

KOS.content

02 | 2014

Ergebnisse der Untersuchungen des
Kompetenzzentrum Open Source der DHBW-Stuttgart

Sommer 2014
band.2

INHALT BAND.2

Inhalt __

Vergleichsstudie zu Open Source Produkten für das Anforderungsmanagement __ 489

Evaluierung von Open Source Defect Management/ Bug-Tracking Tools __ 575

Vergleich und Prototyp von Open-Source-ETL-Tools (2) __ 677

Erweiterungen relationaler Datenbanken in Data-Warehouse-Systemen __ 753

Objektorientierte Datenverwaltung mit relationalen DB-Systemen __ 811

Konzepte und Einsatzszenarien von Wide-Column-Datenbanken __ 895

Das Kompetenzzentrum Open Source (KOS)

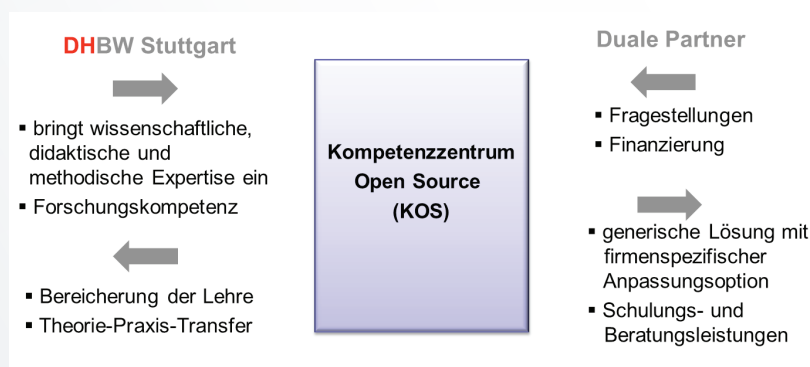
Ziel des Projektes

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 Werkzeugen für das Anforderungsmanagement

-

Open Source und kommerzielle Softwarelösungen
im Vergleich

Schriftliche Ausarbeitung
im Rahmen der Lehrveranstaltung „Integrationsseminar“

Vorgelegt von

Kevin Hahn,
Matthias Hajdu,
Yasemin Kocbinar,
Julika Schacht

am 14.07.2014

Fakultät Wirtschaft
Studiengang Wirtschaftsinformatik
WWI2011V

Inhaltsverzeichnis

Abkürzungsverzeichnis	IV
Abbildungsverzeichnis.....	V
Tabellenverzeichnis.....	VI
1 Herausforderungen für Unternehmen bei der Projektarbeit	1
1.1 Zielsetzung und methodischer Aufbau	2
2 Einsatz von Requirements Engineering im Projektumfeld	3
2.1 Grundlagen und Basiselemente des Requirements Engineering.....	3
2.1.1 Anforderungen – Definition und Merkmale	3
2.1.2 Arten von Anforderungen	4
2.1.3 VOLERE – ein Standard zur Anforderungsanalyse	5
2.1.4 Lebenszyklus des Requirements Engineering.....	6
2.2 Requirements Engineering im Projektumfeld.....	8
2.2.1 Arten von Vorgehensmodellen	8
2.2.2 Herausforderungen bei der Anforderungsanalyse in Projekten.....	9
2.3 Tools zur Unterstützung von Requirements Engineering.....	10
2.3.1 Grundlegende Lösungsarten von Requirements Engineering Tools.....	11
2.3.2 Möglichkeiten zur Erstellung von Marktanalysen.....	12
3 Vorgehensweise zur Erstellung einer unternehmensspezifischen Marktübersicht	15
3.1 Erläuterung der verwendeten Kriterienkategorien.....	15
3.2 Methodik zur Beschreibung eines Kriteriums.....	17
3.3 Erläuterung der verwendeten Kriterien	18
3.3.1 Aufbau des Kriterienkatalogs	18
3.3.2 Funktionale Anforderungen	19
3.3.3 Nichtfunktionale Anforderungen	32
3.4 Durchführung der Toolbewertung.....	34
4 Erstellung der Marktanalyse	37
4.1 Bildung der Shortlist	37
4.1.1 Marktüberblick.....	37
4.1.2 Die Auswahl der kommerziellen Tools für die Shortlist	39
4.1.3 Die Auswahl der Open Source Tools für die Shortlist	43

4.2	Anwendung des Kriterienkatalogs	48
4.2.1	Blueprint Requirements Center	49
4.2.2	Rational DOORS.....	49
4.2.3	Polarion Requirements.....	50
4.2.4	inteGREAT.....	50
4.2.5	Enterprise Architect.....	51
4.2.6	Caliber	52
4.2.7	Jama	52
4.2.8	PTC Integrity	53
4.2.9	Jira	53
4.2.10	Mingle	54
4.2.11	aNimble.....	55
4.2.12	Fazit der Toolbewertungen.....	56
4.3	Grafische Aufbereitung der Shortlist.....	57
4.3.1	Vorgehensweise	57
4.3.2	Beispiel für die Anpassung der Anforderungsgewichtung.....	60
4.3.3	Fazit.....	62
5	Ausblick.....	63
	Anhang.....	65
	Quellenverzeichnisse	75

Abkürzungsverzeichnis

CAD	computer-aided design
INCOSE	International Council on Systems Engineering
PLM	Product-Lifecycle-Management
PTC	Parametric Technology Corporation
RE	Requirements Engineering
REConf	Requirements Engineering Conference
ReqIF	Requirements Interchange Format
TDWG	Tools Database Working Group
WBT	Web Based Training

Abbildungsverzeichnis

Abb. 1: Aufwand für RE und Kostenüberschreitung bei NASA-Projekten	1
Abb. 2: Arten von Anforderungen nach VOLERE	5
Abb. 3: Karteikartenvorlage nach VOLERE	6
Abb. 4: Inhalte des Requirements Engineerings	7
Abb. 5: Einsatz von RE im Verlauf eines Projekts.....	7
Abb. 6: Aufbau des Gartner Magic Quadrant.....	13
Abb. 7: Ebenen des Kriterienkatalogs.....	18
Abb. 8: Beispielhafte Toolbewertung	35
Abb. 9: Tools für große Unternehmen	38
Abb. 10: Tools für mittlere Unternehmen	39
Abb. 11: Tools für kleine Unternehmen	39
Abb. 12: Bewertung des Tools Blueprint Requirements Center	49
Abb. 13: Bewertung des Tools Rational DOORS.....	50
Abb. 14: Bewertung des Tools Polarion Requirements.....	50
Abb. 15: Bewertung des Tools Enterprise Architect.....	51
Abb. 16: Bewertung des Tools Caliber	52
Abb. 17: Bewertung des Tools Jama	52
Abb. 18: Bewertung des Tools PTC Integrity	53
Abb. 19: Bewertung des Tools Jira.....	54
Abb. 20: Bewertung des Tools Mingle	54
Abb. 21: Bewertung des Tools aNimble.....	55
Abb. 22: Legende der Erläuterung der Kategoriefarben nach Erfüllungsgrad	58
Abb. 23: Beispielhafte Ergebniszeile	58
Abb. 24: Ergebnisübersicht der ungewichteten Kategorien.....	59
Abb. 25: Ergebnisübersicht der gewichteten Kategorien	61

Tabellenverzeichnis

Tab. 1: Anforderungsarten nach Rupp, Ebert, Partsch und VOLERE	4
Tab. 2: Kategorien der funktionalen Kriterien.....	15
Tab. 3: Kategorien der nichtfunktionalen Kriterien	16
Tab. 4: Allgemeiner Aufbau der Karteikarte zur Dokumentation einer Anforderung	17
Tab. 5: Basisdefinition einer Anforderung	19
Tab. 6: Änderung einer Anforderung	20
Tab. 7: Kopieren einer Anforderung	20
Tab. 8: Löschen einer Anforderung	20
Tab. 9: Visuelle Detaillierung einer Anforderung.....	21
Tab. 10: Anpassbarkeit der Anforderungsattribute	21
Tab. 11: Aufbereitung der Anforderungsdefinition	22
Tab. 12: Sortieren und Gruppieren von Anforderungen	22
Tab. 13: Filtern von Anforderungen	23
Tab. 14: Suchfunktion	23
Tab. 15: Erstellung von Reports	24
Tab. 16: Verteilung von Anforderungen	24
Tab. 17: Ressourcenplanung	25
Tab. 18: Priorisierung von Anforderungen	26
Tab. 19: Kommentarfunktion	26
Tab. 20: Nachvollziehbarkeit des Fortschritts	27
Tab. 21: Nachvollziehbarkeit einer Änderung	28
Tab. 22: Beziehungsmanagement.....	29
Tab. 23: Berechtigungsmanagement Ebene Anforderung	29
Tab. 24: Berechtigungsmanagement Ebene Tool.....	30
Tab. 25: Mehrbenutzerfähigkeit.....	30
Tab. 26: Import bestehender Anforderungen	31
Tab. 27: Export von Anforderungen.....	31
Tab. 28: Verknüpfung zu weiteren Aufgaben der Softwareentwicklung	32
Tab. 29: Usability und Erlernbarkeit.....	32
Tab. 30: Personalisierbarkeit.....	33
Tab. 31: Sicherheit der Daten (Verschlüsselung)	33
Tab. 32: Sicherung der Daten	33
Tab. 33: Herstellerbewertung	34
Tab. 34: Support	34
Tab. 35: Entscheidungsregel.....	36
Tab. 36: Definition der Unternehmensgrößen.....	37

Tab. 37: Ermittlung der Häufigkeit eines Tools	42
Tab. 38: Ergebnis des Vergleichs	42
Tab. 39: Ausgewählte kommerzielle Tools für die Bewertung mit dem Kriterienkatalog.....	43
Tab. 40: Direkter Ansprechpartner	44
Tab. 41: Release-Abstände	44
Tab. 42: Lizenzierung.....	44
Tab. 43: Aktivität des Forums/der Community	44
Tab. 44: Alter des Produkts	45
Tab. 45: Lizenztypen der Open Source Tools.....	46
Tab. 46: Das Durchschnittsalter der Open Source Tools	47
Tab. 47: Übersicht über die Aufnahme der Open Source Tools in die Shortlist.....	48
Tab. 48: Gesamtüberblick über die Bewertung der Tools	56
Tab. 49: Beispielhafte Gewichtung der Anforderungen.....	61

1 Herausforderungen für Unternehmen bei der Projektarbeit

Die Erstellung von Software wird in vielen Unternehmen in Projekten realisiert. Nach der Analyse und Definition erfolgen in der Regel der Entwurf und die Implementierung des Produkts. Anschließend Tests und die Integration der Software ermöglichen ihren Einsatz. Durch die regelmäßige Wartung der Softwareprodukte können diese weiterentwickelt und verbessert werden. Der richtige Einsatz von Anforderungsmanagement kann den gesamten Prozess unterstützen.

Der Zusammenhang zwischen dem effizienten Umgang mit Anforderungen und dem dadurch ermöglichten Projekterfolg ist nicht neu, aber er ist in der gegenwärtigen Situation bedeutender denn je und eine der Grundvoraussetzungen für den Erfolg von Unternehmen.

Um an dieses Thema heranzuführen, soll die folgende Abbildung verdeutlichen, wie sich Anforderungsmanagement bzw. Requirements Engineering auf die Kosten des Projekts und somit auch auf den Erfolg auswirken kann.

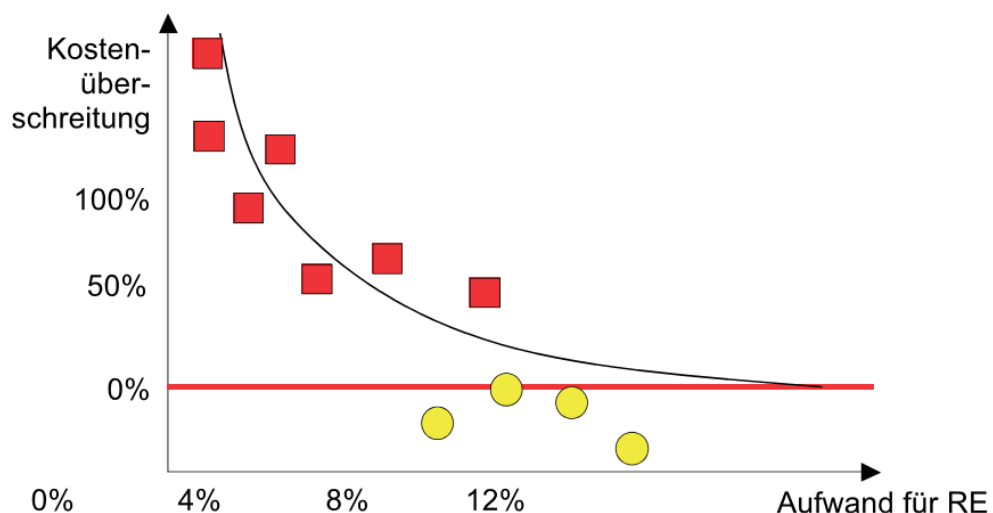


Abb. 1: Aufwand für RE und Kostenüberschreitung bei NASA-Projekten¹

Die Abbildung 1 zeigt die Kostenüberschreitungen von NASA-Projekten. Die roten Quadrate stehen für Projekte, in welche nur ein sehr geringer Aufwand an Requirements Engineering investiert wurde. Sie weisen eine hohe bis extrem hohe Kostenüberschreitung von bis zu 250% auf. Im Gegensatz dazu halten sich die Kosten von Projekten mit einem höheren Aufwand für Anforderungsmanagement (gelbe Punkte) in dem geplanten Rahmen. Dieses Beispiel zeigt, dass sich die Aufwände für Anforderungsmanagement sogar mehr als amortisieren können und die ausreichende Beachtung Konsequenzen, wie Kosten für spät auftretenden

¹ Enthalten in: Partsch, H. (2010), S. 13

de Änderungen oder anfängliche Fehleinschätzungen, auf ein Minimum reduziert.² Ein wichtiges Ziel von Requirements Engineering ist somit die Reduktion von Nacharbeitskosten.

1.1 Zielsetzung und methodischer Aufbau

Trotz der hohen Bedeutung des Anforderungsmanagements für die Projektarbeit wird in vielen Projekten die Anforderungsspezifikation überhaupt nicht, nicht zum richtigen Zeitpunkt oder nicht in einem angemessenen Ausmaß vorgenommen.³ In der vorliegenden Arbeit werden deshalb die Bedeutung und Auswirkungen von Requirements Engineering aufgezeigt und wie Projektbeteiligte dabei durch den Einsatz von Requirements Engineering Tools unterstützt werden können. Die Arbeit beschränkt sich somit auf die Requirements Engineering Tools für Projektarbeit. Darüber hinaus hat sie das Ziel, eine Marktübersicht für Unternehmen zu erstellen, aus der abgeleitet werden kann, welche Tools für welches Unternehmen geeignet sind. Dabei wird besonders auf den Vergleich zwischen proprietären und Open Source Werkzeugen Wert gelegt.

Die vorliegende Arbeit beschreibt im Grundlagenteil wichtige Elemente des Requirements Engineerings. Nach dieser kritischen Auseinandersetzung mit bestehenden Lösungsansätzen in Bezug auf Reifegradmodelle, Standards zur Anforderungsanalyse und Vorgehensweisen zur Auswertung von Tools und der Darstellung deren Ergebnisse erfolgt die Erstellung eines speziell auf diese Arbeit abgestimmte Vorgehensweise inklusive eines Kriterienkatalogs.

Anschließend können die Ergebnisse dieser Auseinandersetzung konkret auf den Markt für Requirements Engineering Tools angewendet werden. Daraus entsteht ein Marktüberblick, der branchenübergreifend von einer breiten Zielgruppe an Unternehmen genutzt und individuell erweitert werden kann. Diese Erweiterungsmöglichkeiten werden in der vorliegenden Arbeit nicht nur aufgezeigt, sondern auch konkrete Vorlagen und Templates mitgegeben. Die zugehörige Anleitung ermöglicht es den Unternehmen den allgemeinen Marktüberblick auf ihre spezifischen Bedürfnisse anzupassen und sich die geeigneten Tools anzeigen zu lassen.

² Vgl. Partsch, H. (2010), 13

³ Vgl. Partsch, H. (2010), 12

2 Einsatz von Requirements Engineering im Projektumfeld

Die Aussicht auf Erfolg ist die treibende Kraft für jede Beauftragung und die nachfolgende Durchführung von Projekten. Wie erfolgreich ein Projekt ist, wird u.a. durch die Effizienz des eingesetzten Requirements Engineering bestimmt. Kapitel 2 gibt ein einheitliches Verständnis über Anforderungen, deren Management und Einsatzgebiete in unterschiedlichen Projekten.

2.1 Grundlagen und Basiselemente des Requirements Engineering

Kapitel 2.1 beinhaltet grundlegende Begriffsbestimmungen und Standards zur Verwaltung von Anforderungen im Rahmen des Requirements Engineerings.

2.1.1 Anforderungen – Definition und Merkmale

Wie in allen Projekten werden auch in Softwareprojekten Möglichkeiten zur Messung des Fortschritts und des Erfolgs und zur Beschreibung von Erwartungen der Kunden an die Produkte bzw. die Dienstleistungen benötigt. Dies wird mithilfe von Anforderungen realisiert.

„Eine **Anforderung** ist eine Aussage über eine Eigenschaft oder Leistung eines Produktes, eines Prozesses oder der am Prozess beteiligten Personen.“⁴

Anhand von festgelegten Anforderungen können Stakeholder des Projekts Regeln bzgl. der Vorgehensweise und Eigenschaften bestimmen. Sie sind Bestandteile von Aufträgen, Verträgen oder Projektplänen und dienen als Grundlage für die Abschätzung, Planung, Steuerung und Kontrolle der Tätigkeiten während der gesamten Projektlaufzeit.⁵ Damit eine Anforderung nach diesen Eigenschaften konkret bewertet werden kann, muss sie gewisse Merkmalsausprägungen aufweisen. Zu diesen Merkmalen gehören u.a.:⁶

- Vollständigkeit
- Korrektheit
- Eindeutigkeit
- ausreichender Umfang

Individuell wird für jedes Projekt und jede Anforderung das nötige Ausmaß der einzelnen Merkmale entschieden. Somit erhalten die Projektbeteiligten ausreichend Informationen über den Grund des Entwurfs des Systems, den Umfang der Leistung und über die einzuhaltenen Einschränkungen.⁷

⁴ Vgl. Rupp, C. (2007), S. 13

⁵ Vgl. Ebert, C. (2005), S. 9

⁶ Vgl. Ebert, C. (2005), S. 10

⁷ Vgl. Partsch, H. (1991), S. 33 f.

2.1.2 Arten von Anforderungen

Um eine ausführliche und vollständige Betrachtung der Anforderungen vorzunehmen, ist neben der oben beschriebenen Charakterisierung der Merkmale **einer** Anforderung auch die Berücksichtigung **aller** zu nennenden Anforderungen notwendig. In der Literatur ist diese Berücksichtigung durch die Nennung aller möglichen Anforderungsarten gegeben. Ist die Aufzählung ausreichend umfangreich, können projektspezifisch die geforderten Anforderungen ausgewählt und beschrieben werden.

Die folgende Tabelle 1 zeigt eine Auswahl an unterschiedlichen Anforderungsarten nach Rupp, Ebert, Partsch und dem Standard VOLERE.

Rupp ⁸	Ebert ⁹	Partsch ¹⁰	VOLERE ¹¹
Funktionale Anforderungen	Prozessanforderungen	Funktionale Anforderungen	Projekttreiber
Technische Anforderungen	Funktionale Anforderungen	Nichtfunktionale Anforderungen	Randbedingungen für das Projekt
Anforderungen an die Benutzungsschnittstelle	Nichtfunktionale Anforderungen	Qualitätsanforderungen der gewünschten Funktionen	Funktionale Anforderungen
Qualitätsanforderungen		Anforderungen an das implementierte System als Ganzes	Nichtfunktionale Anforderungen
Anforderungen an sonstige Lieferbestandteile		Vorgaben für die Durchführung der Systemerstellung	Projektaspekte
Anforderungen an durchzuführende Tätigkeiten		Anforderungen an Prüfung, Einführung, Betrieb und Betreuung	
Rechtlich-vertragliche Anforderungen		Abhängigkeit der nicht-funktionalen Anforderungen von den funktionalen	

Tab. 1: Anforderungsarten nach Rupp, Ebert, Partsch und VOLERE

Die Gegenüberstellung in Tabelle 1 zeigt deutlich, dass die Anforderungskategorien „Funktionale Anforderungen“ und „Nichtfunktionale Anforderungen“ in allen vorgestellten Konzepten berücksichtigt werden.¹²

Auch für den praktischen Teil der vorliegenden Arbeit sollen diese zwei Kategorien übernommen werden. Ausgehend von funktionalen und nichtfunktionalen Anforderungen können Anforderungen definiert und ausformuliert werden und bilden auch im Vergleich zu anderen Ansätzen eine einheitliche Basis, die individuell von Unternehmen erweitert werden kann.

⁸ Vgl. Rupp, C. (2007), S. 15 f.

⁹ Vgl. Ebert, C. (2005), S. 11

¹⁰ Vgl. Partsch, H. (1991), S. 34-38

¹¹ Vgl. Roberson, J./Robertson, S. (2007), S. 2

¹² Rupp benennt zwar keine konkrete Kategorie nichtfunktional, allerdings weist sie ausdrücklich auf diese hin: „Jede Anforderung, die keine funktionale Anforderung ist, ist nichtfunktional.“ Vgl. dazu: Rupp, C. (2007), S. 16

Das folgende Kapitel betrachtet den Standard VOLERE, da dieser nicht nur durch die genannten Anforderungsarten eine umfängliche Auswahl an Anforderungen bietet, sondern auch ein Muster für die Ermittlung von Anforderungen. Somit unterstützt VOLERE die Beschreibung und Berücksichtigung von Anforderungen.

2.1.3 VOLERE – ein Standard zur Anforderungsanalyse

VOLERE ist eine Sammlung von Werkzeugen und Materialien um das Thema Requirements Engineering im Softwareentwicklungsprozess. Da dieses Thema auch die Zielrichtung der vorliegenden Arbeit ist, werden im Folgenden die bereits genannten Anforderungsarten genauer beschrieben und die Methode von VOLERE zur Handhabung von Anforderungen erläutert.

