermeintlich Funktionalitäten in allen fünf bereits festgelegten Kategorien der funktionalen Anforderungen aufweist. Dementsprechend sind sowohl die Teilbereiche Benutzeroberfläche und Darstellungssteuerung, als auch Geschäftslogik, Datenmodell und Architektur abgedeckt.

Darüber hinaus wurde eine Kombination aus jQuery mit jQuery UI ausgewählt. Dies geschah ebenfalls basierend auf der Auswertung des Marktüberblicks. JQuery ist eines der am häufigsten verwendeten Frameworks am Markt, in diesem Fall wird es noch zusätzlich um das Framework JQuery UI erweitert. Diese Kombination ist für den exemplarischen Vergleich besonders interessant, da vor allem die Komponenten, welche in die Kategorien Benutzeroberfläche und Darstellungssteuerung fallen, abgedeckt werden.

Überdies wurde ein weiteres JavaScript Framework bewertet. Hierbei handelt es sich um eine Kombination aus Backbone.js und Underscore.js. Letzteres dient als Erweiterung für das weitverbreitete Backbone.js Framework. Wie aus den Bezeichnungen der beiden Frameworks hervorgeht sind hier eher Komponenten aus den Kategorien Geschäftslogik, Datenmodell und Architektur vertreten.

Der exemplarische Vergleich, der im Folgenden ausgearbeitet und durchgeführt wird, soll die bereits aufgestellten Behauptungen verdeutlichen und aufzeigen, wie sich die Nutzwertanalyse unter den verschiedenen Voraussetzungen verhält.

4.1 Bewertung der Frameworks anhand der Kriterien

Für den exemplarischen Vergleich wird jeweils eine gleichverteilte Gewichtung zugrunde gelegt. Folglich erhalten die funktionalen Anforderungskategorien jeweils 20% und die nichtfunktionalen Kategorien jeweils 33% bzw. 34%. Zudem wird jedes einzelne Kriterium mit dem Gewichtungsfaktor 5 auf der Skala von 0 bis 10 gewichtet.

4.1.1 Nutzwertanalyse von Sencha Touch

Funktionale Anforderungen			
Benutzeroberfläche (20%)			
Feature	Gewichtung	Bewertung	Begründung
Einfache Widgets	5	8	Components verfügbar und über eigene Methoden erstellbar
Zusammengesetzte Widgets	5	7	Lists, Pickers, Grids, etc. Framework mit vielen eingebauten Widgets
Containers	5	6	Verfügbar und eng mit den Methoden der Komponenten verbunden, um diese gruppieren zu können
Visualisierung	5	7	Charts, Carousel, Lists, etc. zur Darstellung verfügbar
Themes	5	9	Spezielle Themes für Betriebssysteme vorhanden, an das Aussehen verschiedener Systeme und Geräte anpassbar und verfügbar in eigener Bibliothek
Styles	5	8	CSS supported via SASS
Event Handling	5	6	Event Listener umfangreich supportet, auch mit zusätzlichen Funktionalitäten
			Score 7,285714
Darstellungssteuerung (20%)			
Layout Manager	5	6	als Layout Class für die Container verfügbar
Interaktionen	5	5	Interaktionen für bspw. Charts vorhanden
Drawing	5	6	möglich mit Draw-Package, aber nicht alle Packages auch kostenlos

			verfügbar
Theming	5	7	Gruppierung von CSS möglich, um Style und Layout der Webseite anzupassen
Templating	5	8	Xtemplate als Templating Engine, vergleichbar mit underscore oder handlebars
Visuelle Effekte	5	6	möglich, auch durch gegebene Vorkonfigurationen
Lokalisierung	5	5	nicht durch vorkonfigurierte Locale-Folder umsetzbar, aber Workaround implementierbar
Bedienmöglichkeiten	5	4	Voice Control über Plug-In in Chrome möglich, aber noch keine mobilen Geräte unterstützt
Browserunterstützung	5	6	IE, Firefox, mobile Browser
View Binding	5	0	Keine Informationen vorhanden
DOM Scripting	5	10	Vollständige DOM Manipulation verfügbar
			Score 5,727273
Geschäftslogik (20%)			
Zustandsmanager	5	3	kein Statemanager, jedoch kann z.B. Sencha.io für diese Funktionalität genutzt werden
Modularität	5	5	MVC Patterns unterstützt, viele neue Funktionalitäten in Version 2, jedoch in Version 1 nicht besonders gut umgesetzt
Testing	5	4	Nicht in Sencha integriert, jedoch problemlos durch weitere Tools zu integrieren und somit durchzuführen
Multimedia	5	7	Video und Audio unterstützt
			Score 4,75
Datenmodell (20%)			
Data Binding	5	5	nur One Way Data Binding (Ext.JS unterstützt 2-Way Data Binding und könnte dies bald an Sencha Touch weitergeben)
Datenobjekte	5	0	Keine Informationen vorhanden
Datenmodelle	5	7	spezielles Datenmodell vorhanden
Persistente Daten	5	9	Sencha kann Daten persistent halten durch sessionStorage, localStorage
Datenquellen	5	0	Keine Informationen vorhanden
Speichermethoden	5	9	Offline Startup über Speichermethoden und Funktionalitäten von Sencha Touch

			umsetzbar
			Score 5

Architektur (20%)			
Schnittstellen (APIs)	5	5	gute Device APIs
Server Calls	5	6	AJAX Calls möglich
Serverbenachrichtigung	5	6	Push Notifications unterstützt
Datenaustauschformate	5	9	XML, JSON, AJAX, Tree, Array
Single-Page Anwendung	5	4	unterstützt
Programmiersprachen	5	8	JS, CSS, HTML
Real-Time Collaboration	0		
Score 5,428571			
Summe funktionaler Anforderungen 28,19156			
Nichtfunktionale Anforderungen			
Reifegrad (33%)			
Feature	Gewichtung	Bewertung	Begründung
Release Date	5	7	erste Version 2010, seit Ende 2013 Version 2.3.1
Anzahl an Releases	5	8	im Schnitt alle 1-2 Monate ein neues Release (alpha, beta, release)
Anzahl an Entwicklern	5	8	keine genaue Zahl bekannt, allerdings Unternehmen mit Vielzahl an Mitarbeitern und Entwicklern, sowie ständig aktualisierte Releases
Dauer zur Fehlerbehebung	5	9	aufgrund der hohen Releasedichte nicht sehr lang
Allgemein bekannte Probleme	5	9	keine speziellen aktuellen Probleme gefunden
Supportmöglichkeiten	5	9	umfangreiche Communities, Fragen allgemein innerhalb eines Tages beantwortet, Entwicklerkonferenzen
Dokumentation	5	9	sehr umfangreich, gut zu bedienen, ausführlich
Score 13,90714			
Verbreitungsgrad (33%)			
Bewertung durch Anwender	5	8	allgemein bekannt als eines der besten und beliebtesten Frameworks
Marktanteil	5	8	2000000 Entwickler, 5000000 Downloads, 50% der Fortune100 nutzen Sencha
Score 13,2			
Nutzung (34%)			
Lizenzmodell	5	9	Commercial, Commercial OEM, Open Source GPLv3

Usability	0		
Erweiterbarkeit	5	7	Extensions verfügbar und pre-approved von Sencha
			Score 13,6
Summe nichtfunktionale Anforderungen 40,70714			
Summe Gesamt 68,8987			

Die vorangegangene Nutzwertanalyse für das Framework Sencha Touch legt dar, dass dieses Entwicklungsframework für den Markt aufgrund seiner Ausgewogenheit und Vielfältigkeit äußerst relevant ist. Sencha Touch kann über alle Bewertungskategorien hinweg gute bis sehr gute Scores aufweisen und sich daher auch gegen spezialisierte Frameworks wie bspw. jQuery behaupten. Da Sencha Touch selbst den Fokus auf ansprechende Oberflächengestaltung für mobile Web Apps legt, ist auch die besonders gute Funktionalität in den Kriterienbereichen Benutzeroberfläche und Darstellungssteuerung nachvollziehbar.

Das Framework erzielt zudem in den nichtfunktionalen Anforderungen gute Ergebnisse. Sencha Touch wird stetig und in kurzen Abständen weiterentwickelt und den jeweiligen Anforderungen angepasst. Das Unternehmen entwickelt außerdem andere Projekte, weshalb eine Vielzahl an Entwicklern zu Verfügung steht, welche das Produkt verbessern. Außerdem gibt es weltweit ein umfangreiches Netzwerk bestehend aus Entwicklern, Foren, Gruppen, Partnern und Meetups, die auch zur Reife des Frameworks beitragen.

Die Beliebtheit von Sencha Touch spiegelt sich auch in den Einsatzgebieten des Frameworks wider: So wird es bspw. von 50% der Fortune 100 Unternehmen genutzt und insgesamt bereits mehr als 5 Mio. Mal heruntergeladen.

4.1.2 Nutzwertanalyse von jQuery und jQuery UI

Funktionale Anforderungen			
Benutzeroberfläche (20%)			
Feature	Gewichtung	Bewertung	Begründung
Einfache Widgets	5	9	Autocomplete, Button, Datepicker, Dialog, Progressbar, Slider, Tabs, Accordion
Zusammengesetzte Widgets	5	8	bietet Widget Factory zum Bau und zur Erweiterung von Widgets (z.B. Datepicker)

Containers	5	6	stellt Component Widget Container bereit
Visualisierung	5	6	jQuery Visualize als Add-On
Themes	5	10	besitzt Themes-GUI namens ThemeRoller (20 vorgefertigte Themes und erweiterbar)
Styles	5	6	unterstützt CSS
Event Handling	5	7	erweitertes Event-System
Score 7,428571			
Darstellungssteuerung (20%)			
Layout Manager	5	5	bietet Plug-Ins wie Magic jQuery oder UI Layout zum Layout Management
Interaktionen	5	8	Drag and Drop, Resizable, Selectable, Sortable
Drawing	5	5	bietet verschiedene Plug-Ins zur Durchführung von Drawings (Draw, jQuery Ink)
Theming	5	6	ebenfalls umsetzbar mit ThemeRoller
Templating	5	6	bietet jQuery Templating Plug-In, um Templates zu laden und zu benutzen
Visuelle Effekte	5	9	Farbanimation, Animation eines Klassenwechsels, Blendeffekte, Bounce, Explode, Fade, Fold, Highlight, Puff, Scale, Shake, Slide
Lokalisierung	5	5	nur über Plug-In möglich (jquery-localize)
Bedienmöglichkeiten	5	5	kann anyway als Plug-In zur Sprachsteuerung einbinden
Browserunterstützung	5	9	Multi-browser Unterstützung
View Binding	5	0	keine Fähigkeit vorhanden
DOM Scripting	5	8	DOM-Manipulation vorhanden
Score 6			
Geschäftslogik (20%)			
Zustandsmanager	5	5	jQuery Page State Manager als Plug-In, um verschiedene Zustände und deren Übergänge zu definieren
Modularität	5	8	jQuery UI ist komplett modular aufgebaut. Ein Entwickler hat die Möglichkeit, nur die Komponenten einzubinden, die er tatsächlich

			benötigt
Testing	5	5	nutzt Qunit, um JavaScript Code zu testen
Multimedia	5	5	jQuery Media Plug-In ermöglicht die Einarbeitung von Flash, Quicktime, Windows Media Player, Real Player, MP3, Silverlight, PDF etc.
			Score 5,75
Datenmodell (20%)			
Data Binding	5	3	eher umständlich, nur über die Anbindung von Plug-Ins möglich
Datenobjekte	5	0	keine Funktionalität
Datenmodelle	5	0	keine Funktionalität
Persistente Daten	5	0	keine Funktionalität
Datenquellen	5	0	keine Funktionalität
Speichermethoden	5	4	bietet nur lokalen Speicher, allerdings lässt sich über Plug-Ins auch Webspeicher implementieren
			Score 1,166667
Architektur (20%)			
Schnittstellen (APIs)	5	7	umfangreiche API Dokumentation
Server Calls	5	0	keine Funktionalität
Serverbenachrichtigung	5	5	Push-Benachrichtigungen nur über Plug-In möglich
Datenaustauschformate	5	7	unterstützt JSON und XML
Single-Page Anwendung	5	2	nur möglich in Verbindung mit anderen Frameworks wie Backbone.js, Underscore.js
Programmiersprachen	5	5	unterstützt Dart, Typescript, HTML 5
Real-Time Collaboration	0	0	keine Funktionalität
			Score 3,714286
Summe funktionaler Anforderungen 24,05952			
Nichtfunktionale Anforderungen			
Reifegrad (33%)			
Feature	Gewichtung	Bewertung	Begründung

Release Date	5	9	17.09.2007, aktuelle Version 26.6.2014
Anzahl an Releases	5	8	Releases bei GitHub: 62
Anzahl an Entwicklern	5	9	GitHub Contributors: 246
Dauer zur Fehlerbehebung	5	6	vermutlich nicht besonders hoch, da sehr aktive und umfangreiche Communities, Blog-Einträge und regelmäßige Releases
Allgemein bekannte Probleme	5	5	keine speziellen aktuellen Probleme gefunden / vereinzelte Probleme ersichtlich allerdings keine schwerwiegenden
Supportmöglichkeiten	5	7	gute Bugtracking-Möglichkeiten, umfangreicher Support (Learning Center, Try jQuery, Chat, Forums)
Dokumentation	5	7	umfangreiche und übersichtliche Dokumentation
Score 12,02143			
Verbreitungsgrad (33%)			
Bewertung durch Anwender	5	9	5/5 bei chip.de, 4,5/5 bei Mozilla Add-Ons
Marktanteil	5	9	Ca. 80% Marktanteil, ca. 60% verwendet auf anderen Seiten
Score 14,85			
Nutzung (34%)			
Lizenzmodell	5	8	MIT Licence (kann vermarktet werden)
Usability	0	0	
Erweiterbarkeit	5	9	durch zahlreiche Plug-Ins erweiterbar (jQuery UI)
Score 14,45			
Summe nichtfunktionaler Anforderungen 41,32143			
Summe Gesamt 65,38095			

Die Ergebnisse der vorangegangenen Nutzwertanalyse für jQuery in Kombination mit jQuery UI zeigen auf, dass es sich bei diesem Framework um ein äußerst marktrelevantes Produkt handelt, welches sich besonders hinsichtlich des Reife- und Verbreitungsgrades auszeichnet. Beispielsweise wird das Framework von über 60% aller Internetseiten genutzt, ebenso arbeiten fast 250 Entwickler auf dem Open Source-Entwicklungsportal GitHub an der ständigen Weiterentwicklung und Verbesserung des Produktes. Betrachtet man die

funktionalen Anforderungen, wird deutlich, dass jQuery UI vor allem in Bezug auf die grafische Benutzeroberfläche sehr umfangreiche Funktionalitäten mit sich bringt. Konkret erhält das Framework in den Bereichen Benutzeroberfläche mit ca. 7,4 und Darstellungssteuerung mit 6 einen sehr hohen Score, während es in den Bereichen Datenmodell mit ca. 1,7 und Architektur mit ca. 3,7 eher schwach bewertet werden muss. Grund dafür ist, dass jQuery zusammen mit jQuery UI hauptsächlich auf die Erstellung von Benutzeroberflächen und Interaktionen ausgelegt ist, während stattdessen die übrigen Bereiche eher vernachlässigt werden. Um diese Schwächen zu umgehen, existiert eine Vielzahl von Plug-Ins, mit Hilfe derer sich jQuery UI sehr umfangreich erweitern lässt, wodurch der Funktionsumfang des Frameworks deutlich wächst. Zusammenfassend erhalten jQuery und jQuery UI einen Score von ca. 65.

4.1.3 Nutzwertanalyse von Backbone.JS und Underscore.JS

Funktionale Anforderungen			
Benutzeroberfläche (20%)			
Feature	Gewichtung	Bewertung	Begründung
Einfache Widgets	5	4	Widgets können hinzugefügt werden allerdings nur mittels Plug-Ins
Zusammengesetzte Widgets	5	3	Widgets können hinzugefügt werden allerdings nur mittels Plug-Ins
Containers	5	5	Als "landing-Place" für den templated Content / Event-Binding zu den Model-Daten darin / Rendering für Views möglich (abhängig von der Existenz von Parent- und Child-Views)
Visualisierung	5	3	Daten-Visualisierung möglich über das Add-On weiterer Frameworks
Themes	5	4	Einbindung möglich (UI-Notifications & Dialogs) über Plug-In Backbone.Notifier
Styles	5	0	keine CSS-Möglichkeiten
Event Handling	5	8	läuft über ein eingebautes Modul ab welches als simples Objekt dargestellt wird / durch Underscore können Backbone Events zu jedem Objekt hinzugefügt werden / Einfach Event-Support zu jedem Objekt hinzuzufügen

Score 3,85714			
Darstellungssteuerung (20%)			
Layout Manager	5	5	nachträglich hinzugefügt / gute Funktionalitäten und einfaches Set-Up / etwas beschränkter Funktionsumfang
Interaktionen	5	7	Routing wird übernommen, sodass das Interaktionsmodell angepasst wird und der aktuelle View geändert wird
Drawing	5	3	Möglichkeiten bestehen mit Backbone.js, allerdings kompliziert / nicht sonderlich ausgefeilt / aktuell keine Plug-Ins verfügbar
Theming	5	3	ebenfalls über Plug-In möglich (siehe Themes)
Templating	5	5	möglich durch die Anwendung von Underscore / Backbone selbst kann auch ohne Template verwendet werden / Platzhalter vorhanden werden mittels Routing-Prozess gefüllt
Visuelle Effekte	5	4	möglich über DOM-Manipulation und Anpassung von Views
Lokalisierung	5	3	Es existieren Möglichkeiten um Inhalt zu lokalisieren und zu übersetzen / wenig Infos verfügbar
Bedienmöglichkeiten	5	2	Über die Anbindung von APIs und Plug-Ins theoretisch möglich / allerdings nahezu keine Infos verfügbar
Browserunterstützung	5	8	Cross-Plattform Unterstützung
View Binding	5	5	läuft über change-events ab / allerdings keine besonders intensive Möglichkeiten
DOM Scripting	5	6	auslösen von Change-Events nach Änderungen an einem Objekt
Score 4,63636			
Geschäftslogik (20%)			
Zustandsmanager	5	5	Backbone.StateManager als Plug-In verfügbar
Modularität	5	8	sehr gute Möglichkeiten zur Strukturierung des Codes
Testing	5	5	möglich mittels einer Erweiterung durch weitere Frameworks wie

			(QUnit und SinonJS)
Multimedia	5	5	multimediale Anbindung ist möglich / zusätzliche Frameworks liefern erweiterte Funktionalitäten
Score 5,75			
Datenmodell (20%)			
Data Binding	5	8	eines der bekanntesten Frameworks für Data-Binding / allerdings kein Two-Way Data Binding
Datenobjekte	5	7	Objekte werden geändert sodass Views angeglichen werden können (möglich durch gutes Data-Binding)
Datenmodelle	5	7	Arbeitet mit Datenmodellen anstelle von DOM-Manipulation (ausführliche Daten-Modelle) / OOP-Möglichkeiten
Persistente Daten	5	6	verschiedene Persistenz Modelle möglich (vornehmlich Ajax)
Datenquellen	5	0	keine Informationen gefunden
Speichermethoden	5	8	local storage (Speicherung der Daten direkt im Browser möglich / keine Anbindung zum Server notwendig); aber auch Speicherung via Ajax möglich
Score 6			
Architektur (20%)			
Schnittstellen (APIs)	5	7	gute Kompatibilitätsmöglichkeiten für UI-orientierte Frameworks und Kommunikation mit Views
Server Calls	5	8	wichtig da eine dauerhafte Kommunikation bestehen muss um die Datenobjekte und Views anzupassen / Möglichkeit von asynchroner Serverkommunikation (Ajax)
Serverbenachrichtigung	5	5	Push-Notifications mittels Back-End Plug-In möglich
Datenaustauschformate	5	8	JSON, REST, AJAX
Single-Page Anwendung	5	7	Besonders gut geeignet für SPA
Programmiersprachen	5	5	JavaScript-Framework / kein CSS
Real-Time Collaboration	0	0	keine Real-Time Collaboration Fähigkeiten ersichtlich
Score 5,71429			

Summe funktionaler Anforderungen 25,9578			
Nichtfunktionale Anforderungen			
Reifegrad (33%)			
Feature	Gewichtung	Bewertung	Begründung
Release Date	5	8	13.10.2010/ aktueller Release (Februar 2014)
Anzahl an Releases	5	4	Releases by GitHub: 21
Anzahl an Entwicklern	5	8	GitHub Contributors: 229
Dauer zur Fehlerbehebung	5	6	vermutlich nicht besonders hoch, da sehr aktive und umfangreiche Communities, Blog-Einträge und regelmäßige Releases
Allgemein bekannte Probleme	5	5	keine speziellen aktuellen Probleme gefunden / vereinzelte Probleme ersichtlich allerdings keine schwerwiegenden
Supportmöglichkeiten	5	6	nicht besonders klar ersichtlich; verteilt (verschiedene Möglichkeiten auf Website + zusätzlichen Entwicklungsforen, gute Community etc)
Dokumentation	5	7	ausführliche Dokumentation und Erklärung der verschiedenen Fusionsmöglichkeiten allerdings fehlen Best Practices / Vielzahl an Büchern und Learning-Ressourcen
			Score 10,3714
Verbreitungsgrad (33%)			
Bewertung durch Anwender	5	8	"web's darling framework" / durchweg gute Benutzerbewertungen, besonders durch die Leight-Weight Eigenschaften und gute Fähigkeiten rund um Key-Components
Marktanteil	5	4	geringer Anteil an JavaScript Libraries (Backbone: 0,1% / Underscore 0,3%) / allerdings Verwendung auf sehr bekannten Webseiten
			Score 9,9
Nutzung (34%)			
Lizenzmodell	5	8	MIT-Lizenz

Usability	0		
Erweiterbarkeit	5	8	sehr gute Erweiterbarkeit durch zusätzliche Frameworks (besonders im Bereich UI) und eine Vielzahl an Plug-Ins
			Score 13,6
Summe nichtfunktionaler Anforderungen 33,8714			
Summe Gesamt 59,8292			

Wie die Auswertung des exemplarischen Vergleichs für das Framework Backbone.js in Kombination mit Underscore.js zeigt, liegen die Schwerpunkte dieses Frameworks in den Bereichen Datenmodell, Geschäftslogik und Architektur. Im Umkehrschluss bedeutet dies ebenfalls, dass eine Erweiterung über Plug-Ins oder weitere Frameworks notwendig ist, um Elemente der Darstellungssteuerung oder Benutzeroberfläche darzustellen.

Für Backbone.js gilt der Grundsatz „alles kann aber nichts muss“. Das bedeutet, dass backbone.js und die Erweiterung Underscore.js viele Komponenten bereits integriert haben, wodurch diese angewendet werden können. Allerdings besteht auch die Möglichkeit, die Komponenten auszulassen, falls diese nicht benötigt werden. Daher gilt es als eines der populärsten light-weight Frameworks am Markt. Es wird ebenfalls auf eine Vielzahl von bekannten Webseiten angewendet und wie sich ebenfalls zeigt, besitzt es eine große und aktive Entwickler-Community, was gut für Support und Fehlerbehebung ist.

In Kombination mit User Interface- und Darstellungs-Komponenten und Fähigkeiten anderer Frameworks bietet sich dem Entwickler einer mobilen Web-Anwendung ein umfangreiches Gerüst.

Da Backbone.js bereits sehr stark in den Bereichen des Datenmodells, Data-Binding, DOM-Manipulation und Scripting sowie einer möglichen Server-Anbindung ist und als eines der bekanntesten Frameworks für Single-Page-Applikationen gilt, ist es eines der meistverwendeten Frameworks für die Back-End Bereiche eines Web-Anwendungs-Frameworks.

4.2 Auswertung

Betrachtet man abschließend auf Basis der Gesamtscores alle drei Frameworks im Vergleich, so befindet sich SenchaTouch auf Platz 1, jQuery + jQuery UI auf Platz 2 und Backbone + Underscore auf Platz 3. Dies bestätigt zunächst die eingangs getroffenen

Erwartungen, da SenchaTouch die meisten Features adressiert. Deutlicher wird dies außerdem bei exklusiver Betrachtung der Auswertung von allen funktionalen Anforderungen. Durch die Funktionsvielfalt von SenchaTouch wird hier ein deutlich höherer Score erzielt.

Vergleicht man allerdings den deutlichen Vorteil von SenchaTouch gegenüber den Konkurrenzprodukten im Bereich der funktionalen Anforderungen mit der eher geringen Gesamtpunktedifferenz zwischen SenchaTouch und jQuery + jQuery UI, so wird klar, dass eine eins zu eins Berücksichtigung von funktionalen und nicht-funktionalen Anforderungen zu einer deutlichen Veränderung des Gesamtergebnisses führen kann. Es ist daher unter Umständen für eine Einschätzung der Rangfolge notwendig, nicht nur den Gesamtscore, sondern auch die Punktzahlen der einzelnen Bereiche zu berücksichtigen und zu hinterfragen. Ebenso besteht die Möglichkeit funktionale, bzw. nicht-funktionale Anforderungen nicht 1:1 zu gewichten. Diese wär über das Hinzufügen einer zusätzlichen Gewichtung der Gesamteilergebnisse möglich, was aber für den Rahmen dieser Ausarbeitung durch den wissenschaftlichen Betreuer nicht gewünscht wurde.

Innerhalb der funktionalen Anforderungen repräsentieren die Verteilungen der Bewertungspunkte außerdem die Stärken der jeweiligen Lösungen. Dies rechtfertigt weiterhin die Annahme, dass eine Kombination aus jQuery + jQuery UI und Backbone + Underscore zu einer Lösung führt, die durch die sich jeweils ergänzenden Kategorien einen höheren Gesamtscore als SenchaTouch erreichen kann. Hierbei handelt es sich um eine Kombination der, in ihren Kategorien bekanntesten Frameworks am Markt. Auch aus diesem Grund handelt es sich hierbei um eine in der Praxis durchaus mögliche und praktikable Lösung zur Entwicklung mobiler Web-Anwendungen.

5 Fazit

5.1 Zusammenfassung

Die innerhalb der Ausarbeitung durchgeführten Untersuchungen haben aufgezeigt, dass die Aktualität und zunehmende Relevanz von clientseitigen, browserbasierten Anwendungen zu einer komplexen heterogenen Marktstruktur in Bezug auf die entsprechenden Anwendungsframeworks führt. Konkret ist der Markt geprägt von einer Vielzahl an kleinen, spezialisierten Frameworks, die zum Teil nicht trennscharf von einer einfachen Bibliothek oder einem Plug-In unterschieden werden können. Dies führt gleichzeitig zu einer enormen Funktionsvielfalt, die bei der Analyse und Bewertung der Frameworks berücksichtigt werden muss. Grundsätzlich ist aber zu sagen, dass durch die offenen Strukturen der meisten Frameworks fehlende Funktionen einfach ergänzt werden können. Deutlich wurde im Rahmen der Recherche allerdings auch, dass einige marktdominierende Frameworks wie z.B. jQuery häufig die Grundlage für durch Plug-Ins erweiterte Frameworks bieten. Diese Kombinationen ermöglichen so eine Spezialisierung je nach Anwendungsfall.

Es erwies sich auf Grund dieser Gegebenheiten als praktikabel, die grundsätzliche Kategorisierung der Frameworks an einer MVC-Struktur zu orientieren, um somit ein Gerüst zu erstellen, in das die Menge an Funktionen der unterschiedlichen Frameworks sinnvoll eingegliedert werden konnte. Die im Rahmen dieser Arbeit entwickelte Nutzwertanalyse steht somit im Fokus der zu liefernden Arbeitsergebnisse, da sie zum einen die wesentlichen am Markt verfügbaren Funktionen abdeckt und zum anderen eine individualisierbare Nutzwertanalyse als Bewertungsgrundlage liefert. Die Arbeitsmappe stellt folglich eine konsequente Reaktion auf die komplexen Marktgegebenheiten dar.

Weiterhin wird durch die Arbeitsergebnisse auch die Möglichkeit des Stackings, also des Zusammenführens mehrerer Frameworks zu einer Gesamtlösung, adressiert. Was aus den vielen Varianten „jQuery plus weitere Frameworks“ erkenntlich wird ist, dass durch komplexe Varianten in den Anforderungen einzelne Framework-Lösungen diesen nur selten vollständig gerecht werden können bzw. Entwickler dazu tendieren, bereits vertraute Frameworks durch notwendige Plug-Ins zu erweitern. Unter Verwendung der Arbeitsmappe ist es möglich, eine parallele Darstellung mehrere spezialisierter Frameworks zu entwickeln, um in einer Kombination alle wichtigen Funktionen abzudecken.

Während der Recherche stellte sich die Beschaffung angemessener Fachliteratur als Herausforderung dar. Der Großteil der am Markt verfügbaren Quellen ist geprägt durch die Hersteller der jeweiligen Frameworks oder lediglich durch sporadische, wenn verfügbare, Dokumentationen abgedeckt.

5.2 Ausblick

Die Ausarbeitungen zeigen, dass für die meisten Entwickler die Marschrichtung fest steht, wenn es um die Entwicklung von funktionsreichen Web-Anwendungen geht. Es gibt einen weitgehend akzeptierten Konsens in der Form von MVC bzw. MVVM Architekturen, was dazu führt, dass die Vorteile von clientseitigen Browseranwendungen konsistent und unabhängig von der Wahl des Frameworks gegeben sind. Der grundlegende Unterschied im Hinblick auf die Architektur ist daher viel mehr durch die Entscheidung für ein umfangreiches Framework oder verschiedene Bibliotheken begründet.

Die Entwicklung im Bereich der Frameworks ist untrennbar mit dem Wandel der zur Verfügung gestellten Services verbunden. Der Einfluss des Umfangs von Inhalten, serverseitigen Ressourcen und bedienten Clients wird nach wie vor die Anwendung der clientseitigen Technologien beeinflussen. Der Trend geht allerdings zu schnelleren Browsern so wie leistungsfähigeren JavaScript-Engines – dies wirkt sich demnach auch direkt auf die Verbreitung der entsprechenden Frameworks aus.

Unklar ist, ob diese Entwicklung zum Fall von serverseitigen Frameworks oder zum Aufstieg serverseitiger JavaScript-Frameworks führt. Twitter überführte beispielsweise 2013 das Frontend vom Client an den Server, da server-rendering nach wie vor Geschwindigkeitsvorteile bietet. Was allerdings in jedem Fall deutlich wird, ist die zunehmende Trennung von Frontend-Code zu Backend-Code.¹³⁰

In Bezug auf clientseitige Frameworks ist aus der Marktübersicht ein Trend zu Lightweight-Frameworks deutlich geworden, die in der Lage sind eine native Haptik und Optik durch Single-Page-Anwendungen zu ermöglichen, ohne sich Auflagen der nativen App-Stores zu unterwerfen oder in verschiedenen Sprachen entwickelt werden zu müssen.

¹³⁰ Vgl. howtoboss.com (2013)

Quellenverzeichnisse

Literaturverzeichnis

- Aktiv Software GmbH (o.J.)** Webapplikationen, Aktiv Software GmbH
programmierter Erfolg, o.J., o.O.
- Flanagan, D. (2006):** JavaScript - The Definitive Guide, 5te Auflage,
o.J., O'Reilly: Sebastopol
- Grüneberger, F. J. (2012):** REAL-TIME COLLABORATION SUPPORT FOR
JAVASCRIPT FRAMEWORKS, Faculty of
Computer Science, o.J., Dresden: Technische
Universität Dresden
- Hansen, R./ Neumann, G. (2002):** Wirtschaftsinformatik I – Grundlagen
betrieblicher Informationsverarbeitung, 8.
Auflage, Stuttgart: Lucius & Lucius
- Kim, Y. / Lee, K. (2009):** A lightweight framework for mobile web services,
6.02.2009, Seoul: Springer-Verlag
- Kouki, J. / Chainbi, W. / Ghedira, K. (2014):** A novel framework for bindings
synchronization of Web services, 3.10.2013,
London: Springer-Verlag
- Lin, F. / Shaw, M. J. / Chuang M. Y. (2005):** A unified framework for managing Web-
based services, 6.10.2005, Illinois Taiwan:
Springer-Verlag
- Metz, C. (2003):** Dynamische Webseiten, in Seminar
Medientechnik, 30.06.2003, o.O.
- Sencha Inc. (2014):** The Modern Web Stack A Taxonomy of Front-
End Technologies As An Aid To Decision Making,
o.J., Redwood City: Sencha Inc.
- Smith, D. / Cearley, D. / Prentice, B. (2013):** HTML5 and Web Technologies Are
Becoming Major Forces in Mobile Application
Development, 24.01.2013, Stanford: Gartner

- Stavroulaki, V. u.a. (2006):** Distributed Web-based Management Framework for Ambient Reconfigurable Services in the Intelligent Environment, Mobile Networks and Applications, o.J., Greece: Springer Science + Business Media
- Züger, D. (2010):** HTML 5 – Überblick und Vergleich, Universität Fribourg Wirtschaft und Sozialwissenschaftliche Fakultät, Fribourg, CH

Verzeichnis der Magazine und Whitepaper

- Wähner, K. (2011):** Webframeworks – Eine Kategorisierung, in: Java Magazin, 2011, Heft 1, S. 34-42
- Brian, D./Knoernschild, K. (2013):** Modern Web App Architecture, in: Gartner – Technical Professional Advice, 26.09.2013, Stamford: Gartner

Verzeichnis der Internet- und Intranet-Quellen

- 4managers (Hrsg.) (o.J.):** Nutzwertanalyse - Wieviel Glück bringt das Produkt?,
<http://4managers.de/management/themen/nutzwertanalyse/>,
 Abruf: 24.05.2014
- Aaken, G. van (2006):** Was ist DOM-Scripting, Webkrauts,
<http://webkrauts.de/artikel/2006/was-ist-dom-scripting>, Abruf: 02.07.2014
- Android Developer (o.J.):** Custom Components,
<http://developer.android.com/guide/topics/ui/custom-components.html#basic>, Abruf: 30.06.2014
- Apache Software Foundation (2004):** Apache License, Version 2.0,
<http://www.apache.org/licenses/LICENSE-2.0.html>, Abruf: 09.07.2014

ASP.NET Resources (2004):	Skinning Web Applications With CSS, http://aspnetresources.com/articles/SkinningWithCSS , Abruf: 29.06.2014
Backbonejs.org (2014):	backbone.js, http://backbonejs.org/ , Abruf: 08.07.2014
Bada Developer (o.J.):	UI: Creating the Application UI Using UI Controls, https://developer.bada.com/help/index.jsp?topic=/com.osp.webappprogramming.help/html/dev_guide/ui/ui_namespace_web.htm , Abruf: 30.06.2014
Blog-Botfrei (2013):	Know-How: Was ist ein Webserver, http://blog.botfrei.de/2013/01/webserver-die-qualder-wahl/ , Abruf 03.06.2014
Controlling-Portal (Hrsg.) (2014):	Die Nutzwertanalyse, http://www.controllingportal.de/Fachinfo/Grundlagen/Die-Nutzwertanalyse.html , Abruf: 24.05.2014
D’Arcy from Winnipeg (2009):	MVVM compared to MVC and MVP, http://geekswithblogs.net/dlussier/archive/2009/11/21/136454.aspx , Abruf: 20.06.2014
Der Dualstudent (Hrsg.) (o.J.):	Definition: Persistenz, http://www.derdualstudent.de/definition-persistenz-informatik.html , Abruf: 01.07.2014
docforge.com (o.J):	Web application framework,: http://docforge.com/wiki/Web_application_framework , Abruf: 04.06.2014
Drupal (2007):	Drawing API, https://www.drupal.org/project/drawing , Abruf: 01.07.2014
dr.web (2003):	CSS Container – Grundlagen, http://www.drweb.de/magazin/css-container-grundlagen/ , Abruf: 30.06.2014

- embarcadero (2012):** Component Events,
http://docwiki.embarcadero.com/HTML5_Builder/en/Component_Events, Abruf: 30.06.2014
- Exploring Java (o.J.):** Chapter 12 - Layout Managers,
http://docstore.mik.ua/oreilly/java/exp/ch12_01.htm, Abruf: 01.07.2014
- Expression Web Tutorial (2014):** 3 Types of CSS Styles, http://www.expression-web-tutorial.com/Types_CSS_Styles.html#.U7nV4xCk3K8, Abruf: 30.06.2014
- Fern-Uni Hagen (2014):** Einführung in Java-Web-Anwendungen: Was sind Web-Anwendungen?,
http://www.isdb.fernuni-hagen.de/wbt/files/demo/jsp/JSP/Kursseite_35130.htm, Abruf: 03.06.2014
- Friedman, V. (2007):** Data Visualization: Modern Approaches,
<http://www.smashingmagazine.com/2007/08/02/data-visualization-modern-approaches/>, Abruf: 01.07.2014
- Gabler Wirtschaftslexikon (Hrsg.) (o.J.):** Nutzwertanalyse,
<http://wirtschaftslexikon.gabler.de/Definition/nutzwertanalyse.html>, Abruf: 24.05.2014
- Gabler Wirtschaftslexikon (o.J.):** Datenmodelle,
<http://wirtschaftslexikon.gabler.de/Definition/datenmodell.html>, Abruf: 29.06.2014
- Gagern, S. von (2013):** Native vs. Web Apps vs. Hybrid: Was ist die perfekte Entwicklernstrategie?,
<https://www.developergarden.com/de/blog/artikel/article/native-vs-web-app-vs-hybrid-was-ist-die-perfekte-entwicklerstrategie/>, Abruf: 21.06.2014
- GNU (2014):** GNU GPLv3: Eine Kurzanleitung,
<http://www.gnu.org/licenses/quick-guide-gplv3>, Abruf: 08.07.2014

- Grammiweb (2014):** Was ist JavaScript,
<http://www.grammiweb.de/informativ/grundlagen/wasistjava.shtml>, Abruf: 03.06.2014
- GWTproject.org (o.J.):** Choosing an Implementation Strategy,
http://www.gwtproject.org/doc/latest/tutorial/client_server.html, Abruf: 29.06.2014
- Heilmann, C. (2007):** Event-Driven Web Application Design,
<http://www.yuiblog.com/blog/2007/01/17/event-plan/>, Abruf: 29.06.2014
- How-Stuff-Works (2012):** How Webservers Work,
<http://computer.howstuffworks.com/web-server2.htm>, Abruf: 04.06.2014
- howtoboss.com (2013):** The Fall of Server Side Web Frameworks,
<http://howtoboss.com/2013/10/08/the-fall-of-server-side-web-frameworks/>, Abruf: 09.07.2014
- Html5rocks.com (o.J.a):** HTML5 Features Storage,
<http://www.html5rocks.com/en/features/storage>,
Abruf: 29.06.2014
- Html5rocks.com (o.J.b):** Client-Side Storage,
<http://www.html5rocks.com/en/tutorials/offline/storage/>, Abruf: 29.06.2014
- Html5rocks.com (o.J.c):** Einführung zu WebSockets – Sockets im Web,
<http://www.html5rocks.com/de/tutorials/websockets/basics/>, Abruf: 29.06.2014
- HTML.net (o.J.):** Lektion 1: Was ist CSS,
<http://de.html.net/tutorials/css/lesson1.php>, Abruf: 03.06.2014
- HTML-Seminar (o.J.):** HTML-Grundlagen: Was ist HTML,
<http://www.html-seminar.de/html-grundlagen.htm>,
Abruf: 03.06.2014
- IBM (o.J.):** Event handling in Rich UI,
<http://pic.dhe.ibm.com/infocenter/rbdhelp/v8r0m0/index.jsp?topic=%2Fcom.ibm.egl.lr.doc%2Ftopic>

- s%2Fregl_ui_richui_event_handling.html, Abruf: 30.06.2014
- Innopact (2014):** Rich-Internet-Applications, https://www.innopact.de/Web-Anwendungen/rich_internet_applications.html, Abruf: 02.07.2014
- IT Wissen (o.J.):** Benutzeroberfläche, http://www.itwissen.info/definition/lexikon/Benutz_eroberflaeche-UI-user-interface.html, Abruf: 09.07.2014
- Kobes, R. (2003):** Simple Widgets, <http://theory.uwinnipeg.ca/perlTk/widgets/>, Abruf: 01.07.2014
- Koebler, R. (2013):** Template Engines, <http://www.simple-is-better.org/template/>, Abruf: 01.07.2014
- Lauke, P. H. (2012):** Touch-Events: Webseiten zum Anfassen, Webkrauts, <http://webkrauts.de/artikel/2012/einfuehrung-touch-events>, Abruf 01.07.2014
- Manager wiki (Hrsg.) (2013):** Nutzwertanalyse – ein Verfahren zur Bewertung verschiedener Optionen, <http://www.manager-wiki.com/methodik/57-nutzwertanalyse>, Abruf: 24.05.2014
- Microsoft.com (2014a):** Data Binding Overview, <http://msdn.microsoft.com/en-us/library/ms752347%28v=vs.110%29.aspx>, Abruf: 29.06.2014
- Microsoft.com (2014b):** Übersicht über Datenquellen, <http://msdn.microsoft.com/de-de/library/8s2t4x15.aspx>, Abruf: 29.06.2014
- Microsoft.com (2014c):** An Introduction to JavaScript Object Notation (JSON) in JavaScript and .NET, <http://msdn.microsoft.com/en-us/library/bb299886.aspx>, Abruf: 29.06.2014

- Microsoft.com (2012d):** Detecting devices and their features,
<http://msdn.microsoft.com/en-us/library/jj149688.aspx>, Abruf: 21.06.2014
- Mosers, C. (2011):** The Model-View-ViewModel Pattern,
<http://wpftutorial.net/MVVM.html>, Abruf: 20.06.2014
- Mozilla (o.J.):** WebAPI, <https://wiki.mozilla.org/WebAPI>, Abruf: 29.06.2014
- MyWebcheck.de (2014):** Datenaustauschformate,
<http://www.mywebcheck.de/datenaustauschformate/>, Abruf: 29.06.2014
- Northover, S./MacLeod, C. (2001):** Creating Your Own Widgets,
<https://www.eclipse.org/articles/Article-Writing%20Your%20Own%20Widget/Writing%20Your%20Own%20Widget.htm>, Abruf: 30.06.2014
- Open Source Initiative (o.J. a):** The BSD 2-Clause License,
<http://opensource.org/licenses/BSD-2-Clause>, Abruf: 08.07.2014
- Open Source Initiative (o.J. b):** The MIT License,
<http://opensource.org/licenses/mit-license.php>, Abruf: 08.07.2014
- Osmani, A. (2012):** Journey Through JavaScript MVC Jungle,
<http://www.smashingmagazine.com/2012/07/27/journey-through-the-javascript-mvc-jungle/>, Abruf: 20.06.2014
- Pimcore.org (2012):** Data Objects,
<http://www.pimcore.org/wiki/display/PIMCORE/Data+Objects>, Abruf: 29.06.2014
- Proggen.org (o.J.):** Grafische Einheiten (Widgets),
<http://www.proggen.org/doku.php?id=frameworks:qt:gui:basic:widget>, Abruf: 01.07.2014
- RWE Einkauf (Hrsg.) (o.J.):** Lieferantenbewertung,
<https://www.rwe.com/web/cms/de/90028/lieferanten/lieferantenmanagement/ueberblick->

- lieferantenmanagement/lieferantenbewertung/,
Abruf: 02.06.2014
- Schwichtenberg, Dr. H. (o.J. a):** Erklärung des Begriffs: Mehrschichtmodell, IT-Visions, <http://www.it-visions.de/glossar/alle/4513/Mehrschichtmodell.aspx>, Abruf: 01.07.2014
- Schwichtenberg, Dr. H. (o.J. b):** Erklärung des Begriffs: Benutzerschnittstellensteuerung (BSS), IT-Visions, [http://www.it-visions.de/glossar/alle/6175/\[TITEL\].aspx](http://www.it-visions.de/glossar/alle/6175/[TITEL].aspx), Abruf: 01.07.2014
- Siegert, Dr. P. F. (o.J.):** Was ist ECMAScript (JavaScript), <http://dok.uni-lueneburg.de/joomla/index.php/javascript-4sem/was-ist-javascript>, Abruf: 03.06.2014
- Springer Gabler (2011):** Gabler Wirtschaftslexikon, Browser, von Prof. Dr. Richard Lackes, <http://wirtschaftslexikon.gabler.de/Definition/browser.html>, Abruf: 03.06.2014
- Stackoverflow (2013):** Client Side Web App Development, <http://stackoverflow.com/questions/18612437/client-side-web-app-development>, Abruf: 04.06.2014
- Underscorejs.org (2014):** underscore.js, <http://underscorejs.org/>, Abruf: 08.07.2014
- Welt der BWL (Hrsg.) (o.J.):** Nutzwertanalyse, <http://www.welt-der-bwl.de/Nutzwertanalyse>, Abruf: 24.05.2014
- W3.org (2014):** HTML Nightly, <http://www.w3.org/html/wg/drafts/html/master/Overview.html>, Abruf. 29.06.2014
- W3.org (1997):** Client-side Scripting and HTML, <http://www.w3.org/TR/WD-script-970314>, Abruf. 29.06.2014

- W3-Schools (2014a):** Introduction to XML,
http://www.w3schools.com/xml/xml_whatism.asp,
Abruf: 03.06.2014
- W3-Schools (2014b):** JavaScript Tutorial,
<http://www.w3schools.com/js/DEFAULT.asp>,
Abruf: 03.06.2014
- W3-Schools (2014c):** Introduction to Ajax,
http://www.w3schools.com/ajax/ajax_intro.asp,
Abruf: 04.06.2014
- W3-Schools (2014d):** PHP 5 Introduction,
http://www.w3schools.com/php/php_intro.asp,
Abruf: 04.06.2014
- W3-Techs (2013):** Usage of server-side programming languages for websites,
http://w3techs.com/technologies/overview/programming_language/all, Abruf: 04.06.2014
- Webmasterpro (2014):** Was ist CSS,
<http://www.webmasterpro.de/coding/article/css-einfuehrung-was-ist-css.html>, Abruf: 03.06.2014
- Wigmore, I. (2013):** Business logic,
<http://whatis.techtarget.com/definition/business-logic>, Abruf: 15.06.2014
- Wikipedia (2014a):** Webserver,
<http://de.wikipedia.org/wiki/Webserver>, Abruf: 03.06.2014
- Wikipedia (2014b):** JavaScript,
<http://de.wikipedia.org/wiki/JavaScript>, Abruf: 03.06.2014
- Wikipedia (2014c):** Ajax (Programmierung) – Vergleich mit herkömmlichen Web-Anwendungen,
[http://de.wikipedia.org/wiki/Ajax_\(Programmierung\)#Vergleich_mit_herk.C3.B6mmlichen_Web-Anwendungen](http://de.wikipedia.org/wiki/Ajax_(Programmierung)#Vergleich_mit_herk.C3.B6mmlichen_Web-Anwendungen), Abruf: 04.06.2014

- Wikipedia (2014d):** PHP, <http://de.wikipedia.org/wiki/PHP>, Abruf: 04.06.2014
- Wikipedia (2014e):** UI data binding, http://en.wikipedia.org/wiki/UI_data_binding, Abruf: 02.07.2014
- Wills, A. (2014):** Top 5 HTML5 Mobile UI Frameworks For HTML5 Powered Apps, <http://ethicalweb.blogspot.de/2014/03/top-5-html5-mobile-ui-frameworks-for-html5-apps.html>, Abruf: 29.06.2014
- Yii Framework (2011):** Theming, <http://www.yiiframework.com/doc/guide/1.1/en/to-pics.theming>, Abruf: 01.07.2014
- Yii Framework (o.J.):** Themes, <http://www.yiiframework.com/doc/guide/1.1/de/to-pics.theming>, Abruf: 30.06.2014
- Xi-Interactive (2014):** Internet, Intranet, Extranet, <http://www.xiinteractiv.com/internet-web-sites.cfm>, Abruf: 03.06.2014

Open Source Workflow-Management-Systeme

Schriftliche Ausarbeitung
im Rahmen der Lehrveranstaltung „Integrationsseminar“
Kompetenzzentrum Open Source (KOS)

Vorgelegt von

Anastasiia Shelukhina,
Laura Schreiber,
Nadine Harff

am 11.07.2014

Fakultät Wirtschaft
Studiengang Wirtschaftsinformatik
International Management for Business and Information Technology

Inhaltsverzeichnis

Abkürzungsverzeichnis	III
Abbildungsverzeichnis.....	V
Tabellenverzeichnis.....	V
1 Einleitung (Shelukhina, Schreiber, Harff).....	1
2 Grundlagen (Shelukhina, Schreiber, Harff).....	2
2.1 Workflow Management.....	2
2.2 Technischer Aufbau eines Workflow-Management-Systems	4
2.3 Open Source.....	6
2.3.1 Definition.....	6
2.3.2 Historische Entwicklung	8
2.3.3 Open-Source Lizenz Modelle	10
3 Exemplarische Vorstellung verschiedener Open Source Workflow-Management Systeme (Shelukhina, Schreiber, Harff).....	11
3.1 Activiti	13
3.2 Bonita BPM 6	15
3.3 Camunda BPM platform.....	16
3.4 Enhydra Shark	17
3.5 Imixs Managed Workflow Server	18
3.6 JBPM	19
3.7 ProcessMaker Open Source	20
3.8 Taverna.....	21
3.9 Workflow Server.....	21
4 Zusammenfassung kommerzieller Workflow-Management-Systeme (Shelukhina, Schreiber, Harff).....	22
5 Bewertungskriterien (Shelukhina, Schreiber, Harff)	24
5.1 Vorstellung der Bewertungskriterien.....	24
5.2 Bewertungstabelle.....	31
6 Installation und Beispielprozess eines Workflow-Management-Systems am Beispiel von Camunda BPM (Shelukhina, Schreiber, Harff)	33
7 Fazit (Shelukhina, Schreiber, Harff).....	37
Quellenverzeichnisse (Shelukhina, Schreiber, Harff).....	38

Abkürzungsverzeichnis

API	Application Programming Interface
BPEL	Business Process Execution Language
BPMN	Business Process Model and Notation
BSD	Berkely Software Distribution
CDI	Contexts and Dependency Injection
ERP	Enterprise Resource Planning
GPL	General Public License
GUI	Graphical User Interface
HTML	Hypertext Markup Language
IBM	International Business Machines
IDE	Integrated Development Environment
JDBC	Java Database Connectivity
JPA	Java Persistence API
JSF	JavaServer Faces
JTA	Java Transaction API
KPI	Key Performance Indicator
LDAP	Lightweight Directory Access Protocol
MS	Microsoft
MySQL	My Structured Query Language
OpenJDK	Open Java Development Kit
OSGi	Open Services Gateway Initiative
PDF	Portable Document Format
PHP	Hypertext Preprocessor
REST	Representational State Transfer

SaaS	Software as a Service
WebDAV	Web-based Distributed Authoring and Versioning
WfMC	Workflow Management Coalition
WfMS	Workflow-Management-System
XML	Extensible Markup Language
XPDL	XML Process Definition Language
XSLT	Extensible Stylesheet Language Transformations

Abbildungsverzeichnis

Abbildung 1: Systemaufbau eines Workflow-Management-Systems	5
Abbildung 2: Activiti Components	14
Abbildung 3: Gartner Magic Quadrant for Intelligent Business Process Management Suites 2014	23
Abbildung 4: Ovum Decision Matrix for Selecting a Business Process Management Solution 2014	24
Abbildung 5: Beispiel Bewertungstabelle	32
Abbildung 6: Eigenschaften der Schritte	34
Abbildung 7: Eigenschaften des gesamten Prozesses	34
Abbildung 8: Camunda BPM Platform Startseite	35
Abbildung 9: Loan Approval Process	36

Tabellenverzeichnis

Tabelle 1: Open Source Lizenz Kategorien	11
Tabelle 2: Vergleich Open Source vs. Enterprise Edition	17

1 Einleitung

Heutzutage muss jede Gruppe verwaltet werden. Gerade in westlichen Staaten ist dies oft auch mit einem administrativen Mehraufwand verbunden. Allerdings hat es sich empirisch erwiesen, dass eine Struktur der Prozesse und Arbeitsabläufe helfen kann, diese zu optimieren, Zeitaufwand und Kosten zu reduzieren und gegebenenfalls auch Qualität zu verbessern. Deshalb unterstützen Workflow-Management Systeme (WfMS) Gruppen bei der Erledigung ihrer Aufgaben. Innerhalb eines privaten Haushalts wird eine IT-gestützte Verwaltung noch nicht angewendet (wobei sich das im Laufe der Zeit durch das Internet der Dinge durchaus irgendwann ändern könnte). Jedoch greifen Unternehmen, gemeinnützige und nichtstaatliche Organisationen, sowie Institutionen, zum Beispiel Bildungseinrichtungen, staatliche Galerien, Museen etc. darauf zurück. Im Laufe dieser Arbeit werden die Begriffe Organisation, Unternehmen und Firma im Sinne des WfMS-Nutzers durchaus synonym verwendet, da sie für alle gleichermaßen relevant sein können.

Ein Workflow-Management-System ist ein (meist java-basierendes System), welches die Arbeitsschritte in einer Organisation abbildet, simuliert und durchführt. Es kann die Logik überprüfen und verschiedene Schritte automatisieren. So würde ein WfMS an einer Hochschule zum Beispiel dafür sorgen in einer Datenbank nachzuschauen, ob die Studenten den Universitätsbeitrag geleistet haben, ggfs. automatische Erinnerungsmails rauschicken, das Druckkontingent der Studenten und Angestellten überwachen, zur Noteneingabe auffordern etc. Dadurch wird der Arbeitsaufwand der Mitarbeiter verringert, bzw. verlagert. Eine Transparenz wird geschaffen, die es einen erleichtert Fehler zu erkennen, zu beheben und einzelne Prozessschritte zu verbessern. WfMS werden von kommerziellen Unternehmen wie International Business Machines (IBM) oder PegaSystems, aber auch von Open Source Unternehmen wie Activiti oder BonitaSoft als Basis-Freeware, bzw. entgeltliche erweiterte Version angeboten. Diese Arbeit soll zukünftige WfMS-Nutzer bei der Auswahl des richtigen Produkts unterstützen.

Hierzu werden zunächst die Begriffe Workflow-Management und Open Source qualitativ erläutert, um ein erstes Grundverständnis zu schaffen. Anschließend wird eine Marktanalyse der kommerziellen und Open Source Produkte durchgeführt, damit ein Überblick entsteht, welches Produkt momentan Marktführer ist und am meisten genutzt wird. Daraufhin werden verschiedene, ins Besondere Open Source Produkte detaillierter vorgestellt. Um die Aktivität der Open Source Unternehmen herauszufinden, wird ohloh.net, ein öffentliches Verzeichnis von Freeware und Open Source Programmen herangezogen werden. Anhand der bisherigen gesammelten Informationen sollen dann Kriterien herausgefiltert und begründet werden, die bei der Auswahl eines Systems zu beachten sind. Daraufhin werden exemplarisch zwei der vorher vorgestellten Open Source Systeme ausgewählt und anhand der Kriterien in einen

eigens dafür vorher im Zusammenhang mit dem Kriterienkatalog erstellten Bewertungsbogen zu beurteilen. Nach einer Auswertung des Vergleichs wird noch ein System anhand eines exemplarischen Prozess demonstriert. Zum Schluss werden die Ergebnisse der Arbeit zusammengefasst.

2 Grundlagen

2.1 Workflow Management

Der Harvard-Professor Michael E. Porter sagte einmal (1999): „Jedes Unternehmen ist eine Ansammlung von Tätigkeiten, durch die sein Produkt entworfen, hergestellt, vertrieben, ausgeliefert und unterstützt wird.“¹

Schewe (w. y.) formuliert Workflow Management im Gabler Wirtschaftslexikon als „Analyse, Modellierung, Simulation, Steuerung und Protokollierung von Geschäftsprozessen (Workflow) unter Einbeziehung von Prozessbeteiligten und (elektronischen Systemen (Workflow Management System)).“² Dementsprechend zielt Workflow Management darauf ab, die Anzahl der Leute, die unter Umständen räumlich verteilt zur Lösung von Teilaufgaben eines Prozesse beisteuern zu koordinieren und den Prozessverlauf und Bearbeitungsstatus zu kontrollieren.³ Die deutsche Bezeichnung für Workflow ist Arbeitsablauf oder Arbeitsfluss. Beim Workflow Management geht es also darum einzelne (operative) Arbeitsschritte eines Geschäftsprozesses genau zu beschreiben, zu unterstützen und transparenter zu machen, um letztlich ihre Optimierung zu erreichen.⁴ Ein Geschäftsprozess wird von einem Kunden angestoßen und am Ende an einen Kunden geliefert. Der kann hierbei extern oder intern sein. Ein Prozess ist dynamisch und messbar und dient der Herstellung eines Produkts oder einer Dienstleistung, den administrativen, wirtschaftlichen oder technischen Gewinn oder Fortschritt einer Organisation. Er funktioniert nach „Input-Throughput-Output“-Prinzip; das heißt es wird etwas reingegeben, verarbeitet und es kommt etwas heraus. Die Schritte werden solange wiederholt, bis das Produkt oder die Dienstleistung lieferbar ist, egal ob es dabei um einen für einen Kunden bestimmten Fernseher oder eine Gehaltsabrechnung eines Angestellten geht. Der Output der einen Aktivität, ist normalerweise der Input der nächsten. Ein Workflow ist die Umsetzung eines Geschäftsprozesses in einzelnen Schritten und beinhaltet die notwendigen Schnittstellen, sowie IT-Unterstützung. Er zeigt ein wie, womit und

¹ Porter, M. E. (1999), S. 67

² Schewe, G (o.J.)

³ Vgl. Ibidem

⁴ Vgl. Helbig, L./ Zhang, J./ Zierke, M. E. (2005), S. 1

was genau im übergeordneten Geschäftsprozess umgesetzt wird. Eine Optimierung kann durch verschiedene Modellierungen erreicht werden, so kann man beispielsweise Schritte parallelisieren, weglassen, auslagern, hinzufügen, zeitlich verkürzen oder deren Reihenfolge ändern. Letztlich muss die Prozessqualität gewährleistet sein, da davon Kundenzufriedenheit und in dem Zusammenhang auch Umsatz und Gewinn abhängen. Oft soll durch Prozessoptimierung auch eine Kostenreduzierung erfolgen.

Man kann Workflows in unterschiedliche Klassifizierungen unterteilen.

So gibt es z.B. den Ad Hoc- Workflow, der sich dadurch auszeichnet, dass er nicht modellierbar ist. Er bietet kaum Automatisierungsmöglichkeiten und ist weder wiederhol- noch strukturierbar. Des Weiteren fehlt die direkte Kontrolle des weiteren Ablaufs sowie direkte Überwachungs- und Eingriffsmöglichkeiten. Ein Beispiel dafür ist ein Telefonanruf.

Der flexible Workflow weist bereits eine geringe Strukturierung auf. Dementsprechend ist er auch geringfügig wiederhol- und automatisierbar. Man kann laufende Vorgänge ändern und eine Zeitplanung, -überwachung und hierarchische Struktur festlegen.

Als dritte Klassifizierung erkennen Helbig, Zhang und Zierke (2005) den strukturierten Workflow an. Wie der Name bereits vermuten lässt herrscht hier ein hoher Strukturierungsgrad, welches eine hohe Wiederholbarkeit und Automatisierbarkeit gewährleistet. Es gibt Überwachungs- und Eingriffsmöglichkeiten. Allerdings sind durch die Automatisierbarkeit keine Änderung zur Laufzeit mehr vorgesehen.⁵

Die Arbeitsschritte werden meist in Workflow Management Systemen abgebildet. Diese dienen zur Modellierung, Instanziierung, Transparenz, Simulation und Überwachung. Häufig sind sie Bestandteil von Enterprise-Resource-Planning- oder Content-Management Systemen.⁶ Ein WfMS dient also zur Planung, Steuerung und Protokollierung der Geschäftsprozesse und somit betrieblicher Abläufe. Es koordiniert zwischen den verschiedenen Mitarbeitern und Personen je nachdem, welche Rolle sie in der Organisation einnehmen und wie ihre Abhängigkeit zum Ablauf aussieht.⁷ Elementare Aktivitäten bilden die Grundlage des Arbeitsablaufs. Es gibt verschiedene Prozessausführungssprachen und Standards, um die Schritte zu modellieren. Die fachlichen Ansätze werden meist in der Business Process Model and Notation (BPMN) dargestellt, die von der Object Management Group durchgesetzt wurde. Damit das Notierte auch ausgeführt werden kann, benötigt man Sprachen wie Business Process Execution Language (BPEL) oder XML Process Definition Language (XPDL).⁸ Bei

⁵ Vgl. Helbig, L./ Zhang, J./ Zierke, M. E. (2005), S. 2 et seqq.

⁶ Vgl. Helbig, L./ Zhang, J./ Zierke, M. E. (2005), S. 1

⁷ Vgl. Böhm, M./ Jablonski, S. (1997); Müller, J. (2005)

⁸ Vgl. Karagiannis, D. (2013)

den Standards bildet die Workflow Management Coalition auch noch eine besondere Rolle, auf die später eingegangen wird. Ein WfMS muss die Arbeitsschritte automatisieren können, mit Menschen interagieren und mit anderen Systemen verbunden werden und zusammenarbeiten können.

Wie die Optimierungsmöglichkeiten bereits erahnen lassen nutzt man Workflow-Management zur Schaffung von Transparenz der Arbeitsschritte. Dies soll zum einen Mängel aufdecken, die man anschließend beheben kann, aber auch sonstige Qualitätsverbesserungspotentiale aufzeigen. Ein WfM-System hilft dabei, in dem man jede Verknüpfung eines Arbeitsschrittes aufzeigen muss und dabei den Schritt an sich noch mal von Grund auf hinterfragt. Außerdem wird es durch die Abbildung deutlich, wenn ein Fehler gehäuft an einer Stelle auftritt. Durch eine Durchlaufzeitverkürzung werden auch Prozesskosten gesenkt, insbesondere dadurch, dass mit Hilfe des Systems genauer ermittelt werden kann, welcher Schritt wie lange dauert, welches Automatisierungspotential er hat und somit auch ggf. als Konsequenz daraus Mitarbeiterkosten (zum Beispiel durch verbesserte Ressourcenplanung) gesenkt werden können. Natürlich ist auch die Anpassung an organisatorische Veränderungen vereinfacht, da bereits ein Überblick besteht. Letztlich werden zum leichteren Bearbeiten gemeinsame Benutzerschnittstellen geschaffen. Schlussendlich dienen all diese Aspekte, wie vorhin erwähnt, der erhöhten Kundenzufriedenheit durch verbesserte Leistung und mehr Information und somit dem Umsatz und Gewinn.

1993 wurde die Workflow Management Coalition (WfMC) gegründet, deren Mitglieder vor allem aus Entwicklern, Beratern, Analysten, Benutzern und Unternehmen bestehen. Ihr primäres Ziel ist es auf die Vorteile von Workflow Management aufmerksam zu machen und Standards durchzusetzen. Von ihnen stammen auch die weitverbreiteten Sprachen der Workflow Management Systeme Wf-XML und XPD. ⁹

2.2 Technischer Aufbau eines Workflow-Management-Systems

Obwohl so viele verschiedene Workflow-Management-Systeme auf dem Markt angeboten werden, ist ein Großteil von ihnen sehr ähnlich aufgebaut. In Abbildung 1 wird der herkömmliche Systemaufbau eines WfMS veranschaulicht.

⁹ Vgl. WfMC (o.J.)

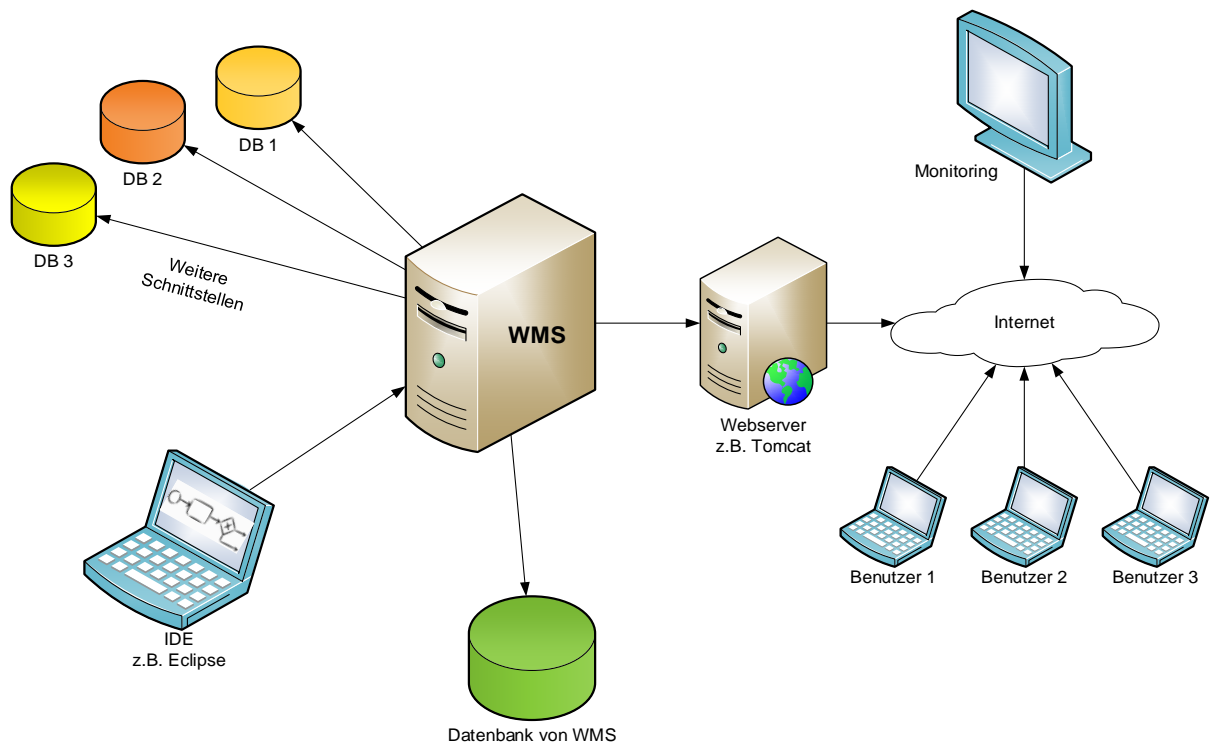


Abbildung 1: Systemaufbau eines Workflow-Management-Systems

Wie es aus der Abbildung ersichtlich ist, wird ein Workflow-Management-System auf einem Rechner bzw. Server installiert. Der Hauptteil des Systems ist die Engine, die dafür sorgt, dass Teilschritte automatisiert stattfinden und das Zusammenspiel zwischen allen Systemkomponenten regelt. Zu diesem System wird eine Datenbank aufgesetzt, auf dem alle für das System benötigten Dateien gespeichert werden. Die Anpassung vom System sowie Erstellung und Bearbeitung verschiedener Workflows werden durch Programmierer mithilfe von IDEs (Integrated Development Environments) durchgeführt. Für die Bearbeitung von vielen in Java geschriebenen Systemen wird oft die Eclipse IDE empfohlen. Um die Erstellung von Workflows zu erleichtern, werden auch oft Plugins für die grafische Darstellung von Workflows mitgeliefert.

Die Daten und Informationen aus dem WfMS werden mithilfe von Webserver, in der Regel Tomcat, ins Internet übertragen. Somit können diese von einem Webbrowser aufgerufen werden. Was genau im Webbrowser mit den Daten gemacht werden kann, hängt von jedem einzelnen System ab. Meistens werden Monitoring und User Webinterfaces angeboten. Mithilfe des Monitoring Interfaces können z.B. alle Workflows beobachtet, gestoppt und gestartet werden. Von einem User Interface kann z.B. jeder einzelne Mitarbeiter ihm zugewiesenen Aufgaben anschauen und erledigen.

In jedem Unternehmen werden mehrere verschiedene Systeme verwendet, mit denen WfMS über diverse Schnittstellen interagieren und auf Daten zugreifen soll. Diese Daten, wie z.B.

E-Mail Adressen von Mitarbeitern, werden für die Durchführung von Workflows benötigt. Ein Teil dieser Schnittstellen, z.B. für solche Standardsysteme wie SAP, werden meistens vom Anfang an mitgeliefert. Andere Schnittstellen können aber in jedem einzelnen Unternehmen speziell entwickelt werden.

2.3 Open Source

2.3.1 Definition

Open Source ist generell eine Ausprägung von Software, die offenen Quellcode besitzt und die jeder nach Belieben studieren, benutzen, verändern und kopieren darf.¹⁰ Der Quellcode beinhaltet hierbei vom Menschen lesbare Befehle¹¹, aus dem die Funktion der Software hervorgeht.¹² Open Source Software lässt sich durch verschiedene Merkmale genauer charakterisieren, die sie von anderen Software Modellen abgrenzt. Mögliche Merkmale sind beispielsweise „offener Quellcode, freie Weiterverbuchung, abgeleitete Programme, Integritätsschutz des Originals, Diskriminierungsverbot, Verbot der Beschränkung auf bestimmte Anwendungsarten, Verbreitung der Lizenzvorschriften, Verbot der Beschränkung auf ein bestimmtes Produkt, Verbot der Ausstrahlung auf andere Software und das Neutralitätsgebot gegenüber anderen Technologien“¹³.

Eine mögliche Definition von Open Source liefert die sogenannte Open Source Initiative (OSI), die ein gemeinnütziges Unternehmen ist. Das Unternehmen wurde 1998 gegründet und hat seinen Sitz im US-Bundesstaat Kalifornien. Die Hauptaufgabe von OSI ist es, die Open Source Definition zu verwalten und Open Source Software Lizenzen auf Konformität zu überprüfen und anzuerkennen. Des Weiteren beteiligt sich das Unternehmen am Aufbau und der Ausbildung einer Open Source Gemeinschaft. Häufig wird Open Source Software von Unternehmen bei Kaufentscheidungen vernachlässigt. Aus diesem Grund betreibt OSI Öffentlichkeitsarbeit, um das Ansehen von nicht-proprietärer Software zu verbessern und das Bewusstsein für die Existenz von professionell anwendbarer Open Source Software zu stärken. Neben wirtschaftlichen Vorteilen kann Open Source Software auch strategische Vorteile im Vergleich zu proprietären Systemen für anwendende Benutzer und Unternehmen mit sich bringen.¹⁴

¹⁰ Vgl. Bundeszentrale für politische Bildung (o.J.)

¹¹ Vgl. Wichmann, T. (2005), S. 3

¹² Vgl. Bandel, G. (2010), S. 7

¹³ Oberhem, C. (2008), S. 11ff

¹⁴ Vgl. Open Source Initiative (2007)

Die sogenannten 10 Gebote von Open Source wurden 1998 von Todd Anderson, Chris Peterson, John maddog Hall, Larry Augustin, Sam Ockman, Eric Raymond, Linus Torvalds und Bruce Perens aufgestellt, die seit diesem Zeitpunkt als Kriterien für Open Source Software gelten:¹⁵

1. Freie Weiterverbreitung

Die mit der Open Source Software erworbene Lizenz darf keine der beteiligten Parteien im Verkauf oder in der Weitergabe der Software als Teil einer aus verschiedenen Quellen zusammengesetzten Softwaredistribution einschränken. Die Open Source Lizenz darf keine Lizenzgebühren oder andersartige Beträge einfordern.

2. Quellcode

Das Programm muss den Quellcode beinhalten. Sowohl die Verbreitung als Quellcode als auch in kompilierter Form sind zu erlauben. Für den Fall, dass ein Teil des Produkts ohne Quellcode veröffentlicht wird, muss ordnungsgemäß publiziert werden, wo der Quellcode für nicht mehr als nachvollziehbare Reproduktionskosten erlangt werden kann. Die bevorzugte Variante hierbei ist der kostenfreie Download über das Internet.

3. Auf dem Programm basierende Werke

Die Lizenz muss sowohl Veränderungen am Programm, auf dem Programm basierende Werke, als auch deren Verbreitung gemäß gleicher Lizenzbedingungen erlauben.

4. Unversehrtheit des Originalcodes

Die Lizenz darf eine Verbreitung von Quellcodes in abgeänderter Form nur in dem Fall einschränken, falls die Lizenz eine Verbreitung von Patchdateien in Verbindung mit dem Originalquellcode gestattet, damit das Programm vor der Benutzung verändert werden kann. Die Lizenz muss explizit die Verbreitung von Software erlauben, die mit verändertem Quellcode erstellt wurde. Die Lizenz erfordert es jedoch unter Umständen, dass modifizierte Arbeiten unter einem anderen Namen oder unterschiedlicher Versionsnummer im Vergleich zur Originalsoftware verbreitet werden.

5. Keine Diskriminierung von einzelnen Personen oder Gruppen

Die Lizenz darf keine einzelnen Personen oder Gruppen diskriminieren, um eine möglichst große Vielfalt an Mitwirkenden zu erlangen.

6. Keine Einschränkung für bestimmte Anwendungsbereiche

Die Lizenz darf keine Einschränkungen für die Verwendung des Programs in bestimmten Einsatzgebieten vorgeben. Beispielsweise darf keine Einschränkung der

¹⁵ Vgl. Müller, M. (1999), vgl. dazu auch: Open Source Initiative (o.J.)

Benutzung im kommerziellen Umfeld oder der Benutzung in der Genforschung gelten.

7. Verbreitung der Lizenz

Die mit dem Programm verbundenen Rechte müssen für alle gelten, die das Programm erhalten, ohne dass hierfür weitere Lizenzen berücksichtigt und erlangt werden müssen.

8. Lizenz darf nicht für ein bestimmtes Produkt gelten

Die zum Programm gehörigen Rechte dürfen nicht in Abhängigkeit dazu stehen, dass das Programm Teil einer bestimmten Softwaredistribution ist. Wird ein spezifische Programm aus der Distribution extrahiert und als solches benutzt oder verbreitet, so gelten für den Anwender und Benutzer die gleichen Rechte, die in der Originaldistribution gewährt wurden.

9. Open Source Lizenzen dürfen andere Software nicht beeinträchtigen

Die Lizenz darf keine andere Software einschränken, die in Kombination mit der lizenzierten Software verbreitet wird. Zum Beispiel darf die Lizenz nicht verlangen, dass alle anderen Programme, die auf dem selben Medium verbreitet werden, Open Source Software sein muss.

10. Technologieneutralität der Lizenz

Keine Klausel der Lizenz darf von einer bestimmten Technologie oder Form des Interfaces abhängig machen.

2.3.2 Historische Entwicklung

Anders als bei vielen geschichtlichen Ereignissen, lässt sich im Rahmen von Open Source Software kein genaues Gründungsdatum finden, sondern die Entwicklung stellt vielmehr einen jahrzehntelangen Prozess dar bis letztendlich der Begriff Open Source mit seiner gültigen Standarddefinition entstanden ist. Da es keine allgemeingültigen Aussagen gibt, sind auch verschiedene Ansätze zur Entstehung von Open Source Software verbreitet. Eine mögliche Entstehungsgeschichte bietet die Hackerkultur der 1960er Jahre. Diese Theorie wird im Rahmen dieser Arbeit jedoch nicht weiter erläutert und es wird auf die im folgenden aufgeführte historische Entwicklung begrenzt.¹⁶

Die Geschichte der Softwareentwicklung und somit auch die Entwicklung von Open Source Software hatte ihren Anfang im 20. Jahrhundert, wobei die Erfindung des Computers eine zentrale Rolle spielt. Zu Beginn war alle Software quelloffen und frei verfügbar, was auch für kommerzielle Computer galt. Software ist hierbei noch nicht als eigenständiges Produkt gesehen worden, sondern als Teil der Hardware. Jedoch war es die Aufgabe des Anwenders

¹⁶ Vgl. Keßler, S. (2013), S.15

dieses Produkt für sich selbst weiter zu schreiben und anzupassen. Somit waren die Nutzer vor allem programmierkompetente Spezialisten. Somit bauten Entwickler auf den wachsenden Bestand von vorhandener Software auf, verbesserten sie, entwickelten sie weiter und veröffentlichten ihre Ergebnisse der Nutzercommunity. Einnahmequelle für Unternehmen war hierbei nicht das Programm, sondern das Programmieren selbst. In den 1960er Jahren dominierten große Hardware-Firmen wie IBM, Hewlett-Packard und Data Central die Computerindustrie. Beim Marktführer IBM beispielsweise konnten Kunden nur ein Computerpaket erwerben, welches neben der Hardware auch Software, Peripheriegeräte, Wartung und Schulung beinhaltete. Somit gab es noch keinen eigenständigen Softwaremarkt.

1969 löste IBM als erster Anbieter die Bündelung auf, was auch als Startschuss zur Bildung einer Softwareindustrie gesehen werden kann.¹⁷ Im gleichen Jahr entwickelten Ken Thompson und Dennis Ritchie in den AT&T Bell Telephone Laboratories die erste Version von UNIX. Da AT&T jedoch einen Kartellrechtsprozess, der sich von 1949 bis 1956 zog, verloren hatte und seine Tätigkeit somit auf den Bereich Telekommunikation beschränken musste,¹⁸ war eine kommerzielle Verwertung von UNIX nicht möglich. Lizenzen wurden jedoch vor allem an Universitäten abgegeben, wo eine große Entwicklergemeinschaft entstand. Koordinationsschnittstelle war hierbei die University of Berkeley, welche unter dem Namen Berkeley Software Distribution (BSD) UNIX weiterentwickelte und an andere Universitäten vertrieb.

Erst im Jahr 1984 änderte sich die Rechtsprechung im amerikanischen Telekommunikationsmarkt, wodurch AT&T nun im Softwaregeschäft tätig sein durfte. Die hierbei geführte Taktik von AT&T war es, die UNIX-Lizenzen durch höhere Preise und Geheimhalten des Quellcodes unter Verschluss zu bringen. Hierdurch konnte die nun bereits sehr große Entwicklergemeinschaft nicht mehr in freien Softwareprojekten arbeiten.

Als Reaktion gründete Richard Stallman das sogenannte GNU-Projekt um ein UNIX ähnliches Betriebssystem zu entwickeln, welches nur aus freien Softwarequellen bestehen sollte.¹⁹ Um einen Gegenpol gegen die Kommerzialisierung von Software zu bilden, gründete er 1983 die Free Software Foundation unter dem Leitsatz GNU ist not Unix.²⁰ Die Akzeptanz freier Software auf dem Softwaremarkt war jedoch nur im privaten und universitären Bereich vorhanden. Die ersten Internet-Service-Provider im Jahr 1990 ermöglichten Privatpersonen den Zugang zum Internet, wodurch Entwickler sich auch außerhalb von Universitäten an Softwareprojekten beteiligen konnten.

¹⁷ Vgl. Grassmuck, V. (2004)

¹⁸ Vgl. Müller, M. (1999)

¹⁹ Vgl. Lämmer, A., u.a. (2001)

²⁰ Vgl. Computerwoche (1998)

1987 entwickelte Andrew Tanenbaum das Betriebssystem Minix für Intels Mikroprozessoren. 1991 entwickelt Linus Torvald das Betriebssystem Minix weiter, welches die Basisideen für Linux liefert. Idee dieses Projekts von Torvald war es nun weitere Codebeiträge aus der Entwicklergemeinschaft zu erhalten, die er im Gegenzug verteilen würde. Auf ähnliche Art und Weise entstehen weitere freie Programme wie Apache und Python. In den folgenden Jahren entwickelte sich starkes Diskussionspotential und eine Art Kleinkrieg zwischen Unternehmen im IT Umfeld.

1998 wurde daraufhin ein Lizenzierungsverfahren für die freiverfügbare Software mit offenem Quellcode, wie dies beispielsweise bei Linux der Fall war, durch Eric Raymond erarbeitet. In diesem Zuge prägte er auch den Begriff Open Source.²¹ Eine offizielle Definition, wie sie in Kapitel 2.3.1 vorgestellt wurde, lieferten Todd Anderson, Chris Peterson, John maddog Hall, Larry Augustin, Sam Ockman, Eric Raymond, Linus Torvalds und Bruce Perens im Rahmen der 1998 gegründeten Open Source Initiative.

2.3.3 Open-Source Lizenz Modelle

Bei dem Einsatz von Open Source Software muss zwischen einerseits der reinen Verwendung des Software Produktes, wie beispielsweise dem Einsatz und Benutzung von OpenOffice.org, und der Verwendung des Quellcodes eines Software Produktes als Basis oder als Teil einer eigenen Software unterschieden werden. Hierbei ist zu beachten, dass die reine Verwendung von Open Source Software keinen Auflagen unterliegt, sowohl im privaten als auch kommerziellen Umfeld, unterliegt, wohingegen für die Verwendung des Sourcecodes oder die Verbreitung Regeln eingehalten werden müssen, welche in den Lizenzbedingungen der jeweiligen Open Source Software bestimmt werden.²²

Durch Unterschiede in der Ausgestaltung dieser Bedingungen und den daraus resultierenden Verpflichtungen, lassen sich drei Hauptkategorien von Open Source Lizenzen definieren. Die hierfür verwendete Begrifflichkeit des Copyleft ist in Anlehnung an das Copyright entstanden, welches die Verbreitung von Daten nur dem Urheber gestattet. „Copyleft besagt, dass sämtliche Änderungen und Weiterentwicklungen einer Open-Source-Software nur unter der gleichen Lizenz als freie Software weitergegeben werden dürfen.“ Diese Definition dient als Grundlage zur Kategorisierung von Open Source Lizenzen. Lizenzen, die keine Abweichung von diesem Prinzip zulassen, besitzen ein starkes Copyleft, wohingegen es auch we-

²¹ Vgl. Computerwoche (1998)

²² Vgl. Schmidt, A. (2012)

niger restriktive (schwaches Copyleft) und Lizenzen ohne Copyleft gibt.²³ Eine Übersicht befindet sich in Tabelle 1.

Art des Copyleft	Starkes Copyleft	Schwaches Copyleft	Kein Copyleft
Kombinationsmöglichkeit mit proprietärer Software	Keine Einbindung in proprietären Code möglich	Statisches und dynamisches Linken von Code mit proprietärer Software möglich. Eigenentwicklungen dürfen als proprietäre Software weitergegeben werden	Keine Vorgaben. Der gesamte Code darf auch als proprietäre Software weitergegeben werden.
Beispiel-Lizenz	GNU General Public Licence	GNU Lesser General Public Licence (LGPL), Mozilla Public Licence, Eclipse Public Licence	BSD, Apache

Tabelle 1: Open Source Lizenz Kategorien²⁴

Aufgrund der Restriktionen, die durch die verschiedenen Vorgaben der Open Source Lizenzen entstehen können, ist es essentiell für ein Unternehmen bei der Entscheidung, ob und wenn ja welche Open Source Software eingesetzt werden soll, neben den Funktionalitäten der jeweiligen Software auch die Lizenzbedingungen zu berücksichtigen, um rechtlichen Schritten zu entgehen und den reibungslosen Einsatz der Software innerhalb des Unternehmens zu garantieren.

3 Exemplarische Vorstellung verschiedener Open Source Workflow-Management Systeme

Die meisten Open Source WfMS werden in der Programmiersprache Java geschrieben. Java-source.net hat mehrere Workflow Engines aufgelistet und kurz beschrieben. Einige davon

²³ Vgl. Kleijn, A. (2006)

²⁴ Vgl. Kleijn, A. (2006), vgl. dazu auch: Wahlers, J. (2013)

werden auch im Folgenden näher erläutert. Zu der Liste auf der Webseite zählen (mit Hauptdifferenzierungsmerkmal):²⁵

- Sarasvati (kann Subprozesse aufsetzen und Aktivitäten rückwärts nachvollziehen)
- jBPM (unten beschrieben)
- Enhydra Shark (unten beschrieben)
- OpenSymphony OSWorkflow (stark anpassbar, durch eigene Programmierung)
- Con:cern (benutzt Pre- und Postkonditionen, um Aktivitäten zu initiieren bzw. zu beenden)
- Werkflow (deckt Vielzahl von großen Unternehmensprozessen bis hin zu kleinen Mensch-Maschine Interaktionen ab)
- ObjectWeb Bonita (ähnlich zu anderen Bonita Produkten)
- Bossa (benutzt Petrinetz-Darstellung)
- Open Business Engine (benutzt XPD L und ist innerhalb von J2EE ausführbar)
- The Open for Business Workflow Engine (basiert auf WfMC-Standards und legt den Fokus auf Entitäten)
- OpenWFE (besteht aus Engine, Arbeitsliste, Webklienten und einer reagierender Komponente, benutzt Python und kann mit REST integriert werden)
- WfMOpen (wird von der WfMC und Object Management Group vorgeschlagen)
- XFlow (Basis für zusammen arbeitende Anwendungen und integrierbare Prozesse)
- JFolder (kann so konfiguriert werden, dass es mit J2EE Applikationsservern und einer Reihe von Datenbanken läuft)
- Taverna (unten beschrieben)
- Freefluo (kann Webservices aufrufen und ist mit Taverna und IBMs WSFL integrierbar)
- Micro-Flow (für Entwickler, die die Kontrolle und die Logik trennen wollen)
- JFlow (kann komplexe Flüsse darstellen und evaluieren)
- YAWL (bietet direkte Unterstützung für alle Arbeitsschritt Muster, basiert auf grundlegender Analyse bestehender WfMS)
- Kbee.workflow (in Eclipse integrierbar, Entwicklerfokussiert)
- Syrup (bietet volle Persistenz, sodass nie ein Schritt verloren geht)
- PXE (eine Laufzeitkomponente für bestimmte Prozesse)
- ActiveBPEL (führt in Business Process Execution Language (BPEL) definierte Aktivitäten für Webservices aus)
- Zebra (Nischenprodukt, kann alle Arbeitsschritte modellieren, die verschiedenen Mustern zugeordnet sind)
- Apache Agila (läuft auch auf Handhelds und ist gut integrierbar)

²⁵ Vgl. Java-source.net (o.J.a)

- Antflow (angeblich performanter, ordnet automatisierte Aktivitäten lokale Verzeichnissen zu)
- Beexee (verwaltet bestehende Webservices)
- Dalma (kann den gerade ausgeführten Status eines Threads erfassen, abspeichern und später wieder aufnehmen)
- Swish (Zusatzengine zu Enhydra Shark, auch für Klienten, die kein Java ausführen)
- BpmScript (kann Prozesse clustern, parallelisieren, halb asynchronisieren und Ähnliches)
- Jawflow (benutzt fast alle WfMC-Standards)
- RUNA WFE (basiert auf JBoss mit graphischer Weboberfläche, bietet eine Vielzahl an Integrationen mit Verzeichnissen, sowie eine Zuordnung zu diversen Ländern)
- Imixs IX Workflow (unten beschrieben)
- Pegasus (ordnet selbstständig den Ressourcen verschiedene Aktivitäten zu)
- Wilos (benutzt Prozesse im XML-Format und den Eclipse Process Framework Composer)
- Apache ODE (besonders auf Webservices fokussiert)
- Tobflow (Objektmodell basierend, Fokus auf Dokumenten und Arbeitsablaufschritte)
- Bonita Open Solution (unten beschrieben)
- Joget Workflow (hauptsächlich Menschen-orientiert, gut integrierbar)
- Activiti (unten beschrieben)

3.1 Activiti

Activiti ist Open ein Source Workflow-Management-System, das der Apache Lizenz Version 2.0 untersteht. Dazu kann Enterprise Support gebucht werden, der von der selbstständigen Firma Alfresco angeboten wird. Der Kern des Activiti Systems ist eine schnelle und solide BPMN 2.0 Prozess Engine für Java. Dieses Projekt wird auch von Alfresco finanziert und wurde von zwei ehemaligen Programmierern von JBoss gegründet. Dabei wurde Activiti auf einer neuen und nicht auf JBoss jBPM Codebasis gesetzt. Die Hauptunterschiede zwischen den beiden Systemen liegen aber nicht an der Engine, sondern an den mitgelieferten Tools.²⁶

Die Abbildung 2 veranschaulicht alle Komponenten, die zu dem ganzen Activiti System kombiniert werden.

²⁶ Vgl. Activiti (2014a)

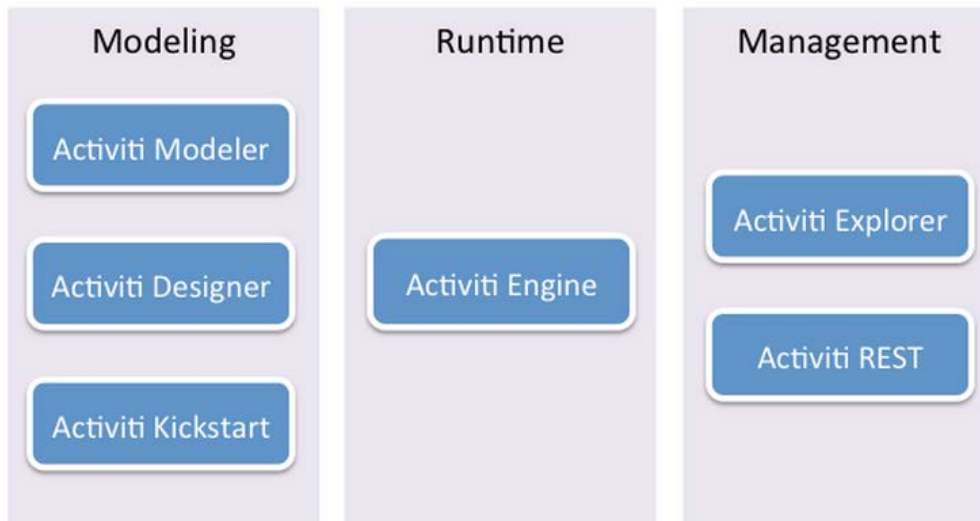


Abbildung 2: Activiti Components²⁷

Der wichtigste Teil und der Kern von Activiti ist die Activiti Engine. Es ist ein Java Prozess Engine, die BPMN 2.0 Prozesse nativ ausführt. Diese Engine verfügt über mehrere wichtige Eigenschaften:²⁸

Das System erlaubt die Benutzer-Updates mit Prozess-Updates innerhalb einer einzigen Transaktion zusammenzufassen. Es läuft unter jeder Java-Umgebung, wie z.B. Spring oder JTA. Das System ist einfach mit Setup Utility zum Laufen zu bringen. Außerdem kann es einfach neue benutzerdefinierte Aktivitätstypen hinzufügen und die Prozesssprachen vervollständigen. Die Engine ist allgemein dafür bekannt sehr zuverlässig und schnell zu sein. Diese Engine bietet die Möglichkeit an, isoliert verschiedene Prozess-Unit Tests durchzuführen.

Die nächste Komponente von Activiti ist der Activiti Explorer. Es ist eine Webanwendung, die den Zugriff zu Activiti Engine Laufzeitumgebung für alle Benutzer des Systems anbietet. Sie beinhaltet Aufgabenmanagement, Inspektion von Prozessinstanzen, Management Funktionen und Anzeige von Berichten, die auf der Grundlage von statistischen und historischen Daten basiert sind.²⁹

Zwei andere wichtigen Komponenten sind Activiti Modeler und Activiti Designer. Activiti Modeler kann nur in Webumgebung verwendet werden. Es gibt die Möglichkeit die Activiti Prozesse zu verwalten und diese mithilfe von BPMN 2.0 grafisch darzustellen. Activiti Designer hingegen ist ein Eclipse Plugin, der es ermöglicht, verschiedene Prozesse auch mit BPMN 2.0 in der IDE-Umgebung zu modellieren.³⁰

²⁷ Enthalten in: Activiti (2014b)

²⁸ Vgl. Activiti (2014b)

²⁹ Vgl. Activiti (2014b)

³⁰ Vgl. Activiti (2014b)

3.2 Bonita BPM 6

Bonitasoft bietet grundsätzlich eine Suite aus drei Produkten an. Auf dem Markt gibt es die Produkte jeweils als Open Source (Freeware) oder als (bezahltes) Abonnement Paket. Grundlegend ist Bonita BPM laut ohloh.net ein sehr aktives Projekt mit fast 60 Entwicklern und regelmäßigen Commits. Es benutzt die GPL2 Lizenz.³¹ Die Funktionen unterteilen sich in verschiedene Bereiche. Für die Open Source-Variante bietet Bonitasoft im Modellierungs- und Entwicklungsbereich Prozessmodellierung nach BPMN2 Standard, sowie Prozessversionierungen und – simulationen. Auch Module wie z.B. BPMN2, XPD, oder andere Framework BMP-Tools dritter Anbieter können importiert werden. Generall ist das Produkt individualisierbar, was die Benutzeroberfläche angeht. Es kann mit den Geschäftsregeln verbunden werden und bietet Entscheidungstabellen zur Hilfe an. Es gibt einen Assistenten um Konnektoren zu externen Systemen zu erstellen, ein paar sind auch schon intergriert, jedoch für weit verbreitete Systeme wie Salesforce.com oder SAP werden die vorgefertigten Konnektoren nur in der bezahlten Version mitgeliefert. Allerdings verfügt selbst die Open Source Version über einen Debugger, eine Benutzerführung, einen Datenmanagement-Editor, sowie einen Webformular-Editor, womit die Grundfunktionen eines Workflow-Management Systems grundsätzlich abgedeckt sind.³² Laut anderen Quellen kann man aber die kostenlose Open Source Version sogar mit Datenbanken wie Oracle, MySQL (My Structured Query Language), Verzeichnissen wie LDAP, aber auch Twitter oder Facebook verbinden.³³

In der Kategorie Ausführung und Deployment sind die Funktionen noch sperrlicher gehalten. Hier sind lediglich Transaktions-Engines, eine asynchrone Ausführung sowie Java- und REST-APIs vorgesehen. Bereits für ein Remote Deployment, andere Ausführungsumgebungen, sowie mandantenfähige Infrastruktur und –basierte Konfiguration wird der Kunde zur Kasse gebeten.

Bei der User Experience und dem Monitoring wurde schließlich am meisten eingespart. Hier stehen lediglich die Erstellung von Geschäftsanwendungen und die Benutzerverwaltung zur Verfügung. Für alles andere, wie zum Beispiel eine Dokumentenverwaltung, KPIs (Key Performance Indicators), Dashboards, die Suchfunktion, jegliche Art des Monitorings usw. muss man bereits bezahlen.³⁴

³¹ Vgl. Ohloh.net Black Duck (2014a)

³² Vgl. Bonitasoft (o.J.a)

³³ Vgl. Java-source.net (o.J.b)

³⁴ Vgl. Bonitasoft (o.J.a)

Allerdings sind alle (auch die bezahlten Produkte) Open Source und Bonitasoft bietet diversen Support an. Es gibt Webinars, Schulungen, Telefonsupport mit verschiedenen Erreichbarkeitszeiten und Remote-Support.³⁵

3.3 Camunda BPM platform

Camunda services GmbH ist ein Unternehmen aus Berlin, das als ein BPM-Beratungshaus gegründet wurde, aber heutzutage primär ein BPM-Softwarehersteller ist. Camunda BPM ist ein Open Source Workflow-Management-System von camunda, das in Java geschrieben ist. Diese Open Source Software untersteht der Apache Lizenz Version 2.0. Mithilfe von diesem System können verschiedene Businessprozesse in BPMN 2.0 definiert und ausgeführt werden. Camunda BPM wurde zuerst auf Activiti BPM aufgebaut. Seit dem Jahr 2012 wurde die Entwicklung von camunda BPM von Activiti abgespalten und getrennt als eigenständiges Produkt weiterentwickelt.³⁶

Die camunda BPM bietet mehrere verschiedene Features an. Eine von diesen Features ist Process Engine. Wie es bereits erwähnt wurde, ist der Kern von camunda BPM eine in Java implementierte und direkt einbettbare native BPMN 2.0 Process Engine. Diese Engine wurde auch speziell für High Concurrency-Szenarien, also für die gleichzeitige Ausführung von mehreren Befehlen und Berechnungen, optimiert. Weiter bietet camunda einen BPMN 2.0 Modeler an. Dieser Modeler ist ein Eclipse Plugin für die BPMN-Prozessmodellierung. Mithilfe von dem Modeler können Java-Entwickler bequem das Prozessmodell bearbeiten und dabei die typischen Eclipse-Hilfsmittel verwenden.³⁷

Mit camunda BPM können Entwickler sehr flexibel und frei in der Wahl von Framework sein. Neben der Integration mit Java EE oder CDI (Context and Dependency Injections) wird auch eine Spring-Integration angeboten. Für die Erstellung von Benutzeroberfläche können JSF (JavaServer Faces) oder andere Java GUI (Graphical User Interface) -Frameworks verwendet werden. Wenn die Entwickler eine andere Client-Technologie nutzen, kann die camunda BPM über REST angesprochen werden. Um die Arbeit noch mehr zu erleichtern, wird neben der camunda BPM auch ein leistungsfähiges Tool für den Prozessbetrieb namens camunda Cockpit bereitgestellt. Mithilfe von diesem Tool können verschiedene Monitoringaufgaben durchgeführt werden, wie z.B. Überwachen, Abbrechen, Wiederanstoßen und Manipulieren von Prozessinstanzen. Ein weiteres Tool, camunda Tasklist, hilft bei der Abarbeitung von Aufgaben durch Fachanwender. Dabei können verschiedene Aufgaben direkt den Benutzern

³⁵ Vgl. Bonitasoft (o.J.b)

³⁶ Vgl. camunda services GmbH (2013a)

³⁷ Vgl. camunda services GmbH (2013a)

zugeordnet werden. Das Tool unterstützt auch die rollenbasierte Aufgabenlisten sowie Claiming, Unclaiming und Delegieren von Aufgaben.³⁸

Von camunda GmbH wird auch eine Enterprise Edition mit bis zu 24/7 Support angeboten. Alle Unterschiede zwischen Open Source und Enterprise Edition werden in der Tabelle 2 definiert.

	Open Source	Enterprise
BPMN 2.0 Process Engine	✓	✓
+ Distributionen für Tomcat, JBoss, Glassfish	✓	✓
+ Distributionen für IBM Websphere AS, Oracle WLS		✓
Modeler	✓	✓
Tasklist	✓	✓
Cycle	✓	✓
Cockpit	✓	✓
Support (bis zu 24/7)		✓
Wartung (Patch Releases)		✓
Haftungsfreistellung		✓

Tabelle 2: Vergleich Open Source vs. Enterprise Edition³⁹

3.4 Enhydra Shark

Enhydra Shark, auch bekannt als „Together XPD and BPMN Workflow Server“, ist ein erweiterbares Open Source Workflow (Arbeitsablauf) Engine Framework (Programmiergerüst). Eine Engine ist ein Teil eines Programms, das meist selbstständig im Hintergrund Berechnungen oder Simulationen ausführt. Es orientiert sich dabei sehr eng an den WfMC- Standards und kann alleine oder integriert als Java Workflow Engine betrieben werden.⁴⁰ Des Weiteren ist es bekannt dafür, sehr flexibel zu sein. Die benutzen Prozessdefinitionen basieren auf den Richtlinien der WfMC-XPDL (XML Process Definition Language) ohne dabei eigene proprietäre Erweiterungen anzubieten. XPDLs können auch selbst mit Hilfe von anderen XPDL Anwendungen, sowie dem BPMN (Business Process Model and Notation) Work-

³⁸ Vgl. camunda services GmbH (2013a)

³⁹ Enthalten in: camunda services GmbH (2013b)

⁴⁰ Vgl. Java-source-net.de (o.J.)

flow Editor erstellt werden. Der Server funktioniert unter anderem auch mit den WfMC Tool Agents API (Application Programming Interface), sprich den WfMC Methodenschnittstellen. Auch andere Toolagents sind für die einfachen Anwendungen bereits mit dabei. Um das Programm zu bedienen gibt es einen Kommandozeilenklienten, einen Swing Administrationsklienten und einen webbasierenden Arbeitslisten- und Administrationsklienten. Für die Prozesse können MS Office-Dokumente, PDFs, Outlook E-Mails und ähnliches direkt im Programm gespeichert und über den Browser per HTML Schnittstelle unter Berücksichtigung des WebDAVs (ein Standard für Dateien im Internet) bearbeitet werden. Sogar Outlooklisten können bearbeitet werden. Hierzu wird ein MS SharePoint simuliert. Für besonders komplizierte Szenarien ist der webbasierende Klient der Xforms und XSLT (beides Dateiformate) vorgesehen. Um sämtliche Prozesse und Aktivitäten zu speichern, zieht Enhydra Shark die Firmeneigene relationale Objektdatenbank heran. Transaktionen können per Standards aber auch in JDBC (Java Database Connectivity) und JTA (Java Transaction API) ausgeführt werden.

Generell kann das Produkt als Java Bibliothek, Webservice, Corba Server, Java Plattformen und anders genutzt werden. Man erwirbt es durch die GPL (General Public License) V3 Lizenz und teilweise bereits als Bestandteil bestehender Open Source Systeme. Es wirbt damit ein schlichtes System ohne unnötige Funktionen zu sein und sich sehr gut mit anderen „Together“- sowie Java-Produkten zu integrieren.⁴¹ Es ist schwierig die Aktivität von Enhydra Shark festzustellen, was unter anderem an der Namensänderung liegt. Das eigentliche Enhydra Shark ist laut der Vergleichsseite ohloh.net seit 2009 inaktiv.⁴²

3.5 Imixs Managed Workflow Server

Dieser Server läuft auf einer virtuellen Umgebung. Das hat den Vorteil, dass etwaige Anschaffungskosten für neue Server entfallen. Die Lösung ist so konzipiert, dass Imixs die Serververwaltung, Softwareupdates und Backups übernimmt. Durch die Virtualisierung wird weiterhin ermöglicht, das System auch in der Cloud aus zu führen. Das System soll maximale Flexibilität durch die Vereinbarkeit mit z. Bsp. MySQL, OpenJDK (Javaplattform), Jenkins (Integrationssystem), Git (Versionsverwaltung), Apache (Webserver) oder PHP (Hypertext Preprocessor; Skriptsprache) liefern. Die Anwendung lässt sich auch spiegeln, um eine höhere Verfügbarkeit zu erreichen. Imixs bietet außerdem telefonischen Support, Beratung und eine online Demo.⁴³ Darüber hinaus bietet die Firma an, verschiedene Dienste zu übernehmen,

⁴¹ Vgl. Together Teamsolutions Co, Ltd. (2011)

⁴² Vgl. Ohloh.net Black Duck (2014b)

⁴³ Vgl. Imixs Software Solution GmbH (o.J.a)

das geht über Beratung bis hin zu Anpassungen, Schnittstellenprogrammierung und den Aufbau von diversen Testumgebungen. Diese Dienste sind kostenpflichtig. Die Preise wurden gestaffelt und fangen bei 119€ im Monat für Support, Beratung, Wartung und regelmäßige Updates an. Für Sonderanfertigungen werden mind. 238€ im Monat verlangt, und um das Programm komplett anzupassen und weiterzuentwickeln muss der Preis individuell verhandelt werden.⁴⁴ Es fällt auf, dass Imixs wenig technische Informationen herausgibt, aber ansonsten auf ihrer Seite gut erklärt, was ein Workflow Management System überhaupt ist, wozu es dient und sogar Kriterien auflistet, welchen Anforderungen genüge getan werden muss. All das geschieht sehr objektiv, ohne gezielt ihr Produkt zu verbreiten. Ohloh.net zu Rate ziehend gibt es außerdem zu bedenken, dass dieses Projekt derzeit nur von zwei Entwicklern unterstützt wird. Es benutzt die GPL-Lizenz.⁴⁵

3.6 JBPM

JBPM ist ein jBoss-Projekt und eigentlich eine Weiterentwicklung dessen. Das Projekt hat derzeit 80 Entwickler und weist eine hohe Aktivität auf.⁴⁶ Auch dieses Produkt bietet eine java-basierte, (leichte,) erweiterbare Workflow Engine. Sie benutzt den BPMN 2 Standard und bietet eclipse und webbasiertes Editionsfunktionen. Die verschiedenen Prozesse und Aufgaben, die in einem Repository abgespeichert werden können, sind über graphische drag und drop-Funktionen sehr leicht einzurichten und zu verändern. Dabei kann der Grad des technischen Wissens, das benötigt wird, um dieses Programm zu bedienen, über verschiedene Methoden und Features eingestellt werden, sodass sowohl Wirtschaftler wie Itler damit umgehen können. Transaktionen und Datenbankschnittstellen werden über JTA und JPA (Java Persistence API) eingerichtet. Des Weiteren sind auch Schnittstellen für Aufgaben vorgesehen, die menschliche Interaktion benötigen, sowie die Integrierbarkeit von Geschäftsregeln und Prozessevents. Von einem mit jBPM abgebildeten Geschäftsprozess können auch nur Teilprozesse individuell ausgeführt werden, was mehr Flexibilität erlaubt. Alle Prozesse können nicht nur statisch, sondern auch adaptiv und dynamisch sein. Eine produkt-eigenen Managementkonsole sorgt für die Verwaltung von Prozessinstanzen, Aufgabenlisten, (individuellen aufgesetzten) Berichten, Management Aufgaben und Ähnlichem. jBPM bietet überdies durch Webtools die Möglichkeit Prozesse zu simulieren und ihre benötigten Daten und Form zu modellieren. Außerdem wird eine Historie aufgezeichnet, die es ermöglicht Analysen zu fahren, das System zu überwachen oder eigene Fragen und Programm-aufgaben zu erzeugen. Darüber hinaus ist jBPM mit anderen Produkten integrierbar, so z.

⁴⁴ Vgl. Imixs Software Solution GmbH (o.J.b)

⁴⁵ Vgl. Ohloh.net Black Duck (o.J.d)

⁴⁶ Vgl. Ohloh.net Black Duck (2014c)

B. mit Seam, Spring, OSGi (Open Services Gateway initiative) oder anderen Plattformen, und auch Git zur Verwaltung der (zusätzlich selbstgeschriebenen) Quellcodes, bzw. Maven zur Erleichterung des eigenständigen Programmieren durch beispielsweise Projekt Objekt Modelle. Der ausführende Server kann ferngesteuert über REST oder JMS angesprochen werden und in Clustern die Prozesse installieren, um die Arbeitslast auszugleichen und eine hohe Verfügbarkeit zu gewährleisten.⁴⁷

3.7 ProcessMaker Open Source

Der ProcessMaker Open Source ist eine von vier Varianten des ProcessMakers, neben einer Cloud Edition einer On-Premise Enterprise Edition und SugarCRM Edition. Die Open Source Variante enthält die wesentlichen BPM und Workflow Funktionalitäten der erweiterten und kostenpflichtigen anderen Optionen. Jedoch stehen hier keine Software Fixes, Workflow Vorlagen, Plug-ins oder Support zur Verfügung. Für die Open Source Variante wird eine GPLv3 Lizenz benutzt.⁴⁸ Im Gegensatz zu anderer Open Source Software, ist bei diesem System 95% des Codes vom direkten Entwicklerteam von ProcessMaker geschrieben worden, welches von einer Community von Entwicklern und Nutzern unterstützt wird.⁴⁹ Aus diesem Grund ist die Aktivität an dieser Software sehr hoch. Die verwendete Programmiersprache ist PHP.⁵⁰ Das System ist webbasiert und erlaubt es Workflows per drag-and-drop System zu erstellen. Nach Angabe des Herstellers ist diese Variante des ProcessMakers vor allem für kleinere Workflow Entwicklungen und zur Unterstützung von Prozessbewertungen geeignet und weniger für große Unternehmen. Die Prozessmodellierung erfüllt die Standards von BPMN 2.0. Trigger, die innerhalb von Prozessen geschaltet werden, sind in der Programmiersprache PHP zu verfassen. Zusätzlich zu diesen entwickelten Triggern können auch Third Party Trigger, wie Microsoft Sharepoint DWS Trigger, SugarCRM, Alfresco und Zimbra, eingebunden werden. Da das System webbasiert ist, kann individueller JavaScript Code eingebaut werden. Dies wird vor allem zur Verbesserung der Anwendungsfreundlichkeit und Steigerung der Funktionalitäten der Prozessschritte angewendet.⁵¹

Wie bereits erwähnt, stellt die Open Source Edition eine in ihrer Funktionalität und Komplexität eingeschränkte Version des ProcessMakers dar, sodass zur Verwendung in Unternehmen ein Entwicklerteam dringend notwendig ist, um mithilfe der vorhandenen Ressourcen, die

⁴⁷ Vgl. jBPM (o.J.)

⁴⁸ Vgl. Colosa, Inc. (o.J.)

⁴⁹ Vgl. Colosa, Inc. (2014a)

⁵⁰ Vgl. Ohloh.net Black Duck (o.J.)

⁵¹ Vgl. Colosa, Inc. (2014b)

durch den Download des ProcessMakers Open Source erlangt werden, komplexe Workflows zu generieren, wie sie in Unternehmen auftreten.

3.8 Taverna

Taverna ist ein weiteres Open Source Workflow-Management-Suite, das von myGrid team in Java entwickelt wurde und der LGPL Lizenz untersteht. MyGrid team ist eine Forschungsgruppe aus Großbritannien, die sich mit der Entwicklung von Tools im eScience Bereich beschäftigt. Taverna Suite wird in der Regel für wissenschaftliche Workflows z.B. in Medizin, Bioinformatik und Chemieinformatik verwendet. Für die Modellierung von Workflows wird ein freier Webservice namens High Performance Computing Online genutzt.⁵²

Taverna Suite besteht aus verschiedenen Tools. Eins der wichtigsten Tools ist Taverna Workbench, das ein Desktop Applikation ist und für Design und Ausführung von Workflows verwendet wird. Dazu kommt ein Command Line Tool, für eine schnelle Ausführung von Workflows vom Terminal. Für die Remote Ausführung von Workflows wird ein Taverna Server angeboten. Zusätzlich wird auch ein Taverna Online mitgeliefert, das die Erstellung von Workflows in einem Webbrowser ermöglicht. Taverna Suite ist leicht erweiterbar und mit vielen anderen Tools und Services, wie z.B. REST kompatibel.⁵³

3.9 Workflow Server

Der Workflow Server von MyControl ist generell eine Verbindung von einem klassischen Workflow Management System mit einem intelligenten und prozessbezogenen Wissensmanagement. Die Aufgabe des Workflow Servers ist es den reibungslosen Ablauf der Geschäftsprozesse zu koordinieren, steuern und überwachen. Die Software generell kann als Freeware direkt auf der Website des Anbieters heruntergeladen werden.⁵⁴ Zusätzlich gibt es die Möglichkeit das Hosting des Workflow Servers von MyControl durchführen zu lassen und somit ein Software as a Service Modell (SaaS) für das Unternehmen zu etablieren. Diese Variante ist kostenpflichtig und enthält neben der Bereitstellung der Software per Internet, das Hosting auf den Hochverfügbarkeits-Clustern von MyControl, automatische Updates auf neuere Versionen, als auch Support und eine Hotline. Der Preis für diese Dienstleistung liegt bei mindestens 79€ monatlich. Des Weiteren werden Beratung, Schulungen, Workflow Ent-

⁵² Vgl. Taverna (2014a)

⁵³ Vgl. Taverna (2014b)

⁵⁴ Vgl. MyControl (o.J.a)

wicklung, individuelle Anpassungen und Schnittstellenprogrammierung als Dienstleistung angeboten.⁵⁵

Die Software erfordert auf Client Seite, dass Mozilla Firefox oder Microsoft Internet Explorer vorhanden sein muss, da das System webbasiert ist.⁵⁶ Die Daten werden nach Industriestandard SSLverschlüsselt für die Kommunikation.

Die Funktionalitäten lassen sich in vier Kategorien untergliedern: das Modellieren der Geschäftsprozesse selbst, das Arbeiten mit dem Workflow Server, das Management mit dem Workflow Server und Wissensmanagement.

Workflows können entweder mithilfe des Workflow Assistent, bei welchem der Nutzer die notwendigen Angaben in vorgefertigte Felder einträgt und automatisch ein Workflow hieraus generiert wird, oder mithilfe des Workflow Editors. Hiermit ist es möglich komplexere Geschäftsmodelle darzustellen, externe Anwendungen in den Prozess zu integrieren oder KPIs zu vergeben. Als integrierte Skriptsprache wird wfScript verwendet. Es besteht die Möglichkeit benutzerdefinierte Felder und Formulare zu erstellen, bedingte Workflows zu erstellen, automatische Transits zu generieren und Rechte zu vergeben und zu verwalten. Zusätzlich zur eigentlichen Software ist es möglich andere Medien und Softwaresysteme zu integrieren. Hierfür wird die auf offenen Standards basierende Anwendungsschnittstelle (SOAP-API) verwendet.⁵⁷

Trotz der genauen Erklärung der verschiedenen Funktionalitäten auf der Website werden kaum technische Informationen erwähnt.

4 Zusammenfassung kommerzieller Workflow-Management-Systeme

Im Gegensatz zu Open Source Freeware für Workflowmanagement Systeme, stehen kommerzielle Systeme. Kommerzielle Software wird meist von Entwicklern innerhalb eines Unternehmens entwickelt und nicht von freien Entwicklern, wie es bei Open Source der Fall ist. Ein Ziel kommerzieller Software ist es einen Profit für das jeweilige Unternehmen zu erwirtschaften. Somit ist nahezu jede kommerzielle Software proprietär. Meist sind die Systeme nicht frei verfügbar, nicht kopierbar und modifizierbar.⁵⁸

⁵⁵ Vgl. Ebenda (o.J.b)

⁵⁶ Vgl. MyControl (o.J.c.)

⁵⁷ Vgl. MyControl (o.J.d)

⁵⁸ Vgl. Buskamp, D. (o.J.)

Als Entscheidungshilfe für den Kauf von kommerzieller Software für ein Workflowmanagement System können Analystenreports wie Gartner Magic Quadrant for Intelligent Business Process Management Suites oder Ovum Decision Matrix for Selecting a Business Process Management Solution herangezogen werden. Beim Vergleich beider Reports lässt sich klar feststellen, dass es drei Marktführer in diesem Bereich gibt, Pega v7 von Pegasystems, Appian v7.2 von Appian und IBM Business Process Manager von IBM.⁵⁹ Die graphischen Darstellungen beider Auswertungen sind in Abbildung 3 und 4 dargestellt.



Abbildung 3: Gartner Magic Quadrant for Intelligent Business Process Management Suites 2014⁶⁰

⁵⁹ Vgl. Jones, T./ Schulte, R./ Cantara, M. (2014); vgl. dazu auch: Barnett, G./ Holt, M. (2014)

⁶⁰ Enthalten in: Jones, T./ Schulte, R./ Cantara, M. (2014)

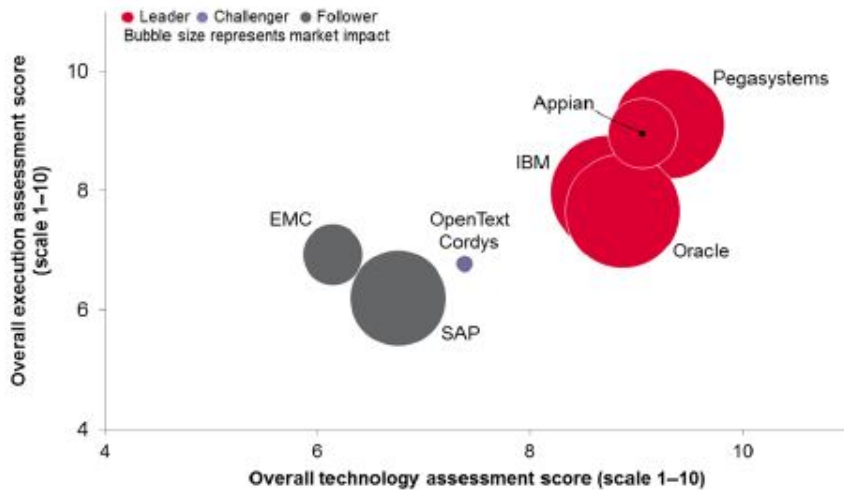


Abbildung 4: Ovum Decision Matrix for Selecting a Business Process Management Solution 2014⁶¹

Die ursprüngliche Idee dieser Seminararbeit war es, kommerzielle Software mit Open Source Software hinsichtlich der Funktionalität, Service, etc. zu vergleichen. Jedoch ist es online nicht möglich stichhaltige Informationen zu den verschiedenen kommerziellen Systemen zu erhalten, da die Internetauftritte der verschiedenen Anbieter ihren Fokus auf Marketingaspekten der Software legen. Aus diesem Grund war es nicht möglich einen Vergleich, der auf belegbaren Informationen beruht, durchzuführen. Dennoch ist es wichtig bei der Entscheidung, ob ein kommerzielles System gekauft werden soll oder ein Open Source System implementiert wird, Kontakt mit verschiedenen Anbietern kommerzieller Software aufzunehmen und sich ausführlich beraten zu lassen. Hierbei muss klar ein Fokus auf technische Daten gesetzt werden, um eine Vergleichbarkeit herzustellen. Die im folgenden Verlauf der Arbeit entwickelten Kriterien können eine Richtlinie hierfür darstellen und als Grundlage für eine Analyse genutzt werden.

5 Bewertungskriterien

5.1 Vorstellung der Bewertungskriterien

Um die Workflow-Management-Systeme gezielt bewerten zu können, sollten verschiedene Kriterien beachtet werden. Dabei kann sich die Relevanz der Kriterien davon unterscheiden, ob es sich um ein kommerzialisiertes oder Open Source Produkt handelt.

⁶¹ Enthalten in: Barnett, G./ Holt, M. (2014)

Hier zunächst Kriterien im Überblick, weitere Erläuterungen befinden sich im nachfolgenden Text:

Qualität der Firma:

- Gründungsdatum
 - Häufigkeit der Updates
 - Datum des letzten Releases
 - Entwickleranzahl
 - Größe der Community
 - Lizenz
 - Support
 - Trainings und Schulungen
 - Beratungsmöglichkeiten
- } Reifegrad

Allgemeine Produktinformationen:

- Test- oder Demoversion
- Verwendete Standards (BPMN, XPD, etc.)
- Generierte Dateiformate
- Integrationsmöglichkeiten mit anderen Systemen
- Hardwareanforderungen
- Virtualisierung
- Sicherheit
- Unterstützte Betriebssysteme

Funktionalität:

- Unterstützte Geräte
- Remotezugriff
- Cloud
- Backups
- Updates
- Schnittstellen
- Import systemfremder Objekte
- Unterstützte Entwicklungssprachen
- Erweiterbarkeit
- Unterstützte Frameworks und Plattformen
- Automatisierungsgrad
- Definition unstrukturierter Arbeitsabläufe

- Simulation und/oder Instanziierung
- Graphische Oberfläche
- Bedienbarkeit
- Adaptierbarkeit
- Monitoring
- Berichte
- Rollenbasierte Aufgaben

Grundsätzlich spielt es (ins Besondere für Open Source Produkte) eine Rolle, wie der Reifegrad der Firma aussieht, das heißt, wie etabliert und zukunftsicher die produzierende Firma ist. Dabei konstituieren *Gründungsdatum*, *Häufigkeit der Updates* und das *Datum des letzten Releases* gute Anhaltspunkte. Eine Firma, die beispielsweise erst vor drei Monaten gegründet wurde, stellt ein höheres Risiko dar, da nicht mit Sicherheit davon ausgegangen werden kann, dass sie auch langfristig bestehen bleibt. Des Weiteren sollten gerade so grundlegende Systeme wie WfMS regelmäßig auf Aktualität überprüft und ggf. angepasst werden. Wenn also das letzte Update vor einigen Jahren rausgekommen ist, könnte es genauso gut sein, dass das Projekt eigentlich schon tot ist. Es können Sicherheitslücken entstehen oder Fortschritt und damit auch wirtschaftliche Wettbewerbsvorteile verpasst werden. Von daher sollte darauf geachtet werden, dass regelmäßig in angemessenen Abständen Patches, Updates und auch neue Releases erscheinen und auch das letzte noch nicht zu lange her ist; somit also die Aktivität des Projekts gewährleistet ist.

Die *Entwickleranzahl* oder auch die *Größe der Community* gibt Auskunft darüber, wie verlässlich das Produkt ist. Wenn ein Produkt nur von zwei Entwicklern programmiert wird und diese sich zerstreiten, das Interesse verlieren oder ihnen sonst etwas geschehen sollte, besteht die Wahrscheinlichkeit, dass das Projekt stirbt. Ähnlich verhält es sich, wenn das Produkt nur von wenigen Unternehmen genutzt wird, könnte sich die Meinung einstellen, dass sich der Aufwand der Programmierung nicht mehr rentiert.

Die *Lizenz* ist insofern von Bedeutung als dass ins Besondere bei Open Source Lizenzen strenge Regeln gelten, was zusätzlich programmiert werden darf, bzw. was gegebenenfalls auch öffentlich zur Verfügung gestellt werden muss. Außerdem muss darauf geachtet werden, welche Art von Lizenzen angeboten werden und welche Sinn machen, z.B. site-Lizenzen, User-Lizenzen oder Floating-Lizenzen. So was spiegelt sich in den Kosten wieder und schränkt je nach dem die Adaptierbarkeit des Programms ein.

Unterschiedliche Firmen bieten unterschiedlichen *Support* an. Einige bieten nur Kommentarfelder auf der Webseite an, auf die dann eventuell geantwortet wird, andere haben Email oder Telefonsupport. Solche Angebote sind teils in Enterprise-Versionen mit inbegriffen, müssen aber in aller Regel in allen Fällen bezahlt werden. Da WfMS so umfangreich und

grundlegend mit den Unternehmensaktivitäten verankert sind, ist Support je nach Kapazität und Kompetenz der eigenen Inhouse-IT durchaus eine sinnvolle Überlegung.

Ähnliches gilt für *Trainings*, *Schulungen* und *Beratungsmöglichkeiten*. Eine grundsätzliche Beratung kann helfen herauszufinden, welches System, oder ob ein bestimmtes System für das entsprechende Unternehmen sinnvoll oder angemessen ist und eine Empfehlung für die Lizenzen geben. Ins Besondere für eher wirtschaftlich orientierte Benutzer helfen Trainings und Schulungen lange Einarbeitungszeiten und Fehler zu vermeiden und die Systemakzeptanz zu fördern. Nicht jedes Unternehmen bietet solche Schulungen und/oder Trainings an und selbstverständlich ist auch hierfür ein Entgelt zu entrichten.

Da die Entscheidung für ein Workflow-Management-System oft kurz- bis mittelfristig irreversibel ist, sollte vorher genauer geprüft werden, wie abhängig man sich vom Produkt und somit von einem anderen Unternehmen macht. Hierbei kann eine kostenlose oder kostengünstige *Test- oder Demoversion* durchaus helfen herauszufinden, ob das System zum Unternehmen passt und auf Benutzerakzeptanz stoßen kann. Außerdem gewinnen dabei vor allen Dingen *Standards*, die unter anderem von der WfMC etabliert wurden, an Bedeutung. BPMN, XPDL und Ähnliche erleichtern grundsätzlich den Wechsel von Anbieter zu Anbieter. Dabei sollten erstellte Objekte und Prozesse und im Prozessverlauf generierte Dateien in gängigen und von mehreren Betriebssystemen und Programmen unterstützten *Dateiformaten* abgespeichert werden, um auch hier eine Flexibilität zu sichern. Es kann auch oft von Vorteil sein, ein WfMS mit anderen System und anderer Software *integrieren* zu können. Oft bedeutet es immense Vorteile ein solches System beispielsweise mit einem Enterprise-Resource-Planning System, diversen Datenbanken oder Ähnlichen integrieren zu können, um eine Interoperabilität und somit den reibungslosesten Arbeitsablauf her zu stellen.

Allgemein soll ein WfMS oft dafür eingesetzt werden, um Kosten zu reduzieren, von daher kann es durchaus auch relevant sein, welche *Hardwareanforderungen* so ein System mit sich bringt. Zum einen ist es wichtig, genug Festplatten- und Cachespeicher bereit zu halten, um die Performanz zu gewährleisten, zum anderen gibt es Programme, die nur auf einen einzig für sie dedizierten Server laufen. Ein Server kann unter Umständen eine Anschaffung von ca. 15.000\$ sein, was unter dem Return on Investment und der Dauer, bis sich die Anschaffung amortisiert hat, zu berücksichtigen ist. Dabei könnten durchaus auch *Virtualisierungsmöglichkeiten* interessant sein. Was bei solch wichtigen Anwendungen mit so vielen Schnittstellen nicht außer Acht gelassen werden sollte, ist die *Sicherheit*. Welche Authentifizierungsmöglichkeiten sind vorgesehen? Es sollte auch eine Möglichkeit geben, verschiedene Zugriffsrechte verwalten zu können, sodass einige zum Beispiel das Recht haben, ganze Prozesse aufzusetzen, andere dürfen sie nur einsehen, oder nur an bestimmte Stellen mit ihnen interagieren. Wie können Zugriffsrechte auf den Server verwaltet werden? Sind etwaige Verbindungen verschlüsselt? Und ähnliche Fragen sollten durchaus Beachtung finden.

Dabei ergibt sich auch die Frage, welche *Betriebssysteme* das System unterstützt. Muss der Server auf Linux oder Unix laufen, oder läuft das Programm auch, wenn er mit Windows aufgesetzt wird.

Daraus ableitend stellt die Funktionalität des WfMS die Hauptkriterien an. Das fängt mit grundlegenden Aspekten wie den *unterstützten Geräten* an, mit denen man die Arbeitsabläufe einsehen, mit ihnen interagieren und ggfs. sie modellieren kann. Einige ermöglichen dies lediglich über Personal Computer, andere auch über mobile Geräte wie Tablets oder Smartphones. Daraus lässt sich der *Remotezugriff* herleiten, der über mobile Geräte generiert wird. Da die *Cloud* immer mehr an Bedeutung gewinnt, könnte es wichtig sein, dass sich das WfMS auch auf eine Cloud anwenden lässt und problemlos auf die dort gespeicherten Dateien zugreifen kann.

Grundsätzlich erscheint es sinnvoll, dass regelmäßige *Backups* generiert werden, um die Arbeitsabläufe und dabei erstellten Dateien und Dokumente zu speichern, da sie für das Unternehmen später weiterhin von Relevanz sein könnten, zum Beispiel um bei einer Kundenbeschwerde nachzuvollziehen, was mit Sicherheit wann und wo eingegangen ist, bzw. wo Fehler aufgetreten sind.

Durch den Umfang und die Kritikalität, die WfMS für Unternehmen darstellen, fallen *Updates* ins Gewicht. Automatische Notifikationen, falls diese bereit stehen, würden dem Unternehmen helfen, Sicherheitslücken zu vermeiden und auf dem aktuellsten Stand zu bleiben. Regelmäßige auf der Produktwebseite selbst nach Updates zu suchen, erscheint zeitraubend, umständlich und tendiert dazu, leicht vergessen zu werden.

Damit ein Workflow-Management-System tatsächlich das ganze Unternehmen abdeckt, bilden *Schnittstellen* eine essentielle Grundlage. Hier haben verschiedene Produkt verschiedene Vorgefertigte, andere bieten die Option selbst welche zu schreiben. Dabei wäre es nützlich, wenn z.B. auch die Geschäftsregeln miteinbezogen werden könnten (was nur einige Systeme anbieten) oder Key Performance Indicators (KPIs) auch einbezogen und dabei noch automatisch berechnet (natürlich nur für Numerische) werden.

Diesem sehr ähnlich unterstützen verschiedene Produkte auch den *Import systemfremder Objekte*, wie z.B. Outlook-Tasklisten, SharePoint-Objekten und Ähnlichem.

Um Schnittstellen gegebenenfalls selbst zu schreiben spielen die *unterstützten Entwicklungssprachen* eine Rolle. Viele Produkte sind Java geschrieben. Unternehmen könnten aber auch die meisten ihrer Anwendungen in C, C++ oder gar gänzlich anderen Sprachen haben, was Schnittstellen erschweren könnte. Zudem man Entwickler braucht, die Java mächtig sind.

Davon wird letztlich die *Erweiterbarkeit* des vorgefertigten Systems abhängig gemacht. Es gibt Hersteller die dies bereits einschränken (z.B. durch Lizenzen, aber auch durch vorprogrammierte Barrieren), aber auch welche, die dabei viel Spielraum lassen, um weitere Funk-

tionalitäten zu implementieren. Dies könnte darüber entscheiden, wie flexibel sich das Produkt an die Organisation und seine Anforderungen anpassen lässt.

Was des Weiteren mit der Erweiterbarkeit und Entwicklung von Schnittstellen relevant ist, sind die *unterstützten Frameworks und Plattformen*, die benutzt werden müssen, um etwas zu entwickeln bzw. zu benutzen, sowie Seam, Spring, OSGi oder andere.

Die beabsichtigte Kostenreduktion wird oft durch eine erhöhte *Automatisierung* der Arbeitsabläufe erzielt. Deswegen spielt der Grad bis zu dem das System dies unterstützt eine entscheidende Rolle, zum Beispiel ob mehrere Arbeitsschritte vollautomatisiert hintereinander ablaufen können.

Wie bereits erläutert, können Arbeitsschritte unterschiedliche Strukturierungen aufweisen. Dieser Arbeitsablauf wird von unterschiedlichen Produkten unterschiedlich definiert. Diese *Definition unstrukturierter Aufgabenabläufe* entscheidend über die Performanz und das Funktionenspektrum.

Es gibt einige Systeme, die hauptsächlich zur *Simulation* dienen. Damit kann man dann schauen, ob die Prozessabläufe richtig erdacht wurden, wo Fehler bestehen etc. Andere Systeme können diese Aktivitäten direkt *instanzieren* und auch ausführen, was grundsätzlich Zeit und Kosten sparen sollte.

Um die Arbeitsschritte zu gestalten und zu erstellen werden je nach Produkt verschiedenen Möglichkeiten angeboten. Einige haben *graphische Oberflächen*, die über ein „drag and drop“-Prinzip funktionieren und je nach Bedarf angepasst werden können. Andere sind nur über Kommandozeile *bedienbar* und modellierbar. Wiederum andere benötigen Plug-Ins oder bieten die Wahl zwischen einer Webedition, einer graphischen Oberfläche oder Ähnlichem an. Hierbei sind einige Produkte so *adaptierbar*, dass je nach Nutzer verschiedene Ansichten gewählt werden können, je nachdem, ob jemand eher einen technischen oder wirtschaftlichen Blickwinkel hat/möchte. Es kann zwischen verschiedenen Bearbeitungs- und Ansichtsmodi gewechselt werden und die Oberflächen sind personalisierbar.

Aus diesem Konzept lassen sich zwei Kriterien ableiten. Einmal das *Monitoring*, was noch nicht standardisiert jedes Produkt anbietet, aber für die hierarchisch höheren Entscheidungsträgern von Bedeutung ist, um bei Fehlern im Ablauf schnell eingreifen zu können und falls etwas schief geht, oder etwas anderes besonders gut, im Nachhinein ausfindig machen zu können, woran das lag und dies entsprechend zu unterbinden oder auf andere Gebiete anzuwenden. Hierfür können automatisch generierte *Berichte* und Log-Dateien unterstützen, die die Fehlerbehebung eines Systems unterstützen, aber auch noch nicht Standard sind. Zum anderen ergeben sich aus der Personalisierbarkeit der Systeme die *rollenbasierten Aufgaben*. Das bedeutet, dass genau festgelegt wird im System an welcher Stelle welcher Mensch mit dem System interagieren muss, um den Arbeitsablauf voranzutreiben und das System diese vom Menschen auszuführenden Tätigkeiten ebenfalls verwalten kann.

Die Kriterien sind stark von den Unternehmensanforderungen abhängig, weshalb keine allgemeine Gewichtung vorgenommen werden kann. Außerdem werden nicht alle Kriterien für alle Organisationen überhaupt relevant sein und andere werden eventuell sogar noch weitere brauchen. Zusätzlich zu den Kriterien muss also eine Organisation zusätzlich abwägen, ob das System zu ihrer Größe und Komplexität passt. Bei einem Handwerker, zum Beispiel einem Fliesenleger, sind die meisten Aktivitäten sehr standardisiert und auch mit der Zeit gibt es wenig Weiterentwicklung. Dann ist es für das System nicht zwangsweise entscheidend, ob die Firma 25 oder 250 Mitarbeiter hat. Eine Designberatung oder allgemein kreativere Unternehmen sind oft komplexer aufgestellt, mit weniger standardisierten Aktivitäten, die sich außerdem zunehmend verändern können. Hier ist dann die Komplexität ein wichtiger Aspekt, der berücksichtigt werden muss und nicht so sehr die Größe. Größere Unternehmen mit durchschnittlich komplexen Aufgaben, zum Beispiel Softwaredienstleister werden die Einbindung an ein Lightweight Directory Access Protocol (LDAP) oder ähnlichen Verzeichnissen benötigen, um ihre Benutzer zu verwalten, da es sonst zu aufwendig wäre, für jeden Mitarbeiter einen neuen Benutzer anzulegen. Hier muss also auf Grund der Größe auf eine Besonderheit geachtet werden.

Auch sollte bei Open Source Produkten nicht vergessen werden, dass diese zwar Schulungen und Trainings anbieten, es jedoch oft auch je nach Produkt und Bedienbarkeit essentiell darauf ankommt, dass die eigene IT-Abteilung in der Organisation schon über gewisse Programmierkenntnisse verfügt, um auch problemlos und ohne großen Zeitaufwand das System verwalten und anpassen kann.

Wer allgemein ein Enterprise-Resource-Planning (ERP)-System sucht, dessen Teil ein Workflow-Management-System sein soll, kann sich auch auf <http://erp-software.org/workflow-management-system/> informieren. Das ist eine Webseite, die in Kooperation mit der Rheinisch-Westfälischen Technischen Hochschule Aachen entwickelt wurde. Sie unterscheidet zunächst zwischen Unternehmensgröße, Branche und Benutzeranzahl. Anschließend wählt man verschiedene Kriterien aus wie Erweiterbarkeit, Anpassungsfähigkeit, (im Allgemeinen einige, die auch oben aufgeführt sind, nur grobkörniger), anschließend Lizenzmodell, Einführungsart und –termin, Preiskategorie usw. Am Ende werden einem alle passenden ERP-Systeme rausgesucht. Wenn man sich vorher noch kostenlos registriert, erhält man noch nähere Informationen über die Anbieter, aber auch so kann die Webseite einen guten ersten Überblick verschaffen und zeigt auch auf, worüber man sich allgemein vor der Einführung (nicht nur der Auswahl) Gedanken machen muss.⁶²

⁶² Vgl. ERP-software (2014)

5.2 Bewertungstabelle

Die Bewertungstabelle dient als Grundlage zur Auswertung verschiedener Open Source WfMS.

Generell ist es gemäß der in Kapitel 5.1 entwickelten Kategorien und den zuzuordnenden Kriterien. Es ist möglich für jede Kategorie separat eine Bewertung der Wichtigkeit auf einer Skala von eins bis zehn abzugeben, wobei eins als unwichtig und zehn als sehr wichtig definiert ist. Selbiges ist für die einzelnen Kriterien durchzuführen. Auf diesen ersten Schritt folgt die Evaluation und Bewertung der einzelnen Kriterien.

Zur Berechnung der Gesamtbewertung des Systems wird folgende Formel verwendet:

$$(Kategorie\ 1 \times (Bewertung\ Kriterium\ 1.1 \times Wichtigkeit\ Kriterium\ 1.1) + (Bewertung\ Kriterium\ 1.2 \times Wichtigkeit\ Kriterium\ 1.2) + \dots) + (Kategorie\ 2 \times (Bewertung\ Kriterium\ 2.1 \times Wichtigkeit\ Kriterium\ 2.1) + (Bewertung\ Kriterium\ 2.2 \times Wichtigkeit\ Kriterium\ 2.2) + \dots) + (Kategorie\ 3 \times (Bewertung\ Kriterium\ 3.1 \times Wichtigkeit\ 3.1) + (Bewertung\ Kriterium\ 3.2 \times Wichtigkeit\ Kriterium\ 3.2) + \dots)$$

Die Bewertungstabelle liefert jedoch keinen optimalen Wert, an den man möglichst nahe heran kommen sollte, sondern basiert vor allem auf den individuellen Anforderungen des Unternehmens an das WfMS. Somit liefern die Ergebnisse der Auswertung vor allem eine Vergleichbarkeit, wie gut die zu vergleichenden Systeme die Anforderungen an das System erfüllen. Als Entscheidungsgrundlage kann das System mit der höchsten Bewertung bevorzugt werden. Somit ist es essentiell verschiedene Systeme miteinander zu vergleichen, um aussagekräftige Resultate zu erlangen.

Im Rahmen dieser Arbeit wurde die Bewertungstabelle exemplarisch für ein fiktives Unternehmen mit Schätzungen ausgefüllt. Die hierbei verglichenen WfMS sind camunda und Bonita BPM 6. Da es sich bei der Ausarbeitung um ein fiktives Unternehmen mit fiktiven Anforderungen an das System handelt, kann das Resultat der Bewertungstabelle nicht als aussagekräftige allgemeingültige Empfehlung gesehen werden, sondern vielmehr als Beispiel wie die Bewertungstabelle anzuwenden ist und wie die Ergebnisse unter Umständen aussehen können. Somit dient das erarbeitete Beispiel, vor allem in Hinblick auf die Wichtigkeit der Kategorien und Kriterien, als Informationsmaterial für die Funktionsweise der Bewertungstabelle. Die Bewertung der WfMS selbst, wurde anhand der Informationen, die im Zuge der Recherche und Informationsbeschaffung für diese Arbeit getätigt wurde, ausgefüllt.

Die Wichtigkeit der Kategorie *Qualität der Firma* wurde für das ausgewählte fiktive Unternehmen mit dem Wert auch bewertet und ist somit wichtig für das Unternehmen. In der Bewertung dieser Kategorie erreicht camunda mehr Punkte als Bonita BPM 6 und erfüllt somit die Erwartungen zu einem größeren Teil. Die zweite Kategorie *Allgemeine Produktinformati-*

onen ist eher unwichtig für das fiktive Unternehmen und wurde deshalb mit einem Wert von 4 belegt. Ausschlaggebend für das Resultat dieser Kategorie war vor allem die Dichte an Informationen, die man zu den verschiedenen Kategorien finden konnte. Waren keine expliziten Informationen zu einem Kriterium verfügbar, ist dieses mit dem Wert 0 versehen worden und einem Kommentar, dass keine Angabe durch den Hersteller verfügbar war. Nichtsdestotrotz erreicht auch camunda in dieser Kategorie eine höhere Gesamtbewertung als Bonita BPM 6. Die wichtigste und dritte Kategorie stellt die *Funktionalität* des WfMS dar. Da dies der Kern der Software ist, erhält es auch die Wichtigkeitswert 10 für das ausgewählte fiktive Unternehmen. Für diese Kategorie ist anzumerken, dass vor allem für das WfMS von camunda einige Informationen nicht vorhanden waren, sodass auch hier Bonita BPM 6 besser abschneidet. Jedoch könnte sich dieser Wert natürlich ändern durch mehr öffentlich zugängliche Informationen. Wenn man alle drei Kategorien berücksichtigt und die Zwischenresultate der Kategorien für jedes System addiert, liegt Bonita BPM 6 vor camunda und wäre somit für das fiktive Unternehmen besser geeignet.

Die voll ausgefüllte Tabelle, wird als Zusatzdokument zu dieser Arbeit abgegeben. Abbildung 5 zeigt einen Ausschnitt aus der Ergebnistabelle für das Produkt camunda.

				camunda		
Kategorie	Wichtigkeit der Kategorie (Skala 1–10)	Kriterium	Wichtigkeit des Kriteriums (Skala 1–10)	Bewertung des Kriteriums	Kommentar	Gesamtbewertung (wird automatisch berechnet)
Qualität der Firma	8	Gründungsdatum	5	3		15
		Häufigkeit der Updates	8	10		80
		Datum des letzten Releases	8	10		80
		Entwickleranzahl	8	6		48
		Größe der Community	8	7		56
		Lizenz	9	9		81
		Support	10	9		90
		Trainings und Schulungen	10	10		100
		Beratungsmöglichkeiten	8	10		80
		Zwischenresultat Kategorie 1:				

Abbildung 5: Beispiel Bewertungstabelle

6 Installation und Beispielprozess eines Workflow-Management-Systems am Beispiel von Camunda BPM

Vor dem Installationsbeginn muss sichergestellt werden, dass es folgende Applikationen und Tools runtergeladen wurden:

- Camunda BPM
- Camunda Modeler
- Java JDK 1.6+
- Apache Maven
- Eclipse
- Webbrowser (Firefox, Chrome, Internet Explorer)

Der benötigte Webserver Tomcat ist bereits im Camunda BPM Paket enthalten.

Nachdem alles runtergeladen wurde, kann mit dem Setup von dem Projekt angefangen werden. Dieses Setup besteht aus vier Schritten. Im ersten Schritt wird ein Maven Projekt in Eclipse erstellt, um später eine WAR-Datei für Tomcat zu erzeugen. Danach werden Camunda Abhängigkeiten in einer pom.xml Datei definiert. Im dritten Schritt wird die @ProcessApplication Klasse erstellt. Sie soll die Schnittstelle zwischen der Applikation und der Prozess Engine sein. Als letztes soll eine META-INF/processes.xml Datei hinzugefügt werden. Diese Datei soll erlauben, dass eine deklarative Konfiguration des Deployments zur Verfügung zu stellen, die die Prozess Applikation erstellt.⁶³

Nachdem diese vier Schritte durchgeführt worden sind, kann der erste Prozess modelliert werden. Dafür wird ein BPMN 2.0 Eclipse Plugin benötigt. Wenn es nicht in Eclipse beinhaltet ist, muss es heruntergeladen werden.

⁶³ Vgl. camunda services GmbH (2013c)

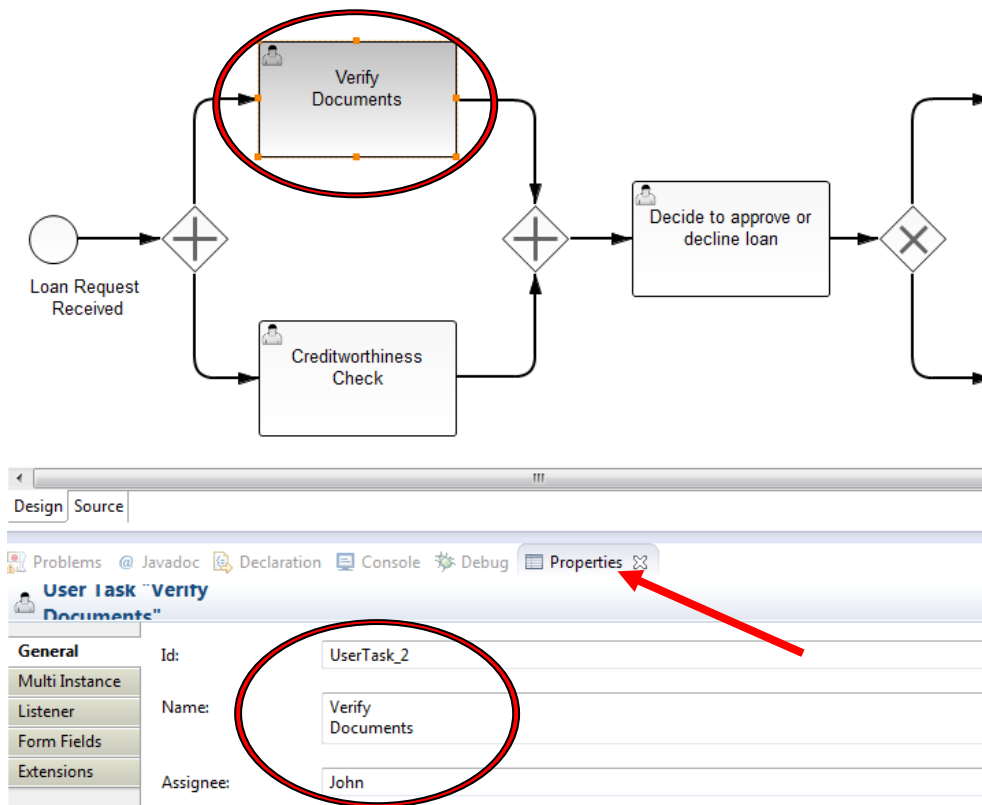


Abbildung 6: Eigenschaften der Schritte

Wie es in Abbildung 6 veranschaulicht wird, soll jedem Schritt eine eigene ID, einen Name und eine für die Aufgabe zuständige Person zugewiesen werden, wenn es sich um eine manuelle Aufgabe handelt. Nachdem die Eigenschaften zu jedem Schritt definiert worden sind, sollen die Eigenschaften für den gesamten Prozess ausgefüllt werden. Wie es in Abbildung 7 zu sehen ist, werden hier die Process ID und der Name definiert. Des Weiteren ist es ein sehr wichtiger Schritt, ein Häkchen bei Is Executable zu setzen, damit dieser Prozess ausführbar wird.

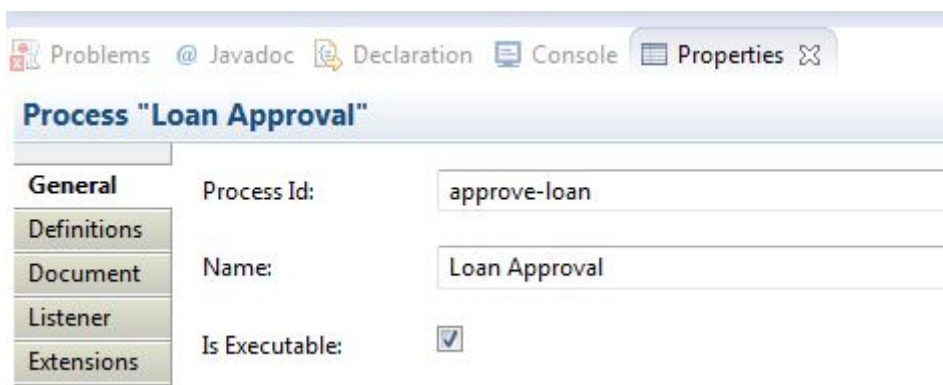


Abbildung 7: Eigenschaften des gesamten Prozesses

Nachdem alle Eigenschaften ausgefüllt sind, kann eine WAR-Datei für Tomcat erzeugt werden. Dafür soll die pom.xml Datei als Maven Install ausgeführt werden. Die daraus erzeugte WAR-Datei soll jetzt in den \$CAMUNDA_HOME/server/apache-tomcat/webapps Ordner kopiert werden. Im folgenden kann der Tomcat Webserver gestartet werden. Dies kann mithilfe der mitgelieferten startup.bat Datei erfolgen.⁶⁴

Wenn der Webserver gestartet ist, wird in dem Browser folgende Seite gestartet: <http://localhost:8080/camunda-welcome/index.html>. Abbildung 8 veranschaulicht die Startseite von Camunda BPM Platform.

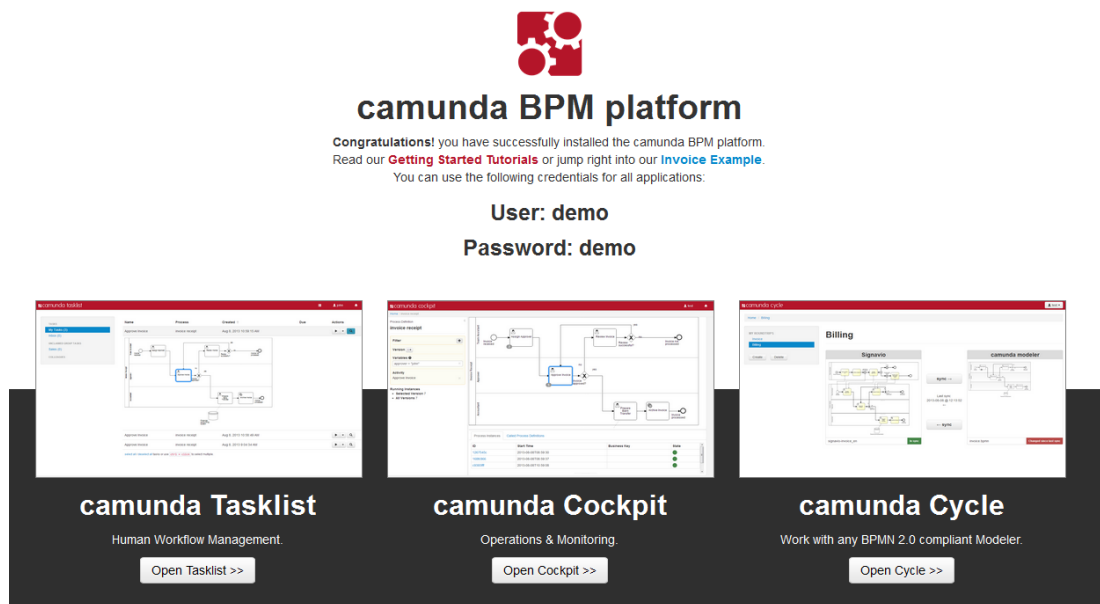


Abbildung 8: Camunda BPM Platform Startseite

Von hier aus können die zwei wichtigsten Tools von Camunda gestartet werden. Das sind die Camunda Tasklist, wo jeder Benutzer ihm zugewiesene Aufgaben sehen und erledigen kann, und Camunda Cockpit, welches als Monitoring Tool dient. Zusätzlich, wenn der Benutzer Administrator Rechte besitzt, kann er z.B. neue Benutzer anlegen, Benutzer Accounts verarbeiten und verschiedene Rechte zuweisen.⁶⁵

Wichtig ist zu erwähnen, dass jede Änderung, die in Eclipse durchgeführt wird, nicht automatisch im Browser angezeigt wird. Um diese Änderungen zu sehen, soll wieder eine neue WAR-Datei erzeugt und in den Tomcat webapps Ordner kopiert werden. Danach soll auch Tomcat neugestartet werden.

Die Abbildung 9 veranschaulicht einen Beispielprozess, der während dieser Arbeit mithilfe von BPMN 2.0 erstellt wurde. Hier wird eine vereinfachte Form des Prozesses der Prüfung eines Kreditantrages abgebildet.