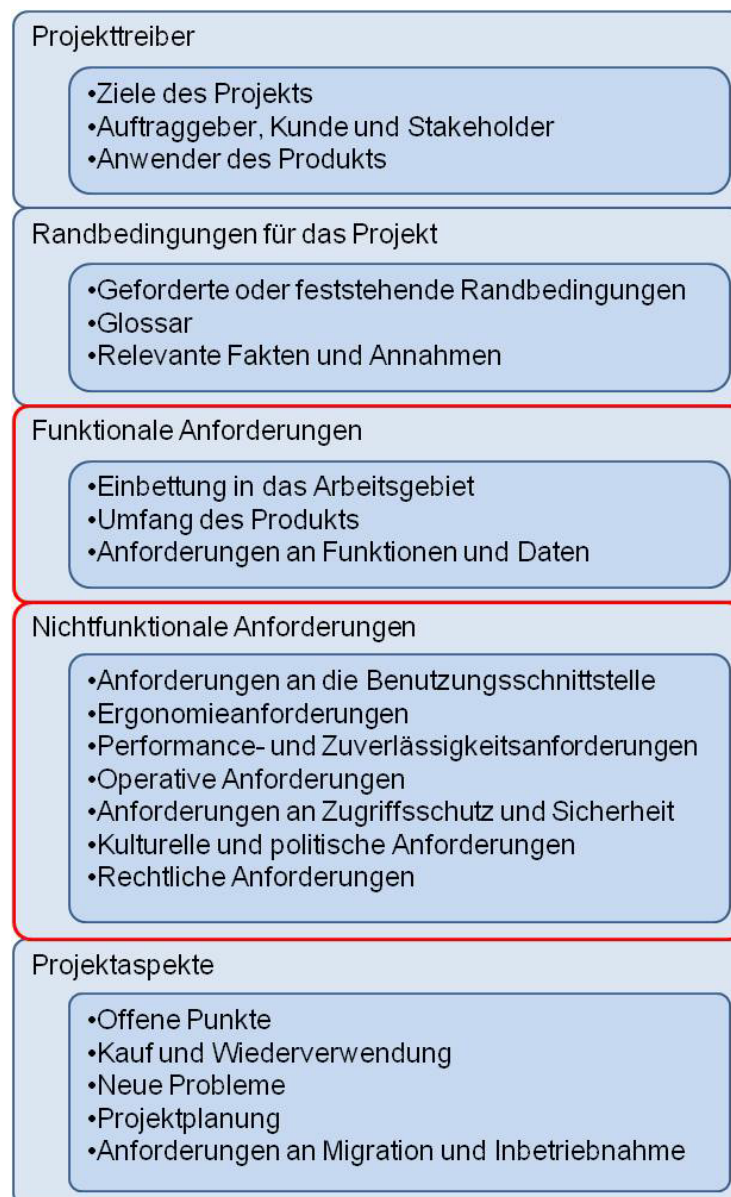


Abb. 2: Arten von Anforderungen nach VOLERE¹³

¹³ Enthalten in: Roberson, J./Robertson, S. (2007), S. 2

Abbildung 2 erläutert mithilfe der Unterkategorien, welche Anforderungen zu den Arten berücksichtigt werden sollen. Der praktische Teil der vorliegenden Arbeit greift den bestehenden VOLERE-Ansatz bezogen auf den rot markierten Bereich wieder auf. Die Anordnung und Bezeichnungen der Unterkategorien werden jedoch im Laufe des praktischen Teils modifiziert und neue Aspekte mit aufgenommen. VOLERE unterstützt mit den Anforderungsarten aus Abbildung 2 das **Finden** von Anforderungen, aber auch das **Dokumentieren** mithilfe der folgenden Karteikartenvorlage für die Erstellung einer Anforderung.

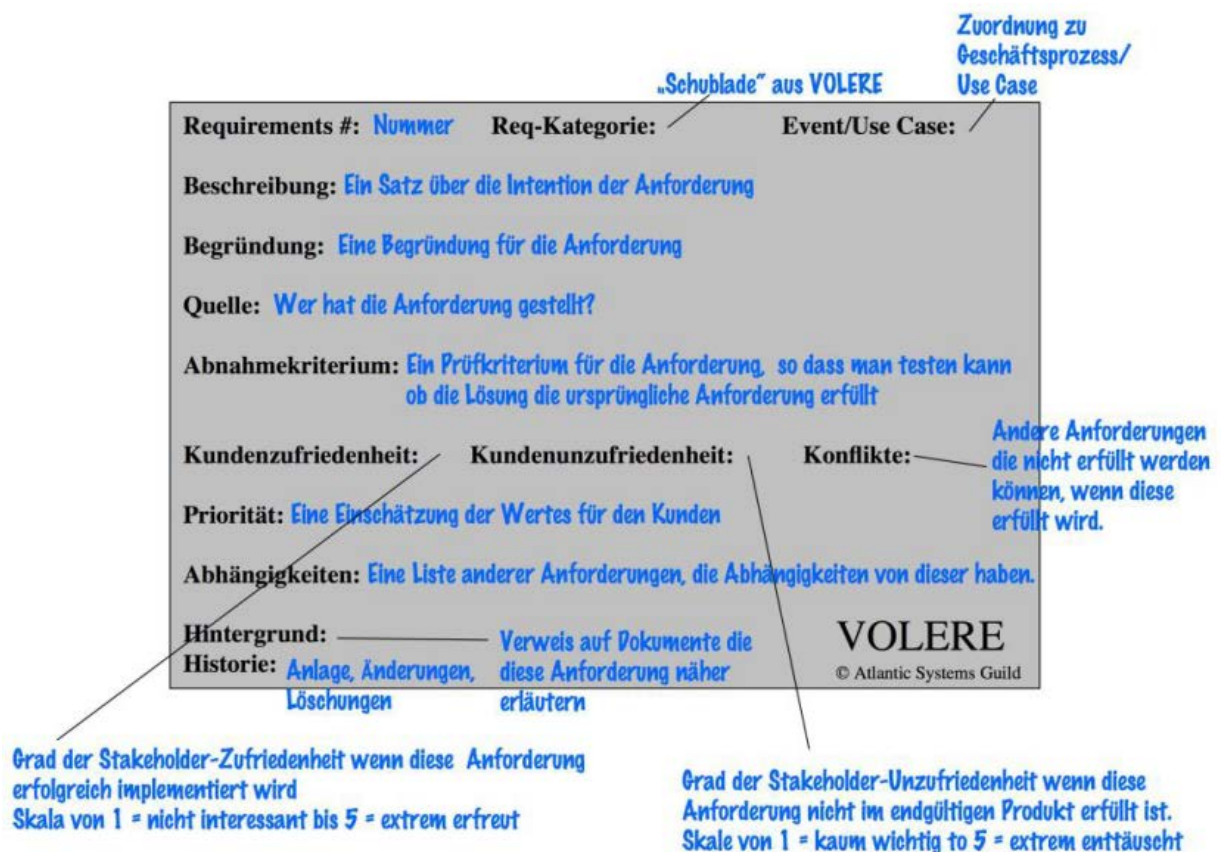


Abb. 3: Karteikartenvorlage nach VOLERE¹⁴

Nachdem die Grundlagen zu einzelnen Anforderungen und deren Arten beschrieben wurden, befasst sich das folgende Kapitel mit der Bearbeitung und dem Management von Anforderungen.

2.1.4 Lebenszyklus des Requirements Engineering

Der bereits genannte Begriff des Requirements Engineerings umfasst das Ermitteln, Formulieren, Validieren und Verwalten von Anforderungen. Die folgende Abbildung visualisiert die Phasen und deren Zusammenspiel.

¹⁴ Enthalten in: Roberson, J./Robertson, S. (2007), S. 5

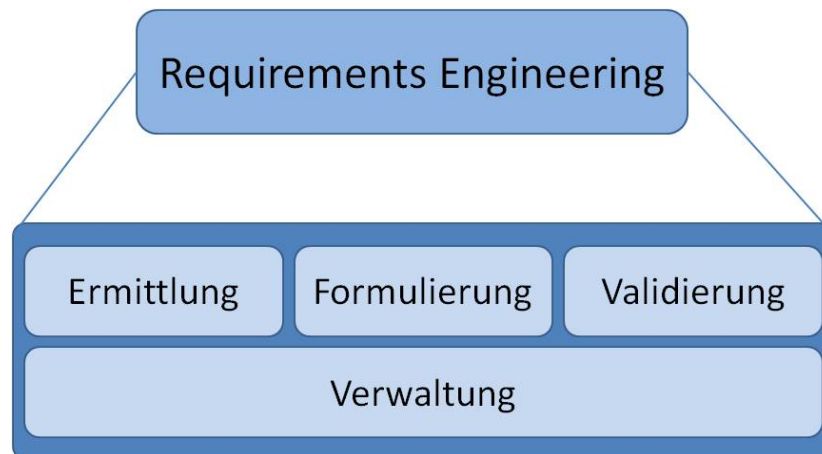


Abb. 4: Inhalte des Requirements Engineerings

Wie die Abbildung 4 zeigt, hat die Phase Verwaltung Einfluss auf die anderen drei Phasen. Beispiele hierfür sind Auswirkungen auf Anforderungen durch das Änderungs- und Umsetzungsmanagement.

Die allgemeine Auswirkung von Requirements Engineering auf den Projekterfolg ist in der folgenden Abbildung dargestellt.

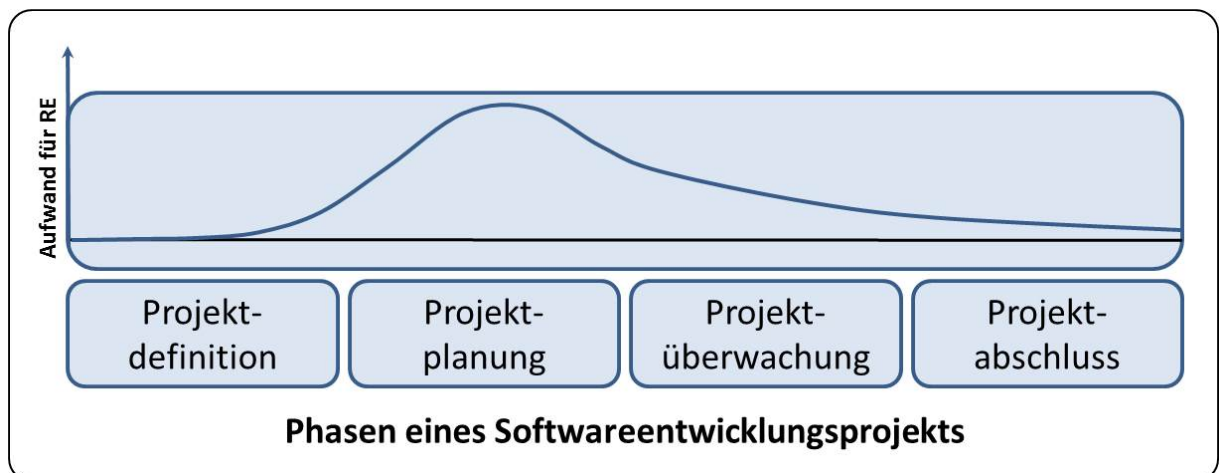


Abb. 5: Einsatz von RE im Verlauf eines Projekts

Wie die Abbildung 5 verdeutlicht, ist in der Regel am Anfang eines Projekts und dort besonders in der Phase der Projektplanung der Aufwand für Requirements Engineering sehr hoch. Das Ziel dieser Methode ist es, durch den hohen Aufwand bei der Planung eine durchgängig geringe Fehlerquote und eine hohe Qualität des Produkts während seines gesamten Lebenszyklus zu erreichen. Wird in Softwareengineeringprojekten nach dieser Methode vorgegangen, werden spät auftretende Fehler und damit verbundene, meist sehr hohe Kosten für die Behebung reduziert und im besten Falle vermieden.

Bestätigt wird diese Vorgehensweise von Balzert, der Requirements Engineering als Aufgabe der Projektplanung und auch als eine der anspruchsvollsten Aufgaben innerhalb der Softwareentwicklung sieht.¹⁵

Das allgemeine Ziel des Requirements Engineering lässt sich als die Bereitstellung einer geeigneten Beschreibungsmaßnahme beschreiben, welche methodisch exakt eingebettet ist und durch qualifizierte Werkzeuge unterstützt wird.¹⁶

2.2 Requirements Engineering im Projektumfeld

„Anforderungen sind die Grundlage jeglicher Projektarbeit.“¹⁷ Aufgrund der hohen Bedeutung von Requirements Engineering für Projekte wird die Beziehung für die unterschiedlichen Arten von Vorgehensmodellen in Projekten genauer betrachtet. Dafür wird im Folgenden ein Überblick über gängige Ausprägungen von Softwareentwicklungsprojekten gegeben und das agile Vorgehensmodell SCRUM vorgestellt.

2.2.1 Arten von Vorgehensmodellen

Fast jedes Vorgehensmodell richtet sich nach einer der folgenden zentralen Begriffe:¹⁸

- **Iterativ-inkrementell:**

Bei einem iterativ-inkrementellen Vorgehen wird der gesamte Ablauf der Softwareentwicklung in zeitliche Elemente unterteilt, sog. Iterationen. Das Produkt wird in Teilen, Inkrementen, realisiert. Die aufeinanderfolgenden Inkremente bilden dabei einen Zuwachs an Funktionalitäten. Während der Bearbeitung von Aufgaben, die sich in mehreren Iterationen wiederholen, können Erfahrungen gemacht werden und bei späteren Iterationen angewandt und in die Entwicklung mit eingebracht werden. Ein weiterer Vorteil dieser Vorgehensweise ist die Überprüfung bereits umgesetzter Funktionalitäten in einem frühen Stadium.

- **Anwendungsfallgetrieben:**

In anwendungsfallgetriebenen Vorgehensmodellen stellen Anwendungsfälle (Use Cases) die Grundlage dar, um den Ablauf der Softwareentwicklung zu strukturieren. Üblicherweise wird bei diesem Vorgehen ein Anwendungsfall einem Inkrement zugeordnet und in diesem vollständig realisiert.

- **Agil bzw. adaptiv:**

Ein agiles bzw. adaptives Vorgehensmodell intendiert eine durchgehende Berücksichtigung und Betrachtung aller Chancen und Risiken. „Diejenige Aktivität wird als Nächstes ausgeführt, welche die zum jeweiligen Zeitpunkt vorhandenen Risiken am

¹⁵ Vgl. Balzert, H. (2009), S. 434

¹⁶ Vgl. Partsch, H. (1991), S. 22

¹⁷ Fahney, R./Herrmann, A. (2013), S. 10

¹⁸ Vgl. Rupp, C. (2007), S. 48

effektivsten minimiert und die Chancen am besten nutzt.“¹⁹ In der Regel bedingt ein agiles Projekt, ein iterativ-inkrementelles Vorgehen.

Die beschriebenen Vorgehensweisen in Projekten haben ganz unterschiedliche Auswirkungen auf die Ermittlung, Formulierung, Validierung und Verwaltung von Anforderungen. Um diese Auswirkung anhand eines konkreten Vorgehensmodells beispielhaft zu erläutern, wird im folgenden SCRUM verkürzt dargestellt.

SCRUM – ein praxisrelevantes, agiles Vorgehensmodell

Die Schlantheit und Flexibilität in der Entwicklung und den Prozessen ist bei SCRUM gegeben, da die Prinzipien der agilen Softwareentwicklung praxisgerecht angewandt werden. Der kundenorientierte und iterative Prozess ist aus mehreren Rückkopplungsschleifen aufgebaut, die ineinander verschachtelt sind.²⁰ Somit ist bei SCRUM auch das iterativ-inkrementelle Vorgehen ein Bestandteil.

2.2.2 Herausforderungen bei der Anforderungsanalyse in Projekten

Die Vorgehensweise des Requirements Engineering hat direkte Auswirkungen auf den Projekterfolg. Jegliche Unzulänglichkeiten oder Mängel im Umgang mit Anforderungen können das Ergebnis negativ beeinflussen. „Fehler, die im engen Zusammenhang mit Anforderungen stehen, stellen mengenmäßig die Häufigsten dar.“²¹ Aus diesem Grund werden im Folgenden einige Hauptprobleme im Umgang mit Anforderungen betrachtet:²²

- **Unklare Zielvorstellungen:**

Meist haben die Produkte aus Softwareprojekten sehr unterschiedliche Gruppen von Anwendern mit entsprechend unterschiedlichen Erwartungen an die Produkte. Verstärkt wird dieser Effekt durch verschiedene Benutzungshäufigkeiten der einzelnen Anwendergruppen oder durch das unterschiedlich hohe Bildungs- und Erfahrungsniveau dieser. Um unklare Zielvorstellungen zu vermeiden, ist es empfehlenswert, die Anforderungen von Repräsentanten aller Stakeholdergruppen formulieren zu lassen. Diese Vorgehensweise spiegelt sich bei VOLERE in der ersten Kategorie wider.

- **Hohe Komplexität der zu lösenden Aufgabe:**

Immer komplizierter werdende Aufgabenstellungen und Prozesse in Unternehmen führen dazu, dass auch die unterstützende Software und Systeme durch wachsende Komplexität gekennzeichnet sind. Darüber hinaus bestehen zwischen den einzelnen Anforderungen wiederum komplexe Abhängigkeitsbeziehungen, die bei jeder Änderung berücksichtigt werden müssen.

¹⁹ Rupp, C. (2007), S. 13

²⁰ Vgl. Grande, M. (2011), S.111

²¹ Rupp, C. (2007), S. 24

²² Vgl. Rupp, C. (2007), S. 25 ff.

- **Änderungen von Zielen und Anforderungen:**
Oft werden Anforderungen während eines laufenden Projekts geändert. Da diese Änderungen sich sukzessive auf das gesamte Projekt und dessen Erfolg auswirken können, ist es notwendig, Folgen und Auswirkungen auf betroffene Entwicklungsergebnisse schnell aufzuzeigen und aufgrund dieser Informationen die richtigen Entscheidungen zu treffen.
- **Schlechte Qualität der Anforderungen:**
Bei der Dokumentation von Anforderungen sind Mehrdeutigkeiten, Redundanzen, Widersprüche und ungenaue Angaben bekannte Kernprobleme. Mehrdeutigkeiten bedingen unterschiedliche Interpretationen über die Bedeutung der jeweiligen Anforderung. Redundante Informationen können zu Fehlern und erheblichem Mehraufwand führen, wenn nur eine Quelle bei einer Änderung aktualisiert wurde. Widersprüche können entstehen, wenn mehrere Benutzer unterschiedliche Wünsche und Erwartungen äußern. Vermieden werden können diese bspw. durch die Anwendung von anwendungsfallbezogenen Vorgehensmodellen. Ungenaue Angaben entstehen oft dadurch, dass z.B. Grundanforderungen von Kunden gar nicht genannt, aber trotzdem erwartet werden. Andererseits werden Leistungsanforderungen an das Produkt oft nur vage ausgesprochen, wodurch die Entwickler große Entscheidungsfreiräume bei der Umsetzung haben. Die Folgen sind häufig unzufriedene und frustrierte Kunden.
- **Ungenaue Planung und Verfolgung des Projektes:**
Unkorrekt formulierte Anforderungen lassen oft nicht auf die exakte und hohe Komplexität des Themas schließen und verursachen so eine falsche Planung von Zeit, Qualität und Kosten. Um dies zu vermeiden, sollten auch individuelle Berichte und Statusreports zeitnahe erstellt und aufbereitet werden.

Die beschriebenen Hauptprobleme im Umgang mit Requirements Engineering zeugen von der hohen Komplexität. Um Projektbeteiligte hierbei effizient und möglichst gut zu unterstützen, bietet der Markt Requirements Engineering Tools an. Einer der größten Vorteile ist hierbei die Automatisierung vieler Funktionen, wie Berichterstellung, Beziehungs- und Berechtigungsmanagement oder auch Prüfungen bei der Verwaltung von Anforderungen.

2.3 Tools zur Unterstützung von Requirements Engineering

Kapitel 2.3 umfasst einen kurzen Einstieg zu den grundlegenden Lösungsarten von Requirements Engineering Tools und der Erstellung von Marktanalysen.

2.3.1 Grundlegende Lösungsarten von Requirements Engineering Tools

Wie bereits oben beschrieben, kann der Umgang mit den in Kapitel 2.2.2 aufgelisteten Hauptproblemen beim Requirements Engineering mithilfe von Tools unterstützt werden. Da Requirements Engineering von unterschiedlichen Interessensgruppen durchgeführt wird, gibt es auch eine große Auswahl an Werkzeugen, die den Anwender dabei assistieren. Um einen Überblick über verschiedene Werkzeuge zugeben, werden im Folgenden zwei unterschiedliche Arten der Einteilung vorgestellt. Dabei erfolgt die erste Einteilung nach der Abdeckung der Aufgaben im Entwicklungsprozess.²³

- **Horizontale Werkzeuge:**

Horizontale Werkzeuge sind in der Regel spezialisiert auf eine bestimmte Phase des Entwicklungsprozesses oder auf den Übergang zweier Phasen. Zu diesen Werkzeugen gehören deshalb beispielsweise Analyse-, Entwurfs-, Wartungs- und Testwerkzeuge.

- **Vertikale Werkzeuge:**

Vertikale Werkzeuge unterstützen im Gegensatz zu den horizontalen Werkzeugen entweder mehrere Phasen, oder sogar den gesamten Entwicklungsprozess. Zu diesen Werkzeugen gehören u.a. Projektmanagement-, Konfigurations-, Qualitätssicherungs- und Wiederverwendungswerkzeuge.

Werden Requirements Engineering Tools für spezielle Phasen am Anfang eines Projekts eingesetzt, so sind sie primär den horizontalen Werkzeugen zuzuordnen. Werkzeuge, mit denen auch die Anforderungen von Kunden und an Produkte gemanagt werden sollen, sind oft vertikale Werkzeuge.

Darüber hinaus lassen sich nach Schienmann die Tools des Requirements Engineering auch abhängig von ihrem Einführungsaufwand grob einteilen.²⁴

- **Große Lösung:**

Bei einer großen Lösung erfolgt die Verwaltung der Anforderungen in einem zentralen Repository, welches oft eigenimplementiert wurde. Die Sicherstellung des Datenaustausches mit anderen Werkzeugen erfolgt über offene Schnittstellen.

- **Mittlere Lösung:**

Zu den mittleren Lösungen werden häufig Standard Requirements Engineering Tools gezählt. Der Vorteil hiervon sind standardisierte Brücken- oder Schnittstellenprogramme, die eine schnelle und einfache Integration mit anderen Tools ermöglichen. Ein Teil dieser Tools bieten zusätzlich Customizing an.

²³ Vgl. Schienmann, B. (2002), S. 279 f.

²⁴ Vgl. Schienmann, B. (2002), S. 280 f.

- **Kleine Lösung:**

Kleine Lösungen entsprechen bereits vorhandenen Entwicklungswerkzeugen, die bspw. für den Modellierungsbereich oder das Konfigurationsmanagement eingesetzt werden. Sie sind somit keine speziellen Requirements Engineering Tools, sondern können auch in anderen Bereichen eingesetzt werden.

Die zwei vorgestellten Arten der Einteilung von Tools nach Schienmann werden im weiteren Verlauf der vorliegenden Arbeit zur besseren Bezeichnung verwendet.