⁶⁴ Vgl. camunda services GmbH (2013c)

⁶⁵ Vgl. camunda services GmbH (2013c)

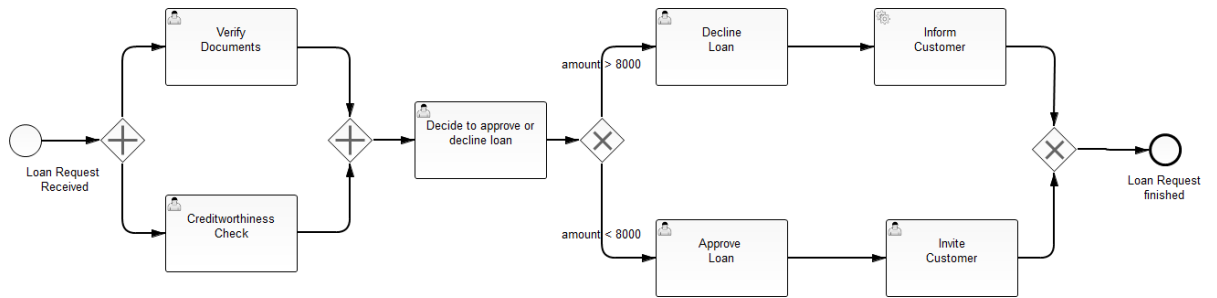


Abbildung 9: Loan Approval Process

Nachdem der Kreditantrag eingegangen ist, was als Start Event abgebildet ist, werden parallel zwei Aufgaben gestartet. Diese Aufgaben stellen zwei manuelle Aufgaben dar, die mit einem Parallel Gateway verbunden sind. Erst nachdem beide Aufgaben erfüllt sind, kann die dritte Aufgabe gestartet werden. Hier soll der Benutzer, anhand der davor gelieferten Daten, entscheiden, ob der Kredit genehmigt oder abgelehnt werden soll. In diesem Fall wird die Höhe des Kredites als Hauptkriterium genommen. Wenn dieser Kredit die Grenze von 8000 Euro nicht überschreitet, wird er automatisch genehmigt. Ansonsten wird er abgelehnt. Die Schritte Decline Loan und Approve Loan sind mit einem Exclusive Gateway verbunden, das einem exklusiven Oder entspricht.

Wenn der Kredit genehmigt wird, bekommt der Benutzer die letzte manuelle Aufgabe, den Antragsteller zu einem Gespräch einzuladen. Nachdem das passiert ist, wird der Prozess beendet, was mit einem End Event abgebildet wird.

Wenn der Kredit abgelehnt wird, wird der Antragsteller automatisch, mithilfe eines Service Tasks, über die Ablehnung informiert. Danach wird der Prozess beendet.

7 Fazit

Generell lässt sich sagen, dass Open Source Workflow-Management-Systeme, abhängig von den Anforderungen des jeweiligen Unternehmens, eine gute Alternative zu kommerziellen Systemen darstellen können. Allerdings ist es nicht möglich eine allgemeingültige Empfehlung über das beste Open Source System zu geben. Jedes Unternehmen besitzt verschiedene Anforderungen und Probleme, die mithilfe eines Workflow-Management-Systems gelöst werden sollen. Aus diesem Grund ist es empfehlenswert zielgerichtete Recherche anhand von vorher ausgewählten Kriterien zu betreiben. Als Grundlage für die Bewertung der näher betrachteten Systeme kann die im Rahmen dieses Projekts erstellte Excel-Bewertungstabelle verwendet werden. Hierdurch werden die verschiedenen Systeme in Anlehnung an spezifische Anforderungen des Unternehmens vergleichbar. Ähnlich bei jeder anderen Standardsoftware besitzt jedes System seine Vor- und Nachteile, die abhängig von den Anforderungen an das Systems, verschieden gravierend sein können und somit abgewägt werden müssen. Da die Beschreibungen der einzelnen Open Source WfMS online oft sehr allgemein und oberflächlich gestaltet sind, ist es ratsam die infragekommenden Systeme mithilfe von Test- oder Demoversionen auszuprobieren und auf Erfüllung der Anforderungen und Benutzerfreundlichkeit zu evaluieren. Im Gegensatz zu kommerziellen Systemen erfordern Open Source Systeme häufig eine starke Einbindung der IT-Abteilung eines Unternehmens. Des Weiteren muss vor allem eine starke Benutzerfreundlichkeit vorhanden sein, um dem Anwender, der häufig nicht Teil der IT-Abteilung ist, die Nutzung der Software zu erleichtern. Nur wenn sowohl die IT-Abteilung als auch der Anwender die neu eingeführte Software akzeptieren, kann diese auch sinnvoll und effizient genutzt werden. So kann das Ziel der Kosteneinsparung und transparenterer Prozesse mit geringerem Administrationsaufwand erreicht werden.

Ergänzend zu der Erfüllung der Anforderung des ausgewählten Systems an das Open Source WfMS, kann auch ein Vergleich mit Unternehmen, die in einem ähnlichen Marktsegment tätig sind, von Vorteil sehen. So wird ein Austausch über bereits getätigte Erfahrungen angestoßen. Es ist relativ wahrscheinlich, dass beide Unternehmen ähnliche Ansprüche an das System haben. Hierdurch können eventuelle Fehlentscheidungen vermieden werden.

Quellenverzeichnisse

Literaturverzeichnis

Bandel, G. (2010): Open Source Software: Fördert oder hemmt diese Art der Softwareentwicklung den Wettbewerb?, Hamburg: Diplomica Verlag GmbH

Böhm, M./ Jablonski, S. (1997): Workflowmanagement. Entwicklung von Anwendungen und Systemen – Facetten einer neuen Technologie, Heidelberg: dpunkt Verlag

Grassmuck, V. (2004): Freie Software – Zwischen Privat-und Gemeineigentum, Bonn: Bundeszentrale für politische Bildung

Helbig, L./ Zhang, J./ Zierke, M. E. (2005): Workflow-Management, ein Vergleich verschiedener Systeme, Augsburg

Keßler, S. (2013): Anpassung von Open-Source-Software in Anwenderunternehmen – Entwicklung und Management von Informationssystemen mit Datenauswertung, Wiesbaden: Springer Fachmedien

Müller, J. (2005): Workflow based integration, Berlin: Springer-Verlag

Oberhem, C. (2008): Vertrags- und Haftungsfragen beim Vertrieb von Open Source Software, Hamburg: Verlag Dr. Kovac

Porter, M. E. (1999): Wettbewerbsvorteile (Competitive Advantage): Spitzenleistungen erreichen und behaupten, 6. Auflage, Frankfurt: Campus Verlag

Wichmann, T. (2005): Linux. Und Open-Source-Strategien, Berlin: Springer Verlag

Internetquellen

Activiti (2014b): Activiti Components, <http://activiti.org/components.html>, Abruf: 09.06.2014

Activiti (2014a): Activiti BPM Platform, <http://www.activiti.org/>, Abruf: 09.06.2014

Barnett, G./ Holt, M. (2014): Ovum Decision Matrix: Selecting a Business Process Management Solution 2014, <http://ovum.com/research/ovum-decision-matrix-selecting-a-business-process-management-solution-2014/>, Abruf: 23.06.2014

Bonitasoft (w. y.a): Produktvergleich, <http://de.bonitasoft.com/produkte/produktvergleich>,
Abruf: 04.06.2014

Bonitasoft (w. y.b): Dienstleistungen, <http://de.bonitasoft.com/dienstleistungen>, Abruf:
04.06.2014

Bundeszentrale für politische Bildung (o.J.): Open Source,
<http://www.bpb.de/gesellschaft/medien/opensource/>, Abruf: 23.06.2014

Busskamp, D. (o.J.): Kapitel 1, <http://www.dbus.de/eip/kapitel01e.html>, Abruf: 23.06.2014

camunda services GmbH (2013a): camunda BPM platform,
<http://camunda.com/de/bpm/features/>, Abruf: 04.06.2014

camunda services GmbH (2013b): camunda BPM Enterprise Edition,
<http://camunda.com/de/bpm/enterprise/>, Abruf: 08.06.2014

camunda services GmbH (2013c): Getting Started, <http://camunda.org/get-started/developing-process-applications.html>, Abruf: 08.07.2014

Colosa, Inc. (2014a): ProcessMaker Open Source – For Small Deployments and Evaluation,
<http://processmaker.com/community-2>, Abruf: 27.06.2014

Colosa, Inc. (2014b): ProcessMaker 2.X Documentation,
http://wiki.processmaker.com/index.php/Main_Page, Abruf: 27.06.2014

Colosa, Inc. (o.J.): ProcessMaker BPM Software: Our Editions,
<http://www.processmaker.com/processmaker-bpm>, Abruf: 27.06.2014

Computerwoche (1998): Zur Geschichte der Open-Source-Software,
<http://www.computerwoche.de/a/zur-geschichte-der-open-source-software,1093333>, Abruf:
08.07.2014

ERP-software (2014): Der grösste unabhängige ERP-Software Vergleich, <http://erp-software.org/workflow-management-system/>, Abruf: 27.06.2014

Imixs Software Solution GmbH: (w. y.a): Managed Workflow Server, <http://www.office-workflow.de/losungen/managed-workflow-server/>, Abruf: 27.06.2014

Imixs Software Solution GmbH: (w. y.b): Unser Angebot, <http://www.office-workflow.de/losungen/imixs-office-workflow/>, Abruf: 27.06.2014

- Java-source-net (w. y.a) : Open Source Workflow Engines in Java, <http://java-source.net/open-source/workflow-engines>, Abruf: 27.06.2014
- Java-source.net (w. y.b): Enhydra Shark, java-source.net/open-source/workflow-engines/enhydra-shark, Abruf: 02.06.2014
- jBPM (w. y): jBPM, <http://jbpm.jboss.org/>, Abruf: 08.06.2014
- Jones, T./ Schulte, R./ Cantara, M. (2014): Magic Quadrant for Intelligent Business Process Management Suites, <http://www.gartner.com/technology/reprints.do?id=1-1RZ9B9J&ct=140318&st=sb>, Abruf: 23.06.2014
- Karagiannis, D. (2013): Enzyklopädie der Wirtschaftsinformatik, Workflow-Management-System, <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/wi-enzyklopaedie/lexikon/is-management/Systementwicklung/Softwarearchitektur/Middleware/Workflow-Management-System/index.html?searchterm=workflow%20management>, Abruf: 28.06.2014
- Kleijn, A. (2006): Open-Source-Lizenzen, <http://www.heise.de/open/artikel/Open-Source-Lizenzen-221957.html>, Abruf: 05.07.2014
- Lämmer, A., u.a. (2001): Open Source: Geschichte und Grundlagen, <http://ig.cs.tu-berlin.de/oldstatic/w2000/ir1/referate1/k-1a/>, Abruf: 08.07.2014
- MyControl (o.J.a): MyControl Workflow Server, <http://www.mycontrol.de/workflow/>, Abruf: 27.06.2014
- MyControl (o.J.b.): Dienstleistungen, <http://www.mycontrol.de/workflow/dienstleistung.thtml>, Abruf: 27.06.2014
- MyControl (o.J.c.): Systemvoraussetzungen, http://wfscentral.mycontrol.de/online_hilfe/devel/admin/systemvoraussetzungen.html, Abruf: 27.06.2014
- MyControl (o.J.d.): Funktionen, <http://www.mycontrol.de/workflow/funktionen.thtml>, Abruf: 27.06.2014
- Ohloh.net Black Duck (2014a): Bonita BPM, <http://www.ohloh.net/p/bonitabpm>, Abruf: 27.06.2014
- Ohloh.net Black Duck (2014b): Enhydra Shark, <http://www.ohloh.net/p/4278>, Abruf: 27.06.2014
- Ohloh.net Black Duck (2014c): jBPM, <http://www.ohloh.net/p/jbpm>, Abruf: 27.06.2014

Ohloh.net Black Duck (2014d): Imixs Workflow, <http://www.ohloh.net/p/imixs>, Abruf: 27.06.2014

Ohloh.net Black Duck (o.J.): ProcessMaker, <http://www.ohloh.net/p/processmaker>, Abruf: 05.07.2014

Open Source Initiative (2007): About the Open Source Initiative, <http://opensource.org/about>, Abruf: 23.06.2014

Schewe, G. (w. y.): Gabler Wirtschaftslexikon, Workflow Management, <http://wirtschaftslexikon.gabler.de/Definition/workflow-management.html>, Abruf: 28.06.2014

Schmidt, A. (2012): Software-Lizenzen – ein Überblick, <http://www.it-economics.de/blog/-/blogs/software-lizenzen-%E2%80%93-ein-uberblick>, Abruf: 05.07.2014

Taverna (2014a): Taverna, <http://www.taverna.org.uk/>, Abruf: 02.07.2014

Taverna (2014b): Taverna features, <http://www.taverna.org.uk/introduction/taverna-features/>, Abruf: 02.07.2014

Together Teamsolutions Co, Ltd. (2011): Together XPDL and BPMN Workflow Server, <http://www.together.at/prod/workflow/tws>, Abruf: 02.06.2014

Wahlers, J. (2013): Open-Source-Lizenzen – Ein Überblick, <http://www.wbs-law.de/it-recht/open-source-lizenzen-ein-uberblick-35752/>, Abruf: 05.07.2014

WfMC (w. y.): Workflow Management Coalition, <http://www.wfmc.org/>, Abruf: 15.06.2014

Konzeption und Implementierung eines Performance-Tests für Key-Value-Datenbanken

Schriftliche Ausarbeitung
im Rahmen der Lehrveranstaltung „Integrationsseminar“

Vorgelegt von

Christian Brummer
Michael Busam
Thorsten Kühner
Klarissa Schmidt
Matthias Welz

am 11.07.2014

Fakultät Wirtschaft
Studiengang Wirtschaftsinformatik
WI 2011 I (IMBIT)

Inhaltsverzeichnis

Abkürzungsverzeichnis	IV
Abbildungsverzeichnis.....	V
Verzeichnis der Quellcodes.....	V
Tabellenverzeichnis.....	VI
1 Einleitung	1
1.1 Motivation	1
1.2 Zielsetzung	2
1.3 Vorgehen	2
2 Theoretische Grundlagen	4
2.1 Relationale Datenbanken.....	4
2.1.1 Grundlegendes Konzept.....	4
2.1.2 Ausgewählte Produktvertreter.....	6
2.2 Key-Value Datenbanken	7
2.2.1 Entwicklung.....	7
2.2.2 Grundlegendes Konzept.....	11
2.2.3 Ausgewählte Produktvertreter.....	12
2.3 Zusammenfassender Vergleich.....	17
3 Konzeption und Umsetzung von Performancetests	19
3.1 Anwendungsfall.....	19
3.2 Konzeption von Testszenarien.....	20
3.2.1 Alle Lokationen auslesen.....	21
3.2.2 Füllstand aller Lokationen auslesen.....	21
3.2.3 Frequentierung der Lokation auslesen.....	21
3.2.4 Gesamte Verweildauer pro Paket in der Logistikkette auslesen.....	22
3.2.5 Historie aller Pakete auslesen	22
3.2.6 Historie zufälliger Pakete auslesen.....	22
3.2.7 Lokation aller Pakete auslesen	23
3.2.8 Lokation zufälliger Pakete auslesen	23
3.2.9 Pakete je Lokation auslesen.....	23
3.2.10 Pakete, die eine Lokation mehrfach besucht haben, auslesen	24
3.2.11 Stammdaten aller Pakete auslesen	24
3.2.12 Stammdaten zufälliger Pakete auslesen.....	24
3.2.13 Pakete bewegen (update).....	25
3.2.14 Archivierung von Paketen und ihrer Historie (delete)	25
3.2.15 Load-Test / Batch-Load-Test	25
3.3 Implementierung	26
3.3.1 Umsetzung der Datenmodelle	26
3.3.2 Testdatengenerierung	36
3.3.3 Testausführung	37
3.4 Durchführung der Messungen.....	40
3.4.1 Konfiguration der Testumgebung.....	41
3.4.2 Hardware.....	42
3.4.3 Konfiguration der Datenbanken	42

4	Auswertung der Messergebnisse.....	43
4.1	Beschreibung der Messergebnisse	43
4.1.1	Alle Lokationen auslesen.....	43
4.1.2	Füllstand aller Lokationen auslesen.....	44
4.1.3	Frequentierung der Lokation auslesen.....	44
4.1.4	Gesamte Verweildauer pro Paket in der Logistikkette auslesen.....	45
4.1.5	Historie aller Pakete auslesen	45
4.1.6	Historie zufälliger Pakete auslesen.....	45
4.1.7	Lokation aller Pakete auslesen	46
4.1.8	Lokation zufälliger Pakete auslesen	46
4.1.9	Pakete je Lokation auslesen.....	47
4.1.10	Pakete, die eine Lokation mehrfach besucht haben, auslesen	47
4.1.11	Stammdaten aller Pakete auslesen	47
4.1.12	Stammdaten zufälliger Pakete auslesen.....	48
4.1.13	Pakete bewegen (update).....	48
4.1.14	Archivierung von Paketen und ihrer Historie (delete).....	49
4.1.15	Load-Test.....	49
4.1.16	Batch-Load-Test.....	49
4.2	Interpretation der Ergebnisse in Hinblick auf die Testszenarien	50
4.2.1	Load	53
4.2.2	Update.....	54
4.2.3	Delete.....	54
4.2.4	Abfragen.....	54
4.3	Verlässlichkeit und Aussagekraft der Ergebnisse.....	57
4.4	Schlussfolgerung	59
5	Schluss.....	61
5.1	Zusammenfassung	61
5.2	Kritische Betrachtung.....	62
5.3	Ausblick	63
	Quellenverzeichnisse	65

Abkürzungsverzeichnis

ACID	Atomicity, Consistency, Isolation, Durability
ANSI	American National Standards Institute
AOF	Append Only File
API	Application Programming Interface
BDB	Berkeley Database
BSON	Binary JavaScript Object Notation
CPU	Computing Processing Unit
CSV	Comma-separated values
DB	Datenbank
DBMS	Database Management System
DCL	Data Control Language
DDL	Data Definition Language
DI	Dependency Injection
DML	Data Manipulation Language
FIFO	First In First Out
GC	Garbage Collection
IT	Informationstechnologie
JDBC	Java Database Connectivity
JSON	JavaScript Object Notation
LIFO	Last In First Out
MB	Megabyte
NoSQL	Not only SQL
PHP	PHP: Hypertext Preprocessor
SQL	Structured Query Language
TDD	Test-Driven Development

Abbildungsverzeichnis

Abb. 1: Veranschaulichung einer Datenbanktabelle	5
Abb. 2: Kategorisierung von NoSQL DBMS	9
Abb. 3: Value Typen in Redis	13
Abb. 4: Normalisiertes Datenmodell in MongoDB	16
Abb. 5: Denormalisiertes Datenmodell in MongoDB	17
Abb. 6: Umsetzung des Datenmodells in Klassen	26
Abb. 7: Drei Collections in MongoDbRepository	34

Verzeichnis der Quellcodes

Code 1: Anlegen der Datenbanktabellen	29
Code 2: Ermittlung von Paketen mit mehrfachem Lokationsbesuch	33
Code 3: Berechnung der Gesamtzeit eines Pakets für MongoDB.....	35
Code 4: Testdaten aus Wikipedia extrahieren	36

Tabellenverzeichnis

Tabelle 1: Methoden des Repository-Interfaces	26
Tabelle 2: Klassen des Repository-Interfaces pro Datenbank	28
Tabelle 3: SQL-Anweisungen für die Methoden des Repository-Interfaces	29
Tabelle 4: Keys der Objekte zur Verwaltung von Abhängigkeiten in Redis	31
Tabelle 5: Übersicht der Testklassen	37
Tabelle 6: Konfigurationsparameter der Testausführung	39
Tabelle 7: Konfiguration der Testumgebung	41
Tabelle 8: Hardware der Testumgebung	42
Tabelle 9: Ergebnis GetAllLocationsTest.....	44
Tabelle 10: Ergebnis GetAllLocationPackageCount	44
Tabelle 11: Ergebnis GetAllLocationTotalPackageCount	44
Tabelle 12: Ergebnis GetRandomPackageTrackingTime	45
Tabelle 13: Ergebnis GetAllPackageHistories	45
Tabelle 14: Ergebnis GetRandomPackageHistories	46
Tabelle 15: Ergebnis GetAllPackageLocations	46
Tabelle 16: Ergebnis GetRandomPackageLocations.....	46
Tabelle 17: Ergebnis GetPackagesAtAllLocations	47
Tabelle 18: Ergebnis GetAllMultipleVisitors	47
Tabelle 19: Ergebnis GetAllPackagesTest	48
Tabelle 20: Ergebnis GetRandomPackages	48
Tabelle 21: Ergebnis MoveRandomPackages.....	48
Tabelle 22: Ergebnis ArchiveRandomPackages.....	49
Tabelle 23: Ergebnis LoadTest.....	49
Tabelle 24: Ergebnis BatchLoadTest.....	50
Tabelle 25: Gesamtausführungszeit aller Tests.....	51
Tabelle 26: Abfragen nach Häufigkeit ihres Auftretens	51
Tabelle 27: Gesamtausführungszeit aller Tests gewichtet nach Szenarienhäufigkeit	52
Tabelle 28: Summe der Ausführungszeit der Load-Tests	53
Tabelle 29: Komplexe und einfache Abfragen	55
Tabelle 30: Ergebnis einfacher Abfragen.....	56
Tabelle 31: Ergebnis komplexer Abfragen.....	56
Tabelle 32: Komplett- und Zufallsabfragen	56
Tabelle 33: Ergebnis Komplettabfragen.....	57
Tabelle 34: Ergebnis Zufalls-/Teilabfragen	57

1 Einleitung

(Christian Brummer, Michael Busam, Thorsten Kühner, Klarissa Schmidt, Matthias Welz)

“Going forward, we see the bifurcation between relational and NoSQL DBMS markets diminishing over time.”¹ - So lautet die Meinung von Nick Heudecker, Research Director bei Gartner Inc. für Big Data und NoSQL, über die Zukunft operativer Datenbanksysteme. In der Gartner Studie „Magic Quadrant for Operational Database Management Systems“ wird ausgehend von der hohen Nachfrage an Datenbanksystemen in allen Unternehmensindustrien das derzeitige Angebot detailliert untersucht:² Obwohl relationale Datenbanken der großen IT-Unternehmen Oracle, IBM und Microsoft noch immer den Markt beherrschen, entwickeln sich in der jüngeren Vergangenheit zunehmend innovative Ansätze, um die wachsende Datenmenge verwalten zu können. Der IDC Studie „The Digital Universe in 2020“ zufolge wird im kommenden Jahrzehnt die weltweite Datenmenge auf 40.000 Exabytes ansteigen, was einem Wachstumsfaktor von 300 basierend auf dem Jahr 2005 entspricht³. Neben Schlagworten wie Big Data und Cloud Computing etabliert sich der Begriff von NoSQL Datenbanksystemen, die sich als Alternative zu den traditionellen relationalen Datenbanksystemen präsentieren und einen entscheidenden Beitrag für die Performance- und Skalierungsherausforderungen der wachsenden Datenmenge leisten können.⁴

Im Folgenden wird auf die Motivation sowie Problemstellung dieser Arbeit eingegangen und ein Vorgehen gewählt, um die definierte Zielsetzung zu erfüllen.

1.1 Motivation

Das Vorhandensein von Alternativen zu den etablierten relationalen Datenbanken wirft die Frage auf, ob NoSQL-Ansätze tatsächlich Vorteile bieten, und ob und wo sie leistungsfähiger sind als relationale Datenbanken. Dies ist insbesondere für Key-Value-Datenbanken noch wenig untersucht, sodass die Frage, ob in einem gegebenen Nutzungsszenario eine relationale oder eine Key-Value Datenbank zum Einsatz kommt, oft mehr auf den immer gleichen, theoretischen Argumenten beruht, aber nicht auf empirisch ermittelten Untersuchungen zur Leistungsfähigkeit.

¹ Heudecker, N. (2013)

² Vgl. Feinberg, D. / Adrian M. / Heudecker; N. (2013)

³ Vgl. Gantz, J. / Reinsel, D. (2012), S. 1

⁴ Vgl. Grolinger, K. / u.a. (2013), S. 1

1.2 Zielsetzung

Das zentrale Ziel dieser Arbeit ist es, die Leistungsfähigkeit verschiedener relationaler Datenbanken mit derjenigen einer Key-Value Datenbank auf der Basis verschiedener Testszenarien zu vergleichen. Zugleich wird es möglich sein, über diese standardisierten Tests die Leistungsfähigkeit verschiedener relationaler und NoSQL-Ansätze untereinander zu vergleichen. Die Leistungstests sollen auf einem möglichst realen Anwendungsfall eines Paketdienstleisters basieren, um tatsächliche Nutzungsszenarien für die verschiedenen Datenbanken und Klassen von Datenbanken aufzeigen zu können.

Neben dem hauptsächlichen Ergebnis dieser Arbeit, welches die Dokumentation, Diskussion und Interpretation der Ergebnisse und Einschränkungen des Leistungsvergleichs der Datenbanken ist, soll diese Arbeit weitere Teilergebnisse hervorbringen:

Eine Synopsis des derzeitigen Forschungsstands zu relationalen und insbesondere NoSQL- und Key-Value-Datenbanken wird Unterschiede, Gemeinsamkeiten und mögliche Anwendungsszenarien der verschiedenen Konzepte aufzeigen.

Schließlich soll die für die Testdurchführung entwickelte Java-Anwendung inklusive Testdatengenerator durch ihren modularen Aufbau die Möglichkeit bieten, auch über die im Rahmen dieser Arbeit zu testenden Szenarien hinaus weitere, modifizierte Szenarien zu testen.

1.3 Vorgehen

Diese Arbeit ist in vier Teile untergliedert: Die theoretischen Grundlagen zu relationalen und NoSQL-Datenbanken, die praktische Umsetzung der Leistungstests, die Diskussion der Messergebnisse und die Zusammenfassung und Reflexion.

Vor der tatsächlichen Konzeption und Umsetzung der vergleichenden Performancetests verschiedener relationaler und NoSQL-Datenbanken dient Kapitel 2 zunächst der theoretischen Diskussion der zugrundeliegenden Konzepte und Funktionsweisen der relationalen und NoSQL-Datenbanken, sowie der jeweiligen in der praktischen Implementierung genutzten Vertreter. Dabei liegt der Schwerpunkt auf der Beschreibung und Diskussion der NoSQL-, und dabei insbesondere der Key-Value-Datenbanken, während es zu den in der Literatur bereits ausführlich diskutierten relationalen Datenbanken nur eine kurze Einführung geben wird. Abgeschlossen wird der Theorieteil von einem zusammenfassenden Vergleich der beiden grundlegenden Konzepte.

Basierend auf dem Wissen aus dem Theorieteil beschreibt Kapitel 3 die tatsächliche Umsetzung der Leistungstests, beginnend mit der Beschreibung des Anwendungsfalles eines Paketdienstleisters, der die Grundlage für die Testdurchführung bilden wird. Ziel ist, ein möglichst realistisches Szenario zu erstellen, sodass im Anschluss für die verschiedenen Test-

szenarien neben der technischen Aspekte auch die wirtschaftliche Motivation erläutert wird. Im weiteren Verlauf beschreibt Kapitel 3 die technische Umsetzung der Testumgebung und -szenarien, und beschreibt die Konfiguration, unter welcher die Leistungsmessungen durchgeführt werden.

Innerhalb des praktischen Teils wird eine Untergliederung in die Konzeption und Implementierung der Leistungstests einerseits, und die Ergebnisauswertung andererseits vorgenommen, sodass Kapitel 4 die Auswertung der Messergebnisse enthalten wird. Diese beinhaltet eine detaillierte Beschreibung, Interpretation und Überprüfung der Ergebnisse sowie Schlussfolgerungen bezüglich der Eignung verschiedener Datenbanken und Datenbankkonzepte.

Kapitel 5 wird mit dem Schluss schließlich eine Klammer um die gesamte Arbeit ziehen, die theoretischen und praktischen Erkenntnisse zusammenfassen und erläutern, die Methodik der Arbeit kritisch reflektieren sowie mögliche Weiterentwicklungen der Arbeit aufzeigen.

2 Theoretische Grundlagen

(Christian Brummer, Michael Busam, Thorsten Kühner, Klarissa Schmidt, Matthias Welz)

Dieser Teil der Arbeit schafft die theoretischen Grundlagen für die im Praxisteil konzeptionierten, durchgeführten und ausgewerteten Performancetests. Die nachfolgenden Abschnitte geben eine Einführung zu relationalen sowie Key-Value-Datenbanken, die jeweils das grundlegende Konzept und Funktionsweise, sowie die spezifischen Charakteristika der zum Einsatz kommenden Datenbankprodukte hervorhebt. Abschließend wird ein kurzer Vergleich zwischen relationalen und Key-Value-Datenbanken gezogen, um die insbesondere im Hinblick auf die Performancetests relevanten Unterscheidungsmerkmale zusammenzufassen.

2.1 Relationale Datenbanken

Eine Datenbank (DB) enthält eine formalisierte Anordnung von in Verbindung stehenden Daten, die häufig Objekte der realen Welt darstellen und mit ihnen interagieren.⁵ Es ist eine „gut organisierte Sammlung von Daten, die auf aussagekräftige Weise in Beziehung zueinander stehen und auf die über verschiedene logische Arten zugegriffen werden kann“.⁶ Ein Datenbank Management System (DBMS) ermöglicht Nutzern administrative Funktionen, wie Beschreiben, Anlegen und Manipulieren von Datenstrukturen.⁷ Dabei wird der parallele Zugriff durch mehrere verschiedene Benutzer ausdrücklich unterstützt.⁸ Die vier wesentlichen Hauptaufgaben lassen sich durch Datenverfügbarkeit, Datenintegrität, Datensicherheit und Datenunabhängigkeit zusammenfassen.⁹ Ein DBMS verwendet als Grundlage ein Datenmodell, das im Grunde die Infrastruktur zur Darstellung der Objekte der realen Welt bereitstellt. Es verwaltet dabei zwei Arten von Daten: Zum einen die Daten, die ein Anwender benötigt und zum anderen Metadaten, welche die Struktur und die Form der Erstgenannten näher beschreiben.¹⁰ Im Folgenden werden neben dem grundlegenden Konzept einer Datenbank die im Praxisteil relevanten Produktvertreter vorgestellt.

2.1.1 Grundlegendes Konzept

Dr. E. F. Codd beschreibt in seiner 1970 veröffentlichten Arbeit „A Relational Model of Data for Large Shared Data Banks“¹¹, wie Daten in Form von Tabellen abgelegt werden können und prägt damit den Begriff der Datenbanken. Die Organisation der Daten basiert auf einem mathematischen Modell, dem relationalen Modell, welches Daten in Relationen verwaltet,

⁵ Vgl. Elmasri, R. / Navathe, S. B. (2011), S. 2

⁶ Übersetzt aus Sumathi, S. / Esakkirajan, S. (2007), S. 2

⁷ Vgl. Schneider, U. (2012), S. 441

⁸ Vgl. Sumathi, S. / Esakkirajan, S. (2007), S. 3

⁹ Vgl. Sumathi, S. / Esakkirajan, S. (2007), S. 4

¹⁰ Vgl. Sumathi, S. / Esakkirajan, S. (2007), S. 2

¹¹ Vgl. Codd, E. F. (2002), S. 263 ff.

d.h. Tabellen. Datensätze oder Zeilen werden über Schlüssel (engl. Keys) eindeutig identifiziert und enthalten Eigenschaften oder Attribute, welche die Datensätze beschreiben, wobei Redundanzen durch das Konzept der Normalformen verhindert werden. Weiterhin stellen Indices sicher, dass ein performanter Zugriff auf die Daten anhand bestimmter Attribute möglich ist, ohne, dass sämtliche Datensätze dazu durchsucht werden müssen.¹² Das DBMS gewährleistet, dass alle Datenbankoperationen, sogenannte Transaktionen, dem Atomicity, Consistency, Isolation, Durability (ACID)-Prinzip folgen, welches Atomizität, Konsistenz, Isolation und Dauerhaftigkeit fordert¹³ und dadurch Datenverlust oder Inkonsistenzen verhindert.

Abb. 1 veranschaulicht beispielhaft eine Datenbanktabelle namens „STUDENT“. Diese Tabelle speichert Daten über Studenten in einer formalisierten Struktur, da jede Spalte durch eine Überschrift definiert ist. Diese Spaltendefinitionen dienen als sogenannte Attribute von dem Objekt „STUDENT“. Die Zellen der Spalten enthalten Werte eines bestimmten Datentyps, der entsprechend dem Attribut gewählt wurde - beispielsweise: Zeichenkette, Zahl, Datum und Wahrheitswert oder null, wenn die Zelle leer bleibt.¹⁴ Am Beispiel bleibend repräsentiert jede Zeile der Datenbanktabelle einem Datensatz „STUDENT“: Der Student der ersten Zeile hat die Attribute „Peter“ als „Name“ und kommt aus der „City“ „San Marcos“.

STUDENT			
<i>ID</i>	<i>Name</i>	<i>Phone</i>	<i>City</i>
293021	Peter	234-483920	San Marcos
392013	Carla	324-382930	San Diego
293020	Nick	221-020039	La Jolla

Abb. 1: Veranschaulichung einer Datenbanktabelle

Dies stellt ein sehr stark vereinfachtes Beispiel einer relationalen Datenbank dar, in der alle Daten in Werten ausgedrückt werden und Nutzer über diese Werte immer Zugang zu allen Daten haben. In anderen Worten heißt das, dass alle Daten durch ihre Werte miteinander verknüpft sind.¹⁵ Um einen bestimmten Datensatz aus der „STUDENT“ Tabelle auszuwählen, muss es eine eindeutige Identifizierungsmöglichkeit geben. Im Beispiel der Abb. 1 ist das die Spalte „ID“ mit einem sechsstelligen numerischen Wert, der so gewählt wurde, dass er nur einmalig in der Tabelle vorhanden ist und somit die Zeile eindeutig identifiziert. Die Spalte „ID“ wird deshalb primärer Schlüssel genannt.¹⁶ Durch den „ID“ Wert „293021“ erhält ein Nutzer alle Werte, die in den Spalten (Attributen) „Name“, „Phone“ und „City“ dieses bestimmten Datensatzes der Tabelle „STUDENT“ gespeichert sind.

¹² Vgl. Aho, A. V. / Ullman, J. D. (1994), S. 403 ff.

¹³ Vgl. Sumathi, S. / Esakkirajan, S. (2007), S. 320 ff.

¹⁴ Vgl. Kleuker, S. (2011), S. 62

¹⁵ Vgl. Unterstein, M. / Matthiessen, G. (2012), S. 7

¹⁶ Vgl. ebenda, S. 30

Dieses theoretische Konzept relationaler Datenbanken kann durch die Structured Query Language (SQL) implementiert werden, die als deklarative Programmiersprache zur Kommunikation und Steuerung von DBMSs eingesetzt wird. SQL bietet eine Reihe von Befehlen, um beispielsweise Tabellen anzulegen und diese durch Einfüge- und Aktualisierungsbefehle mit Daten zu befüllen.¹⁷ Ein Datenmodell wird deshalb durch die Data Definition Language (DDL) für die grundlegenden Eigenschaften und Struktur der Datenobjekte und die Data Manipulation Language (DML), um die gespeicherten Datensätze auszugeben, zu bearbeiten und zu löschen oder neue hinzuzufügen, beschrieben. Die Data Control Language (DCL) übernimmt die Rechteverwaltung und Transaktionskontrolle.¹⁸

2.1.2 Ausgewählte Produktvertreter

Im Folgenden werden die im Praxisteil implementierten relationalen Datenbankprodukte, MySQL, PostgreSQL und Derby vorgestellt und ihre Auswahl begründet.

MySQL wurde 1994 vom schwedischen Unternehmen MySQL AB entwickelt und geht namentlich auf die Tochter *My* des Mitbegründers Michael „Monty“ Widenius und SQL zurück.¹⁹ Nachdem MySQL AB 2008 von Sun Microsystems übernommen wurde, wurden diese 2010 von Oracle aufgekauft, sodass MySQL nun von Oracle weiterentwickelt wird.²⁰ Mit rund 50 Millionen Installationen im Jahr 2013 ist es weltweit das populärste relationale DBMS.^{21 22} MySQL stellt die Grundlage vieler Webservices, wie unter anderem Flickr²³, YouTube²⁴, Facebook²⁵, Google²⁶ und Twitter²⁷ dar und wird häufig in Kombination mit einem Apache Webserver und PHP: Hypertext Preprocessor (PHP) als Skriptsprache verwendet. Neben einer Open-Source-Version gibt es auch eine kommerzielle Version, die in der jüngeren Vergangenheit verstärkt für heftige Kritik sorgte, da Oracle wenig Interesse an der freien Entwicklergemeinschaft zeige und neue Funktionen wie auch die Bug-Datenbank der Open-Source-Version vorenthalte.²⁸

¹⁷ Vgl. Beaulieu, A. (2009), S. 7 ff.

¹⁸ Vgl. Kemper, A. / Eickler, A. (2011), S. 23

¹⁹ Vgl. Oracle (o.J.)

²⁰ Vgl. Unterreiner, V. (2009)

²¹ Vgl. Schmitz, L. (2013)

²² Vgl. Oracle (2008)

²³ Vgl. O'Reilly (2007)

²⁴ Vgl. Oracle (o.J. a)

²⁵ Vgl. Sobel, J. (2007)

²⁶ Vgl. Claburn, T. (2007)

²⁷ Vgl. Cole, J. (2011)

²⁸ Vgl. Pearce, R. (2013)

PostgreSQL ist ein objektrelationales DBMS und wird seit 1997 als Open-Source weiterentwickelt.²⁹ Es orientiert sich stark an dem SQL-Standard American National Standards Institute (ANSI)-SQL 2008 und erfüllt somit das ACID-Prinzip, bietet erweiterte Datentypen, Funktionen, Operatoren und Aggregate.³⁰ Neben einer umfangreichen Dokumentation bietet PostgreSQL Schnittstellen für Erweiterungen, um beispielsweise individuelle Such- und Sortierverfahren zu ermöglichen.³¹ PostgreSQL hat in der Vergangenheit eine Vielzahl von Preisen und Auszeichnungen erhalten und gehört somit unbestritten zu den wichtigsten DBMS.³²

Ursprünglich wurde Derby als JBMS von Cloudscape im Jahr 1997 entwickelt und erhielt erst den Namen „Derby“ bei der Übereignung 2004 mit einer Freie-Software-Lizenz von IBM an die Apache Software Foundation. Nachdem Sun Microsystems sich an der Entwicklung beteiligte, wurde Derby 2006 als Java DB im Java Development Kit Version 6 integriert und wird nun von Oracle weiterentwickelt.³³ Apache Derby zeichnet sich insbesondere durch eine sehr kleine Auslieferungsdatei (ca. 2 Megabyte (MB)), der Plattformunabhängigkeit durch Java, der Standardisierung bei der Unterstützung von Java Database Connectivity (JDBC) und ANSI SQL und die Kompatibilität zwischen verschiedenen Betriebssystemen aus.³⁴ Demnach eignet sich Derby vor allem für Java-Projekte.

2.2 Key-Value Datenbanken

Relationale Datenbanken konnten sich über die Jahre fest in die IT-Landschaft von Unternehmen für Aufgaben der Datenverwaltung integrieren und etablieren. Andere Ansätze, wie die der objektorientierten Datenbanken, konnten diese Dominanz nicht beenden.³⁵ In den letzten Jahren traten vermehrt NoSQL Datenbanken in den Vordergrund. Diese versprechen Skalierbarkeit und Flexibilität in Zeiten von Big Data.³⁶ Eine Unterkategorie von NoSQL Datenbanken sind die Key-Value Datenbanken, auf denen der Fokus in diesem Kapitel gerichtet ist.

2.2.1 Entwicklung

Key-Value Datenbanken gehören zu den ältesten Vertretern von NoSQL. Bereits in den 70er Jahren traten die ersten Systeme hervor.³⁷ Als erster Vertreter kann Ken Thompson mit sei-

²⁹ Vgl. PostgreSQL (o.J.)

³⁰ Vgl. PostgreSQL (o.J. a)

³¹ Vgl. PostgreSQL (o.J. b)

³² Vgl. PostgreSQL (o.J. c)

³³ Vgl. Oracle (o.J. b)

³⁴ Vgl. Apache DB (o.J.)

³⁵ Vgl. Fowler, M. / Sadalage, P. (2012), S. 3

³⁶ Vgl. Trivedi, V. (2014), S. 50

³⁷ Vgl. FH Köln (2013)

ner Key/Hash-Datenbank namens DBM angesehen werden.³⁸ Im folgenden Jahrzehnt wurde die Berkeley DB (BDB)³⁹ entwickelt, die 2006 von Oracle gekauft und bis heute angeboten wird.⁴⁰ Auch jüngere Vertreter wie Redis⁴¹ bereichern das Angebot an Key-Value Datenbanken. Die Popularität solcher Systeme stieg erst in den letzten Jahren, als das Thema NoSQL an Zuspruch gewann.

2.2.1.1 Einordnung in NoSQL-Datenbanken

Erst in 2009 wurde der Begriff NoSQL zum ersten Mal in einem Weblog von Eric Evans als Bezeichnung für ein Event in San Francisco⁴², das von einem Team um Johan Oskarsson zum Thema „distributed data storage systems“ organisiert wurde, gebraucht. Die Entwicklung von NoSQL DBMS kann auf den eingangs erwähnten Trend von Big Data zurück geführt werden. Dieser sagt aus, dass immer mehr Informationen (Volume) immer schneller generiert (Velocity) wird und dabei von unterschiedlichen Quellen in verschiedenster Form vorliegen (Variety).⁴³ Relationale DBMS wurden dafür nicht konzipiert und stoßen deswegen an ihre Grenzen.⁴⁴ Dieser Begriff NoSQL hat sich daher als Kontrast zu relationalen DBMS gebildet.⁴⁵ Dabei ist zu beachten, dass NoSQL nicht als Ablehnung zu SQL („no to SQL“) verstanden werden soll. In der Praxis hat sich die Bedeutung des Akronymes auf „Not only SQL“ gelegt.⁴⁶

Dennoch gibt es bisher keine eindeutige Definition.⁴⁷ Am ehesten kann NoSQL als ein Sammelbegriff für DBMS, die nicht (vollständig) den Konventionen relationaler Datenbanken folgen, beschrieben werden.⁴⁸ Edlich et al.⁴⁹ nennen eine Liste von Eigenschaften, mithilfe derer DBMS zu NoSQL zugeordnet werden können, wobei nicht alle Punkte erfüllt sein müssen.

1. Kein relationales Datenmodell
2. Ausrichtung auf eine verteilte und horizontale Skalierbarkeit von Anfang an
3. Das System ist Open Source

³⁸ Vgl. Edlich, S. / u.a. (2010), S. 1

³⁹ Vgl. Oracle (o.J. c)

⁴⁰ Vgl. Celko, J. (2013), S. 83

⁴¹ Vgl. Redis (o.J. b)

⁴² Vgl. Meetup (o.J.)

⁴³ Vgl. Trivedi, V. (2014), S. 50 f.

⁴⁴ Vgl. Klein, D. / Tran-Gia, P. / Hartmann, M. (2013), S. 321f.

⁴⁵ Vgl. Edlich, S. / u.a. (2010), S. 1

⁴⁶ Vgl. Walker-Morgen, D. (2010)

⁴⁷ Vgl. Fowler, M. / Sadalage, P. (2012), S. 10

⁴⁸ Vgl. Janssen, C. (w.y.)

⁴⁹ Edlich, S. / u.a. (2010)

4. Schemafreiheit oder schwächere Schemarestriktionen
5. Unterstützung einer einfachen Datenreplikation aufgrund der verteilten Architektur
6. Das System bietet ein einfaches Application Programming Interface (API)
7. Konsistenzmodelle wie Eventually Consistent und Base (nicht ACID!) als mögliche Grundlagen

Fowler und Sadalage⁵⁰ beschreiben ähnliche Eigenschaften zur Abgrenzung von NoSQL zu relationalen DBMS. Damit wird ein erstes, allgemeines Verständnis des Begriffes NoSQL sichtbar.

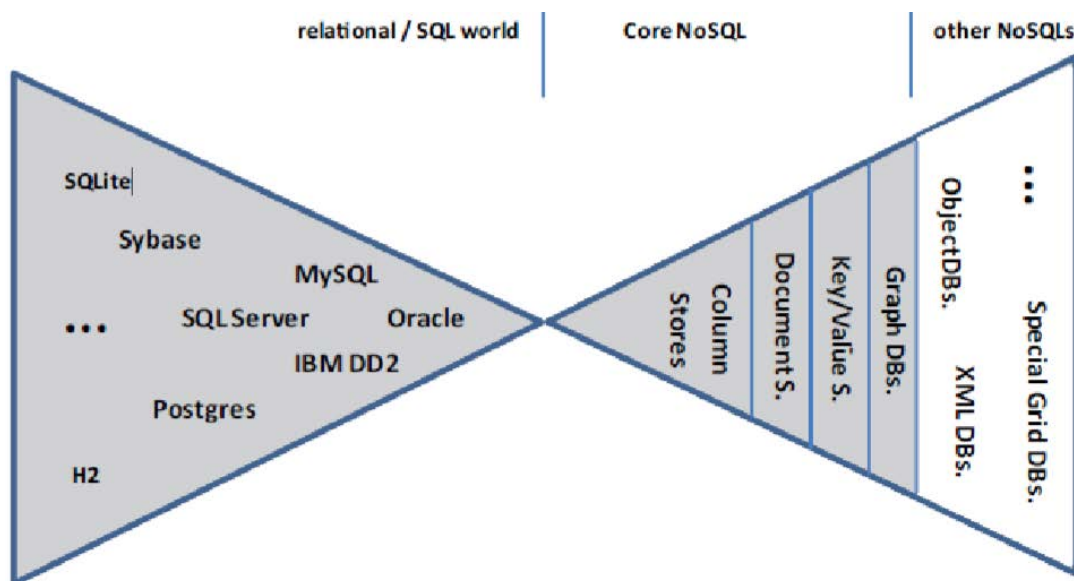


Abb. 2: Kategorisierung von NoSQL DBMS⁵¹

Die Liste von Eigenschaften erlaubt viel Spielraum bei der Konzeption von NoSQL DBMS. Mit der Zeit bildeten sich verschiedene Modelle, die in der Literatur in vier Kategorien eingeteilt werden. Abb. 2 zeigt die Einteilung nach Edlich et. al. Diese nennen folgende Hauptkategorien („Core NoSQL“):

- Graphdatenbanken
- Key-Value Datenbanken
- Dokumentenorientierte Datenbanken
- Spaltenorientierte Datenbanken

Anderen Werke⁵² teilen diese Kategorisierung, wobei keine Unterscheidung in „Core NoSQL“ und „Other NoSQL“ getroffen wird. Letztere Kategorie ist ein Sammelbegriff für NoSQL Datenbanken, die noch keine größere Produktvariation aufweisen.

⁵⁰ Fowler, M. / Sadalage, P. (2012)

⁵¹ Enthalten in Edlich, S. / u.a. (2010), S. 6

⁵² Celko, J. (2013), Tiwari, S. (2011), Fowler, M / Sadalage, P. (2012), Klein, D. / Tran-Gia, P. / Hartmann, M. (2013)

Key-Value Datenbanken werden als einfachste NoSQL Ausprägung angesehen.⁵³ Dies ist auf deren simples zugrundeliegendes Modell zurückzuführen. Mehr hierzu ist in Kapitel 2.2.2 beschrieben.

2.2.1.2 Anwendungsgebiete

Anwendung finden NoSQL Datenbanken in Bereichen, in denen relationale Datenbanken an ihre Grenzen stoßen. Dies ist vor allem im Bereich des sogenannten Web 2.0 zutreffend. Dort ist das Schlagwort Big Data allgegenwärtig und auf die wachsende Anzahl an (mobilen) Endgeräten und damit Benutzer im Internet⁵⁴ zurückzuführen. Die erhöhte Nutzerzahl in Kombination mit der stetig wachsenden Anzahl an dort verfügbaren Diensten sorgen für ein immenses Datenaufkommen, das beispielsweise Nutzdaten, Profildaten und statistischen Daten einschließt.⁵⁵ Um diese Daten verwalten, speichern und verarbeiten zu können, wird der Anspruch der Skalierbarkeit und hohen Verfügbarkeit an die jeweiligen Datenbanken gestellt. Diese können von NoSQL DBMS erfüllt werden, da in deren Begriffsdefinition (siehe Kapitel 2.2.1.1) diese Ansprüche angesprochen sind. Relationale DBMS sind dagegen mit einer vertikalen Skalierbarkeit konzipiert, was der Zentralisierung der Daten auf einen Knotenpunkt gleichgestellt ist. Damit ist es auch möglich, große Datenmengen zu verwalten, bringt aber einen erhöhten Aufwand und damit Kosten, eine gesteigerte Anfälligkeit an Fehler sowie Limitierungen bei der Kapazität mit sich.⁵⁶ Damit eignen sich NoSQL DBMS besonders für das Verwalten und Verarbeiten von großen Datenmengen in verteilten Systemen.⁵⁷

NoSQL DBMS kommen bei verschiedenen Unternehmen, die im Web 2.0 tätig sind, zum Einsatz. Das Apache Projekt Cassandra ist eine der bekanntesten NoSQL DBMS, das sich bei der Kategorie der spaltenorientierten Datenbanken einordnet. Es kommt unter anderem bei Twitter⁵⁸, eBay, GitHub, Instagram, Netflix und Reddit zum Einsatz.⁵⁹ All diese Einsatzgebiete haben die Gemeinsamkeit, dass sehr viele Zugriffe (Lese- und Schreibtransaktionen) in sehr kurzer Zeit auf den Datenbestand vollzogen werden und das Datenvolumen stetig mit hoher Geschwindigkeit steigt. Dies stimmt mit dem eingangs erwähnten Anspruch an Datenbanken in diesem Umfeld über ein.

⁵³ Vgl. MongoDB (2014)

⁵⁴ Vgl. Klein, D. / Tran-Gia, P. / Hartmann, M. (2013), S. 1

⁵⁵ Vgl. ebenda, S. 1

⁵⁶ Vgl. MongoDB (2014)

⁵⁷ Vgl. Trivedi, T. (2014), S. 44 und Fowler, M. / Sadalage, P. (2013), S. 12

⁵⁸ Vgl. Lai, E. (2010)

⁵⁹ Vgl. Cassandra (o.J.)

Zudem bieten bekannte Vertreter wie Amazon und Google eigene NoSQL Lösungen an. Amazon DynamoDB⁶⁰ ist eine Open Source Key-Value Datenbank. Google BigTable hingegen ist eine proprietäre NoSQL Datenbank, die in verschiedenen Google Diensten zum Einsatz kommt.⁶¹

Ein weiteres Anwendungsgebiet wird in der agilen Softwareentwicklung gesehen. Durch die Agilität bei der Entwicklung, werden Funktionen dem System in Iterationen eingeführt. Dies könnte Veränderungen an den zugrundeliegenden Datenbanken einschließen. Relationale DBMS verlangen, dass das Datenmodell bereits von Anfang an vordefiniert ist, was im Widerspruch mit der oben beschriebenen Agilität steht. Die Schemafreiheit von NoSQL Datenbanken erlaubt das Einfügen von Daten ohne ein vordefiniertes Schema. Damit wird die Notwendigkeit für Serverwartungen für Datenmigrationen wegen einer Anpassung des Datenmodells eliminiert, was Entwicklungen erleichtert.⁶²

2.2.2 Grundlegendes Konzept

Key-Value Datenbanken stellen die einfachste Form von NoSQL Datenbanken dar. Grundsätzlich können diese in zwei weitere Untervarianten aufgeteilt werden. Dazu gehören die In-Memory Varianten, die Daten im Hauptspeicher (Arbeitsspeicher) halten, und die On-Disk Varianten, die permanenten Speicher benutzen.⁶³

Die Funktionsweise von Key-Value Datenbanken ist bereits im Namen angedeutet. Key-Value Datenbanken funktionieren nach dem Schlüssel- und Wertschemas⁶⁴, bei dem eine Sammlung von Schlüsseln mit zugeordneten Werten abgespeichert werden. Die Schlüssel müssen hierbei einzigartig sein und können jeglichen Datentyp annehmen.⁶⁵ Dabei können diese aus strukturierten oder willkürlichen Zeichenketten bestehen. Die Wahl des Schlüssels ist bei Key-Value DBMS eine wichtige Aufgabe, da über diese jegliche Abfragen laufen. Die dazu gehörigen Werte müssen immer einen Schlüssel zugeordnet sein und können dabei jegliche Form annehmen, wie z.B. Strings, Listen, Sets oder Hashes.⁶⁶ Die Werte sind dabei frei von jeglichen Standardwerten (defaults) und Einschränkungen (constraints).

Damit wird die Zielsetzung von Key-Value Datenbanken deutlich: Diese sind für das Abfragen von großen Datenmengen bzw. das Hinzufügen zum Datenbestand konzipiert, wobei die

⁶⁰ Vgl. Amazon (o.J.)

⁶¹ Vgl. Chang, F. / u.a. (2006), S. 1

⁶² Vgl. MongoDB (2014) und Fowler, M. / Sadalage, P. (2012), S. 12

⁶³ Vgl. Klein, D. / Tran-Gia, P. / Hartmann, M. (2013), S. 321 f.

⁶⁴ Vgl. FH Köln (2013)

⁶⁵ Vgl. Celko, J. (2013), S. 84

⁶⁶ Vgl. FH Köln (2013)

Beziehung zwischen den Elementen aus Sicht der Datenbank von keiner Bedeutung ist.⁶⁷ Dies wird auch in den vier grundlegenden Transaktionsarten wiedergespiegelt. Es wird unterschieden zwischen dem Einfügen eines Schlüssel-Wert Paares, dem Löschen eines solchen, dem Aktualisieren von existierenden Paaren und dem Suchen eines Werten über den identifizierenden Schlüssel .⁶⁸

Dieser geringe Funktionsumfang schränkt das Anwendungsgebiet von Key-Value Datenbanken stark ein. Diese sind daher für das Speichern und Abfragen von Daten, die immer in derselben Form vorliegen, geeignet. Dabei ist der Datensatz zu einer spezifizierten Schlüssel zugeordnet, ohne dabei eine Verbindung zu anderen Daten aufzuweisen. Als Beispiel ist die Verwaltung von Sessions in einem Websystem zu nennen, bei dem der Cookie Wert als Schlüssel, der aktuelle Zustand des Websession als Wert hinterlegt wird.⁶⁹

Diese Einfachheit schränkt zwar das Anwendungsgebiet ein, bringt aber in den Feldern, in denen Key-Value Datenbanken genutzt werden können, große Vorteile. Es bietet durch das einfache Schema eine schnelle und effiziente Datenverwaltung sowie die nötige Skalierbarkeit für die Anforderungen von Big Data.⁷⁰

2.2.3 Ausgewählte Produktvertreter

Die Implementierung im Praxisteil stützt sich auf die Key-Value Datenbankvertreter Redis und die dokumentenorientierte Datenbank MongoDB. Dieses Kapitel erklärt die Datenmodelle dieser Vertreter und begründet deren Auswahl für die Implementierung.

2.2.3.1 Redis

Redis ist eine Open-Source Key-Value Datenbank, die sowohl In-Memory als auch On-Disk betrieben werden kann. Es wird häufig auch als Datenstrukturserver angesehen, da es verschiedenste Datenstrukturen für die zu speichernden Daten anbietet.⁷¹ Redis zählt zu den beliebtesten Key-Value Datenbanken.⁷² Als Grund für die Popularität kann der hoch entwickelte Befehlssatz mit mehr als 120 Befehlen⁷³ und die hohe Performanz bei Lese- und Schreiboperationen⁷⁴ genannt werden. Zudem bietet Redis eine umfangreiche Client-

⁶⁷ Vgl. Celko, J. (2013), S. 84

⁶⁸ Vgl. ebenda, S. 81 f.

⁶⁹ Vgl. Jansen, R. (2011)

⁷⁰ Vgl. FH Köln

⁷¹ Vgl. Tiwari, S. (2011), S. 15 und Redmond, E. / Wilson, J. (2012), S. 285

⁷² Vgl. DB-Engines (o.J.)

⁷³ Vgl. Redmond, E. / Wilson, J. (2012), S. 334

⁷⁴ Vgl. Redis (o.J. c)

Bibliotheken an, die alle gängigen Sprachen umfasst.⁷⁵ Redmond und Wilson charakterisieren Redis als „[...] eine perfekte Ergänzung jedes polyglotten Ökosystems.“⁷⁶ Sie nennen als Anwendungsgebiet Caching, die Verwaltung von Nachrichten oder als Helfer für Datentransformationen.

Das Datenmodell von Redis basiert auf dem Prinzip eines Schlüssel-Wert-Paares, womit die Zuordnung zu den Key-Value Datenbanken begründet wird. Die Funktionsweise ist daher analog mit der in Kapitel 2.2.2 beschriebenen. Der Schlüssel wird in Redis als Zeichenkette gespeichert. Reine Key-Value Datenbanken können nur Suchanfragen über den Schlüssel handhaben.⁷⁷ Damit wird die Wichtigkeit einer überlegten Definition der Schlüsselwerte bei der Datenmodellierung unterstrichen.

Die dem Schlüssel zugeordneten Werte enthalten die eigentlichen Daten. Redis unterstützt dabei mehrere Datenstrukturen, von einfachen Strings bis hin zu Bitmaps, wobei Redis für jede Struktur einen eigenen Befehlssatz liefert.⁷⁸ Abb. 3 gibt einen Überblick über die drei gängigsten Datenstrukturen in Redis und veranschaulicht deren Funktionsweise.

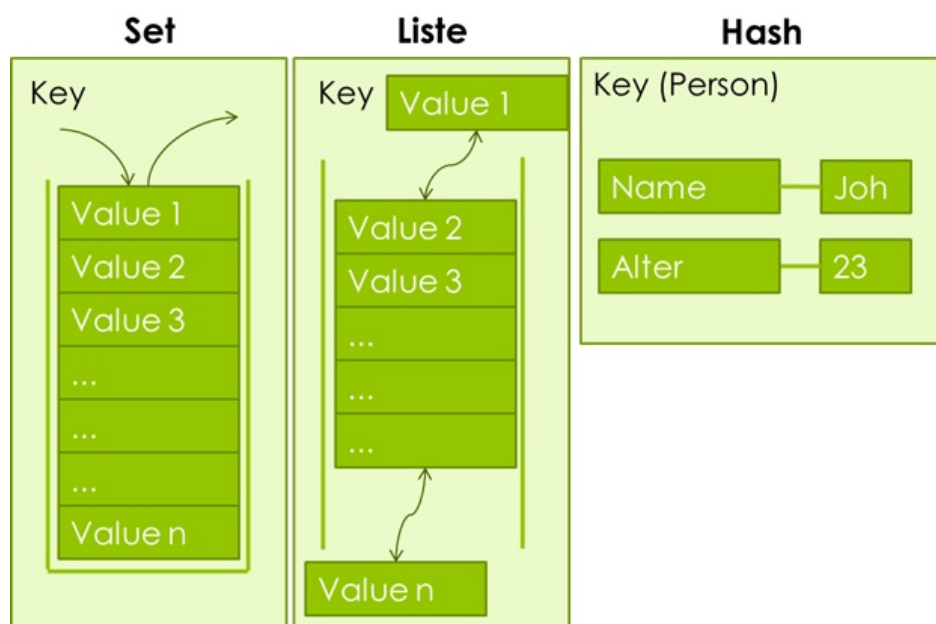


Abb. 3: Value Typen in Redis⁷⁹

Eine Möglichkeit der Datenstrukturierung von Werten in Redis sind Hashes. Es handelt sich dabei um „[...] verschachtelte Redis-Objekte, die eine beliebige Anzahl von Schlüssel/Wert-

⁷⁵ Vgl. Tiwari, S. (2011), S. 15 und Redmond, E. / Wilson, J. (2012), S. 286

⁷⁶ Redmond, E. / Wilson, J. (2012), S. 333

⁷⁷ Vgl. Sequin, K. (2012)

⁷⁸ Vgl. Redis (o.J.)

⁷⁹ Enthalten in: FH Köln (2013)

Paaren enthalten können.⁸⁰ Damit ist es möglich, Daten ohne künstliche Schlüssel-Präfixe zu speichern und Felder innerhalb eines Hashes zu definieren. Hashes eignen sich somit zur Darstellung von Objekten.⁸¹

Listen enthalten mehrere Werte unter einem Schlüssel. Dabei können Elemente entweder von oben auf oder von unten an die bestehende Liste angefügt werden. Hierbei benutzt Redis eine Indizierung auf einer Nullbasis, womit das erste Element mit 0, das zweite mit 1 und so weiter beziffert sind.⁸² Damit funktionieren Listen wie Queues oder Stapel⁸³ und agieren nach den gleichen Prinzipien, wie beispielsweise dem First In First Out (FIFO).

Sets teilen mit Listen eine ähnliche Funktionsweise und beschreiben eine Sammlung von verschiedenen Werten unter einem Schlüssel. Die Besonderheit ist hierbei, dass jedes Element in einem Set einzigartig sein muss. Sets gibt es in zwei Ausprägungen. Unsortierte Sets folgen der Funktionsweise eines Stapel und damit dem Prinzip des Last In First Out (LIFO). Der von Redis angebotene Befehlssatz enthält beispielsweise Operationen, die das Testen auf existierende Elemente in einer Liste oder das Ausführen einer Suche nach gleichen Werten in verschiedenen Sets umfasst. Sortierte Sets charakterisieren die zweite Ausprägung von Sets. Bei diesen werden Gleitkommawerte (Elementwert genannt) jedem Element zugeordnet, anhand deren die Sortierung erfolgt. Redis ermöglicht es, Befehle wie eine lexikografische Abfrage bei solchen Sets anzuwenden.⁸⁴

2.2.3.2 MongoDB

MongoDB gehört zu den dokumentenorientierten Datenbanken, die, wie der Name suggeriert, Daten in Form von Dokumenten abspeichert. Der Name MongoDB, oder kurz Mongo, leitet sich vom englischen Wort *humongous* (riesig) ab, was auf die Fähigkeiten der Datenbank im Umgang mit großen Datenmengen anspielen soll.⁸⁵ Dabei nutzt MongoDB eine horizontale Aufteilung (kurz *Sharding*), um eine sichere und schnelle Verarbeitung zu gewährleisten.⁸⁶ Als weitere Stärken werden die Leistungsfähigkeit, die einfache Nutzung, das große Aufgabenspektrum und die Flexibilität genannt.⁸⁷ Letztere begründet die Aussage von Redmond und Wilson, die MongoDB beschreiben als „[...] eine wesentlich natürlichere Antwort auf viele gängige Problembereiche anwendungsgesteuerter Datenmengen als relationa-

⁸⁰ Redmond, E. / Wilson, J. (2012), S. 290

⁸¹ Vgl. Redis (o.J.)

⁸² Vgl. ebenda

⁸³ Vgl. FH Köln (2013) und Redmond, E. / Wilson, J. (2012), S. 291

⁸⁴ Vgl. Redis (o.J.)

⁸⁵ Vgl. Hawkins, T. / Membrey, P. / Plugge, E. (2010), S. 1

⁸⁶ Vgl. Redmond, E. / Wilson, J. (2012), S. 184

⁸⁷ Vgl. ebenda, S. 147

le Datenbanken.“⁸⁸ Auch Unternehmen haben bereits MongoDB für sich entdeckt. So nutzen Foursquare, bitly und auch das Forschungszentrum CERN diese Datenbank.⁸⁹

Die Kategorie der Dokumentenorientierten Datenbanken unterscheidet sich durch einen größeren Funktionsumfang von Key-Value Datenbanken. Dabei enthält MongoDB im wesentlichen die Funktionalitäten einer Key-Value Datenbank und erweitert diese durch beispielsweise der Möglichkeit, das zugrundeliegende Datenmodell mittels eines relationalen Ansatzes zu gestalten, wobei ein Schema wiederum nicht zwangsläufig nötig ist. Eine Ausprägung davon ist, dass Daten verschachtelt festgehalten werden können. In anderen Worten, eine Zeile einer relationalen Tabelle kann ohne Schema mit beliebig tiefer Verschachterlung aufgenommen werden. MongoDB nutzt dafür ein Format, das in starker Anlehnung an JavaScript Object Notation (JSON) konzipiert wurde, nämlich Binary-JSON (BSON)⁹⁰, das leichter zu verarbeiten ist, einen größeren Funktionsumfang anbieten und das Suchen in einem Dokument erleichtert.⁹¹ JSON ist im Allgemeinen ein sehr expressives Format für den Datenaustausch, das in der Regel bei Webapplikationen Anwendung findet. Es beschreibt den Inhalt eines Dokumentes in einer für den Menschen leicht lesbaren Weise (vgl. Abb. 4 und Abb. 5) und ist dabei schemafrei.⁹² Dies ermöglicht es, Dokumente einzeln abzuändern, ohne dabei andere Dokumente zu beeinflussen. Das Datenmodell von MongoDB ist um sogenannte *Collections* zentriert, welche die Daten in Form von Dokumenten halten. Collections sind analog zu Tabellen in relationalen Modellen zu verstehen. Der Unterschied besteht darin, dass die Anwendung eines Schemas nicht erzwungen wird, wodurch jedes Dokument innerhalb einer Collection verschiedene Felder aufweisen kann.⁹³ In der Modellierung der Datenstruktur kennt MongoDB zwei verschiedene Ansätze.

Der Erste ist auf das Verwenden von Referenzen ausgelegt. Dabei werden die Beziehungen zwischen den Daten separat in Form von Links oder Referenzen von einem Dokument zu einem anderen abgespeichert.⁹⁴ Dies stellt den relationalen Ansatz bei MongoDB dar. Abb. 4 zeigt beispielhaft ein solches Schema. Hierbei sind drei Dokumente unter einer Collection gesammelt, wobei ein Dokument (*user document*) nur für das Speichern der Beziehung zwischen den anderen beiden Dokumenten zuständig ist. Solch ein normalisiertes Datenmodell

⁸⁸ Ebenda, S. 191

⁸⁹ Vgl. ebenda, S. 147

⁹⁰ Vgl. Redmond, E. / Wilson, J. (2012), S. 147 ff.

⁹¹ Vgl. Hawkins, T. / Membrey, P. / Plugge, E. (2010), S. 5 ff.

⁹² Vgl. Hawkins, T. / Membrey, P. / Plugge, E. (2010), S. 5

⁹³ Vgl. MongoDB (2014 b)

⁹⁴ Vgl. MongoDB (2014 c)

wird häufig als Baumstruktur dargestellt, welche die verschiedenen Beziehungen innerhalb einer Collection veranschaulicht.⁹⁵

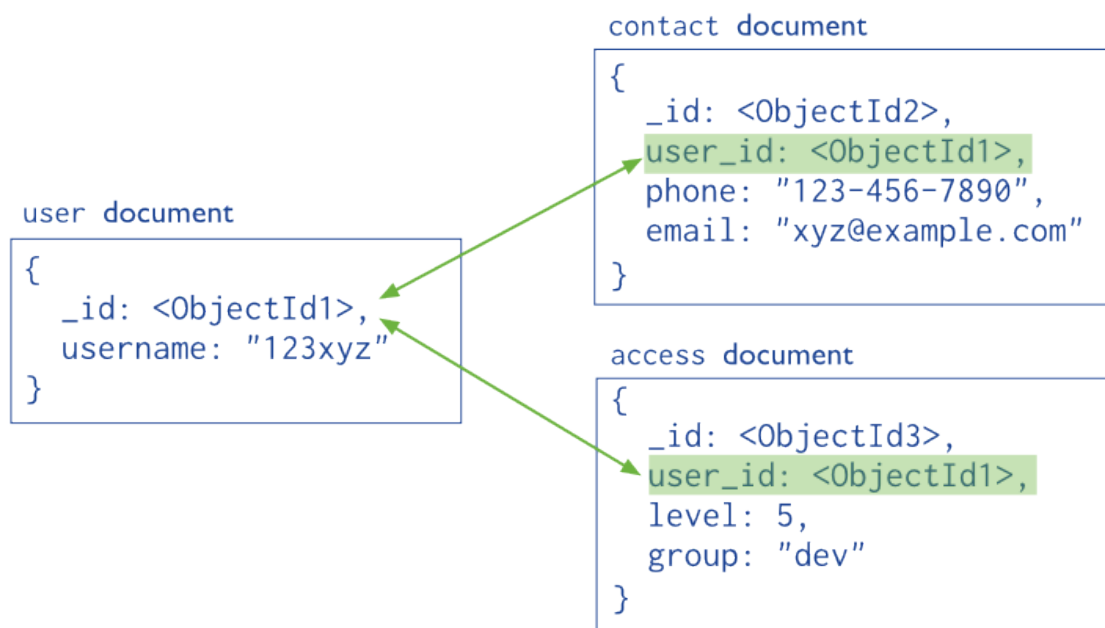


Abb. 4: Normalisiertes Datenmodell in MongoDB⁹⁶

Ein solches normalisiertes Datenmodell kann Probleme mit Schreiboperationen verursachen, da das Handhaben der Atomizität bei Transaktionen in Umgebungen mit mehreren Benutzern zu gewährleisten ist.

Der zweite Ansatz zur Datenmodellierung hat dieses Problem nicht. Bei diesem gilt, dass die Daten in sich geschlossen sind. Das bedeutet, dass alle relevanten Daten zu einem Objekt in einem Dokument gespeichert werden. Somit können in einem Dokument mehrere Ebenen an Daten geschachtelt werden. Abb. 5 veranschaulicht ein solches Dokument in einem denormalisierten Datenmodell. Hierbei ist das Dokument verschachtelt aufgebaut und enthält zwei weitere Unterdokumente für die Kontaktdaten und die Zugangsberechtigung. Solche Unterdokumente haben die selben Eigenschaften wie normale Dokumente und können daher weiter verschachtelt werden. Dies in Kombination mit der Verwendung von BSON sind die beiden wesentlichen Unterschiede zu Redis.

Das Speichern von Daten in solch einem denormalisierten Modell macht diesen Ansatz ideal für verteilte Systeme, da jeder Server nur seinen Datenbestand abfragen muss und nicht auf Daten von anderen Servern angewiesen ist (Prinzip des *Sharding*). Dies führt zu einer beinahe linearen Skalierung bei Lese- und Schreiboperationen.⁹⁷ Ein Problem dieses Ansatzes

⁹⁵ Vgl. MongoDB (2014 d)

⁹⁶ Enthalten in MongoDB (2014 c)

⁹⁷ Vgl. Hawkins, T. / Membrey, P. / Plugge, E. (2010), S. 7

ist, dass Dokumente unaufhörlich wachsen können. MongoDB speichert zu große Dokumente im Massenspeicher. Dabei kann es passieren, dass nur Teile on-Disk geschrieben werden, während der Andere noch im Arbeitsspeicher gehalten wird. Diese Datenfragmentierung verschlechtert die Schreibleistung. In solchen Fällen ist ein normalisiertes Datenmodell zu betrachten.⁹⁸



Abb. 5: Denormalisiertes Datenmodell in MongoDB⁹⁹

2.3 Zusammenfassender Vergleich

Im Folgenden werden die bereits in den vorigen Kapiteln angesprochenen Eigenschaften von relationalen und Key-Value Datenbanken gegenübergestellt und zusammenfassend verglichen. Als wesentliche Unterscheidungsmerkmale dienen dabei die historische Entwicklung, Flexibilität, Skalierbarkeit und Konsistenz der beiden Datenbankenarten:

Durch die jahrzehntelange Entwicklung von relationalen Datenbanken sind eine Reihe von verlässlichen Systemen auf dem Markt, die sich eines breiten Anwenderkreises erfreuen. Durch die erst in der jüngeren Vergangenheit wachsende Popularität von NoSQL Datenbanken mangelt es hier noch an etablierten Standards, sodass Anwender schnell abhängig von einzelnen Anbietern werden und eventuell langfristig Probleme mit noch nicht vollständig ausgereiften Systemen bekommen können.

Relationale Datenbanken trennen in ihrem Datenmodell die Metadaten von den eigentlichen Daten und erfordern somit ein Schema. Dieses vordefinierte Datenschema liefert zwar eine grundlegende Struktur der Relationen, erfordert allerdings einen höheren Aufwand für nach-

⁹⁸ Vgl. MongoDB (2014 e)

⁹⁹ Enthalten in MongoDB (2014 c)

trägliche Änderungen. Key-Value Datenbanken setzen hier auf einen schemafreien Ansatz, der ein erhöhtes Maß an Flexibilität während der Entwicklung ermöglicht. Spätere Anpassungen am Datenmodell sind ohne großen Aufwand möglich und genügen den Anforderungen agiler Vorgehen, was sie unter anderem zugänglicher macht, da Prinzipien wie Normalform nicht berücksichtigt werden müssen.

Im Hinblick auf den jüngsten anhaltenden Trend von Big Data müssen zukünftige Systeme den Anforderungen immer schneller steigender Datenmengen aus unterschiedlichsten Quellen gewachsen sein. Relationale Datenbanksysteme werden dem durch eine vertikale Skalierbarkeit nur bedingt und unter hohem (Kosten-)aufwand gerecht. Key-Value Datenbanken hingegen setzen auf eine horizontale Skalierbarkeit, sodass durch eine Verteilung auf mehrere Knoten eine hohe Verfügbarkeit gewährleistet ist.

Um Datenverlust oder Inkonsistenzen bei Datenbankoperationen zu verhindern, folgen relationale Datenbanken dem ACID-Prinzip, welches Atomizität, Konsistenz, Isolation und Dauerhaftigkeit fordert. Key-Value Datenbanken setzen hierbei auf Konsistenzmodelle wie Eventually Consistent und Base, die sich zunehmend an das Qualitätsniveau von relationalen Datenbanksystemen annähern, aber für manche Einsatzgebiete, in denen die Konsistenz der Daten von entscheidender Bedeutung ist, ungeeignet sind.

Zusammenfassend ist festzuhalten, dass sowohl relationale als auch NoSQL-Datenbanken ihre Berechtigung haben und je nach Projektsituation die jeweiligen Eigenschaften analysiert und bewertet werden müssen. Im weiteren Verlauf dieser Arbeit wird insbesondere die Performance als differenzierendes Merkmal der beiden Systeme untersucht.

3 Konzeption und Umsetzung von Performancetests

(Christian Brummer, Michael Busam, Thorsten Kühner, Klarissa Schmidt, Matthias Welz)

Nachdem im vorigen Abschnitt die grundlegenden theoretischen Aspekte von relationalen und Key-Value Datenbanken dargelegt wurden, beschäftigt sich dieser Abschnitt mit der Performancemessung der beiden Systeme. Zuerst wird der Anwendungsfall, gefolgt von den konzeptionierten Testszenarien, vorgestellt. Anschließend wird die technische Implementierung erläutert und letztlich die Durchführung der Messungen beschrieben.