2.3.2 Möglichkeiten zur Erstellung von Marktanalysen

In der Praxis lassen sich viele Tools finden, mit denen Aufgaben des Requirements Engineering abgedeckt werden können. Die Aufgabe von Marktanalysen ist es, einen Markt nach vorher festgelegten Kriterien zu strukturieren. Die beiden auf Informationstechnik spezialisierten Marktforschungsunternehmen Gartner und Forrester haben mit dem Gartner Magic Quadrant und der Marktforschungsmethode Forrester Wave Möglichkeiten zur Durchführung und visuellen Aufbereitung von Marktanalysen entwickelt, die in der betrieblichen Praxis auf große Akzeptanz stoßen. Im Folgenden soll besonders der Magic Quadrant von Gartner näher betrachtet werden. Aus den Ergebnissen der Betrachtung wird in Kapitel 3 anschließend eine Vorgehensweise zur Erstellung einer flexiblen Marktübersicht für das Segment der Requirements Management Tools vorgestellt, die im Rahmen dieser Forschungsarbeit entwickelt wurde.

Der Gartner Magic Quadrant teilt Anbieter von Software gemäß ihrer Marktposition in die vier Quadranten „Challengers“, „Leaders“, „Niche Players“ und „Visionaries“ ein. Die Marktposition eines Anbieters wird dabei durch den Erfüllungsgrad der Kriterien „Ability to Execute“ und „Completeness of Vision“ bestimmt, die jeweils aus verschiedenen Komponenten zusammengesetzt sind. Ausschlaggebend für einen hohen Erfüllungsgrad des Faktors „Ability to Execute“ sind Kriterien, wie Marketing, Support, finanzielle Situation des Anbieters, Effektivität der Vertriebskanäle und die Qualität der angebotenen Produkte. Unter der „Completeness of Vision“ dagegen werden Kriterien, wie Innovations-, Marketing-, Verkaufsstrategie, und das Geschäftsmodell des Anbieters zusammengefasst.^{25 26}

²⁵ Vgl. Black, D. / Thomas, J. (2013)

²⁶ Vgl. Gartner (o.J.)

Abbildung 6 fasst den grundlegenden Aufbau von Gartners Magic Quadrant zusammen.

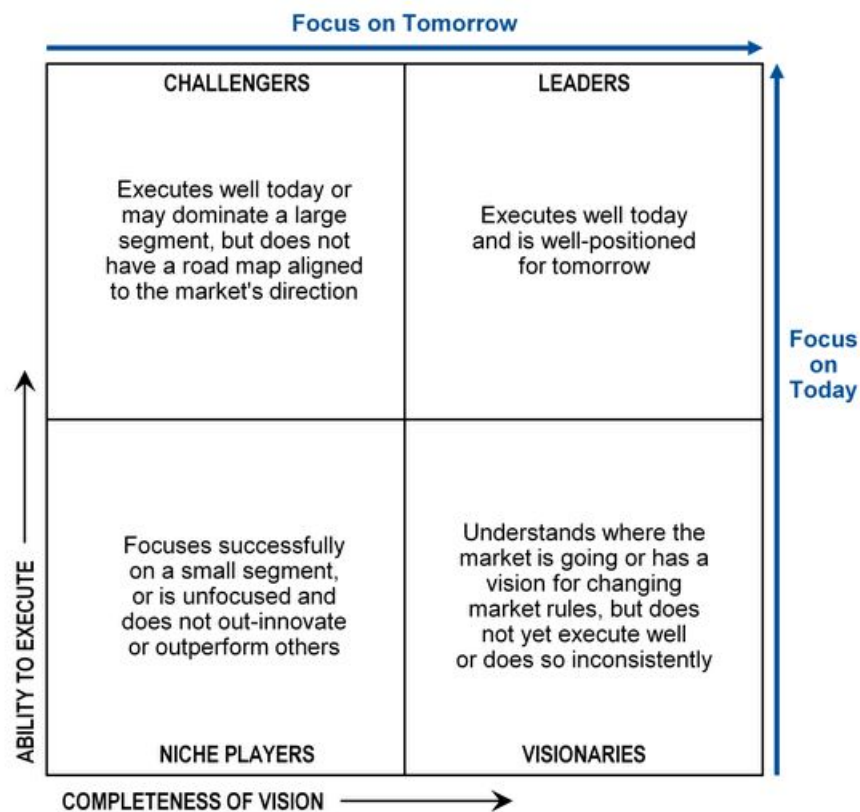


Abb. 6: Aufbau des Gartner Magic Quadrant²⁷

„Challengers“ erfüllen die Anforderungen des Faktors „Ability to Execute“ in einem hohen Maß, d.h. sie haben eine starke Position auf dem Markt. Jedoch fehlt ihnen eine auf die Zukunft ausgerichtete Vision (geringe „Completeness of Vision“). In diesem Segment finden sich beispielsweise finanzkräftige Anbieter mit einem reifen Produkt. Um vom „Challenger“ zum „Leader“ zu werden, sollten Anbieter ihre Strategie weiter entwickeln.²⁸

„Leaders“ sind dadurch gekennzeichnet, dass sie sowohl die „Ability to Execute“, als auch die „Completeness of Vision“ in einem hohen Maß erfüllen. In diesem Segment befinden sich Marktführer, die in den Erhalt ihrer Marktführerschaft investieren. Im Allgemeinen decken „Leader“ die Anforderungen vieler Kundengruppen ab. Für spezielle Einsatzzwecke können diese Anbieter in manchen Fällen dennoch nicht geeignet sein.²⁹

„Niche Players“ können erfolgreich in einem spezialisierten Marktsegment sein. Dennoch erfüllen sie sowohl die „Ability to Execute“, als auch die „Completeness of Vision“ nur in einem geringen Maß. In diesem Bereich sind beispielsweise Anbieter zu finden, die zwar ein funktional ausgewogenes Produkt anbieten, jedoch beispielsweise keinen ausreichenden Support gewährleisten können. Auch Anbieter, die gerade erst in den Markt eingetreten sind,

²⁷ Enthalten in: Black, D. / Thomas, J. (2013)

²⁸ Vgl. Black, D. / Thomas, J. (2013)

²⁹ Vgl. Black, D. / Thomas, J. (2013)

werden häufig zunächst als „Niche Players“ eingestuft. Die Auswahl eines „Niche Players“ als Tool-Anbieter ist für ein Unternehmen mit einem erhöhten Risiko verbunden.³⁰

„Visionaries“ haben eine auf die Zukunft ausgerichtete Vision, sind aber im Moment nicht in der Lage nachhaltig zu wirtschaften (geringe „Ability to Execute“). In vielen Fällen sind „Visionaries“ verantwortlich für Innovationen. Falls ein Anbieter die Innovation auf dem Markt durchsetzen kann, wird er sich zum „Leader“ entwickeln.³¹

Gartner weist darauf hin, dass die Einteilung der Anbieter einem Unternehmen lediglich einen Anhaltspunkt für Investitions- und Kaufentscheidungen bietet, da die spezifischen Anforderungen eines Unternehmens ausschlaggebend sein sollten.

Auch bei der Marktforschungsmethode Forrester Wave werden die Anbieter von Software anhand vorher festgelegter Kriterien bewertet und die Ergebnisse anschließend in Matrixform dargestellt. Im Gegensatz zu Gartners Magic Quadrant werden die Anbieter nicht in Quadranten eingeordnet, sondern in die Bereiche „Risky Bets“, „Contenders“, „Strong Performers“ und „Leaders“.³²

Für die Erstellung der Marktanalyse in dieser Projektarbeit wurde eine Marktforschungsmethode speziell für das Segment der Requirements Management Tools entwickelt. Von den Marktanalysemethoden von Gartner und Forrester wird dabei lediglich die Idee der Aufbereitung der Analyseergebnisse in Matrixform übernommen.

³⁰ Vgl. Black, D. / Thomas, J. (2013)

³¹ Vgl. Black, D. / Thomas, J. (2013)

³² Vgl. Forrester (o.J.)

3 Vorgehensweise zur Erstellung einer unternehmensspezifischen Marktübersicht

Im Rahmen der vorliegenden Arbeit wurde eine Methode zur Marktanalyse entwickelt, die die aus den oben vorgestellten Standards gewonnenen Erkenntnisse berücksichtigt, im Kern jedoch von diesen Standards abweicht. Das Ziel der entwickelten Methode die Erstellung einer flexiblen, vom Unternehmen individuell anpassbaren Gestaltung der Marktübersicht für das Segment der Anforderungsmanagement-Tools.

Die angewandte Methodik basiert auf einem Kriterienkatalog, der Anforderungen an ein Anforderungsmanagement-Tool enthält. Diese Anforderungen werden in Kategorien eingeteilt und erhalten eine individuelle Gewichtung. Jedes Tool wurde anschließend bezüglich der Erfüllung der Kriterien überprüft und die erhaltenen Ergebnisse grafisch aufbereitet.

In Kapitel 3 werden die einzelnen Komponenten der im Projekt entwickelten Marktanalyse-methode vorgestellt. Ziel dieses Kapitels ist es, die Methodik auf einer granularen Ebene zu erläutern. Durch diese Transparenz kann ein Unternehmen die variablen Bestandteile der Methode jederzeit eigenständig anpassen und erweitern.

3.1 Erläuterung der verwendeten Kriterienkategorien

Wie bereits in Kapitel 2.1.2 dargelegt, kommt den Anforderungsarten „Funktionale Anforderungen“ und „Nichtfunktionale Anforderungen“ in der Literatur eine besondere Rolle zu. Im Rahmen der Erstellung der Marktübersicht für Anforderungsmanagement-Tools werden alle Kriterienkategorien jeweils diesen beiden Anforderungsarten zugeordnet.

Bei der Kategorisierung der Kriterien selbst wurde auf eine Anlehnung an einen Standard verzichtet, da besonders bei den funktionalen Anforderungen die Kategorien nach spezifischen Gesichtspunkten der Anforderungsmanagement-Tools gebildet wurden. Der Fokus der vorliegenden Marktanalyse wurde mit dem Ziel der Erstellung einer branchen- und unternehmensunabhängigen Marktübersicht auf die funktionalen Anforderungen gelegt, da nicht-funktionale Anforderungen stark von dem jeweiligen Einsatzgebiet des Tools im Unternehmen abhängen.

Funktionale Anforderungen
Basisfunktionalitäten
Erweiterung der Basisfunktionalitäten
Darstellung und Aufbereitung von Anforderungen
Unterstützung der Teamarbeit
Schnittstellenfunktionalitäten

Tab. 2: Kategorien der funktionalen Kriterien

Nichtfunktionale Anforderungen
Bedienbarkeit
Sicherheit
Herstellereigenschaften
(Open Source)

Tab. 3: Kategorien der nichtfunktionalen Kriterien

Die Tabellen 2 und 3 geben einen Überblick über die gebildeten Kriterienkategorien, welche im Folgenden erläutert werden:

- **Basisfunktionalitäten:** Die Kategorie Basisfunktionalitäten enthält alle Funktionalitäten, die ein Anforderungsmanagement-Tool bereitstellen soll, um grundlegende Aufgaben des Requirements Engineering unterstützen zu können. Alle Kriterien in dieser Kategorie sind aus diesem Grund Pflichtkriterien. Falls ein Tool ein Pflichtkriterium nicht erfüllt, wird es im Rahmen dieser Marktübersicht nicht weiter betrachtet.
- **Erweiterung der Basisfunktionalitäten:** Unter Erweiterung der Basisfunktionalitäten werden alle Anforderungen zusammengefasst, die die Definition einer Anforderung im Tool funktional ergänzen.
- **Darstellung und Aufbereitung von Anforderungen:** Alle Funktionalitäten, die die Anzeige der Anforderungen strukturieren und die Ableitung von Entscheidungen erleichtern, finden sich in der Kategorie Darstellung und Aufbereitung von Anforderungen.
- **Unterstützung der Teamarbeit:** An dem Requirements Engineering Prozess in einem Unternehmen im Regelfall unterschiedliche Personen beteiligt. In der Kategorie Unterstützung der Teamarbeit werden alle Funktionalitäten zusammengefasst, die die Projektarbeit erleichtern. Da der Schwerpunkt der vorliegenden Forschungsarbeit auf das Requirements Engineering in Projekten gelegt wurde, kommt dieser Kategorie eine besondere Bedeutung zu. Dies ist an der Anzahl der enthaltenen Kriterien zu erkennen.
- **Schnittstellenfunktionalitäten:** Das Anforderungsmanagement-Tool soll in die vorhandene Infrastruktur eines Unternehmens integriert werden. Die Kategorie Schnittstellenfunktionalitäten enthält Anforderungen, die eine Anbindung an andere Tools sicherstellen.

Diese Kategorien im Bereich der funktionalen Anforderungen werden um weitere Kategorien im Bereich der nichtfunktionalen Anforderungen ergänzt.

- **Bedienbarkeit:** Funktionen von Anforderungen, die den Nutzer beim Umgang mit dem Tool unterstützen, werden in der Kategorie Bedienbarkeit zusammengefasst.
- **Sicherheit:** Die Kategorie Sicherheit beschreibt die Kriterien, anhand derer bewertet wird, wie im Tool verwaltete Daten sowohl unter Informationssicherheitsaspekten, als auch unter dem Aspekt der Wiederherstellbarkeit, gesichert werden.

- **Herstellereigenschaften:** Die Kategorie Herstellereigenschaften enthält die Kriterien, die Anforderungen an den Hersteller eines Tools beinhalten.
- **Open Source:** Die Kategorie Open Source fasst spezifische Anforderungen zur Bewertung des Reifegrades eines Open Source Tools zusammen. Diese Kategorie wurde nicht in den Kriterienkatalog aufgenommen, um die Vergleichbarkeit zwischen allen betrachteten Tools zu erhalten, da nur Anforderungen geprüft werden, die alle Tools aufweisen. Mithilfe dieser Kriterien wurden jedoch die Open Source Tools ausgewählt, die im Rahmen der Marktanalyse mithilfe des Kriterienkatalogs bewertet wurden (siehe Kapitel 4.1).

3.2 Methodik zur Beschreibung eines Kriteriums

Die Beschreibung der Kriterien erfolgt mithilfe einer Karteikarte. Dies stellt sicher, dass die Kriterien einheitlich und strukturiert beschrieben werden. Zudem werden durch die Verwendung der Karteikarte wichtige Merkmale einer Anforderung, wie sie in Kapitel 2.1.1 beschrieben sind, berücksichtigt. Hier ist besonders der Einfluss der Karteikarte auf die Vollständigkeit, Einheitlichkeit und den Umfang einer Anforderungsdefinition zu nennen. Die entwickelte Karteikarte ist an die Methodik der in Kapitel 2.1.3 beschriebenen Karteikartenvorlage des Standards VOLERE angelehnt. Im Aufbau unterscheidet sie sich jedoch von der VOLERE Karteikarte. Tabelle 4 enthält die modifizierte, im Rahmen der Projektarbeit weiterentwickelte Karteikarte zur Beschreibung einer Anforderung.

Anforderungskategorie	
Anforderungsname	Gewichtung
Beschreibung	Erläuterungen der Anforderung
Begründung	Auswirkungen auf den Kunden? Welchen Profit erhält die Organisation durch Erfüllung dieses Kriteriums?
Abnahmekriterien	Wann ist diese Anforderung konkret erfüllt? Angabe des SOLL-Zustands und Beurteilung durch fehlende, teilweise und volle Erfüllung des Kriteriums

Tab. 4: Allgemeiner Aufbau der Karteikarte zur Dokumentation einer Anforderung

Neben der Anforderungskategorie enthält die Karteikarte auch den Namen der Anforderung. Zudem bietet sie die Möglichkeit der individuellen Gewichtung jedes Kriteriums, wodurch eine Priorisierung vorgenommen werden kann. Ein Unternehmen kann diese Gewichtung seinem individuellen Bedarf anpassen. Die Beschreibung der Anforderung erläutert die zu prüfende Funktion. Die jeweilige Begründung lässt auf die mögliche Gewichtung des Kriteriums schließen und verdeutlicht, welche Folgen ein Unternehmen bei Nichterfüllung riskiert. Die Struktur der Karteikarte erlaubt durch die Angabe von ein oder mehreren Abnahmekriterien eine granulare Beschreibung des Erfüllungsgrads jeder einzelnen Anforderung. Da ein Abnahmekriterium nur genau eine Funktion auf detaillierter Ebene abfragt, kann eindeutig und schnell zugeordnet werden, ob ein Tool das Abnahmekriterium erfüllt.

Enthält eine Anforderung mehrere Abnahmekriterien, wird eine Entscheidungsregel hinterlegt, die entsprechend der Kombination erfüllter Abnahmekriterien dem Tool einen der Erfüllungsgrade nicht, teilweise oder voll erfüllt für die jeweilige Anforderung zuweist. Aufbauend auf dieser Logik können essenzielle Abnahmekriterien als Pflichtabnahmekriterien definiert werden. Das heißt, sie müssen zwingend erfüllt sein, um die Anforderung teilweise oder voll zu erfüllen. **Entscheidungsregeln für Pflichtabnahmekriterien** werden nach der folgenden Regel aufgebaut:

- Nicht alle Pflichtabnahmekriterien erfüllt
→ Anforderung nicht erfüllt
- Genau alle Pflichtabnahmekriterien erfüllt
→ Anforderung teilweise erfüllt
- Genau alle Pflichtabnahmekriterien und x weitere Abnahmekriterien erfüllt
→ Anforderung voll erfüllt

Der erste Fall tritt auch ein, wenn nur Abnahmekriterien und nicht alle Pflichtabnahmekriterien erfüllt sind.

3.3 Erläuterung der verwendeten Kriterien

Kapitel 3.3 stellt den in der Projektarbeit entwickelten Kriterienkatalog vor. Nach der Beschreibung des Aufbaus folgt die detaillierte Darstellung der funktionalen und nichtfunktionalen Anforderungen.

3.3.1 Aufbau des Kriterienkatalogs

Im Rahmen des Projekts wurde der Kriterienkatalog so aufgebaut, dass trotz der Komplexität der Anforderungsbewertung eine übersichtliche Ergebnisdarstellung gegeben werden kann. Dies erfolgt durch eine Aufteilung in die drei Ebenen Kriterienkatalog, Anforderung und Abnahme- und Pflichtabnahmekriterien, die in der Abbildung 7 dargestellt wird.

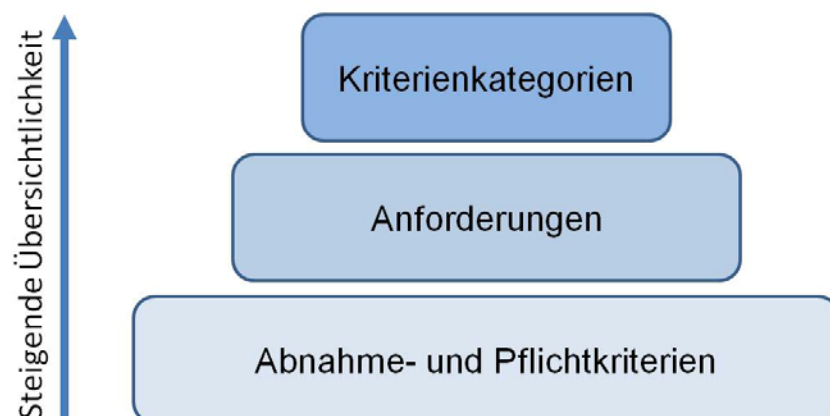


Abb. 7: Ebenen des Kriterienkatalogs

Wie die Abbildung 7 verdeutlicht, besitzen die Kriterienkategorien das höchste Abstraktionsniveau. Der Bewertung einer der acht Kategorien gibt einen Überblick über die aggregierten Erfüllungsgrade der in der Kategorie enthaltenen Anforderungen. Die Komplexität, eine Anforderung zu beurteilen wird somit durch die Bearbeitung auf der granularen Ebene der Abnahmekriterien gemindert.

Durch die Transparenz bei der Bestimmung des Erfüllungsgrades einer Anforderung, kann ein Unternehmen ohne großen Aufwand die vorhandenen Abnahmekriterien um eigene Kriterien ergänzen, bzw. spezifische Priorisierungen durch Veränderung der Pflichtabnahmekriterien vornehmen. Ein weiterer Vorteil des entwickelten Kriterienkatalogs besteht darin, dass dieser auf alle Anforderungsmanagement-Tools angewendet werden kann, unabhängig davon, ob es sich bei dem Tool um eine Speziallösung für das Requirements Engineering oder eine Application Lifecycle Development Suite (umfassendes Tool zur Verwaltung des gesamten Lebenszyklus einer Software) mit einer Requirements Engineering Komponente handelt. Durch diese Universalität entsteht eine Vergleichbarkeit zwischen den verschiedenen Toolkategorien.

Im Folgenden werden die Anforderungen erklärt, die in den Kriterienkatalog aufgenommen wurden. Dabei wird die in Kapitel 3.2 dargestellte Karteikarte verwendet. Falls zu einer Anforderung hinsichtlich der Softwareentwicklung nach dem Vorgehensmodell SCRUM (siehe Kapitel 2.2.1) Besonderheiten zu berücksichtigen sind, wird darauf in einer Anmerkung unter der Karteikarte hingewiesen. Ein Unternehmen, das SCRUM anwendet, kann diese Hinweise für die Modifizierung der Gewichtung einer Anforderung verwenden.

3.3.2 Funktionale Anforderungen

Basisfunktionalitäten

Funktionale Anforderung: Basisfunktionalitäten							
Basisdefinition einer Anforderung							
Gewichtung: x-fach							
Beschreibung	Eine neue Anforderung kann im System in Textform erfasst werden. Das Tool unterstützt dabei die Abfrage von Standard-Attributen, wie beispielsweise Anforderungsbezeichnung, Bearbeiter, Erfasser, usw.						
Begründung	Die textuelle Erfassung einer Anforderung ist eine Basisfunktion des Requirements Engineering.						
Abnahmekriterien	<p>1. Abfrage von Standardattributen bei der Definition einer Anforderung</p> <table border="1" style="width: 100%;"> <tr> <td style="color: red;">nicht erfüllt:</td> <td>0 erfüllt</td> </tr> <tr> <td style="color: blue;">teilweise erfüllt:</td> <td>-</td> </tr> <tr> <td style="color: green;">voll erfüllt:</td> <td>1 erfüllt</td> </tr> </table>	nicht erfüllt:	0 erfüllt	teilweise erfüllt:	-	voll erfüllt:	1 erfüllt
nicht erfüllt:	0 erfüllt						
teilweise erfüllt:	-						
voll erfüllt:	1 erfüllt						

Tab. 5: Basisdefinition einer Anforderung

Funktionale Anforderung: Basisfunktionalitäten							
Änderung einer Anforderung	Gewichtung: x-fach						
Beschreibung	Eine bereits definierte Anforderung kann nachträglich verändert werden.						
Begründung	Die Änderung einer Anforderung ist eine Basisfunktion des Requirements Engineering. Die Änderung von Anforderungen erhöht die Flexibilität im Anforderungsmanagementprozess.						
Abnahmekriterien	1. Änderungsfunktion						
	<table border="1"> <tr> <td>nicht erfüllt:</td> <td>0 erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>-</td> </tr> <tr> <td>voll erfüllt:</td> <td>1 erfüllt</td> </tr> </table>	nicht erfüllt:	0 erfüllt	teilweise erfüllt:	-	voll erfüllt:	1 erfüllt
nicht erfüllt:	0 erfüllt						
teilweise erfüllt:	-						
voll erfüllt:	1 erfüllt						

Tab. 6: Änderung einer Anforderung

Funktionale Anforderung: Basisfunktionalitäten							
Kopieren einer Anforderung	Gewichtung: x-fach						
Beschreibung	Eine bereits definierte Anforderung kann kopiert werden.						
Begründung	Das Kopieren einer Anforderung ist eine Basisfunktion des Requirements Engineering. Diese Funktion reduziert den Aufwand der manuellen Erfassung.						
Abnahmekriterien	1. Kopierfunktion						
	<table border="1"> <tr> <td>nicht erfüllt:</td> <td>0 erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>-</td> </tr> <tr> <td>voll erfüllt:</td> <td>1 erfüllt</td> </tr> </table>	nicht erfüllt:	0 erfüllt	teilweise erfüllt:	-	voll erfüllt:	1 erfüllt
nicht erfüllt:	0 erfüllt						
teilweise erfüllt:	-						
voll erfüllt:	1 erfüllt						

Tab. 7: Kopieren einer Anforderung

Funktionale Anforderung: Basisfunktionalitäten							
Löschen einer Anforderung	Gewichtung: x-fach						
Beschreibung	Eine bereits definierte Anforderung kann gelöscht werden.						
Begründung	Das Löschen einer Anforderung ist eine Basisfunktion des Requirements Engineering.						
Abnahmekriterien	1. Löschfunktion						
	<table border="1"> <tr> <td>nicht erfüllt:</td> <td>0 erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>-</td> </tr> <tr> <td>voll erfüllt:</td> <td>1 erfüllt</td> </tr> </table>	nicht erfüllt:	0 erfüllt	teilweise erfüllt:	-	voll erfüllt:	1 erfüllt
nicht erfüllt:	0 erfüllt						
teilweise erfüllt:	-						
voll erfüllt:	1 erfüllt						

Tab. 8: Löschen einer Anforderung

Erweiterung der Basisfunktionalitäten

Funktionale Anforderung: Erweiterung der Basisfunktionalitäten							
Visuelle Detaillierung einer Anforderung	Gewichtung: x-fach						
Beschreibung	Das Tool bietet Möglichkeiten, um eine Anforderung bzw. die damit zusammenhängenden Prozesse und Architekturen, grafisch zu modellieren.						
Begründung	Durch eine grafische Modellierung wird unter anderem die Dokumentation verbessert und eine einheitliche Basis für die weitere Kommunikation geschaffen.						
Abnahmekriterien	<ol style="list-style-type: none"> 1. Es ist ein grafischer Editor vorhanden, in dem Prozesse und Architekturen modelliert werden können 2. Die entstandenen Modelle können mit mindestens einer textuellen Beschreibung einer Anforderung verknüpft werden <table border="1" style="width: 100%;"> <tr> <td>nicht erfüllt:</td> <td>< voll erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>-</td> </tr> <tr> <td>voll erfüllt:</td> <td>alle erfüllt</td> </tr> </table>	nicht erfüllt:	< voll erfüllt	teilweise erfüllt:	-	voll erfüllt:	alle erfüllt
nicht erfüllt:	< voll erfüllt						
teilweise erfüllt:	-						
voll erfüllt:	alle erfüllt						

Tab. 9: Visuelle Detaillierung einer Anforderung

Funktionale Anforderung: Erweiterung der Basisfunktionalitäten							
Anpassbarkeit der Anforderungsattribute	Gewichtung: x-fach						
Beschreibung	Die verwalteten Attribute einer Anforderung sind an die individuellen Bedürfnisse in einem bestimmten Einsatzbereich anpassbar. Dabei kann festgelegt werden, welche Attribute bei der Einführung des Tools in einem Unternehmen bei einer Anforderungsdefinition standardisiert abgefragt werden. Bei der Festlegung dieser Attribute können Ausprägungen wie, Attributtyp, obligatorische oder optionale Befüllung, vordefinierte Werteliste und Validierungsregeln definiert werden.						
Begründung	Durch die Anpassbarkeit der Anforderungsattribute können die individuellen Spezifika eines Unternehmens berücksichtigt werden. Ein Tool, das dieses Kriterium erfüllt, kann besser in den einsatzspezifischen Anforderungsmanagementprozess eines Unternehmens integriert werden.						
Abnahmekriterien	<ol style="list-style-type: none"> 1. Definition eines individuellen Attributs 2. Attributtyp festlegbar 3. Obligatorische / optionale Befüllung 4. Vordefinierte Werteliste 5. Regeln zur Validierung definierbar <table border="1" style="width: 100%;"> <tr> <td>nicht erfüllt:</td> <td>< teilweise erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>erstes und zweites Kriterium erfüllt</td> </tr> <tr> <td>voll erfüllt:</td> <td>> teilweise erfüllt</td> </tr> </table>	nicht erfüllt:	< teilweise erfüllt	teilweise erfüllt:	erstes und zweites Kriterium erfüllt	voll erfüllt:	> teilweise erfüllt
nicht erfüllt:	< teilweise erfüllt						
teilweise erfüllt:	erstes und zweites Kriterium erfüllt						
voll erfüllt:	> teilweise erfüllt						

Tab. 10: Anpassbarkeit der Anforderungsattribute

Funktionale Anforderung: Erweiterung der Basisfunktionalitäten							
Aufbereitung der Anforderungsdefinition	Gewichtung: x-fach						
Beschreibung	Aufnahme von formatierten Texten (z.B. Tabellen) und Anhängen; zusätzlich wird eine Rechtschreibprüfung angeboten.						
Begründung	Übersichtlichkeit bei ausführlichen Anforderungsdefinitionen, Verständlichkeit durch Zusatzinformationen; Erhöhung der Qualität der Anforderungsdefinitionen.						
Abnahmekriterien	<ol style="list-style-type: none"> 1. Aufnahme von formatierten Texten 2. Aufnahme von Anhängen 3. Rechtschreibprüfung <table border="1"> <tr> <td>nicht erfüllt:</td> <td>0 erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>1-2 erfüllt</td> </tr> <tr> <td>voll erfüllt:</td> <td>3 erfüllt</td> </tr> </table>	nicht erfüllt:	0 erfüllt	teilweise erfüllt:	1-2 erfüllt	voll erfüllt:	3 erfüllt
nicht erfüllt:	0 erfüllt						
teilweise erfüllt:	1-2 erfüllt						
voll erfüllt:	3 erfüllt						

Tab. 11: Aufbereitung der Anforderungsdefinition

Darstellung und Aufbereitung von Anforderungen

Funktionale Anforderung: Darstellung und Aufbereitung von Anforderungen							
Sortieren und Gruppieren von Anforderungen	Gewichtung: x-fach						
Beschreibung	Anforderungen können zu Kategorien zusammengefasst werden. Auf diese Weise kann eine Struktur für die Verwaltung der Anforderungen aufgebaut werden.						
Begründung	Für die effiziente Verwaltung einer großen Menge an Anforderungen ist die Abbildung von Kategorien und Strukturen wichtig. Durch eine Kategorisierung können zwischen den Anforderungen Zusammenhänge nachvollziehbar dargestellt werden. Die Übersichtlichkeit für die Anwender wird erhöht.						
Abnahmekriterien	<ol style="list-style-type: none"> 1. Das Tool ermöglicht das Kategorisieren und Ordnen von Anforderungen; die gebildeten Kategorien werden in einer Ordnerstruktur übersichtlich dargestellt <table border="1"> <tr> <td>nicht erfüllt:</td> <td>0 erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>-</td> </tr> <tr> <td>voll erfüllt:</td> <td>1 erfüllt</td> </tr> </table>	nicht erfüllt:	0 erfüllt	teilweise erfüllt:	-	voll erfüllt:	1 erfüllt
nicht erfüllt:	0 erfüllt						
teilweise erfüllt:	-						
voll erfüllt:	1 erfüllt						

Tab. 12: Sortieren und Gruppieren von Anforderungen

Funktionale Anforderung: Darstellung und Aufbereitung von Anforderungen							
Filtern von Anforderungen	Gewichtung: x-fach						
Beschreibung	Ein Anwender kann durch Eingabe eines Filters die Anzeige der Anforderungen auf diejenigen Anforderungen einschränken, die alle Filterkriterien erfüllen. Es handelt sich dabei nicht um einen statischen Report, sondern um eine dynamische Darstellung der Anforderungsübersicht.						
Begründung	Durch das Verwenden eines Filters wird die Übersichtlichkeit für den Anwender erhöht, da er sich nur aktuell relevante Anforderungen anzeigen lassen kann und so die Geschwindigkeit bei routinierten Tätigkeiten erhöht wird. Auf diese Weise wird dabei die Produktivität des Anwenders gesteigert, da keine Aufwände für manuelle Sortier- und Filter-Tätigkeiten anfallen.						
Abnahmekriterien	<ol style="list-style-type: none"> 1. Es besteht eine Möglichkeit zur Erstellung eines Filters 2. Die Anzeige der Anforderungsübersicht ändert sich dynamisch je nach eingestellten Filterkriterien <table border="1" data-bbox="544 813 1461 920"> <tr> <td>nicht erfüllt:</td> <td>< voll erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>-</td> </tr> <tr> <td>voll erfüllt:</td> <td>Erstes und zweites Kriterium erfüllt</td> </tr> </table>	nicht erfüllt:	< voll erfüllt	teilweise erfüllt:	-	voll erfüllt:	Erstes und zweites Kriterium erfüllt
nicht erfüllt:	< voll erfüllt						
teilweise erfüllt:	-						
voll erfüllt:	Erstes und zweites Kriterium erfüllt						

Tab. 13: Filtern von Anforderungen

Funktionale Anforderung: Darstellung und Aufbereitung von Anforderungen							
Suchfunktion	Gewichtung: x-fach						
Beschreibung	Das Tool unterstützt den Anwender bei der Suche nach bestimmten Anforderungsinhalten, indem nach Begriffen und Kategorien gesucht werden kann.						
Begründung	Manueller Suchaufwand wird reduziert; Produktivität und Bearbeitungsgeschwindigkeit werden gesteigert.						
Abnahmekriterien	<ol style="list-style-type: none"> 1. Das Tool bietet eine Suchfunktion <table border="1" data-bbox="544 1350 1461 1458"> <tr> <td>nicht erfüllt:</td> <td>0 erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>-</td> </tr> <tr> <td>voll erfüllt:</td> <td>1 erfüllt</td> </tr> </table>	nicht erfüllt:	0 erfüllt	teilweise erfüllt:	-	voll erfüllt:	1 erfüllt
nicht erfüllt:	0 erfüllt						
teilweise erfüllt:	-						
voll erfüllt:	1 erfüllt						

Tab. 14: Suchfunktion

Funktionale Anforderung: Darstellung und Aufbereitung von Anforderungen							
Erstellung von Reports							
Gewichtung: x-fach							
Beschreibung	Mithilfe der Erstellung von Reports können Auswertungen von abgeschlossenen und bestehenden Anforderungen durchgeführt werden. Aufgrund dieser Informationen können Verbesserungsmaßnahmen für zukünftige Anforderungen abgeleitet werden, aber auch die Umsetzung bestehender Anforderungen optimiert werden. Eine Exportfunktion ermöglicht die Verwendung der Reports auch außerhalb des Tools. Eine automatische grafische Aufbereitung der Analyseergebnisse erhöht die Verständlichkeit der Reports.						
Begründung	Das Erstellen von Reports ist besonders bei der effizienten Verwaltung von Anforderungen von großer Bedeutung. Aus den Auswertungen müssen Verantwortliche die richtigen Schlüsse für die Steuerung der Anforderungsumsetzung ziehen können.						
Abnahmekriterien	<ol style="list-style-type: none"> 1. Das Tool ermöglicht das Erstellen von unterschiedlichen Reports und Analysen 2. Die Reports können auf die individuellen Bedürfnisse des Erstellers angepasst werden 3. Grafische Aufbereitung des Reports 4. Ein erstellter Report kann exportiert werden, sodass er auch außerhalb des Tools verwendet werden kann <table border="1" style="width: 100%; margin-top: 10px;"> <tr> <td>nicht erfüllt:</td> <td>< teilweise erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>Erstes Kriterium erfüllt</td> </tr> <tr> <td>voll erfüllt:</td> <td>> teilweise erfüllt</td> </tr> </table>	nicht erfüllt:	< teilweise erfüllt	teilweise erfüllt:	Erstes Kriterium erfüllt	voll erfüllt:	> teilweise erfüllt
nicht erfüllt:	< teilweise erfüllt						
teilweise erfüllt:	Erstes Kriterium erfüllt						
voll erfüllt:	> teilweise erfüllt						

Tab. 15: Erstellung von Reports

Unterstützung der Teamarbeit

Funktionale Anforderung: Unterstützung der Teamarbeit							
Verteilung von Anforderungen							
Gewichtung: x-fach							
Beschreibung	Mitarbeiter, die Schreibzugriff auf Anforderungen besitzen, können diesen – unabhängig vom Ersteller der jeweiligen Anforderung – Namen von Bearbeitern und Ansprechpartnern zuordnen. Aktualisierungen und Änderungen in diesem Bereich lösen bei den zuständigen Mitarbeitern eine Benachrichtigung aus.						
Begründung	Berechtigte Personen können Änderungen hinsichtlich der Verteilung vornehmen, wodurch die gesamte Umsetzung der Anforderungen organisiert werden kann. Zudem können außerplanmäßige Änderungen in der Projektarbeit so berücksichtigt werden (z.B. Krankheit, Urlaub).						
Abnahmekriterien	<ol style="list-style-type: none"> 1. Berechtigte Mitarbeiter können Bearbeiter und Ansprechpartner einer Anforderung festlegen und ändern 2. Benachrichtigungen erfolgen bei Änderung <table border="1" style="width: 100%; margin-top: 10px;"> <tr> <td>nicht erfüllt:</td> <td>0 erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>Erstes Kriterium erfüllt</td> </tr> <tr> <td>voll erfüllt:</td> <td>2 erfüllt</td> </tr> </table>	nicht erfüllt:	0 erfüllt	teilweise erfüllt:	Erstes Kriterium erfüllt	voll erfüllt:	2 erfüllt
nicht erfüllt:	0 erfüllt						
teilweise erfüllt:	Erstes Kriterium erfüllt						
voll erfüllt:	2 erfüllt						

Tab. 16: Verteilung von Anforderungen

Die in Tabelle 16 dargestellte Anforderung zur Verteilung von Anforderungen sollte bei der Anwendung von SCRUM besonders berücksichtigt werden, da es möglich sein soll, den Bearbeiter einer Task flexibel festzulegen und zu ändern.

Funktionale Anforderung: Unterstützung der Teamarbeit							
Ressourcenplanung							
Gewichtung: x-fach							
Beschreibung	Das Tool unterstützt die Eingabe, Planung und Verwaltung von Ressourcen, wie Personal-, Budget-, Zeit- und Materialaufwand. Darüber hinaus können Mitarbeiter individuelle Aufwände neu erstellen und von dem Tool verwalten lassen. Die Aufwände werden von den zuständigen Mitarbeitern aktualisiert und in grafischer Darstellung aufbereitet angezeigt. Vor dem Erreichen kritischer Abweichungen erfolgen automatisiert Warnungen bzw. Hinweise an die zuständigen Mitarbeiter.						
Begründung	Eine in sich stimmige, jederzeit aktuelle und zentrale Ressourcenplanung ist essenziell für die Existenz und den Erfolg eines Projekts. Warnhinweise unterstützen die Steuerung und ermöglichen es, frühzeitig Gegenmaßnahmen einzuleiten.						
Abnahmekriterien	<ol style="list-style-type: none"> 1. Eingabe der Ressourcenaufwände 2. Neue Ressourcenfelder erstellen 3. Aktualisierung der Aufwände 4. Automatisierte Aufbereitung 5. Automatisierte Hinweismeldungen <table border="1" style="width: 100%; margin-top: 10px;"> <tr> <td style="color: red;">nicht erfüllt:</td> <td>0 erfüllt</td> </tr> <tr> <td style="color: blue;">teilweise erfüllt:</td> <td>Erstes und drittes Kriterium erfüllt</td> </tr> <tr> <td style="color: green;">voll erfüllt:</td> <td>> teilweise erfüllt</td> </tr> </table>	nicht erfüllt:	0 erfüllt	teilweise erfüllt:	Erstes und drittes Kriterium erfüllt	voll erfüllt:	> teilweise erfüllt
nicht erfüllt:	0 erfüllt						
teilweise erfüllt:	Erstes und drittes Kriterium erfüllt						
voll erfüllt:	> teilweise erfüllt						

Tab. 17: Ressourcenplanung

Ist die Anforderung aus Tabelle 17 erfüllt, kann auch die im Sprint Planning stattfindende Ressourcenplanung der einzelnen User Stories im Tool abgebildet werden.

Funktionale Anforderung: Unterstützung der Teamarbeit							
Priorisierung von Anforderungen	Gewichtung: x-fach						
Beschreibung	Für jede Anforderung kann eine Priorisierung vorgenommen werden, aufgrund derer der Umsetzungszeitpunkt festgelegt wird. Die Priorität bestimmt die Höhe des Beitrags einer Anforderung zum Erfolg des Projekts. Diese Informationen verwaltet das Tool und sendet die aufbereiteten, aktualisierten Informationen an alle beteiligten Mitarbeiter, falls eine Änderung erfolgt. Priorisierungen dürfen nur mit den entsprechenden Berechtigungen vorgenommen werden.						
Begründung	Projektbeteiligte wissen zu jeder Zeit, welche Anforderungen in welcher Reihenfolge zu erledigen sind.						
Abnahmekriterien	<ol style="list-style-type: none"> 1. Zuweisen einer Priorität 2. Ändern einer Priorität 3. Automatisierte Verteilung von Informationen an beteiligte Mitarbeiter über Änderungen von Prioritäten <table border="1"> <tr> <td>nicht erfüllt:</td> <td>< teilweise erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>Erstes Kriterium erfüllt</td> </tr> <tr> <td>voll erfüllt:</td> <td>> teilweise erfüllt</td> </tr> </table>	nicht erfüllt:	< teilweise erfüllt	teilweise erfüllt:	Erstes Kriterium erfüllt	voll erfüllt:	> teilweise erfüllt
nicht erfüllt:	< teilweise erfüllt						
teilweise erfüllt:	Erstes Kriterium erfüllt						
voll erfüllt:	> teilweise erfüllt						

Tab. 18: Priorisierung von Anforderungen

Um die Anforderung aus Tabelle 18 für SCRUM-Projekte zu erfüllen, muss die Änderung der Prioritäten (Abnahmekriterium 2) im Tool möglich sein, da die Priorität maßgeblich die Position einer Anforderung im Sprint Backlog und damit ihren Umsetzungszeitpunkt beeinflusst. Bei der Anwendung von SCRUM ist es daher empfehlenswert, das Ändern einer Priorität als Pflichtabnahmekriterium festzulegen.

Funktionale Anforderung: Unterstützung der Teamarbeit							
Kommentarfunktion	Gewichtung: x-fach						
Beschreibung	Jede Anforderung kann von allen berechtigten Projektbeteiligten kommentiert werden. Das Tool bietet die Möglichkeit, die Kommentare ein- und auszublenden. Das Tool kann weitere Hilfsfunktionen bei der Erstellung von Kommentaren anbieten (Rechtschreibprüfung, farbliche Darstellung von unterschiedlichen Autoren, usw.).						
Begründung	Die Kommentarfunktion bietet die Möglichkeit zum Informationsaustausch und zur Beantwortung von Fragen. Auf diese Weise können Änderungswünsche bzw. -vorschläge vor der tatsächlichen Umsetzung einer Anforderungsänderung diskutiert werden. Unnötige Änderungen können so vermieden werden. Weiterführende Informationen zu einer bestimmten Anforderung können gezielt abgelegt werden.						
Abnahmekriterien	<ol style="list-style-type: none"> 1. Kommentare erstellen 2. Übersichtliche Strukturierung der Kommentare 3. Weitere Funktionen (Rechtschreibprüfung, farbliche Darstellung von unterschiedlichen Autoren, usw.) <table border="1"> <tr> <td>nicht erfüllt:</td> <td>< teilweise erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>Erstes Kriterium erfüllt</td> </tr> <tr> <td>voll erfüllt:</td> <td>> teilweise erfüllt</td> </tr> </table>	nicht erfüllt:	< teilweise erfüllt	teilweise erfüllt:	Erstes Kriterium erfüllt	voll erfüllt:	> teilweise erfüllt
nicht erfüllt:	< teilweise erfüllt						
teilweise erfüllt:	Erstes Kriterium erfüllt						
voll erfüllt:	> teilweise erfüllt						

Tab. 19: Kommentarfunktion

Funktionale Anforderung: Unterstützung der Teamarbeit							
Nachvollziehbarkeit des Fortschritts							
Gewichtung: x-fach							
Beschreibung	Einer Anforderung können unterschiedliche Status zugeordnet werden. Nach dieser Information können gewünschte Anforderungen sortiert und aufbereitet dargestellt werden. Für die Erreichung eines Status können Bedingungen (z.B. Zustimmung des Projektleiters) eingestellt und abgefragt werden. Hinweise und Warnungen erfolgen, wenn Anforderungen über einen bestimmten Zeitraum einen bestimmten Status besitzen.						
Begründung	Überblick über Anforderungsumsetzung und Fortschritt des Projekts wird ermöglicht. Zusätzlich können Gegenmaßnahmen eingeleitet werden. Kontrollen durch Abfrage von erfüllten Bedingungen steigern die Sicherheit. Die Funktionen „Bedingungen festlegen“ und „Hinweise ausgeben“ können zur Qualitätssicherung beitragen, indem beispielsweise definiert wird, dass der Projektleiter die Umsetzung einer Anforderung freigeben muss.						
Abnahmekriterien	<ol style="list-style-type: none"> 1. Zuteilung von Status 2. Darstellung des Umsetzungsfortschritts 3. Festlegung von Bedingungen zur Erfüllung eines Status 4. Automatisierte Hinweise bei gewünschten Ereignissen <table border="1" style="width: 100%; margin-top: 10px;"> <tr> <td style="background-color: #f2f2f2;">nicht erfüllt:</td> <td>< teilweise erfüllt</td> </tr> <tr> <td style="background-color: #f2f2f2;">teilweise erfüllt:</td> <td>Erstes und zweites Kriterium erfüllt</td> </tr> <tr> <td style="background-color: #f2f2f2;">voll erfüllt:</td> <td>> teilweise erfüllt</td> </tr> </table>	nicht erfüllt:	< teilweise erfüllt	teilweise erfüllt:	Erstes und zweites Kriterium erfüllt	voll erfüllt:	> teilweise erfüllt
nicht erfüllt:	< teilweise erfüllt						
teilweise erfüllt:	Erstes und zweites Kriterium erfüllt						
voll erfüllt:	> teilweise erfüllt						

Tab. 20: Nachvollziehbarkeit des Fortschritts

Die in der Tabelle 20 dargestellte Anforderung zur Nachvollziehbarkeit des Fortschritts ist bei der Anwendung von SCRUM besonders im Rahmen der Erstellung von Burn-Down-Charts zur Steuerung des Projekts wichtig.