3.1 Anwendungsfall

Das für die Tests der relationalen und Key-Value Datenbanken angedachte Geschäftsmodell handelt von einem Paketdienstleister. Dieses Modell wurde auf Grund seiner Vielseitigkeit an Testszenarien und seinem notwendig hohen Datenbestand, welcher für umfangreiche Test benötigt wird, ausgewählt.

Für die Wirtschaftlichkeit eines Paketdienstleisters sorgt vor allem eine starke Vernetzung im zu beliefernden Gebiet. Dies bedeutet, dass ein dichtes Standortnetz gewährleistet sein muss um einen möglichst effizienten und effektiven Paketdienst anbieten zu können. Dieses Netz an Lokationen sollte sowohl national, als auch international ausgebreitet sein, um die Anforderungen der Kunden bestmöglich abzudecken. Zudem würde ein separates Express-Netz die schnelle Lieferung mancher Pakete garantieren können. Durch standardisierte und einfache Prozesse solle eine pünktliche Lieferung und ein geringer Schadensausfall garantiert werden. Die Prozessgegebenheit hängt von den Prozesskosten, der Prozessqualität und der Prozesskomplexität ab. Diese drei Variablen haben den größten Einfluss auf die Prozess Gegebenheit.

Das für die Tests angewandte Geschäftsmodell beinhaltet drei verschiedene Hauptbestandteile, diese wären die Lokation, die Pakete selbst und die Historie der Pakete. Jeder Lokation wird eine Lokationsnummer zugewiesen. Jede Lokationsnummer ist daher sowohl mit Namen der Lokation also auch deren Adresse verbunden. Die Pakete erhalten, sobald sie von einem Kunden abgegeben werden, eine Paketversandnummer, welche einen Startpunkt und ein Zielpunkt zugeordnet wird. Zudem wird noch eine Historie angelegt, welche einen Überblick über alle Pakete und deren durchlaufenen Lokationen und Zeitangaben liefert. Jedem Historieneintrag ist ein bestimmtes Paket, eine vom Paket durchlaufene Lokation und das dazugehörige Datum der dortigen Paketankunft zugeteilt.

Folgendes Beispiel soll den Prozess welcher von jedem Paket durchlaufen wird im Zusammenhang mit den drei soeben beschriebenen Variablen ausführlicher erklären: Ein Kunde gibt bei einer Lokation sein Paket bei dem Paketdienstleister ab. Hierbei wird dem Paket eine

durchlaufende Nummer zugeteilt. Dieser Paketnummer wird desweiteren die Startlokation, also die der Abgabe, und eine vom Kunden gewünschte Zieladresse zugeordnet. Daraufhin erscheint dieses Paket im System. Das Paket durchläuft nun mehrere Lokationen, welche wie zuvor beschrieben einer Nummer und Adresse zugeordnet sind, bis es seine Zieladresse erreicht. In dem Zeitraum der Lokationswechsel wird jeweils im System ein neuer Eintrag erstellt, welcher wie oben beschrieben die Paketnummer, die Lokation und das Datum beinhaltet. Dies ermöglicht eine Überwachung der einzelnen Pakete und eine mögliche Zurückverfolgung dieser. Sobald das Paket an seiner Zieladresse angekommen ist, kann der gesamte Weg, welchen das Paket eschritten hat, anhand dieser Historie nachvollzogen werden. Diesen Prozess durchlaufen alle Pakete, somit erweitert sich das Modell um jedes Paket, das neu dazukommt, damit steigt die Durchlaufnummer der Pakete stark an und auch die Historie wird ständig erweitert. Die Angaben zu den einzelnen Lokationen bleiben jedoch bestehen, da sich diese in unserem Geschäftsmodell nicht verändern. Das bedeutet, es wird keine geschlossen und auch keine neu eröffnet.

Mögliche weitere Szenarien, welche aufbauend auf das soeben beschriebene Geschäftsmodell entstehen, werden im folgenden Kapitel ausführlich beschrieben.

3.2 Konzeption von Testszenarien

In diesem Kapitel werden die Testszenarien, welche für die Überprüfung von Unterschieden zwischen relationalen und Key Value Datenbanken herangezogen werden, beschrieben. In der Beschreibung der Testszenarien wird Fokus auf den wirtschaftlichen Hintergrund der Szenarien gelegt. Fragen wie beispielsweise: „Welchen Nutzen hat dieses Szenario?“, „Wer greift auf diese Informationen zu?“ und „Welche Variablen werden in dieser Abfrage benötigt?“ werden in der Beschreibung beantwortet.

Die Szenario Beschreibungen sind jeweils in demselben tabellarischen Schema aufgebaut. Zuerst wird ein Überblick über das Szenario und dessen Inhalte geliefert. Darauf folgt eine Auflistung des Inputs, welcher benötigt wird für die Durchführung und des Outputs, welcher als Ergebnis geliefert wird. Zusätzlich wird die Motivation beschrieben, welche die Szenario Abfrage auslöst. Zuletzt wird der Klassenname des Szenarios aufgeführt, welcher in Kapitel 3.3.3 beschrieben wird.

3.2.1 Alle Lokationen auslesen

Überblick	Für jede Lokation werden die Informationen ausgelesen. Diese beinhalten die Daten aller Lokationen.
Input	-
Output	Eine Liste der Lokationen, welche derzeit vom Paketdienstleister genutzt werden, mit der Lokationsadresse, -nummer und -name.
Motivation	Dadurch kann überprüft werden, welche Lokationen des Paketdienstleisters derzeit genutzt werden.
Klassenname	GetAllLocationsTest

3.2.2 Füllstand aller Lokationen auslesen

Überblick	Liefert die Anzahl der Pakete, die sich zum aktuellen Zeitpunkt an einer angegebenen Lokation befinden.
Input	-
Output	Anzahl der Pakete im Zeitpunkt der Abfrage.
Motivation	Somit kann ein Mitarbeiter des Paketdienstleisters überprüfen wie ausgelastet eine Lokation im Zeitpunkt der Abfrage ist.
Klassenname	GetAllLocationPackageCount

3.2.3 Frequentierung der Lokation auslesen

Überblick	Ermittelt alle Pakete, die sich zu irgendeinem Zeitpunkt in ihrer Historie an der angegebenen Lokation befunden haben.
Input	Die Paket- und Lokationsnummer zur eindeutigen Identifikation.
Output	Eine Ausgabe der Pakete, welche sich in irgendeinem Zeitpunkt in einer Lokation befunden haben, mit der Paketnummer.
Motivation	Somit kann überprüft werden welche Lokation in der Historie am meisten frequentiert wurde.
Klassenname	GetAllLocationTotalPackageCount

3.2.4 Gesamte Verweildauer pro Paket in der Logistikkette auslesen

Überblick	Ermittlung der Anzahl an Tagen zwischen der ersten und der letzten Lokation in der Historie eines Paketes.
Input	Die Paketnummer zur eindeutigen Identifikation des Pakets in dessen Historie.
Output	Differenz zwischen dem Datum des ersten Historieneintrags und dem Datum des letzten Historieneintrags eines bestimmten Paketes.
Motivation	Somit kann überprüft werden ob die, den Kunden versprochene, Auslieferungszeit der Pakete eingehalten wurde.
Klassenname	GetRandomPackageTrackingTime

3.2.5 Historie aller Pakete auslesen

Überblick	Für jedes Paket wird die Historie ausgelesen. Diese beinhaltet die Daten jeder Lokation, die das Paket durchlaufen hat und das dazugehörige Datum.
Input	-
Output	Pro Paket eine Liste der Lokationen, die jedes Paket durchlaufen hat mit der Lokationsadresse, -nummer, -name und das Datum.
Motivation	Zur Ablage aller wichtigen Informationen des Paketdienstleisters wird diese Abfrage durchgeführt.
Klassenname	GetAllPackageHistories

3.2.6 Historie zufälliger Pakete auslesen

Überblick	Für ein zufälliges Paket wird die Historie ausgelesen. Diese beinhaltet die Daten jeder Lokation, die das Paket durchlaufen hat und das dazugehörige Datum.
Input	Die Paketnummer zur eindeutigen Identifikation.
Output	Eine Liste der Lokationen, die das Paket durchlaufen hat mit der Lokationsadresse, -nummer, -name und dem Datum.
Motivation	Ein Kunde zeigt sich die Historie seines versendeten Pakets an um es nach zu verfolgen.
Klassenname	GetRandomPackageHistories

3.2.7 Lokation aller Pakete auslesen

Überblick	Für jedes Paket wird die Lokation ausgelesen. Dies beinhaltet die Daten der Lokationen, in welchen sich die Pakete befinden.
Input	-
Output	Pro Paket eine Liste der Lokation, in welcher sich jedes Paket befindet mit der Lokationsadresse, -nummer und -name.
Motivation	Zur Ablage aller wichtigen Informationen über die Lokationen wird diese Abfrage durchgeführt.
Klassenname	GetAllPackageLocations

3.2.8 Lokation zufälliger Pakete auslesen

Überblick	Für ein zufälliges Paket wird die Lokation ausgelesen. Diese beinhaltet die Daten der Lokation, in welcher sich das Paket befindet.
Input	Die Paketnummer zur eindeutigen Identifikation.
Output	Eine Ausgabe der Lokation, in welcher sich das Paket derzeit befindet mit der Lokationsadresse, -nummer und -name.
Motivation	Ein Kunde zeigt sich den derzeitigen Aufenthaltsort seines versendeten Pakets an.
Klassenname	GetRandomPackageLocations

3.2.9 Pakete je Lokation auslesen

Überblick	Für jede Lokation werden die Pakete ausgelesen. Diese beinhaltet die Daten der Pakete, welche sich in den Lokationen befindet.
Input	Die Lokationsnummer zur eindeutigen Identifikation.
Output	Eine Ausgabe der Pakete, welche sich in den Lokationen derzeit befindet mit der Paketnummer.
Motivation	Zur Inventur einer Lokation liest ein Mitarbeiter eine Liste aus, welche alle Pakete der Lokation ausliefert.
Klassenname	GetPackagesAtAllLocations

3.2.10 Pakete, die eine Lokation mehrfach besucht haben, auslesen

Überblick	Liefert eine Liste von Paketen, die eine beliebige Lokation mehrfach besucht haben.
Input	Die Historie aller Pakete.
Output	Die Paketnummern zur eindeutigen Identifikation der Pakete, welche laut Historie eine Lokation mehrfach besucht haben.
Motivation	Somit kann überprüft werden ob die Pakete eine logische Abfolge der Lokationen während des Auslieferungsprozess durchlaufen.
Klassenname	GetAllMultipleVisitors

3.2.11 Stammdaten aller Pakete auslesen

Überblick	Für jedes Paket werden die Stammdaten ausgelesen. Diese beinhalten wichtige Informationen aller Pakete.
Input	-
Output	Eine Liste der Pakete, welche derzeit versendet werden, mit der Paketnummer, Absende- und Empfängeradresse.
Motivation	Zur Ablage aller wichtigen Informationen über die Pakete wird diese Abfrage durchgeführt.
Klassenname	GetAllPackagesTest

3.2.12 Stammdaten zufälliger Pakete auslesen

Überblick	Für ein gewähltes Paket werden die Stammdaten ausgelesen. Diese beinhalten wichtige Informationen des Pakets.
Input	Die Paketnummer zur eindeutigen Identifikation.
Output	Die Stammdaten des gewählten Pakets, welche derzeit versendet wird, mit der Paketnummer, Absende- und Empfängeradresse.
Motivation	Abfrage zur Überprüfung der Paketinformationen für die weitere Versendung.
Klassenname	GetRandomPackages

3.2.13 Pakete bewegen (update)

Überblick	Aktualisiert den Ort eines Paketes und dokumentiert die Änderung in der Historie.
Input	Die Paket- und Lokationsnummer zur eindeutigen Identifikation.
Output	-
Motivation	Sobald beispielsweise ein Paket von einer Lokation zu einer anderen Lokation versendet wurde und dort angekommen ist, muss diese Lokationswechsel in der Datenbank festgehalten werden.
Klassenname	MoveRandomPackages

3.2.14 Archivierung von Paketen und ihrer Historie (delete)

Überblick	Archiviert ein Paket, indem seine Daten und seine Historie aus der Datenbank entfernt werden.
Input	Die Paketnummer zur eindeutigen Identifikation.
Output	-
Motivation	Um die Datenbank, der zu versendenden Pakete, auf dem aktuellen Stand zu halten, müssen ausgelieferte Pakete aus der aktuellen Aufträge Datenbank gelöscht werden.
Klassenname	ArchiveRandomPackages

3.2.15 Load-Test / Batch-Load-Test

Überblick	Informationen über Pakete, Lokationen und Historien in eine bestehende Datenbank einfügen.
Input	Daten über Pakete (Paketnummer, Absender- und Empfängeradresse), Lokationen (Lokationsadresse, -nummer und -name) und Historie (Paketnummer, Lokationsnummer und Datum).
Output	-
Motivation	Um neue Pakete und Lokationen in die Datenbank einzufügen, als neue Pakete verschickt oder neue Lokationen gegründet werden.
Klassenname	LoadTest / BatchLoadTest

3.3 Implementierung

Das zuvor entwickelte Datenmodell soll nun in die ausgewählten Datenbanken umgesetzt und die Szenarien in Form von Performancetests durchgeführt werden. Hierzu wird eine Java-Anwendung entwickelt, welche automatisch alle Testszzenarien für alle Datenbanken durchführt und dabei die Zeit misst, welche die Ausführung in Anspruch nimmt.

Diese Anwendung ist im Paket "de.dhbwstuttgart.wi2011i.keyvalueperformance" implementiert. Die Klasse *Program* stellt dabei die *main*-Methode und damit den Einstiegspunkt dar. Zur Verwaltung der Abhängigkeiten zur Laufzeit mittels Dependency Injection (DI) sowie zur Injektion von Konfigurationsparametern wird das Spring-Framework verwendet. Während des Kompilierens werden Abhängigkeiten, die in der Datei "pom.xml" angegeben werden, mittels Maven automatisch aufgelöst. Tests verwenden das JUnit Framework.

3.3.1 Umsetzung der Datenmodelle

Bevor Performancetests durchgeführt werden können, muss das Datenmodell auf die jeweils auf die Datenbanken angepasst werden. Hierzu wird zunächst das Datenmodell in die drei Klassen auf Abb. 6 umgesetzt.

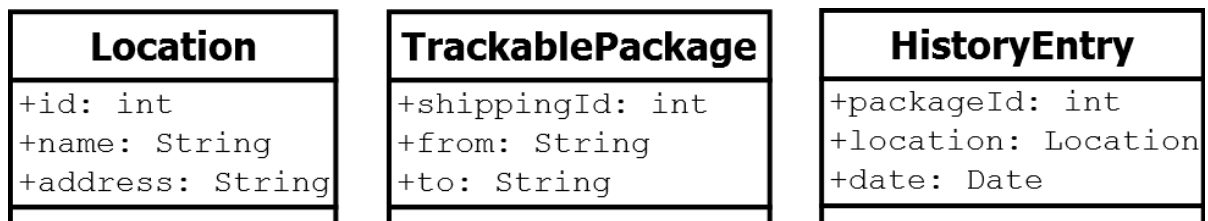


Abb. 6: Umsetzung des Datenmodells in Klassen

3.3.1.1 Das Repository-Interface

Um eine Duplizierung der späteren Testlogik zu vermeiden, wird ein Interface *Repository* bereitgestellt, welches in der Architektur eine generische Schnittstelle zur Datenbank bereitstellt, ohne sich auf eine konkrete Technologie zu beziehen. Aus Sicht der Anwendung kann somit ohne große Änderungen transparent ausgetauscht werden, welche Datenbank verwendet werden soll, was auch in der Praxis eine sinnvolle Designentscheidung darstellt. Die Methoden des Repository-Interfaces sind in Tabelle 1 aufgeführt und beschrieben und richten sich nach den Testszzenarien.

Tabelle 1: Methoden des Repository-Interfaces

Methoden	Beschreibung
List<Location> getAllLocations()	Liefert alle Lokationen, die in der Datenbank gespeichert sind.

Location getLocationById(int id)	Liefert eine spezifische Lokation in der Datenbank anhand ihrer ID.
void addLocation(Location locations)	Fügt eine Lokation in den Datenbestand ein.
List<TrackablePackage> getAllPackages()	Liefert alle Pakete.
TrackablePackage getPackageById(int id)	Liefert ein einzelnes Paket anhand dessen Sendungsverfolgungsnummer.
void addPackage(TrackablePackage pack)	Fügt ein Paket in den Datenbestand ein.
Location getPackageLocation(int id)	Liefert den aktuellen Ort eines Paketes anhand dessen Sendungsverfolgungsnummer.
void updatePackageLocation(int packageId, Location location, Date date)	Aktualisiert den Ort eines Paketes und dokumentiert die Änderung in der Historie.
int getTotalPackageCount(int locationId)	Ermittelt die Gesamtzahl der Pakete, die sich zu irgendeinem Zeitpunkt in ihrer Historie an der angegebenen Lokation befunden haben.
void archivePackage(int shippingId)	Archiviert ein Paket, indem seine Daten und seine Historie aus der Datenbank entfernt werden.
int getTotalTrackingTime(int shippingId)	Ermittelt die Anzahl an Tagen zwischen der ersten und der letzten Lokation in der Historie eines Paketes.
List<TrackablePackage> getReoccurringVisitors()	Liefert eine Liste von Paketen, die eine beliebige Lokation mehrfach besucht haben.
List<HistoryEntry> getHistory(int id)	Liefert die Historie eines Paketes.
int getLocationPackageCount(int id)	Liefert die Anzahl der Pakete, die sich zum aktuellen Zeitpunkt an einer angegebenen Lokation befinden.
List<TrackablePackage> getPackagesAtLocation(int id)	Liefert eine Liste aller Pakete, die sich zum aktuellen Zeitpunkt an einer angegebenen Lokation befinden.
void batchInsertLocations(List<Location> locations)	Fügt mehrere Lokationen gleichzeitig in die Datenbank ein.
void batchInsertPackages(Map<TrackablePackage, List<HistoryEntry>> packages)	Fügt mehrere Pakete und Historien gleichzeitig in die Datenbank ein.

Da während der Performancetests aus Gründen der Performance auf eine Überprüfung der Korrektheit der jeweiligen Rückgabewerte der Methoden verzichtet werden soll, ist es wichtig, sicherzustellen, dass die Implementierungen fehlerfrei funktionieren und tatsächlich die erwarteten Ergebnisse liefern. Andernfalls könnte ein später gemessener Performancevorteil einer bestimmten Datenbank lediglich aufgrund einer fehlerhaften Implementierung entstanden sein. Im Sinne von Test-Driven Development (TDD) werden die Tests vor der eigentlichen Implementierung vorgenommen. Die Tests sind in der Klasse *RepositoryTests* implementiert, welche jede Methode des Repository-Interfaces einzeln und für jede konkrete Implementierung automatisiert testet. Hierzu werden jeweils definierte Datenbestände in der Datenbank erzeugt und anschließend die Rückgabewerte der einzelnen Methoden mit den erwarteten Werten verglichen. Kommt es zu einer Abweichung, wird eine Ausnahme ausgelöst und der Test schlägt fehl.

Für jede Datenbank wird nun eine Implementierung des Repository-Interfaces bereitgestellt, was zu den in Tabelle 2 aufgeführten Klassen führt. Bei der Erstellung einer Instanz einer der Klassen liegt es außerdem in der Verantwortung der Implementierung, die zugrundeliegende Datenbank zu löschen und neu anzulegen bzw. sämtliche Daten aus ihr zu entfernen, um einen einheitlichen Ausgangszustand herstellen zu können.

Tabelle 2: Klassen des Repository-Interfaces pro Datenbank

Klasse	Datenbank
DerbyDbRepository	Apache Derby (File mode)
DerbyInMemoryRepository	Apache Derby (In-Memory Mode)
MongoDbRepository	MongoDB
MySqlRepository	Oracle MySQL
PostgresRepository	Postgres
RedisRepository	Redis

3.3.1.2 SQL-Datenbanken

Da die Implementierung der Repositories für die SQL-Datenbanken weitestgehend identisch ist und in der Basisklasse *JdbcRepository* zusammengefasst wird, sollen sie an dieser Stelle gemeinsam erläutert werden. Die Daten werden in drei Tabellen abgelegt, welche am Beispiel von MySQL mit folgenden drei SQL-Anweisungen angelegt werden:

```
CREATE TABLE LOCATIONS(ID INTEGER NOT NULL AUTO_INCREMENT, NAME
VARCHAR(100) NOT NULL, ADDRESS VARCHAR(250) NOT NULL, PRIMARY
KEY(ID));
```

```
CREATE TABLE PACKAGES(ID INTEGER NOT NULL AUTO_INCREMENT, SENDER
VARCHAR(250) NOT NULL, RECEIVER VARCHAR(250) NOT NULL, LOCATION_ID
INTEGER, PRIMARY KEY(ID), FOREIGN KEY(LOCATION_ID) REFERENCES
LOCATIONS(ID));
```

```
CREATE TABLE PACKAGE_HISTORY(PACKAGE_ID INTEGER NOT NULL,
LOCATION_ID INTEGER NOT NULL, DATE DATETIME NOT NULL, PRIMARY
KEY(PACKAGE_ID, DATE), FOREIGN KEY (LOCATION_ID) REFERENCES
LOCATIONS(ID), FOREIGN KEY(PACKAGE_ID) REFERENCES PACKAGES(ID) ON
DELETE CASCADE;
```

Code 1: Anlegen der Datenbanktabellen

Die Tabelle *LOCATIONS* hält die Informationen aller Lokationen. In *PACKAGES* sind die Stammdaten der Pakete hinterlegt. Außerdem wird mittels eines Foreign-Key die aktuelle Position eines Paketes gespeichert. *PACKAGE_HISTORY* speichert schließlich die Historie eines Paketes, wobei die Verweise auf das Paket und die Lokation ebenfalls mithilfe von Foreign-Keys realisiert werden. Tabelle 3 führt auf, welche SQL-Anweisungen genutzt werden, um die Methoden des *Repository*-Interfaces zu implementieren. Die Anweisungen werden dabei unter Nutzung von Java Database Connectivity (JDBC) zunächst als *Prepared-Statement* vorbereitet. Fragezeichen ('?') geben dabei variable Parameter an, welche zur Ausführungszeit gesetzt werden.

Tabelle 3: SQL-Anweisungen für die Methoden des Repository-Interfaces

Methode	SQL-Anweisung
getAllLocations	SELECT ID, NAME, ADDRESS FROM LOCATIONS;
getLocationById	SELECT ID, NAME, ADDRESS FROM LOCATIONS WHERE ID = ?;
addLocation	INSERT INTO LOCATIONS(NAME, ADDRESS) VALUES(?, ?);
getAllPackages	SELECT ID, SENDER, RECEIVER FROM PACKAGES;
getPackageById	SELECT ID, SENDER, RECEIVER FROM PACKAGES WHERE ID = ?;
addPackage	INSERT INTO PACKAGES(SENDER, RECEIVER) VALUES(?, ?);
getPackageLocation	SELECT l.ID, l.NAME, l.ADDRESS FROM LOCATIONS l JOIN PACKAGES p ON p.LOCATION_ID = l.ID WHERE p.ID = ?;

updatePackage-Location	UPDATE PACKAGES SET LOCATION_ID = ? WHERE ID = ?; INSERT INTO PACKAGE_HISTORY(PACKAGE_ID, LOCATION_ID, DATE) VALUES(?, ?, ?);
getTotalPackageCount	SELECT COUNT(*) FROM PACKAGE_HISTORY WHERE LOCATION_ID = ?;
archivePackage	DELETE FROM PACKAGES WHERE ID = ?;
getTotalTrackingTime	SELECT DATEDIFF(MAX(DATE), MIN(DATE)) FROM PACKAGE_HISTORY WHERE PACKAGE_ID = ? GROUP BY PACKAGE_ID; (Unterscheidet sich je nach Datenbank)
getReoccurringVisitors	SELECT p.ID, p.SENDER, p.RECEIVER FROM PACKAGE_HISTORY pl JOIN PACKAGES p ON p.ID = pl.PACKAGE_ID GROUP BY p.ID, p.Sender, p.Receiver HAVING COUNT(DISTINCT pl.LOCATION_ID) <> COUNT(pl.LOCATION_ID);
getHistory	SELECT l.ID, l.NAME, l.ADDRESS, pl.DATE FROM PACKAGE_HISTORY pl JOIN LOCATIONS l ON pl.LOCATION_ID = l.ID WHERE pl.PACKAGE_ID = ? ORDER BY pl.DATE ASC;
getLocationPackageCount	SELECT COUNT(*) FROM PACKAGES WHERE LOCATION_ID = ?;
getPackagesAt-Location	SELECT p.ID, p.SENDER, p.RECEIVER FROM PACKAGES p JOIN LOCATIONS l ON p.LOCATION_ID = l.ID WHERE l.ID = ?;
batchInsertLocations	INSERT INTO LOCATIONS(NAME, ADDRESS) VALUES(?, ?);
batchInsertPackages	INSERT INTO PACKAGES(SENDER, RECEIVER, LOCATION_ID) VALUES(?, ?, ?); INSERT INTO PACKAGE_HISTORY(PACKAGE_ID, LOCATION_ID, DATE) VALUES(?, ?, ?);

Anzumerken ist, dass sich lediglich die Anweisung für getTotalTrackingTime zwischen den einzelnen Datenbanken unterscheidet, da sie eine Funktion nutzt, welche nicht im SQL-

Standard enthalten ist und von den Herstellern mit unterschiedlichen Namen und Parametern implementiert ist. Unter Postgres lautet der entsprechende Teil beispielsweise "DATE_PART('day', MAX(DATE) - MIN(DATE))" und unter Derby "{fn TIMESTAMPDIFF(SQL_TSI_DAY, MIN(DATE), MAX(DATE))}". Weiterhin werden bei den *batch*-Methoden die *addBatch* und *executeBatch* Methoden des *PreparedStatements* genutzt, um mehrere Statements zusammenzufassen und damit die Ausführungsgeschwindigkeit zu steigern.

3.3.1.3 Redis

Redis als Key-Value-Datenbank unterstützt kein SQL. Folglich muss ein eigenständiges Datenmodell entwickelt und implementiert werden und auch Abfragen lassen sich nicht wiederverwenden. Die Implementierung nutzt in Anlehnung an das offizielle Tutorial¹⁰⁰ Hashmaps, um die Eigenschaften einzelner Objekte zu speichern, und nutzt Listen mit Keys zur Verwaltung von Abhängigkeiten. Tabelle 4 führt auf, nach welchen Mustern Keys aufgebaut sind, welchen Typ der dahinterstehende Wert hat und was darin abgelegt ist.

Tabelle 4: Keys der Objekte zur Verwaltung von Abhängigkeiten in Redis

Key	Typ	Beschreibung
location	Set	Enthält die Keys aller Lokationen in der Datenbank
location:{id}	Hash	Enthält die Informationen einer Lokation: <ul style="list-style-type: none"> ▪ id: ID der Lokation ▪ name: Name der Lokation ▪ address: Adresse der Lokation
location:key	String	Speichert die nächste freie ID für eine Lokation
location:{id}:packages	Set	Enthält die Keys aller Pakete, welche sich derzeit an einer bestimmten Lokation befinden.
package	Set	Enthält die Keys aller Pakete in der Datenbank
package:key	String	Speichert die nächste freie ID für ein Paket
package:{id}	Hash	Enthält die Informationen eines Paketes: <ul style="list-style-type: none"> ▪ id: Id des Paketes ▪ from: Absender des Paketes ▪ to: Empfänger des Paketes ▪ location: Key der Lokation, an der sich das Paket befindet
package:{id}:history:key	String	Speichert die nächste freie ID für einen Historieneintrag für ein Paket

¹⁰⁰ Vgl. Redis (o.J.)

package:{id}:history	Set	Enthält die Keys der Historieneinträge eines Paketes
package:{id}:history:{id}	Hash	Enthält die Informationen zu einem Historieneintrag eines Paketes: <ul style="list-style-type: none"> ▪ locationId: Id der Lokation ▪ locationAddress: Adresse der Lokation ▪ locationName: Name der Lokation ▪ date: Datum ▪ package: Id des Paketes

Zum Zugriff auf Redis aus Java wird jedis¹⁰¹ verwendet. Da Redis keine Prüfung der Daten auf Konsistenz durchführt, zeigen sich Fehler in der Implementierung erst während der Laufzeit oder bleiben unentdeckt. Außerdem bietet Redis keine Abfragesprache, was dazu führt, dass komplexere Abfragen in Java-Code implementiert werden müssen. Beispielsweise müssen zur Archivierung eines Paketes folgende Schritte durchgeführt werden:

1. Löschen aller Historieneinträge für das Paket:

```
int historyEntryCount = this.newId("package:" + packageId +
    ":history");
for (int historyEntryId = 0; historyEntryId <
    historyEntryCount; historyEntryId++) {
    String historyEntryKey =
    this.getHistoryEntryKey(packageId, historyEntryId);
    this.jedis.del(historyEntryKey);
}
```

2. Löschen des Sets, das die Historieneinträge enthält:

```
this.jedis.del("package:" + packageId + ":history");
```

3. Löschen der nächsten ID für Historieneinträge:

```
this.jedis.del("package:" + packageId + ":history:key");
```

4. Löschen des Keys des Paketes aus dem Set der Lokation, an der es sich gerade befindet:

```
String currentLocationKey = jedis.hget(packageKey, "location");
this.jedis.srem(currentLocationKey + ":packages", packageKey);
```

5. Löschen des Keys des Paketes aus der Liste der Pakete:

```
jedis.srem("package", packageKey);
```

6. Löschen der Paketdaten:

```
this.jedis.del(packageKey);
```

¹⁰¹ Vgl. Jedis (o.J.)

Die Ermittlung von Paketen, welche eine Lokation mehrfach besucht haben, kann ebenfalls nur in Java-Code erfolgen:

```
List<TrackablePackage> result = new ArrayList<TrackablePackage>();
Set<String> packageKeys = jedis.smembers("package");
for (String packageKey : packageKeys) {
    String historyKey = packageKey + ":history";
    Set<String> packageLocations = jedis.smembers(historyKey);
    HashSet<String> locations = new
HashSet<String>(packageLocations.size());
    for (String entryKey : packageLocations) {
        String key = this.jedis.hget(entryKey, "locationId");
        if (!locations.add(key)) {
            TrackablePackage pack =
this.toPackage(this.jedis.hgetAll(packageKey));
            if (!result.contains(pack))
                result.add(pack);
            break;
        }
    }
}
return result;
```

Code 2: Ermittlung von Paketen mit mehrfachem Lokationsbesuch

3.3.1.4 MongoDB

Die Fähigkeiten von MongoDB gehen über die von Redis hinaus, was dazu führt, dass ein separates Datenmodell entwickelt werden sollte. Das *MongoDbRepository* nutzt drei Collections, in denen die Daten abgelegt werden, welche in Abb. 7 dargestellt sind:

- counters: Enthält Informationen, um IDs für Lokationen und Pakete zu generieren
- locations: Enthält Informationen zu den Lokation
- packages: Enthält Informationen zu den Paketen und deren Historie

MongoDB erlaubt auch komplexe Abfragen über eine eigene Abfragesprache, die auf Objekten basiert. Diese Funktionalität wird genutzt, um die Methoden zu implementieren. Beispielsweise kann die Gesamtzahl der Pakete, die je eine bestimmte Lokation aufgesucht haben (*getTotalPackageCount*), mit folgendem Query ermittelt werden:

1. Zunächst werden die Felder *_id* und *history* jedes Paketes ausgelesen:

```
query.add(BasicDBObjectBuilder.start().add("$project",
BasicDBObjectBuilder.start().add("history",
BasicDBObjectBuilder.start().add("_id",1).get()).get()).get());
```

- Im Anschluss wird ein sogenannter Unwind des Arrays history durchgeführt, wodurch das Array quasi "ausgepackt" wird und für jedes Element darin ein Objekt mit dessen Properties und der `_id` des Paketes erstellt wird:

```
query.add(BasicDBObjectBuilder.start().add("$unwind",
"$history").get());
```

- Es folgt eine Prüfung der Id der Lokation, auf die sich der Historieneintrag bezieht:

```
query.add(BasicDBObjectBuilder.start().add("$match",
BasicDBObjectBuilder.start().add("history._id",
locationId).get()).get());
```

- Zum Schluss wird die Anzahl der Ergebnisse ermittelt:

```
query.add(BasicDBObjectBuilder.start().add("$group",
BasicDBObjectBuilder.start().add("_id", null).add("count",
BasicDBObjectBuilder.start().add("$sum", 1).get())
.get()).get());
```

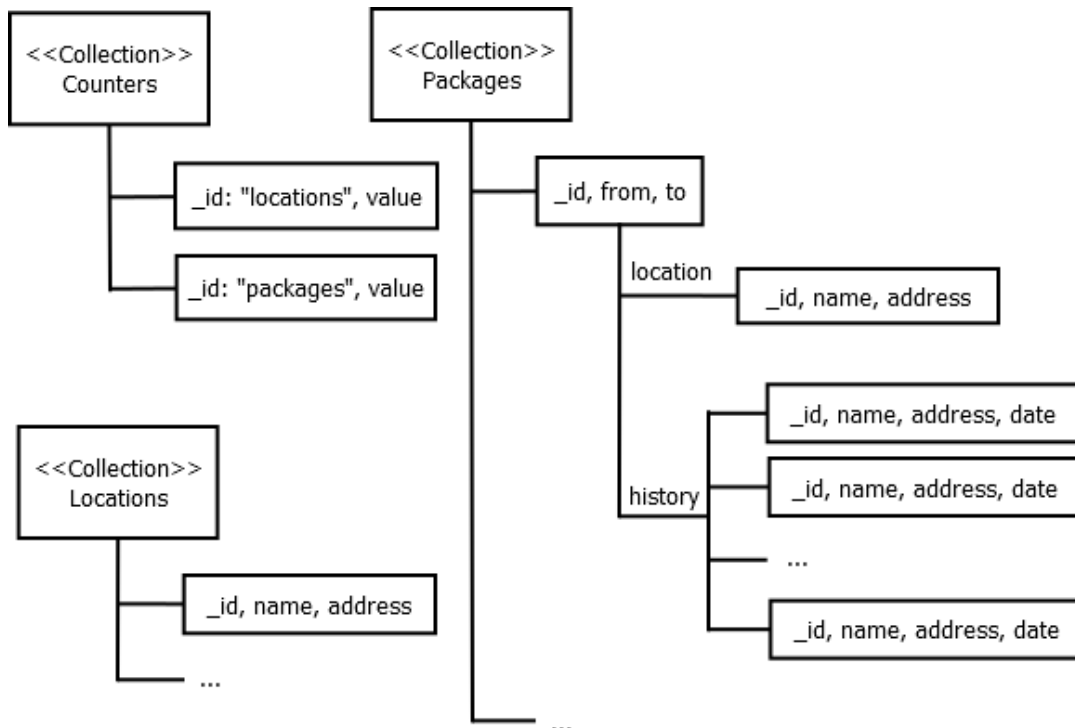


Abb. 7: Drei Collections in MongoDBRepository

Auch die Gesamtzeit, die ein Paket im Tracking verbraucht hat (`getTotalTrackingTime`) kann größtenteils von MongoDB ermittelt werden; lediglich die Differenz muss in Java-Code berechnet werden. Hierzu wird zunächst mittels des "\$match"-Operators das entsprechende Paket ermittelt, ein Unwind der Historie durchgeführt, um im Anschluss das kleinste und größte Datum zu ermitteln, welche zurückgegeben und dann im Java-Code weiterverarbeitet werden:

```
List<DBObject> query = new ArrayList<DBObject>();
query.add(BasicDBObjectBuilder.start().add("$match",
BasicDBObjectBuilder.start().add("_id", shippingId).get()).get());
```



```

query.add(BasicDBObjectBuilder.start().add("$unwind",
"$history").get());
query.add(BasicDBObjectBuilder.start().add("$project",
BasicDBObjectBuilder.start().add("date",
"$history.date").get()).get());
query.add(BasicDBObjectBuilder.start().add("$group",
BasicDBObjectBuilder.start().add("_id", "$_id").add("minDate",
BasicDBObjectBuilder.start().add("$min",
"$date").get()).add("maxDate",
BasicDBObjectBuilder.start().add("$max",
"$date").get()).get()).get());
AggregationOutput aggregate = this.packages.aggregate(query);
for (DBObject resultObj : aggregate.results()) {
    Date minDate = (Date) resultObj.get("minDate");
    Date maxDate = (Date) resultObj.get("maxDate");
    return Days.daysBetween(new DateTime(minDate.getTime()), new
DateTime(maxDate.getTime())).getDays();
}

```

Code 3: Berechnung der Gesamtzeit eines Pakets für MongoDB

Pakete, bei denen dieselbe Lokation mehrfach in der Historie auftaucht (*getReoccurringVisitors*) können ebenfalls mittels eines Queries ermittelt werden:

1. Absender, Empfänger und Historie werden ermittelt:

```

BasicDBObjectBuilder.start().add("$project",
BasicDBObjectBuilder.start().add("from", 1).add("to",
1).add("history", BasicDBObjectBuilder.start().add("_id",
1).get()).get()).get()

```

2. Ein Unwind wird auf die Historie angewandt:

```

BasicDBObjectBuilder.start().add("$unwind", "$history").get()

```

3. Das Ergebnis wird nach Paket-ID (*_id*) und Id der Lokation (*\$history._id*) gruppiert und jeweils die Anzahl der Elemente mit ebendieser Kombination ermittelt:

```

BasicDBObjectBuilder.start().add("$group",
BasicDBObjectBuilder.start().add("_id",
BasicDBObjectBuilder.start().add("locationId",
"$history._id").add("package",
BasicDBObjectBuilder.start().add("from", "$from").add("to",
"$to").add("_id", "$_id").get()).get()).add("count",
BasicDBObjectBuilder.start().add("$sum", 1).get()).get()).get()

```

4. Mithilfe von \$match werden alle Gruppen mit einer Elementzahl größer als eins ermittelt:

```

BasicDBObjectBuilder.start().add("$match",
BasicDBObjectBuilder.start().add("count",
BasicDBObjectBuilder.start().
add("$gt", 1).get()).get()).get()

```

5. Zum Schluss werden die Gruppen wieder aufgelöst und die Pakete ermittelt:

```
BasicDBObjectBuilder.start().add("$project",
BasicDBObjectBuilder.start().add("_id", 0).add("package",
"$_id.package").get()).get()
BasicDBObjectBuilder.start().add("$group",
BasicDBObjectBuilder.start().add("_id",
"$package").get()).get()
```

3.3.2 Testdatengenerierung

Um die Szenarien in einer praxisnahen Umgebung ausführen zu können, ist es notwendig, realitätsnahe Testdaten zu verwenden. Diese manuell zu generieren, ist zwar möglich, aber sehr aufwendig. Aus diesem Grund wurde eine automatische Testdatengenerierung implementiert, welche bei allen Performancetests zum Einsatz kommt. Insgesamt sind drei Arten von Testdaten zu generieren:

1. Namen für die Absender- und Empfängeradresse der Pakete
2. Adressen für die Adressen der Pakete und für die Lokationen
3. Datumsangaben für die Historien

Die Generierung von Datumsangaben für die Historie durch die Klasse *DateGenerator* ist sehr einfach gehalten: Zu dem 01.01. eines Jahres zwischen 2005 und 2012 wird eine zufällige Anzahl an Tagen zwischen 0 und 365 addiert. Im Anschluss werden aus dem Ergebnis weitere Tage generiert, die zufällig zwischen 1 und 13 Tage aufeinander folgen.

Namen werden von *NameGenerator* erzeugt, indem ein zufälliger Vorname mit einem zufälligen Nachnamen kombiniert wird. Dabei können die Vornamen und Nachnamen, aus denen ausgewählt wird, in der Konfigurationsdatei hinterlegt werden. Standardmäßig sind als Vornamen Namen aus einem Wikipediaartikel¹⁰² eingetragen, während die Nachnamen ebenfalls aus einem Wikipediaartikel¹⁰³ stammen, welche mit folgendem Javascript-Code (unter Verwendung von jQuery¹⁰⁴) extrahiert wurden:

```
var text = $("#div").text();
var re = /([A-Za-zäöüß]+)/g;
var result = [];
while (match = re.exec(text)) {
    var name = match[1];
    result.push(name);
}
$("#div").text(result.join());
```

Code 4: Testdaten aus Wikipedia extrahieren

¹⁰² Vgl. Wikipedia (2014 a)

¹⁰³ Vgl. Wikipedia (2014 b)

¹⁰⁴ Vgl. jQuery (o.J.)

Adressen folgen dem Format “[Straßenname] [Hausnummer], [PLZ] [Ort]”. Hausnummer und PLZ sind Zufallszahlen, während Straßen- und Ortsnamen auf sehr ähnliche Weise wie Namen generiert werden: Sie werden zufällig aus zwei Bestandteilen zusammengesetzt, wobei der erste für Straßen und Orte identisch ist (“Test”, “Beispiel”, ...) und der zweite aus häufigen Suffixen besteht (z.B. “stadt” oder “dorf” für Orte und “straße” oder “weg” für Straßen).

3.3.3 Testausführung

Tests werden von der Klasse *PerformanceTestRunner* ausgeführt, welche einen Performancetest koordiniert. Jedes zu testende Szenario wird in einer Klasse implementiert, welche von *PerformanceTestBase* erbt, die unter anderem die beiden Methoden *initialize* und *execute* bereitstellt. Erstere Methode erlaubt dem Test, die Datenbank entsprechend vorzubereiten, ohne, dass dies die gemessene Zeit beeinflusst. Gemessen wird nur die Ausführungszeit von *execute*. Jede Instanz einer *PerformanceTestBase* wird über mehrere Testdurchläufe verwendet, um sicherzustellen, dass alle getesteten Datenbanken mit identischen Testdaten gemessen werden. Performancetests treten ausschließlich über das zuvor eingeführte Repository-Interface in Kontakt zu den Datenbanken.

Als Unterklasse von *PerformanceTestBase* existiert *DataTestBase*, welche bei ihrer Erzeugung eine in der Konfigurationsdatei konfigurierbare Anzahl an Lokationen, Paketen und Historieneinträgen erzeugt und diese bei der Initialisierung in die Datenbank schreibt. Von ihr erben wiederum *RandomTestBase* und *AllTestBase*, welche gemeinsame Logik für weitere Testklassen kapseln. Tabelle 5 listet alle Testklassen.

Tabelle 5: Übersicht der Testklassen

Klassenname	Beschreibung
GetAllPackageHistories	Historie aller Pakete auslesen
GetRandomPackageHistories	Historie zufälliger Pakete auslesen
GetAllPackagesTest	Stammdaten aller Pakete auslesen
GetRandomPackages	Stammdaten zufälliger Pakete auslesen
GetAllPackageLocations	Lokation aller Pakete auslesen
GetRandomPackageLocations	Lokation zufälliger Pakete auslesen
GetPackagesAtAllLocations	Pakete je Lokation auslesen
GetAllLocationsTest	Alle Lokationen auslesen
GetAllLocationPackageCount	Füllstand aller Lokationen auslesen

GetAllLocationTotalPackageCount	Frequentierung der Lokation auslesen
MoveRandomPackages	Pakete bewegen (update)
GetRandomPackageTrackingTime	Gesamte Verweildauer pro Paket in der Logistikkette auslesen
GetAllMultipleVisitors	Pakete, die eine Lokation mehrfach besucht haben, auslesen
ArchiveRandomPackages	Archivierung von Paketen und ihrer Historie (delete)
LoadTest	Load Test
BatchLoadTest	Batch Load Test

Bei der Ausführung des Testprogrammes werden alle Tests mit allen Repositories beliebig oft ausgeführt. Vor jeder Testausführung werden die Datenbanken gelöscht bzw. geleert, um für jeden Test identische Ausgangsbedingungen zu schaffen. Neben der Messung der Ausführungszeit mithilfe einer *Stopwatch* Instanz stellt der *PerformanceTestRunner* auch sicher, dass ein Test nach Erreichen eines Timeouts abgebrochen wird. In diesem Fall kann davon ausgegangen werden, dass der Test weitaus länger dauern würde als es sinnvoll im Rahmen einer Messung wäre. Zudem wird vor jeder Testausführung eine Garbage Collection (GC) angestoßen und abgewartet, um das Risiko zu minimieren, dass ein GC während der Testausführung auftritt und folglich das Ergebnis verzerrt.

Neben einer Protokollierung der aktuellen Aktivität in die Standardausgabe, werden alle Testergebnisse in eine comma separated-values (CSV)-Datei exportiert. Diese besteht aus drei Spalten:

1. Test gibt an, welcher Test ausgeführt wurde
2. DB gibt an, welche Datenbank verwendet wurde
3. Duration enthält die Ausführungsdauer in Nanosekunden. Es empfiehlt sich, die Zeit auf Millisekunden umzurechnen, da die zusätzliche Präzisions nur wenig Aussagekraft besitzt. Ist der Wert in Duration kleiner als 0, ist ein Fehler aufgetreten: -1 steht für einen Timeout des Testes, während -2 eine Exception während der Ausführung kennzeichnet.

Durch eine Konfigurationsdatei mit Schlüssel-Wert-Paaren kann die Testausführung konfiguriert werden. Die Parameter werden in Tabelle 6 aufgeführt und erklärt:

Tabelle 6: Konfigurationsparameter der Testausführung

Name	Standard	Beschreibung
redis.host	localhost	Hostname des Redis-Servers
redis.port	6379	Portnummer des Redis-Servers
mongoDb.host	localhost	Hostname des MongoDB-Servers
mongoDb.port	27017	Portnummer des MongoDB-Servers
mongoDb.db	test	Datenbankname auf dem MongoDB-Server
mySqlUrl	jdbc:mysql://localhost	Connection-URL zu MySQL
postgresUrl	jdbc:postgresql://localhost:5432/	Connection-URL zu Postgres
derbyUrl	jdbc:derby:test;create=true	Connection-URL zu Derby
derbyInMemoryUrl	jdbc:derby:memory:test;create=true	Connection-URL zu Derby (in memory)
resultPath	C:\	Ordner, in dem die Ergebnisdateien (CSV) erzeugt werden
numberOfTestRuns		Gibt an, wie oft alle Tests durchgeführt werden sollen
tests.timeout		Die Zeit in Sekunden, nach der ein Timeout bei der Testausführung ausgelöst werden soll
RandomTest.Count		Die Anzahl der Pakete / Lokationen, die in einem Random-Test verwendet werden soll
GetAllTests.Count	3	Die Anzahl, wie oft ein Test durchgeführt werden soll, der alle Pakete / Lokationen verarbeitet
tests	...	Eine durch Kommata getrennte Liste der auszuführenden Tests

repositories	...	Eine durch Kommata getrennte Liste mit Repositories, die verwendet werden sollen
part1	...	Siehe Abschnitt Testdatengenerierung
streetPart	...	Siehe Abschnitt Testdatengenerierung
cityPart	...	Siehe Abschnitt Testdatengenerierung
firstNames	...	Siehe Abschnitt Testdatengenerierung
lastNames	...	Siehe Abschnitt Testdatengenerierung
dataTest.locationCount		Die Anzahl der Lokationen, die generiert werden soll
dataTest.packageCount		Die Anzahl der Pakete, die generiert werden soll
data-Test.minPackageLocations		Die minimale Anzahl an Historieneinträgen pro Paket
data-Test.maxPackageLocations		Die maximale Anzahl an Historieneinträgen pro Paket

3.4 Durchführung der Messungen

In allen Testszenarien ist die Abarbeitungsdauer die abhängige Variable, die als leistungsbestimmender Faktor gewählt wird. Vor der tatsächlichen Testdurchführung ist es wichtig, zu berücksichtigen, die Auswahl welcher Rahmenbedingungen und Faktoren signifikante Unterschiede in der Leistungsfähigkeit der verschiedenen Datenbanksysteme bzw. Klassen von Datenbanksystemen aufzeigen kann, und dabei die Vergleichbarkeit der Ergebnisse zu erhalten. Aufgrund der begrenzten Zeit und Ressourcen für die Testdurchführung ist es darüber hinaus wichtig, durch eine gute Planung den Aufwand für die Testdurchführung möglichst gering zu halten. Dieses Kapitel soll der Beschreibung der für die Testdurchführung gewählten Ausprägung aller relevanten Parameter dienen, sowie der Bestimmung der unabhängigen Variable.

Zu diesen veränderbaren Faktoren der Testdurchführung gehören zunächst die *Konfiguration der Testumgebung* (3.4.1), die durch die Parameter aus Tabelle 6 festgelegt wird. Zusätzlich ist es zur Begründung der Ergebnisse nicht nur relevant die *Hardware* (3.4.2), auf der die Datenbanken ausgeführt werden, zu kennen, sondern zu gewährleisten, dass alle Datenbanken auf vergleichbarer Hardware ausgeführt werden. Weiterhin spielt die *Konfiguration der Datenbanken* (3.4.3) eine Rolle, etwa bezüglich der Persistenz.

3.4.1 Konfiguration der Testumgebung

Nicht alle Parameter aus Tabelle 7 sind für die Testausführung von Relevanz. Im Folgenden wird auf die Festlegung der Relevanten eingegangen:

Tabelle 7: Konfiguration der Testumgebung

Parameter	Relevanz	Wert	Begründung
numberOfTestRuns	Eine Verfälschung der Testergebnisse durch Ausreißerwerte kann durch die Berechnung der Durchschnittszeit aus mehreren Ausführungen verhindert werden	3	Der Durchschnitt aus drei Ausführungszeiten desselben Szenarios stellt einen Kompromiss dar zwischen Verlässlichkeit der Ergebnisse und angemessener Ausführungszeit
Tests.timeout	Um eine unverhältnismäßig lange Testausführung, insbesondere auch im Fehlerfall, zu vermeiden, muss die Ausführungsdauer einzelner Tests beschränkt werden	120s	Bei längeren Ausführungszeiten würde die Ausführungszeit unverhältnismäßig lang; für den Fall eines Timeouts wird die Ausführungsdauer mit 120s ausgewiesen
RandomTest.Count	Die Anzahl der Datensätze, die bei einer Zufallsauswahl ausgewählt werden, hat direkten Einfluss auf die Ausführungszeit	2000	Der Wert ist kleiner als die kleinste gewählte Anzahl an Paketen, aber groß genug, um eine ausreichende Ausführungszeit zu gewährleisten
data-Test.locationCount	Die Anzahl der Lokationen bestimmt die Komplexität der Datenbank und damit die Ausführungszeit	50	Für das gewählte Beispiel eines Paketdienstleisters ist ein Netzwerk von 50 Umschlagplätzen realistisch
dataTest.minPackageLocations	Die Anzahl der Stationen eines Paketes bestimmt die Größe der Datenmenge und damit die Ausführungszeit einiger Szenarien	1	Zwischen 1 und 10 (Zwischen)-stationen je Paket sind realistisch für einen Paketdienstleister
dataTest.maxPackageLocations		10	
dataTest.packageCount	Die Anzahl der Pakete bestimmt die Größe der Datenmenge und damit die Ausführungszeit der meisten Szenarien	5000 10000 20000 40000 80000	Die Anzahl der Pakete in der Datenbank wird als unabhängige Variable gewählt; es werden zur späteren Vergleichbarkeit der Ergebnisse für jedes Szenario die gleichen Paketzahlen gewählt

3.4.2 Hardware

Um die Vergleichbarkeit der Leistungstests zu gewährleisten, muss die Hardware der Testumgebung bei allen Szenarien gleich sein. Datenbankclient und -server auf jeweils eigener Hardware auszuführen, würde einerseits die Prozessor- und Speichernutzung des Clients als verfälschenden Einflussfaktor auf die Ausführungszeit ausschließen, würde jedoch die Leistungsfähigkeit des Netzwerks als weitere Unbekannte hinzufügen. Für die Tests in dieser Arbeit wurde deshalb auf eine physische Trennung von Client und Server verzichtet.

Alle Tests wurden auf einem Thinkpad T420 ausgeführt:

Tabelle 8: Hardware der Testumgebung

Betriebssystem	Windows 7 Professional 64-bit
Prozessor	Intel Core i5-2520M, 2 physikalische und 2 virtuelle Kerne, 2,5 GHz
Arbeitsspeicher	4 GB

3.4.3 Konfiguration der Datenbanken

Alle getesteten Datenbanken mit Ausnahme von Redis und Derby sind vollständig persistent. Bei Redis und Derby besteht die Möglichkeit, den Einfluss eines Verzichts auf einen gewissen Grad an Persistenz auf die Ausführungsdauer zu testen. Dazu wird Derby neben der vollständig persistenten Version auch allein im Arbeitsspeicher (in Memory) ausgeführt. Bei Redis lässt sich der Grad der Persistenz über die Konfiguration des Append Only File (AOF) steuern. Zum einen kann der derzeitige Stand sekundlich protokolliert werden, oder, um volle Persistenz zu erhalten, bei jeder Transaktion. Alle Testszenarien werden für beide Konfigurationsmöglichkeiten durchgeführt.

Um eine Verfälschung der Ausführungszeiten durch Initialisierungszeiten auszuschließen, laufen während eines Testlaufs alle Datenbankprozesse. Vor jeder Szenarioausführung wird jede Datenbank zudem vor Start der Zeitmessung neu aufgesetzt, womit auch eine Verfälschung der Ausführungszeiten durch z.B. den Datenbankcache oder etwaige Query Optimizer ausgeschlossen werden kann. Einschränkend ist hierbei anzumerken, dass der AOF bei Redis nur vor einem kompletten Testlauf geleert wird, nicht aber vor jeder einzelnen Szenarioausführung.

4 Auswertung der Messergebnisse

(Christian Brummer, Michael Busam, Thorsten Kühner, Klarissa Schmidt, Matthias Welz)

Dieses Kapitel dient der Beschreibung und Interpretation der Messergebnisse. Im Anschluss werden die Ergebnisse im Hinblick auf ihre Plausibilität bewertet. Abschließend werden auf Basis der Testergebnisse Schlussfolgerungen für mögliche Nutzungsszenarien der einzelnen Datenbanken gezogen.

Während die Darstellung und Beschreibung der Messergebnisse für jedes der sechzehn Testszenarien im Einzelnen erfolgt, ist es sinnvoll, die Einzelergebnisse für den Zweck der Interpretation zu gruppieren. Einerseits können zwar bereits einzelne Szenarien Rückschlüsse auf die Leistungsfähigkeit einer bestimmten Datenbank oder Klasse von Datenbanken zulassen, andererseits ist die Gefahr von Fehlinterpretationen hoch, etwa weil eine bestimmte Datenbank in einem bestimmten Anwendungsszenario allein aufgrund der gewählten Testumgebung im Vorteil war. Insbesondere bei den vielfältigen verschiedenen Abfragen besteht dann das Risiko, eine Datenbank zu bevorzugen.

4.1 Beschreibung der Messergebnisse

Für jedes der 16 Testszenarien werden im Folgenden die mittleren Ausführungszeiten jeder Datenbank für die verschiedenen Paketzahlen n in Millisekunden in tabellarischer Form aufgeführt. Die kursive Darstellung der Ausführungszeit markiert das Auftreten eines Timeouts. Für die Szenarien, bei denen nur 2.000 der Pakete betroffen sind, wird dies in der ersten Spalte der Ergebnistabelle durch die Darstellung 2.000/x kenntlich gemacht. Derby (iM) steht dabei für die in Memory-Ausführung von Derby. Redis (T) steht für die Ausführung von Redis mit Ergänzung des AOF bei jeder Transaktion, während Redis (1s) die sekundliche Ergänzung des AOF bezeichnet. Zusätzlich werden die auffälligsten Ergebnisse kurz hervorgehoben. Somit legt dieses Kapitel dar, wie die einzelnen Datenbanken die spezifischen Testszenarien bewältigen können.

4.1.1 Alle Lokationen auslesen

Tabelle 9 zeigt die Leistungstestergebnisse der Abfrage aller Lokationsdaten, die von fast allen Datenbanken am schnellsten ausgeführt wird. Auffallend ist, dass trotz der Abfrage der Lokationsdaten die Abfragezeiten bei allen Datenbanken im Test mit den ansteigenden Paketzahlen korrelieren, was nicht erwartbar ist. Die NoSQL-Datenbanken schneiden insgesamt schlechter ab als die relationalen, wobei MongoDB für jede Paketzahl die langsamste Datenbank ist.

Tabelle 9: Ergebnis GetAllLocationsTest

GetAllLocationsTest							
	Relationale Datenbanken				NoSQL Datenbanken		
n	MySQL	PostgreSQL	Derby	Derby (iM)	MongoDB	Redis (T)	Redis (1s)
5000	11	7	9	8	24	18	17
10000	8	7	7	12	33	20	25
20000	14	16	13	12	56	41	36
40000	26	27	29	36	89	67	65
80000	43	61	72	67	154	102	104

4.1.2 Füllstand aller Lokationen auslesen

Besonders auffällig in den Ergebnissen dieses Szenarios in Tabelle 10 ist die starke Diskrepanz in der Ausführungszeit zwischen den beiden NoSQL-Vertretern, da die Ausführungszeiten von MongoDB oft etwa hundertmal länger sind als die von Redis. Die Ausführungszeit von Redis kann unabhängig von der AOF-Konfiguration mit MySQL, der schnellsten Datenbank im Test, mithalten.

Tabelle 10: Ergebnis GetAllLocationPackageCount

GetAllLocationPackageCount							
	Relationale Datenbanken				NoSQL Datenbanken		
n	MySQL	PostgreSQL	Derby	Derby (iM)	MongoDB	Redis (T)	Redis (1s)
5000	58	254	84	49	11087	25	26
10000	142	474	152	146	21206	81	61
20000	315	1225	491	502	41265	302	221
40000	803	2905	1605	1431	90435	1187	1205
80000	2465	8664	7054	7781	120000	4528	3165

4.1.3 Frequentierung der Lokation auslesen

Bei dieser Abfrage schneiden alle NoSQL-Datenbanken bis um den Faktor 1.000 schlechter ab als alle relationalen Datenbanken, wobei die tatsächlichen Ausführungszeiten bereits ab der Auswahl aus 40.000 Paketen aufgrund von Timeouts nicht mehr korrekt dargestellt werden können. Unter den relationalen Datenbanken fällt Tabelle 11 insbesondere die stark überdurchschnittliche Ausführungszeit von PostgreSQL auf.

Tabelle 11: Ergebnis GetAllLocationTotalPackageCount

GetAllLocationTotalPackageCount							
	Relationale Datenbanken				NoSQL Datenbanken		
n	MySQL	PostgreSQL	Derby	Derby (iM)	MongoDB	Redis (T)	Redis (1s)
2000/5000	78	824	57	56	59665	70466	68774
2000/10000	110	1156	91	77	72276	88903	97804
2000/20000	152	1605	183	182	102119	115424	111778
2000/40000	302	2664	416	401	120000	120000	120000
2000/80000	421	3658	1216	1762	120000	120000	120000

4.1.4 Gesamte Verweildauer pro Paket in der Logistikkette auslesen

Tabelle 12 zeigt die Messergebnisse der Abfrage der gesamten Verweildauer eines bestimmten Paketes in der Logistikkette. Auch in diesem Szenario schneiden alle NoSQL-Datenbanken gleichermaßen signifikant schlechter als alle relationalen Datenbanken ab, jedoch deutlich weniger ausgeprägt als im vorangehenden Szenario. Darüber hinaus ist der überproportionale Anstieg der Ausführungszeit für die in Memory Version von Derby auffällig.

Tabelle 12: Ergebnis GetRandomPackageTrackingTime

GetRandomPackageTrackingTime							
	Relationale Datenbanken				NoSQL Datenbanken		
n	MySQL	PostgreSQL	Derby	Derby (iM)	MongoDB	Redis (T)	Redis (1s)
2000/5000	244	176	197	98	578	966	789
2000/10000	297	255	341	176	667	1098	1265
2000/20000	383	292	708	644	799	1225	1187
2000/40000	487	778	781	772	1702	2165	1967
2000/80000	605	1523	1267	1899	3880	5427	4728

4.1.5 Historie aller Pakete auslesen

Die Ausführungszeit dieses Szenarios ist für alle Datenbanken ähnlich, wobei für kleinere Paketzahlen ein Vorteil für Derby festgestellt werden kann, für größere Paketzahlen ein Vorteil für die dokumentenbasierte MongoDB. Gemäß der Messergebnisse in Tabelle 13 schneidet allein Redis deutlich schlechter ab, unabhängig von der AOF-Konfiguration.

Tabelle 13: Ergebnis GetAllPackageHistories

GetAllPackageHistories							
	Relationale Datenbanken				NoSQL Datenbanken		
n	MySQL	PostgreSQL	Derby	Derby (iM)	MongoDB	Redis (T)	Redis (1s)
5000	4226	3903	2192	2066	4423	7544	7065
10000	8766	8445	6776	5122	9665	15443	15667
20000	19901	19706	18904	16704	19993	32726	33445
40000	48809	44906	52703	43221	39057	83225	80064
80000	117803	113926	120000	120000	95546	120000	120000

4.1.6 Historie zufälliger Pakete auslesen

Tabelle 14 zeigt, dass MySQL dieses Szenario durchgehend in der kürzesten Ausführungszeit bewältigt. Währenddessen brauchen auch hier alle NoSQL-Vertreter etwa zwei- bis dreimal länger für die Ausführung als die relationalen Datenbanken, wobei weder für MongoDB noch Redis jeweils deutliche Vorteile zeigen.

Tabelle 14: Ergebnis GetRandomPackageHistories

GetRandomPackageHistories							
	Relationale Datenbanken				NoSQL Datenbanken		
n	MySQL	PostgreSQL	Derby	Derby (iM)	MongoDB	Redis (T)	Redis (1s)
2000/5000	155	156	344	302	655	905	903
2000/10000	176	163	708	705	854	1144	1097
2000/20000	271	182	1004	933	1142	1383	1271
2000/40000	371	557	2117	2004	2629	2744	2644
2000/80000	755	905	2795	3011	5121	4901	5176

4.1.7 Lokation aller Pakete auslesen

Auffällig in den Ergebnissen dieses Szenarios in Tabelle 15 ist die unterdurchschnittliche Ausführungszeit beider Konfigurationen von Redis, während insbesondere Derby einen stark überproportionalen Anstieg der Ausführungszeit für große Paketzahlen zeigt.

Tabelle 15: Ergebnis GetAllPackageLocations

GetAllPackageLocations							
	Relationale Datenbanken				NoSQL Datenbanken		
n	MySQL	PostgreSQL	Derby	Derby (iM)	MongoDB	Redis (T)	Redis (1s)
5000	2871	2088	883	497	3655	2551	2508
10000	6566	5122	2466	1676	7908	4904	4109
20000	14992	12722	7881	5880	17604	10002	9163
40000	36779	30008	31226	32657	42165	22768	22401
80000	99043	86549	114376	108773	107567	67198	68092

4.1.8 Lokation zufälliger Pakete auslesen

Tabelle 16 beinhaltet die Messergebnisse der Abfrage der Lokation zufälliger Pakete. Während die NoSQL-Datenbanken für dieses Szenario wiederum eine signifikant höhere Ausführungszeit haben als die relationalen Datenbanken, fällt insbesondere der im Vergleich schwache Anstieg der Ausführungszeiten bei größeren Paketzahlen auf.

Tabelle 16: Ergebnis GetRandomPackageLocations

GetRandomPackageLocations							
	Relationale Datenbanken				NoSQL Datenbanken		
n	MySQL	PostgreSQL	Derby	Derby (iM)	MongoDB	Redis (T)	Redis (1s)
2000/5000	178	204	85	91	374	556	521
2000/10000	251	209	167	144	489	655	604
2000/20000	299	296	308	314	671	704	809
2000/40000	478	609	402	684	802	1203	1324
2000/80000	871	811	1675	1561	1723	1992	1792

4.1.9 Pakete je Lokation auslesen

Die Ausführungszeiten für MySQL bei den relationalen Datenbanken und für MongoDB bei den NoSQL-Datenbanken sind stark unterdurchschnittlich, wobei MongoDB die zweit-schnellste Datenbank in diesem Szenario ist. Hingegen sind die Ausführungszeiten für Redis signifikant höher als für alle anderen getesteten Datenbanken, wie aus Tabelle 17 ersichtlich wird.

Tabelle 17: Ergebnis GetPackagesAtAllLocations

GetPackagesAtAllLocations							
	Relationale Datenbanken				NoSQL Datenbanken		
n	MySQL	PostgreSQL	Derby	Derby (iM)	MongoDB	Redis (T)	Redis (1s)
2000/5000	41	778	221	206	57	8262	7344
2000/10000	54	1122	441	322	81	10443	9105
2000/20000	81	1312	885	771	108	12855	12609
2000/40000	99	2708	1551	1335	176	21436	19925
2000/80000	167	4874	3417	2705	671	29043	27698

4.1.10 Pakete, die eine Lokation mehrfach besucht haben, auslesen

Bei der Ausführung dieses Szenarios fällt das sehr uneinheitliche Ergebnis sowohl bei den relationalen wie auch bei den NoSQL-Datenbanken auf. Während PostgreSQL die mit Abstand besten Ausführungszeiten unter den relationalen Datenbanken wie auch insgesamt liefert, sind gemäß Tabelle 18 die Ausführungszeiten für Derby und MongoDB stark unterdurchschnittlich, wobei MongoDB etwa fünfmal langsamer ist als Redis.

Tabelle 18: Ergebnis GetAllMultipleVisitors

GetAllMultipleVisitors							
	Relationale Datenbanken				NoSQL Datenbanken		
n	MySQL	PostgreSQL	Derby	Derby (iM)	MongoDB	Redis (T)	Redis (1s)
5000	262	31	1370	671	7155	1484	1712
10000	676	102	3662	2772	15442	2988	3678
20000	1556	229	9755	9022	34808	6704	6792
40000	4484	679	36474	29788	87669	18702	16554
80000	15232	2294	108772	115601	120000	44023	45192

4.1.11 Stammdaten aller Pakete auslesen

Tabelle 19 zeigt, dass bei diesem Szenario die Ausführungszeiten von Redis vergleichbar mit den meisten relationalen Datenbanken sind, während MongoDB für jede Paketzahl die langsamsten Ausführungszeiten liefert. MySQL führt die Abfrage der Stammdaten aller Pakete am schnellsten aus.

Tabelle 19: Ergebnis GetAllPackagesTest

GetAllPackagesTest							
	Relationale Datenbanken				NoSQL Datenbanken		
n	MySQL	PostgreSQL	Derby	Derby (iM)	MongoDB	Redis (T)	Redis (1s)
5000	55	244	21	16	1566	165	121
10000	121	391	67	65	2875	362	241
20000	279	802	217	242	7125	676	671
40000	599	1667	707	819	14255	1328	1181
80000	1607	5144	4165	3774	29088	3765	3481

4.1.12 Stammdaten zufälliger Pakete auslesen

Im Gegensatz zum vorherigen Szenario sind in Tabelle 20 wiederum die Ausführungszeiten von MongoDB vergleichbar mit den relationalen Datenbanken Derby und MySQL, die jeweils die besten Ausführungszeiten liefern. Hingegen liefert PostgreSQL bei den relationalen und Redis bei den NoSQL-Datenbanken signifikant schlechtere Ausführungszeiten.

Tabelle 20: Ergebnis GetRandomPackages

GetRandomPackages							
	Relationale Datenbanken				NoSQL Datenbanken		
n	MySQL	PostgreSQL	Derby	Derby (iM)	MongoDB	Redis (T)	Redis (1s)
2000/5000	165	788	69	156	302	1702	1765
2000/10000	233	908	122	252	404	2252	2288
2000/20000	308	1054	185	229	433	2655	2802
2000/40000	601	2114	576	451	721	4276	4075
2000/80000	1106	4791	1190	1062	1341	5022	5176

4.1.13 Pakete bewegen (update)

Tabelle 21 illustriert, dass das Update-Szenario von MySQL am schnellsten abgearbeitet wird, während die Abarbeitungszeiten von Redis bis zu zwanzig Mal schlechter sind als die von allen anderen Datenbanken, inklusive MongoDB.

Tabelle 21: Ergebnis MoveRandomPackages

MoveRandomPackages							
	Relationale Datenbanken				NoSQL Datenbanken		
n	MySQL	PostgreSQL	Derby	Derby (iM)	MongoDB	Redis (T)	Redis (1s)
2000/5000	877	2252	544	271	676	24708	24555
2000/10000	1676	2904	1221	705	988	35448	34655
2000/20000	2465	3887	2219	2055	1878	49902	47655
2000/40000	3775	7336	4022	3775	3766	80981	82155
2000/80000	5885	12054	12903	11704	8329	120000	120000

4.1.14 Archivierung von Paketen und ihrer Historie (delete)

Bei Betrachtung der Ergebnisse dieses Szenarios in Tabelle 22 fällt insbesondere die starke Diskrepanz in den Abarbeitungszeiten zwischen den einzelnen Vertretern der relationalen Datenbanken auf mit sehr langsamen Ausführungszeiten für Derby, während MongoDB und Redis sich im Mittelfeld eingruppierten.

Tabelle 22: Ergebnis ArchiveRandomPackages

ArchiveRandomPackages							
	Relationale Datenbanken				NoSQL Datenbanken		
n	MySQL	PostgreSQL	Derby	Derby (iM)	MongoDB	Redis (T)	Redis (1s)
2000/5000	576	422	2455	766	704	2807	2331
2000/10000	945	988	3544	1176	957	3974	3790
2000/20000	1484	1602	7154	3155	1602	5443	5464
2000/40000	2806	3321	18971	14565	5441	8705	7992
2000/80000	7129	8122	46721	51228	14429	15871	14376

4.1.15 Load-Test

Abgesehen von PostgreSQL wird dieses Szenario von allen Datenbanken nur sehr langsam ausgeführt. Bei den relationalen Datenbanken ist insbesondere die im Gegensatz zu PostgreSQL bis zu zehnmal langsamere Ausführung durch MySQL hervorhebenswert, der schlechtesten Datenbank in diesem Szenario. Tabelle 23 zeigt außerdem die um den Faktor zwei langsamere Ausführung des Szenarios durch die Key-Value-Datenbank im Vergleich zur dokumentenorientierten MongoDB.

Tabelle 23: Ergebnis LoadTest

LoadTest							
	Relationale Datenbanken				NoSQL Datenbanken		
n	MySQL	PostgreSQL	Derby	Derby (iM)	MongoDB	Redis (T)	Redis (1s)
5000	20655	1803	14144	5990	10566	17906	15671
10000	41211	4052	27099	10854	17605	37529	37801
20000	67605	7225	63794	30886	33127	59443	52766
40000	119226	17122	120000	90224	59223	108598	97373
80000	120000	46502	120000	120000	120000	120000	120000

4.1.16 Batch-Load-Test

Während MySQL und Derby den Batch-Load-Test in Übereinstimmung mit den Ausführungszeiten beim Load-Test sehr langsam ausführen, fallen in Tabelle 24 die im Vergleich signifikant schlechteren Ausführungszeiten für PostgreSQL auf, das in diesem Szenario durchweg schlechter abschneidet als MySQL, sowie die signifikant schnellere Ausführung von MongoDB, das dieses Szenario etwa doppelt so schnell abarbeitet als Derby, die zweit-schnellste Datenbank in diesem Szenario..

Tabelle 24: Ergebnis BatchLoadTest

BatchLoadTest							
	Relationale Datenbanken				NoSQL Datenbanken		
n	MySQL	PostgreSQL	Derby	Derby (iM)	MongoDB	Redis (T)	Redis (1s)
5000	6766	13664	2766	2609	1244	20566	20208
10000	14114	24335	6022	6766	2122	36447	37659
20000	28603	44879	15609	15812	4231	64837	58435
40000	58022	101445	39903	40041	17665	111905	106229
80000	118582	120000	119029	112767	57114	120000	120000

4.2 Interpretation der Ergebnisse in Hinblick auf die Testszenarios

Nach der Darstellung der Ergebnisse der Einzelszenarien dient dieses Kapitel der eingehenderen Deutung der Ergebnisse. Zugrundeliegende Fragen sind, wie die verschiedenen Datenbanken und Klassen von Datenbanken insgesamt abschneiden, in welchen Arten von Szenarien die Stärken und Schwächen einzelner Datenbanken liegen und ob sich die Testergebnisse mit den Erwartungen bezüglich der Leistungsfähigkeit von relationalen und NoSQL-Datenbanken decken. Diese Aussagen werden insbesondere in Kapitel 4.4 Schlussfolgerungen bezüglich der tatsächlichen Eignung verschiedener Datenbanken und Klassen von Datenbanken in verschiedenen Nutzungsszenarien zulassen.

Tabelle 25 zeigt die Gesamtausführungszeit aller Szenarien für jede der Datenbanken im Test. Dazu ist zunächst anzumerken, dass das Verhältnis der Gesamtausführungszeiten tendenziell zu Ungunsten der Datenbanken mit kürzeren Ausführungszeiten dargestellt wird, da für Timeouts, die insbesondere bei den langsamen Datenbanken auftreten, die Maximalausführungszeit von 120s angenommen wird. Zudem ist die Gesamtausführungszeit nicht nach dem in der Realität tatsächlich auftretenden Verhältnis von verschiedenen Aktivitäten gewichtet, beinhaltet allerdings einige Load-, Update-, sowie Delete-Operationen und eine Vielzahl an Abfragen und somit alle relevanten Datenbankoperationen, die auch in der Realität die Mehrheit der Aktionen stellen sollten.

Einerseits lässt sich zwar keine eindeutige Reihenfolge der Datenbanken feststellen, andererseits erlaubt allerdings bereits dieses relativ kleine Datenset, signifikante Unterschiede und Trends in der Ausführungszeit aufzuzeigen: Zunächst schneiden alle relationalen Datenbanken für jede Paketzahl in der Gesamtausführungszeit besser ab als alle NoSQL-Datenbanken, was angesichts des zugrundeliegenden, sehr strukturierten Datenmodells nicht unerwartbar ist.

Aufgrund der Schwächen dieser Analyse ist nicht auszumachen, welcher Vertreter jeder Klasse von Datenbanken am performantesten ist. Die Ergebnisse deuten bei den relationalen Datenbanken jedoch an, dass Derby insbesondere in der in Memory Version und insbe-

sondere bei kleinen Paketzahlen erhebliche Vorteile hat, während die relationalen Vertreter MySQL und PostgreSQL bei großen Paketzahlen am besten abschneiden, ein Ergebnis, das angesichts der vielfachen Verwendung und intensiven Weiterentwicklung dieser etablierten Datenbanken wenig erstaunlich ist. Bei den NoSQL-Datenbanken sind die Unterschiede weniger deutlich, wobei insbesondere bei kleinen Paketzahlen die dokumentenorientierte MongoDB leichte Vorteile vor dem Key-Value-Store von Redis zu haben scheint.

Beim Vergleich der beiden AOF-Konfigurationen von Redis ist, aufgrund des leicht verringerten Aufwandes für die Protokollierung, ein leichter Vorteil in der Ausführungsdauer für die sekundliche Protokollierung vor der transaktionsbasierten Protokollierung erwartbar. Dieser hat sich für vier von fünf Paketzahlen bestätigt, ist angesichts der geringen Differenz allerdings kaum signifikant.