Funktionale Anforderung: Unterstützung der Teamarbeit							
Nachvollziehbarkeit einer Änderung							
Gewichtung: x-fach							
Beschreibung	Für jede Anforderung wird eine Änderungshistorie hinterlegt und verwaltet, auf die die Anwender Lesezugriff besitzen. Eine Änderung an einer Anforderung soll protokolliert werden, sodass erkennbar ist, wer eine Anforderung zu welchem Zeitpunkt wie verändert hat. In diesem Zusammenhang kommt auch der Versionierung der Anforderungsdefinitionen eine wichtige Rolle zu. Dabei bleiben überarbeitete Versionsstände erhalten, sodass ein Zugriff darauf zu jeder Zeit möglich ist. Eine grafische Visualisierung der Änderungen zwischen zwei Versionen erhöht die Übersichtlichkeit, beispielsweise durch farbliche Markierung neu hinzugekommener Textpassagen.						
Begründung	Möglichkeit der Nachvollziehbarkeit von Änderungen auch ohne Ansprechpartner oder bei länger zurückliegenden Änderungen. Durch die Protokollierung wird die Nachvollziehbarkeit einer Änderung gewährleistet. Durch den Zugriff auf vorherige Versionen einer Anforderungsdefinition wird das Rückgängigmachen von Änderungen im Bedarfsfall erleichtert.						
Abnahmekriterien	<ol style="list-style-type: none"> 1. Anlage einer Änderungshistorie 2. Lesezugriff für Anwender 3. Versionierung der Änderungen 4. Grafische Aufbereitung der Änderungen <table border="1" style="width: 100%; margin-top: 10px;"> <tr> <td>nicht erfüllt:</td> <td>< teilweise erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>Erstes, zweites und drittes Kriterium erfüllt</td> </tr> <tr> <td>voll erfüllt:</td> <td>alle erfüllt</td> </tr> </table>	nicht erfüllt:	< teilweise erfüllt	teilweise erfüllt:	Erstes, zweites und drittes Kriterium erfüllt	voll erfüllt:	alle erfüllt
nicht erfüllt:	< teilweise erfüllt						
teilweise erfüllt:	Erstes, zweites und drittes Kriterium erfüllt						
voll erfüllt:	alle erfüllt						

Tab. 21: Nachvollziehbarkeit einer Änderung

Funktionale Anforderung: Unterstützung der Teamarbeit							
Beziehungsmanagement	Gewichtung: x-fach						
Beschreibung	<p>Durch ein vom Tool verwaltetes Beziehungsmanagement werden die Anforderungen strukturiert aufeinander referenziert. Beziehungen zwischen Anforderungen werden bei Änderungen vom Tool berücksichtigt und bei Bedarf erfolgen Warnungen (z.B. beim Löschen einer Anforderung, die eine Referenz zu einer anderen Anforderung enthält). Zudem sind Verweise zwischen den einzelnen Anforderungen möglich, zum Beispiel zu vorher zu beendeten Anforderungen.</p> <p>Erweiterung: Anzeige abhängiger Anforderungen, bei Änderung der Priorität einer referenzierten Anforderung. So wird dem Bearbeiter angezeigt, welche referenzierten Anforderungen bzgl. der Priorität anzupassen sind.</p>						
Begründung	Vom Tool verwaltete Beziehungen ermöglichen ein besseres Verständnis und verdeutlichen die Abhängigkeiten, die bei Änderungen zu berücksichtigen sind.						
Abnahmekriterien	<ol style="list-style-type: none"> 1. Generierung von Beziehungen und Abhängigkeiten zwischen Anforderungen 2. Verwaltung der Beziehungen 3. Generierung von Hinweismeldungen <table border="1" style="width: 100%; margin-top: 10px;"> <tr> <td>nicht erfüllt:</td> <td>0 erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>Erstes Kriterium erfüllt</td> </tr> <tr> <td>voll erfüllt:</td> <td>> teilweise erfüllt</td> </tr> </table>	nicht erfüllt:	0 erfüllt	teilweise erfüllt:	Erstes Kriterium erfüllt	voll erfüllt:	> teilweise erfüllt
nicht erfüllt:	0 erfüllt						
teilweise erfüllt:	Erstes Kriterium erfüllt						
voll erfüllt:	> teilweise erfüllt						

Tab. 22: Beziehungsmanagement

Funktionale Anforderung: Unterstützung der Teamarbeit							
Berechtigungsmanagement Ebene Anforderung	Gewichtung: x-fach						
Beschreibung	Das Tool ermöglicht die Berechtigungsvergabe (Lese- und/oder Schreibberechtigung) für jede Anforderung. Darüber hinaus wird die Verwaltung in einem Rollenkonzept angeboten.						
Begründung	Besonders in Unternehmen mit vielen Mitarbeitern ist es wichtig, den Zugriff auf vertrauliche Anforderungen nur für einen bestimmten Berechtigtenkreis freizugeben.						
Abnahmekriterien	<ol style="list-style-type: none"> 1. Vergabe von Lese- und Schreibzugriff auf Anforderungen für alle Anwender 2. Rollenkonzept <table border="1" style="width: 100%; margin-top: 10px;"> <tr> <td>nicht erfüllt:</td> <td>0 erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>Erstes Kriterium erfüllt</td> </tr> <tr> <td>voll erfüllt:</td> <td>2 erfüllt</td> </tr> </table>	nicht erfüllt:	0 erfüllt	teilweise erfüllt:	Erstes Kriterium erfüllt	voll erfüllt:	2 erfüllt
nicht erfüllt:	0 erfüllt						
teilweise erfüllt:	Erstes Kriterium erfüllt						
voll erfüllt:	2 erfüllt						

Tab. 23: Berechtigungsmanagement Ebene Anforderung

Funktionale Anforderung: Unterstützung der Teamarbeit							
Berechtigungsmanagement Ebene Tool	Gewichtung: x-fach						
Beschreibung	Das Tool unterstützt die Rechteverwaltung für den Zugriff auf das Tool. Darüber hinaus werden die Verwaltung in einem Rollenkonzept und die Anbindung an Directory Services angeboten.						
Begründung	Um die Informationssicherheit zu gewährleisten, werden nur die Mitarbeiter für den Zugriff auf das Tool berechtigt, die diesen für die Erfüllung ihrer Aufgaben benötigen. Für eine effiziente Benutzerverwaltung benötigen besonders große Unternehmen Funktionalitäten, die eine rollenbasierte Berechtigungsvergabe und eine Anbindung an Directory Services ermöglichen.						
Abnahmekriterien	<ol style="list-style-type: none"> 1. Zugriffsverwaltung durch Vergabe von Berechtigungen 2. Unterstützung des rollenbasierten Berechtigungsmanagements 3. Anbindung an Directory Services <table border="1"> <tr> <td>nicht erfüllt:</td> <td>< teilweise erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>Erstes Kriterium erfüllt</td> </tr> <tr> <td>voll erfüllt:</td> <td>> teilweise erfüllt</td> </tr> </table>	nicht erfüllt:	< teilweise erfüllt	teilweise erfüllt:	Erstes Kriterium erfüllt	voll erfüllt:	> teilweise erfüllt
nicht erfüllt:	< teilweise erfüllt						
teilweise erfüllt:	Erstes Kriterium erfüllt						
voll erfüllt:	> teilweise erfüllt						

Tab. 24: Berechtigungsmanagement Ebene Tool

Funktionale Anforderung: Unterstützung der Teamarbeit							
Mehrbenutzerfähigkeit	Gewichtung: x-fach						
Beschreibung	Das Tool ermöglicht, dass eine große Anzahl an Mitarbeitern in einem gemeinsamen Projektraum zusammenarbeitet. Wird auf eine Anforderung von mehr als einem Mitarbeiter gleichzeitig zugegriffen, verhindert das Tool die Entstehung von Inkonsistenzen, indem der Zugriff auf die Anforderung während der Bearbeitungszeit gesperrt wird. Bei Aufruf der gesperrten Anforderung wird ein Hinweis angezeigt.						
Begründung	Die gleichzeitige Benutzung des Tools durch mehrere Mitarbeiter ist für eine effektive, teambasierte Projektarbeit essenziell.						
Abnahmekriterien	<ol style="list-style-type: none"> 1. Mehrbenutzerfähigkeit wird ermöglicht 2. Bearbeitungskonflikte werden vermieden <table border="1"> <tr> <td>nicht erfüllt:</td> <td>< teilweise erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>Erstes Kriterium erfüllt</td> </tr> <tr> <td>voll erfüllt:</td> <td>> teilweise erfüllt</td> </tr> </table>	nicht erfüllt:	< teilweise erfüllt	teilweise erfüllt:	Erstes Kriterium erfüllt	voll erfüllt:	> teilweise erfüllt
nicht erfüllt:	< teilweise erfüllt						
teilweise erfüllt:	Erstes Kriterium erfüllt						
voll erfüllt:	> teilweise erfüllt						

Tab. 25: Mehrbenutzerfähigkeit

Schnittstellenfunktionalitäten

Die Object Management Group definierte ein einheitliches Austausch-Format für Anforderung, das sogenannte Requirements Interchange Format (ReqIF). Durch die Verwendung des einheitlichen Formats soll die Verwendung von Anforderungen in unterschiedlichen Systemen erleichtert werden.³³

Funktionale Anforderung: Schnittstellenfunktionalitäten							
Import bestehender Anforderungen							
Gewichtung: x-fach							
Beschreibung	In anderen Tools oder in einem Textverarbeitungsprogramm erstellte Anforderungen können von dem Tool importiert werden. Zudem wird das Format ReqIF unterstützt.						
Begründung	In vielen Unternehmen werden Anforderungen vor der Einführung eines Requirements Engineering Tools in einem Textverarbeitungsprogramm, wie beispielsweise Microsoft Word, erstellt und verwaltet. Um bestehende Anforderungsdefinitionen nicht manuell in das Tool übernehmen zu müssen, ist eine Möglichkeit zum Import bestehender Anforderungen wichtig.						
Abnahmekriterien	<ol style="list-style-type: none"> 1. Es besteht die Möglichkeit, in einem Textverarbeitungsprogramm erstellte Anforderungen zu importieren 2. ReqIF wird unterstützt 3. Eine Importfunktion aus Datenbanken steht zur Verfügung 						
	<table border="1"> <tr> <td>nicht erfüllt:</td> <td>< teilweise erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>Erstes und zweites Kriterium erfüllt</td> </tr> <tr> <td>voll erfüllt:</td> <td>> teilweise erfüllt</td> </tr> </table>	nicht erfüllt:	< teilweise erfüllt	teilweise erfüllt:	Erstes und zweites Kriterium erfüllt	voll erfüllt:	> teilweise erfüllt
nicht erfüllt:	< teilweise erfüllt						
teilweise erfüllt:	Erstes und zweites Kriterium erfüllt						
voll erfüllt:	> teilweise erfüllt						

Tab. 26: Import bestehender Anforderungen

Funktionale Anforderung: Schnittstellenfunktionalitäten							
Export von Anforderungen							
Gewichtung: x-fach							
Beschreibung	Im Tool verwaltete Anforderungen können exportiert werden. Dabei soll ReqIF unterstützt werden. Auch ein Export in Textverarbeitungsprogramme, wie beispielsweise Microsoft Word, soll angeboten werden.						
Begründung	Eine Exportmöglichkeit lässt es zu, Anforderungen zwischen verschiedenen Interessengruppen auszutauschen, die keinen Zugriff auf das Tool erhalten sollen. Zudem können die Anforderungen durch einen Export mit einem einheitlichen Datenformat auch in anderen Tools verwendet werden. Auf diese Weise wird manueller Aufwand bei der Anforderungsdefinition verhindert.						
Abnahmekriterien	<ol style="list-style-type: none"> 1. ReqIF wird unterstützt 2. Der Export in Textverarbeitungsprogramme wird unterstützt 3. Es steht eine Exportfunktion für Datenbanken zur Verfügung 						
	<table border="1"> <tr> <td>nicht erfüllt:</td> <td>< teilweise erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>Erstes und zweites Kriterium erfüllt</td> </tr> <tr> <td>voll erfüllt:</td> <td>> teilweise erfüllt</td> </tr> </table>	nicht erfüllt:	< teilweise erfüllt	teilweise erfüllt:	Erstes und zweites Kriterium erfüllt	voll erfüllt:	> teilweise erfüllt
nicht erfüllt:	< teilweise erfüllt						
teilweise erfüllt:	Erstes und zweites Kriterium erfüllt						
voll erfüllt:	> teilweise erfüllt						

Tab. 27: Export von Anforderungen

³³ Vgl. Object Management Group (o.J.)

Funktionale Anforderung: Schnittstellenfunktionalitäten							
Verknüpfung zu weiteren Aufgaben der Softwareentwicklung	Gewichtung: x-fach						
Beschreibung	Für eine effektive Einbindung des Tools in das Software Engineering werden Schnittstellenfunktionalitäten benötigt. Damit ist es möglich, weitere Systeme, wie Bug-Tracker oder Testmanagement-Tools, an das Tool anzubinden.						
Begründung	Durch die Verknüpfung von Anforderungen mit weiteren Funktionen des Software Engineerings werden Beziehungen auf einem höheren Abstraktionsgrad dargestellt. Durch die nun mögliche, umfassende Informationsbasis wird die effiziente Zusammenarbeit zwischen den oftmals auf mehrere Unternehmensbereiche verteilten Funktionalitäten ermöglicht. Beispielsweise können Testfälle mit den zugehörigen Anforderungen verknüpft werden.						
Abnahmekriterien	<p>1. Möglichkeit zur Anbindung von Tools</p> <table border="1"> <tr> <td>nicht erfüllt:</td> <td>0 erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>-</td> </tr> <tr> <td>voll erfüllt:</td> <td>Erstes Kriterium erfüllt</td> </tr> </table>	nicht erfüllt:	0 erfüllt	teilweise erfüllt:	-	voll erfüllt:	Erstes Kriterium erfüllt
nicht erfüllt:	0 erfüllt						
teilweise erfüllt:	-						
voll erfüllt:	Erstes Kriterium erfüllt						

Tab. 28: Verknüpfung zu weiteren Aufgaben der Softwareentwicklung

3.3.3 Nichtfunktionale Anforderungen

Bedienbarkeit

Nichtfunktionale Anforderung: Bedienbarkeit							
Usability und Erlernbarkeit	Gewichtung: x-fach						
Beschreibung	Das Tool ist intuitiv zu bedienen und unterstützt den Anwender mit Funktionen, wie einer Hilfe-Funktion, Web Based Training (WBT) zum Erlernen der Toolfunktionalitäten oder einem „erste-Schritte“-Tutorial. Weitere softwareergonomische Kriterien, wie Barrierefreiheit, Steuerbarkeit oder Erwartungskonformität, werden von dem Tool erfüllt.						
Begründung	Hohe Akzeptanz bei den Anwendern; gesteigerte Produktivität durch minimierte Einarbeitungszeiten; gesteigerte Effizienz im Umgang mit dem Tool.						
Abnahmekriterien	<p>1. Unterstützung der Anwender durch Funktionen (Hilfe-Funktion, WBTs zum Erlernen der Toolfunktionalitäten oder „erste-Schritte“-Tutorial)</p> <p>2. Intuitive Bedienbarkeit durch Erfüllung aller softwareergonomischen Kriterien</p> <table border="1"> <tr> <td>nicht erfüllt:</td> <td>< teilweise erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>Erstes Kriterium erfüllt</td> </tr> <tr> <td>voll erfüllt:</td> <td>> teilweise erfüllt</td> </tr> </table>	nicht erfüllt:	< teilweise erfüllt	teilweise erfüllt:	Erstes Kriterium erfüllt	voll erfüllt:	> teilweise erfüllt
nicht erfüllt:	< teilweise erfüllt						
teilweise erfüllt:	Erstes Kriterium erfüllt						
voll erfüllt:	> teilweise erfüllt						

Tab. 29: Usability und Erlernbarkeit

Nichtfunktionale Anforderung: Bedienbarkeit							
Personalisierbarkeit	Gewichtung: x-fach						
Beschreibung	Die Oberfläche des Tools ist an die individuellen Bedürfnisse des Anwenders anpassbar, beispielsweise durch personalisierte Anordnung der Oberflächenelemente. Des Weiteren können Arbeitsabläufe durch die Anwendung von Tastenkombinationen oder der Erstellung von Makros verkürzt werden.						
Begründung	Durch die individuelle Gestaltung können vor allem routinierte Tätigkeiten schneller und effizienter ausgeführt werden.						
Abnahmekriterien	<ol style="list-style-type: none"> 1. Anpassbarkeit der Oberfläche 2. Tastenkombinationen und Makros anwendbar <table border="1"> <tr> <td>nicht erfüllt:</td> <td>< teilweise erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>Erstes Kriterium erfüllt</td> </tr> <tr> <td>voll erfüllt:</td> <td>> teilweise erfüllt</td> </tr> </table>	nicht erfüllt:	< teilweise erfüllt	teilweise erfüllt:	Erstes Kriterium erfüllt	voll erfüllt:	> teilweise erfüllt
nicht erfüllt:	< teilweise erfüllt						
teilweise erfüllt:	Erstes Kriterium erfüllt						
voll erfüllt:	> teilweise erfüllt						

Tab. 30: Personalisierbarkeit

Sicherheit

Nichtfunktionale Anforderung: Sicherheit							
Sicherheit der Daten (Verschlüsselung)	Gewichtung: x-fach						
Beschreibung	Die im Tool abgespeicherten Daten werden durch ein geeignetes Verschlüsselungsverfahren vor unberechtigten Zugriffen geschützt.						
Begründung	Der Datensicherheit kommt eine wichtige Rolle zu.						
Abnahmekriterien	<ol style="list-style-type: none"> 1. Verschlüsselungsverfahren <table border="1"> <tr> <td>nicht erfüllt:</td> <td>0 erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>-</td> </tr> <tr> <td>voll erfüllt:</td> <td>Erstes Kriterium erfüllt</td> </tr> </table>	nicht erfüllt:	0 erfüllt	teilweise erfüllt:	-	voll erfüllt:	Erstes Kriterium erfüllt
nicht erfüllt:	0 erfüllt						
teilweise erfüllt:	-						
voll erfüllt:	Erstes Kriterium erfüllt						

Tab. 31: Sicherheit der Daten (Verschlüsselung)

Nichtfunktionale Anforderung: Sicherheit							
Sicherung der Daten	Gewichtung: x-fach						
Beschreibung	Funktionen zur Erstellung von Backups und zur Wiederherstellung von Daten tragen zur Sicherung der Daten bei. Das Tool besitzt Funktionalitäten, um die Sicherung der Daten zu erfüllen.						
Begründung	Der Verlust von Daten muss vermieden werden.						
Abnahmekriterien	<ol style="list-style-type: none"> 1. Verfahren zur Sicherung der Daten (Backup, Wiederherstellung) <table border="1"> <tr> <td>nicht erfüllt:</td> <td>0 erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>-</td> </tr> <tr> <td>voll erfüllt:</td> <td>Erstes Kriterium erfüllt</td> </tr> </table>	nicht erfüllt:	0 erfüllt	teilweise erfüllt:	-	voll erfüllt:	Erstes Kriterium erfüllt
nicht erfüllt:	0 erfüllt						
teilweise erfüllt:	-						
voll erfüllt:	Erstes Kriterium erfüllt						

Tab. 32: Sicherung der Daten

Herstellereigenschaften

Nichtfunktionale Anforderung: Herstellereigenschaften							
Herstellerbewertung	Gewichtung: x-fach						
Beschreibung	Der Hersteller wird bewertet durch eine stabile Marktposition, aussagekräftige Referenzen und die Höhe des Umsatzes.						
Begründung	Da nach der Einführung des Tools eine Abhängigkeit zwischen dem Hersteller und dem Unternehmen besteht, muss sichergestellt werden, dass der Hersteller weiterhin für die Betreuung und Weiterentwicklung des Tools ausreichend zur Verfügung steht.						
Abnahmekriterien	<ol style="list-style-type: none"> 1. Stabile Marktposition (entsprechender Umsatz) 2. Aussagekräftige Referenzen <table border="1" style="width: 100%; margin-top: 5px;"> <tr> <td>nicht erfüllt:</td> <td>< teilweise erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>Erstes Kriterium erfüllt</td> </tr> <tr> <td>voll erfüllt:</td> <td>> teilweise erfüllt</td> </tr> </table>	nicht erfüllt:	< teilweise erfüllt	teilweise erfüllt:	Erstes Kriterium erfüllt	voll erfüllt:	> teilweise erfüllt
nicht erfüllt:	< teilweise erfüllt						
teilweise erfüllt:	Erstes Kriterium erfüllt						
voll erfüllt:	> teilweise erfüllt						

Tab. 33: Herstellerbewertung

Nichtfunktionale Anforderung: Herstellereigenschaften							
Support	Gewichtung: x-fach						
Beschreibung	Der Hersteller wird auch bewertet durch den angebotenen Support. Neben einer entsprechenden Verfügbarkeit des Supports, sollen auch geringe Reaktionszeiten und die Möglichkeit zum Abschluss eines Wartungsvertrages gegeben sein.						
Begründung	Auftretende Probleme sollen vom Support möglichst schnell behoben werden, um Produktivitätseinbußen zu verhindern bzw. zu minimieren.						
Abnahmekriterien	<ol style="list-style-type: none"> 1. Ausreichende Verfügbarkeit (Verhältnis: Unternehmensgröße ↔ Erreichbarkeit) 2. Geringe Reaktionszeiten 3. Möglichkeit zum Abschluss eines Wartungsvertrages <table border="1" style="width: 100%; margin-top: 5px;"> <tr> <td>nicht erfüllt:</td> <td>< teilweise erfüllt</td> </tr> <tr> <td>teilweise erfüllt:</td> <td>Erstes und zweites Kriterium erfüllt</td> </tr> <tr> <td>voll erfüllt:</td> <td>> teilweise erfüllt</td> </tr> </table>	nicht erfüllt:	< teilweise erfüllt	teilweise erfüllt:	Erstes und zweites Kriterium erfüllt	voll erfüllt:	> teilweise erfüllt
nicht erfüllt:	< teilweise erfüllt						
teilweise erfüllt:	Erstes und zweites Kriterium erfüllt						
voll erfüllt:	> teilweise erfüllt						

Tab. 34: Support

3.4 Durchführung der Toolbewertung

Für die Toolbewertungen werden die Bedienungsanleitungen und Funktionalitätsbeschreibungen der Hersteller verwendet. Anhand dieser wird überprüft, ob ein Tool ein Abnahmekriterium erfüllt oder nicht. Durch die granulare Definition der Abnahmekriterien, kann ein solches entweder erfüllt sein, oder nicht. Der Erfüllungsgrad teilweise erfüllt ist auf Ebene der Abnahmekriterien nicht anwendbar.

Für die Toolbewertung wurde eine Excel-Datei erstellt, in der die Ergebnisse der Toolbetrachtungen eingetragen werden, wodurch die Übersichtlichkeit erhöht wird. Darüber hinaus

wird interessierten Unternehmen die Excel-Datei zur Verfügung gestellt. Die Unternehmen können die Toolbewertungen nachvollziehen und eigene Bewertungen für in der Marktanalyse nicht berücksichtigte Tools durchführen. Zudem kann die Gewichtung der Kriterien und des Kriterienkatalogs an das Einsatzgebiet des Tools im Unternehmen angepasst werden. Hier ist beispielsweise die Erweiterung des Kriterienkatalogs nach VOLERE möglich, sodass auch Spezifika, wie Projektziele, Randbedingungen für das Projekt und Projektspezi- (siehe Kapitel 2.1.3) in die Bewertung aufgenommen werden.

Abbildung 8 zeigt eine beispielhafte Toolbewertung in der Kategorie „Erweiterung der Basisfunktionalitäten“.

Kategorien	Kriterien	Gewichtung des	Abnahme	Tool 1	Tool 2
Erweiterung der Basisfunktionalitäten	Visuelle Detaillierung einer Anforderung	1			
			1.	j	
			2.	j	
	Erfüllungsgrad			voll erfüllt	nicht erfüllt
	Anpassbarkeit der Anforderungsattribute	1			
			1.	j	
			2.	j	
			3.	n	
			4.	n	
			5.	n	
	Erfüllungsgrad			teilw. erfüllt	nicht erfüllt
	Aufbereitung der Anforderungsdefinition	1			
			1.	j	
			2.	j	
			3.	j	
Erfüllungsgrad			voll erfüllt	nicht erfüllt	
Kategorienbewertung				83%	0%

Abb. 8: Beispielhafte Toolbewertung

Für „Tool 1“ ist für jedes Abnahmekriterium die Erfüllung eingetragen. Ein „j“ steht dabei für erfüllt, ein „n“ für nicht erfüllt. Für die Bewertung eines Tools muss lediglich eingetragen werden, ob ein Tool ein Abnahmekriterium erfüllt oder nicht. Der Erfüllungsgrad der zugehörigen Anforderung wird durch eine hinterlegte Formel automatisch berechnet. Auch die Kategorienbewertung wird automatisch gebildet. Jedes Unternehmen kann individuell festlegen, mit welcher Gewichtung eine Anforderung bei der Berechnung der Kategorienbewertung berücksichtigt werden soll. Im Beispiel werden alle Kriterien mit dem Faktor 1 gewichtet.

Falls ein Unternehmen ein Tool, das im Rahmen der Marktanalyse nicht berücksichtigt wurde, nach den vorhandenen Kriterien bewerten möchte, genügt es eine weitere Spalte im Excel hinzuzufügen und die hinterlegten Formeln zu übernehmen. Daraufhin wird eingetragen, welche Abnahmekriterien von dem Tool erfüllt werden, um das Tool in die Marktanalyse zu integrieren.

Wenn ein Unternehmen spezifische Abnahmekriterien berücksichtigen oder optionale Kriterien als Pflichtabnahmekriterien festlegen möchte, ist eine Änderung der hinterlegten Formel zur Bestimmung des Erfüllungsgrades notwendig. Zur Verdeutlichung wird die **Funktionsweise der Formel** beispielhaft anhand der Anforderung „Anpassbarkeit der Anforderungsattribute“ erläutert.

- Jedes Abnahmekriterium erhält für die interne Berechnung eine festgelegte Punktzahl. Pflichtabnahmekriterien müssen bei Erfüllung aggregiert mehr Punkte erhalten, als die optionalen Abnahmekriterien zusammen erhalten können.
- Im Beispiel sind die Abnahmekriterien 1. und 2. Pflichtabnahmekriterien. Sie erhalten bei Erfüllung daher jeweils 6 Punkte. Alle anderen Abnahmekriterien erhalten bei Erfüllung jeweils einen Punkt.
- Nach der Beurteilung aller Abnahmekriterien werden die erhaltenen Punkte intern summiert. Der Grad der Erfüllung leitet sich anschließend aus der erhaltenen Punktzahl ab. Diese Regel ist in Tabelle 35 visualisiert.

Entscheidungsregeln zur Erfüllung der Anforderung „Anpassbarkeit der Anforderungsattribute“			
<i>Anzahl der Punkte</i>	weniger als 12	genau 12	mehr als 12
<i>Erfüllungsgrad</i>	nicht erfüllt	teilweise erfüllt	voll erfüllt

Tab. 35: Entscheidungsregel

Bei Veränderung der Pflichtkriterien oder dem Hinzufügen neuer Abnahmekriterien kann die Formel mit geringem Aufwand entsprechend dieser Logik angepasst werden. Die Ergebnisse der Toolbewertungen werden anschließend grafisch aufbereitet, um einen aussagekräftigen Überblick über die Marktanalyseergebnisse zur Verfügung zu stellen.

4 Erstellung der Marktanalyse

In diesem Kapitel der Arbeit wird das oben beschriebene Vorgehen zur Erstellung der Marktübersicht umgesetzt. Zunächst erfolgt die Bildung der Shortlist, die aus einer durchgeführten Marktanalyse erarbeitet wird. Anschließend werden die Tools der Shortlist mit dem Kriterienkatalog bewertet. Dabei wird das Tool kurz beschrieben und die in der jeweiligen Kategorie erzielten Ergebnisse zusammengefasst. Ein Gesamtüberblick über die bewerteten Kriterien bietet eine Vergleichsmöglichkeit aller Tools. In Kapitel 4.3 werden die Tools der Shortlist grafisch aufbereitet. Dabei werden die Unterschiede zwischen einem gewichteten und dem ungewichteten Kriterienkatalog dargestellt.

4.1 Bildung der Shortlist

Um eine geeignete Auswahl an Tools für die Shortlist treffen zu können, die mit dem Kriterienkatalog bewertet werden sollen, wird zunächst ein Marktüberblick über die derzeit angebotenen Werkzeuge für Requirements Engineering erstellt. Hierzu findet eine Betrachtung von Open Source Tools und kommerziellen Programmen statt. Um explizit für alle Unternehmensgrößen passende Empfehlungen geben zu können, werden die Tools zunächst drei Unternehmensgrößen zugeordnet. Anschließend findet die Bildung einer Shortlist von Requirements Engineering Tools statt, die aus einer überschaubaren Anzahl von Vertretern der jeweiligen Unternehmensgröße besteht.

4.1.1 Marktüberblick

Zu Beginn werden die Tools den drei Unternehmensgrößen *klein*, *mittel* und *groß* zugeordnet. Bei der Einteilung der Unternehmen wurde der Paragraf § 267 des Handelsgesetzbuchs (HGB) herangezogen. Folgende Tabelle gibt einen Überblick über die Eigenschaften der jeweiligen Größe eines Unternehmens.

	Mitarbeiteranzahl (Jahresdurchschnitt)	Bilanzsumme (in Millionen Euro)	Umsatzerlöse (in Millionen Euro)
Kleine Unternehmen	Bis zu 50	Bis zu 4,84	Bis zu 9,68
Mittlere Unternehmen	Mehr als 50 und weniger als 250	Bis zu 19,25	Bis zu 38,5
Großunternehmen	Mehr als 250	Mehr als 19,25	Mehr als 38,5

Tab. 36: Definition der Unternehmensgrößen

Um den Bekanntheitsgrad sowie Relevanz der Tools in der Praxis durch Empfehlungen identifizieren zu können, wurden vertrauenswürdige Webseiten ausgewählt, die sich zum Teil auf das Thema spezialisiert haben.

Eine dieser Seiten ist *The Making of Software*³⁴, die Informationen zu dem Thema Software Engineering anbieten. Die umfassende Auflistung der 120 Tools basiert auf IncoSE, VOLE-RE, Ian Alexander, Tools Journal und zusätzlichen Websuchen.³⁵ Ausgehend von dieser Liste wurde eine komprimierte Liste von 17 Tools erstellt, die auf dem Markt am meisten verbreitet, vielversprechende und zukunftsorientierte Funktionen bieten. Als Anhaltspunkt für die Marktpräsenz wurden Erfahrungsberichte und Aussteller der *Requirements Engineering Conference (REConf) 2012-2014* herangezogen. Hinzu kamen Tools, die im *Forrester Wave des Application Lifecycle Management von 2012 (Q4)* und in der Marktforschung (2013) *Gartner Magic Quadrant for Integrated Software Quality Suite* enthalten waren.³⁶

Eine weitere Quelle ist das Whitepaper *How to Evaluate and Select a Requirements Management Tool* des Beratungsunternehmens für Business-Analysen Seilevel.³⁷ In diesem Whitepaper wird erläutert, wie ein Requirements Engineering Tool ausgewählt werden kann. Zunächst wird hier ein umfassender Kriterienkatalog erstellt. Dieser wird auf die wesentlichen Kriterien komprimiert und die einzelnen Kriterien werden priorisiert. Anhand der komprimierten Kriterienliste werden wenige Tools für die Evaluierung ausgewählt. Diese 15 Tools werden anschließend auf die umfassenden Kriterien geprüft und repräsentieren eine Punktzahl für deren Leistungen. Die Ergebnisse sind im Whitepaper *Seilevel's Evaluations of Requirements Management Tools: Summaries and Scores* enthalten.³⁸

Diese Vorgehensweise ist ähnlich wie bei International Council on Systems Engineering (INCOSE), jedoch finden die Überprüfungen der Kriterien nicht von den jeweiligen Herstellern statt, sondern von außenstehenden Forschern, was eine Verzerrung des Ergebnisses verhindert.

Die dritte Quelle für den Toolvergleich ist die Internetseite des International Council on Systems Engineering, welches eine gemeinnützige Organisation für Software Engineering ist. Die Tools Database Working Group (TDWG) und die INCOSE Requirements Management Working Group entwickelten einen Fragebogen für Anforderungsmanagement-Tools. Die Umfrageergebnisse, sowie deren Bewertungen für die Erfüllung der Kriterien werden von den jeweiligen Werkzeugherstellern zur Verfügung gestellt. Die Ergebnisse werden dennoch

³⁴ Vgl. *The Making of Software* (2014a)

³⁵ Vgl. *The Making of Software* (2014b)

³⁶ Vgl. *The Making of Software* (2014b)

³⁷ Vgl. Beatty, J. / Ferrari, R. (2011)

³⁸ Vgl. U.a. Beatty, J. (o.J.)

von der TDWG überprüft und ggf. korrigiert. Die Datenbank enthält derzeit die Ergebnisse des Fragebogens von 34 Tools.³⁹

Weitere Quellen sind Auflistungen von Werkzeugen auf der Website von VOLERE⁴⁰ und in Wikipedia⁴¹. Folgende Tabelle enthält die Tools, die auf den eben erläuterten Webseiten enthalten sind und deren Auftrittshäufigkeit. Ein Tool kann innerhalb dieser Tabelle die maximale Summe von fünf haben, wenn dieses auf allen fünf Webseiten enthalten ist.

	Making-ofsoftware	Sei-level	Incose	VO-LERE	Wiki-pedia	Häufigkeit
Blueprint Requirements Center	✓	✓	✓	✓		4
Caliber	✓	✓		✓	✓	4
codeBeamer Requirements Management	✓			✓		2
Enterprise Architect	✓	✓	✓	✓		4
HP Agile Manager	✓					1
HP Quality Center	✓			✓		2
IBM Rational DOORS	✓	✓	✓	✓	✓	5
IBM Rational Requirements Composer	✓	✓		✓		3
Innovator for Business Analysts	✓					1
inteGREAT	✓	✓	✓	✓		4
Jama	✓	✓	✓	✓	✓	5
Jira	✓					1
Jira Agile	✓					1
Kovair ALM Studio	✓	✓	✓			3
Mingle	✓					1
Polarion Requirements	✓	✓	✓	✓	✓	5
PTC Integrity	✓	✓	✓	✓	✓	5
Rally	✓					1
Serena Requirements Manager	✓		✓		✓	3
TestTrack RM	✓				✓	2
VersionOne	✓					1
Visure Requirements	✓			✓	✓	3
Cradle		✓	✓	✓		3
Application Lifecycle Management		✓				1
Team Foundation Server		✓				1
Qpack		✓				1
Teamcenter		✓	✓		✓	3
TopTeam Analyst		✓			✓	2

³⁹ Vgl. TDWG (2010)

⁴⁰ Vgl. Volere (2014)

⁴¹ Vgl. Wikipedia (2014a)

TraceCloud		✓	✓			2
Accept Requirements			✓	✓		2
Acclaro			✓			1
Aligned Elements			✓			1
Avenqo PEP			✓	✓		2
Cameo Requirements			✓			1
Case Spec			✓	✓		2
Cognition Cockpit			✓			1
Core			✓			1
Envision VIP			✓	✓		2
Rational RequisitePro			✓	✓		2
IRQA			✓			1
MagicDraw			✓			1
PACE			✓			1
Psoda			✓			1
RaQuest			✓	✓		2
ReqMan			✓	✓		2
Reqtify			✓	✓		2
Requirements Manager			✓	✓		2
RTIME			✓			1
What To Do Next			✓			1
Workspace			✓	✓		2

Tab. 37: Ermittlung der Häufigkeit eines Tools

Für die Auswahl der Tools, die mit dem Kriterienkatalog bewertet werden sollen, kommen jene infrage, die eine Häufigkeit von 4 und 5 haben. Dies sind folgende Tools:

Tool	Hersteller	Häufigkeit
Blueprint Requirements Center	Blueprint Software Systems	4
Caliber	Borland	4
Enterprise Architect	Sparx Systems	4
IBM Rational DOORS	IBM	5
inteGREAT	eDev Technologies	4
Jama	Jama Software	5
Polarion Requirements	Polarion Software	5
PTC Integrity	MKS	5

Tab. 38: Ergebnis des Vergleichs

Die im Vergleich am häufigsten auftretenden Tools mit der Häufigkeit 4 und 5 sind lediglich Tools, die sich für große und mittlere Unternehmen eignen. Deshalb ist die Auswahl der Tools für kleine Unternehmen nicht anhand des Vergleichs zu treffen, womit an dieser Stelle eine individuelle Auswahl notwendig ist. Da es auf dem Markt nur wenige Tools gibt, die speziell für den Einsatz in kleineren Unternehmen geeignet sind, werden lediglich zwei Tools aus diesem Bereich betrachtet. Das Tool Fogbugz wird nicht zur Bewertung herangezogen, weil es ein Tool für Bugtracking ist. Da das Tool RaQuest die Anwendung Enterprise Architect voraussetzt und somit kein eigenständiges Tool ist, wird dieses ebenfalls nicht für die

Bewertung im Kriterienkatalog ausgewählt. Somit werden die Tools Jira und Mingle als Vertreter der Tools für kleine Unternehmen betrachtet.

Zusammenfassend ergeben sich folgende kommerzielle Tools, die im weiteren Verlauf dieser Arbeit für die Bewertung mit dem Kriterienkatalog betrachtet werden.

Tool	Hersteller	Unternehmensgröße
Blueprint Requirements Center	Blueprint Software Systems	Groß
IBM Rational DOORS	IBM	Groß
Polarion Requirements	Polarion Software	Groß
inteGREAT	eDev Technologies	Groß
Enterprise Architect	Sparx Systems	Groß
Caliber	Borland	Mittel
Jama	Jama Software	Mittel
PTC Integrity	MKS	Mittel
Jira	Atlassian	Klein
Mingle	ThoughtWorks	Klein

Tab. 39: Ausgewählte kommerzielle Tools für die Bewertung mit dem Kriterienkatalog

4.1.3 Die Auswahl der Open Source Tools für die Shortlist

Im Open Source-Umfeld wurden die folgenden acht Tools begutachtet:

- OSRMT
- Rmtoo
- JavaRequirementsTracer
- Lifecycle Artefact Manager
- UNICASE
- ORMF
- ProR
- aNimble

Da die Qualität in diesem Umfeld von Tool zu Tool variiert, mussten spezifische Open Source-Kriterien erfüllt sein, um in die Shortlist aufgenommen zu werden. Für die Definition dieser Anforderungen wird die aus Kapitel 3.2 bekannte Karteikarte verwendet.

Open Source	
Direkter Ansprechpartner	
Beschreibung	Auf der Homepage ist eine E-Mail-Adresse oder eine Telefonnummer angegeben, mit der ein Experte kontaktiert werden kann.
Begründung	Ein direkter Ansprechpartner bietet unkomplizierte und schnelle Hilfe bei auftretenden, individuellen Problemen.
Abnahmekriterien	1. E-Mail-Adresse oder Telefonnummer sollen auf der Homepage vorhanden sein

Tab. 40: Direkter Ansprechpartner

Open Source	
Release-Abstände	
Beschreibung	Der Release-Abstand ist die Zeit, die seit der letzten veröffentlichten Version vergangen ist.
Begründung	Ein kurzer Zeitraum stellt sicher, dass das Produkt sicherheitstechnisch auf dem neuesten Stand ist.
Abnahmekriterien	1. Letztes Release ist jünger als ein Jahr

Tab. 41: Release-Abstände

Open Source	
Lizenzierung	
Beschreibung	Handelt es sich um eine Softwarelizenz, die das Anpassen des Open Source Tools beinhaltet oder nicht.
Begründung	Um Unternehmen die Möglichkeit zu geben das Programm anzupassen, muss die Lizenz des Open Source Tools die Modifikation des Programms unterstützen.
Abnahmekriterien	1. Die Software darf modifiziert werden

Tab. 42: Lizenzierung

Open Source	
Aktivität des Forums/der Community	
Beschreibung	Diese Anforderung ist erfüllt, wenn ein Forum für Fragestellungen vorhanden ist. Die Aktivität der Mitglieder und des Forums wird mit der Mitgliederzahl und der Anzahl der erstellten Einträge überprüft.
Begründung	Gibt es im Forum eine hohe Anzahl an Einträgen (sog. Topics), ist die Supportwahrscheinlichkeit bei Problemen höher als in Foren mit wenigen Community-Beiträgen.
Abnahmekriterien	1. Ein Forum muss vorhanden sein 2. Die Community ist in dem Forum aktiv

Tab. 43: Aktivität des Forums/der Community

Open Source	
Alter des Produkts	
Beschreibung	Das Alter des Produkts gibt an, wie lange das Produkt auf dem Markt existiert.
Begründung	Ein hohes Alter bietet eine höhere Garantie für eine längere Beständigkeit des Produkts. Junge Produkte können sich eventuell am Markt nicht etablieren. Das ermittelte Durchschnittsalter beträgt fünf Jahre.
Abnahmekriterien	1. Das Alter eines Tools muss größer als das zum Vergleich herangezogene Durchschnittsalter von fünf Jahren.

Tab. 44: Alter des Produkts

Die wichtigsten Kriterien, die für eine Aufnahme in die Shortlist erfüllt sein müssen, sind die Kriterien „Direkter Ansprechpartner“ und „Aktivität des Forums/der Community“. Ein aktuelles Release ist notwendig, damit das Programm aktuelle Standards unterstützt, Sicherheitspatches und Fehlerbehebungen erhält. Die Aktivität im Forum und der Community zeigt, inwiefern Interesse an einer Weiterentwicklung an dem Tool besteht.

Das Kriterium „Direkter Ansprechpartner“ wird von dem Tool Rmtoo und dem Tool aNimble erfüllt. Bei Rmtoo sind auf der Homepage eine Anschrift, eine E-Mail-Adresse sowie eine Telefonnummer hinterlegt. Auf der Homepage von aNimble kann unter der Rubrik Contact Us können die Entwickler über ein Formular kontaktiert werden. Die anderen Tools bieten keinen direkten Ansprechpartner.

Bei den Release-Abständen gibt es große Unterschiede zwischen den Tools. Das Tool OS-RMT ist bereits seit dem 26. März 2007 in der Version 1.5 veröffentlicht. In den dazu veröffentlichten Release Notes wird von Fehlerbehebungen und dem Hinzufügen weiterer Features gesprochen. Ein weiteres Release ist nicht angekündigt und eine Internetrecherche dahin gehend ergab, dass das Tool durch aNimble weitergeführt werden soll. Aufgrund des Alters der letzten Version (sieben Jahre) und der in Zukunft fehlenden Updates erfüllt das Tool dieses Kriterium nach der oben angeführten Definition nicht.

Rmtoo, ein Tool ohne grafische Oberfläche, erhielt das letzte Update im Frühjahr 2012. Über die zukünftige Entwicklung des Programms bzw. die Einführung weiterer Features liegen keine Informationen vor. Aufgrund der fehlenden Hinweise über weitere Updates und dem Zeitpunkt der letzten veröffentlichten Version, wird das Kriterium der „Release-Abstände“ nicht erfüllt.

Das letzte Update des JavaRequirementTracer war im Jahr 2013. Informationen über die Fortführung des Projekts sind im Internet per Suchanfrage nicht zu finden. Durch die oben definierte Regelung, allen Tools, deren letztes Update länger als ein Jahr zurückliegt und bei denen in naher Zukunft kein Release geplant ist, mit einem „nicht erfüllt“ zu bewerten, erhält der JavaRequirementTracer ein „nicht erfüllt“ bei dieser Anforderung.

Die letzte Version des Lifecycle Artefact Managers ist aus dem Jahr 2013. Informationen über weitere Updates sind zu diesem Tool nicht zu finden. Weiter sei der Support bei diesem Tool mangelhaft, schreibt der Hersteller selbst auf www.sourceforge.net. Aufgrund des fehlenden Supports und des Alters des letzten Release erfüllt der Lifecycle Artefact Manager das Kriterium des „Release-Abstands“ nicht.

Das Tool UNICASE der Technischen Universität München erhielt seit 2012 kein weiteres Update und ist zurzeit in der Version 0.5.2 erhältlich. Weitere Informationen über geplante Versionen sind auf der Homepage nicht genannt und auch per Suchanfrage im Internet nicht zu finden. Ebenso wie bei den bis jetzt bewerteten Tools, erfüllt auch das Tool UNICASE aufgrund des Alters der letzten Version und der fehlenden Roadmap die Anforderung nicht.

Über die beiden Eclipse-Plugins, ORMF und ProR, lassen sich weder auf der offiziellen Homepage, noch per Internetsuche Informationen über den aktuellen Entwicklungsstand und den weiteren Updateverlauf finden. Seit wann die Plug-ins in der derzeit aktuellen Version auf dem Markt existieren, ist ebenfalls unklar. Ebenso ist der Support der Tools sehr eingeschränkt. Diese Eigenschaften sprechen nach den oben erklärten Kriterien gegen eine Aufnahme in die Shortlist.