Tabelle 25: Gesamtausführungszeit aller Tests

Gesamtzeit	Relationale Datenbanken				NoSQL Datenbanken		
	MySQL	PostgreSQL	Derby	Derby (iM)	MongoDB	Redis (T)	Redis (1s)
n							
5000	37218	27594	25441	13852	102731	160631	154310
10000	75346	50633	52886	30970	153572	241691	249849
20000	138708	97034	129310	87343	266961	364322	345104
40000	277667	218846	311483	262204	485795	589290	565154
80000	491714	419878	664652	663695	804963	781872	778980

Die Aussagekraft einer Gesamtbetrachtung kann durch die Gewichtung der Ausführungszeiten der Szenarien nach der Häufigkeit ihres Auftretens erhöht werden, da insbesondere seltene Operationen wie die Loadoperationen oft überdurchschnittlich hohe Ausführungszeiten aufweisen. Gemäß Tabelle 26 werden dazu die Testszzenarien nach der angenommenen Ausführungshäufigkeit im Anwendungsfall des Paketdienstleisters in drei Klassen eingeteilt. Während einige Abfragen und insbesondere die Updateoperation *MoveRandomPackages* häufig ausgeführt werden, werden die Loadoperationen vergleichsweise selten ausgeführt. In Ermangelung empirischer Ausführungshäufigkeiten bei einem Paketdienstleister wird angenommen, dass jede der häufigen Aktionen 10% aller Operationen ausmacht, jeder der mittelhäufigen 5% aller Operationen, und jede der selten auftretenden Operationen 2,5% aller Operationen.

Tabelle 26: Abfragen nach Häufigkeit ihres Auftretens

Abfragenhäufigkeit		
Hoch	Mittel	Niedrig
GetRandomPackageTrackingTime	GetAllLocationsTest	BatchLoadTest

GetRandomPackageHistories	ArchiveRandomPackages	LoadTest
GetRandomPackageLocations	GetAllLocationPackageCount	
MoveRandomPackages	GetAllLocationTotalPackageCount	
GetRandomPackages	GetAllPackageHistories	
	GetAllPackageLocations	
	GetPackagesAtAllLocations	
	GetAllMultipleVisitors	

Tabelle 27 illustriert die nach diesen Faktoren gewichteten Ausführungszeiten. Während sich an den Gesamtverhältnissen im Gegensatz zu den unverändert aufsummierten Ausführungszeiten in Tabelle 25 nur wenig ändert, was gegen eine deutliche Verfälschung der akkumulierten Ergebnisse durch die unterschiedliche Häufigkeit der verschiedenen Operationen spricht, manifestiert sich bei näherer Untersuchung insbesondere ein deutlich stärkerer Progressionseffekt der Ausführungszeiten bei den relationalen Datenbanken im Gegensatz zu den NoSQL-Datenbanken: Während bei einfacher Aufsummierung das Verhältnis der Gesamtausführungszeit aller Szenarien für MySQL bei 80.000 Paketen dreizehn Mal höher liegt als bei 5.000 Paketen, liegt es bei Gewichtung der Ausführungszeiten zweiundzwanzig Mal höher. In ähnlichem Ausmaß zeigt sich diese Entwicklung auch für PostgreSQL und Derby, nicht jedoch für MongoDB und Redis. Bei Redis ist der Progressionseffekt insgesamt am schwächsten und für beide – die einfachen wie die gewichteten – Gesamtzeiten bei Faktor fünf.

Dieser deutlich niedrigere Faktor bestärkt die Annahme, dass NoSQL-Datenbanken insgesamt und insbesondere die Key-Value-Datenbank Redis bei deutlich größeren als den getesteten Datenmengen Vorteile gegenüber den relationalen Datenbanken haben können.

Tabelle 27: Gesamtausführungszeit aller Tests gewichtet nach Szenarienhäufigkeit

Gewichtete Gesamtzeit							
n	Relationale Datenbanken				NoSQL Datenbanken		
	MySQL	PostgreSQL	Derby	Derby (iM)	MongoDB	Redis (T)	Redis (1s)
5000	5861	6409	4746	2840	41634	58337	56515
10000	11907	11652	10678	6988	61387	81200	84244
20000	24122	23109	26650	20834	105435	117026	113503
40000	54337	51624	76601	66270	188334	181548	177487
80000	126300	123349	204473	206518	292407	262710	261385

Sowohl die Untersuchung der Ergebnisse auf Szenarioebene, als auch in der Gesamtsicht kann wichtige Ergebnisse verschleiern und ist wenig aussagekräftig, entweder wegen zu großem oder zu geringem Detailgrad. Eine Alternative für eine aussagekräftige Interpretation

der Ergebnisse stellt die Betrachtung der Performance der Datenbanken in bestimmten Szenariogruppen dar. Dazu werden die Szenarien nach der Art der Datenbankoperation gruppiert, was auch den Abgleich von erwarteten mit tatsächlichen Ergebnissen erleichtert, da Annahmen zwar für eine bestimmte Art von Abfragen bestehen, sinnvollerweise allerdings nicht für jeder einzelne Abfrage. Deshalb werden die Ergebnisse in diesem Kapitel in den folgenden Untergliederungen diskutiert:

- Load (4.2.1)
- Update (4.2.2)
- Delete (4.2.3)
- Abfragen nach Komplexität (4.2.4.1)
- Abfragen nach Anzahl der Datensätze (4.2.4.2)

Für jede dieser Gruppen werden nachfolgend beinhaltete Testszzenarien, erwartete Ergebnisse und tatsächliche Ergebnisse beschrieben.

4.2.1 Load

Die Gruppe Load umfasst die Testszzenarien LoadTest sowie BatchLoadTest. Bezüglich der Ausführungszeit besteht für die Loadoperationen keine Hypothese, da weder die relationalen, noch die NoSQL-Ansätze für einen besonders performanten Load bekannt sind. Zudem ist dieses Ereignis im Vergleich etwa zu Abfragen eher selten, weshalb eine besonders hohe Performanz nicht zwingend erforderlich ist.

Auffallend in der Ergebnisbetrachtung in Tabelle 28 ist das verhältnismäßig einheitliche Abschneiden aller getesteten Datenbanken im Gesamtbild, mit leichten Vorteilen für PostgreSQL bei den relationalen und MongoDB bei den NoSQL-Ansätzen. Während Derby den auch bei den anderen Szenarien typischen, progressiven Anstieg in der Ausführungszeit aufweist, fällt trotz der insgesamt deutlich längeren Ausführungszeit zudem der stark unterdurchschnittliche Anstieg der Abarbeitungszeit bei Redis auf. Für noch größere Paketzahlen kann folglich ein noch besseres Abschneiden dieser Key-Value-Datenbank im Vergleich zu den anderen Datenbanken im Test angenommen werden.

Tabelle 28: Summe der Ausführungszeit der Load-Tests

Load							
	Relationale Datenbanken				NoSQL Datenbanken		
n	MySQL	PostgreSQL	Derby	Derby (iM)	MongoDB	Redis (T)	Redis (1s)
5000	27421	15467	16910	8599	11810	38472	35879
10000	55325	28387	33121	17620	19727	73976	75460
20000	96208	52104	79403	46698	37358	124280	111201
40000	177248	118567	159903	130265	76888	220503	203602
80000	238582	166502	239029	232767	177114	240000	240000

4.2.2 Update

Die getestete Update-Operation `MoveRandomPackages`, deren Ergebnisse in Kapitel 4.1.13 aufgeführt sind, besteht zunächst aus einer Abfrage, da die zu ändernden Datensätze aufgefunden werden müssen. Über die Ausführungsdauer des tatsächlichen Update-Vorgangs kann im Vorfeld keine Annahme gemacht werden, das schnelle Auffinden von (zufälligen) Datensätzen sollte allerdings eher einen Vorteil für die unter anderem dafür entwickelte Key-Value-Datenbank Redis sein.

Das Ergebnis dieses Tests zeigt jedoch das exakte Gegenteil, und Redis ist bei der Update-Operation gegenüber allen anderen getesteten Datenbanken weit abgeschlagen. Während MySQL die Operation insgesamt am schnellsten bewerkstelligt, ist insbesondere die im Gegensatz zu Redis relativ schnelle Ausführungszeit von MongoDB bemerkenswert. Dieser Test widerlegt die Annahme der schnelleren Änderbarkeit von Datensätzen in Key-Value-Datenbanken somit deutlich.

4.2.3 Delete

Ähnlich der Änderungsoperation beinhaltet auch die Delete-Operation zunächst das Auffinden der zu löschenden Datensätze, und damit eine zufällige Abfrage, was ebenfalls einen leichten Vorteil für die NoSQL-Ansätze erwarten lassen sollte. Die Ergebnisse der getesteten Löschoperation `ArchiveRandomPackages` sind in Kapitel 4.1.14 dargestellt.

Im Gegensatz zu den unerwarteten Ergebnissen bei der Update-Operation sind die Ausführungszeiten der Löschoperation ausgeglichener zwischen den relationalen und den NoSQL-Datenbanken. Zwar ist Derby insbesondere bei einer großen Datenbankgröße die mit Abstand langsamste Datenbank, jedoch können die etablierten relationalen Datenbanken MySQL und PostgreSQL Redis und MongoDB auch bei dieser Operation signifikant unterbieten.

4.2.4 Abfragen

Bei der Mehrheit der Testszenarien, 12 von 16, handelt es sich um Datenbankabfragen, die also auf keine Manipulation, sondern nur eine Auslese von Daten aus der Datenbank abzielen. Aufgrund der Verschiedenheit von Abfragen – und den unterschiedlichen Anforderungen an eine Datenbank, die sich daraus ergeben – werden die Abfragen auf zweifache Art untergliedert: Nach dem Grad ihrer Komplexität und nach der Anzahl der abgefragten Datensätze.

4.2.4.1 Abfragen nach Komplexität

Als Unterscheidungsmerkmal zwischen komplexen und einfachen Abfragen dient die Anzahl der Relationen, die für die Bewältigung einer Abfrage im relationalen Modell konsultiert wer-

den müssen. Sobald mehr als eine Relation an der Abfrage beteiligt ist, es also mindestens einen Join zwischen Datenbanktabellen gibt, gilt eine Abfrage als komplex. Die zugrunde liegende Frage ist, ob Abfragen, die für relationale Datenbanken einen bestimmten Schwierigkeitsgrad haben, von den NoSQL-Datenbanken schneller bewältigt werden können. Erwartbar ist dabei, dass die relationalen Datenbanken, denen ein sehr strukturiertes Datenmodell zugrundeliegt und die gerade für schwierige, dafür aber selten auftretende Abfragen, optimiert sind, die komplexen Abfragen schneller ausführen, während bei den einfachen insbesondere der Key-Value-Store im Vorteil sein sollte.

Tabelle 29: Komplexe und einfache Abfragen

Komplexe Abfragen	Einfache Abfragen
GetAllPackageHistories	GetAllLocationsTest
GetAllPackageLocations	GetAllLocationPackageCount
GetAllLocationTotalPackageCount	GetAllMultipleVisitors
GetRandomPackageHistories	GetAllPackagesTest
GetRandomPackageLocations	GetRandomPackageTrackingTime
GetPackagesAtAllLocations	GetRandomPackages

Die Ergebnisse der Analyse der Abfragen nach ihrem Komplexitätsgrad kann nicht alle Ergebnishypothesen bestätigen. Zwar bewältigt der Key-Value-Store Redis gemäß Tabelle 30 die einfachen Abfragen tatsächlich deutlich besser als Derby und MongoDB, ist allerdings für jede Paketzahl durchweg signifikant schlechter als MySQL und PostgreSQL. Es scheint bei einfachen Abfragen also keinen Vorteil der NoSQL-Datenbanken zu bestehen.

Die Abarbeitungszeiten der komplexen Abfragen, illustriert in Tabelle 30, ist zunächst einmal erwartungsgemäß signifikant höher als die der einfachen Abfragen. Ebenfalls bestätigt sich die Hypothese, dass die relationalen Datenbanken durchweg besser abschneiden als MongoDB und Redis, wobei zwischen den NoSQL-Datenbanken kein signifikanter Unterschied festzustellen ist. Auffällig ist der degressive, stark unterdurchschnittliche Anstieg der Ausführungszeit der NoSQL-Datenbanken bei höheren Paketzahlen im Gegensatz zu den relationalen Datenbanken, der im Wesentlichen jedoch auf den Degressionseffekt durch die Deckelung der Ausführungszeit auf 120 Sekunden im Falle eines Timeouts zurückgeführt werden kann. Die Hypothese bezüglich komplexer Abfragen kann folglich bestätigt werden, relationale Datenbanken sind in diesem Fall angesichts einer gerade bei kleinen Paketzahlen bis zu zehnmal schlechteren Ausführungszeit der NoSQL-Datenbanken diesen deutlich überlegen..

Tabelle 30: Ergebnis einfacher Abfragen

Einfache Abfragen							
n	Relationale Datenbanken				NoSQL Datenbanken		
	MySQL	PostgreSQL	Derby	Derby (iM)	MongoDB	Redis (T)	Redis (1s)
5000	795	1500	1750	998	20712	4360	4430
10000	1477	2137	4351	3423	40627	6801	7558
20000	2855	3618	11369	10651	84486	11603	11709
40000	7000	8170	40172	33297	194871	27725	25047
80000	21058	22477	122520	130184	274463	62867	61846

Tabelle 31: Ergebnis komplexer Abfragen

Komplexe Abfragen							
n	Relationale Datenbanken				NoSQL Datenbanken		
	MySQL	PostgreSQL	Derby	Derby (iM)	MongoDB	Redis (T)	Redis (1s)
5000	7549	7953	3782	3218	68829	90284	87115
10000	15923	16217	10649	8046	91273	121492	128386
20000	35696	35823	29165	24784	141637	173094	169075
40000	86838	81452	88415	80302	204829	251376	246358
80000	219060	210723	243479	237812	330628	343134	342758

4.2.4.2 Abfragen nach Anzahl der Datensätze

Der zweite signifikante Unterschied in den Abfrageszenarien besteht darin, dass einige Szenarien eine Abfrage sämtlicher Datensätze in der Datenbank erfordern, während für andere Szenarien nur bestimmte, (zufällig) ausgewählte Datensätze abgefragt werden. Nachfolgend werden erstere als Komplettabfragen, letztere als Zufalls- oder Teilabfragen bezeichnet. Während zur Ausführungszeit bei Komplettabfragen keine Hypothese getroffen werden kann, ist eine deutlich geringere Ausführungszeit der NoSQL-Datenbanken bei den Zufallsabfragen erwartbar, da gerade das schnelle Auffinden von Datensätzen in großen Datenmengen einer der Hauptzwecke für die Entwicklung von NoSQL-Datenbanken, insbesondere von Key-Value-Datenbanken ist und deshalb auch eine Stärke dieser Art von Datenbanken sein sollte.

Tabelle 32: Komplet- und Zufallsabfragen

Zufalls-/Teilabfragen	Komplettabfragen
GetRandomPackageTrackingTime	GetAllLocationsTest
GetRandomPackages	GetAllLocationPackageCount
GetAllLocationTotalPackageCount	GetAllMultipleVisitors
GetRandomPackageHistories	GetAllPackagesTest
GetRandomPackageLocations	GetAllPackageHistories
GetPackagesAtAllLocations	GetAllPackageLocations

Die Ergebnisse der Tests können die Annahme in keiner Weise bestätigen. Während bei den Ausführungszeiten der Komplettabfragen in Tabelle 33 kein eindeutiger Trend erkennbar ist und sowohl MySQL und PostgreSQL, als auch Redis überdurchschnittlich gute Ergebnisse erzielen können, schneiden gemäß der Ergebnisse in Tabelle 34 die NoSQL-Datenbanken bei den Zufallsabfragen signifikant schlechter ab. Zwar sind die Zufallstests tendenziell schneller als die Kompletttests, was sich wohl vor allem auf die geringere Anzahl an abgefragten Datensätzen zurückführen lässt, dieser Effekt zeigt sich bei den relationalen Datenbanken jedoch deutlich stärker. Weitere Untersuchungen zeigen, dass insbesondere die erheblich überdurchschnittliche Ausführungszeit des Szenarios *GetAllLocationTotalPackageCount* das Ergebnis der Zufallsabfragen zu Ungunsten der NoSQL-Datenbanken verschiebt. Jedoch bleiben auch bei Berichtigung um dieses Szenario signifikant schlechtere Ausführungszeiten.

Tabelle 33: Ergebnis Komplettabfragen

Komplettabfragen							
n	Relationale Datenbanken				NoSQL Datenbanken		
	MySQL	PostgreSQL	Derby	Derby (iM)	MongoDB	Redis (T)	Redis (1s)
5000	7483	6527	4559	3307	27910	11787	11449
10000	16279	14541	13130	9793	57129	23798	23781
20000	37057	34700	37261	32362	120851	50451	50328
40000	91500	80192	122744	107952	273670	127277	121470
80000	236193	216638	354439	355996	472355	239616	240034

Tabelle 34: Ergebnis Zufalls-/Teilabfragen

Zufalls-/Teilabfragen							
n	Relationale Datenbanken				NoSQL Datenbanken		
	MySQL	PostgreSQL	Derby	Derby (iM)	MongoDB	Redis (T)	Redis (1s)
2000/5000	861	2926	973	909	61631	82857	80096
2000/10000	1121	3813	1870	1676	74771	104495	112163
2000/20000	1494	4741	3273	3073	105272	134246	130456
2000/40000	2338	9430	5843	5647	126030	151824	149935
2000/80000	3925	16562	11560	12000	132736	166385	164570

4.3 Verlässlichkeit und Aussagekraft der Ergebnisse

Die in dieser Arbeit durchgeführten Tests geben nur einen ersten Eindruck von der Leistungsfähigkeit ausgewählter relationaler und NoSQL-Datenbanken und erheben keinen Anspruch auf Vollständigkeit. Eine Verallgemeinerung der Ergebnisse ist aufgrund der beschränkten Testumgebung nur bedingt möglich. Es existieren zahlreiche Unsicherheitsfaktoren und weitere interessante Testszenarien können aufgrund der begrenzten Ressourcen nicht geprüft werden. Dieser Abschnitt soll einen Einblick in mögliche Fehlerquellen und Unsicherheiten während der Analyse geben und diese gegebenenfalls diskutieren.

Wie bereits im vorangehenden Kapitel aufgezeigt werden konnte, entsprechen zahlreiche Testergebnisse nicht den Erwartungen, insbesondere im Falle der angenommenen Vorteile der Key-Value- und NoSQL-Ansätze. Würden unerwartete Werte nur in einigen Messungen auftreten, könnte dies ein Indiz für mangelnde Plausibilität der Messergebnisse insgesamt sein; die ermittelte Tendenz, dass die NoSQL-Ansätze weniger performant sind, tritt jedoch konsistent in mehreren Messungen auf, sodass die Ergebnisse signifikant zu sein scheinen.

Dennoch gibt es bei einzelnen Messungen trotz der redundanten, dreifachen Testausführung immer wieder Ausreißerwerte, sodass entgegen dem allgemeinen Trend eine Datenbank ein bestimmtes Szenario für eine bestimmte Paketzahl besonders schnell oder langsam abarbeitet. Dies kann mehrere Ursachen haben, die zwar einer Nachbesserung bedürfen, jedoch alle die Validität der sonstigen Testergebnisse nicht in Frage stellen. Erstens können solche Ausreißerwerte auf eine instabile Leistungsfähigkeit eines Datenbankprozesses hindeuten, also der jeweiligen Datenbank inhärent sein. Dann wäre das Messergebnis vollkommen legitim und würde eine Schwäche der jeweiligen Datenbank aufzeigen. Zudem kann ein Ausreißerwert in der Konfiguration der Testumgebung und Datenbanken sowie des Datenmodells begründet sein, etwa wenn durch die Auswahl eines bestimmten Parameters eine bestimmte Datenbank in einem bestimmten Szenario bevorzugt ist. Eine solche Verschiebung der Testergebnisse ist nur sehr schwer zu ermitteln. Schließlich kann die instabile Ausführung der Datenbanken auf der Testhardware zu Ausreißerwerten führen, etwa wenn ein anderer Prozess während eines Testlaufs die Computing Processing Unit (CPU) sehr unregelmäßig beansprucht. Dies könnte nur durch eine bessere Abschottung der Testumgebung erreicht werden, etwa durch Ausführung jeder Datenbank auf dezidiert Hardware.

Bei den Testergebnissen ist der oft fast lineare Anstieg der Ausführungszeiten auffällig. Dies kann dadurch begründet werden, dass die Hardware mit zunehmend größerer Paketanzahl dennoch kaum stärker belastet wird, sondern im Wesentlichen nur länger – schließlich werden die Anfragen seriell abgearbeitet, was bei einer größeren Datenbank länger dauert. Dies könnte geändert werden, indem mehrere Anfragen den Datenbankserver gleichzeitig erreichen, indem also mehrere Clients simuliert würden.

Bezüglich der Testszenarien ist festzustellen, dass ein sehr strukturiertes Datenmodell in Form eines Paketdienstleisters angewendet wurde. Für strukturierte Datenmodelle sind jedoch eher die relationalen Datenbanken optimiert, nicht aber die NoSQL-Ansätze, sodass dadurch mitunter eine strukturelle Benachteiligung der NoSQL-Datenbanken gegeben sein könnte. Allerdings ist eine der zugrundeliegenden Fragestellungen dieser Arbeit, inwiefern NoSQL-Datenbanken eine adäquate, vielleicht sogar bessere Alternative zu relationalen Datenbanken darstellen können – gerade auch in der Domäne der relationalen Datenbanken.

Bezüglich der Ergebnisse für Redis könnte schließlich die Größe des AOF ein verfälschender Faktor sein, tendenziell zu Ungunsten von Redis. Für die meisten Szenarien weichen die Ergebnisse für Redis allerdings deutlich von denen anderer Datenbanken ab. Es ist einerseits unwahrscheinlich, dass für eine derart starke Abweichung allein die AOF-Konfiguration verantwortlich ist. Dies würde andererseits an der Aussage, dass Redis im jeweiligen Szenario deutlich besser oder schlechter abschneidet, nichts ändern, da zur Erreichung von Persistenz die Protokollierung auch im Produktionsbetrieb notwendig ist.

4.4 Schlussfolgerung

Dieses Kapitel soll neben einer kurzen Zusammenfassung relevanter Ergebnisse vor allem die Fragen beantworten, inwiefern zusätzliche Tests weitere Aussagen über die Leistungsfähigkeit der verschiedenen Datenbanken und Klassen von Datenbanken zulassen können, und, basierend auf den Testergebnissen, nach welchen Kriterien eine Datenbank für eine bestimmte Aufgabe ausgewählt werden sollte.

Zusammenfassend deuten die Testergebnisse zwar einige Stärken und Schwächen der einzelnen Datenbanken an, lassen jedoch angesichts der im vorigen Kapitel besprochenen Limitierungen der Testdurchführung und –umgebung kein abschließendes Urteil zu. Was die Testergebnisse allerdings insgesamt andeuten, ist, dass die NoSQL-Datenbanken in Form der dokumentenorientierten MongoDB und der Key-Value-Datenbank Redis nicht nur die angenommenen Vorteile nicht ausspielen können, sondern vielmehr in den meisten Szenarien den relationalen Datenbanken sogar unterlegen sind. Dies tritt insbesondere bei der Untersuchung der Zufalls- und Teilabfragen sehr deutlich hervor. Man muss den NoSQL-Ansätzen zwar zu Gute halten, dass sie deutlich neuer sind und auch (noch) deutlich seltener zum Einsatz kommen also die etablierten relationalen Vertreter MySQL und PostgreSQL, die bereits einen viel höheren Reifegrad erreichen konnten. Es kann aus den Ergebnissen also nicht gefolgert werden, dass NoSQL-Ansätze grundsätzlich weniger performant sind, da sie mitunter noch ein deutlich größeres Optimierungs- und Weiterentwicklungspotential bergen im Gegensatz zu den relationalen Ansätzen. Jedoch können weitere Optimierungsmöglichkeiten auch bei den relationalen Datenbanken bestehen, etwa durch Denormalisierung.

Es hat sich während der Testdurchführung gezeigt, dass die Testumgebung einige Schwächen aufweist, die durch zusätzliche Leistungstests in Zukunft behoben werden könnten. So sollten etwa nicht nur serielle, sondern auch parallele Zugriffe auf die Datenbanken berücksichtigt werden.

Schließlich stellt sich die Frage, inwiefern die in dieser Arbeit ermittelten Testergebnisse Aussagen über die Auswahlkriterien und Einsatzbereiche der verschiedenen Datenbanken zulassen. Zunächst legen die Ergebnisse nahe, dass gerade in einem sehr strukturierten,

festgelegten Datenmodell wie etwa bei einem Paketdienstleister, das recht wenig Flexibilität erfordert, relationale Datenbanken für die meisten Nutzungsszenarien und insgesamt die besseren Ausführungszeiten liefern. Diese Aussage lässt sich angesichts der Beschränkungen während der Testdurchführung nicht verallgemeinern und ist sehr stark vom gewählten Szenario abhängig. Zumindest für das in dieser Arbeit untersuchte Modell eines Paketdienstleisters scheinen die untersuchten relationalen Datenbanken der Key-Value-Datenbank Redis und der dokumentenorientierten MongoDB allerdings überlegen zu sein.

Gerade angesichts der zahlreichen Unwägbarkeiten in Testdurchführung und -ergebnissen, sowie der Erfahrungen bei der Implementierung des Testszenarios, sollte die Entscheidung für oder gegen eine Datenbank jedoch weniger auf der in einem – mitunter nicht übertragbaren – Szenario ermittelten Leistungsfähigkeit beruhen, sondern vielmehr den konkreten Zweck der Datenbankanwendung berücksichtigen. Während sich für stark strukturierte Daten eher die Möglichkeiten zur Gliederung im relationalen Modell anbieten, können NoSQL-Datenbanken für unstrukturierte Daten Vorteile bieten. Ein zweites zentrales Kriterium sollte der Entwicklungs- und Unterhaltungsaufwand einer Datenbank sein, der etwaige Leistungsvorteile im Betrieb schnell zunichte machen kann.

5 Schluss

(Christian Brummer, Michael Busam, Thorsten Kühner, Klarissa Schmidt, Matthias Welz)

Dieser Abschnitt fasst noch einmal die gesammelten Erkenntnisse zusammen, betrachtet das Vorgehen dieser Arbeit kritisch und gibt zuletzt einen Ausblick in diesem Themengebiet.

5.1 Zusammenfassung

Ausgehend von der zunehmenden Marktetablierung alternativer Konzepte zu relationalen Datenbanken wurde im Rahmen dieser Arbeit empirisch die Leistungsfähigkeit der Key-Value Datenbank Redis sowie der dokumentenorientierten Datenbank MongoDB, die beide zu den Vertretern von NoSQL-Konzepten gehören, untersucht. Der möglichst reale Anwendungsfall eines Paketdienstleisters bietet eine Auswahl verschiedener Testszenarien, sodass für die spezifischen Datenbanken tatsächliche Nutzungsszenarien konzeptioniert werden konnten. Dabei wurde bei der Konzeption der sechzehn Testszenarien darauf geachtet, alle Arten von Datenbankoperationen zu berücksichtigen, sodass im Ergebnis eine Unterteilung in Load-, Update-, Delete- sowie komplexen und einfachen Abfrageoperationen möglich ist. Die technische Implementierung in Form einer Java-Anwendung führt alle konzeptionierten Testszenarien automatisch für alle Datenbanken durch und misst die entsprechenden Ausführungszeiten. Die Datenbanken enthalten dabei realitätsnahe Daten aus einem im Rahmen dieser Arbeit entwickelten automatischen Testdatengenerator.

Nach der Durchführung und Auswertung der Messungen wurde eine umfangreiche, aber nicht abschließende Aussage zur Leistungsfähigkeit der getesteten Datenbanken und Klassen von Datenbanken getroffen: Die Testergebnisse skizzieren einige Stärken und Schwächen der Datenbanken, lassen sich aber aufgrund von ressourcenbedingten Limitierungen nicht abschließend beurteilen. Zwar zeigt sich, dass die dokumentenorientierten Datenbank MongoDB und die Key-Value-Datenbank Redis hinter den angenommenen Vorteilen und Erwartungen gegenüber den relationalen Datenbankvertretern zurückbleibt, was allerdings unter Beachtung der viel höheren Entwicklungsreife relationaler Datenbanken nicht zur Schlussfolgerung führt, NoSQL-Ansätze seien grundsätzlich weniger performant. Dies muss auch bei der Folgefragestellung, bei der Bewertung der Testergebnisse als Auswahlkriterium für eine Datenbank, berücksichtigt werden: Die konzeptionierten Szenarien und das stark strukturierte Datenmodell eines Paketdienstleisters, lässt sich nur beschränkt auf andere Anwendungsfälle übertragen, sodass die ermittelte Leistungsfähigkeit nicht verallgemeinert werden kann und die Auswahl einer Datenbankanwendung stark unter dem spezifischen Anwendungszweck getroffen werden sollte.

5.2 Kritische Betrachtung

Dieses Kapitel setzt sich mit der in dieser Arbeit gewählten Vorgehensweise kritisch auseinander und gibt Empfehlungen für ein mögliches Weiterentwicklungspotential in künftigen Arbeiten.

Die technische Umsetzung der Testumgebung weist aufgrund begrenzter Ressourcen einige Unsicherheitsfaktoren auf, sodass nur ausgewählte Testszenarien gemessen werden konnten. Auch bei mehrfachen Ausführungen dieser Messungen traten immer wieder Ausreißerwerte auf, die auf instabile Datenbankprozesse, der Parameterkonfiguration der Testumgebung und die Testhardware selbst zurückzuführen sind. Der Einsatz dedizierter Hardware bietet einen Ansatz, um die Testumgebung möglichst isoliert betreiben zu können. Gleichzeitig könnte der ermittelte lineare Anstieg der Ausführungszeiten auf das serielle Abarbeiten der Datenbankabfragen zurückzuführen sein und durch den Einsatz mehrerer Clients und parallelen Anfragen beeinflusst werden. Die Messung mit nur einem Client ermöglicht keine Vollaustlastung auf Seiten der Datenbank, da entweder die Datenbank oder der Client aktiv sind. Unter Verwendung eines Parallelbetriebs muss die Rechenleistung gleichmäßig auf mehrere Clients verteilt werden, was die Messung der tatsächlichen Limits der Datenbank ermöglichen würde. Darüberhinaus ist festzuhalten, dass die Aussagekraft künstlicher Messungen unter Laborbedingungen generell hinterfragt werden müssen, da in der Praxis Funktionen wie das Caching von DB Statements eine große Rolle spielen.

Aufgrund der noch überschaubaren und greifbaren Komplexität fiel die Wahl des Anwendungsfalles auf einen Paketdienstleisters und hat in der Folge zur Konzeption von Testszenarien mit einem stark strukturierten Datenmodell geführt. Da hierfür ursprünglich relationale Datenbanken entwickelt wurden, könnte dies eine gewisse Benachteiligung der Key-Value Datenbank Redis bzw. der dokumentenorientierten Datenbank MongoDB nach sich ziehen. Da die Zielsetzung dieser Arbeit eine Untersuchung von NoSQL-Ansätzen als adäquate Alternative zu den etablierten relationalen Datenbanken vorsieht, ist die Verwendung strukturierter Daten jedoch gut vertretbar. Die Evaluierung der Leistungsfähigkeit auf Basis eines anderen Anwendungsfalles mit unstrukturierten Daten als Grundlage könnte eine interessante Erweiterung dieser Arbeit darstellen.

Da bei der Entwicklung der Java-Anwendung für die Testdurchführung, wie auch des Testdatengenerators für die Befüllung der Datenbanken, besonders auf einen modularen Aufbau geachtet wurde, eignen sich die Ergebnisse dieser Arbeit für die Konzeption weiterer modifizierter Szenarien, die über die im Rahmen dieser Arbeit getesteten Szenarien hinausgehen. Somit ist es möglich, den Anwendungsfall des Paketdienstleisters zu erweitern und auf ande-

ren Datenbanksystemen zu testen, sodass der Fokus zukünftiger Arbeiten in dieser Thematik noch stärker auf die zugrunde liegenden Szenarien gerückt werden kann.

Die zu Anfang dieser Arbeit durchgeführte Auseinandersetzung mit den theoretischen Grundlagen zu relationalen und Key-Value Datenbanken bot einen wichtigen grundlegenden Einstieg in die Thematik. Die eingehende theoretische Beschäftigung war äußerst hilfreich für die Konzeption und Umsetzung dieser Arbeit, sodass auf Basis der Literaturrecherche vor allem im Hinblick auf die junge, aufstrebende NoSQL-Technologie Erwartungen geweckt wurden, die jedoch teilweise im Laufe der Arbeit widerlegt werden konnten. Die Key-Value Datenbankvertreter konnten im Hinblick auf die Leistungsmessungen nicht überzeugen, sondern hatten eher im Gegenteil das Nachsehen gegenüber den relationalen Datenbanken. Dies wird im folgenden Ausblick, Kapitel 5.3, noch einmal unter der Fragestellung, ob eine Datenbankkategorie unter dem Leistungskriterium zu empfehlen ist, aufgegriffen.

5.3 Ausblick

Das zentrale Ziel dieser Arbeit war die Leistungsfähigkeit verschiedener relationaler Datenbanken mit derjenigen einer Key-Value Datenbank auf der Basis verschiedener Testszenarien zu vergleichen. Dies führt zur Folgefragestellung, ob daraus eine Empfehlung für die zu bevorzugende Wahl einer Datenbankkategorie ausgesprochen werden kann.

Aus Sicht der in dieser Arbeit durchgeführten Performance-Tests für den Anwendungsfall eines Paketdienstleisters sind Key-Value Datenbanken nicht empfehlenswert. Allerdings muss dabei deutlich gemacht werden, dass dies nicht zu Folge hat, dass relationalen Datenbanken die grundsätzlich bessere Wahl im Vergleich zu Key-Value und andern NoSQL Vertretern sind. Im Hinblick auf die Entscheidung zum Einsatz einer Datenbank gilt es, weitere Faktoren zu berücksichtigen, worunter nach Ansicht der Autoren die Performance nur ein zweitrangiges Kriterium darstellt. Im Kontext von „premature optimization is the root of all evil“¹⁰⁵ sollten Bemühungen zu Leistungsverbesserungen nicht zu Beginn der Entwicklung im Mittelpunkt stehen, sondern erst im späteren Verlauf angegangen werden, um die Verständlichkeit, Pflege und Wartung von Programmen und Systemen nicht bereits von Anfang an durch zusätzlichen Code einzuschränken.

Im Vergleich relationaler mit Key-Value Datenbanken treten vor allem aufgrund des unterschiedlichen Entwicklungsstands entscheidende Merkmale hervor: Relationale Datenbanken sind weitestgehend standardisiert, umfangreich dokumentiert und weit verbreitet. Der Kenntnisstand bei Entwicklern ist verhältnismäßig hoch und ein Fortbestand der Technologie ist durch den breiten Nutzerkreis sehr sicher. Da in der Vergangenheit viel Zeit und Geld in die

¹⁰⁵ Knuth, D. (1974), S.268

die Entwicklung geflossen ist, gelten relationale Datenbanken als erprobte und zuverlässige Technologie. Key-Value Datenbanken als Variante von NoSQL-Konzepten haben diesen Entwicklungsstand noch nicht erreicht. Sie bieten keinen gemeinsamen Standard, sondern präsentieren sich sehr individuell, sodass es eine Vielzahl von verschiedenen Herstellern mit unterschiedlichen Schwerpunkt gibt. Zwar wird häufig mit hoher Flexibilität geworben, sodass der Aufwand zum Entwicklungsbeginn vergleichsweise gering ist und kurzfristige Vorteile gesichert sind. Allerdings ist dies mit dem Risiko langfristiger Nachteile verbunden, da beispielsweise eine Abhängigkeit vom Hersteller vorliegt, der die fortlaufende Unterstützung seiner Technologie nicht garantiert. Im Hinblick auf das Leistungspotential der Systeme sind für relationale Datenbanken verschiedene Technologien im Einsatz, die durch ausgereifte Algorithmen automatisch Leistungsoptimierungen vornehmen, wohingegen dies bei Key-Value Datenbanken noch unter erhöhtem Arbeitsaufwand manuell erfolgen muss.

Da Key-Value Datenbanken noch am Anfang ihrer Entwicklung stehen, ist es nicht überraschend, dass in verschiedenen Aspekten noch Nachholbedarf besteht. Allerdings sind sie bereits teilweise etablierten Technologien überlegen, sodass das vorhandene Potential in Zukunft mit Spannung verfolgt werden sollte und künftige Entwicklungen als mögliche Alternative zu jetzigen Lösungen kritisch untersucht werden sollten.

Quellenverzeichnisse

1 Literaturverzeichnis

- Aho, A. V. / Ullman, J. D. (1994):* Foundations of Computer Science: C Edition, W. H. Freeman
- Beaulieu, A. (2009):* Einführung in SQL, 2. Aufl., Köln: O'Reilly Verlag
- Celko, J. (2013):* Joe Celko's complete guide to NoSQL, What every SQL professional needs to know about non-relational databases, Waltham: Morgan Kaufmann
- Chang, F. / u.a. (2006):* Bigtable: A Distributed Storage System for Structured Data, Google
- Codd, E. F. (2002):* A Relational Model of Data for Large Shared Data Banks, in: Software Pioneers, Berlin, Heidelberg: Springer Verlag, S.263-294
- Edlich, S. / u.a. (2010):* Einstieg in die Welt nichtrelationaler Web-2.0-Datenbanken, München: Hanser
- Elmasri, R. / Navathe, S. B. (2011):* Database System - Models, languages, design, and application programming, 6. Aufl., Boston: Pearson
- Fowler, M. / Sadalage, P. (2012):* NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Boston/Mass/London: Addison-Wesley
- Hawkins, T. / Membrey, P. / Plugge, E. (2010):* The definitive guide to MongoDB, w.l.: Apress
- Kemper, A. / Eickler, A. (2011):* Datenbanksystem - Eine Einführung, 8. Aufl., München: Oldenbourg Verlag
- Klein, D. / Tran-Gia, P. / Hartmann, M. (2013):* Big Data, in: Informatik-Spektrum, Vol. 36, Nr. 3, w.l.: Springer, p. 319-323
- Kleuker, S. (2011):* Grundkurs Datenbankentwicklung - Von der Anforderungsanalyse zur komplexen Datenbankanfrage, 2. Aufl., Wiesbaden: Vieweg+Teubner Verlag (Springer Fachmedien)
- Knuth, D. (1974):* StructuredProgrammingWithGoToStatements, in: Computing Surveys, Aufl. 6, Nr. 4, S.261-301
- Redmond, E. / Wilson, J. (2012):* Sieben Wochen, sieben Datenbanken, Köln: O'Reilly
- Schneider, U. (Hrsg.) (2012):* Taschenbuch der Informatik, 7. Aufl., München: Carl Hanser Verlag
- Sumathi, S. / Esakkirajan, S. (2007):* Fundamentals of Relational Database Management Systems
- Tiwari, S. (2011):* Professional NoSQL, Indianapolis: John Wiley & Sons
- Trivedi, V. (2014):* How to speak Tech, The non-techie's guide to technology basics in business, New York: Apress
- Unterreiner, V. (2009):* Oracle kauft Sun für rund 7,4 Milliarden Dollar, in: Die Welt Wirtschaft, am 20.04.2009

Unterstein, M. / Matthiessen, G. (2012): Relationale Datenbanken und SQL in Theorie und Praxis, 5. Aufl., Berlin, Heidelberg: Springer Vieweg

2 Internet

Amazon (Hrsg.) (o.J.): Amazon DynamoDB, Abruf: 02.07.2014

<http://aws.amazon.com/de/dynamodb/>

Apache DB (o.J.): Apache Derby, Abruf: 12.06.2014

<http://db.apache.org/derby/index.html>

Cassandra (o.J.): Cassandra Apache Project Website, Abruf: 02.07.2014

<http://cassandra.apache.org/>

Claburn, T. (2007): Google Releases Improved MySQL Code, Abruf: 12.06.2014

<http://www.informationweek.com/google-releases-improved-mysql-code/d/d-id/1054428?>

Cole, J. (2011): Big and Small Data at Twitter: MySQL CE 2011, Abruf: 12.06.2014

<http://nosql.mypopescu.com/post/4687379038/big-and-small-data-at-twitter-mysql-ce-2011>

DB-Engines (o.J.): DB-Engines Ranking von Key-Value Stores, Abruf: 02.07.2014

<http://db-engines.com/de/ranking/key-value+store>

Feinberg, D. / Adrian M. / Heudecker, N. (2013): Magic Quadrant for Operational Database Management Systems, Abruf: 04.06.2014

<http://www.gartner.com/technology/reprints.do?id=1-1M9YEHW&ct=131028&st=sb#>

FH Köln (2013): Key/Value-Datenbanksysteme, Abruf: 28.06.2014

http://wikis.gm.fh-koeln.de/wiki_db/Datenbanken/KeyValueSysteme

Gantz, J. / Reinsel, D. (2012): The Digital Universe 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East, Abruf: 04.06.2014

<http://idcdocserv.com/1414>

Grolinger, K. / u.a. (2013): Data management in cloud environments: NoSQL and NewSQL data stores, Abruf: 04.06.2014

<http://www.journalofcloudcomputing.com/content/pdf/2192-113X-2-22.pdf>

Heudecker, N. (2013): Operational Database Management Systems. Interview with Nick Heudecker, Abruf: 04.06.2014

<http://www.odbms.org/blog/2013/12/operational-database-management-systems-interview-with-nick-heudecker/>

Janssen, C. (o.J.): NoSQL, Abruf: 28.06.2014

[http://www.techopedia.com/definition/27689/nosql-database,](http://www.techopedia.com/definition/27689/nosql-database)

Jansen, R. (2011): NoSQL: Key-Value-Datenbank Redis im Überblick, Abruf: 28.06.14

<http://www.heise.de/developer/artikel/NoSQL-Key-Value-Datenbank-Redis-im-Ueberblick-1233843.html>

- Jedis (Hrsg.)* (o.J.): jedis - A blazingly small and sane redis java client, Abruf: 23.06.2014
<https://code.google.com/p/jedis/>
- jQuery (Hrsg.)* (o.J.): jQuery, write less, do more, Abruf: 23.06.2014
<https://jquery.com/>
- Lai, E. (2010)*: Twitter growth prompts switch from MySQL to 'NoSQL' database, Abruf: 28.06.2014
http://www.computerworld.com/s/article/9161078/Twitter_growth_prompts_switch_from_MySQL_to_NoSQL_database
- MongoDB (2014a)*: NoSQL Databases Explained, Abruf: 28.06.2014
<http://www.mongodb.com/nosql-explained>,
- MongoDB (2014b)*: Glossary, Abruf: 02.07.2014
<http://docs.mongodb.org/manual/reference/glossary/#term-collection>
- MongoDB (2014c)*: Data Modeling Introduction, Abruf: 02.07.2014
<http://docs.mongodb.org/manual/core/data-modeling-introduction/>,
- MongoDB (2014d)*: Model Tree Structures, Abruf: 02.07.2014
<http://docs.mongodb.org/manual/applications/data-models-tree-structures/>,
- MongoDB (2014e)*: Data Model Design, Abruf: 02.07.2014
<http://docs.mongodb.org/manual/core/data-model-design/>
- O'Reilly (Hrsg.) (2007)*: YouTube, Flickr, and Wikipedia to Share their Secrets of Success at the 2007 MySQL Conference & Expo, Abruf: 12.06.2014
<http://www.oreilly.com/pub/pr/1712>
- Oracle (Hrsg.)* (o.J.): History of MySQL, Abruf: 12.06.2014
<http://dev.mysql.com/doc/refman/5.1/en/history.html>
- Oracle (Hrsg.)* (o.J. a): MySQL Customers by Industry : Web: Social Networks, Abruf: 12.06.2014
<http://www.mysql.com/customers/industry/?id=85>
- Oracle (Hrsg.)* (o.J. b): JavaDB, Abruf: 12.06.2014
<http://www.oracle.com/technetwork/java/javadb/overview/index.html>
- Oracle (Hrsg.)* (o.J. c): Oracle Berkeley DB 12c, Abruf: 02.07.2014
<http://www.oracle.com/technetwork/database/database-technologies/berkeleydb/overview/index.html>
- Oracle (Hrsg.) (2008)*: Marktanteil nach Gartner Studie, Abruf: 12.06.2014
<http://www.mysql.de/why-mysql/marketshare/>
- Pearce, R. (2013)*: Dead database walking: MySQL's creator on why the future belongs to MariaDB, Abruf: 12.06.2014
http://www.computerworld.com.au/article/457551/dead_database_walking_mysql_creator_why_future_belongs_mariadb/
- PostgreSQL (Hrsg.)* (o.J.): PostgreSQL About, Abruf: 12.06.2014
<http://www.postgresql.org/about/>

- PostgreSQL (Hrsg.) (o.J. a):* PostgreSQL Feature Matrix, Abruf: 12.06.2014
<http://www.postgresql.org/about/featurematrix/>
- PostgreSQL (Hrsg.) (o.J. b):* PostgreSQL GiST Manual Introduction, Abruf: 12.06.2014
<http://www.postgresql.org/docs/9.1/static/gist-intro.html>
- PostgreSQL (Hrsg.) (o.J. c):* PostgreSQL Awards, Abruf: 12.06.2014
<http://www.postgresql.org/about/awards/>
- Redis (Hrsg.) (o.J.):* Tutorial: Design and implementation of a simple Twitter clone using PHP and the Redis key-value store, Abruf: 23.06.2014
<http://redis.io/topics/twitter-clone>
- Redis (Hrsg.) (o.J. b):* Redis Project Website, Abruf: 02.07.2014
<http://redis.io>
- Redis (Hrsg.) (o.J. c):* How fast is Redis?, Abruf: 02.07.2014
<http://redis.io/topics/benchmarks>
- Meetup (Hrsg.) (o.J.):* San Francisco NoSQL Meetup Group, Abruf: 02.07.2014
<http://www.meetup.com/SFNoSQL/>
- Sequin, K. (2012):* Data Modeling In Redis, Abruf: 02.07.14
<http://openmymind.net/Data-Modeling-In-Redis/>,
- Schmitz, L. (2013):* "Die Zukunft von MySQL gestalten wir", Abruf: 12.06.2014
<http://www.computerwoche.de/a/die-zukunft-von-mysql-gestalten-wir,2533344,2#>
- Sobel, J. (2007):* Keeping Up, Abruf: 12.06.2014
<https://www.facebook.com/notes/facebook/keeping-up/7899307130>
- Walker-Morgen, D. (2010):* NoSQL im Überblick, Abruf: 28.06.2014
<http://heise.de/-1012483>
- Wikipedia (2014 a):* Liste deutscher Vornamen germanischer Herkunft, Abruf: 23.06.2014
http://de.wikipedia.org/wiki/Liste_deutscher_Vornamen_germanischer_Herkunft
- Wikipedia (2014 b):* Liste der häufigsten Familiennamen in Deutschland, Abruf: 23.06.2014
http://de.wikipedia.org/wiki/Liste_der_h%C3%A4ufigsten_Familiennamen_in_Deutschland

Methodik/Systematik einer Beratungsdienstleistung für das Open Source Beratungszentrum Baden-Württemberg

Schriftliche Ausarbeitung
im Rahmen der Lehrveranstaltung "Integrationsseminar"
Kompetenzzentrum Open Source (KOS)

Vorgelegt von

Maurice Goerges
Manuel Meyer
David Pohlmann
Thomas Scherer

am 11.07.2014

Fakultät Wirtschaft
Wirtschaftsinformatik
Kurs WWI2011I

Inhaltsverzeichnis

Inhaltsverzeichnis.....	II
Abbildungsverzeichnis.....	IV
1 Einleitung (Manuel Meyer).....	1
2 Grundlagen Open Source (Thomas Scherer)	2
3 Methodisches Vorgehen (Maurice Goerges).....	3
3.1 Grundlagen empirischer Forschung	3
3.2 Untersuchungsdesign	7
3.3 Das Experteninterview als Erhebungsmethode	7
4 Interviewergebnisse.....	11
4.1 Startups (Manuel Meyer).....	11
4.1.1 Ergebnisse Startups	11
4.2 Jungunternehmen (Maurice Goerges).....	13
4.2.1 Ergebnisse Jungunternehmen	13
4.3 Kleine und mittlere Unternehmen (Thomas Scherer).....	14
4.3.1 Ergebnisse kleine und mittlere Unternehmen	15
4.4 Open Source Anbieter (David Pohlmann)	16
4.4.1 Ergebnisse Open Source Anbieter	17
5 Handlungsempfehlungen.....	21
5.1 Zielgruppendefinition (Manuel Meyer)	21
5.2 Services / Portfolio	22
5.2.1 Internetpräsenz (Manuel Meyer).....	22
5.2.2 Rating / Testberichte / Neuheiten (Manuel Meyer).....	24
5.2.3 FAQ Management (Manuel Meyer).....	26
5.2.4 Vor Ort Service (Thomas Scherer)	27
5.2.5 Support, wo kein Anbieter-Support (David Pohlmann).....	27
5.2.6 Helpdesk (Thomas Scherer).....	28
5.2.7 Expertenvermittlung (Manuel Meyer)	29

5.2.8	Aufklärung (David Pohlmann)	31
5.2.9	Service für Open Source Anbieter (David Pohlmann)	32
5.2.10	Software Stacks (David Pohlmann)	33
5.2.11	Test- bzw. Demoumgebung (David Pohlmann).....	36
5.3	Öffentlichkeitsarbeit (Maurice Goerges)	37
5.3.1	Referenzmanagement	37
5.3.2	Social Media / Foren.....	37
5.3.3	Präsenz auf Messen.....	38
5.4	Organisation (Thomas Scherer)	38
5.4.1	Finanzierung.....	38
5.4.2	Partnermanagement.....	40
6	Kritische Reflexion / Fazit (gemeinsam).....	42
	Anhang.....	44
	Quellen.....	48

Abbildungsverzeichnis

Abb. 1: Prozess empirischer Forschung	4
Abb. 2: Zielgruppendefinition	22
Abb. 3: Internetpräsenz	24
Abb. 4: Rating, Testberichte, Neuheiten	25
Abb. 5: FAQ Management.....	26
Abb. 6: Vor-Ort Service	27
Abb. 7: Support, wo Community und Hersteller nicht ausreichen	28
Abb. 8: Helpdesk.....	29
Abb. 9: Expertenvermittlung	31
Abb. 10: Mittlung zwischen Anbieter, Dienstleister und Unternehmen	33
Abb. 11: Beispiel Software Stack.....	35
Abb. 12: Demoumgebung für Sektoren	36
Abb. 13: Geschäftsmodelle	40
Abb. 14: Partnermanagement	41

1 Einleitung

Diese Seminararbeit zum Thema „Methodik/Systematik einer Beratungsdienstleistung für das Open Source Beratungszentrum Baden-Württemberg“ wurde im Rahmen des Integrationsseminars im sechsten Semester des Studiengangs „International Business for Management and Information Technology“ an der DHBW Stuttgart erstellt. Hintergrund dieser Arbeit ist das Vorhaben des Landes Baden-Württemberg in Kooperation mit der MFG Innovationsagentur für IT und Medien Baden-Württemberg ein Open Source Beratungszentrum in Baden-Württemberg einzurichten.

Aus dem zunehmenden Angebot von Open Source Software und der damit einhergehenden Verbreitung dieser, sind viele entsprechende Beratungsdienstleistungen zu diesem Thema entstanden. Diese sind teilweise jedoch sehr spezifisch oder zu generisch was viele Anforderungen an Problemlösungen nicht abdeckt. Aus diesem Grund soll das Open Source Beratungszentrum für eine persönliche und individuelle Erstberatung zur Hilfe stehen.

Im ersten Schritt dieser Arbeit geht es um das Herausarbeiten von möglichen Anforderungen an ein solches Angebot eines Open Source Beratungszentrums. Diese Anforderungen werden Anhand einer empirischen Forschung mittels Experteninterviews eruiert. Zunächst wird die methodische Vorgehensweise einer solchen Umfrage näher betrachtet. Anschließend wird ein Fragenkatalog erstellt, der Fragen zu Themen wie der Anwendung von Open Source Software, benötigte Supportangebote, dem Software-Kaufentscheidungsprozess sowie die generelle Anforderung an eine Beratungsdienstleistung, stellt. Mithilfe dieses Fragenkatalogs werden Experteninterviews durchgeführt. Die Experten stammen aus verschiedenen Unternehmen wie Startups, jungen Unternehmen, kleinen und mittleren Unternehmen sowie Anbietern von Open Source Software.

In einem zweiten Schritt werden die Antworten zu den jeweiligen Unternehmensgruppen evaluiert und ausgewertet. Ziel ist es für jede Unternehmensart Gemeinsamkeiten herauszufinden, welche Anforderungen an ein Open Source Beratungszentrum betreffen. Diese Ergebnisse sollen auch als Grundlage für den folgenden Schritt dienen.

Im dritten Schritt folgt die Erarbeitung eines Beratungs- und Service-Portfolios inklusive einer Priorisierung dieser. Dabei werden nicht nur Handlungsempfehlungen für die Beratung in Bezug auf spezielle Open Source Software gegeben, sondern auch Möglichkeiten erarbeitet, auf welche Weise ein solcher Beratungsservice aufgebaut werden kann und welche Resource dafür benötigt werden. Wichtig ist dabei Beratungsdienstleistungen zu entwickeln, die allen vier Zielunternehmensgruppen gerecht wird.

2 Grundlagen Open Source

Open Source bedeutet Offen und Quelle, dies bezieht sich auf den Quellcode der Software.

Die Open Source Initiative definiert Kriterien für Open Source Software. Dies bedeutet nicht nur, dass der Quellcode einsehbar sein muss. Die wichtigsten Rechte sind hierbei das Recht die Software ohne Einschränkungen weiterzugeben, den Quellcode zu erhalten und auch diesen zu verändern. Dieser veränderte Text darf dann auch weitergegeben werden. Zudem ist es erlaubt die Software für jeden Zweck einzusetzen. Bekannte Lizenz die diese Kriterien erfüllen sind beispielsweise die Apache License 2.0, die GNU General Public License oder die MIT License. Open Source ist kein Markenzeichen. Trotzdem besteht ein breiter Konsens was darunter zu verstehen ist. Daher ähneln sich die zuvor genannten Lizenzen stark.¹

¹ Vgl.: Diedrich (o.J.)

3 Methodisches Vorgehen

3.1 Grundlagen empirischer Forschung

Die Begrifflichkeit *Empirie* (griech. *empeiria* = Erfahrung) wird als Wissenschaftsverständnis verstanden, bei dem alle Erkenntnisse anhand von Beobachtung gemachten Erfahrungen zurückgeführt werden.² Empirische Wissenschaft gilt nach FRÜH als die „*systematische und intersubjektiv nachvollziehbare Sammlung, Kritik und Kontrolle von Erfahrungen*“³, womit die Kernelemente noch einmal zusammengefasst sind. In anderen Worten kann Empirie als Erhebung und Interpretation von Daten über Gegebenheiten und Vorgänge im soziokulturellen Bereich (soziale Tatbestände) verstanden werden.⁴ Deshalb wird im Zuge der Empirie auch oft von der empirischen Sozialforschung gesprochen. Sozial wahrnehmbare Tatbestände sind beobachtetes menschliches Verhalten oder durch Sprache vermittelte Informationen über Erfahrungen, Meinungen, Werturteile oder Einstellungen.⁵

Um solche Informationen zu erfassen werden in der empirischen Forschung diverse methodische Instrumente wie bspw. die Befragung, die Beobachtung oder die Inhaltsanalyse genutzt.⁶

Als primäre Zielsetzung empirischer Forschung gelten u.a. die Gewinnung von neuen Erkenntnissen und Hypothesen, die Überprüfung von Hypothesen und Theorien, die Fundierung von rationalen Planungs- und Entscheidungsprozessen sowie die Bewältigung von praktischen Problemen.⁷

Der systematisch ablaufende Prozess der empirischen Forschung kann wie folgt vereinfacht dargestellt werden.

² Vgl. Institut für Medien und Bildungstechnologie (2014)

³ Früh, W. (2012), S.1

⁴ Vgl. Hillmann, K.-H./Hartfiel, G. (1994), S.179

⁵ Vgl. Atteslander, P. (2003), S.4

⁶ Vgl. Ebenda, S.5

⁷ Vgl. Pez (2004), S.8



Abb. 1: Prozess empirischer Forschung⁸

In der **Problembenennung** wird ein zu untersuchendes Problem in Form von wissenschaftlichen Fragestellungen formuliert (z.B. welche Rolle spielt OS-Software im Unternehmensumfeld?). Die **Gegenstandsbenennung** definiert das Untersuchungsobjekt, welches zur Klärung der Fragestellung untersucht werden soll. Die **Auswahl und Anwendung der Forschungsmethode** sorgt für die Erhebung der Daten. Dabei werden mit quantitativen- und qualitative Methoden zwei Ansätze empirischer Forschungsinstrumente unterschieden, die sich allerdings nicht gegenseitig ausschließen.⁹ Die gewonnenen Daten werden im nächsten Schritt ausgewertet (**Auswertungsverfahren**). Die aus der Auswertung resultierenden Ergebnisse können dann in einem letzten Schritt z.B. in Form der Definition von Handlungsempfehlungen verwendet werden (**Verwendung von Ergebnissen**).

Die Durchführung der empirischen Forschung im Zuge dieser Ausarbeitung orientiert sich an diesem Prozess.

Der folgende Abschnitt soll den Unterschied zwischen den quantitativen und qualitativen empirischen Forschungsmethoden aufzeigen. Daraufhin erfolgt die Vorstellung des Experteninterviews als Beispiel einer qualitativen Erhebungsmethode.

3.1.1 Qualitative- und Quantitative Forschung

Empirische Forschung kann in großzahlige, quantitative- und kleinzahliger, qualitative Forschung unterschieden werden.

Die Variabilität von Merkmalen eines Untersuchungsobjekts bei der **qualitativen Forschung** wird anhand von wenigen (kleinzahlig) gezielt ausgewählten Einzelfällen aufgezeichnet und

⁸ Eigene Darstellung in Anlehnung an Atteslander, P. (2003), S.22ff.

⁹ Vgl. Atteslander, P. (2003), S.6

verbal beschrieben.¹⁰ Das daraus entstehende qualitative Material wird dabei meist in Textform oder grafisch bzw. nichtnumerisches ausgedrückt und anschließend interpretativ ausgewertet und dient als verbale Beschreibung der Erfahrungswirklichkeit.¹¹ Qualitative Interviews, Beobachtungen und Gruppendiskussionen zählen zu den gängigsten Erhebungsformen qualitativen Materials.¹² In qualitativen Interviews werden meist unstandardisierte, offene Fragen gestellt, die von den Befragten weitgehend frei beantwortet werden können.¹³

Quantitative Forschung untersucht die Variabilität eines Merkmals über die definierte Zuordnung von messbaren Zahlenwerten.¹⁴ Anders als bei der verbalen Beschreibung qualitativer Forschung, erfolgt die Messung von Ausschnitten der Beobachtungsrealität dabei durch die statistische Verarbeitung und Interpretation von Messwerten.¹⁵ Der Hauptunterschied liegt darin, dass bei *quantitativen* Untersuchungen eine sehr viel höhere Anzahl (großzählig) an Objekten untersucht wird.¹⁶ Aus diesem Grund werden in quantitativen Interviews oder schriftlichen Fragebögen standardisierten Befragungen von Personen durchgeführt. In diesen wird den Befragten ein vordefiniertes Raster an geschlossenen Fragen zur Beantwortung gestellt, bei denen diverse Antwortmöglichkeiten vorgegeben sind. Somit besteht eine sehr hohe Vergleichbarkeit der Antworten, allerdings entfällt die freie Antwortgestaltung durch die individuelle Interpretation der Befragten.

Die Menge aller Merkmalsmessungen der Untersuchung wird bei beiden Methoden als Daten bezeichnet. Aufgrund der standardisierten Erhebung quantitativer Daten, lassen sich diese erheblich einfacher auswerten als qualitative Daten. Andererseits bietet das Ergebnis quantitativer Untersuchungen, aufgrund der Standardisierung, nur sehr generellen Erkenntnissen.¹⁷ Beide Methoden schließen sich nicht gegenseitig aus und können durchaus in Kombination angewandt werden.¹⁸

Um neue Erkenntnisse zu den Zusammenhängen der Befragung zu erlangen und den Ermessensspielraum der Befragten nicht durch vorgefertigte Antwortmuster einzuschränken entschieden sich die Verfasser dieser Ausarbeitung für die Durchführung einer qualitativen

¹⁰ Vgl. Albers, S. et al. (2009), S.7

¹¹ Vgl. Bortz, J./Döring, N. (2006), S.296

¹² Vgl. Ebenda, S.296

¹³ Vgl. Kutscher, N. (2004)

¹⁴ Vgl. Albers, S. et al. (2009), S.7

¹⁵ Vgl. Bortz, J./Döring, N. (2006), S.269

¹⁶ Vgl. Bauer, A./Mulley, U. (2014)

¹⁷ Vgl. Albers, S. et al. (2009), S.7

¹⁸ Vgl. Kelle, U. (2008), S.14

empirischen Forschung und das Experteninterview als Forschungsmethode gewählt, welche im Folgenden vorgestellt wird.

3.1.2 Das Experteninterview als qualitative Erhebungsmethode

Das Experteninterview ist eine qualitative Methodik zur Datenerhebung durch Befragung.¹⁹ Die Methodik zielt auf die Abfrage der Wissensbestände von Experten im Kontext ihrer verantwortlichen Zuständigkeit ab, die für die Problemlösung als Gegenstand des Forschungsinteresses relevant sind.²⁰ Aus diesem Grund werden Experteninterviews häufig in den Bereichen der industriesoziologischen Forschung, der Organisationsforschung, der Bildungsforschung oder Politikforschung eingesetzt.²¹

Im Gegensatz zu anderen Interviewformen gilt es in einem Experteninterview bei „[...] geeigneten Personen [Experten] zeiteffektiv erfahrungsgestütztes Expertenwissen abzuholen.“²² Unter Expertenwissen kann nach SPRONDEL „[...] notwendig erachtetes Sonderwissen“ in einer arbeitsteiligen Gesellschaft verstanden werden.²³ Die Definition des Expertenbegriffs gründet in der wissenssoziologischen Unterscheidung von Laien und Experten sowie der entsprechenden Unterscheidung von Allgemeinwissen und spezialisiertem Sonderwissen (Expertenwissen).²⁴ Somit wird als Experte eine Person verstanden, die sich durch eine „institutionalisierte Kompetenz zur Konstruktion von Wirklichkeit“ im Rahmen des Forschungsfeldes auszeichnet.²⁵

Normalerweise erfolgt das Experteninterview persönlich oder telefonisch und findet zwischen dem Forscher und dem von ihm ausgewählten Experten statt. Ein Leitfaden dient als Strukturelement bei der Durchführung und definiert Themen, die im Fokus der Befragung stehen. Um mögliche Themendimensionierungen durch den Experten nicht zu behindern sollte ein Leitfaden mit offenen Fragen und flexibel genutzt und nicht standardisiertes Ablaufschema verstanden werden.²⁶ So ist dem Experten die Möglichkeit gegeben, frei zu berichten und bestimmte Situationen anhand von Beispielen zu erläutern.

¹⁹ Vgl. Mayer, H. O. (2008), S.37f; Flick, U. (1995), S.109f.

²⁰ Vgl. Pfadenhauer, M. (2009), S.452

²¹ Vgl. Meuser, M./Nagel, U. (2009), S.465

²² Mieg, H./Brunner, B. (2004), S.199

²³ Sprondel, W. M. (1979), S.148

²⁴ Vgl. Meuser, M./Nagel, U. (2009), S.467

²⁵ Vgl. Hitzler, R./Honer, A./Maeder, C. (1994), S.13ff.

²⁶ Vgl. Langenbacher-König, R. (2013)

Besonders wenn es um die Perspektive von Experten auf Prinzipien des Sachverhalts, auf Probleme und ihre Ursachen oder auf Lösungen und ihre Konsequenzen geht, gilt die Methode des Experteninterviews als dienliches Instrument der empirischen Sozialforschung.²⁷

Nach der Durchführung des Experteninterviews werden die grundlegenden Erkenntnisse aus den erhobenen Daten schriftlich zusammengefasst, weshalb es sich anbietet das Interview audiographisch aufzunehmen.²⁸

3.2 Untersuchungsdesign

Das Untersuchungsdesign orientiert sich am Prozess der empirischen Forschung (vgl. **Fehler! Verweisquelle konnte nicht gefunden werden.**). Die folgende Untersuchung soll dabei die in der Problemstellung (vgl. **Fehler! Verweisquelle konnte nicht gefunden werden.**) definierten Leitfragen untersuchen.

Zur Durchführung der empirischen Forschung bedienen sich die Autor dem empirischen Verfahren von Experteninterviews. Diese wurden mit ausgewählten Experten aus den Segmenten StartUps, junge Unternehmen, SMB und OS-Anbieter durchgeführt. Durch die verschiedenen Sichtweisen kann so eine objektivere Sammlung von Informationen gewährleistet werden. Zur Auswertung der erhobenen Daten erfolgt als Zusammenfassung der Grunderkenntnisse in einem jeweiligen Segment. Anschließend werden anhand der gewonnenen Erkenntnisse Handlungsempfehlungen ausgesprochen.

3.3 Das Experteninterview als Erhebungsmethode

Für die Durchführung der empirischen Untersuchung, haben sich die Autoren für die Methode des leitfadengestützten Experteninterviews entschieden. Diese Vorgehensweise bietet sich an, weil bei der Erforschung organisationaler Zusammenhänge die Befragung von Experten als Informanten weit verbreitet und anerkannt ist.²⁹ Durch diese Vorgehensweise können segmentspezifischen Erkenntnissen identifiziert und allgemein gültige Aussagen anhand des Fachwissens der Experten gewonnen werden. Da die Erhebung in verschiedenen Segmenten durchgeführt wird, erwarten die Autoren heterogene Ergebnisse. Kritisch zu betrachten ist, dass die unterschiedlichen Interpretationen von offen gestellten Fragen in dieser Methodik zu Verzerrung führen können. Durch die sorgfältige Auswahl diverser Befragter soll

²⁷ Vgl. Pfadenhauer, M. (2009), S.459f.

²⁸ Vgl. Meuser, M./Nagel, U. (2009), S.476f.

²⁹ Vgl. Mitchell, V. W. (1994)

dies aber vermieden werden, um ein allgemeingültiges Bild anhand von überwiegend gleichartig geäußerten Antworten zu erhalten.

3.3.1 Auswahl der Experten

In einem ersten Schritt wurde festgelegt, dass jeder der vier Autoren ein spezielles Segment bearbeitet. Daraufhin wurde von jedem der Autoren Ansprechpartner im jeweiligen Segment gesucht, die als Informationsquelle in Frage kommen. Für die umfassende Beantwortung der im Leitfaden skizzierten Fragen ist eine umfassende Unternehmenskenntnis sowie Fachkenntnis zu IT-Zusammenhängen nötig.

Insgesamt wurden von den Autoren 16 Experten interviewt. Die Experten wurden dabei teilweise anhand von Empfehlungen, über Kontakte aus dem privaten Umfeld oder anhand von Suchergebnissen im Internet ausgewählt. Der Erstkontakt fand i.d.R. via Email statt. Unter den Experten waren hochrangige Mitarbeiter in den berufsspezifischen Rollen: Geschäftsführer oder Chief Information Officer (CIO) vertreten. Die Auswahl geeigneter Experten ist den Verfassern nicht ganz leicht gefallen. Neben den wenigen privaten oder empfohlenen Kontakten konnten überwiegend nur Unternehmen anhand von einer Internetrecherche gefunden und angeschrieben werden. Die Antwortrate und Bereitschaft für ein Interview bei Letzteren war dabei sehr gering.

3.3.2 Leitfadenentwicklung

Die Einzelinterviews wurden leitfadengestützt, mit offenen Fragen, durchgeführt, sodass die Befragten die Möglichkeit hatten, sich frei zu äußern und das wiederzugeben, was ihnen bezüglich der Frage als wichtig erscheint. Der Vorgang entlang des Leitfadens dient dazu, dass in verschiedenen Interviews die gleichen Thematiken angesprochen werden und ermöglicht so eine gewisse Vergleichbarkeit der einzelnen Interviews. Dies bedeutet nicht, dass alle Fragen exakt gleich gestellt werden müssen. Die Vorgehensweise entlang eines Leitfadens ermöglicht eine Strukturierung des Interviews und bietet gleichzeitig Flexibilität für den Autor sich an die Gesprächssituation anzupassen. Dadurch kommt ein natürlicherer Gesprächsverlauf zustande. Dem Interviewer hilft der Leitfaden lediglich als Orientierung und es ist ihm überlassen, spontan neue Fragen zu verwenden oder gewisse Fragen mehr zu vertiefen und dafür bspw. andere Fragen auszulassen. Frageformulierungen oder eine genau vorgegebene Reihenfolge der Fragen sind also nicht bindend.

Um eine aufwändige Strukturierung bei der Auswertung des in den Interviews erhobenen Informationsmaterials zu vermeiden, wurden schon im Leitfaden thematische Schwerpunkte

gesetzt, die als Kriterien bei der Auswertung dienen. Der Leitfaden wurde vorab in die folgenden Kategorien gegliedert:³⁰

- Anwendung
- Support
- Kaufentscheidungsprozess
- OS-Beratungsdienstleistung

In der ersten Kategorie handelt es sich um die Erfassung der aktuellen Einschätzung der Befragten über die gegenwärtige Nutzung von Software-Komponenten. Dabei wurde speziell nach dem Anteil von OS-Lösungen unter den zurzeit genutzten Anwendungen gefragt. Beim Punkt „*Support*“ richtet sich der Blick auf die derzeit in Anspruch genommenen Support-Dienstleistungen im Zuge der zurzeit genutzten Anwendungen und wie die Erfahrungen mit diesem aussehen. Danach wird gezielt auf den Entscheidungsprozess zur Einführung einer Software eingegangen. Abschließend wird von noch auf die Rolle von Beratungsdienstleistungen im Zusammenhang mit Software behandelt. Hierbei wird speziell auf den Bedarf von OS-Beratungsdienstleistungen eingegangen.

3.3.3 Durchführung der Interviews

Die 16 Experteninterviews wurden im Zeitraum vom 25.05.2014 bis zum 01.07.2014 in Form einer telefonischen Befragung durchgeführt. Dies ermöglichte eine flexiblere Termingestaltung und Erreichbarkeit der Experten. Um die Gespräche vertraulich zu behandeln, wurde jeweils zu Beginn vereinbart, die durch das Gespräch erhobenen Inhalte anonymisiert zu verwenden, weshalb in der Auswertung der Interviewinhalte keine Namen der Befragten oder zugehörige Unternehmen veröffentlicht werden. Zusätzlich wurde das Einverständnis der Interviewpartner für die digitale Aufzeichnung des Gespräches eingeholt. Diese erfolgte in allen Interviews. Durch das audiographische Aufzeichnen des Interviews, konnten die Autoren auf eine unmittelbare Mitschrift verzichten und sich völlig auf das jeweilige Gespräch konzentrieren.

Zu Beginn der Experteninterviews fand jeweils eine kurze Einleitung der Autoren statt, in welcher eine persönliche Vorstellung und das Thema des Integrationsseminars erläutert wurden. Danach wurden die Befragten gebeten, sich selbst kurz vorzustellen. Die Gesprächsdauer der Experteninterviews lag zwischen 20 und 35 Minuten, was sich auf die unterschiedlich ausführlichen Antworten der Interviewten zurückführen lässt. Die Fragen im Interview orientierten sich an der Konzeption des Leitfadens. Dabei wurden von den Verfassern in jedem Interview individuell Vertiefungsfragen gestellt, wenn es die Gesprächssituation anbot. Zum Ende eines jeden Interviews bedankte sich der jeweilige Interviewführer für

³⁰ siehe Anhang

die genommene Zeit und klärte ob es von Seiten des Experten noch Fragen gibt. Einige Male wurde diesbezüglich das Interesse geäußert, über die Ergebnisse der letztendlichen Ausarbeitung informiert zu werden.

Im Nachgang wurden die wesentlichen Inhalte aus den audiographischen Aufzeichnungen des Interviews durch den jeweiligen Verfasser schriftlich zusammengefasst.

4 Interviewergebnisse

4.1 Startups

Die Experteninterviews für Gründer und Startups in Bezug auf deren Erfahrung mit Open Source Software im Unternehmensumfeld, wurden mit drei Startups aus Stuttgart, Darmstadt und Berlin durchgeführt. Bei Startups handelt es sich um „junge, noch nicht etablierte Unternehmen, die zur Verwirklichung einer innovativen Geschäftsidee [...] mit geringem Startkapital gegründet werden“.³¹ Gerade das geringe Startkapital ist für diese Branche oftmals eine Herausforderung, auch was die Beschaffung von Software betrifft. Aus diesem Grund ist es wichtig, Experten aus Startups zu interviewen um deren Anforderungen zu erkennen. Diese Erkenntnisse sollen anschließend in die Entwicklung von Handlungsempfehlungen für das Open Source Beratungszentrum einfließen.

4.1.1 Ergebnisse Startups

Anwendung

Aus den Experteninterviews wird ersichtlich, dass Open Source Software bei Gründern und Startups von großer Bedeutung ist und daher häufig verwendet wird. Dabei kommt Open Source Software hauptsächlich im Backend-Bereich zum Einsatz, vorwiegend bei Servertechnik, Datenbanken und der Programmierung. Vor allem die Webserver-Software Nginx und das Betriebssystem Linux spielen eine wichtige Rolle (vgl. Anhang 1). Für die Anwendung von Open Source Software in Startups gibt es verschiedene Gründe. Zum einen ist die Anschaffung von Open Source Software oftmals kostenlos. Des Weiteren hat man bei Open Source Software in den meisten Fällen die Möglichkeit selbst in den Code einzugreifen um diese mit bestimmten Funktionalitäten zu erweitern. Dies ist allerdings lizenzabhängig. Das bewirkt, dass dem Entwickler bei der Integration eine gewisse Flexibilität eingeräumt wird.

Auf der anderen Seite kommt bei Startups auch kommerzielle Software zum Einsatz. Dies ist vor allem in den Bereichen der Textverarbeitung, der Kommunikation und Planung sowie der Bildbearbeitung der Fall. Hierbei werden hauptsächlich Software wie MS Office, Projektplanungstools wie Basecamp oder Podio, Kommunikationssoftware wie Skype oder Google Hangouts, sowie Adobe Photoshop für die Bildbearbeitung verwendet. Vorteil ist hier vor allem die Benutzerfreundlichkeit.