Das Tool aNimble setzt auf dem Tool OSRMT der Version 1.5 auf. Das letzte Update des Tools wurde im April 2014 durchgeführt. Informationen über weitere Versionen sind nicht vorhanden. Durch die aktuelle Version, die jünger als ein Jahr ist, erfüllt aNimble die Anforderung an die Release-Abstände.

Die Tools sind unter verschiedenen Lizenzen veröffentlicht. Die folgende Tabelle gibt eine Übersicht über die von den Tools verwendeten Lizenzen.

Tool	Lizenztyp
OSRMT	GNU General Public License Version 2.0 (GPLv2)
Rmtoo	GPL v3
Java Requirement Tracer	Apache License V 2.0
Lifecycle Artefact Manager	GPL v2
UNICASE	Eclipse Public License Version 1.0 (EPL)
ORMF	Keine Angabe
ProR	Keine Angabe
aNimble	GPL v2

Tab. 45: Lizenztypen der Open Source Tools

Trotz der unterschiedlichen der Lizenztypen sind diese hinsichtlich der Verwendung und Modifikationen, die an den Tools vorgenommen werden dürfen, identisch. So muss ein Tool, dass angepasst wird und wieder veröffentlicht werden soll unter der gleichen Lizenz wie die nicht-modifizierte Version zuvor veröffentlicht werden. Die Lizenzen Apache License V2 und GPL v3 sind zueinander kompatibel. Das bedeutet, dass eine Lizenzänderung von Apache License V2 zu GPL v3 und umgekehrt keine Lizenzprobleme mit sich bringt.

Die Anforderung, ein aktives Forum und eine aktive Community zu besitzen, wird nur bei den Tools ORMF und aNimble erfüllt. Das Forum von ORMF ist jedoch kein ORMF-Forum, sondern das allgemeine Forum von Eclipse. Bei einer Suche nach Topics mit dem Begriff ORMF wurden keine Einträge gefunden. Durch das Fehlen von Einträgen zu dem Tool ORMF, erfüllt ORMF diese Anforderung nicht.

Bei dem Tool aNimble existieren zwei Foren. Eine Möglichkeit, um an Informationen zu gelangen, ist ein durch aNimble eingerichtetes GoogleForum zu nutzen. Hierfür ist ein GoogleAccount Voraussetzung für den Zugang. Bei einer Testanmeldung kam die Meldung, dass in dieser Gruppe keine Meldungen und Beiträge vorhanden seien. Bei der zweiten Möglichkeit, dem von aNimble bereitgestellten Forum, muss ebenfalls ein Account (bei aNimble selbst) angelegt werden. Nach der Registrierung hat der User Zugriff auf viele Topics, unter anderem Tutorials zur Installation und Anwendung des Tools.

Ob ein Tool das Kriterium des Produktalters erfüllt, hängt davon ab, ob es älter ist als das ermittelte Durchschnittsalter von allen Tools von fünf Jahren. Die Bewertung folgt dem Gedanken, dass langjährige Projekte sich gegen die Konkurrenz durchsetzen konnten und aufgrund dessen noch auf dem Markt existieren. Wie lange die einzelnen Tools auf dem Markt sind, ist in der nachfolgenden Tabelle zusammengefasst. Bei den Tools UNICASE, ORMF und ProR sind keine Zahlen über die erste Veröffentlichung angegeben. Diese Tools erhalten eine neutrale Bewertung und fließen nicht in die Berechnung des Durchschnitts mit ein.

Tool	Alter
OSMRT	2007
Rmtoo	2010
Java Requirement Tracer	2009
Lifecycle Artefact Manager	2008
UNICASE	/
ORMF	/
ProR	/
aNimble	2010
Durchschnittliches Erscheinungsjahr	2008,8 (~2009)

Tab. 46: Das Durchschnittsalter der Open Source Tools

Das Durchschnittsalter der Tools beträgt fünf Jahre. Die Tools OSRMT, JavaRequirement Tracer und Lifecycle Artefact Manager erhalten ein erfüllt. Die übrigen Tools (Rmtoo und aNimble) sind jünger als das ermittelte Durchschnittsalter von fünf Jahren und erhalten ein nicht erfüllt.

Abschließend folgt eine Zusammenfassung in Tabellenform, welche Tools welches Kriterium erfüllen und ob das Tool in der Shortlist berücksichtigt wird. Ein grüner Haken bedeutet, dass das Kriterium erfüllt, ein rotes Kreuz, dass das Kriterium nicht erfüllt wurde. Der blaue Kreis bedeutet, dass eine Anforderung aufgrund fehlender Informationen nicht bewertet werden kann.

Tool	Direkter Ansprechpartner	Release-Abstände	Lizenzierung	Aktivität des Forums/der Community	Alter des Produkts	Berücksichtigung in Shortlist
OSRMT	✗	✗	✓	✗	✓	Nein
Rmtoo	✓	✗	✓	✗	✗	Nein
JavaRequirements Tracer	✗	✗	✓	✗	✓	Nein
Lifecycle Artefact Manager	✗	✗	✓	✗	✓	Nein
UNICASE	✗	✗	✓	✗	●	Nein
ORMF	✗	✗	●	✗	●	Nein
ProR	✗	✗	●	✗	●	Nein
aNimble	✓	✓	✓	✓	✗	Ja

Tab. 47: Übersicht über die Aufnahme der Open Source Tools in die Shortlist

Die beiden Anforderungen, „Release-Abstände“ und „Aktivität des Forums/der Community“, die für eine Aufnahme in die Shortlist erfüllt werden müssen, erfüllt ausschließlich das Tool aNimble. Die restlichen Tools werden aufgrund der Nichterfüllung dieser beider Anforderungen nicht in die Shortlist aufgenommen.

Nach einer Betrachtung der Produkt- und Herstellerwebsites, der Produktfeatures und den bewerteten Kriterien, folgt die Einschätzung, dass die Open Source Tools sich primär für kleine Unternehmen eignen. Sollen die Tools eine spezielle Aufgabe in mittelständischen oder großen Unternehmen übernehmen und das Tool erfüllt die Anforderungen, die diese Aufgabe voraussetzt, kann über einen Einsatz der Tools hierfür nachgedacht werden.

4.2 Anwendung des Kriterienkatalogs

In diesem Kapitel werden die Tools der Shortlist nacheinander vorgestellt und mit dem Kriterienkatalog bewertet. Dafür werden Informationen aus den jeweiligen User Guides und von den Websites der Hersteller bezogen. Die Kriterien werden nicht durch praktische Erfahrungen mit dem Tool bewertet. Sind Informationen weder im User Guide, noch auf der Website des Herstellers beschrieben, wird bei der Bewertung davon ausgegangen, dass diese Kriterien nicht erfüllt werden. Am Ende des jeweiligen Tool-Abschnitts folgt eine Grafik, die einen Überblick über das Ergebnis, das das Tool in der jeweiligen Unterkategorie erreicht hat, geben soll. Um die Erfüllung der Abnahmekriterien nachvollziehen zu können, ist in Anhang 1 eine detaillierte Darstellung der ungewichteten Toolbewertungen enthalten.

Bei den nachfolgend betrachteten Tools handelt es sich ausschließlich um vertikale Tools, da diese nicht nur eine, sondern mehrere Phasen des Entwicklungsprozesses unterstützen.

4.2.1 Blueprint Requirements Center

Das Tool Blueprint Requirements Center ist das Kernprodukt der Firma Blueprint Software Systems. Die Firma wurde 2003 gegründet und ist spezialisiert auf Requirements Management. Mit 50 bis 150 Mitarbeitern gehört die Firma zu den kleineren Unternehmen,⁴² hat jedoch zahlreiche namhafte Kunden.⁴³

Eine Stärke des Tools ist die Definition von Anforderungen, die insbesondere mit dem grafischen Editor erfolgen kann. Zudem bietet Blueprint eine moderne GUI, die der Oberfläche von Microsoft Office ähnelt und webbasiert ist, sodass die Anwendung mit jedem beliebigen Webbrowser betrieben werden kann. Dadurch wird eine gewisse Flexibilität erreicht. Anforderungen können lediglich aus Microsoft Excel Dateien importiert werden. Es bietet jedoch mehrere Exportfunktionen zu anderen Softwareengineering Tools.⁴⁴

	Basisfunktionalitäten	Erweiterung der Basisfunktionalitäten	Darstellung und Aufbereitung	Unterstützung der Teamarbeit	Schnittstellenfunktionalitäten	Bedienbarkeit	Sicherheit	Herstellereigenschaften
Erfüllung	100%	100%	100%	95%	33%	50%	50%	50%
Erfüllung der Kategorien	90%					50%		

Abb. 12: Bewertung des Tools Blueprint Requirements Center

4.2.2 Rational DOORS

Das Tool Rational DOORS ist eine Softwarelösung der Firma IBM. Es ist ein Tool speziell für das Anforderungsmanagement für Systeme und IT-Anwendungen. Es bietet beispielhaft keine Funktionalität für die Verfolgung eines Status und den Fortschritt eines Projekts. Es gibt auch nicht die Möglichkeit, Anforderungen visuell darzustellen bzw. zu generieren. Dennoch bietet das Tool viele Möglichkeiten für das Customizing und die Skalierbarkeit, sodass genügend viele Benutzer mit dem Werkzeug arbeiten können. Das Tool eignet sich für Unternehmen, die mit großen Mengen an Anforderungen arbeiten und dementsprechend viele Benutzer das Tool verwenden.

⁴² Vgl. Glassdoor (o.J.)

⁴³ Vgl. Blueprint (o.J.)

⁴⁴ Vgl. Blueprint (2011)

Die Oberfläche des Tools ist schlicht gehalten und nicht modern, was sich bei der Usability auswirkt. Dennoch ist Rational DOORS eines der Standardprodukte für Requirements Engineering und ist schon lange Zeit auf dem Markt, was auf eine stabile Marktposition hindeutet.⁴⁵

	Basisfunktionalitäten	Erweiterung der Basisfunktionalitäten	Darstellung und Aufbereitung	Unterstützung der Teamarbeit	Schnittstellenfunktionalitäten	Bedienbarkeit	Sicherheit	Herstellereigenschaften
Erfüllung	100%	100%	100%	85%	100%	25%	100%	100%
Erfüllung der Kategorien	94%					75%		

Abb. 13: Bewertung des Tools Rational DOORS

4.2.3 Polarion Requirements

Das Tool Polarion Requirements wird von der Firma Polarion Software angeboten. Seit 2004 bietet diese Lösungen für Anforderungsmanagement, Application Lifecycle Management und Test Management an.

Polarion Requirements bietet einen webbasierten Client, der eine übersichtliche Benutzeroberfläche besitzt. Eine Besonderheit ist, dass das Tool zusammen mit Microsoft Word benutzt werden kann, da diese miteinander synchronisiert werden. Dennoch ist es einer der wenigen Tools, welches keine Rechtschreibprüfung anbietet.⁴⁶

	Basisfunktionalitäten	Erweiterung der Basisfunktionalitäten	Darstellung und Aufbereitung	Unterstützung der Teamarbeit	Schnittstellenfunktionalitäten	Bedienbarkeit	Sicherheit	Herstellereigenschaften
Erfüllung	100%	83%	100%	95%	67%	100%	50%	100%
Erfüllung der Kategorien	92%					83%		

Abb. 14: Bewertung des Tools Polarion Requirements

4.2.4 inteGREAT

Das Produkt inteGREAT des Herstellers eDevTechnologies kooperiert im Gegensatz zu den anderen Werkzeugen am stärksten mit Microsoft Produkten, sodass inteGREAT als eine

⁴⁵ Vgl. IBM (2010)

⁴⁶ Vgl. Polarion Software (2014)

Ergänzung der vorhandenen Microsoft Umgebung dienen soll. Dazu gehören z.B. Microsoft Visual Studio Team Foundation Server, Microsoft Share Point und die Microsoft Office Produkte Excel, Word und Project.⁴⁷

Da kein frei zugänglicher User Guide im Internet zur Verfügung steht, konnte keine Bewertung des Tools mit dem Kriterienkatalog stattfinden. Es ist jedoch erwähnenswert, dass das Tool im Bewertungssystem von Seilevel die höchste Punktzahl zwischen 17 Tools erzielen konnte.⁴⁸

4.2.5 Enterprise Architect

Das Tool Enterprise Architect der australischen Firma SparxSystems ist aktuell in der Version 10 auf dem Markt erhältlich. Das Tool kann durch diverse Add-ons mit Lösungen von anderen Herstellern zusammenarbeiten. Z.B. wird ein Add-on für IBMs Doors angeboten.⁴⁹ Zu den von Enterprise Architect erfassten Anforderungen können durch den User Reports erstellt werden, z.B. in Form von Text oder Diagrammen. Zusätzlich können die Reports exportiert und mit anderen Tools geöffnet werden, sodass die Weitergabe der Berichte an weitere Mitarbeiter vereinfacht wird. Eine Verknüpfung zu anderen Tools des Softwareengineering kann nicht hergestellt werden.

	Basisfunktionalitäten	Erweiterung der Basisfunktionalitäten	Darstellung und Aufbereitung	Unterstützung der Teamarbeit	Schnittstellenfunktionalitäten	Bedienbarkeit	Sicherheit	Herstellereigenschaften
Erfüllung	100%	33%	75%	60%	0%	50%	0%	100%
Erfüllung der Kategorien	58%					50%		

Abb. 15: Bewertung des Tools Enterprise Architect

⁴⁷ Vgl. inteGREAT (o.J.)

⁴⁸ Vgl. u.a. Beatty, J. (o.J.)

⁴⁹ Vgl. EA AddOn (2014)

4.2.6 Caliber

Der Hersteller des Produkts Caliber ist Borland, der ein Tochterunternehmen von Micro Focus ist. Borland bietet verschiedene Lösungen für den gesamten Zyklus der Softwareentwicklung an.

Die Oberfläche ist nicht zeitgemäß und kann zudem auch nicht personalisiert werden. Negativ zu bewerten ist, dass sich Anforderungen ausschließlich aus Microsoft Word importieren lassen.⁵⁰

	Basisfunktionalitäten	Erweiterung der Basisfunktionalitäten	Darstellung und Aufbereitung	Unterstützung der Teamarbeit	Schnittstellenfunktionalitäten	Bedienbarkeit	Sicherheit	Herstellereigenschaften
Erfüllung	100%	100%	100%	100%	33%	25%	100%	100%
Erfüllung der Kategorien	92%					75%		

Abb. 16: Bewertung des Tools Caliber

4.2.7 Jama

Jama ist eine Lösung für die Produktentwicklung und ist gleichzeitig das einzige Produkt, das die Firma Jama Software bietet. Mit ca. 65 Mitarbeitern und der Gründung in 2007 ist Jama Software ein junges und kleines Unternehmen.⁵¹

Das webbasierte Tool Jama soll den gesamten Lebenszyklus eines Produkts begleiten und unterstützt neben den Requirements Engineering Funktionalitäten auch das Test Management. Die Stärken des Produkts sind die Berichterstattung, sowie die Unterstützung der Teamarbeit. Ein großer Nachteil ist der fehlende Editor für das Modellieren im Tool.⁵²

	Basisfunktionalitäten	Erweiterung der Basisfunktionalitäten	Darstellung und Aufbereitung	Unterstützung der Teamarbeit	Schnittstellenfunktionalitäten	Bedienbarkeit	Sicherheit	Herstellereigenschaften
Erfüllung	100%	67%	100%	100%	33%	50%	50%	50%
Erfüllung der Kategorien	88%					50%		

Abb. 17: Bewertung des Tools Jama

⁵⁰ Vgl. Borland (2004)

⁵¹ Vgl. Inc. (o.J.)

⁵² Vgl. Jama Software (2014)

4.2.8 PTC Integrity

Parametric Technology Corporation (PTC), die 1985 gegründet wurde, bietet „Software für computer-aided design (CAD) und Product-Lifecycle-Management (PLM), sowie für Management- und Dokumentationslösungen.“⁵³ Die Firma beschäftigt etwa 6000 Mitarbeiter und ist seit 1989 börsennotiert.

Das Tool PTC Integrity wird von PTC seit der Übernahme der Firma MKS angeboten und ist eine „Plattform für das Änderungs- und Konfigurationsmanagement, die Unternehmenslösungen für das Änderungsmanagement in der technischen Entwicklung sowie das Application Lifecycle Management [...] unterstützt.“⁵⁴ So ist das Requirements Engineering nur ein Teil der Lösung und bietet darüber hinaus weitere Funktionen für das Softwareengineering.

Das Tool ist nicht in der Lage, Importe bzw. Exporte durchzuführen und benötigt zusätzlich das Produkt PTC Integrity Gateway, das eine Integrationslösung ist. Auch Verknüpfung zu anderen Tool sind nur mit dieser Zusatzsoftware möglich. Ein anderer Nachteil ist die nicht moderne Oberfläche.⁵⁵

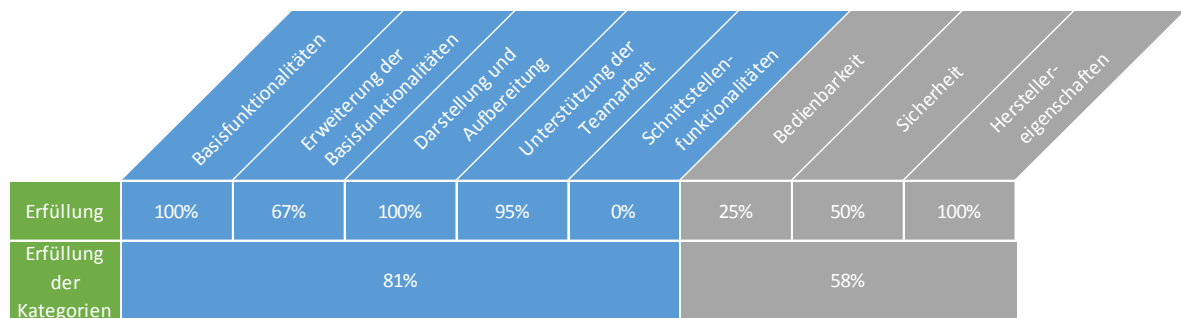


Abb. 18: Bewertung des Tools PTC Integrity

4.2.9 Jira

Der Hersteller des Tools Jira ist die Firma Atlassian. Die erste Version erschien kurz nach der Gründung des Unternehmens Jahr 2002. Bis heute ist es das Kernprogramm in der Produktpalette des Herstellers.⁵⁶ Neben dem Anforderungsmanagement-Tool Jira gibt es Lösungen für andere Aufgabenbereiche, wie beispielsweise Confluence, das es Teams ermöglicht, Daten strukturiert an einem Ort abzulegen. Aktuell arbeiten ca. 400 Mitarbeiter bei Atlassian.

Das Tool Jira bietet eine grafische Oberfläche, mit der die Anforderungen erstellt werden können. Anpassbare Attribute sind hingegen nicht möglich. Änderungen an Anforderungen werden in einer Historie gespeichert und können durch die Mitarbeiter kommentiert werden.

⁵³ Wikipedia (2014b)

⁵⁴ PTC (2012)

⁵⁵ Vgl. MKS (2009)

⁵⁶ Vgl. Atlassian (2014)

Der Import und Export von Anforderungen ist aus einer Datenbank oder Microsoft Excel möglich, ReqIf hingegen wird nicht unterstützt.

	Basisfunktionalitäten	Erweiterung der Basisfunktionalitäten	Darstellung und Aufbereitung	Unterstützung der Teamarbeit	Schnittstellenfunktionalitäten	Bedienbarkeit	Sicherheit	Herstellereigenschaften
Erfüllung	100%	50%	100%	65%	0%	50%	50%	50%
Erfüllung der Kategorien	67%					50%		

Abb. 19: Bewertung des Tools Jira

4.2.10 Mingle

Die Firma ThoughtWorks besteht seit 1993. Aktuell beschäftigt die Firma mehr als 2500 Mitarbeiter in zwölf verschiedenen Ländern.⁵⁷ In Mingle können User im Zuge des Berechtigungsmanagements mit Rechten ausgestattet werden, die Ihnen die Nutzung bestimmter Toolfunktionen erlaubt oder verbietet. Hierbei unterstützt das Tool ein rollenbasiertes Rechtesystem, das die Rechtevergabe und -verwaltung erleichtert. Importe und Exporte von Anforderungen können ausschließlich aus Microsoft Excel-Dateien gemacht werden. Schnittstellen, um das Tool an andere Tools anzubinden, sind in der aktuellen Version vorhanden.

	Basisfunktionalitäten	Erweiterung der Basisfunktionalitäten	Darstellung und Aufbereitung	Unterstützung der Teamarbeit	Schnittstellenfunktionalitäten	Bedienbarkeit	Sicherheit	Herstellereigenschaften
Erfüllung	100%	33%	100%	75%	33%	100%	50%	25%
Erfüllung der Kategorien	73%					58%		

Abb. 20: Bewertung des Tools Mingle

⁵⁷ Vgl. ThoughtWorks (2014)

4.2.11 aNimble

Das Open Source Tool aNimble ist seit dem Jahr 2010 auf dem Markt. Der Hersteller des Tools ist IdeaStub.⁵⁸ Das Tool besitzt einen grafischen Editor und erlaubt es, Attribute nach den eigenen Anforderungen an Namen zu erstellen. Standardwerte für die Attribute können jedoch nicht hinterlegt werden. An eine modellierte Anforderung können Anhänge angefügt, sowie die Anforderung selbst mit formatierten Texten gestaltet werden. Das Tool unterstützt im Rahmen des Beziehungsmanagements das Verbinden von Anforderungen untereinander und gibt eine Hinweismeldung aus, falls eine vom User durchgeführte Aktion neben der bearbeiteten Anforderung noch eine weitere betrifft.

	Basisfunktionalitäten	Erweiterung der Basisfunktionalitäten	Darstellung und Aufbereitung	Unterstützung der Teamarbeit	Schnittstellenfunktionalitäten	Bedienbarkeit	Sicherheit	Herstellereigenschaften
Erfüllung	100%	50%	75%	55%	0%	100%	0%	0%
Erfüllung der Kategorien	58%					33%		

Abb. 21: Bewertung des Tools aNimble

⁵⁸ Vgl. IdeaStub (2014)

4.2.12 Fazit der Toolbewertungen

Um abschließend einen Überblick über die Ergebnisse der Tools in den verschiedenen Kriterien zu gewinnen und die Tools untereinander vergleichen zu können, dient die folgende Grafik. Dort sind alle geprüften Tools und alle Kriterien je nach Erfüllungsgrad mit Farben hinterlegt. Grün bedeutet voll erfüllt, gelb teilweise erfüllt und rot nicht erfüllt.