Support

³¹ Springer Gabler Verlag (o. J.a)

Bezüglich des Supports geben die Interviewergebnisse der Startups auch ein ähnliches Bild ab. Der Support spielt besonders bei Gründern und noch sehr jungen Startups eine recht untergeordnete Rolle. Die sogenannte „Do it yourself“-Mentalität ist hier vorherrschend, was sicherlich auch mit dem vorhandenen Kapital zusammenhängt. Des Weiteren werden wenige bezahlpflichtige Supportdienstleistungen in Anspruch genommen. Viel mehr werden bei Problemen, Lösungen in entsprechenden Communities und Support-Foren gesucht oder das nähere Umfeld bis hin zu Experten von Hochschulen befragt.

Dennoch gaben alle interviewten Startups an, in Zukunft auch auf bezahlten Support zurückgreifen zu wollen, hauptsächlich dann, wenn die Mitarbeiterzahl deutlich größer sei. Von bezahltem Support würde jedoch ein qualitativ guter Support erwartet.

Kaufentscheidung

Der Anschaffungsprozess von neuer Software wird hauptsächlich durch neue Anforderungen angestoßen. Dabei gaben alle Startups an, sowohl Open Source als auch kommerzielle Software in Betracht zu ziehen. Teilweise würde dies auch eine untergeordnete Rolle spielen, da eine gute, hilfreiche und vor allem funktionierende Lösung gesucht werde. Dabei sieht der Prozess der Evaluation bei allen Startups ähnlich aus. Um sich über das bestehende Angebot an bestimmter Software zu informieren, werden Quellen wie Google, Blogs oder Online-Communities wie Twitter herangezogen. Gerade Blogs und Online-Foren haben den Vorteil, dass Softwareangebote gefunden werden, die durch ihre Leistung überzeugen und nicht durch Promotion. Eines der wichtigsten Kriterien bei der Auswahl der Software sei vor allem die Kompatibilität zu anderen bestehenden Softwareanwendungen.

Beratungsdienstleistung

Für Beratung bezüglich Open Source Software verwenden Startups hauptsächlich das Internet, da es ohne großen Aufwand, kostenlos abrufbar ist und einen großen Pool an Information beherbergt. Dennoch sind die interviewten Unternehmen einem solchen Angebot nicht abgeneigt, sofern es einen Mehrwert darstellen kann, der den zusätzlichen Aufwand des Kontaktaufbaus rechtfertigt. Anforderungen diesbezüglich sind beispielsweise kostenlose Kennenlernertermine und Erstsupport. Zudem sollten die Angebote zielgruppenspezifisch sein um dediziert auf die Ratsuchenden zu reagieren. Des Weiteren sollten die Antwortzeiten schnell sein um den Vorteilen des Internets entgegenzukommen. Weitere Anforderungen waren die Vermittlung von Experten sowie eine attraktive Internetpräsenz.

4.2 Jungunternehmen

Eine genaue Definition von Jungunternehmen gibt es in der Literatur nicht, da es unterschiedliche Auffassungen über den zeitlichen Rahmen eines solchen Unternehmens gibt. Gemeinsam haben aber alle Definitionen, dass ein Jungunternehmen sich über das Alter des Unternehmens definiert. Im Verständnis dieser Ausarbeitung bezieht sich der Begriff Jungunternehmen auf jene Unternehmen, die seit ihrer Gründung mindestens 2 und maximal fünf Jahre im Betrieb sind. Wenn beispielsweise ein Betrieb im Jahr 2010 gegründet wurde, gilt er bis 2015 als junges Unternehmen.

4.2.1 Ergebnisse Jungunternehmen

Für das Segment Jungunternehmen wurden fünf Unternehmen aus verschiedenen Branchen befragt. Darunter waren zwei lokal agierende und drei international tätige Unternehmen mit einer Angestelltenzahl zwischen 2 bis 50 Mitarbeitern vertreten.

Anwendung

Die Software-Anwendungsbereiche lagen bei den Firmen in den Kategorien: Bürosoftware, Warenwirtschaftssystem, Designsoftware, Kassensysteme sowie Online-Shop und Websysteme. In der derzeitigen Anwendung wurden lediglich zwei Open Source Lösungen von den Unternehmen genannt. Zu diesen zählte ‚Filezilla‘ sowie ‚Open Office‘ im Bereich der Bürosoftware. Für alle anderen Softwarebereiche wurde nach Angaben der Befragten mit Hilfe von kommerzieller Software unterstützt. Bei Bürosoftware war dies i.d.R. Microsoft Office. Zwei der an der Umfrage teilnehmenden Firmen hatten eine Lexware-Lösung als Warenwirtschaftssystem im Gebrauch.

Support

Der Software-Support gestaltete sich bei den Firmen unterschiedlich. Drei Unternehmen nehmen On-Demand Support in Anspruch, welcher bei jeder Support-Dienstleistung individuell abgerechnet wird. Davon waren zwei Services über eine gebührenpflichtige Hotline und einer über einen Online Service, bei welchem der entsprechende Support-Mitarbeiter Online Zugriff auf die entsprechenden Systeme hat. Eines der Unternehmen nutzt einen monatlich festgeschriebenen Dienstleistungsvertrag mit einem Support-Dienstleister und eine Firma löst nach eigenen Angaben sämtliche Probleme selbst, ohne fremde Support-Leistungen in Anspruch zu nehmen.

Generell werden die bisherigen Erfahrungen mit dem jeweiligen Software-Support als gut bis ausgezeichnet beschrieben. Da keines der befragten Unternehmen seine Kernkompetenz in der IT hat, wird ein Hauptkriterium darin gesehen, dass die bestehenden Anwendungen und

Systeme fehlerlos funktionieren. Dies wird gut mit der Aussage beschrieben: „Software ist ein Nebenschauplatz und soll einfach nur funktionieren und man sollte sich möglichst wenig Gedanken darüber machen müssen.“ Aus diesem Grund hat ein guter Support eine hohe Wichtigkeit, allerdings soll dieser mit möglichst wenig eigenen Aufwand von statten gehen.

Kaufentscheidung

Der Entscheidungsprozess zur Einführung einer Software sieht bei allen Unternehmen ähnlich aus. Zu Beginn muss eine klare Aufgabenstellung definiert werden, deren Bedarf sich i.d.R. an einer Problemstellung innerhalb des Unternehmens orientiert. In einem nächsten Schritt werden Systeme, die zur Problemlösung in Frage kommen gesucht. Ob Open Source- oder proprietäre Lösungen spielt nach Angaben der Befragten zuerst keine Rolle. Für die Befragten kommen möglichst Standard Lösungen in Betracht, mit der Begründung, dass Individualsoftware i.d.R. teurer und meist nur kurzlebig ist, weil andere Standardlösungen weiterentwickelt worden sind. Die Informationssuche geschieht meist im Internet oder auf Messen. Zwei der fünf befragten Jungunternehmen haben sich vor der Anschaffung Ihrer aktuellen Systeme kommerziell beraten lassen. Als weitere Alternative zur Informationsbeschaffung gilt der Austausch von Erfahrungswerten mit befreundeten Unternehmen. Nach der Auswahl potenzieller Software bestellen die Unternehmen oftmals vorab eine Testversion zur besseren Erprobung. Anhand der Erfahrungen aus dem Test werden dann konkrete Angebote bei dem jeweiligen Anbietern eingeholt. In einem letzten Schritt erfolgt dann die Entscheidung für eine entsprechende Anwendung unter Abwägung der Vor- und Nachteile, woraufhin dann i.d.R. der Kauf erfolgt.

Beratungsdienstleistung

Hinsichtlich von Beratungsdienstleistungen in Bezug auf Open Source Software zeigten sich die befragten Unternehmen sehr defensiv. Lediglich einer der Befragten äußerte generelles Interesse an einer solchen Dienstleistung. Die anderen Unternehmen begründeten ihr Desinteresse in dem fehlenden Bedarf in der Anschaffung neuer Software bzw. dem unverhältnismäßig hohem Aufwand ein bestehendes System auszutauschen.

4.3 Kleine und mittlere Unternehmen

Kleinere und mittlere Unternehmen werden in Deutschland auch als Mittelstand bezeichnet. Die Abgrenzung von anderen Unternehmen erfolgt durch verschiedene Faktoren. Die können Bilanzsumme, Umsatz oder Anzahl der Beschäftigten sein. Geläufig ist hierbei die Abgrenzung von Unternehmen durch eine Umsatzgröße zwischen einer Million und fünfzig Millionen Euro und einer Mitarbeiterzahl von unter 500. Neben diesen quantitativen Kriterien, werden oftmals noch qualitative Kriterien betrachtet. Ein Beispiel hierfür ist, dass bei den KMUs in der Regele Eigentümer und Geschäftsführer oftmals die gleiche Person ist, bzw.

aus der gleichen Familie stammen. Die Abgrenzung wird in der Praxis beispielsweise relevant bei der Mittelstandsförderung.³²

4.3.1 Ergebnisse kleine und mittlere Unternehmen

Anwendung

Die befragten Unternehmen haben eher wenig Open Source Software im Einsatz. Oftmals wird Open Source dort verwendet, bei dem der Implementierungsaufwand gering ist. Ein Beispiel hierfür ist die Verwendung von Greenshot, einem Open Source Programm zur Erstellung und Bearbeitung von Screenshots. Des Weiteren wurde auch Linux als Betriebssystem für Server und teilweise in geringer Anzahl auch von Desktop Computern verwendet. NetBeans als Entwicklungsumgebung für Software wurde in Unternehmen ebenfalls verwendet. Bei der Unternehmenssoftware wurde kaum auf Open Source Lösungen zurückgegriffen. Lediglich ein Unternehmen verwendete mit Redmine, einer Projektmanagement Software, eine Open Source Variante. In einigen Unternehmen wird Software selbst entwickelt auch wenn dies nicht zu der Kernkompetenz des Unternehmens zählt. Bei den Betriebssystemen wird meistens auf kommerzielle Lösungen zurückgegriffen. Neben Microsoft, welches auf Servern und Endgeräten verwendet wird, werden auch Mac OS auf Computern verwendet. Auf mobile Endgeräte wurde nicht eingegangen. Das Office Paket von Microsoft wurde in allen befragten Unternehmen verwendet. Bei ERP Systemen wurde sowohl auf Eigenentwicklung als auch proprietäre Software zurückgegriffen. Als Barriere wurde angegeben, dass der Leistungsumfang oftmals nicht ausreichend ist, 100 prozentige Kompatibilität zum Industriestandard nicht gegeben ist oder einfach keine deutsche Sprachversion verfügbar ist. Eine weitere große Barriere ist es, dass bei komplexeren Lösungen es schwierig ist einen geeigneten Einführungspartner zu finden. Punkte die für eine Open Source Lösung sprechen, wurden eine breite Community, teilweise performantere Software (Hierbei wurde Linux als Betriebssystem für Server genannt), sowie die Vielzahl an kostenlosen Plug Ins die es für manche Lösungen gibt. Teilweise wurde der Vorteil genannt das Open Source Lösungen kostenlos sind. Wichtig bei den Unternehmen ist jedoch der Kosten Nutzen Effekt. Deshalb sind diese in der Regel auch dazu bereit Geld auszugeben. Ein Nachteil der Open Source mit sich bringt, ist dadurch dass hinter den Lösungen kein kommerzielles Interesse der Community besteht, die Weiterentwicklung fragwürdig ist und sehr abhängig ist. Generell stehen die Unternehmen der Nutzung von Open Source Software offen gegenüber.

³² Prof. Dr. Klodt (o.J.)

Support

Die befragten Unternehmen halten den Software Support für wichtig. Teilweise ist dies auch einer der wichtigsten Entscheidungskriterien bei der Neuanschaffung von Software. Mit steigender Komplexität nimmt auch der Bedarf an Support zu. Besonders in und kurz nach der Einführung der Software hat der Support einen hohen Stellenwert. Bevorzugt wird von allen Unternehmen der persönliche Support oder per Telefon. Bei weniger komplexen und wichtigen Angelegenheiten kann dieser auch per E-Mail oder Chat stattfinden. Eine besondere negative Erfahrung hat ein Unternehmen mit einem Support gemacht, welcher aus einem Off-Shore Land stattgefunden hat. Das Problem verursachte eine Verfügbarkeit von nur noch 80 Prozent der Rechenleistung. Der Support machte einen Fehler und sorgte dafür dass die komplette IT ausgefallen ist. Allgemein wird ein Off oder Near Shore Support nicht gerne in Anspruch genommen. Die Mehrheit der Unternehmen ist bereit zusätzliche finanzielle Mittel bereitzustellen um einen guten Support zu erhalten.

Kaufentscheidung

Bei allen Unternehmen wird Software angeschafft weil ein Bedarf besteht, seine Unternehmensprozesse informationstechnologisch abzubilden oder für die Entwicklung Software benötigt wird. Bevor diese angeschafft wird, geht in der Regel zuerst eine Recherche voraus. Diese erfolgt im Internet. Zudem werden die in den Unternehmen etablierten IT Berater um Rat gefragt. Wichtig für den Entscheidungsprozess sind Reviews, Testberichte und in vielen Fällen eine Live Demo des Produkts. Auf die Beratung des etablierten IT Lieferanten wird viel Wert gelegt. Kann dieser eine Lösung implementieren und/oder supporten sind Unternehmen eher dazu geneigt die Entscheidung für eine solche Lösung zu treffen.

Beratungsdienstleistung

Keiner der befragten Unternehmen hat bisher dedizierte Open Source Beratungsdienstleistungen in Anspruch genommen. Bevorzugt wird die Beratung von bereits bekannten IT Lieferanten. Der Anspruch an eine Beratung und Einführung von Open Source wäre, dass diese nicht teurer ist als der Kauf von gleichwertiger proprietärer Software. Wichtig ist, dass keine unnötigen Verzögerungen, wie etwa durch lange Anfahrtswege entstehen. Zudem wird Beratung eigentlich nur in größeren Implementierungsprojekten benötigt. Fragen zur Open Source Software die sich am meisten Stellen, sind die Zukunftsaussichten sowie die Kompatibilität zu Industrie Standards.

4.4 Open Source Anbieter

Open Source Software kann grundsätzlich auf zwei verschiedene Arten entwickelt werden. Zum einen existiert die Kategorie von Open Source Software, deren Entwicklung tatsächlich ausschließlich von der Community vorangetrieben und durchgeführt wird. Dabei arbeiten

viele verschiedene Entwickler, die teilweise keinerlei Verbindung haben und auch unterschiedliche Standorte haben, an dem selben Software-Projekt. Auch wenn eine gewisse grundsätzliche Organisationsstruktur bzw. eine Administration unabdingbar ist, kann grundsätzlich jeder im Rahmen seiner zeitlichen, fachlichen und auch finanziellen Möglichkeiten ein Projekt dieser Art unterstützen.

Daneben besteht aber auch die Möglichkeit, dass eine wirtschaftlich orientierte Unternehmung (oder auch eine Non-Profit-Organisation) in Eigenregie und mit einem festen Entwicklerstamm eine Software unter eine Open Source Lizenz stellt und den Quellcode dementsprechend der gesamten Öffentlichkeit zur Verfügung stellt. Dies können beispielsweise darauf spezialisierte Software-Firmen sein, grundsätzlich ist aber auch hier jede Konstellation denkbar. Der Unterschied zu einem reinen Community-Projekt ist, dass die allgemeine Öffentlichkeit zwar Zugang zu einem Teil oder auch dem kompletten Quellcode einer Applikation erhält, die Anwendung aber grundsätzlich nur von dem jeweiligen entwickelt und weiterentwickelt wird. Oftmals existieren aber auch hier sogenannte „forks“ (=Abspaltung), wodurch beispielsweise eine entsprechende „Community-Version“ der jeweiligen Applikation entsteht.

Die tatsächlichen Community-Open Source Projekte nehmen ihrer Anzahl nach zwar einen deutlichen größeren Rahmen ein, spielen aber gerade im geschäftlichen Einsatz eine eher untergeordnete Rolle, da die für Unternehmen sehr wichtigen Aspekte wie bspw. Support, Anpassung und Weiterentwicklung einer Applikation nur schlecht oder überhaupt nicht gegeben sind.

Aus diesem Grund wurden im Rahmen dieser empirischen Untersuchung auch vier deutsche Unternehmen, die ausschließlich oder zum Teil Open Source Software entwickeln bzw. als Dienstleister integrieren, interviewt.

Aufgrund der hohen Relevanz in Unternehmen wurden dafür Hersteller von Software aus den Bereichen ERP, Groupware und Service Management bzw. Kundenservice ausgewählt. Darüber hinaus wurde auch ein Integrator für Open Source Software aller Art befragt.

4.4.1 Ergebnisse Open Source Anbieter

Anwendung

Wie nicht anders zu erwarten, verwenden die Open Source Anbieter, welche im Rahmen dieser empirischen Untersuchung befragt wurden, alle zumindest für bestimmte Anwendungsgebiete auch Open Source Software anderer Hersteller. So setzten alle befragten Unternehmen auf diverse Linux-Distributionen als Betriebssystem und OpenOffice bzw. LibreOffice als Office-Suite. Auch im Bereich Server und Datenbanken sind vorrangig Open Source Produkte im Einsatz. So verwendete ebenfalls alle befragten Unternehmen Zarafa

Server als Groupware-Applikation. Als Datenbank kam bei einigen Unternehmen beispielsweise die Open Source Lösung PostgreSQL zum Einsatz.

Daneben lag ein weiterer Schwerpunkt auf dem Bereich Programmierung. Alle Unternehmen, die selbst Open Source Software herstellen oder zumindest als Dienstleister in diesem Bereich arbeiten, setzen auch selbst zur Entwicklung der Software auf Open Source Produkte wie beispielsweise die Entwicklungsumgebung Eclipse.

Darüberhinaus kamen im Web-Bereich vereinzelt die Analyse-Plattform Piwik und als Wiki-System MediaWiki zum Einsatz. Ein Hersteller verwendete im Bereich Service Management außerdem die OTRS-Suite.

Aus dem Bereich der kommerziellen Software kamen bzw. kommen bei den befragten Unternehmen nur sehr wenige Applikationen zum Einsatz. Das Anwendungsgebiet liegt dabei vorrangig im grafischen bzw. Medien-Bereich, wo vor allem Adobe Photoshop als Grafik- bzw. Bildbearbeitungsprogramm eingesetzt wird. Außerdem setzt ein Unternehmen als Fernwartungssystem bzw. für Webcasts auf TeamViewer.

Die Gründe, warum die befragten Unternehmen vorrangig auf Open Source Software setzen sind sehr einheitlich. So gaben alle Unternehmen an, aufgrund dessen, dass sie selbst Anbieter im Open Source Bereich sind, ebenfalls selbst auf diese Art von Anwendungen zu setzen. Weitere Gründe seien die Stabilität und Qualität des Quellcodes sowie die geringen Kosten bei der Anschaffung. Grundsätzliche Vorteile werden im hohen Integrationsgrad und der Flexibilität von Open Source Software gesehen. Außerdem wurde die Möglichkeit gelobt, selbst in den Quellcode der Applikation eingreifen und so diesen auch überwachen zu können. Dadurch sei es vor allem im sicherheitsrelevanten Bereich möglich, die entsprechenden Applikationen zu auditieren und auf Aspekte wie Datenschutz zu untersuchen. Ein Unternehmen habe so die Möglichkeit nachzuvollziehen, wie Daten erfasst bzw. verarbeitet und gespeichert werden.

Kontroverse Aussagen wurden teilweise bzgl. Nachteilen von Open Source Software gemacht. So gab ein Unternehmen an, gerade bei Open Source Software, die der Community entstammt, sei die Stabilität der Applikation oftmals nicht so gegeben, wie es bei proprietärer Software der Fall ist. Ebenfalls kontrovers zu den bereits genannten Vorteilen bemängelt ein Unternehmen die teilweise noch nicht vorhandene Integration von Open Source Software. So sei das gerade von Windows bekannte „Plug-and-Play“-Gefühl bei Open Source Software oftmals noch nicht vorhanden und eine manuelle Installation bzw. Konfiguration nötig. Von allen Unternehmen nachteilig wird außerdem der oft nicht oder nicht ausreichend vorhandene Support gesehen. Darüberhinaus sei der Funktionsumfang bei Open Source Produkten oft sehr unterschiedlich und oft nicht mit den Standards proprietärer Software zu vergleichen. Einen erheblichen Nachteil, gerade auch im Vergleich zu aktuellen kommerziellen Produk-

ten, sehen die befragten Unternehmen in der grafischen Benutzeroberfläche von Open Source Software. Diese sei entweder überhaupt nicht, oder gemessen an aktuellen Benutzeroberflächen im proprietären Bereich, nur unzureichend vorhanden und nicht ansprechend bzw. zielführend gestaltet. Als Grund bzw. verantwortlich für dieses Defizit wird der zumeist nicht vorhandene Screen-Designer bzw. das bei den Programmieren oft nicht vorhandene Know-How im Bereich Design genannt.

Support

Da die Unternehmen den Support bei Open Source Software grundsätzlich als Nachteil sehen, sei besonders wichtig, dass Anbieter von professioneller Open Source Software einen guten und umfangreichen Support - auch in verschiedenen Ausprägungen - bereit stellen. Dadurch würde ursprüngliche Nachteil eliminiert und den Unternehmen ein mögliches Geschäftsmodell geboten. Grundsätzlich halten alle Unternehmen den Support von Open Source Software, gerade in kritischen Bereichen, für sehr wichtig. Dabei wurde dies teilweise sowohl selbst von den Unternehmen übernommen, oder auch von Dienstleistern oder Herstellern direkt bezogen. Neben Geschäftsanwendungen und komplexen Applikationen, wo Support unabdingbar sei, gäbe es aber auch Bereiche, wo Support auch bei Open Source Anwendungen nicht notwendig sei. Dazu zählen gerade einfach Applikationen im Konsumenten-Bereich wie das Grafik-Tool GIMP, wo die Community als Support-Quelle ausreiche. Außerhalb davon bevorzugen alle Unternehmen aber tatsächlichen Support eines verlässlichen Dienstleisters bzw. übernehmen den Support, durch die Möglichkeit, Einblick in den Quellcode zu haben, selbst. Die Community könne die gleichbleibende Qualität und Konsistenz, die beim Support in kritischen Bereichen notwendig ist, nicht bieten. Alle Unternehmen sind bereit, für einen entsprechenden Support in diesen Bereichen auch die notwendigen finanziellen Bereit zu stellen, erwarten dann aber auch die notwendige Gegenleistung. Dies sei insbesondere bei Open Source Software der Fall. Negative Erfahrungen mit einer bezahlten Support-Dienstleistung hatte nur ein Unternehmen, welches in einem bestimmten Fall das Problem einer Applikation nach einmonatigem Stillstand selbst, durch Einsicht in den Quellcode, lösen musste.

Kaufentscheidung

Wenn der Bedarf besteht, neue Software anzuschaffen bzw. dies für die Lösung eines Problems notwendig ist, gaben die Unternehmen an, zunächst ein Anforderungsprofil zu erstellen und anschließend nach entsprechenden Lösungen zu suchen. Dabei steht an erster Stelle die Funktionalität gefolgt von den Kosten der Anschaffung. Grundsätzlich sei dabei zunächst unerheblich, ob es sich um proprietäre, Open Source oder Individualsoftware handelt. Bei gleicher Leistung werden Open Source Lösungen jedoch bevorzugt. Außerdem werde vor allem in kritischen bzw. sicherheitsrelevanten Bereichen auf Open Source Software gesetzt,

um einerseits Einblick in den Quellcode haben und andererseits sich im Notfall selbst behelfen zu können. In erster Linie gehe es aber darum, dass in wichtigen Bereichen ein tatsächliches Unternehmen hinter dem Produkt steht, was in der Lage ist, Support zu leisten.

Beratungsdienstleistung

Bezüglich einer Beratungsdienstleistung zum Thema Open Source gaben die Unternehmen an, dass gerade in diesem Bereich die Unternehmen sehr unerfahren sind und deshalb sich eher auf eine Beratung einlassen und auf diese verlassen würden. Gerade bei diesen unerfahrenen Unternehmen sei das Thema Open Source oft negativ behaftet und für sie fraglich, ob ausreichend Support gegeben, das Produkt bzw. Unternehmen verlässlich ist und auch in Zukunft noch existent sei und die Weiterentwicklung vorangetrieben wird. Grundsätzlich fragen Open Source fremde Unternehmen auch danach, wie der Hersteller der Software denn überhaupt Geld verdiene. Nachdem Vorurteile wie „was nichts kostet, ist nichts wert“ geklärt sind, werden schnell inhaltliche Fragen gestellt und die Funktion der Anwendung und nicht die Tatsache, dass es sich um Open Source Software handelt, steht im Vordergrund.

Die Anbieter von Open Source Produkten haben an eine Beratungsdienstleistung die Anforderung, dass diese Fragen zunächst geklärt und anfragende Unternehmen für das Thema Open Source sensibilisiert werden. Gerade Gründer kämen oft durch den Kostendruck zu Open Source und es sei gerade in dieser Startphase wichtig, die Unternehmen mit Open Source Software zu konfrontieren und dadurch die Verbreitung zu vergrößern. Außerdem soll ein auf Open Source basierendes Gesamtkonzept, wie ERP-Systeme und ähnliches, propagiert werden. Grundsätzlich stehen alle befragten Unternehmen einem Open Source Beratungszentrum positiv gegenüber und begrüßen, dass dadurch Einstiegshürden genommen bzw. verringert werden.

Die befragten Anbieter von Open Source Software kommen allesamt aus den für Unternehmen relevanten Bereichen wie ERP, Groupware und Kundenservice. Die Zielgruppen der Hersteller reichen dabei von kleinen und kleineren mittelständischen Unternehmen bis hin zu großen Mittelständlern und sogar internationalen Konzernen, über alle Branchen hinweg. Bis auf einen Anbieter vertreiben alle Unternehmen ausschließlich Open Source Software, diese teilweise jedoch auch als Enterprise-Edition mit nutzerabhängigen bezahlten Lizenzen. Grundsätzlich setzen aber alle befragten Unternehmen vorrangig auf ein ein Subscription-Modell bzw. Support- und Wartungsverträge als primäres Geschäftsmodell und bieten die eigentliche Software zumindest in bestimmten Versionen kostenfrei zum Download an.

5 Handlungsempfehlungen

5.1 Zielgruppendefinition

Damit das Open Source Beratungszentrum ein Serviceportfolio anbieten kann, welches perfekt auf die Anforderung der Ratsuchenden zugeschnitten ist, ist es von essentieller Bedeutung eine Zielgruppendefinition vorzunehmen. Darauf basierend können dann dementsprechende Angebote und Beratungsdienstleistungen erstellt werden.

Eine grobe, übergeordnete Definition der Zielgruppe ist, dass das Open Source Beratungszentrum eine Zielgruppe ansprechen möchte, die Open Source Software einsetzt beziehungsweise diverse Berührungspunkte mit dieser Art von Software hat. Auf einer detaillierteren Ebene betrachtet können das Privatpersonen, Unternehmen, Experten oder Anbieter von Open Source Software sein. Diese vier Gruppen werden im Folgenden nochmal genauer betrachtet.

Privatpersonen

Eine mögliche Zielgruppe können Privatpersonen sein, die zum Beispiel anstelle von der kommerziellen Software Microsoft Office, die Open Source Variante OpenOffice bevorzugen. Diese Personen können Probleme im Umgang mit der Software haben und sich bei Fragen eventuell an das Open Source Beratungszentrum wenden. Da es sich hierbei jedoch oft um recht triviale Probleme handelt kann man hier mit vorgefertigten Informationsbroschüren und Gebrauchsanweisungen agieren.

Unternehmen

Bei Unternehmen, die Open Source Software im Unternehmenskontext verwenden, muss man nach Art des Unternehmens unterscheiden. Wie bereits für die Experteninterviews definiert, kann man hier drei wesentliche Gruppen unterscheiden: Gründer und Startups, junge Unternehmen sowie kleine und mittlere Unternehmen. Dabei kann man feststellen, dass die Anforderungen dieser drei Unternehmenstypen an Open Source Software recht unterschiedlich sein können. Während kleine und mittlere Unternehmen Open Source Software oftmals als Ergänzung zu ihrem bestehenden Softwareportfolio verwenden, sind Gründer und Startups oftmals aus Kostengründen ausschließlich auf diese angewiesen. Bei der Erstellung von Beratungsangeboten sollten diese unterschiedlichen Anforderungen beachtet werden, so dass für jede dieser Teilzielgruppen ein ansprechendes Angebot besteht.

Anbieter von Open Source Software

Auch Anbieter von Open Source Software stellen für das Open Source Beratungszentrum eine entscheidende Zielgruppe dar. Womöglich nicht in der Form, dass diese Unternehmen eine Beratungsdienstleistung beanspruchen. Viel mehr besitzen diese Firmen jedoch ein gewisses Know-how in Bezug auf Open Source Software, welche für das Beratungszentrum interessant ist. Durch entsprechende Prozesse könnte zwischen dem Beratungszentrum und den Open Source Software Anbietern ein Wissensaustausch entstehen, von welchem auf der einen Seite die Anbieter profitieren um einen größeren Marktüberblick zu erhalten. Auf der anderen Seite kann das Beratungszentrum das gewonnene Wissen direkt in die Beratungsdienstleistungen einfließen lassen, die somit ein gesteigertes Qualitätsniveau repräsentieren.

Experten

Ebenso wie Open Source Software Anbieter besitzen auch Experten von Hochschulen ein umfangreiches Wissen bezüglich Open Source Software. Auch von dem Wissen dieser Experten könnte das Beratungszentrum profitieren um das Beratungs- und Serviceangebot auszubauen. Daher ist es auch wichtig hierfür entsprechende Prozesse für den Wissenstransfer einzurichten.

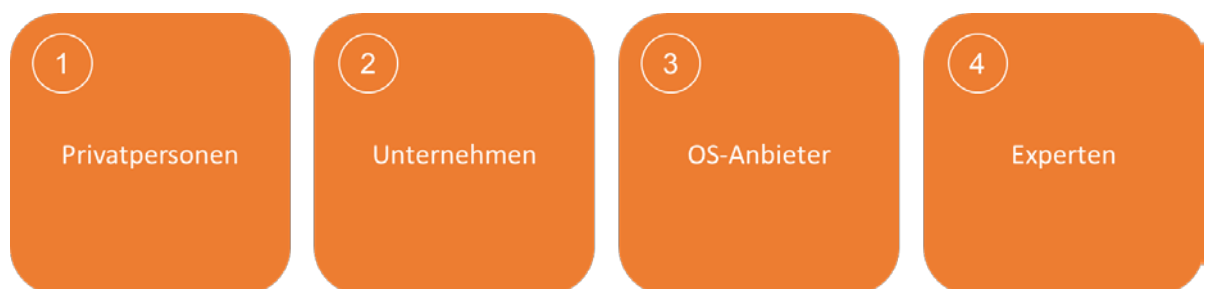


Abb. 2: Zielgruppendefinition

5.2 Services / Portfolio

5.2.1 Internetpräsenz

Die Ergebnisse der Experteninterviews weisen auf, dass der benötigte Support von Open Source Software häufig über das Internet abgerufen wird. Dort haben sich über die Zeit einige Foren und Communities gebildet, die Open Source Software und auftretende Probleme diskutieren. Dabei gibt es sehr umfangreiche Foren, wie zum Beispiel „Stack Overflow“.³³ Bei dieser Webseite handelt es sich um eine Plattform, auf der Benutzer Fragen bezüglich Softwareproblemen stellen, gleichzeitig aber auch Antworten auf gestellte Fragen liefern können. Anschließend können andere Benutzer Antworten auf ihre Richtigkeit bewerten. Somit ist für

³³ Vgl. Stack Exchange, Inc. (2014a)

den Nutzer ersichtlich welche Antworten mit hoher Wahrscheinlichkeit hilfreich sein können. Eine Alternative zu „Stack Overflow“ bietet zum Beispiel „Experts Exchange“.³⁴

Diese Online-Foren sind im Open Source Software Support sehr populär, da sie einige positive Anreize bieten. Auf der einen Seite sind sie schnell abrufbar. Dies ermöglicht es den Ratsuchenden, Fehler, die erkannt werden, unmittelbar zu recherchieren um mögliche Lösungen zu erörtern. Zudem kann der Nutzer bei diesen Online-Foren auf einen recht großen Erfahrungsschatz zurückgreifen, wodurch es recht wahrscheinlich ist, dass ein Fehler schon einmal bei anderen Nutzern aufgekommen ist und über diesen bereits diskutiert wurde.

Ein weiterer positiver Aspekt ist die Tatsache, dass viele der Benutzer der die Fragen beantworten selbst Experten auf diesen Gebieten sind. Zudem werden hilfreiche Ratschläge, wie bereits erwähnt, durch andere Nutzer kenntlich gemacht, was es für den Ratsuchenden nochmals vereinfacht, eine nützliche Lösung zu finden.

Ein dritter positiver Punkt ist die Möglichkeit des kostenlosen Supports. Die meisten Online-Foren für Open Source Software Support sind kostenlos, was gerade für Gründer, Startups und junge Unternehmen mit wenig Kapital sehr ansprechend ist. Oftmals gibt es zusätzlich zu den Foren noch Live-Chat Angebote, bei welchen Nutzer mit Experten in Kontakt treten können.³⁵

Diese Beleuchtung der Vorteile von Internet-Support soll aufzeigen, dass die Konkurrenz für ein Open Source Beratungszentrum recht umfangreich agiert. Um Anreize für Ratsuchende schaffen zu können, sodass diese anstelle von Online-Foren das Beratungszentrum aufsuchen, muss die Internetpräsenz des Zentrums ebenfalls umfangreiche Angebote aufweisen. Dies kann durch, zu Online-Foren, ähnliche Funktionen geschehen. Eine weitere Möglichkeit ist die Differenzierung durch Angebote, welche Online-Foren nicht bereitstellen.

Ein Angebot das dem Angebot eines Online-Forums entsprechen könnte, wäre die Möglichkeit Fragen zu stellen und Antworten zu erhalten. In diesem Fall ist es jedoch wichtig zu erkennen, dass renommierte und frequentiert aufgerufene Online-Foren auf eine breite Masse an Benutzern zurückgreifen und somit zu sehr vielen Problemen Lösungen anbieten. Um als Support-Webseite des Beratungszentrums von Benutzern aufgesucht zu werden, ist es wichtig hier ein Angebot zu schaffen, das einen Mehrwert bietet. Eine Option wäre es im Rahmen eines solchen Online-Support-Forums Experten von Universitäten aus der Region, sowie Experten von Anbietern von Open Source Software zu gewinnen. Diese könnten mit theoretisch fundiertem Wissen, bestimmte Fragestellungen beantworten und somit plausible und korrekte Antworten liefern. Bei Online-Foren wie „Stack Overflow“ oder „Experts

³⁴ Vgl. Experts Exchange, LLC. (2014)

³⁵ Vgl. Stack Exchange, Inc. (2014b)

Exchange“ hingegen ist es nicht klar ersichtlich, ob Antworten auf theoretischem Wissen oder auf reinem Erfahrungsschatz beruhen. Durch diese gezielte Einbindung von Experten könnte ein klarer Mehrwert geschaffen werden, der dem Kunden zusätzliche Sicherheit in der Lösungsfindung bietet. Auf diese Weise könnten auch Live-Chat-Angebote mit Experten zu bestimmten Zeiten angeboten werden.

Um auch für Einsteiger eine Anlaufstelle darzustellen, wäre es sinnvoll Tutorialangebote online bereitzustellen. Gerade Programmierneulingen fehlt es oft an einer guten Anlaufstelle mit Tutorials, die aber auch eng verknüpft ist mit einer Online-Support-Community um aufkommende Fragen direkt zu beantworten ohne zwischen verschiedenen Plattformen wechseln zu müssen. Generell ist es bei diesem Thema wichtig, dass Fragen die von Ratsuchenden gestellt werden möglichst zeitnah beantwortet werden. Dies ist der Fall bei herkömmlichen Communities und wird daher von Benutzern oftmals als Normalfall angesehen. Gerade am Anfang, wenn die Nutzeranzahl noch nicht sehr hoch ist, kann dies eine gewisse Herausforderung darstellen, der man sich bewusst sein sollte.

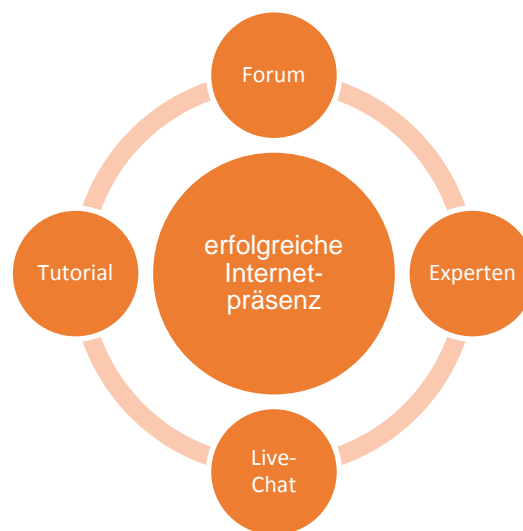


Abb. 3: Internetpräsenz

5.2.2 Rating / Testberichte / Neuheiten

Benutzer von Online-Support-Foren haben bei einem Besuch auf einer solchen Webseite die Intention, möglichst schnell eine Lösung für ein Problem zu erhalten. Zudem sollte es sich oftmals um eine qualitativ hochwertige Lösung handeln, welche ressourcensparend und nicht zu komplex ist. Diese Anforderungen können durch eine große Anzahl an beteiligten Personen erzielt werden, da die Wahrscheinlichkeit von vorhandenen Experten bei steigender Anzahl von Nutzern höher wird. Werden Antworten dann noch aufgrund ihrer Qualität positiv bewertet, ist es relativ wahrscheinlich eine zufriedenstellende Lösung zu erhalten. Trotzdem bleibt eine Ungewissheit.

Eine andere Möglichkeit um das Vertrauen der Ratsuchenden zu erhalten ist die Tatsache, dass man auf dem Gebiet ein Experte ist und das nicht nur anhand von Erfahrung festmachen sondern auch andere Referenzen aufweisen kann. Hierzu gibt es drei Optionen die das Open Source Beratungszentrum in Erwägung ziehen kann. Diese Möglichkeiten sollen den Ratsuchenden dabei als Orientierungshilfe dienen um sich in dem breitgefächerten Angebot an Open Source Software zurechtzufinden.

Zum einen kann eine Bewertung von Software beziehungsweise ein Vergleich zwischen verschiedenen Open Source Softwareangeboten erstellt werden. Dies kann branchenspezifisch oder aber auch Anwendungsspezifisch vorgenommen werden. Der Kunde kann dann das sogenannte Ranking online abrufen und beispielsweise eine Auflistung an Open Source Softwareangeboten für den Bürogebrauch finden. Anhand des Rankings und der Bewertung kann er dann das für ihn passende herausuchen.

Eine zweite Option ist das Veröffentlichen von Testberichten von bestimmten Open Source Softwareangeboten. Dabei kann es sich zum Beispiel um Lastentests von Open Source NoSQL-Datenbanken unter bestimmten Umständen handeln. Diese Testberichte können den Ratsuchenden dann bei Kaufentscheidungsprozessen bezüglich Open Source Software unterstützen.

Als letzter Punkt bietet sich ein regelmäßiger Bericht über Neuheiten und Neuerscheinungen von Open Source Software an. Dies ermöglicht den Ratsuchenden ständig auf dem neuesten Informationsstand bezüglich Open Source Angeboten zu sein, um dieses Fachwissen in Lösungsfindungsprozesse zu integrieren.



Abb. 4: Rating, Testberichte, Neuheiten

All diese drei Vorschläge könnten hierbei auf der Internetseite des Open Source Beratungszentrums veröffentlicht werden. Testberichte könnten auch gezielt in anderen Foren platziert werden um Benutzern den Weg auf die Webseite des Beratungszentrums zu erleichtern.

5.2.3 FAQ Management

FAQ steht für „Frequently Asked Questions“³⁶ und bedeutet so viel wie „Zusammenstellung von Informationen zu besonders häufig gestellten Fragen“.³⁷ Dabei handelt es sich um ein Thema, welches häufig vernachlässigt wird, allerdings eine nicht unbedeutende Rolle spielt. FAQs befinden sich normalerweise unter einer eigenen Rubrik auf einer Webseite und sollen eine Auflistung der von Nutzern gestellten Fragen darstellen, welche frequentiert nachgefragt werden. Für viele Ratsuchende ist dies oft eine erste Anlaufstelle auf einer Support-Webseite. Finden die Benutzer keinen nützlichen Rat mit Hilfe der FAQs verlassen viele auch direkt im Anschluss wieder die Webseite und suchen nach anderen hilfreichen Webseiten.

Auf der anderen Seite können hilfreiche FAQs den Benutzer motivieren diese Webseite öfters aufzusuchen, da er überzeugt ist, dass er dort mit hoher Wahrscheinlichkeit eine Antwort auf sein Problem findet. Aus diesem Grund ist es von enormer Bedeutung ein angemessenes FAQ-Management zu betreiben.

Damit das Open Source Beratungszentrum ein erfolgreiches FAQ-Management verfolgt müssen einige Punkte beachtet werden. Einerseits ist es wichtig, dass es einen fortlaufenden Prozess gibt, bei dem Ratsuchende ihre Fragen stellen können und sie bei einer Häufung auch in die FAQs aufgenommen werden. Dabei ist es auch wichtig, Fragen die an Relevanz verlieren wieder aus den FAQs zu nehmen. Ein zweiter Punkt ist, dass die Antworten auf die FAQs regelmäßig auf ihre Gültigkeit und Aktualität überprüft werden. Zudem ist es wichtig eine angemessene und übersichtliche Anzahl an FAQs in der Auflistung zu führen. Dies erfordert eine Priorisierung dieser.



Abb. 5: FAQ Management

³⁶ Springer Gabler Verlag (o. J.b)

³⁷ Bibliographisches Institut GmbH (2013)

5.2.4 Vor Ort Service

Um die Hemmschwelle für eine Beratung zu minimieren, sollte es für den Kunden möglich sein im eigenen Unternehmen die Beratungsdienstleistung zu erlangen. Hierbei bekommt der Berater einen Einblick in das zu beratende Unternehmen. Spezifische Probleme und Umstände können somit besser analysiert und somit eine passende Beratung erbracht werden.

Die Anfahrtskosten sollten ohne Marge dem Kunden in Rechnung gestellt werden. Dies senkt die Kosten für das Beratungszentrum und für den Kunden ist dieser Service preiswert. Dennoch sorgt die Kostenweitergabe dafür, dass dieser Service nur bei ernsthaftem Interesse in Anspruch genommen wird.

Beispielsweise überlegt ein Unternehmen die Einführung eines CRM Systems. Um überhaupt gegen den etablierten IT Lieferant des Unternehmens, welcher eine proprietäre Lösung verkaufen möchte, konkurrieren zu können, müssen die Kosten für den möglichen Kunden gering gehalten werden. Der etablierte IT Lieferant, kennt das Unternehmen meistens sehr gut. Daher muss beim Vor Ort Service ein fähiger Berater eingesetzt werden, welcher auch idealerweise über ein Industriespezifisches Know-How verfügt.



Abb. 6: Vor-Ort Service

5.2.5 Support, wo kein Anbieter-Support

Eines der größten Bedenken bzw. Ängste gegenüber Open Source Software ist der befürchtete, nicht vorhandene oder schlechte Support.

Tatsächlich ist es so, dass einige Hersteller nicht in der Lage sind, einen ausreichenden Support leisten zu können bzw. bestimmte Open Source Produkte tatsächlich von der Community entwickelt werden. Gerade bei dieser Art Software ist es zwingend notwendig, dass

sich das Open Source Beratungszentrum eigenes Know-How der jeweiligen Applikation aneignet und in der Lage ist, zumindest grundlegenden Support leisten zu können.

Da es im Vorhinein nur schwer abzuschätzen ist, welche Applikationen hier von besonderer Bedeutung sind ist es notwendig, sich den Bedürfnissen der anfragenden Unternehmen anzupassen und sein Know-How-Portfolio stets dahingehend zu erweitern, dass die am häufigsten vorkommenden Applikationen und Probleme abgedeckt werden können.

Als erster Schritt sollte dennoch ein gewisses Grundportfolio mit wichtigen Basiskenntnissen zu den populären Open Source Applikationen ohne direkten Herstellersupport aufgebaut werden. Dies sind zum Beispiel die OpenOffice-Suite sowie die Produkte der Apache Foundation.

Um die Kenntnisse der Support-Mitarbeiter des Open Source Beratungszentrums stets zu erweitern und den Bedürfnissen der Kunden anpassen zu können empfiehlt sich zudem die Einrichtung einer sogenannten Known-Error-Database, wo Lösungen zu Fehlern strukturiert abgespeichert und verwaltet werden können.

Für den Kontakt mit dem Kunden ist außerdem zu überprüfen, ob der Einsatz einer Kundenservice- oder Help-Desk-Lösung - wie zum Beispiel das Open Source Produkt OTRS von c.a.p.e. IT - notwendig bzw. sinnvoll ist und die Arbeit erleichtern und effizienter gestalten könnte.



Abb. 7: Support, wo Community und Hersteller nicht ausreichen

5.2.6 Helpdesk

Der Helpdesk sollte zunächst zwei Aufgaben erfüllen. Zum einen für potentielle Kunden eine Art Anlaufstelle, zum anderen bei Problemen mit Partnern eine Möglichkeit zur Eskalation zu bieten.

Der Erstkontakt muss einfach möglich sein und so konzipiert sein, dass die Hemmschwelle gering ist. Dies kann erreicht werden, in dem verdeutlicht wird, dass dieser Kontakt kostenlos ist. Bei dem Erstkontakt sollten Fragen zur Beratung beantwortet werden und Informationen über den Kunden gesammelt werden um dann den richtigen Berater zu involvieren.

Sollten bei einer Abwicklung von Projekten oder bei der Erbringung einer Service Dienstleistung zwischen einem Kunden und einem Partner Probleme auftreten, könnten sich beide Parteien an den Helpdesk wenden um eine Schlichtung oder Klärung zu schaffen.

Mögliche Kontaktformen wären Telefonanrufe, E-Mail oder bei entsprechender Größe auch einen Live Chat.

Als Zusatzleistung könnte auch ein kostenpflichtiger Helpdesk für OS Lösungen errichtet werden, allerdings macht dies nur Sinn wenn dies nicht mit dem Geschäftsmodell der Partner konkurriert und genug Kunden diesen in Anspruch nehmen.



Abb. 8: Helpdesk

5.2.7 Expertenvermittlung

Wie im Kapitel der Handlungsempfehlung „Internetpräsenz“ deutlich wird, stellt das Internet die größte Konkurrenz für das Open Source Beratungszentrum dar. Daher gilt es einen Mehrwert zu schaffen, der Anreiz genug ist, dass Kunden den zeitlichen Mehraufwand in Kauf nehmen um mit dem Open Source Beratungszentrum in Kontakt zu treten. Dieser Kontakt kann vor Ort, telefonisch oder online geschehen.

Im Raum Stuttgart sind einige Hochschulen angesiedelt, die auch Studienangebote im informationstechnologischen Bereich anbieten. Dazu gehören zum Beispiel die Universität Stuttgart³⁸, die Hochschule der Medien³⁹ oder die Duale Hochschule Baden-Württemberg⁴⁰ um nur einige wenige zu nennen. Aus den Experteninterviews ging hervor, dass sich Ratsuchende aus der Unternehmensgruppe Gründer und Startups, die oftmals selbst noch studieren, an Experten aus Freundeskreisen, an Kommilitonen oder aber auch an Professoren von entsprechenden Studiengängen wenden. Der Vorteil gegenüber dem Internet sei hier vor allem auch die Diskussionen über bestimmte Probleme, die oft neue Möglichkeiten und hilfreiche, problemspezifische Lösungen hervorbringen.

Diese Experten lassen sich nicht nur an Hochschulen finden. Im Raum Stuttgart gibt es auch viele Open Source Anbieter aber auch Firmen, die Open Source Software einsetzen und Experten für diese Gebiete anstellen.

Ein Angebot des Open Source Beratungszentrums, Ratsuchende an Experten von Hochschulen, Open Source Anbietern und Firmen zu vermitteln um für diese eine Möglichkeit der Diskussion zu bieten, würde einen klaren Vorteil gegenüber den Internetangeboten darstellen. Der mögliche Wissenstransfer der durch den Austausch mit Experten erzielt werden kann, stellt einen deutlichen Mehrwert dar, der den zusätzlichen zeitlichen Aufwand rechtfertigt.

Um den Prozess der Expertenvermittlung für den Ratsuchenden übersichtlich zu gestalten, sollten die Experten in einem sogenannten Expertenpool gelistet werden. Dabei ist es sinnvoll diese nach Rubriken beziehungsweise Spezialgebieten von Open Source Software zu unterteilen. Somit kann der Ratsuchende nach Experten in dem Fachgebiet suchen, in welchem er aktuell Probleme hat und Lösungen braucht. Eine Anfrage kann dann entweder telefonisch beim Open Source Beratungszentrum gestellt werden oder über eine Online-Request-Funktion auf der Internetseite des Beratungszentrums eingehen. Über diese zwei Kontaktwege kann dann ein Termin vereinbart werden.

Eine Herausforderung ist hierbei jedoch überhaupt Experten für diese Einrichtung zu gewinnen. Diese haben oftmals geregelte Arbeitszeiten und sind terminlich eingespannt, sodass man Anreizmodelle entwickeln muss, damit diese für eine Beratungsdienstleistung zur Verfügung stehen. Hierfür gibt es zwei mögliche Ansätze. Auf der einen Seite kann eine Beratungsdienstleistung über ein entsprechendes Entgelt erbracht werden, welches sich zum Beispiel auf einer stündlichen Basis ermisst. Auf der anderen Seite kann ein Experte seine Beratungsdienstleistung auch als Referenz nutzen. Da die Experten aus Qualitätssiche-

³⁸ Vgl. Universität Stuttgart (2014)

³⁹ Vgl. Hochschule der Medien (2014)

⁴⁰ Vgl. Duale Hochschule Baden-Württemberg Stuttgart (o. J.)

Maßnahmen von dem Open Source Beratungszentrum geprüft werden um für eine solche Dienstleistung zugelassen zu werden, kann dies alleine auch schon als Anreiz genügen. Eine Dienstleistung eines Experten könnte sich auf eine bestimmte wöchentliche Uhrzeit, von einer Dauer von zum Beispiel zwei Stunden, beschränken.

Dieses Angebot ist ein sehr aufwändiges, was aber auch einen einzigartigen Mehrwert bietet. Aus diesem Grund ist es wichtig, dass diese Dienstleistung den Anforderungen der Ratsuchenden entspricht, damit diese dann eventuell auch noch auf weitere Angebote des Open Source Beratungszentrums zurückgreifen, sodass dieses zu ihrer zentralen Anlaufstelle in Fragen zu Open Source Software wird. Daher ist es notwendig einen entsprechenden Feedback-Prozess zu modellieren, der es ermöglicht, detailliertes Feedback des Ratsuchenden zu erhalten. Dies kann in Form eines Online-Fragebogens umgesetzt werden, der nach jedem Expertengespräch auszufüllen ist. Dieser enthält Rückmeldung über die Art des Problems, der Erfolg des Gesprächs und das Resultat. Dies dient zur Ergebnissicherung. Anhand der Rückmeldungen können dann kommende Expertengespräche optimiert werden und bestimmte Themen und Fachgebiete, zu welchen vermehrt Fragen auftreten, forciert werden.

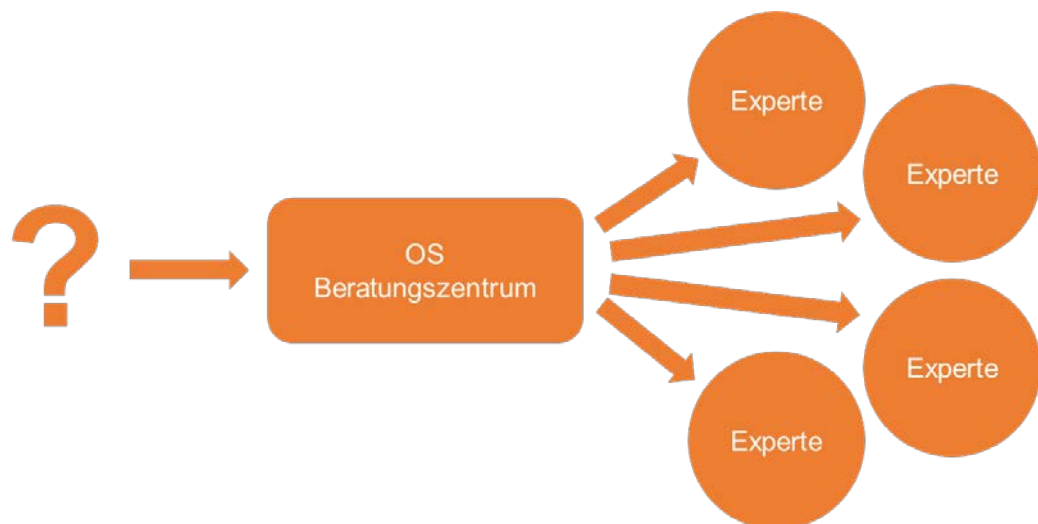


Abb. 9: Expertenvermittlung

5.2.8 Aufklärung

Ein ebenfalls großes Problem, was bei der Auswertung der Interviews deutlich wurde, ist die Tatsache, dass viele Vorurteile zum Thema Open Source existieren. Dabei geht es um Themen wie mangelnden Support, nicht vorhandene oder stagnierende Weiterentwicklung oder die fehlende Verantwortlichkeit eines Unternehmens. Aus diesem Grund ist es wichtig, einen großen Schwerpunkt der Arbeit des Open Source Beratungszentrums auf das Thema Aufklärung zu legen. Dabei muss den anfragenden Unternehmen und Gründern deutlich gemacht werden, dass lediglich ein geringer - und für den Unternehmenseinsatz eher unwichtiger -

Teil aller Open Source Produkte durch die Community entwickelt, supportet und verwaltet werden. Die hier bereits vorgestellten Produkte zum Abbilden betriebswirtschaftlicher Prozesse werden alle von soliden, dahinterstehenden Unternehmen entwickelt und supportet. Durch die Tatsache, dass diese Unternehmen ebenfalls alle gewinnorientiert sind und ein solides Geschäftsmodell haben, ist auch die Frage, womit Open Source Hersteller ihr Geld verdienen, beantwortet. Daraus resultiert auch die von mit Open Source nicht vertrauten Unternehmen oft vorherrschende Angst, die Existenz bzw. weitere Entwicklung eines Produktes sei nicht sichergestellt.

Ebenfalls von hoher Relevanz im Bereich der Aufklärung von nicht affinen Unternehmen sind die generelle Sensibilisierung mit dem Thema Open Source. Dabei ist es vor allem wichtig, dass die oft unter Kostendruck stehenden Gründer und Jungunternehmer mit Open Source konfrontiert werden und so die Verbreitung der entsprechenden Software gefördert wird. Dazu gehört ebenfalls, dass das Open Source Beratungszentrum in der Lage ist, eine Art Grundausstattung von Open Source Produkten für Unternehmen verschiedener Branchen nennen und auch erklären zu können bzw. auch für spezielle Anforderungen oder Anwendungsgebiete ein passendes Open Source Softwareprodukt zu kennen.

5.2.9 Service für Open Source Anbieter

Während sich ein Großteil der Interviews mit Unternehmen auf der Nutzer-Seite befasste, wurden auch einige Anbieter von Open Source Produkten befragt. Um die Zielgruppe des Open Source Beratungszentrums zu erweitern sollten auch diese Unternehmen mit einbezogen werden. Neben den grundsätzlichen angebotenen Dienstleistungen, die auf Anbieter von Open Source Software genauso zutreffen wie auf herkömmliche Unternehmen, ist es sinnvoll, auch dedizierte Fragestellungen und Probleme der Anbieter-Seite adressieren zu können.

Dabei geht es neben Unternehmen, die bereits Open Source Software entwickeln und vertreiben, auch um solche, die in Erwägung ziehen auf ein Open Source Modell umzustellen oder sogar Gründer aus dem Software-Bereich, die vor der Entscheidung stehen, Ihre Produkte proprietär oder Open Source anzubieten.

Fragestellungen, auf die das Open Source Beratungszentrum dabei eingehen sollte und die es noch konkreter zu definieren und zu beantworten gilt, beziehen sich neben den grundsätzlichen Vorteilen davon, auf ein Open Source anstatt auf ein Lizenz-Geschäftsmodell vor allem darauf, wie ein wirtschaftlich agierendes Unternehmen trotz der kostenfreien Bereitstellung von Software und Know-How erfolgreich am Markt sein kann. Dazu müssen die verschiedenen Möglichkeiten der Umsatzgenerierung wie Support- und Wartungsverträge, In-

tegrationsdienstleistungen, das Hosting von Applikationen oder auch tatsächlich trotz Open Source ein klassischer Lizenzvertrieb erklärt werden können.

Darüberhinaus ist ein weiterer Bestandteil des Service für Open Source Anbieter, dass Nutzer und Anbieter von Software hier miteinander in Kontakt gebracht werden und durch gegenseitiges Feedback voneinander profitieren können.

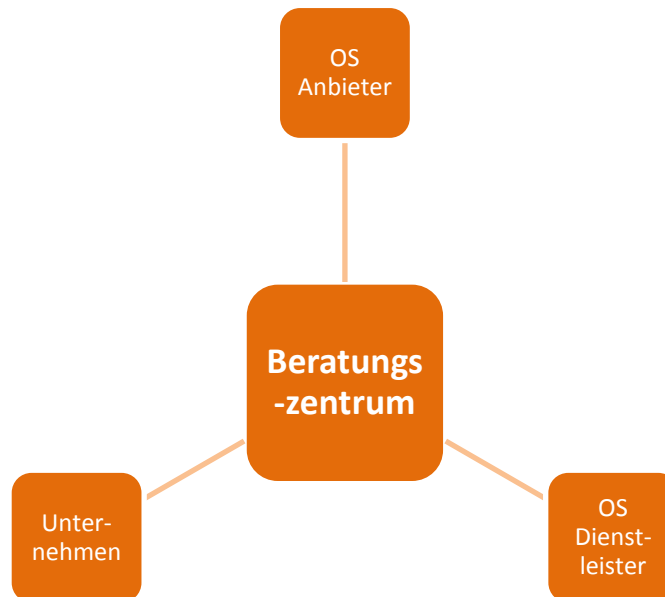


Abb. 10: Mittlerung zwischen Anbieter, Dienstleister und Unternehmen

5.2.10 Software Stacks

Durch die verschiedenen Interviews wurde deutlich, dass ein großes Problem darin besteht, dass viele Unternehmen mit Open Source Software lediglich Produkte wie OpenOffice oder GIMP verbinden und die für Unternehmen wirklich relevanten Produkte eher unbekannt sind. Aus diesem Grund wurde ein Software-Stack entwickelt, der die wichtigsten und populärsten Produkte für Unternehmen aus dem Open Source Bereich beinhaltet.

Dabei wurde auf der einen Seite versucht, betriebliche Prozesse mittels Open Source Software möglichst umfangreich abzudecken (z.B. mit ERP-Systemen etc.), auf der anderen Seite wurde auch ein Portfolio mit oft eingesetzten Anwendungsprogrammen, wie z.B. zur Bild- oder Textverarbeitung erstellt.

Grundsätzlicher Ausgangspunkt eines Open Source Gesamtkonzeptes ist definitiv das Betriebssystem. Aufgrund der vielen Unterschiedlichen Linux-Distributionen, die sowohl den Desktop- als auch für den Servereinsatz und mit Android sogar mobile Geräte abdecken, ist hier bereits ein hoher Grad an Open Source Technologie möglich.

Neben Red Hat und Suse Enterprise bieten auch die meisten anderen, renommierten Linux-Distributionen einen kostenpflichtigen Support- bzw. Wartungsvertrag an, was für den professionellen Einsatz im Unternehmen unabdingbar ist.

Serverprogramme für verschiedene Einsatzgebiete bilden die nächste Ebene. Im Webserver-Bereich bietet das Apache-Projekt für die meisten anzutreffenden Anwendungsbereiche ein geeignetes Serversystem inklusive verschiedener Datenbanken. Diese reichen neben klassischen relationalen Datenbanken bis hin zum NoSQL-Bereich mit z.B. Cassandra und CouchDB. Weitere populäre Datenbanken unter Open Source Lizenz sind PostgreSQL im relationalen und z.B. MongoDB im NoSQL-Segment.

Im Bereich von Groupware bzw. Groupware-Servern ist im Open Source Bereich der deutsche Hersteller Zarafa mit den entsprechenden Produkten marktführend. Auch hier steht ein professionelles Unternehmen hinter dem Projekt, wodurch Support- oder Integrationsdienstleistungen unproblematisch sind.

Der zumindest in Deutschland im Bereich ERP wichtigste Anbieter ist Helium mit der branchenübergreifenden ERP-Lösung Helium V. Das komplett quelloffene System sieht sich selbst als vollwertiges ERP-System und deckt sämtliche betriebswirtschaftlichen Prozesse wie Einkauf, Verkauf, Fertigung, Finanzbuchhaltung etc. ab und kann sowohl im Handels- & Produktions- als auch im Dienstleistungsgewerbe eingesetzt werden. Zielgruppe sind kleinere Unternehmen sowie der klassische, kleinere deutsche Mittelstand. Neben der komplett kostenfreien Version existiert auch eine lizenzgebührenpflichtige Enterprise-Edition, die aber ebenfalls Open Source ist. Helium bietet wie die anderen hier vorgestellten professionellen Open Source Applikationen ebenfalls Unterstützung durch verschiedene Subscription-Modelle⁴¹.

Neben Helium existieren mit IntarS, Nuclos und sopen darüberhinaus auch noch weitere ERP-Lösungen, auf die aber nicht näher eingegangen wird. Die sopen bietet zudem ein an das ERP anknüpfendes Customer-Relationship-Management-System⁴².

Für den Bereich des Service Managements bzw. Kundenservice- und Help-Desk-Anwendungen steht mit OTRS ein matures System der c.a.p.e. IT GmbH zur Verfügung, was grundsätzlich alle Branchen abdeckt, die internen oder externen Kundenservice zu liefern haben⁴³. Das können sowohl eine interne IT-Organisationen als auch z.B. ein externer technischer Kundenservice sein. Da dies vor allem auf größere Unternehmen zutrifft, ist die Hauptzielgruppe hier ab 300 Nutzer bis hin zu internationalen Konzernen zu sehen. Da die

⁴¹ Vgl. Helium V (o. J.)

⁴² Vgl. OSB Alliance (o. J.)

⁴³ Vgl. c.a.p.e. IT (2014)

OTRS-Suite ausschließlich als freie Open Source Variante zur Verfügung steht, wird auch hier der Umsatz mit entsprechenden Wartungs- bzw. Serviceverträgen erzielt.

Für weitere betriebliche Abläufe und Aufgaben existieren mit Erpal im Bereich Projekt- und mit agorum im Bereich Dokumentenmanagement ebenfalls Open Source Applikationen deutscher Hersteller.

Bezüglich der Administration, Verwaltung und dem Monitoring von IT-Infrastrukturen hat sich die Univention GmbH als deutscher Marktführer im Bereich Open Source etabliert, dessen Produkte bei vielen befragten Unternehmen im Einsatz sind. Des weiteren kann mit den quelloffenen Backup- und Recovery-Produkten der SEP AG die Datensicherheit und Integrität einer IT-Organisation gewahrt und gewährleistet werden⁴⁴.

Bezüglich Anwendungstools - auch für den Endkunden - hatten die befragten Unternehmen vorrangig mit Open- bzw. LibreOffice Kontakt. Daneben auch teilweise das Open Source Grafik- und Zeichenprogramm GIMP. Besonders im Bereich Programmierung wird aber sehr stark auf Open Source Produkte wie beispielsweise die integrierte Entwicklungsumgebung Eclipse gesetzt.

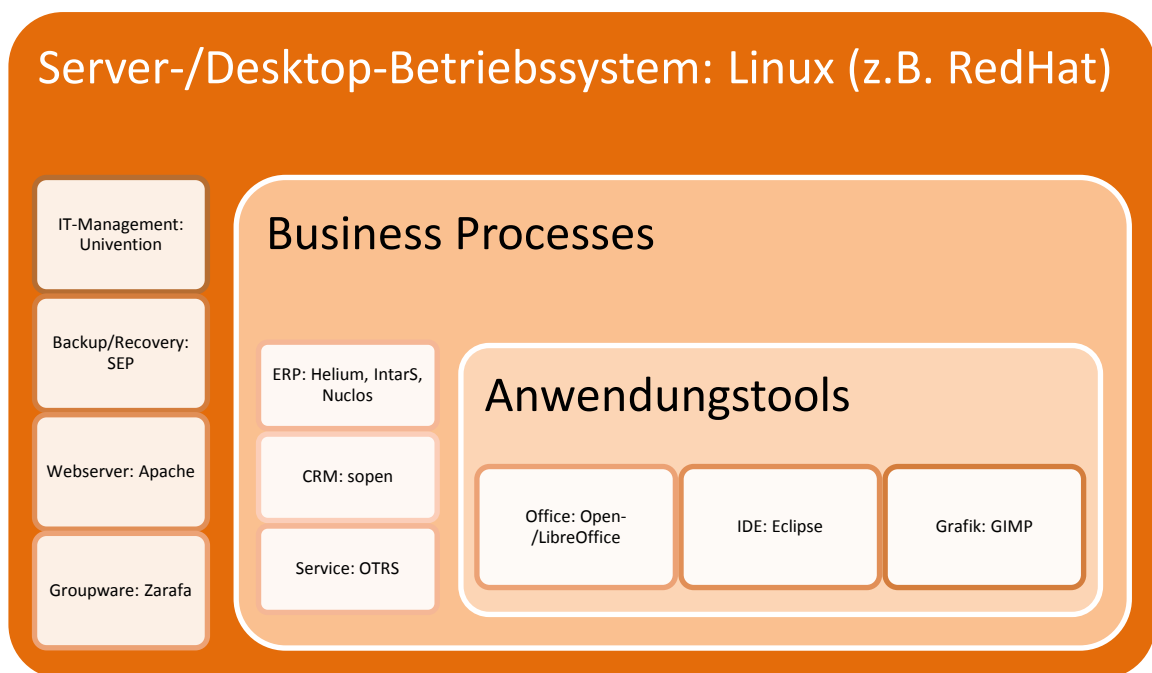


Abb. 11: Beispiel Software Stack

⁴⁴ Vgl. SEP AG (o. J.)

5.2.11 Test- bzw. Demoumgebung

Neben der theoretischen Ausarbeitung der verschiedenen Applikationen und Lösungen im Bereich Open Source ist sinnvoll, auch eine Demo- bzw. Testumgebung mit den wichtigsten Applikationen aufzubauen.

Dabei geht es vorrangig um betriebswirtschaftliche Applikationen wie ERP-Systeme und weniger um Anwendungstools wie z.B. OpenOffice, die ohnehin jeder Nutzer ohne größeren Aufwand installieren und testen kann.

Vielmehr sollten verschiedene Unternehmens- bzw. Branchen-Szenarien entwickelt werden, für die Beispiele eines kompletten Software-Stacks aufgebaut werden.

Dabei bietet es sich an, zum Beispiel für die drei Bereiche Produktion, Handel und Dienstleistung je einen tatsächlichen oder virtuellen Server aufzusetzen, der die für den jeweiligen Sektor relevanten Open Source Produkte beinhaltet.

Um konsequent zu sein, sollte in jedem Fall auf den Einsatz von proprietärer Software verzichtet werden, sofern dies möglich ist.

Als Betriebssystem bietet sich jeweils eine Linux-Distribution an. Außerdem sollte ein für das jeweilige Szenario passendes ERP-System installiert und mit Beispieldaten gefüllt werden. Ein Zimbra-Groupware-Server und ein Backup- bzw. Recovery-System der SEP AG komplettieren die Grundausstattung. Darüberhinaus könnten für die einzelnen Szenarien noch spezifisch passende Anwendungen eingerichtet werden. Dies könnte beispielsweise eine OTRS Help-Desk-Applikation im Handels- oder die Erpal-Projektmanagement-Lösung im Dienstleistungs-Szenario sein.

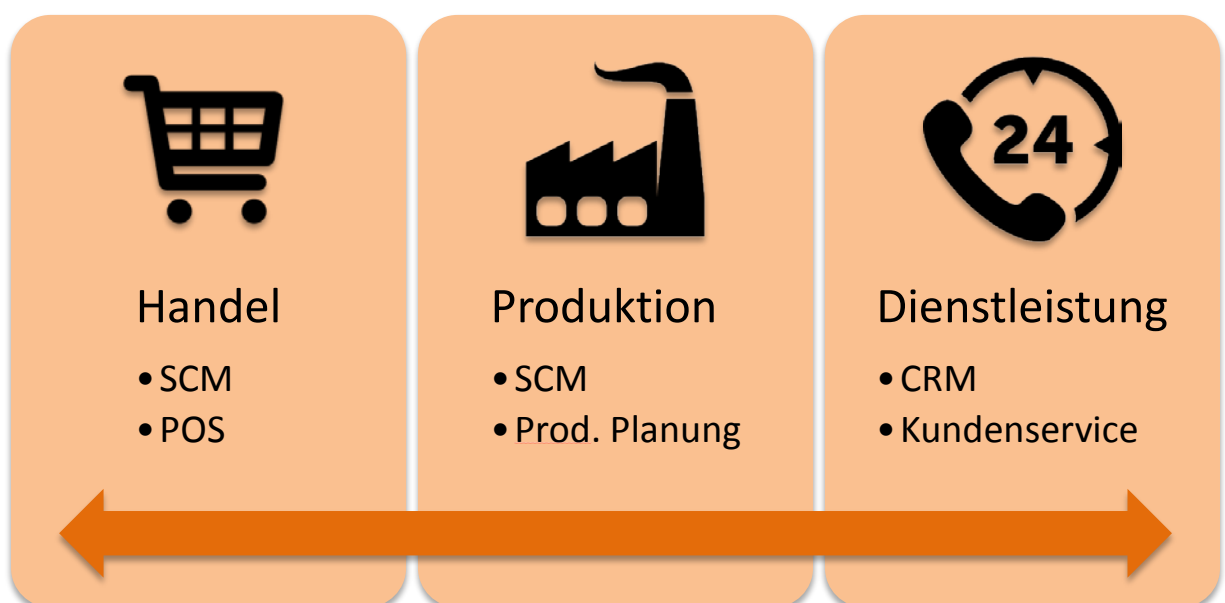


Abb. 12: Demoumgebung für Sektoren

Bezüglich der Einrichtung einer solchen Testumgebung ist es denkbar, die beteiligten Software-Anbieter sowohl bei der Implementierung als auch finanziell um Unterstützung zu bitten.

Bei unzureichenden Ressourcen für verschiedene Szenarien ist zudem denkbar, zunächst einen Server mit einer generischen Ausstattung zu installieren und ggf. im Laufe der Zeit weitere Server für unterschiedliche Anwendungsbeispiele bereitzustellen.

5.3 Öffentlichkeitsarbeit

5.3.1 Referenzmanagement

Ein gutes Referenzmanagement sorgt für eine gute Reputation und kann bei bestehenden- und Neukunden für ein größeres Vertrauen in die angebotene Dienstleistung sorgen. Kunden haben die Möglichkeit sich bei der entsprechenden Referenz über die Qualität der angebotenen Dienstleistung zu informieren.

Gleichzeitig kann das Image eines renommierten Unternehmens, für welches bereits gearbeitet wurde, einen positiven Effekt auf das eigene Unternehmensimage haben. Aus diesem Grund ist es gerade für überwiegend unbekannte Unternehmen wichtig bspw. auf der eigenen Website eine Reihe Unternehmen als Referenz aufzulisten, für welche bereits Dienstleistungen erbracht wurden. Als Beispiel könnte auf der Website des Beratungszentrums ein Kästchen eingerichtet werden, in dem die Unternehmenslogos der besten Referenzkunden aufgezeigt sind.

Veröffentlichungen von Success- und Case Studies mit detaillierteren Informationen helfen dabei mit konkreten Beispielen den Mehrwert der angebotenen Leistungen zu beschreiben. Diese sollten gut sichtbar bspw. zum Download angeboten werden und können als Werbung in anderen Medien, wie bspw. Social Media oder Fachzeitschriften dienen. Besonders die Veröffentlichungen von Case Studies in Fachzeitschriften eignen sich gut, um eine gewünschte Zielgruppe gezielt anzusprechen und sich gleichzeitig eine Reputation aufzubauen.

5.3.2 Social Media / Foren

Das Internet spielt in der heutigen Gesellschaft und der Ökonomie eine tragende Rolle. Wie die Interviews gezeigt haben gilt es besonders auch im Softwareumfeld von Unternehmen meist als erste Informationsquelle bei der Suche nach Lösungen zu gängigen Problemen. Aus diesem Grund sollte sich das Beratungszentrum um eine starke Präsenz in Internetforen zu entsprechenden Softwarezusammenhängen wie bspw. auf „stackoverflow.com“ sowie auf gängigen Social Media Plattformen wie Facebook oder Twitter bemühen.

Besonders Karriereportale wie Xing oder LinkedIn liefern eine gute Möglichkeit die Zielgruppe im B2B-Bereich zu erreichen. Hierzu können bspw. Gruppen eröffnet werden, in welchen regelmäßig auf die eigenen Leistungen, Veranstaltungen oder Veröffentlichungen aufmerksam gemacht wird. Gleichzeitig besteht die Möglichkeit ein großes Netzwerk an Kontakten zu Kunden und Partnern aufzubauen, mit denen in Interaktion getreten werden kann. Mit Hilfe von Posts können diese dann leicht kontaktiert und auf Neuigkeiten aufmerksam gemacht werden.

Eine weitere Möglichkeit ist es, ein Forum auf der eigenen Website zu implementieren und zu pflegen, mit dem Versuch das OS-Beratungszentrum als eine der ersten Suchadressen für Anwender bei OS-Fragestellungen zu etablieren. Hierzu sollten stets aktuelle Themen dazu beitragen eine nach und nach wachsende und aktiv teilnehmende Community aufzubauen.

5.3.3 Präsenz auf Messen

Eine Präsenz auf gängigen Software-Messen, -Konferenzen oder -Kongressen, könnte potenzielle Kunden verstärkt auf das Beratungszentrum aufmerksam machen und bietet eine gute Möglichkeit mit potenziellen Neukunden in Kontakt zu treten.

Aus den Experteninterviews ging hervor, dass viele Unternehmer sich auf Fachmessen über neue Software informieren. Durch das intensive Networking auf solchen Veranstaltungen, könnte die Wahrnehmung des OS-Beratungszentrums gesteigert werden und es können Kontakte zu Kunden und Partnern aufgebaut werden.

Mit „IT & Business“, „CRM-Expo“, „DMS Expo Stuttgart“, „Focusreseller“ oder „Office-Management-Messe“ sind nur einige Beispiele für Messen und Fachkongresse mit Softwarerelevanz zu nennen, die z.B. 2014 in Stuttgart stattfinden bzw. stattgefunden haben.⁴⁵

Um also den Bekanntheitsgrad des OS-Beratungszentrums zu steigern, erachten die Autoren es als sinnvoll in den Auftritt auf solchen Veranstaltungen zu investieren und zumindest auf den wichtigsten auch als Aussteller vertreten zu sein.

5.4 Organisation

5.4.1 Finanzierung

Es gibt verschieden mögliche Geschäftsmodelle für das Source Beratungszentrum.

⁴⁵ www.messen.de

Honorarbasierendes Geschäftsmodell

Bei diesem Geschäftsmodell bekommt das Beratungszentrum, repräsentativ der Berater ein Stunden Satz basierendes Honorar. Vorteil hierbei ist, dass die Kosten direkt zugerechnet und in Rechnung gestellt werden. Sollte sich das beratende Unternehmen gegen eine Open Source Lösung entscheiden, wurde dennoch Geld erwirtschaftet. Der Vorteil für das Beratungszentrum ist, dass die Arbeit in jedem Falle entlohnt wird. Der Nachteil ist, dass eine erfolgreiche Beratung keine monetären Auswirkungen haben. Für die Kunden ist ein solches Geschäftsmodell möglicherweise zunächst einmal ein Hindernis, da die Beratung Geld kostet. Der Vorteil für den Kunden ist, dass er von seinem Berater eine absolut offene, ehrliche und finanziell unabhängige Beratung erwarten kann.

Provisionsbasierendes Geschäftsmodell

Das Beratungszentrum erhält hierbei, bei einer erfolgreichen Vermittlung zu einem Partner, welcher dann ein zum Beispiel ein Implementierungsprojekt durchführt, eine Provision. Diese ist umsatzabhängig und wird entweder vom Partner oder dem Kunden bezahlt. Der Vorteil für den Kunden hierbei ist, dass er zunächst einmal eine kostenlose Beratung erhält. Nur bei Vertragsschluss würde er für diese bezahlen. Der Nachteil ist, dass ein Berater möglicherweise eine teurere Lösung anbieten könnte, als nötig, da diese eine höhere Provision einbringt. Der Vorteil für das Beratungszentrum ist, dass große Projekte viel Umsatz einbringen können: Nachteilig ist, dass wenn es zu keinem Vertragsabschluss kommt, kein Umsatz gemacht wird.

Hybrides Geschäftsmodell

Hierbei handelt es sich um eine Mischform beider zuvor erklärten Geschäftsmodelle. Möglich wäre ab einem gewissen Stundenbetrag an Beratungsdienstleistungen, ein gewisses Honorar abzurechnen. Dadurch ergibt sich eine gewisse finanzielle Absicherung für das Beratungszentrum. Das Beratungszentrum vermittelt dem Kunden einen Partner für die Implementierung oder einer anderen Dienstleistung. Bei Vertragsabschluss zwischen dem Partner und dem Kunden erhält das Beratungszentrum eine Provision, abhängig von dem Vertragsvolumen. Diese wird von dem Partner bezahlt. Zuvor gezahlte Honorarleistungen werden hiermit verrechnet. Honorarleistungen sollten erst dann in Rechnung gestellt werden, wenn eine Entscheidung gefallen ist, um eine komplizierte Rückabwicklung zu ersparen.

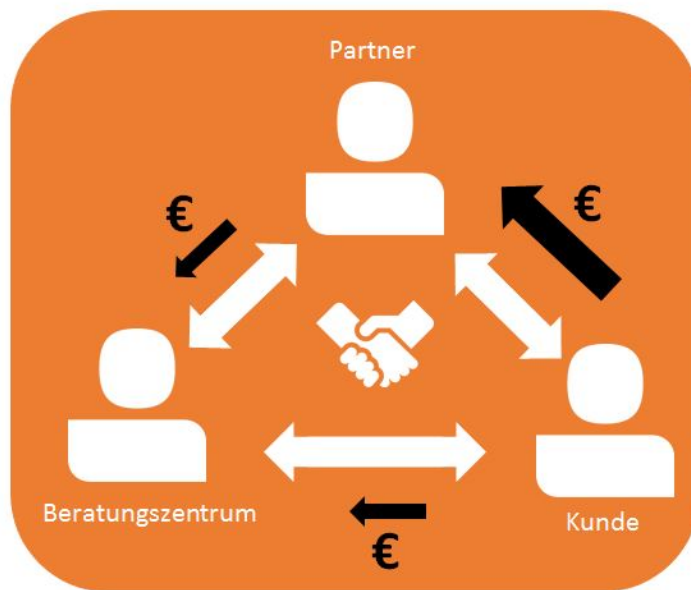


Abb. 13: Geschäftsmodelle

Für das Open Source Beratungszentrum wird empfohlen ein hybrides Geschäftsmodell zu verwenden, um sich finanziell etwas abzusichern, aber den möglichen Kunden gewisse finanzielle Vorteile zu gewähren.

5.4.2 Partnermanagement

Für das Open Source Beratungszentrum ist ein gutes und effektives Partnermanagement unerlässlich. Es werden qualifizierte Anbieter von Open Source Software und damit verbundene Dienstleistungen benötigt, um den Kunden eine vollständige Beratung und vor allem eine Umsetzung zu liefern. Je nach Geschäftsmodell ist dies auch essentiell, weil sonst kein Umsatz generiert werden kann.

Das Ziel des Partnermanagement muss sein, jedes mögliche Kundenbedürfnis mit einem Partner zu befriedigen. Im Idealfall ermöglicht das Open Source Beratungszentrum den Partnern, bei komplexeren Projekten die Vermittlung weiterer eigener Partner. Diese Vermittlung könnte auch ein weiterer Geschäftsbereich des Beratungszentrums werden.

Kommunikation ist bei Partnermanagement sehr wichtig. Umso mehr Austausch zwischen dem Beratungszentrum und dem Partner stattfindet, desto besser kann später der richtige Partner für den Kunden gefunden werden. Partnern muss die Zusammenarbeit attraktiv gemacht werden, da diese zunächst einmal auch eine gewisse, vor allem zeitliche Investition beansprucht. Die Zusammenarbeit mit den Partnern muss für beide Seiten wenig komplex sein, dies erhöht die Attraktivität und senkt auch die Kosten.



Abb. 14: Partnermanagement

Wichtig für den Erfolg ist außerdem den Konflikt zwischen Partnern möglichst gering zu halten. Die Empfehlung hierbei ist, für bestimmte Produkte nur eine begrenzte Anzahl an Partnern in dem gleichen Einzugsgebiet zu qualifizieren.

Eine weitere Möglichkeit für eine Partnerschaft wäre mit Unternehmensberatern, speziell diejenigen, welche Existenzgründerseminare abhalten. Hierbei könnte man, im Rahmen eines solchen Existenzgründerseminars, einen kleinen Workshop über Unternehmenssoftware anbieten. Hierbei wird vorgestellt welche Software die Gründer benötigen um ihren Geschäftsbetrieb zu unterstützen. Sowohl proprietäre Software als auch Open Source Lösungen werden vorgestellt. Es muss ein echter Mehrwert für die Workshop Teilnehmer geschaffen werden. Dies sollte keine Werbeveranstaltung für Open Source oder das Open Source Beratungszentrum werden. Zufriedene Teilnehmer werden dann bei Bedarf den Kontakt aufnehmen, wenn diese den Service in Anspruch nehmen wollen. Kommt es später durch diese Aktivitäten zu einem Kontakt, ist dies ein sehr hoch qualifizierter Lead. Nicht nur Unternehmensberater eignen sich hierfür. Möglich sind auch Kooperationen mit Institutionen wie der Industrie und Handelskammern.

6 Kritische Reflexion / Fazit

Die auf den Ergebnissen der verschiedenen Interviews basierenden Handlungsempfehlungen sollen dem Open Source Beratungszentrum einige Hinweise geben, wie eine entsprechende Dienstleistung aufgebaut werden kann und welche grundlegenden Elemente sie sinnvollerweise beinhaltet sollte. Dabei dienen die Empfehlungen nicht als strikter Plan, der eins zu eins umgesetzt werden soll, sondern lediglich als grundlegende Orientierung und Hilfestellung. Die Umsetzung der einzelnen Empfehlungen bedarf zusätzlicher Planung und Anpassung an die Gegebenheiten und Bedürfnisse des Open Source Beratungszentrums.

Bei der Durchführung der Interviews sind bezogen auf die jeweilige Gruppe unterschiedliche Herausforderungen aufgetreten. Diese und wichtige Schlüssel-Aussagen werden im folgenden je Unternehmenskategorie zusammengefasst:

Open Source Software findet bei Gründern und Startups Anwendung in sehr vielen Bereichen, wie zum Beispiel Servertechnik, Datenbanken und der Programmierung. Sie sehen die Vorteile der Open Source Software vor allem in der meist kostenlosen Anschaffung, in der Flexibilität, sowie in der Möglichkeit die Software um bestimmte Funktionalitäten zu erweitern. Wichtiges Auswahlkriterium ist dabei die Kompatibilität mit anderer bereits verwendeter Software. Kommerzielle Software hingegen kommt vor allem in den Bereichen Textverarbeitung, Kommunikation, Planung sowie Bildbearbeitung zum Einsatz. Software Support spielt zumeist noch keine große Rolle. Bei auftretenden Problemen werden vorwiegend Online-Foren sowie Experten von Hochschulen befragt.

Die befragten Jungunternehmen nutzen Software in den Bereichen Büro, Warenwirtschaft, Design, Kasse, Online-Shop und Website. Mit OpenOffice und Filezilla waren allerdings lediglich zwei Open Source Lösungen in Anwendung. Support spielt für die Unternehmen eine wichtige Rolle, weil so sichergestellt werden muss, dass alle Anwendungen problemlos funktionieren. Für die Informationssuche bei dem Anschaffungsprozess von neuer Software dienen das Internet, Fachzeitschriften, Messen oder der Austausch mit anderen Unternehmen. In Bezug auf Beratungsdienstleistungen zeigten die Befragten geringes Interesse.

Open Source Software ist bei den kleinen und mittleren Unternehmen bekannt und auch im Einsatz. Die Verbreitung ist jedoch nicht besonders groß. Beratung wird aus einer Hand bevorzugt. Die Ursache der geringeren Verbreitung liegt daher bei den IT Lieferanten. Wenn diese, vor allem bei komplexeren Projekten, keine Open Source Lösung anbieten, sind die Unternehmen nicht gewillt unbedingt eine Open Source Lösung in Betracht zu ziehen. Unternehmen mit starker IT affiner Geschäftsführung setzt eher auf Open Source allerdings ist der Beratungsbedarf sehr gering. Eine Beratung käme für diese nur in Frage, wenn diese

sich die Internet Recherche ersparen könnten. Ziel des Beratungszentrum sollte eine Zusammenarbeit mit den etablierten IT Dienstleistern der kleinen und mittleren Unternehmen sein.

Anders gestaltet sich das Bild bei den Anbietern von Open Source Software. Zum einen verwenden diese Unternehmen aufgrund der Tatsache, dass sie selber in jenem Bereich aktiv sind, fast ausschließlich Open Source Produkte. Dabei reichen die Einsatzgebiete von klassischer Anwendungssoftware wie OpenOffice oder GIMP bis hin zu hochkomplexen Geschäftsapplikationen wie ERP- oder CRM-Systeme, Kundenservice-Tools oder Produkte zur Administration von IT-Infrastrukturen und im Bereich Backup/Recovery. Darüberhinaus ist der Bereich der Softwareentwicklung selbst ein sehr großes Einsatzgebiet von Open Source Produkten. Neben Entwicklungsumgebungen geht es dabei auch um Programmier-Bibliotheken und z.B. Webserver. Kommerzielle Software kommt – wenn überhaupt – lediglich im Bereich der Grafik- oder Bildbearbeitung zum Einsatz. Einer Beratungsdienstleistungen stehen die Anbieter von Open Source Software sehr positiv gegenüber und begrüßen, dass das Thema Open Source dadurch verbreitet wird.

Zusammenfassend lässt sich sagen, dass durch die Interviews deutlich wurde, dass das Thema Open Source bei jungen Unternehmen und dem klassischen deutschen Mittelstand eine untergeordnete oder gar keine Rolle spielt. Lediglich Startups und die Anbieter selbst scheinen eine gewisse Affinität für Open Source Produkte aufzuweisen. Umso wichtiger ist es, dieses Thema in Deutschland voranzutreiben, wozu ein entsprechendes Beratungszentrum sicherlich einen Beitrag leisten kann. Mit den in dieser Forschungsarbeit vorgestellten Handlungsempfehlungen sollen erste Grundlagen bezüglich der Entwicklung und Einrichtung eines solchen Zentrums geschaffen werden.

Anhang

Anhang 1: Leitfragebogen.....	45
Anhang 2: Startups Ergebnistabelle.....	46
Anhang 3: Jungunternehmen Ergebnistabelle	46
Anhang 4: Kl. und mittlere Unternehmen Ergebnistabelle	47
Anhang 5: Open Source Anbieter Ergebnistabelle	47

Anhang 1: Leitfragebogen

1. Anwendung

- a. Nutzen Sie OS Software?
 - i. JA:
 1. Welche OS Software verwenden Sie? Seit wann? Erfahrungen?
 2. Welche kommerzielle Software?
 3. Für welche Zwecke? In welchen Bereichen?
 4. Warum OS Software und nicht kommerzielle Software?
 - ii. NEIN:
 1. Was haben Sie als Alternative? Standard/ Individualsoftware? Seit wann? Erfahrungen?
 2. Wieso keine OS Software? Barrieren?
 3. Schon über Verwendung von OS nachgedacht? In wie weit? Wieso nicht?
- b. Welche grundsätzlichen Vor-/Nachteile sehen Sie bei OS Software im Vergleich zu kommerzieller Software?
- c. Überlegen Sie in Zukunft auch/generell OS Software anzuschaffen?

2. Support

- a. Welche Rolle spielt Software Support für Sie?
- b. Welche Art Support bevorzugen Sie? (Persönlich, telefonisch, Forum)
- c. In welchem zeitlichen/finanziellen Umfang wird Support in Anspruch genommen? (allgemein und SW-spezifisch)
- d. Positive/negative Erfahrungen mit Support? (allgemein und SW-spezifisch)
- e. Sind Sie bereit für Support zusätzliche finanzielle Mittel abzustellen?

3. Kaufentscheidungsprozess

- a. Was bewegt Sie dazu neue Software anzuschaffen?
- b. Wie gehen Sie dabei vor?
- c. Welche Arten ziehen Sie in Betracht? (OS, Standard, Individual) Kriterien bei der Auswahl?

4. OS Beratungszentrum (MFG)

- a. Nehmen Sie Beratungsdienstleistung im Bezug auf OS Software bereits in Anspruch? Wenn nicht, würde Sie es in Erwägung ziehen? Wo?
- b. Welche Fragen bewegen Sie am ehesten im Bezug auf OS Software?
- c. Welche Anforderungen hätten Sie an solche eine Beratungsdienstleistung? (Inhaltlich/Organisatorisch/Finanziell)
- d. *Für Anbieter:* Welche Chancen sehen Sie für sich als Anbieter?

5. Spezifisch für Anbieter von OS Software

- a. Welche OS Software bieten Sie an?
- b. Was sind die Hauptanwendungsgebiete/-zielgruppen?
- c. Bieten Sie auch kommerzielle Software an?
- d. Was hat Sie dazu bewegt Ihre Software unter eine OS Lizenz zu stellen?
- e. Welche Lizenzmodelle verfolgen Sie?
- f. Was ist Ihr Geschäftsmodell?

Anhang 2: Startups Ergebnistabelle

		Startup A	Startup B	Startup C	
Anwen-dung	Open Source SW	<ul style="list-style-type: none"> •OS Libraries •DB-Frameworks •Linux 	<ul style="list-style-type: none"> •Nginx •Webserver •Anwendungssoftware •Linux 	<ul style="list-style-type: none"> •Nginx •Linux •Node.js •CouchDB 	<ul style="list-style-type: none"> •Grunt •GitHub •Travis CI •Homebrew
	Kommerzielle SW	<ul style="list-style-type: none"> •MS Office •Podio (PM) 	<ul style="list-style-type: none"> •MS Office 	Webbasiert: <ul style="list-style-type: none"> •Talky •Hangouts •Google Apps/Docs •Basecamp •Slack 	Desktop: <ul style="list-style-type: none"> •Mac OS •Skype •Photoshop •Sublime Text •Chrome/Safari
Support		<ul style="list-style-type: none"> •Ausschließlich Communities und Experten von Universitäten 	<ul style="list-style-type: none"> •Support momentan keine Frage •Communities 	<ul style="list-style-type: none"> •"Do it yourself"-Mentalität •Bei Bezahlsupport wird gutes Ergebnis erwartet •Für Unternehmen wichtig 	
Kaufentscheidung		<ul style="list-style-type: none"> •aufgetretenes Problem lösen •Leistung vs. Preis •sowohl OS und kommerzielle SW in Betracht ziehen •Informationsbeschaffung durch Google, Blogs, Tests 	<ul style="list-style-type: none"> •Suche nach Standardvariante •Kompatibilität •Anforderungen müssen erfüllt sein (OS oder kommerziell) 	<ul style="list-style-type: none"> •Hauptsächlich Informationsprozess über Communities (z.B. Twitter) 	
Anforderung Beratungsdienstleistung		<ul style="list-style-type: none"> •kostenlose Kennenlernertermine •schnelle Reaktionszeit •Kompetente Beratung •Vermittlung von Experten 	<ul style="list-style-type: none"> •Zielgruppenspezifisch •Für Enduser, Entwickler und Entscheider •Attraktives Angebot im Internet 		

Anhang 3: Jungunternehmen Ergebnistabelle

		Jungunternehmen 1	Jungunternehmen 2	Jungunternehmen 3	Jungunternehmen 4	Jungunternehmen 5
Anwendung	Open Source SW	keine	<ul style="list-style-type: none"> - Filezilla SDP Client - Internetbrowser: Firefox 	Open Office	keine	keine
	Kommerzielle SW	<ul style="list-style-type: none"> •MS Office •Lexware 	<ul style="list-style-type: none"> •MS Office •Warenwirtschaftssystem •Design Software 	Warenwirtschaft: Haufe Webbrowser: Firefox	Kasse: Vectro Commander Büro: MS Office Online Shop: Gambio	Büro: MS Office Warenwirtschaft: Lexware
Support		<ul style="list-style-type: none"> •Online Support der direkt auf PC zugreifen kann 	<ul style="list-style-type: none"> •"Do it yourself"-Mentalität 	Telefonhotline, die pro Minute abgerechnet wird	IT-Laden im Dorf mit monatl. Basisvertrag	Telefonhotline on Demand
Kaufentscheidung		<ul style="list-style-type: none"> •aufgetretenes Problem lösen •Leistung vs. Preis •sowohl OS und kommerzielle SW in Betracht ziehen •Informationsbeschaffung durch Google, Blogs, Tests 	<ul style="list-style-type: none"> •Suche nach Standardvariante •Kompatibilität •Anforderungen müssen erfüllt sein (OS oder kommerziell) •Testversion 	<ul style="list-style-type: none"> •Informationsprozess im Internet •Anforderungen müssen erfüllt sein (OS oder kommerziell) •Kosten/Leistung 	<ul style="list-style-type: none"> •Bedarf weil etwas fehlt im Prozess •Angebote einholen •befreundete Unternehmen fragen 	<ul style="list-style-type: none"> •Bedarf •Infosuche (Internet, Zeitschriften, Messen) •Auswahl
Anforderung Beratungsdienstleistung		<ul style="list-style-type: none"> •kommerzielle Beratung vor Anschaffung sämtlicher Software •kein Interesse an Beratung, da kein Bedarf zu Neuerungen 	<ul style="list-style-type: none"> •in wenigen Worten den Mehrwert einer Software aufzeigen und Argumentation warum gerade diese die beste Lösung ist •sollte kostenfrei sein 	generelles Interesse an Beratungs-DL für Open Source	Kein Interesse an Beratung	Kein Interesse an Beratung

Anhang 4: KI. und mittlere Unternehmen Ergebnistabelle

		KMU A	KMU B	KMU C
Anwendug	Open Source SW	<ul style="list-style-type: none"> •NetBeans u.A. •REDmine •MySQL •Linux (Server) 	<ul style="list-style-type: none"> •Raspbian 	<ul style="list-style-type: none"> •Greenshot
	Kommerzielle SW	<ul style="list-style-type: none"> •MS Office •Windows (Desktop) •Mac OS •Adobe Produkte 	<ul style="list-style-type: none"> •MS Office •Windows (Desktop) •VM Ware •Windows (Server) •Visual Studio 	<ul style="list-style-type: none"> •MS Office •Windows (Desktop) •Microsoft (Server) •VM Ware •SAP ERP
Support		<ul style="list-style-type: none"> •Communities, Dokumentationen, bei komplexeren Projekten Implementationspartner, Support darf bei entsprechender Leistung auch etwas mehr Geld kosten 	<ul style="list-style-type: none"> • IT Lieferant, bei kleinen Problemen Recherche, Support vor alle in der Anfangszeit wichtig 	<ul style="list-style-type: none"> •Service Desk oder IT Lieferant, falls eigen IT Abteilung nicht lösen kann, keine Budgetprobleme für Suupordienstleistungen
Kaufentscheidung		<ul style="list-style-type: none"> •Kommerziell oder OS egal, Kosten Nutzen Verhältnis entscheidet, für gute Software bereit viel Geld auszugeben •Informationsbeschaffung durch Internet, Reviews und Demos wichtig 	<ul style="list-style-type: none"> •Bestes Preis Leistungs Verhältnis •Kompatibilität •Informationsbeschaffung durch Internet und IT Lieferant, Erfahrungsberichte und Demos wichtig 	<ul style="list-style-type: none"> •aufgetretener Bedarfbedrüdigen •Leistung vs. Preis •sowohl OS und kommerzielle SW in Betracht ziehen •Berater und IT Dienstleister als auch Internet Recherche wichtig
Anforderung Beratungsdienstleistung		<ul style="list-style-type: none"> •kein Bedarf, da hohe IT Affinität 	<ul style="list-style-type: none"> •Muss finanzielle Vorteile bringen, da dies ein Mehraufwand mit sich bringt, mehrere IT Dienstleister zu haben 	<ul style="list-style-type: none"> • Eher uninteressant, da IT Berater schon im Haus sind

Anhang 5: Open Source Anbieter Ergebnistabelle

		OS Anbieter 1	OS Anbieter 2	OS Anbieter 3
Anwendug	Open Source SW	<ul style="list-style-type: none"> •Linux •Zarafa •PostgreSQL •OpenOffice/LibreOffice 	<ul style="list-style-type: none"> •Linux •Zarafa •LibreOffice •MediaWiki •OTRS •Piwik •Eclipse 	<ul style="list-style-type: none"> •Linux •Zarafa •Backup •Webserver
	Kommerzielle SW	<ul style="list-style-type: none"> •Photoshop 	<ul style="list-style-type: none"> •TeamViewer 	<ul style="list-style-type: none"> •keine
Support		<ul style="list-style-type: none"> •Do it yourself •Community •Hersteller-Support 	<ul style="list-style-type: none"> •Verlässliche Dienstleister •NICHT Community 	<ul style="list-style-type: none"> •Eigener Support •Teilweise Herstellersupport •Quellcode betrachten
Kaufentscheidung		<ul style="list-style-type: none"> •Zunächst Funktionalität, dann Kosten •Im kritischen Bereich Open Source 	<ul style="list-style-type: none"> •Sowohl proprietär als auch OS •Bestes Kosten/Nutzen-Verhältnis •Wichtig: solides Unternehmen im Hintergrund 	<ul style="list-style-type: none"> •Tendenziell eher OS •Außer wenn proprietär besser •OS wenn kritisch (z.B. Security)
Anforderung Beratungsdienstleistung		<ul style="list-style-type: none"> •Frage/Vorurteile bzgl. OS müssen geklärt werden •OS „Komplettpaket“ für Gründer 	<ul style="list-style-type: none"> •Klärung, wie OS Anbieter Geld verdienen •Unternehmen Einstiegshürden in Thema OS nehmen 	<ul style="list-style-type: none"> •Unternehmen in Startphase mit OS konfrontieren •Verbreitung generell vergrößern
Zusatz: OS Anbieter		<ul style="list-style-type: none"> •Produkt: ERP System •Zielgruppe: kleine bis mittlere Unternehmen •Branchen: Handel, Produktion, Dienstleister •Geschäftsmodell: Subscription / Lizenzkauf 	<ul style="list-style-type: none"> •Produkt: Kundenservice •Zielgruppe: mittlere und große Unternehmen •Branchen: alle, wo Serviceprozesse relevant •Geschäftsmodell: Wartungs-/Serviceverträge 	<ul style="list-style-type: none"> •Produkt: Integration von OS-Software •Zielgruppe: Mittelstand (< 1000 Mitarbeiter) •Branchen: alle

Quellen

Literaturverzeichnis

- Albers, S./Klapper, D. et al. (2009):** Methodik der empirischen Forschung, 3., überarb. und erw. Aufl, Wiesbaden: Gabler.
- Atteslander, P. (2003):** Methoden der empirischen Sozialforschung, 10., neubearb. und erw. Aufl, Berlin: Walter de Gruyter.
- Bortz, J./Döring, N. (2006):** Forschungsmethoden und Evaluation für Human- und Sozialwissenschaftler, Mit 156 Abbildungen und 87 Tabellen, 4. überarbeitete Aufl, Heidelberg: Springer Medizin Verlag.
- Flick, U. (1995):** Qualitative Forschung, Theorie, Methoden, Anwendung in Psychologie und Sozialwissenschaften, Orig.-Ausg, Reinbek bei Hamburg: Rowohlt Taschenbuch-Verl.
- Hillmann, K.-H./Hartfiel, G. (1994):** Wörterbuch der Soziologie, 4., überarbeitete und erg. Aufl, Stuttgart: A. Kröner.
- Hitzler, R./Honer, A. et al. (1994):** Expertenwissen, Die institutionalisierte Kompetenz zur Konstruktion von Wirklichkeit, Opladen: Westdt. Verl.
- Homburg, C. (2000):** Quantitative Betriebswirtschaftslehre, Entscheidungsunterstützung durch Modelle ; mit Beispielen, Übungsaufgaben und Lösungen, 3., überarb. Aufl, Wiesbaden: Gabler.
- Kelle, U. (2008):** Die Integration qualitativer und quantitativer Methoden in der empirischen Sozialforschung, Theoretische Grundlagen und methodologische Konzepte, 2., Aufl, Wiesbaden: VS Verlag für Sozialwissenschaften.
- Mayer, H. O. (2008):** Interview und schriftliche Befragung, Entwicklung, Durchführung und Auswertung, 4., überarb. und erw. Aufl, München, Wien: Oldenbourg.
- Meuser, M./Nagel, U. (2009):** Methoden der vergleichenden Politik- und Sozialwissenschaft, Neue Entwicklungen und Anwendungen, 1. Aufl, Wiesbaden: VS Verlag für Sozialwissenschaften.
- Mieg, H./Brunner, B. (2004):** Experteninterviews, in Schweizerische Zeitschrift für Soziologie, Nr. 30, S.199-222
- Mitchell, V. W. (1994):** Using industrial key informants, Some guidelines.
- Pfadenhauer, M. (2009):** Qualitative Marktforschung, Konzepte - Methoden - Analysen, 2., überarb. Aufl, Wiesbaden: Gabler.
- Sprondel, W. M. (1979):** Experte und Laie, Zur Entwicklung von Typenbegriffen in der Wissenssoziologie, Stuttgart: Enke.

Verzeichnis der Internetquellen

- Bauer, A./Mulley, U. (2014):** Unterschied und Gemeinsamkeiten qualitativer und quantitativer Forschung, <http://www.mahara.at/artefact/file/download.php?file=17388&view=681>, Abruf: 14.03.2014.
- Bibliographisches Institut GmbH (2013):** FAQ, die, <http://www.duden.de/rechtschreibung/FAQ>, Abruf: 05.07.2014
- c.a.p.e. IT (2014):** c.a.p.e. IT, OTRS, <http://www.cape-it.de/de/service-management-system-otrs.html>, Abruf: 04.07.2014
- Dr. Diedrich, O (o. J.):** Die Woche: Was ist Open Source: <http://www.heise.de/open/artikel/Die-Woche-Was-ist-Open-Source-222121.html>, Heise online, Abruf: 27.06.2014
- Dr. Klodt, H (o. J.):** Mittelstand: <http://wirtschaftslexikon.gabler.de/Archiv/71994/mittelstand-v9.html>, Gabler Wirtschaftslexikon, Abruf: 23.06.2014
- Duale Hochschule Baden-Württemberg Stuttgart (o. J.):** Duale Hochschule Baden-Württemberg Stuttgart, <http://www.dhbw-stuttgart.de/home.html>, Abruf: 03.07.2014
- Experts Exchange, LLC. (2014):** The Network for Technology Professionals, <http://www.experts-exchange.com/>, Abruf: 03.07.2014
- Früh, W. (2012):** Zusammenfassung Frueh Empirische Forschung, <http://de.scribd.com/doc/93083479/103-Zusammenfassung-Frueh-Empirische-Forschung>, Abruf: 13.03.2013.
- Helium V (o. J.):** Helium V, Lizenzmodelle, <http://www.heliumv.com/heliumv-Preise-3.html>, Abruf: 04.07.2014
- Hochschule der Medien (2014):** Hochschule der Medien, <http://www.hdm-stuttgart.de/>, Abruf: 03.07.2014
- Institut für Medien und Bildungstechnologie (2014):** Der Empiriebegriff, <http://qsf.e-learning.imb-uni-augsburg.de/node/505>, Abruf: 14.03.2014.
- Kutscher, N. (2004):** Was sind qualitative und quantitative Forschungsmethoden?, <https://www.unibielefeld.de/Universitaet/Einrichtungen/Zentrale%20Institute/IWT/FWG/Jugend%20online/qualitativ.html>, Abruf: 14.03.2014.
- Langenbacher-König, R. (2013):** Experteninterview, <https://www.ph-freiburg.de/de/quasus/einstiegstexte/erhebungsinstrumente/interview/experteninterview.html>, Abruf: 15.03.2014.
- OSB Alliance (o. J.):** OSB Alliance, Mitglieder, <http://www.osb-alliance.de/mitglieder>, Abruf: 04.07.2014
- Pez (2004):** Empirische Sozialforschung, http://www.uni-lueneburg.de/fallstudie/downloads/Vortrag_1_Sozialforschung.pdf, Abruf: 14.03.2014.
- SEP AG (o. J.):** SEP sesam, Funktionen & Vorteile,

<http://www.sep.de/de/produkte/entdecke-sep-sesam/funktionen-vorteile/>, Abruf: 04.07.2014

Springer Gabler Verlag (o. J.a): Start-up-Unternehmen, <http://wirtschaftslexikon.gabler.de/Archiv/427/start-up-unternehmen-v7.html>, Abruf: 03.07.2014

Springer Gabler Verlag (o. J.b): FAQ, <http://wirtschaftslexikon.gabler.de/Archiv/78644/faq-v9.html>, Abruf: 07.07.2014

Stack Exchange, Inc. (2014a): Stackoverflow, <http://stackoverflow.com/>, Abruf: 03.07.2014

Stack Exchange, Inc. (2014b): Room, <http://chat.stackoverflow.com/>, Abruf: 03.07.2014

Universität Stuttgart (2014): Universität Stuttgart, Willkommen an der Universität Stuttgart, <http://www.uni-stuttgart.de/home/>, Abruf: 03.07.2014

Textanalyse am Beispiel von Beschwerdeschreiben

Umsetzung mit PostgreSQL

Schriftliche Ausarbeitung
im Rahmen der Lehrveranstaltung „Integrationsseminar“
Kompetenzzentrum Open Source (KOS)

Vorgelegt von

Reutter, F., Grötzinger, H.,
Kaiser, M., Putze, M.

am 14.07.2014

Fakultät Wirtschaft
Studiengang Wirtschaftsinformatik
WWI2011v

Inhaltsverzeichnis

Abkürzungsverzeichnis	III
Abbildungsverzeichnis.....	III
1 Einleitung	1
2 Aufgabenstellung & Ausgangslage.....	2
3 Theoretische Grundlage & Vorüberlegungen	3
3.1 BayesKlassifikator	3
3.1.1 Beschreibung	3
3.1.2 Berechnung.....	3
3.2 Sentiment Analyse	6
3.3 Phraseanalyse	6
3.4 PostgreSQL	7
4 Realisierung	9
4.1 Projektablauf	9
4.2 Planung & Projektdurchführung.....	10
4.2.1 Vorgehensweise	10
4.2.2 Herausforderungen	11
4.2.3 Beschreibung der Anwendung	13
4.2.4 Installation der Anwendung	15
5 Ausblick.....	17
5.1 Grammatikanalyse	17
5.2 Abschaffung analoger Medieneingänge	17
5.3 Datenbank Routinen	17
5.4 Verwendung kommerzieller Lösungen	18
5.5 Rechtschreibprüfung	18
6 Fazit.....	19
Quellenverzeichnisse	20

Abkürzungsverzeichnis

DHBW	= Duale Hochschule Baden-Württemberg
GUI	= Graphical User Interface
OCR	= Optical Character Recognition

Abbildungsverzeichnis

Abbildung 1: Zahlenbeispiel – Trainingsdaten	4
Abbildung 2: Senti Wortschatz Statistiken	6
Abbildung 3 Übersicht Projektplan.....	9
Abbildung 4 Feasibility Study am Beispiel Shared Service Center nach PwC.....	10
Abbildung 5 Screenshot der Anwendung.....	13
Abbildung 6 Info-Fenster zum Bayes-Klassifikator	14
Abbildung 7 Info-Fenster zum Sentiment-Faktor	14
Abbildung 8 Info-Fenster zu gefundenen Phrasen.....	15

1 Einleitung

Die Geschäftstätigkeit von Versicherungsunternehmen führt zu einem hohen Ausmaß an Kundenkontakt. Kern der Geschäftstätigkeit ist es, dass Kunden Versicherungen abschließen und entsprechende Leistungen in Anspruch nehmen. Dieser Umstand führt unweigerlich dazu, dass sich Kunde und Versicherungsunternehmen untereinander austauschen. Dieser Austausch kann auf unterschiedlichsten Wegen stattfinden, so kann persönlich vor Ort in einer Filiale, am Telefon, per Email oder per Brief kommuniziert werden.

Die Bundesanstalt für Finanzdienstleistungsaufsicht (BaFin) ist dazu verpflichtet jährlich eine Beschwerdestatistik eines jeden Versicherungsunternehmens anzufertigen. Aus diesem Umstand ergibt sich die Verpflichtung der einzelnen Versicherungsunternehmen Beschwerde-schreiben die sie erhalten an die BaFin weiterzuleiten.

Eine automatisierte Erkennung, ob es sich bei einem Schreiben um eine Beschwerde handelt oder nicht, kann hier für Versicherungsunternehmen einen Zeit- und somit auch eine Kostenersparnis bedeuten. Der Auftraggeber hat Zusammenhang erkannt und sich um den Themenkomplex der Beschwerdenerkennung näher zu untersuchen, entschlossen mit dem KOS- Projekt an der Dualen Hochschule Baden-Württemberg (DHBW) zusammenzuarbeiten.

2 Aufgabenstellung & Ausgangslage

Wie die meisten Versicherungsunternehmen erhält der Auftraggeber täglich eine Vielzahl an Briefen. Mitunter kann eine Beschwerde Grund für ein solches Schreiben sein. Wie oben bereits beschrieben besteht die Pflicht solche Beschwerdeschreiben an die BaFin weiterzuleiten, aber auch unternehmensintern ist ein Erkennen der Beschwerdeschreiben sinnvoll, um solche Dokumente ohne Zeitverlust an die richtigen Stellen weiterzuleiten.

Schreiben, die bei dem Auftraggeber in analoger Form eingehen, werden beim Auftraggeber wie folgt behandelt: Die Briefe werden automatisiert geöffnet und eingescannt. Zusätzlich wird die erste Seite jedes Schreibens mit Hilfe eines Texterkennungsprogramms (OCR-Software) in eine Computerlesbare Form übersetzt. Die eingegangenen Schreiben werden daraufhin versucht an die richtigen Mitarbeiter weiterzuleiten.

Eine große Erleichterung in diesem Zusammenhang würde eine automatisierte Erkennung von Beschwerden liefern, deshalb soll untersucht werden inwiefern mit Hilfe von Textanalysemethoden Beschwerdeschreiben als solche zuverlässig erkannt werden können. Im Rahmen des KOS-Projekts an der DHBW soll der Sachverhalt von zwei Gruppen näher untersucht werden. Die eine Gruppe soll dabei das Datenbanksystem MongoDB verwenden, während die andere Gruppe mit PostgreSQL arbeiten soll. Diese Ausarbeitung erläutert die Ergebnisse der Gruppe, die PostgreSQL verwendet hat.

Ergebnis dieser Projektarbeit soll sein, Lösungsansätze im Bereich der Beschwerdenerkennung zu erarbeiten und diese zu dokumentieren. Fragestellungen sind dabei:

- Wie können Beschwerdeschreiben automatisiert erkannt werden?
- Wie lässt sich diese Automation mit bestimmten Datenbanksystemen realisieren? (PostgreSQL / MongoDB)

Vorerst könnte der zukünftige Arbeitsvorgang beim Eingang von Schreiben beim Auftraggeber dann wie folgt aussehen:

Schreiben gehen wie bisher auch beim Unternehmen ein, werden eingescannt und von der OCR Erkennung interpretiert. Der computerlesbare Teil wird mit Hilfe eine neu entwickelten Programms untersucht und es angezeigt ob es sich um eine Beschwerde oder um keine Beschwerde handelt. Ein Mitarbeiter überprüft in der Anfangsphase noch das Ergebnis dieser Untersuchung. Das heißt der Mitarbeiter bekommt das Schreiben mit dem Ergebnis des Programms vorgelegt. Aufgrund dieses Ergebnisses kann der Mitarbeiter das Schreiben wesentlich schneller überfliegen und die Prognose des Programms bestätigen oder dementieren. Die Ergebnisse dieser manuellen Kategorisierung kann dafür verwendet werden das Programm weiter zu optimieren.

3 Theoretische Grundlage & Vorüberlegungen

Auf das wesentliche herunter gebrochen, erfordert die Aufgabenstellung das Einordnen von Textdokumenten in die Gruppen „Beschwerde“ und „keine Beschwerde“. Ebendieses Einordnen wird häufig durch Data-Mining, dem „Schürfen“ nach Informationen, realisiert. Eine weit verbreitete Möglichkeit zur Klassifikation von Daten, ist der Bayes-Klassifikator. Bei den zu klassifizierenden Daten handelt es sich in diesem Fall um einzelne Wörter.

3.1 BayesKlassifikator

3.1.1 Beschreibung

Bei der Klassifizierung durch das Bayes wird jedes Objekt (in diesem Fall jedes Wort) mit einer bestimmten Wahrscheinlichkeit einer Klasse zugeordnet (Class-Membership-Probability). Dazu ist es nötig im Vorfeld sog. Trainingsdaten, für die die Klassifizierung bekannt ist, auszuwerten. Da es sich hier um eine Fallunterscheidung handelt, liegen „nur“ zwei Klassen vor, was den Prozess vereinfacht und die Genauigkeit der Aussage erhöht.

Der Bayes-Klassifikator wird häufig im Zusammenhang mit Text-Analysen verwendet, um zum Beispiel die Stimmung oder die Themen in Beiträgen auf sozialen Netzwerken zu analysieren. Vor allem aber auch bei der Erkennung von Spam E-Mails ist diese Methode grundlegend.

Vorteil des Bayes-Klassifikators ist, dass er keine harten Aussagen wie „true / false“ trifft, sondern eine Wahrscheinlichkeit für die Zuordnung zu einer Klasse zurück gibt. Diese Wahrscheinlichkeit wird mit jedem weiteren analysierten Dokument genauer, da sich Trainingsdaten mit Lerndaten verbinden lassen. Daher gehört die Klassifikation nach Bayes zum Bereich „supervisedlearning“ (überwachtes lernen) beim Data-Mining.

3.1.2 Berechnung

Satz von Bayes:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

Der Satz von Bayes gibt Auskunft über die Wahrscheinlichkeit, dass ein Ereignis A eintritt, unter der Bedingung, dass ein Ereignis B eingetreten ist. Für die Elemente gilt hierbei:

$P(A|B)$ = Wahrscheinlichkeit für Ereignis A, unter der Bedingung, dass Ereignis B eingetreten ist.

$P(B|A)$ = Wahrscheinlichkeit für Ereignis B, unter der Bedingung, dass Ereignis A eingetreten ist.

$P(A)$ = A-priori-Wahrscheinlichkeit für A

$P(B)$ = A-priori-Wahrscheinlichkeit für B

Theoretisches Beispiel: Klassifizierung von Spam/nicht Spam (Ham) E-Mails

$P(A)$ entspricht dem Verhältnis von Spam Mails im Verhältnis zu allen vorhandenen Mails in der Trainingsdatenbank. Für $P(B)$ gilt dasselbe analog.

$P(B|A)$ ist die Wahrscheinlichkeit, dass ein Wort B in einer Spam-Mail auftaucht (Anzahl der Spam-Mails mit dem Wort B im Verhältnis zur Anzahl aller Spam-Mails)

Da E-Mails in der Regel nicht nur aus einem Wort bestehen, muss $P(B|A)$ natürlich für jedes einzelne Wort berechnet werden.

Zahlenbeispiel

Abbildung 1 zeigt die zu Grunde liegenden Trainingsdaten, auf deren Basis die Bewertung einer weiteren E-Mail geschehen soll.

Spam-Mails:	Ham-Mails:
offer is secret	play sports today
click secret link	went play sports
secret sports link	secret sports event
	sports is today
	sports costs money

Abbildung 1: Zahlenbeispiel – Trainingsdaten

Zu klassifizierender Satz: „todayissecret“

$$\frac{\frac{3}{8} * \frac{0}{9} * \frac{1}{9} * \frac{2}{9}}{- " - + \frac{5}{8} * \frac{2}{15} * \frac{1}{15} * \frac{1}{15}}$$

A-priori-Wahrscheinlichkeiten: 3 aus 8 Trainingsdateien sind Spam, 5 aus 8 sind Ham.

Wort für Wort:

„today“ taucht in den 9 Trainingswörtern für Spam 0-mal auf, in den 15 für Ham 2-mal.

„is“ taucht in den 9 Trainingswörtern für Spam 1-mal auf, in den 15 für Ham 1-mal.

„secret“ taucht in den 9 Trainingswörtern für Spam 2-mal auf, in den 15 für Ham 1-mal.

Bei der Berechnung der Wahrscheinlichkeit fällt auf, dass durch das Fehlen von Treffern für „today“ im Katalog für Spam das Produkt über dem Bruchstrich 0 ergibt, woraus eine Wahrscheinlichkeit von 0% entstehen würde. Da aber abgesehen von diesem einen Wort trotzdem eine Aussage getroffen werden kann, wendet man nun die sog. Laplacesche Glättung an. Dabei wird dem Zähler ein Wert n hinzugefügt und dem Nenner das Produkt aus n und den verschiedenen Ausprägungen der Klasse. N kann in diesem Fall klein ($n=1$) gewählt

werden, da es nur darum geht, null als Faktor zu verhindern. Dies wird für alle Einzelwahrscheinlichkeiten durchgeführt.¹

Konkret formuliert: 3 aus 8 Trainingsmails sind Spam. Es gibt zwei Klassen, Spam und Ham.

Daraus folgt: $\frac{3+1}{8+(1*2)} = \frac{4}{10} = \frac{2}{5}$

Analog gilt für die Wahrscheinlichkeiten der einzelnen Wörter: „today“ hat eine Grundwahrscheinlichkeit von $\frac{0}{9}$. Das Wörterbuch beinhaltet 12 verschiedene Wörter, was eine geglättete Wahrscheinlichkeit von $\frac{0+1}{9+(1*12)} = \frac{1}{21}$ ergibt.

Für die gesamte Berechnung ergibt sich also folgender Term:

$$\frac{\frac{2}{5} * \frac{1}{21} * \frac{2}{21} * \frac{4}{21}}{- " - + \frac{3}{5} * \frac{3}{27} * \frac{2}{27} * \frac{2}{27}}$$

Die Wahrscheinlichkeit, dass das Dokument mit dem Inhalt „todayissecret“ eine Spam-Nachricht ist, liegt damit bei 0,4858, oder 48,58%.

¹ Schira 2005

3.2 Sentiment Analyse

Sentiment kommt aus dem französischen und bedeutet so viel wie Stimmung, Empfindung oder Gefühl. Bei der Sentiment Analyse werden die einzelnen Wörter eines Dokuments mit denen in einer Datenbank verglichen, deren Stimmung bereits im Voraus festgestellt wurde.

Als Datenbasis bedient sich die Anwendung zweier Dateien eines Projekts der Universität Leipzig mit dem Namen „SentiWS“ (Sentiment Wortschatz). Die beiden Dateien sind aufgeteilt in positive und negative Begriffe. In jeder Datei steht die Grundform eines Wortes, seine verschiedenen Formen und ein zugeordneter Sentiment-Wert. Dieser Wert liegt zwischen -1 und +1, wobei -1 als sehr negativ und +1 als sehr positiv zu betrachten ist.

Zählt man alle Ausprägungen so ergibt sich für je positiv und negativ ein ungefähre Wortschatz von 14000 Wörtern (Siehe Abbildung 2).

		Positive	Negative
Adjectives	Baseforms	784	698
	Inflections	11,101	9,992
Adverbs	Baseforms	6	4
	Inflections	0	0
Nouns	Baseforms	548	686
	Inflections	649	979
Verbs	Baseforms	312	430
	Inflections	2,249	2,843
All	Baseforms	1,650	1,818
	Inflections	13,999	13,814

Abbildung 2: Senti Wortschatz Statistiken

Mithilfe von statistischen Kennzahlen (Summe, Durchschnitt,...) lässt sich aus den einzelnen Sentiment-Werten für die Wörter ein Gesamtwert ermitteln, anhand dessen sich die Stimmung des Dokuments erkannt werden kann.

3.3 Phraseanalyse

Die „WordClusterAnalyse“ beruht auf der Idee anhand von gängigen Phrasen die Bedeutung eines Textes zu definieren. Die Phrasenerkennung kann wichtige Erkenntnisse für spätere Analysen liefern. Sie liefert bei einem Treffer eine relativ sichere Auskunft über die Bedeutung eines Textes. Wird mit der Phrasenanalyse nur überprüft ob die Bedeutung eines Textes mit einer zuvor aufgestellten Richtung übereinstimmt oder nicht, so kann die Phrasenerkennung den Text sehr genau ein kategorisieren. Für eine Phrasen-/Word-Cluster-Analyse ist es gut die Wörter in ihrer Grundform abzuspeichern und lediglich ihre Beziehung zueinander abzuspeichern. Dadurch dass die Wörter in ihrer Grundform gespeichert werden, können die Wörter in beliebiger Konjunktion in weiteren Texten vorkommen und werden immer als

die eigentliche Phrase erkannt.² Es ist denkbar zudem mit abzuspeichern wie häufig eine Phrase in einem Text gefunden wurde, wie viele Texte durchsucht wurden und wie häufig eine Text mit gefundenen Phrasen auch ein Text der gewünschten Kategorie war. Mit diesen Kennzahlen kann die bedingte Wahrscheinlichkeiten, die aussagt mit welcher Wahrscheinlichkeit eine Text ein Text einer gewünschten Kategorie ist wenn darin eine oder mehrere Phrasen der Kategorie gefunden wurden. Dazu wird betrachtet wie Wahrscheinlich es grundsätzlich ist, dass eine Phrase in einem Text vorkommt. Diese Wahrscheinlichkeit wird mit der Wahrscheinlichkeit, dass es sich bei dieser Phrase um eine Phrase in einem Text mit der gewünschten Kategorie handelt, in Verbindung gesetzt. So können Phrasen gewichtet werden, um dadurch eine genauere Aussage treffen zu können ob es sich bei einer Erkennung auch um einen positiven Treffer handelt.

3.4 PostgreSQL

„PostgreSQL ist ein objektrelationales Datenbankmanagementsystem (ORDBMS), das als Open-Source-Programm [...] frei verfügbar ist und ohne Lizenzierung heruntergeladen und benutzt werden darf.“³ PostgreSQL Datenbank basiert auf dem SQL Standards 92 und 99 und verfügt über einige eigene Erweiterungen. Das Datenbanksystem baut auf einer typischen Client-Server-Architektur auf. Dadurch werden entfernte Zugriffe ermöglicht und Hardware-Komponenten müssen nicht lokal zusammen stehen. Innerhalb von PostgreSQL übernimmt das Serverprogramm „postmaster“ die Verwaltung des Systems. Es ist zuständig für den Verbindungsaufbau von Client-Anfragen, sowie das Ausführen von Aktionen auf der Datenbank. PostgreSQL's „postmaster“ ist kompatibel mit vielen verschiedenen Clientprogrammen. Mit der PostgreSQL-Installation wird bereits „psql“ mitgeliefert. Dieser stellt einen rudimentären Datenbankmonitor dar. Unter Linux wird oft pgAccess mitgeliefert, welches auch unter Windows verwendet werden kann.⁴

Unter PostgreSQL können Trigger und Regeln verwendet werden. Trigger stellen Prozeduren dar. Diese sind auf dem Datenbankserver gespeichert und können mit einer Bedingung versehen werden. Tritt diese Bedingung dann ein, beispielsweise wenn ein neuer Datensatz eingefügt wird, so wird die Prozedur ausgelöst. Durch diese Prozeduren kann beispielsweise mitgeschrieben werden, wer sich mit einer Datenbank wann verbunden hat und welche Aktionen er auf dieser durchgeführt hat. Sie können unter anderem mit Java, Perl, Python, Ruby, Tcl, C/C++ und der von PostgreSQL entwickelten Sprache PL/pgSQL geschrieben werden⁵. Regeln können das Abfrageverhalten manipulieren. Auch hierfür können Bedingungen festgelegt werden, welche eine Regel auslösen. Mit Regeln können beispielsweise Delete-Statements in Update-Statements umgewandelt werden.⁶ Sinn von solchen Regeln könnte sein, dass Informationen nie gelöscht werden sollen, sondern nur als vom Anwender gelöscht markiert werden sollen oder dann von einem Administrator gelöscht werden können.

² Vgl. Lobin, Mehler 2004

³ Vgl. Boenigk, Burger 2014

⁴ Vgl. Boenigk, Burger 2014

⁵ Vgl. PostgreSQL 2014

⁶ Vgl. Boenigk, Burger 2011

PostgreSQL zählt zu den leistungsstärksten und im Jahre 2012 zu den am schnellsten wachsenden Datenbanksystemen der Welt.⁷ Es läuft unter den wichtigsten Betriebssystemen wie Linux, UNIX(zum Beispiel Mac OS X) und Windows. Natürlich unterliegt PostgreSQL auch Einschränkungen. So ist eine Tabelle auf 32TB beschränkt, wobei ein Eintrag maximal aus 1,6TB bestehen kann und ein Feld maximal eine Datenmenge von 1GB aufnimmt. Die Anzahl der Reihen sind dahingegen nur durch die Maximalgröße der Tabelle beschränkt. Jede Tabelle kann aus 250-1600 Spalten bestehen, die genaue Anzahl ist von den verwendeten Typen der Spalten abhängig. Pro Tabelle können beliebig viele Indexe angelegt werden. Insgesamt kann PostgreSQL eine beliebig große Datenbank verwalten.

⁷ Vgl. Fröhlich 2011

4 Realisierung

4.1 Projektablauf

Nr.	Vorgangname	Dauer	Anfang	Ende	Integrationsseminartermine
1	Phase 1	13 Tage	Don 15.05.14	Mon 02.06.14	1
6	Phase 2	8 Tage	Mon 02.06.14	Mit 11.06.14	2
16	Phase 3	12 Tage	Mit 11.06.14	Don 26.06.14	3
22	Phase 4	4 Tage	Don 26.06.14	Die 01.07.14	4
28	Phase 5	10 Tage	Die 01.07.14	Mon 14.07.14	5

Abbildung 3 Übersicht Projektplan

Der Projektplanübersicht können die einzelnen Termine entnommen werden. Das Projekt war für einen Zeitraum von 47 Tagen angesetzt, wobei die reine Arbeitszeit geringer ausfiel. Im Folgenden werden die Kerninhalte der einzelnen Phasen berichtet, um den Ablauf des Projektes transparent zu gestalten.

Charakteristisch für Phase eins war das Einlesen die Thematik. Das Team verschaffte sich zunächst einen theoretischen Hintergrund zur gegebenen Problematik. So wurden ebenso in der ersten Phase erste Lösungsansätze entworfen. Für die Vorbereitung dieser wurde die PostgreSQL Datenbank ausgesetzt, sowie ein einfacher Algorithmus geschrieben, der die Auftrittshäufigkeit jeder Wörter der Beispieldaten zählt und diese in der Datenbank speichert.

Dank des entwickelten Algorithmus aus Phase eins konnten in der zweiten Phase damit die Beispieldaten analysiert werden, sodass daraus eine Wortliste generiert werden konnte. Basierend auf dieser Liste konnten theoretische Ansätze wie von Bayes oder die Sentiment Untersuchung realisiert werden. Ziel dieser Phase war ein lauffähiger Prototyp zur praktischen Belegung der theoretischen Ansätze. Dieses Ziel wurde auch erreicht. Abschluss fand diese Phase mit der Dokumentation der ersten Ergebnisse sowie einem Feedback-Gespräch mit den Auftraggebern.

Die dritte Phase war von der Gestaltung und Entwicklung der Oberfläche geprägt. Mitunter half dies um leichter die weiten Schritte zu vollziehen. Im Anschluss an die Erstellung der Oberfläche konnten die in der Vorphase erstellten Algorithmen genauer überprüft werden. Dadurch konnten Diese genauer angepasst werden. Abschluss fand diese Phase in der Erstellung der GUI-Beschreibung sowie im Nachpflegen der Dokumentation bis dato.

Eine genaue Analyse der Gesamtsituation leitete die vierte Phase ein. Daraus resultierte die Idee der Phrasenanalyse. Somit wurde diese Idee im nächsten Schritt implementiert und getestet. Bei der Implementierung mussten Kompromisse gemacht werden. Auch diese Phase fand ihren Abschluss in der Überarbeitung der Dokumentation, begleitet durch die Anfertigung eines Leitfadens zur Installation der Datenbank sowie des Prototyps.

Ziel der letzten Phase war der Feinschliff des Prototyps und der Ausarbeitung. Dazu wurden überflüssige Kommentare und Ausgaben aus dem Quellcode entfernt. Der bereinigte Prototyp wurde dann in einem Installationspaket zusammengestellt, sodass künftig die Installati-

on anhand des Installationsleitfadens erfolgen kann. Nach diesen Aufgaben muss nur noch die Präsentation angefertigt werden, sodass auch Phase fünf ihr Ende finden wird.

4.2 Planung & Projektdurchführung

4.2.1 Vorgehensweise

Die Auswahl eines geeigneten Vorgehensmodells zur Lösung einer gegebenen Problemstellung stellt den initialen Baustein eines Projektes dar. Ein Vorgehensmodell bestimmt, wie ein Projekt ablaufen wird, daher ist es wichtig in Abhängigkeit diverser Rahmenbedingungen des Projektes ein passendes Vorgehensmodell zu wählen. So wäre es wenig sinnvoll ein Wasserfall-Modell zur Entwicklung einer Software zu nutzen, wenn es den Auftraggebern auf schnelle und sichtbare Erfolge ankommt. Mitunter spielen Umfang, Themengebiet, zeitliches Ausmaß und Sicherheitsaspekte eine wichtige Rolle in der Bestimmung eines Vorgehensmodells. Im Folgenden wird nun das Vorgehensmodell von PricewaterhouseCoopers (PwC) vorgestellt und auf seine Relevanz überprüft.

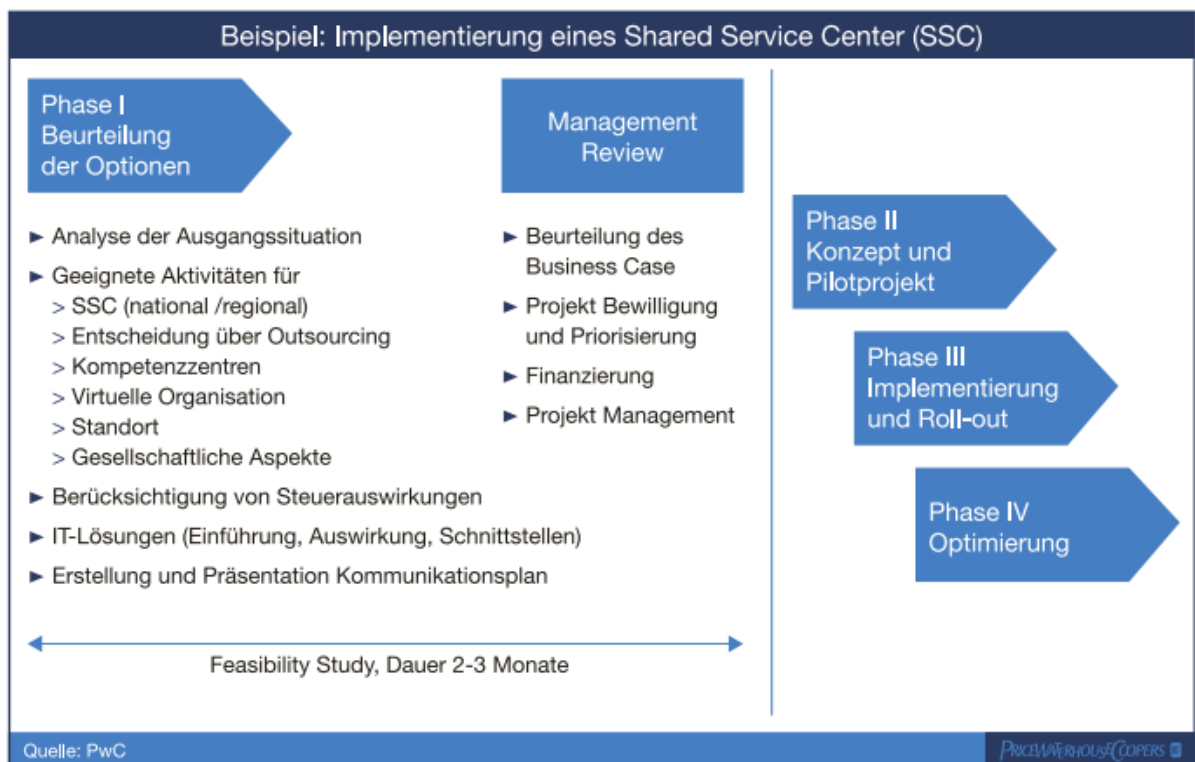


Abbildung 4 Feasibility Study am Beispiel Shared Service Center nach PwC⁸

PwCs Vorgehensmodell zielt auf Machbarkeitsstudien ab. Die konkrete Ausgestaltung hängt vom jeweiligen Projekt ab. Eine Machbarkeitsstudie stellt nach PwC Phase eins eines jeden Projekts dar. PwC unterteilt die Studie in zwei Unterphasen. Unterphase eins richtet sich, wie in Abbildung 4 exemplarisch zu sehen ist, der Erschließung und Beurteilung von Möglichkeiten. Es wird dazu die aktuelle Ist-Situation betrachtet. Nach der Erfassung aller relevanten Daten müssen diese ausgewertet werden. Zur Findung konkreter Lösungsansätze kann die

⁸ Enthalten in: Mörtl, J. (2007)

Phase um einen Ideenfindungsprozess erweitert werden. Sollte es dabei zu mehreren konkurrierenden Lösungsansätzen kommen, sollen diese laut PwC durch eine Portfolioanalyse betrachtet und bewertet werden. Der daraus entstehende Ansatz zur Lösung der gegebenen Problemstellung, wird im zweiten Schritt der Studie in einer Konzeption niedergeschrieben. Die Konzeptionserstellungsphase schließt mit einer Empfehlung oder einem Ausblick ab.⁹ Generell können Machbarkeitsstudien, vor allem bei Projekten in der Softwareentwicklung oder im Ingenieurwesen, durch die Entwicklung eines Prototyps überprüft werden. Somit kann unter relativ nahen Realitätsbedingungen das Ergebnis der Studie bekräftigt oder, im schlechteren Falle, widerlegt werden. Eine Widerlegung einer Machbarkeitsstudie durch eine praxisnahe Überprüfung anhand eines Prototyps hätte je nach Studie eine erneute Studie zur Folge.

4.2.2 Herausforderungen

Im Laufe des Projektes ergaben sich einige Herausforderungen welche in diesem Kapitel mit den jeweiligen Lösungsansätzen beschrieben werden.

Bei Projektbeginn bestand die Herausforderung zunächst darin, sich in das Thema einzuarbeiten, da Vorwissen zum Thema Text Mining nicht vorhanden war. Nach einiger Recherche wurde festgelegt, dass eine Kombination der hier bereits beschriebenen Bayes-Klassifikator und Sentiment Analyse verwendet werden soll. Für die tatsächliche Einordnung des Kundenschreibens als Beschwerde wurde diese Kombination später noch mit der Phrasenanalyse ergänzt.

Die größte Herausforderung entstand allerdings bei der Umsetzung des Bayes-Klassifikators. Um so größer der verwendete Wortschatz im Laufe des Projektes wurde, desto kleiner wurden die Zahlen, welche für die Berechnungen verwendet wurden. Diese Situation entstand dadurch, dass die Wahrscheinlichkeit eines Wortes in einem Schreiben aufzutreten der Anteil des Wortes an sämtlichen zur Berechnung verwendeten Schreiben darstellt. Als nun immer mehr Schreiben für die Berechnung miteinbezogen wurden, vergrößerten sich die Nenner der Brüche (siehe Kapitel 3.1.2) beständig. Dadurch wurde der gesamte Bruch immer kleiner, konnte aber als Dezimalzahl noch in einer Java-Variable vom Typ Double gespeichert werden. Da aber bei der Berechnung letztendlich der Term vor einer Addition getrennt berechnet werden muss, wurde das Ergebnis des Terms durch die Multiplikation der vielen kleinen Zahlen auch immer kleiner. Deshalb konnte dieses Ergebnis nicht in einem gängigen Java-Datentyp wie einem Double gespeichert werden. Da die Werte so klein waren wurden sie auf Null gerundet, was im Endeffekt zu einem verfälschten Bayes-Klassifikator geführt hatte.

Die Lösung dieser Herausforderung besteht nun darin, dass der natürliche Logarithmus der einzelnen Wortwahrscheinlichkeiten für die Berechnung genutzt wird. Auch wird für die Berechnung Logarithmische Algebra verwendet. Daher ist das eigentliche Ergebnis, also der Bayes-Klassifikator, das Endergebnis der Berechnung hoch der Eulerschen Zahl. So konnten die kleinen Zahlen, welche sich nicht in eine Double-Variable speichern ließen, vermieden werden.

Darüberhinaus ergab sich eine Herausforderung bei der Phrasenanalyse. Wenn eine markierte Textstelle eine andere markierte Stelle überschneidet, oder sogar ganz überdeckte, wurden anfänglich beide Textstellen als Phrasen abgespeichert. Es musste also eine Lösung

⁹ Vgl. Mörtl, J. (2007), S. 4ff

gefunden werden, wie gewährleistet wird, dass bestimmte Textstellen nicht mehrfach abgespeichert werden. Dies wurde nun so gelöst, dass von der Anwendung geprüft wird, welche Markierungen sich überschneiden, so dass diese zu einer Markierung zusammengefasst werden können. Eine Überschneidung liegt dann vor, wenn der Beginn der zweiten Markierung zwischen dem Beginn und Ende der ersten Markierung liegt. Auch für den Fall, dass zwei Markierungen sich nicht überschneiden, aber direkt aneinander grenzen, werden diese zu einer Markierung zusammengefasst. Dazu wird jeweils der Beginn der im Gesamttext zuerst stehenden Markierung als Beginn der neuen Markierung genommen. Als Ende der neuen Markierung wird nun das Ende der zweiten Markierung verwendet. So entsteht eine einzelne Markierung, welche die sich überschneidenden Markierungen umfasst. Durch dieses Vorgehen ist die Herausforderung der Vermeidung von Mehrfachspeicherung einer Phrase, beziehungsweise eines bestimmten Textteiles, bewältigt worden.

4.2.3 Beschreibung der Anwendung

In diesem Kapitel wird der Aufbau und die Benutzung der grafischen Oberfläche der Anwendung zur Beschwerdeerkennung beschrieben. Die grafische Oberfläche besteht hauptsächlich aus einem Fenster, welches in folgendem Screenshot zu sehen ist.

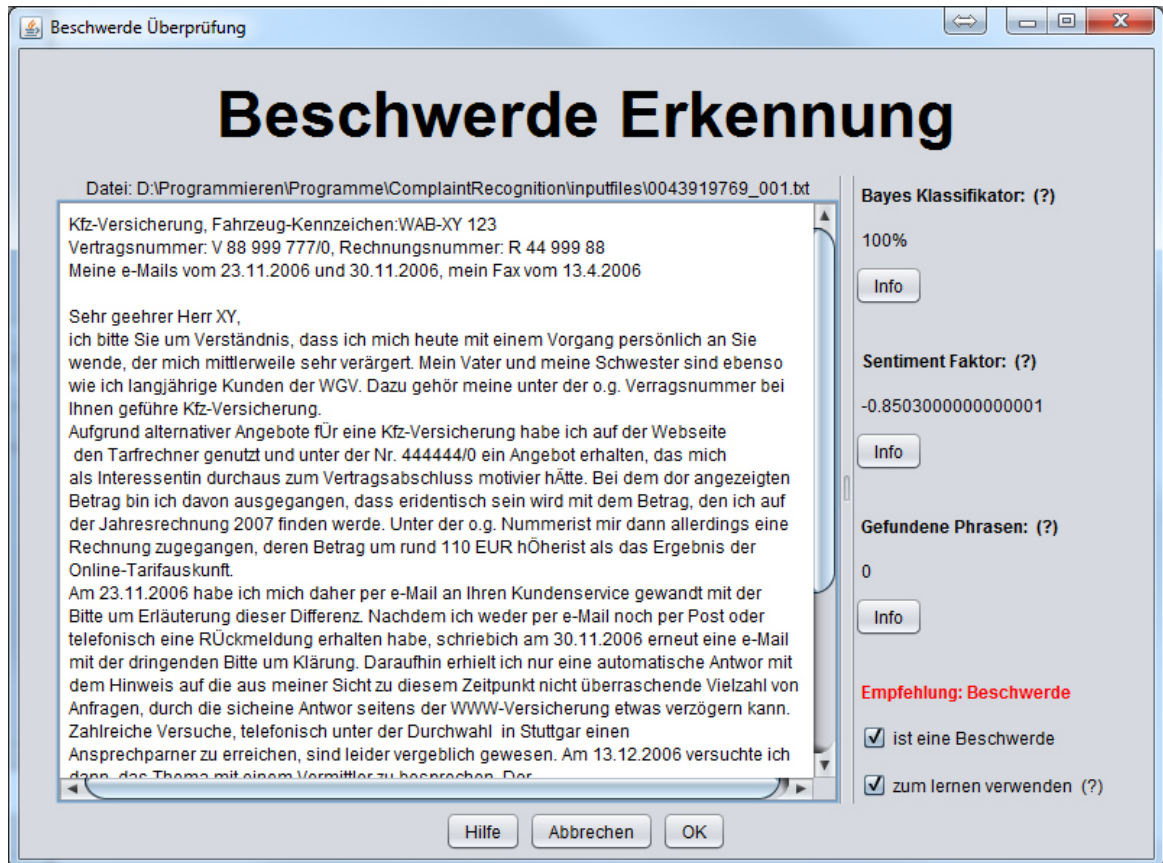


Abbildung 5 Screenshot der Anwendung

Im linken Bereich des Fensters befindet sich ein Textfeld, in welchem das auf eine Beschwerde geprüfte Kundenschreiben dargestellt wird. In diesem lassen sich auch Aktionen ausführen, worauf später eingegangen wird. Oberhalb des Textfeldes befindet sich die Pfadangabe zum Speicherort der überprüften Datei. Unterhalb des Textfeldes befinden sich die Buttons "Hilfe", "Abbrechen" und "OK". Mit dem Hilfe-Button lässt sich ein Dialogfenster mit einer Kurzeinweisung öffnen. Mit dem Abbrechen-Button wird die Anwendung geschlossen, ohne dass eventuell geänderte Angaben übernommen werden. Mit dem OK-Button werden geänderte Angaben übernommen und anschließend die Anwendung beendet.

Im rechten Bereich des Fensters ist die Auswertung der Beschwerdeanalyse mit ihren einzelnen Faktoren dargestellt. Zuerst wird der Bayes Klassifikator angegeben. Fährt der Anwender mit dem Mauscursor darüber, erhält er eine kurze Erläuterung zu diesem Wert durch einen sogenannten Tooltip. Durch den Info-Button, welcher sich unter dem Bayes-Wert befindet, wird ein neues Fenster geöffnet, in welchem die Werte genauer dargestellt sind. Dieses Fenster ist im nächsten Screenshot abgebildet.

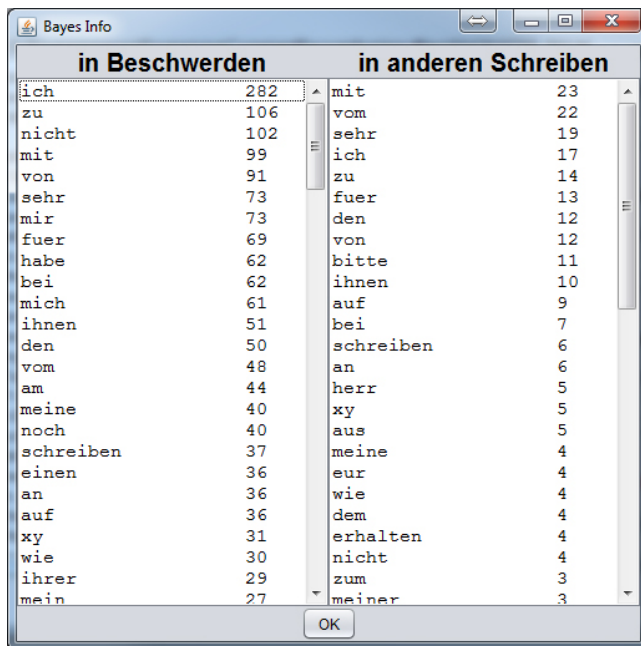


Abbildung 6 Info-Fenster zum Bayes-Klassifikator

Hier werden in der linken Spalte die für die Bayes-Klassifikation verwendeten Wörter der überprüften Schreiben nach Häufigkeit, in welcher diese Wörter in Beschwerden vorkommen, aufgelistet. In der rechten Spalte werden nach dem gleichen Prinzip die Wörter aufgelistet, welche in einem Schreiben vorkommen, bei welchen es sich nicht um Beschwerden handelt. Klickt man in einer der beiden Listen auf ein bestimmtes Wort, so wird dieses entsprechend auch in der anderen Liste markiert. So kann der Anwender beispielsweise sehen, welche Wörter besonders oft in Beschwerden verwendet werden.

Unterhalb der Bayes-Klassifikation befindet sich im Hauptfenster die Angabe des Sentiment-Faktors. Auch hier bekommt der Anwender durch einen Tooltip eine kurze Erläuterung des Sentiment-Faktors. Über den Info-Button unter der Angabe lässt sich wiederum ein neues Fenster öffnen, welches sich nun von dem des Bayes-Klassifikator unterscheidet. Hier werden die überprüften Wörter mit ihren dazugehörigen Bewertungen in einer einzelnen Liste darstellt. Dies ist auf dem folgenden Screenshot zu erkennen.

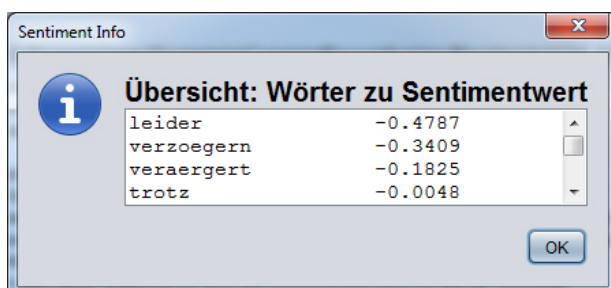


Abbildung 7 Info-Fenster zum Sentiment-Faktor

Nach dem Sentiment-Faktor folgt im Hauptfenster die Angabe der Phrasenanalyse. Hier wird angegeben, wie viele zuvor gespeicherte Phrasen, die auf eine Beschwerde hindeuten, im Text gefunden wurden. Auch an dieser Stelle lässt sich über den entsprechenden Info-Button ein neues Fenster öffnen, das wie das Fenster des Sentiment-Faktors aufgebaut ist. Die Lis-

te enthält hier aber lediglich die Phrasen, welche im überprüften Schreiben gefunden wurden. Dies ist im nächsten Screenshot zu sehen.

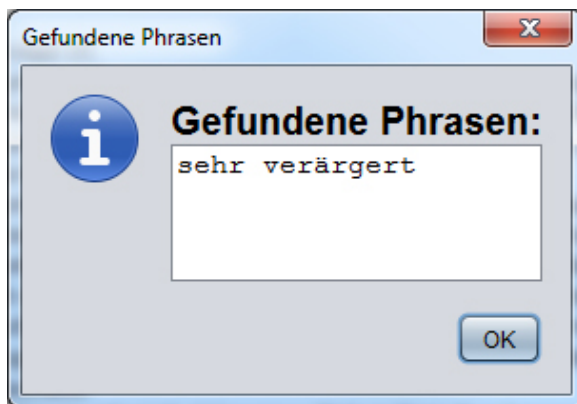


Abbildung 8 Info-Fenster zu gefundenen Phrasen

Im Hauptfenster wird von der Anwendung eine Empfehlung basierend auf dem Bayes-Klassifikator, dem Sentiment-Faktor und der Phrasenanalyse gegeben. Diese befindet sich unter den anderen Angaben im rechten Bereich des Hauptfensters. Handelt es sich um eine Beschwerde, ist dort in roter Schrift zu lesen: "Empfehlung: Beschwerde". Ist das Schreiben allerdings keine Beschwerde, steht an dieser Stelle in grüner Schrift: "Empfehlung: keine Beschwerde".

Unterhalb dieser Empfehlung befindet sich eine sogenannte Checkbox mit der Beschriftung "ist eine Beschwerde". Entsprechend der eben erwähnten Empfehlung ist diese Checkbox bei einer vorliegenden Beschwerde schon vorbelegt. Darunter befindet sich eine weitere Checkbox mit der Beschriftung "zum Lernen verwenden". Ist diese aktiviert, wird das aktuell dargestellte Schreiben für alle folgenden Überprüfungen ebenfalls mit in den Bayes-Klassifikator eingerechnet. Das bietet sich besonders dann an, wenn das Schreiben eine Beschwerde ist, dies aber von der Anwendung nicht erkannt wurde.

Des Weiteren lassen sich vom Anwender Phrasen, welche für eine Beschwerde typisch sind, im Textfeld mit der Maus markieren. Fährt man nun mit dem Mauscursor über eine solche markierte Textstelle, ändert sich der Cursor zu einem roten Kreuz. Klickt man nun in die Textstelle, wird die Markierung für diese Phrase wieder aufgehoben. Wird nun die Anwendung über den OK-Button geschlossen, so werden die markierten Phrasen in die Phrasenanalyse für folgende Beschwerdeprüfungen übernommen.

4.2.4 Installation der Anwendung

Aufsetzen der Datenbank

Die Datenbank wird wie unter Windows üblich durch ein Ausführen der "postgresql-9.3.4-3-windows-x64.exe"-Datei installiert. Hierbei wird nach einem Passwort für den Superuser namens "postgres" gefragt. Dieses sollte am besten notiert werden, da es später noch in der Anwendung eingetragen werden muss. Im nächsten Installationsschritt wird nach dem Port

gefragt. Den dort vorhandenen Wert bitte auf "5433" abändern. Nach der Installation wird gefragt, ob zusätzlich noch der PostGres-StackLoader installiert werden soll. Dieser ist für die Anwendung nicht notwendig. Deshalb kann dieser Schritt übersprungen werden. Damit ist das Vorbereiten der Datenbank abgeschlossen. Es müssen keine Tabellen erstellt werden. Das wird von der Anwendung übernommen.

Die Anwendung in Eclipse übernehmen

Für die Entwicklung der Anwendung wurde die Entwicklungsumgebung Eclipse genutzt. Um das Projekt der Anwendung dorthin zu importieren, muss zunächst in Eclipse ein leeres Java-Projekt erstellt werden. Über einen Rechtsklick auf dieses und dem Menüpunkt "Import" kann nun als File System der Ordner "ComplaintRecognition" importiert werden. Hierbei müssen alle Daten übernommen werden.

Nachdem die Anwendung erfolgreich importiert wurde muss noch etwas angepasst werden. Es befindet sich im Ordner "metafiles" des Projektes die Datei "db.properties". In dieser werden die grundlegenden Parameter für die Datenbankverbindung festgelegt. Hier muss der Parameter "password" abgeändert werden. An dieser Stelle wird nun das Passwort, welches bei der Installation der Datenbank festgelegt wurde, eingetragen. Danach muss die Datei gespeichert werden.

Erster Start der Anwendung

Die Anwendung lässt sich über die Klasse "Start.java", welche sich im Projektordner "src" im Package "launch" befindet, starten. Es erscheint nun die Entwicklungs-GUI. Sind die benötigten Datenbanktabellen nicht vorhanden, werden diese nun selbstständig von der Anwendung erstellt. Danach kann die Anwendung genutzt werden.

5 Ausblick

5.1 Grammatikanalyse

Die Grammatikanalyse zielt darauf ab, mit sprachwissenschaftlichen Vorgehensweisen die Bedeutung eines Textes durch eine pragmatische Untersuchung dessen zu analysieren. Dazu muss zu Beginn der Untersuchung eine Vielzahl an zu untersuchenden Texten vorliegen. Die Texte sollten im vorliegenden Beispiel aus Beschwerden sowie aus nicht Beschwerden bestehen um daraus typische Strukturen und grammatikalische, sowie linguistische Merkmale ableiten zu können die für eine Beschwerde sprechen. Für diese Analysetechnik sind sprachwissenschaftliche Fähigkeiten erforderlich. Die erkannten Strukturen können dann in einem System berücksichtigt werden, sodass mithilfe eines auf Basis dieser Analyse entwickelten Algorithmus das System erkennen kann ob es sich bei einem Dokument um eine Beschwerde handelt oder nicht.

5.2 Abschaffung analoger Medieneingänge

Bisher gehen Beschwerden mit vielen anderen Briefen ein. Ein Lösungsansatz um die ganze Sache einfacher zu gestalten könnte die Minimierung analoger Posteingänge sein. Denkbar ist die Implementierung eines Online-Kontaktformulars wobei Anwender aus einer vordefinierten Liste mit Gründen den Grund ihrer Kontaktaufnahme auswählen. So könnten Anwender beispielsweise „Beschwerde“ als Grund angeben. Dadurch könnten eingegangene Beschwerden recht einfach erkannt und entsprechend weitergeleitet werden. Um Kunden die auf traditionelle Medien setzen nicht zu verärgern sollte auch in Zukunft die Möglichkeit geschaffen werden diese zu nutzen. Denkbar wäre die Einführung spezieller Formulare, wodurch das OCR-System automatisch erkennt, dass es sich bei dem Briefstück um eine Beschwerde handelt. Natürlich wird es immer Menschen geben, die diese Medien nicht benutzen werden und weiterhin herkömmliche Briefe schreiben, doch die Anzahl dieser Briefstücke würde weniger werden, wodurch der Aufwand der Erkennung kein so gewichtiger Prozess ist.

5.3 Datenbank Routinen

Bisher laufen alle Methoden in einer Java Applikation ab. Unter Performance Gesichtspunkten ist dies, vor allem was Sortier- und Bereinigungsverfahren angeht, keine ideale Lösung. Gerade bei den genannten Operationen werden große Datenmengen zwischen der Applikation und der Datenbank hin und her geschoben. Genau hier könnten Datenbank Routinen eingesetzt werden. Diese könnten beispielsweise das Bereinigen der Datenbank übernehmen. So könnte diese Routine von der Datenbank selbst gestartet werden. Dies kann so eingestellt werden, dass die Datenbank diese recht aufwendige Routine startet wenn sie gerade nicht eingesetzt wird, dies kann beispielsweise über Nacht sein. Ebenso könnten die Algorithmen zur Untersuchung der Texte auf der Datenbank direkt implementiert werden, um auch so einen unnötigen Datenaustausch zu vermeiden. Sehr wahrscheinlich ist die Datenbank bei solchen Abfrage lästigen Algorithmen sogar deutlich schneller als eine auf Java basierte Anwendung. Allerdings sollten dazu die Dateien unter PostgreSQL als Strings in der Datenbank vorliegen, da PostgreSQL nicht als Dokumentendatenbank optimiert ist.

5.4 Verwendung kommerzieller Lösungen

Da es sich bei der Erkennung von Beschwerden um ein Problem handelt, mit dem nahezu alle Unternehmen zu kämpfen haben, gibt es dafür schon kommerzielle Lösungen auf dem Markt. „Turnfriendly“ von der FIOROX. AG, ist ein Beschwerdemanagement System mit der Fähigkeit Inhalte eines Schreibens interpretieren zu können. Somit kann die Software Beschwerden erkennen. „Turnfriendly“ wird bereits von vielen, teils großen, Unternehmen eingesetzt. Mitunter von der Deutschen Bank, der Commerzbank und der Ergo Direkt Versicherung.¹⁰ Sicherlich profitiert ein weit verbreitetes Produkt davon. Durch die weite Verbreitung stehen dem Algorithmus viele Daten zur Verfügung um zu analysieren, ob es sich bei einem Schreiben um eine Beschwerde handelt oder nicht. Da davon auszugehen ist, dass ein lernender Algorithmus verwendet wird, der mit steigender Anzahl an Beispieldaten immer genauer wird, kann davon ausgegangen werden, dass FIOROX mit ihrer Software eine sehr gute Trefferquote erzielt.

5.5 Rechtschreibprüfung

Die Verwendung einer gut funktionierenden Rechtschreibprüfung kann weiterhin helfen das Beschwerdenerkennungsprogramm zu verfeinern. Da die meisten Schreiben, die beim Auftraggeber eingehen, von Menschen verfasst werden, beinhalten diese natürlich auch Rechtschreibfehler. Zusätzlich kann es vorkommen, dass die OCR Erkennung gerade bei handschriftlichen Schreiben Buchstabendreher erkennt oder selber für Rechtschreibfehler sorgt. Unter Umständen werden auch einzelne Buchstaben anstatt sinnvoller Wörter erkannt. Ein gut funktionierendes Rechtschreibprogramm könnte diese Fehler, bedingt durch die OCR Erkennung oder die Autoren der Schreiben selbst ausbügeln und somit die Datenbankgrundlage für die Beschwerdeerkennung verbessern in dem ungewollt verschiedene Schreibweisen der gleichen Wörter entfernt werden. Ähnlich wie bei der Grammatikanalyse handelt es sich bei einer umfassenden Rechtschreibprüfung keinesfalls um einen simplen Programmbaustein. Daher muss auch hier untersucht werden ob ein Fremdbezug eine denkbare Alternative darstellt oder ob eine Eigenentwicklung mehr Sinn macht. Wie bei jeder Finanzierungsentscheidung muss außerdem auch hier entschieden werden, ob sich die Investition insgesamt überhaupt bezahlt macht. Das heißt im speziellen: Kann durch die Einführung der Rechtschreibprüfung eine signifikante Verbesserung des Beschwerdeerkennungsprogramms erreicht werden?

¹⁰ o.A. o.J.

6 Fazit

Die verwendeten Methoden, welche in diesem Projekt zur Erkennung einer Beschwerde eingesetzt wurden, erwiesen sich in Kombination miteinander als sehr gutes Vorgehen, um die Kundenschriften präzise in Beschwerde und keine Beschwerde zu erkennen. Der Bayes-Klassifikator liefert allein schon durch die Tatsache, dass nur zwei mögliche Ergebnisse (Beschwerde / keine Beschwerde) erreicht werden können eine sehr genaue Aussage. Dabei wird jedoch nur die Wahrscheinlichkeit, mit welcher es sich bei einem bestimmten Schreiben um eine Beschwerde handelt ermittelt, und keine endgültige Einstufung ermittelt. Die Genauigkeit, mit welcher der Bayes-Klassifikator ermittelt wird, lässt sich durch Schreiben, welche vom Nutzer der Anwendung, aber nicht von der Anwendung selbst, als Beschwerde erkannt werden, noch erhöhen. Die Anwendung kann also dazulernen und so ihre Effizienz erhöhen.

Auch die Sentiment Analyse bietet eine gute Kategorisierung des Kundenschreibens. Da bei diesem Vorgehen die Grundstimmung des Schreibens ermittelt wird, lässt sich darüber eine Aussage treffen. Ist der Sentiment-Faktor beispielsweise im negativen Bereich, ist es wahrscheinlicher, dass es sich um eine Beschwerde handelt. Zur Ermittlung dieses Wertes werden die Wörter des Textes anhand einer bestehenden Datenbasis bewertet und miteinander verrechnet. In dieser Datenbasis sind jeweils circa 14000 positive und negative Wörter mit ihren individuellen Sentiment-Werten gespeichert. So kann eine relativ genaue Aussage über die Grundstimmung des Schreibens ermittelt werden.

Als letzte in diesem Projekt verwendete Methode kategorisiert die Phrasenanalyse anhand von erkannten Textstücken das Kundenschreiben. Hierbei kann der Nutzer der Anwendung aktiv am Lernprozess der Anwendung teilnehmen, indem die Möglichkeit besteht, besonders auffällige Textstellen zu markieren und abzuspeichern. Diese Textstücke werden von nun an ebenfalls zur Phrasenanalyse genutzt. Umso mehr Phrasen abgespeichert werden, desto treffsicherer kann auch die Phrasenanalyse eine Beschwerde erkennen.

Zusammenfassend ergibt die nun so entstandene Kombination aus Bayes-Klassifikator, der Sentiment Analyse und der Phrasenanalyse somit eine sehr zuverlässige Aussage, ob es sich bei einem bestimmten Kundenschreiben um eine Beschwerde handelt.

Die Erkennung eines Kundenschreibens als eine Beschwerde lässt sich durch verschiedene weitere Analysen und Techniken durchaus noch verbessern. Eine wichtige wäre beispielsweise eine Ergänzung der Phrasenanalyse, so dass nicht nur die markierten Phrasen vorliegen, sondern auch deren Konjugationen. So könnte die Phrasenanalyse noch effektiver funktionieren. Allerdings konnte eine Umsetzung dieser und weiterer Ideen aber aus zeitlichen Gründen im Rahmen dieses Projektes nicht realisiert werden.

Quellenverzeichnis

Literaturverzeichnis

- Boenigk, C./Burger, R. (28.11.2011):** PostgreSQL, das fortschrittlichste Open Source Datenbanksystem, <http://www.postgresql.de/features.html#tr>,
Abruf: 06.07.2014
- Boenigk, C./Burger, R. (06.07.2014):** PostgreSQL, das fortschrittlichste Open Source Datenbanksystem, <http://www.postgresql.de/index.whtml#pg>,
Abruf: 06.07.2014
- Franke, J. (2003):** Text mining, in: Advances in Soft Computing, 1. Aufl, Heidelberg [u.a.]: Physica-Verl.
- Fröhlich, L. (2011):** PostgreSQL 9,
Praxisbuch für Administratoren und Entwickler, in: PostgreSQL 9
- Heyer, G./Quasthoff, U./Wittig, T. (2006):** Text Mining: Wissensrohstoff Text,
Konzepte, Algorithmen, Ergebnisse, in: IT lernen, Herdecke [u.a.]: W3L-Verl.
- Lobin, H./Mehler, A. (2004):** Automatische Textanalyse,
Systeme und Methoden zur Annotation und Analyse natürlichsprachlicher Texte, 1. Aufl, Wiesbaden: VS, Verl. für Sozialwiss.
- Mörzl, J. (o. J.):** Schwerpunktthema, Machbarkeit am Beispiel Tourismus, in: Tipps & Trends,
Schwerpunkt-thema Machbarkeits-studien Feasibility Study – Machbarkeit unter der Lupe
- o. A. (2014):** PostgreSQL: About,
<http://www.postgresql.org/about/>,
Abruf: 2014
- Schira, J. (2005):** Statistische Methoden der VWL und BWL,
Theorie und Praxis, in: Wirtschaft, 2. Aufl, München, Boston [u.a.]: Pearson Studium
- TurnFriendly (14.07.2014):** Inhaltserkennung - TurnFriendly,
<http://www.turnfriendly.com/de/module/inhaltserkennung/>,
Abruf: 14.07.2014
- Weiss, S. M. (2005):** Text mining,
Predictive methods for analyzing unstructured information, New York: Springer

Textanalyse am Beispiel von Beschwerdeschreiben

Schriftliche Ausarbeitung
im Rahmen der Lehrveranstaltung „Integrationsseminar“
Kompetenzzentrum Open Source (KOS)

Fakultät Wirtschaft
Studiengang Wirtschaftsinformatik
WWI2011V

Inhaltsverzeichnis

Abbildungsverzeichnis	III
Tabellenverzeichnis	IV
Quelltextverzeichnis	V
1 Einleitung	1
2 NoSQL-Datenbanken	2
2.1 Definition und Grundlagen	2
2.2 Dokumentenorientierte Datenbanken	3
2.3 MongoDB	4
3 Klassifikation	6
3.1 Grundlagen der Klassifikation	6
3.2 Der naive Bayes-Klassifikator	8
3.2.1 Die Formel von Bayes	8
3.2.2 Funktionsweise	9
4 Implementierung	11
4.1 Die Klasse BOOT	12
4.1.1 Methode tokenizer	13
4.1.2 Methode create_knowledge	15
4.1.3 Das REST-Interface	17
4.2 Das User-Interface	17
5 Fazit	22
6 Quellenverzeichnisse	23

Abbildungsverzeichnis

1	Übersicht NoSQL DB	2
2	Google Search Trends	3
3	MongoDB Einordnung	4
4	Erfolgsquote im Verlauf der Anzahl eingelernter Dokumente bei variabler Wortlänge	14
5	Erfolgsquote im Verlauf der Anzahl eingelernter Dokumente bei variabler Wahrscheinlichkeitsgrenze	14
6	Erfolgsquote des Algorithmus im Verlauf der Anzahl eingelernter Dokumente bei variabler Häufigkeit der Wörter	17
7	Ansicht des Frontends	18
8	Ansicht des Frontends 2	19
9	Ansicht des Backends	20
10	Ansicht des Backends 2	21

Tabellenverzeichnis

1 Vergleich von MongoDB mit PostgreSQL	5
--	---

Quelltextverzeichnis

1 Die Methode tokenizer	13
2 Die Methode create_knowledge	15
3 JSON eines Wort-Eintrags	16
4 JSON des Statistik-Eintrags	16

1 Einleitung

Im heutigen Dienstleistungsgewerbe spielt die Kundenzufriedenheit eine große Rolle. Der Kontakt zum Kunden sollte entsprechend intensiv gepflegt werden, damit Unternehmen am Markt bestehen können. Ein wichtiger Teilbereich stellt dabei das Beschwerdemanagement dar. Dieses umfasst die Planung, Durchführung und Kontrolle sämtlicher Maßnahmen, die ein Unternehmen im Kontext mit Beschwerden ergreift. Durch das Beschwerdemanagement sollen die Geschäftsbeziehungen, die sich aufgrund von Beschwerden verschlechtert haben, durch Problemlösungen verbessert werden.¹ Der Eingang einer Beschwerde in digitaler oder nicht digitaler Form ist der erste Kontakt eines Versicherungsunternehmens mit einer Kundenbeschwerde. Nach der Erfassung einer Beschwerde, hat jedes Versicherungsunternehmen individuelle Prozesse zur Bearbeitung. Um schnell auf Beschwerden eingehen zu können, müssen die Erfassung und Kategorisierung einer Nachricht möglichst schnell ablaufen. Das Ziel dieser Seminararbeit ist es, die Erfassung und Kategorisierung von eingehenden Nachrichten zu automatisieren. Zur schnellen und effizienten Erfassung bieten sich dabei NoSQL Datenbanken an, da diese im Vergleich zu relationalen Datenbanken eine höhere Skalierbarkeit und Performance aufweisen. In dieser Seminararbeit wird die Datenbank MongoDB verwendet. MongoDB genießt eine hohe Akzeptanz unter den NoSQL Datenbanken. Die Kernziele sind eine hohe Performance und ein einfacher Datenzugriff. Der erste Teil der Seminararbeit befasst sich näher mit NoSQL, dokumentenorientierten Datenbanken und speziell mit MongoDB. Hier werden einige technische Grundlagen erläutert, um die oben genannte Problemstellung zu lösen. Der nächste Teil der Arbeit thematisiert den Bayes-Klassifikator. Dieser bildet die mathematische Grundlage zur Klassifizierung von potentiellen Beschwerdeschreibern. Anschließend wird die Umsetzung und Lösung der genannten Problemstellung vorgestellt. Abschließend erfolgen eine Schlussbetrachtung und ein Ausblick.

¹ Vgl. Leiner, V. (2007), S. 2 f

2 NoSQL-Datenbanken

2.1 Definition und Grundlagen

Der Begriff NoSQL setzt sich aus zwei Wörtern zusammen: No und SQL. NoSQL ist eine SQL Technologie und bedeutet Not Only SQL. Der Begriff wird als Sammelbegriff für alle Datenbanken benutzt, die nicht dem klassischen, relationalen Ansatz folgen. NoSQL ist kein allein-stehendes Produkt, sondern ein Sammelbegriff für eine Vielzahl von Produkten und Konzepten zur Speicherung und Manipulation von Daten.² Der Einsatzbereich von NoSQL Datenbanken liegt hauptsächlich in der Verarbeitung von großen Datenmengen (beispielsweise „BigTable“ von Google). Allerdings gewährleisten sie die Datenkonsistenz nicht in dem von relationalen Datenbanken gewohnten Umfang.³

Merkmal		Beispiele
Dokumentenorientierte Datenbanken		Apache Jackrabbit, BaseX, CouchDB, eXist, Lotus Notes, MongoDB
Graphdatenbanken	Generisch	Neo4j, InfoGrid, HyperGraphDB, Core Data, DEX
	RDF-Zentriert	AllegroGraph, 4store, Virtuoso, andere ↗
Verteilte ACID-Datenbanken		MySQL Cluster
Key-Value-Datenbanken	Festplattenspeicher	Chordless, Google BigTable, GT.M, InterSystems Caché
	Caches im RAM	Membase, memcached, Redis
	Eventually-consistente Speicher	Amazon Dynamo, Project Voldemort, Riak
	Sortierte Key-Value-Speicher	Berkeley DB, Memcachedb
Multivalue-Datenbanken		OpenQM, Rocket U2
Objektdatenbanken		Db4o, ZODB
Spaltenorientierte Datenbanken		Apache Cassandra, Google BigTable, Hbase, SimpleDB

Abb. 1: Übersicht NoSQL DB⁴

Die Abbildung zeigt eine Übersicht über einige NoSQL Datenbanken. In der rechten Spalte werden die Produkte und in der linken Spalte die zugehörigen Merkmale dargestellt. Die nachfolgenden Kapitel betrachten im wesentlichen dokumentenorientierte Datenbanksysteme und speziell MongoDB, da es in dieser Seminararbeit praktisch eingesetzt wird.

²Vgl. Tiwari, S. (2011), S. 4

³Vgl. u.a. Bachmann (2014), S. 157

⁴Enthalten in: Wikipedia (o.J.a)

2.2 Dokumentenorientierte Datenbanken

Dokumentenorientierte Datenbanken sind Datenbanken, bei denen Dokumente die Speichereinheit von Daten bilden. Das bekannte dokumentenorientierte Datenbanksystem ist Lotus Notes. Im Unterschied zu relationalen Datenbanksystemen, die aus Datenbanktabellen und festen Datenbankschemata bestehen, enthalten dokumentenorientierte Datenbanken einzelne Dokumente. Die Dokumente können strukturierte Dateien mit einem Standard Dateiformat sein. Weiterhin sind aber auch Binary Large Objects möglich, die aus datenbanktechnischer Sicht nicht weiter strukturiert sind. Die Dateien können nach einem frei festlegbaren Schema strukturiert werden. Die in der Datei strukturierten Datenfelder bestehen je aus einem Schlüssel-Wert-Paar. Die Dokumente innerhalb der Datenbank werden über einen eindeutigen Identifikator angesprochen.

Folgende dokumentenorientierte Datenbanksysteme sind die am Markt Bekanntesten:

1. BaseX
2. CouchDB
3. eXist
4. MongoDB
5. RavenDB
6. MarkLogic Server⁵

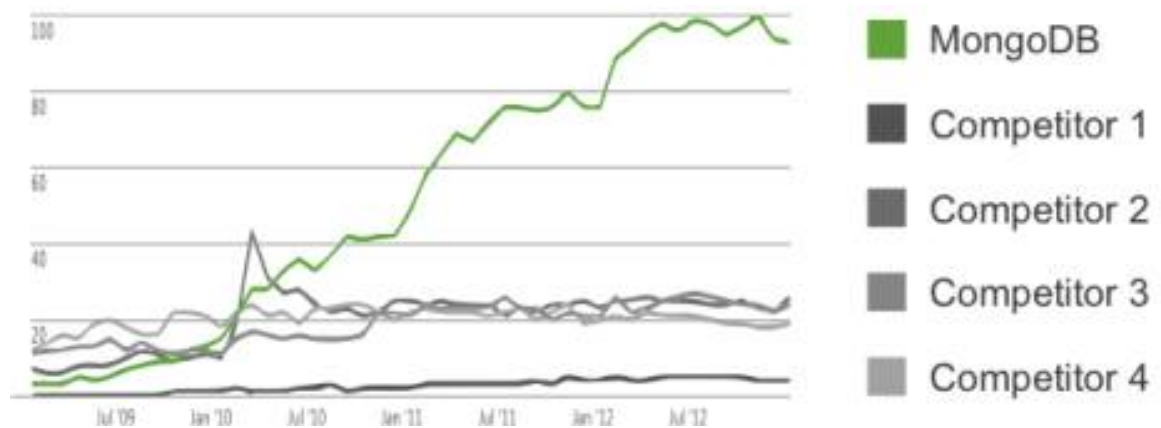


Abb. 2: Google Search Trends⁶

Die Abbildung zwei stellt einen Graphen von Google Search Trends dar. Sie zeigt, wie häufig die Begriffe verschiedener NoSQL Datenbanksysteme bei Google eingegeben wurden. Der Begriff MongoDB wurde viermal häufiger eingegeben als Begriffe von anderen NoSQL Daten-

⁵Vgl. Wikipedia (o.J.b)

⁶Enthalten in: MongoDB (o.J.a)

banksystemen. Daraus lässt sich ableiten, dass MongoDB entsprechend häufiger im Einsatz ist als andere Datenbanksysteme.

2.3 MongoDB

MongoDB ist eine Open Source Datenbank, die von Unternehmen jeder Größe verwendet werden kann. Die Kernziele der Datenbank sind eine hohe Skalierbarkeit sowie Performance.⁷

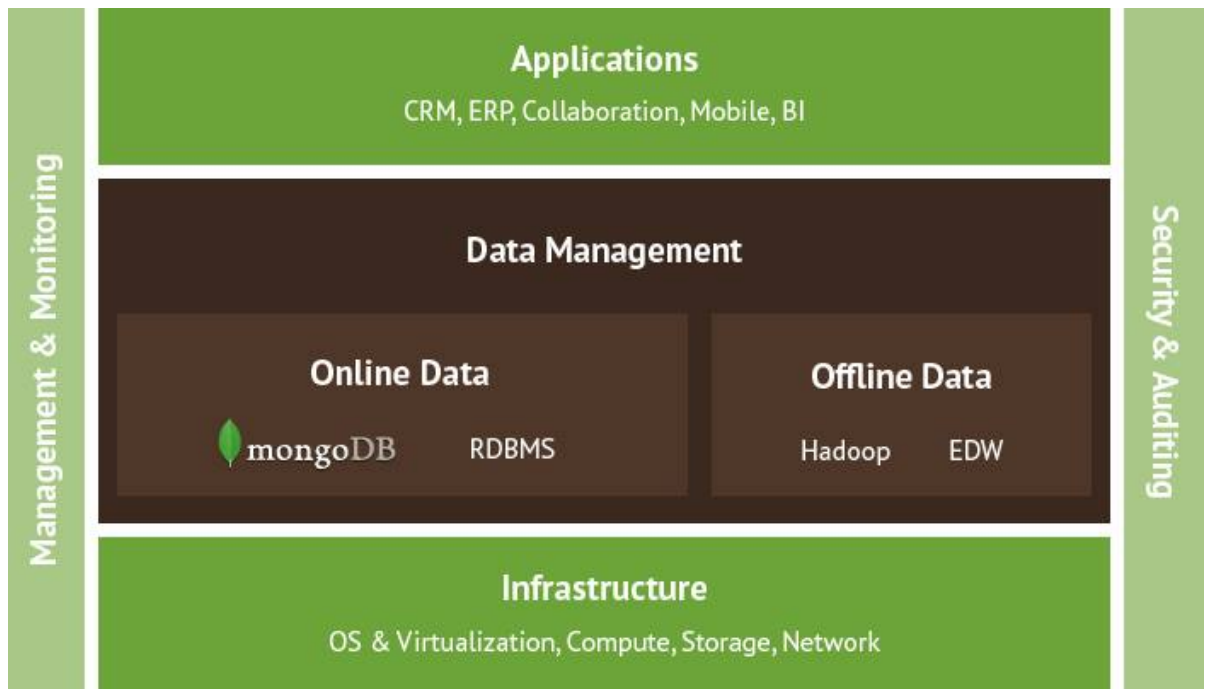


Abb. 3: MongoDB Einordnung⁸

Die Abbildung drei stellt die Einordnung von MongoDB in die unternehmerische Anwendungsarchitektur dar. Neben relationalen Datenbanksystemen, befindet sich MongoDB in der Datenhaltungsschicht im Bereich der Onlinedaten.

Die abgebildete Tabelle zeigt einige Eigenschaften von Mongo DB im Vergleich zum Datenbanksystem PostgreSQL. Die wesentlichen Unterschiede sind, dass Mongo DB ein dokumentenorientiertes und PostgreSQL ein relationales Datenbanksystem ist. Des Weiteren ist MongoDB ein wesentlich neueres und innovativeres System – es erschien 2009. PostgreSQL hingegen erschien bereits 1989. Beide Systeme unterliegen einer Open Source Lizenz. Die Quellcodes sind entsprechend öffentlich zugänglich und können von jedem Nutzer editiert werden. Die von PostgreSQL verwendete Lizenz ist ähnlich der BSD oder MIT Lizenz. Ein Nachteil von MongoDB ist die fehlende Trigger Funktion, die PostgreSQL hingegen bie-

⁷Vgl. MongoDB (o.J.b)

⁸Enthalten in: ebenda

Name	MongoDB	PostgreSQL
Kurzbeschreibung	Einer der meistverwendeten Document Stores	Nachfolger des objektrelationalen DBMS Postgre
DB-Engines Ranking	Rang 5	Rang 4
Erscheinungsjahr	2009	1989
Lizenz	AGPL	PostgreSQL Lizenz
Datenbankmodell	Document Store	Relationalen DBMS
Server-seitige Scripts	JavaScript	Benutzerdefinierte Funktionen
Triggers	Nein	Ja
MapReduce	Ja	Nein

Tabelle 1: Vergleich von MongoDB mit PostgreSQL

tet. Mit Hilfe von Datenbanktrigger lässt sich Code ausführen, wenn ein bestimmtes Ereignis eintritt. Typische Ereignisse sind das Einfügen, Ändern oder Löschen von Datensätzen. Trigger können selbst Daten in Datenbanken ändern, was das Auslösen weiterer Trigger bewirken kann. Ein Vorteil hingegen von MongoDB im Vergleich zu PostgreSQL ist die MapReduce Funktion. MapReduce ist ein Algorithmus für parallele Verarbeitung in großen Datenbeständen. Diese Funktion wird hauptsächlich in NoSQL Datenbanksystemen verwendet und gewährleistet unter anderem die hohe Performance der Datenbanken. Ein zusätzlicher wesentlicher Unterschied liegt darin, dass die Daten in MongoDB in Form von JSON Dateien abgespeichert werden. Die Daten in PostgreSQL werden in den herkömmlichen Typen, wie beispielsweise Integer, String o.ä. gespeichert.

3 Klassifikation

3.1 Grundlagen der Klassifikation

Klassifikation kann als Teilgebiet des Data Mining verstanden werden. Das Ziel des Data Mining ist, aus Daten Wissen zu extrahieren. Unter dem Begriff Wissen versteht man interessante Muster. Interessant sind jene Muster, die allgemein gültig sind, nicht trivial, nützlich, neu und verständlich. Wie stark diese Eigenschaften auf die erkannten Muster zutreffen, muss je nach Anwendung neu bestimmt werden. Hierbei muss mit Experten dieser Anwendung zusammengearbeitet werden. Dadurch, dass Experten einbezogen werden, entsteht ein rückgekoppelter Prozess, der oft mehrfach durchlaufen wird, bis ein zufriedenstellendes Resultat erreicht ist. In den meisten Anwendungen stehen erstmal nur Rohdaten zur Verfügung. Diese sind in der Regel fehlerhaft, unvollständig, ungeordnet, teilweise redundant oder unwichtig. Bevor also die eigentliche Analyse der Daten durchgeführt werden kann, ist eine Vorverarbeitung und Vorbereitung der Daten notwendig. Zur Datenvorbereitung gehört, dass aus den teilweise unübersichtlichen Daten geeignete Merkmale generiert und die überhaupt interessanten Daten ausgewählt werden. Nach der Datenvorbereitung steht ein Rohdatensatz zur Verfügung, der in der anschließenden Vorverarbeitung normalisiert, gesäubert, gefiltert oder transformiert wird. Fehlende Einträge werden ergänzt und Fehler korrigiert. Der auf diese Weise erhaltene vorverarbeitete Datensatz kann anschließend mit verschiedenen Verfahren der Datenanalyse verarbeitet werden.

Bei der Klassifikation werden Daten von Objekten betrachtet, die jeweils einer von c Klassen angehören mit $c \in \{2, 3, \dots\}$. Als Beispiel können hierfür Untersuchungsergebnisse in der Medizin genannt werden. Diese Ergebnisse liefern meist numerische Merkmale wie Temperatur oder Blutdruck und dienen der Beurteilung, ob ein Patient entweder krank oder gesund ist. Dies entspricht einer Klassifikation in die Klasse der Gesunden und die der Kranken. Eine Vorschrift, die eine solche Klassifikation vornimmt, wird Klassifikator genannt.

Für Klassifikatoren können bestimmte Kenngrößen ermittelt werden, die eine Beurteilung der Klassifikationsgüte ermöglichen. Dafür wird eine der möglichen Klassen ausgewählt, für das obige Beispiel etwa die Klasse der gesunden Patienten. Ein idealer Klassifikator klassifiziert einen gesunden Patienten stets auch als einen solchen. Im Fehlerfall würde er als krank klassifiziert werden. Um diesbezüglich Aussagen über die Klassifikationsgüte treffen zu können, werden vier Fälle bei der Klassifikation unterschieden:

1. richtig positiv oder true positive (TP) bedeutet, dass ein kranker Patient als krank klassi-

fiziert wird

2. richtig negativ oder true negative (TN) bedeutet, dass ein gesunder Patient als gesund klassifiziert wird
3. falsch positiv oder false positive (FP) bedeutet, dass ein gesunder Patient als krank klassifiziert wird
4. falsch negativ oder false negative (FN) bedeutet, dass ein kranker Patient als gesund klassifiziert wird

Allgemein wird der Fall FP als Fehler erster Art und FN als Fehler zweiter Art bezeichnet. Diese Kenngrößen bilden die Basis für weitere abgeleitete Kenngrößen:

- Korrektklassifikationen $T=TP+TN$ (Anzahl der richtig klassifizierten Patienten)
- Falschklassifikationen $F=FP+FN$ (Anzahl der falsch klassifizierten Patienten)
- Relevanz $R=TP+FN$ (Anzahl der kranken Patienten)
- Irrelevanz $I=FP+TN$ (Anzahl der gesunden Patienten)
- Positivität $P=TP+FP$ (Anzahl der als krank klassifizierten Patienten)
- Negativität $N=TN+FN$ (Anzahl der als gesund klassifizierten Patienten)
- Korrektklassifikationsrate T/n (Wahrscheinlichkeit, dass ein Patient richtig klassifiziert wird)
- Falschklassifikationsrate F/n (Wahrscheinlichkeit, dass ein Patient falsch klassifiziert wird)
- Richtig-Positiv-Rate, Sensitivität, Trefferquote oder Recall $TPR=TP/R$ (Wahrscheinlichkeit, dass ein kranker Patient als krank klassifiziert wird)
- Richtig-Negativ-Rate oder Spezifität $TNR=TN/I$ (Wahrscheinlichkeit, dass ein gesunder Patient als gesund klassifiziert wird)
- Falsch-Positiv-Rate, Ausfallrate oder Falschalarmrate $FPR=FP/I$ (Wahrscheinlichkeit, dass ein gesunder Patient als krank klassifiziert wird)
- Falsch-Negativ-Rate $FNR=FN/R$ (Wahrscheinlichkeit, dass ein kranker Patient als gesund klassifiziert wird)
- positiver Vorhersagewert, Genauigkeit oder Präzision TP/P (Wahrscheinlichkeit, dass ein krank klassifizierter Patient krank ist)
- negativer Vorhersagewert TN/N (Wahrscheinlichkeit, dass ein gesund klassifizierter Patient gesund ist)
- negative Falschklassifikationsrate FN/N (Wahrscheinlichkeit, dass ein gesund klassifi-

zierter Patient krank ist)

- positive Falschklassifikationsrate FP/P (Wahrscheinlichkeit, dass ein krank klassifizierter Patient gesund ist)⁹

3.2 Der naive Bayes-Klassifikator

3.2.1 Die Formel von Bayes

Die Formel von Bayes basiert auf dem Multiplikationssatz für unabhängige Ereignisse und der Formel für die totale Wahrscheinlichkeit. Zwei Ereignisse bezeichnet man als unabhängig, wenn gilt:

$$P(B|A) = P(B) \text{ mit } P(A) > 0 \text{ bzw.}$$

$$P(A|B) = P(A) \text{ mit } P(B) > 0 \text{ bzw.}$$

$$P(A \cap B) = P(A) \cdot P(B)$$

Davon ist die letzte Beziehung der Multiplikationssatz für unabhängige Ereignisse. Treten zwei Ereignisse also unabhängig voneinander auf, so ist die Wahrscheinlichkeit für das gleichzeitige Auftreten dieser Ereignisse gleich der Multiplikation ihrer Wahrscheinlichkeiten.

Die Formel für die totale Wahrscheinlichkeit lässt sich wie folgt darstellen. Für beliebige unvereinbare Ereignisse $E_1, E_2, \dots, E_n = \Omega$ gilt

$$P(A) = \sum_{k=1}^n P(A \cap E_k) = \sum_{k=1}^n P(E_k)P(A|E_k)$$

Aus dem Multiplikationssatz und der Formel für die totale Wahrscheinlichkeit ergibt sich die folgende Beziehung, die nach Thomas Bayes benannt ist:

$$P(E_j|A) = \frac{P(E_j)P(A|E_j)}{P(A)} = \frac{P(E_j)P(A|E_j)}{\sum_{k=1}^n P(E_k)P(A|E_k)}$$

⁹Vgl. Runkler, T. (2010), S. 85 ff.

Hierbei sind E_1, E_2, \dots, E_n unvereinbare Ereignisse. Aus deren Vereinigung ist A ein beliebiges Ereignis. Die Formel von Bayes berechnet also die Wahrscheinlichkeit, dass eines der Ereignisse E_j unter der Bedingung von A eintritt. Speziell für den Fall, dass nur zwei Ereignisse $E_1 = E$ und $E_2 = \bar{E}$ auftreten lautet die Formel von Bayes:

$$P(E|A) = \frac{P(E)P(A|E)}{P(E)P(A|E) + P(\bar{E})P(A|\bar{E})}$$

Damit sind die Grundlagen des naiven Bayes-Klassifikator, dessen Funktionsweise Thema das folgenden Abschnittes ist, behandelt worden.¹⁰

3.2.2 Funktionsweise

Anhand eines Beispiels soll nun die Funktionsweise eines naiven Bayes-Klassifikator dargestellt werden. Der Klassifikator wird oft bei Spam-Filtern eingesetzt, um möglichst zuverlässig anhand des Inhaltes von Mails eine Einordnung in Spam oder Nicht-Spam durchzuführen. Dies sind die zwei Klassen von E-Mails.

$$C = \{Spam, \overline{Spam}\}$$

Eine Mail besteht aus einzelnen Worten W_i . Für diese Worte kann man die Wahrscheinlichkeit schätzen, dass sie in Nicht-Spam- oder Spam-Mails vorkommen mit den folgenden Formeln.

$$P(W_i|Spam) = \frac{\text{Anzahl der Spam - Mails mit Wort } W_i}{\text{Anzahl der Spam - Mails}}$$

$$P(W_i|\overline{Spam}) = \frac{\text{Anzahl der Nicht - Spam - Mails mit Wort } W_i}{\text{Anzahl der Nicht - Spam - Mails}}$$

Wenn eine neue E-Mail im Postfach ankommt muss entschieden werden, ob $P(Spam|W) < P(\overline{Spam}|W)$, also die Wahrscheinlichkeiten dass, wenn Mail W eintrifft, sie Spam ist oder nicht. Besteht die Mail W aus den Wörtern W_1, \dots, W_n , so kann folgender Quotient berechnet werden.

$$Q = \frac{P(Spam|W)}{P(\overline{Spam}|W)} = \frac{P(W_1|Spam) \dots P(W_n|Spam)P(Spam)}{P(W_1|\overline{Spam}) \dots P(W_n|\overline{Spam})P(\overline{Spam})}$$

Ist dieser Quotient größer 1, ist die Wahrscheinlichkeit, dass die Mail Spam ist rechnerisch

¹⁰Vgl. Teschl, G./Teschl, S. (2013), S. 253 ff.

höher, als dass sie kein Spam ist. Die klassifizierte Mail wird anschließend in die Menge der Mails aufgenommen und dient der anschließenden Klassifikation weiterer Mails. So wird der Bayes-Klassifikator mit wachsender Menge an Objekten genauer.¹¹

Ein großer Vorteil des naiven Bayes-Klassifikators ist die Effizienz, da die Trainingsdaten zur Bestimmung der relativen Häufigkeit nur einmal durchlaufen werden müssen. Unvollständige Einträge in den Daten oder Einträge, die nicht plausibel sind, können einfach ignoriert werden. Der Bayes-Klassifikator kann nicht nur mit binären Merkmalen umgehen, sondern auf beliebige diskrete Merkmale erweitert werden.¹²

¹¹Vgl. Wikipedia (2014)

¹²Vgl. Runkler, T. (2010), S. 91

4 Implementierung

Für die Implementierung wurde ein Bayes-Klassifikator genutzt. Dazu wurde eine Klasse *BOOT* geschrieben. *BOOT* ist als Akronym für Bayes Objekt Orientierte Textanalyse zu verstehen. Neben der Klasse *BOOT* gibt es noch ein weiteres Skript *rest.py*, welches die Funktionalität für einen rudimentären REST-Service darstellt.

REST steht für Representational State Transfer und stellt ein Programmierparadigma dar. Ziel ist es, einfache Services zu schreiben, die anschließend von beliebig vielen anderen Services genutzt werden können. So wurde für eine rudimentäre Oberfläche eine Webseite programmiert, die mittels JavaScript auf diesen REST-Service zugreift. Für eine weitere Verwendung könnte man nun beispielsweise den Posteingang ebenfalls dort ansbinden und die Scans der Briefe dort einleiten. REST gibt fünf Eigenschaften vor, die der Service haben muss, die individuelle Implementierung bleibt jedoch offen.

1. **Adressierbarkeit** Ein Service muss eine eindeutige Adresse haben. Des Weiteren muss jede Ressource ebenso eine eindeutige Adresse haben.
2. **Repräsentationen** Ein Service kann verschiedene Repräsentationen ausliefern, beispielsweise JSON und/oder XML. Dabei sollte jedoch mindestens ein solcher Standard verwendet werden. Das einfache zurückgeben eines Datenbankeintrags im Plain-Text sollte unterbleiben.
3. **Zustandslosigkeit** Ein REST-Service darf keine Zustände speichern, das heißt er sollte jede Anfrage wie die erste eines neuen Clients ansehen. Sessions die im Webumfeld sehr häufig zum Einsatz kommen dürfen nicht verwendet werden.
4. **Operationen** Da HTTP konform zu REST ist, sollten die Möglichkeiten von HTTP verwendet werden. So schreibt HTTP vor, dass die Methode *GET* sicher ist, also keine Datenänderung zur Folge hat. Die anderen Methoden *PUT*, *HEAD* und *DELETE* müssen idempotent sein, also bei mehrmaliger gleicher Ausführung genauso wie ein einfacher Aufruf wirken.
5. **Hypermedia** REST-Services sollten es auch unterstützen, auf andere Services zu verweisen.

4.1 Die Klasse BOOT

Die Klasse BOOT besitzt einige statische Variablen, die Konfigurationsparameter enthalten. Dort ist zum Beispiel gespeichert, wie die Datenbank heißt, oder der Pfad zu den Wörterbüchern der Rechtschreibprüfung.

Des Weiteren kommen noch zwei statische Methoden zum Einsatz. Die Funktion *tokenizer* diese macht aus einem Fließtext eine Liste mit den einzelnen Wörtern. Die Funktion *spell.check* überprüft einen Text auf Rechtschreibfehler und verbessert diese dann mit der erst besten Möglichkeit. Dazu kommt ein externes Programm namens Hunspell zum Einsatz. Dieses wird auch bei anderen Programmen wie etwa Open-Office eingesetzt. Die hier verwendete Implementierung beherrscht leider nur das Überprüfen von einzelnen Wörtern, eine kontextorientierte Überprüfung ist leider derzeit nicht möglich.

Die nicht statische Methode *insert* schreibt einen Text, den rechtschreibgeprüften Text und die Wahrscheinlichkeit, ob es sich um eine Beschwerde handelt in die Datenbank. Für die Ermittlung der Wahrscheinlichkeit wird die Methode *rate* benutzt. Diese berechnet die Bayes-Wahrscheinlichkeit.

Mittels der Methode *categorize* ist es möglich einen Brief anhand der DokumentenID später noch als Beschwerde oder keine Beschwerde einzustufen. Dies ist wichtig um den Algorithmus später noch zu verbessern. Dabei sollte diese Einstufung nur von einem Menschen und nicht anhand der Ergebnisse des Algorithmus stattfinden.

Die Methode *show* dient dazu, die Briefe anzuzeigen.

Die Methode *create_knowledge* erzeugt das Wissen des Algorithmus. Dabei werden nur die manuell eingestuft Briefe analysiert und eine Wissensdatenbank gefüllt. Auf diese greift dann die Methode *rate* zu, um die Wahrscheinlichkeiten zu ermitteln.

Im folgenden werden nun die wichtigsten Methoden der Klasse *BOOT* beschrieben. Dabei wurden verschiedene Parameter empirisch bestimmt. Für diese Bestimmung wurden Tests durchgeführt. Diese Tests waren so aufgebaut, dass zunächst fünf zufällig gewählte Beschwerde und fünf Nicht-Beschwerde Schreiben ausgewählt wurden und eingelernt wurden. Daraufhin wurde die Wahrscheinlichkeit zu allen Test-Schreiben, auch den bereits eingelernten, ermittelt und überprüft ob diese richtig eingestuft werden. Dabei wurde eine Grenzwahrscheinlichkeit benötigt die teilweise im nachfolgenden *Grenze* genannt wird. Der Vorgang, inklusive dem auswählen der Schreiben, wurde nun 99 mal wiederholt. Anschließend wurde die Anzahl der einzulernenden Schreiben um 5 erhöht und abermals der Test 99 mal durchgeführt.

4.1.1 Methode tokenizer

Die Methode *tokenizer* ist wie folgt implementiert.

```

1 | @staticmethod
2 | def tokenizer(text):
3 |     import re
4 |     text = re.sub(u"^[a-z\säüöß]", u"␣", text.lower())
5 |     return_set = set()
6 |     for word in text.split():
7 |         if len(word) > 4:
8 |             return_set.add(word)
9 |     return return_set

```

Quelltext 1: Die Methode tokenizer

Dabei werden alle Zeichen abgesehen von Buchstaben durch Leerzeichen ersetzt und in Kleinbuchstaben geändert. Dies hat den Vorteil, dass man sich im Verlauf nicht um Groß- und Kleinschreibung kümmern muss. anschließend wird der String *text* bei den Leerzeichen in eine Liste gesplittet. Jedes Wort wird nun geprüft, ob es mehr als 4 Zeichen hat und anschließend in ein Set gespeichert. Dies hat den Vorteil, dass man sich zum einen nicht um kleine Füllwörter wie etwa Artikel, oder Adverbien kümmern muss. Diese kommen vermutlich gleich oft in Beschwerden und nicht Beschwerden vor und können ignoriert werden. Das Speichern in einem Set hat den Vorteil, dass jedes Wort nur einmal in dem Set gespeichert ist, man muss sich also um die Häufigkeit keine Gedanken machen.

Die Wortlänge von 4 oder mehr Zeichen wurde empirisch ermittelt. Dabei wurden schrittweise immer mehr Schreiben eingelernt und nach jeweils 5 Schreiben getestet, wieviele aller Schreiben der Algorithmus richtig kategorisiert. Die Grenze, mit der ein Schreiben als Beschwerde kategorisiert wird, betrug 100%. Also musste der Algorithmus eine Wahrscheinlichkeit von 100% ermitteln. Erst dann galt ein Brief als Beschwerde. Des Weiteren wurde in der Wissensdatenbank jedes Wort gespeichert, dass mindestens einmal vorkam. Eine Beschränkung auf häufige Worte fand nicht statt.

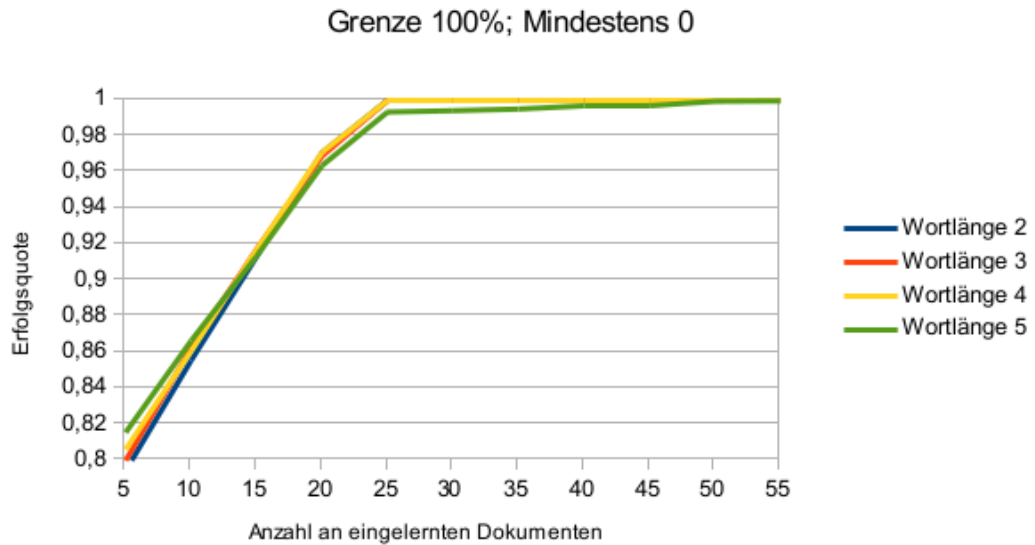


Abb. 4: Erfolgsquote im Verlauf der Anzahl eingelernter Dokumente bei variabler Wortlänge

Wie hier zu sehen ist, gab es nur einen sehr geringen Unterschied zwischen den Ergebnissen. Die Grenze von 100% wurde auch empirisch ermittelt. Dabei wurde die Wortlänge von 4 oder mehr Zeichen genutzt und wieder jedes Wort in der Wissensdatenbank gespeichert und genutzt. Die Grenze, ab welcher Wahrscheinlichkeit ein Schreiben als Beschwerde beziehungsweise nicht als solches gewertet wurde, lag bei 90% beziehungsweise bei 100%.

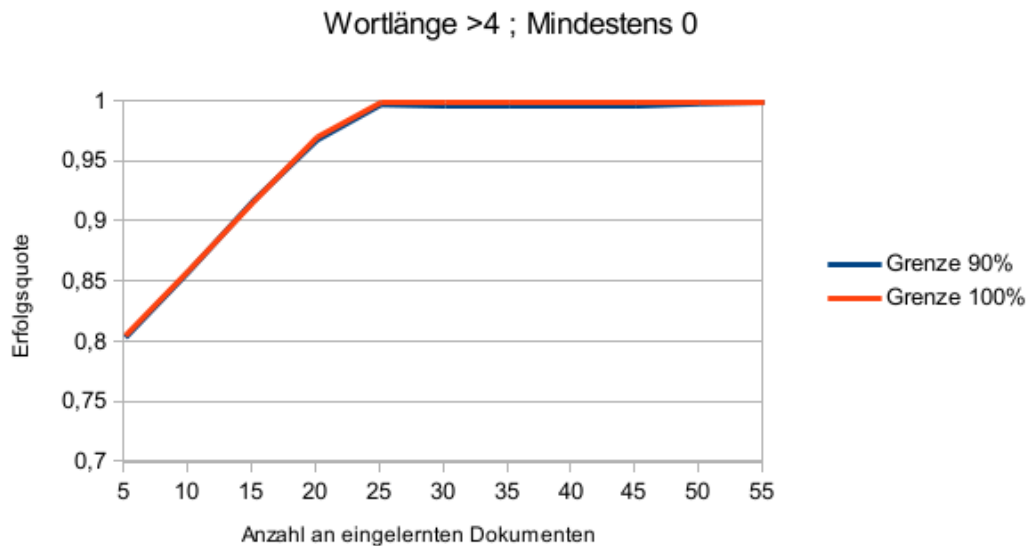


Abb. 5: Erfolgsquote im Verlauf der Anzahl eingelernter Dokumente bei variabler Wahrscheinlichkeitsgrenze

Hier ist der Graph mit der Grenze von 100% etwas weiter oben und daher wurde die Grenze von 100% gewählt.

4.1.2 Methode create_knowledge

Die Methode *create_knowledge* ist wie folgt implementiert.

```

1 def create_knowledge(self):
2     self.words.remove({})
3     complaints = 0
4     for doc in self.letters.find({"is_complaint": True}):
5         text = BOOT.tokenizer(doc["hunspell"])
6         for word in text:
7             ret = self.words.update({"word": word}, {"$inc":
8                 {"complaint_commonness": +1}})
9             if not ret["updatedExisting"]:
10                self.words.insert({"word": word,
11                    "complaint_commonness": 1,
12                    "non_complaint_commonness": 0})
13                complaints += 1
14
15        non_complaints = 0
16        for doc in self.letters.find({"is_complaint": False}):
17            text = BOOT.tokenizer(doc["hunspell"])
18            for word in text:
19                ret = self.words.update({"word": word}, {"$inc":
20                    {"non_complaint_commonness": +1}})
21                if not ret["updatedExisting"]:
22                    self.words.insert({"word": word,
23                        "complaint_commonness": 0,
24                        "non_complaint_commonness": 1})
25                    non_complaints += 1
26
27        self.words.insert({"stats": True,
28            "complaints": complaints,
29            "non_complaints": non_complaints})
30        #self.words.remove({"complaint_commonness": {"$lt": 5}, "
31            non_complaint_commonness": {"$lt": 5}})

```

Quelltext 2: Die Methode create_knowledge

Zunächst wird die Wissensdatenbank, beziehungsweise die Collection geleert. Anschließend werden alle manuell als Beschwerde eingestuftem Schreiben durchiteriert. Diese besitzen das Attribut *is_complaint* mit dem booleschen Wert *true*. Von diesen Schreiben wird der recht-schreibgeprüfte Text genutzt der unter dem Attribut *hunspell* gespeichert ist. Dieser wird mittels des *tokenizer* in einzelne Worte geteilt und diese werden wiederum iteriert. Für jedes Wort wird der Datenbankeintrag geupdated und das Attribut *complaint_commonness* um eins erhöht. Falls es das Wort in der Datenbank noch nicht gab, wird es eingefügt. Dabei wird ein JSON mit folgendem Format eingefügt:

```

1 | {
2 |   "word": "Wort",
3 |   "complaint_commonness": 1,
4 |   "non_complaint_commonness": 0
5 | }
```

Quelltext 3: JSON eines Wort-Eintrags

Dazu wird der Zähler *complaints* um eins erhöht. Dieser zählt die Schreiben die als Beschwerde eingelernt wurden.

Nun wird das selbe für die als nicht Beschwerde kategorisierten Schreiben wiederholt. Der Ablauf ist analog.

Anschließend wird noch ein weiteres JSON in die Collection gespeichert. Dieses enthält die Anzahl der eingelernten Schreiben:

```

1 | {
2 |   "stats": True,
3 |   "complaints": 10,
4 |   "non_complaints": 10
5 | }
```

Quelltext 4: JSON des Statistik-Eintrags

Um Wörter nicht zu berücksichtigen, die nur selten vorkommen, könnten nun noch die Einträge gelöscht werden die nur selten vorkommen. Auch hier wurde empirisch die Auswirkung auf die Erfolgsquote ermittelt. Dabei wurden wieder nur Wörter berücksichtigt die mindestens vier Zeichen lang sind und einer Schwellwahrscheinlichkeit von 100% verwendet. Nun wurden in einzelnen Durchläufen keine Wörter aus der Wissensdatenbank gelöscht (*min_0*), Wörter, die seltener als 5 mal in entweder in Beschwerden oder nicht Beschwerden vorkamen (*min_5*),

oder Wörter, die weniger 10 mal vorkamen (*min_10*). Dabei wurden jedoch die Häufigkeiten nicht kumuliert. Ein Wort, das beispielsweise 9 mal in einer Beschwerde und 5 mal in nicht Beschwerde Schreiben vorkam, wurde gelöscht. Der folgende Graph zeigt den Verlauf der Erfolgsquote anhand der eingelernten Dokumente.

Wie man sieht dauert es bei den Testläufen mit nur geringer Einlern-Schreiben länger bis der

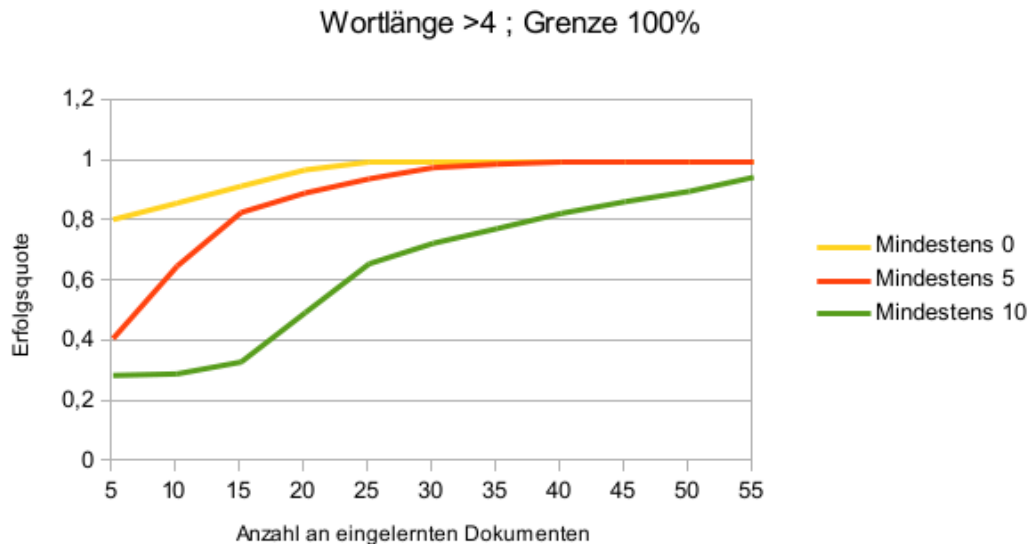


Abb. 6: Erfolgsquote des Algorithmus im Verlauf der Anzahl eingelernter Dokumente bei variabler Häufigkeit der Wörter

Erfolg einsetzt. Dies ist auch nachvollziehbar.

4.1.3 Das REST-Interface

Um die Komplexität so gering wie möglich zu halten, wurde hier nur die HTTP-Methode *POST* implementiert. Dabei wird ein JSON übergeben und anhand des Parameters *action* ermittelt was getan werden soll. So wird bei der Aktion *message* eine neue Nachricht, die wiederum im Attribut *text* gespeichert ist, in die Datenbank gespeichert. Dazu wird die Methode *insert* der *BOOT*-Klasse verwendet.

Weitere Aktionen sind *show* zum Anzeigen der Briefe, *categorize* zum manuellen Einstufen der Briefe und die Aktion *rebuild_knowledge* zum erneuten erzeugen des Wissens.

4.2 Das User-Interface

Um die Verwendung des *BOOT*-Services besser veranschaulichen zu können wurde eine Weboberfläche erstellt. Dazu kam das Bootstrap-Framework zum Einsatz. Es ist relativ einfach zu implementieren und hat viele gute Built-In Funktionen. Des weiteren bringt es direkt die

Unterstützung für responsives Design, also die Unterstützung von kleineren Displays, beispielsweise Smartphones.

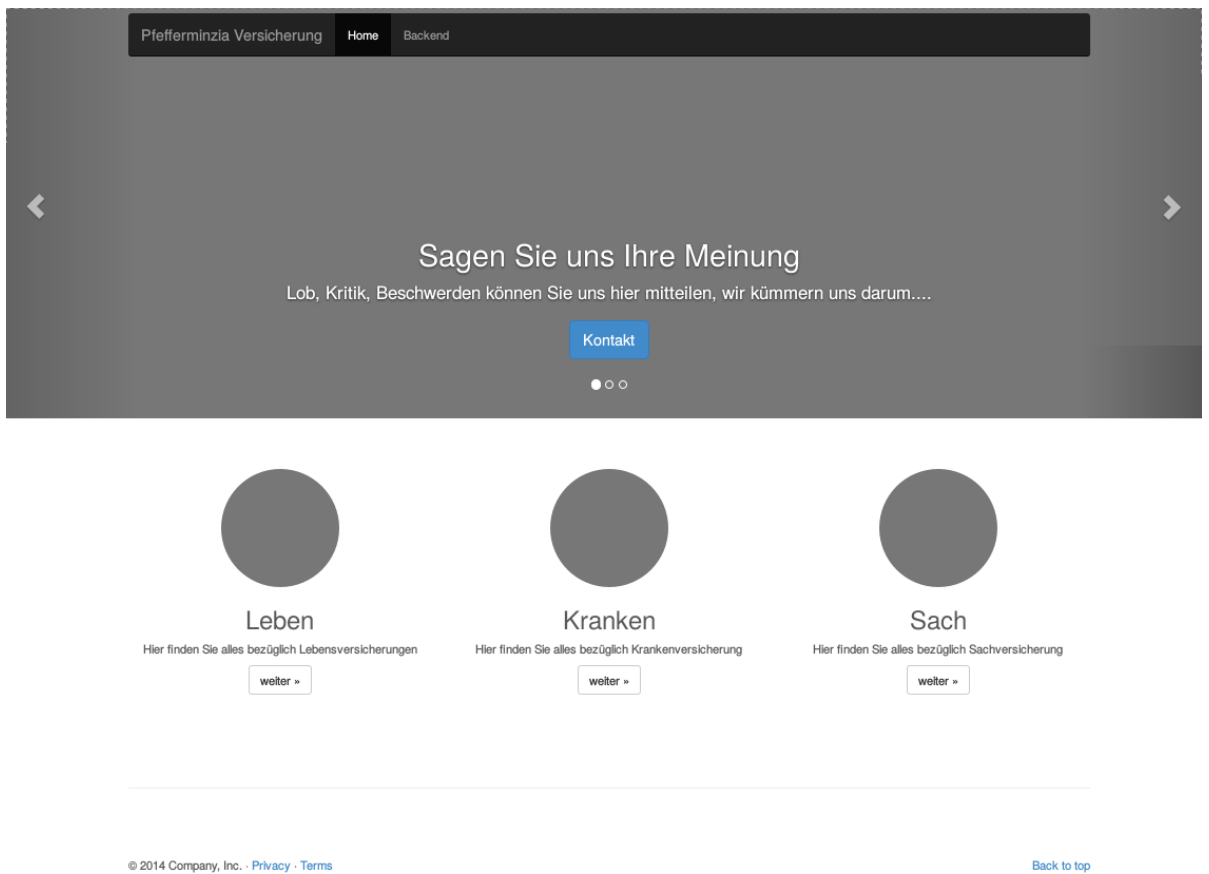


Abb. 7: Ansicht des Frontends

Hier ist dargestellt wie ein Kunde auf die Webseite einer fiktiven Versicherung gelangt. Nun kann er über den Button Kontakt ein Popup öffnen. Dieses bietet dem Kunden nun die Möglichkeit eine Nachricht an die Versicherung zu senden.

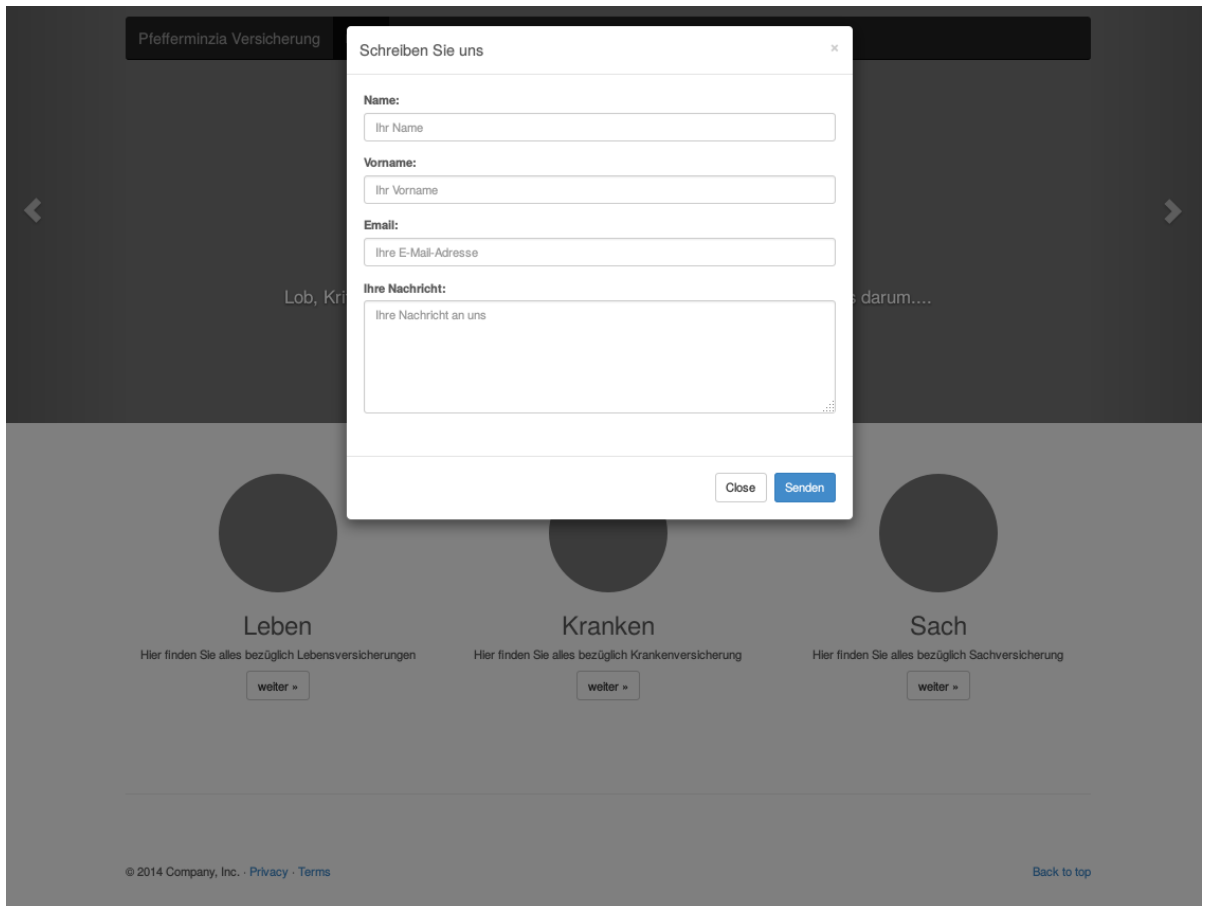


Abb. 8: Ansicht des Frontends 2

Nach dem Absenden wird dieser Brief umgehend in die Brief-Datenbank gespeichert und eine Wahrscheinlichkeit ermittelt, ob es sich um eine Beschwerde handelt.

Neben diesem Frontend, in dem die Kunden der Versicherung etwas schreiben können, gibt es noch ein Backend, in dem ein Versicherungsangestellter die Kundenschriften verwalten kann. Da dieses Backend nur zu Demonstrationszwecken erstellt wurde, wurde nur die notwendige Funktionalität implementiert.

Pfefferminzia Versicherung Home Backend Wissen neu machen	
Text	Wahrscheinlichkeit
Sehr geehrte Damen und Herren, die auf oben angegebenem Anwesen angemeldete Grundstücksbelastung ge	0
Sehr geehrte Damen und Herren, für die Gebäudeversicherung existiert eine Realrechtsbestätigung diese	0
Entschädigung Nichträumung ab September 2012 bis Auszug Sehr geehrte Damen und Herren, in vorbeze	0
Kündigung der Versicherung Kunden-Nr.:P 1234 VK 123 Sehr geehrte Damen und Herren, Hiermit möchte	0
Rechtsschutzversicherung V 007 Sehr geehrte Damen und Herren, Ich möchte Sie hiermit informieren, da	0
Sehr geehrte Damen und Herren, Ihr Versicherungsnehmer hat uns mit der Interessenwahrnehmung und Ver	0
Schadensnummer: 12-1 Sehr geehrte Damen und Herren, anbei erhalten Sie die Schadensrechnung mit A	0
Sehr geehrte Damen und Herren, in der obigen Angelegenheit habe ich vom Amtsgericht xy-Hasengarten	0
WIDERSPRUCH Kraftfahrtversicherung Nr. V 123456/0 Sehr geehrte Damen und Herren, meine Kfz-Versich	1
Hausratsversicherung-Vers.Nr. P 123456 Schaden vom 19.11.2006 Massive Probleme bei der Schadensregul	1
Sehr geehrter Herr XY, der PKW meines verstorbenen Vaters soll ab Januar mit einem Oldtimerwechselken	1
Vers. Nr. V 1231 u. V 2345 u. V 5679 u. V 5896 Vers. Nr. P 3456 u. P 5479 Hier: Reklamation Sehr	1
Günstiger gehts's kaum/Ihr Schreiben an mich vom 02.11.2006/ Tarfsauskunftsnummer- Sehr geehrte Dam	1
Sehr geehrter Herr XY, vielen Dank für Ihr Schreiben vom 22.November 2006. Leider kann ich mit Ihrer E	1
Sehr geehrte Damen und Herren, bekanntlich ist die Versicherung eine der günstigsten Versicherungen,	1

< zurück | weiter >

© 2014 Pfefferminzia Versicherung · Privacy · Terms Back to top

Abb. 9: Ansicht des Backends

In der Backend-Ansicht sind nun die Schreiben der Kunden zu sehen. Aus Gründen der Übersichtlichkeit werden pro Seite nur 15 Schreiben angezeigt. Mit den Links *weiter* beziehungsweise *zurück* kann man durch die Ansicht durchblättern. Dabei werden in der ersten Spalte die ersten Worte des Schreibens angezeigt und in der zweiten die vom Algorithmus ermittelten Wahrscheinlichkeiten. Wenn das Schreiben manuell als Beschwerde oder Nicht-Beschwerde kategorisiert wurde wird die Wahrscheinlichkeit auf 1 also 100% oder 0 gerundet. Schreiben mit einer Wahrscheinlichkeit von unter 80% werden grün hinterlegt. Schreiben mit einer Wahrscheinlichkeit zwischen 80% und 100% werden gelb hinterlegt und die Schreiben mit einer Wahrscheinlichkeit von 100% rot.

The screenshot shows a web application interface for 'Pfefferminzia Versicherung'. A modal window titled 'Großansicht' is open, displaying a text snippet and three buttons: 'Beschwerde', 'keine Beschwerde', and 'Close'. The background shows a table with columns 'Text' and 'Wahrscheinlichkeit' (Probability). The table contains several rows of text and probability values (0 or 1). The page header includes 'Pfefferminzia Versicherung', 'Home', 'Backend', and 'Wissen neu machen'. The footer includes '© 2014 Pfefferminzia Versicherung · Privacy · Terms' and 'Back to top'.

Text	Wahrscheinlichkeit
Sehr geehrte Damen und Herren, die auf ob...	0
Sehr geehrte Damen und Herren, für die Geb...	0
Entschädigung Nichträumung ab September	0
Kündigung der Versicherung Kunden-Nr.: P 1	0
Rechtsschutzversicherung V 007 Sehr geehr	0
Sehr geehrte Damen und Herren, Ihr Versich	0
Schadensnummer: 12-1 Sehr geehrte Dame	0
Sehr geehrte Damen und Herren, in der obig	0
WIDERSPRUCH Kraftfahrtversicherung Nr. V 123456/0 Sehr geehrte Damen und Herren, meine Kfz-Versich	1
Hausratversicherung-Vers.Nr. P 123456 Schaden vom 19.11.2006 Massive Probleme bei der Schadensregul	1
Sehr geehrter Herr XY, der PKW meines verstorbenen Vaters soll ab Januar mit einem Oldtimerwechselken	1
Vers. Nr. V 1231 u. V 2345 u. V 5679 u. V 5896 Vers. Nr. P 3456 u. P 5479 Hier: Reklamation Sehr	1
Günstiger gehts's kaum/Ihr Schreiben an mich vom 02.11.2006/ Tarfsauskunftsnummer- Sehr geehrte Dam	1
Sehr geehrter Herr XY, vielen Dank für Ihr Schreiben vom 22.November 2006. Leider kann ich mit Ihrer E	1
Sehr geehrte Damen und Herren, bekanntlich ist die Versicherung eine der günstigsten Versicherungen,	1

Abb. 10: Ansicht des Backends 2

Klickt man nun auf eine Zeile erscheint ein Pop-Up mit dem gesamten Text, sowie drei Buttons *Beschwerde*, *keine Beschwerde* und *Close*. Mit den ersten beiden Buttons kann ein Angestellter die Schreiben einzeln kategorisieren. Über den Button *Close* sowie das *X* rechts oben kann das Pop-Up wieder geschlossen werden. Über den Link *Wissen neu machen* in der Navigationsleiste kann eine neu Initialisierung der Wissens-Datenbank angestoßen werden. Dies geschieht noch in Echtzeit da dieser Vorgang bei der derzeitigen Anzahl an Schreiben relativ kurz ist, wenn die Menge aber steigt sollte dieser Vorgang asynchron ausgeführt werden.

5 Fazit

Die Ergebnisse des hier implementierten Bayes-Klassifikator sind sehr zufriedenstellend und lohnen sich durchaus mit einer deutlich größeren Menge an Schreiben getestet zu werden. Dabei sollten die Parameter weiter getestet werden, da es durchaus sein kann, dass aus Performance-Gründen eine Anpassung nötig wird.

Die Performance bei dieser Implementierung war durchweg zufriedenstellend, jedoch war die Wissensdatenbank aufgrund der geringen Anzahl an Testschreiben auch relativ klein. Eine Beschränkung wie in 4.1.2 beschrieben kann bei einer größeren Wissensdatenbank durchaus notwendig werden.

Auch eine Implementierung dieses Algorithmus in einer anderen Programmiersprache ist denkbar. Da Python eine interpretierte Sprache ist, ist diese im Vergleich zu einer Kompilersprache etwas langsamer. Eine Implementierung in C wäre anzuraten. Auch eine Implementierung in Cython könnte sich lohnen. Dabei wird Python-Quelltext in C oder kompiliert, ohne das man den Quellcode in großem Umfang umschreiben muss.

6 Quellenverzeichnisse

Literaturverzeichnis

u.a. Bachmann (2014): Big Data – Fluch oder Segen?: Unternehmen im Spiegel gesellschaftlichen Wandels, Hamburg: Verlagsgruppe Hühthig Jehle Rehm GmbH

Leiner, V. (2007): Beschwerdemanagement – ein Instrument zur Kundenbindung, Norderstedt: Grin Verlag

Teschl, G./Teschl, S. (2013): Mathematik für Informatiker, 4. Aufl., Berlin/Heidelberg: Springer-Verlag

Tiwari, S. (2011): Professional NoSQL, Indianapolis: John Wiley & Sons, Inc.

Runkler, T. (2010): Data Mining, Wiesbaden, GWV Fachverlage GmbH

Verzeichnis der Internet- und Intranetquellen

MongoDB (o.J.a): NoSQL Database, <http://www.mongodb.com/leading-nosql-database>, Abruf: 13.07.2014

MongoDB (o.J.b): MongoDB Überblick, <http://www.mongodb.com/mongodb-overview>, Abruf: 13.07.2014

Wikipeda (o.J.a): NoSQL, <http://de.wikipedia.org/wiki/NoSQL>, Abruf: 13.07.2014

Wikipeda (o.J.b): Dokumentenorientierte Datenbank, http://de.wikipedia.org/wiki/Dokumentenorientierte_Datenbank, Abruf: 13.07.2014

Wikipedia (2014): Bayes-Klassifikator, <http://de.wikipedia.org/wiki/Bayes-Klassifikator>, Abruf: 13.07.2014