Anforderungen	Blueprint R.C.	Rational DOORS	Polarion R.	Enterprise Architect	Caliber	Jama	PTC Integrity	Jira	Mingle	aNimble
Basisdefinition einer Anforderung	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün
Änderung einer Anforderung	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün
Kopieren einer Anforderung	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün
Löschen einer Anforderung	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün
Visuelle Detaillierung einer Anforderung	Grün	Grün	Grün	Grün	Grün	Rot	Rot	Grün	Grün	Rot
Anpassbarkeit der Anforderungsattribute	Grün	Grün	Grün	Rot	Grün	Grün	Grün	Rot	Rot	Grün
Aufbereitung der Anforderungsdefinition	Grün	Grün	Gelb	Rot	Grün	Grün	Grün	Gelb	Rot	Gelb
Sortieren und Gruppieren von Anforderungen	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Rot
Filtern von Anforderungen	Grün	Grün	Grün	Rot	Grün	Grün	Grün	Grün	Grün	Grün
Suchfunktion	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün
Erstellung von Reports	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün
Verteilung von Anforderungen	Grün	Grün	Grün	Rot	Grün	Grün	Grün	Gelb	Gelb	Rot
Ressourcenplanung	Grün	Grün	Grün	Rot	Grün	Grün	Grün	Rot	Rot	Gelb
Priorisierung von Anforderungen	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün
Kommentarfunktion	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün
Nachvollziehbarkeit des Fortschritts	Grün	Rot	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün
Nachvollziehbarkeit einer Änderung	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün
Beziehungsmanagement	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün
Berechtigungsmanagement Ebene Anforderung	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün
Berechtigungsmanagement Ebene Tool	Grün	Grün	Grün	Rot	Grün	Grün	Grün	Rot	Rot	Rot
Mehrbenutzerfähigkeit	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün
Import bestehender Anforderungen	Rot	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün
Export von Anforderungen	Rot	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün
Verknüpfung zu weiteren ALM-Aufgaben	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün
Usability und Erlernbarkeit	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün
Personalisierbarkeit	Rot	Rot	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün
Sicherheit der Daten	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün
Sicherung der Daten	Rot	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün
Herstellerbewertung	Rot	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün
Support	Rot	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün	Grün

Tab. 48: Gesamtüberblick über die Bewertung der Tools

Aus der Grafik ist ersichtlich, dass alle Tools die Basisanforderungen voll erfüllen. Die zweite Kategorie wird von den Tools für große Unternehmen ebenfalls voll erfüllt, mit der Ausnahme von Enterprise Architect, welches kein Tool speziell für das Requirements Engineering ist und somit die spezifischen Kriterien nicht erfüllt. Auch die Tools für kleine Unternehmen sowie das Open Source Produkt erfüllen die Kriterien dieser Kategorie nicht oder nur teilweise.

Die Kriterien der dritten Kategorie werden von allen Tools voll erfüllt, mit Ausnahme von zwei Kriterien, die von Enterprise Architect und aNimble nicht erfüllt werden. In der vierten Kategorie wird ein Großteil der Kriterien voll erfüllt. Die Tools für große und kleine Unternehmen erfüllen hier einige Kriterien nur teilweise. Das Open Source Produkt erfüllt in diesem Bereich die wenigsten Kriterien. Die Kriterien der fünften Kategorie werden von den meisten Tools nicht unterstützt. Grund dafür sind die fehlenden Import- und Exportfunktionalitäten, da die meisten Tools Importe und Exporte nur für Textverarbeitungsprogramme wie Microsoft Word oder Excel ermöglichen und das Format ReqIF, sowie Datenbanken nicht unterstützt werden. Rational DOORS erfüllt diese beiden Anforderungen zu 100%. Ebenso lassen sich wenige Tools personalisieren und bieten keine Sicherungsmöglichkeit, sodass diese manuell vom Administrator gesichert werden müssen. Was der Grafik ebenfalls als Gesamtüberblick entnommen werden kann, sind große Unterschiede zwischen den Tools. So erfüllt beispielsweise das Tool Rational DOORS von IBM viele, aber nicht alle geprüften Kriterien. Ist einem Unternehmen z.B. die Nachvollziehbarkeit des Fortschritts wichtig, bietet sich es sich nicht an, hierfür das Produkt Rational DOORS einzusetzen. Die meisten Kriterien insgesamt erfüllen die Tools Rational DOORS von IBM sowie Caliber von Borland. Weiter kann der Grafik entnommen werden, dass das Open Source Tool aNimble ausschließlich in den Kriterien Schwächen zeigt, in denen die kommerziellen Lösungen ebenfalls Schwächen aufweisen. Bei Kriterien, die von den kommerziellen Tools erfüllt werden, erfüllt aNimble ebenfalls einen Großteil dieser.

Der komplette Kriterienkatalog mit den genauen Erfüllungsgraden aller Tools ist am Ende der Arbeit im Anhang zu finden (siehe Anhang 1).

4.3 Grafische Aufbereitung der Shortlist

In diesem Kapitel werden die Ergebnisse des Kriterienkatalogs grafisch aufbereitet, um eine bessere Vergleichbarkeit der Tools zu ermöglichen. Zusätzlich werden die Auswirkungen einer Gewichtung der Kriterien dargestellt.

4.3.1 Vorgehensweise

Um sich einen gezielten Überblick über die Ergebnisse der jeweiligen Kategorien eines Tools zu verschaffen, dient Abbildung 24. Dabei wird jedes Tool in einer Ergebniszeile abgebildet. Diese setzt sich aus den Erfüllungsgraden der jeweiligen Kategorien und dem Namen des Tools zusammen, wie Abbildung 23 verdeutlicht.

Die Kategorien sind als Kästchen dargestellt, die sich je nach Erfüllungsgrad farblich unterscheiden und den Erfüllungsgrad in Prozent enthalten. So sind z. B. Kategorien, die einen Erfüllungsgrad zwischen 50% und 59% erreichen, gelb markiert. Die Füllfarben der Kästchen

werden in der Excel-Datei mit der Eintragung des jeweiligen Erfüllungsgrades durch eine bedingte Formatierung automatisch angepasst.



Abb. 22: Legende der Erläuterung der Kategoriefarben nach Erfüllungsgrad

Eine in allen acht Kategorien bewertete Ergebniszeile zeigt Abbildung 23.

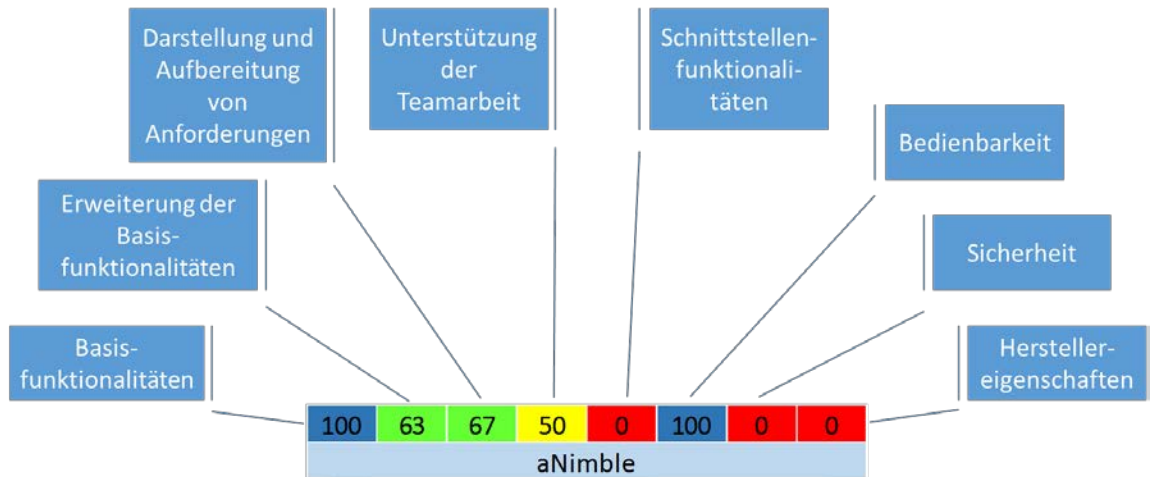


Abb. 23: Beispielhafte Ergebniszeile

Nach diesem Vorgehen werden für die Tools aus der Shortlist die jeweiligen Ergebniszeilen gebildet. Die einzelnen Ergebniszeilen der kommerziellen und der Open Source Tools werden in einer Grafik gegenübergestellt. Die Einteilung auf der vertikalen Ebene erfolgt nach den Unternehmensgrößen. Dadurch entsteht eine Aufteilung in sechs Segmente. Dabei enthält die Platzierung der einzelnen Ergebniszeilen innerhalb eines Segments keine Aussage auf den Erfüllungsgrad der Kategorien.

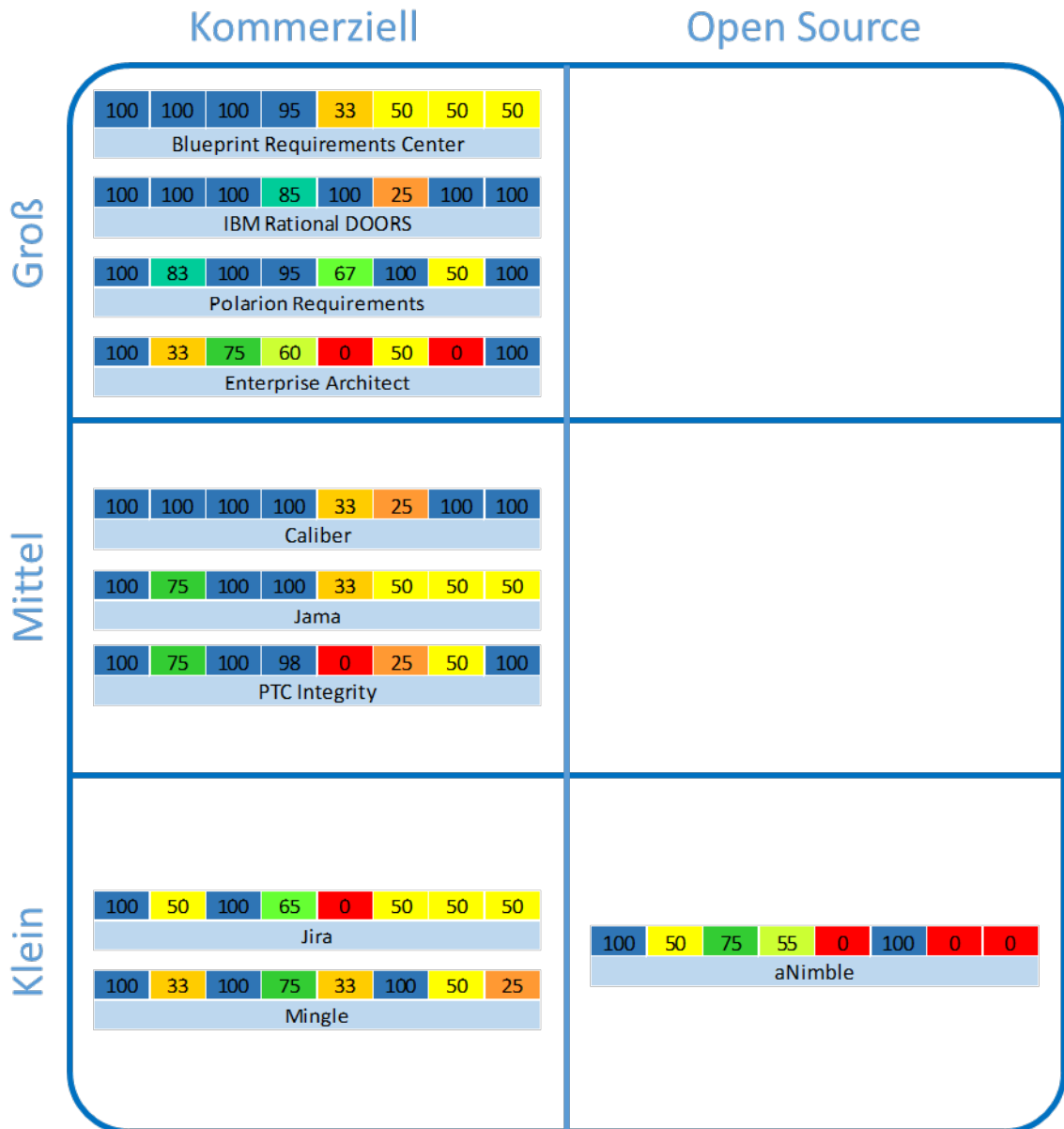


Abb. 24: Ergebnisübersicht der ungewichteten Kategorien

Die Matrixdarstellung der Ergebnisübersicht ermöglicht Unternehmen ein zielgerichtetes Suchen nach einem geeigneten Requirements Engineering Tool. Z.B. kann ein großes Unternehmen, welches Interesse an einem kommerziellen Tool hat, die Ergebniszeilen der Tools innerhalb des Teilbereichs „Groß“ und „Kommerziell“ vergleichen.

Die Tools, die kategorieübergreifend die höchsten Erfüllungsgrade erzielen konnten, sind IBM Rational DOORS und Polarion Requirements. Diese Tools eignen sich für Unternehmen, die eine umfassende Lösung mit vielen Funktionalitäten benötigen. Falls ein Unternehmen besonderen Wert auf die Erfüllung der funktionalen Kriterien legt, stehen mehrere Tools zur Auswahl. Die funktionalen Anforderungen werden von den Tools Blueprint Requirements Center, IBM Rational DOORS, Polarion Requirements, Caliber, Jama und PTC Integrity fast vollständig erfüllt. Eine Ausnahme stellen im Bereich der funktionalen Anforder-

rungen die Schnittstellenfunktionalitäten dar. Hier gibt es große Abweichungen zwischen den Tools. IBM Rational DOORS und Polarion Requirements erfüllen diese Kategorie gut, während Blueprint Requirements Center, Caliber, Jama und PTC Integrity die Funktionalitäten dieser Kategorie nicht ausreichend bieten. Da Enterprise Architect ein Softwaremodellierungstool ist und Requirements Engineering daher nur eine Teilfunktionalität des Funktionsumfangs darstellt, bietet es nicht die Funktionalitäten eines klassischen Requirements Engineering Tools.

Die Tools Jira und Mingle bieten in einigen Kategorien viele Funktionalitäten, im Vergleich zu den oben beschriebenen Tools jedoch mit einem insgesamt geringeren Erfüllungsgrad. Diese Tools können sich dennoch für Unternehmen eignen, für die die gebotenen Funktionalitäten ausreichen.

Falls ein Unternehmen die Kategorien Schnittstellen, Hersteller, und Sicherheit weniger stark gewichtet, ist das Open Source Tool aNimble für den Einsatz in kleinen Unternehmen geeignet, da die übrigen Kategorien ausreichend erfüllt werden.

4.3.2 Beispiel für die Anpassung der Anforderungsgewichtung

Bei der Erläuterung der Anforderungen in Kapitel 3.3 wurde auf Besonderheiten hingewiesen, die bezüglich der Durchführung von Projekten nach dem Vorgehensmodell SCRUM zu beachten sind. Das Kapitel 4.3.2 greift diese Anmerkungen auf und zeigt eine beispielhafte Möglichkeit zur Gewichtung der Kriterien. Neben den Anforderungen, die Anmerkungen zur Anwendung von SCRUM enthalten, werden im Beispiel auch die Anforderungen stärker gewichtet, die im Rahmen dieser Projektarbeit für besonders wichtig erachtet wurden. Das Ziel des Kapitels ist es, den Effekt der Anpassung der Anforderungsgewichtung zu demonstrieren. Tabelle 49 zeigt die dazu angewendete Gewichtung der Anforderungen.

Name der Anforderung	Gewichtung
Basisdefinition einer Anforderung	1
Änderung einer Anforderung	1
Kopieren einer Anforderung	1
Löschen einer Anforderung	1
Visuelle Detaillierung einer Anforderung	1
Anpassbarkeit der Anforderungsattribute	2
Aufbereitung der Anforderungsdefinition	1
Sortieren und Gruppieren von Anforderungen	2
Filtern von Anforderungen	1
Suchfunktion	1
Erstellung von Reports	2
Verteilung von Anforderungen	3
Ressourcenplanung	3
Priorisierung von Anforderungen	3
Kommentarfunktion	1
Nachvollziehbarkeit des Fortschritts	3
Nachvollziehbarkeit einer Änderung	2
Beziehungsmanagement	3

Berechtigungsmanagement Ebene Anforderung	1
Berechtigungsmanagement Ebene Tool	1
Mehrbenutzerfähigkeit	1
Import bestehender Anforderungen	1
Export von Anforderungen	1
Verknüpfung zu weiteren Aufgaben der Softwareentwicklung	1
Usability und Erlernbarkeit	1
Personalisierbarkeit	1
Sicherheit der Daten (Verschlüsselung)	1
Sicherung der Daten	1
Herstellerbewertung	1
Support	1

Tab. 49: Beispielhafte Gewichtung der Anforderungen

Analog der Vorgehensweise in Kapitel 4.3.1, ergibt sich für die gewichteten Kategorien die folgende Ergebnisübersicht:

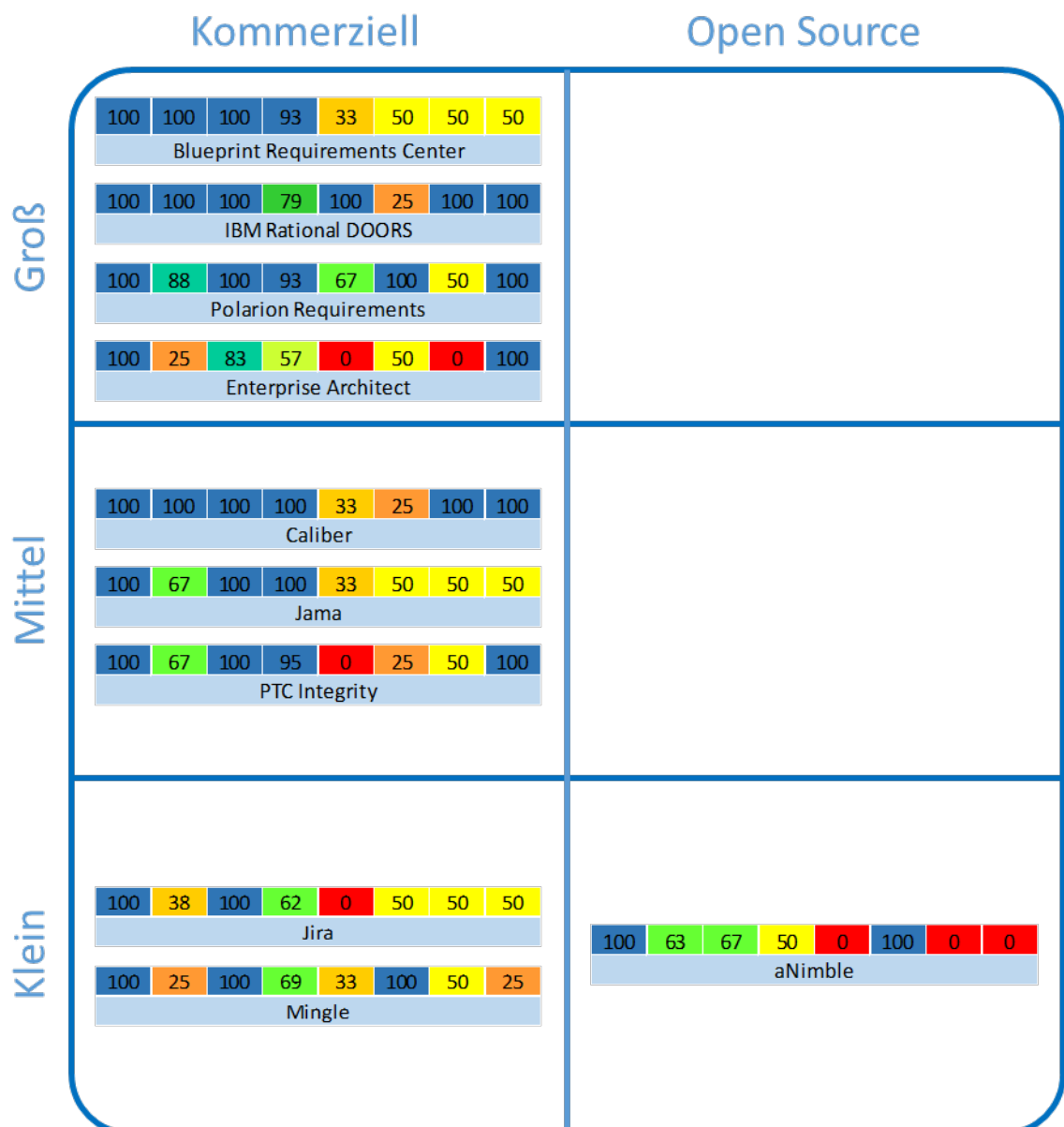


Abb. 25: Ergebnisübersicht der gewichteten Kategorien

Eine höhere Gewichtung der SCRUM-relevanten Anforderungen führt zu veränderten Werten in den Kategorien zwei und vier. Diese verlieren im Vergleich zu der ungewichteten Version der Auswertung an Punkte. Die Farben der Kategorien ändern sich dementsprechend. Einen direkten Vergleich bietet Anhang 3.

4.3.3 Fazit

Die in Kapitel 4.3.1 definierte Ergebniszeile ermöglicht eine Zusammenfassung der Stärken und Schwächen der Tools in den jeweiligen Kategorien. Die anschließende Einordnung der Ergebniszeilen in die Ergebnisübersicht, die in Unternehmensgrößen, sowie in kommerzielle und Open Source-Produkte eingeteilt ist, erlaubt eine zielgerichtete Betrachtung der Tools. Die Darstellungsform gestattet es dem Unternehmen, einzelne Kategorieerfüllungen mehrerer Tools zu vergleichen. Im Gegensatz zu Darstellungen, die ausschließlich eine Gesamtpunktzahl ausweisen, ist ein Kategorienvergleich aussagekräftiger.

Bei einem Vergleich der beiden Ergebnisübersichten (siehe Anhang 3) ist zu sehen, dass die Gewichtung einer oder mehrerer Anforderungen zu unterschiedlichen Ergebnissen führt. Je nach den Bedürfnissen eines Unternehmens, können die Gewichtungen der einzelnen Anforderungen angepasst werden. Dadurch erhält das Unternehmen ein Ergebnis, das seinen spezifischen Anforderungen an den Leistungsumfang eines Requirements Engineering Tools entspricht.

5 Ausblick

Die vorliegende Seminararbeit zeigt die hohe Bedeutung von Requirements Engineering, besonders für den Bereich der teambasierten Abwicklung von Softwareprojekten. Dabei wurde dargelegt, dass sich für eine optimierte Unterstützung des Anforderungsmanagementprozesses der Einsatz von Requirements Engineering Tools anbietet. Aufgrund der hohen Anzahl solcher Werkzeuge wurde der Markt für Anforderungsmanagement-Tools im Rahmen der Seminararbeit strukturiert und segmentiert. Das Ergebnis ist eine Einteilung der Tools gemäß ihrer Eignung für einen Einsatz in kleinen, mittleren oder großen Unternehmen. Ein Unternehmen, das den Einsatz eines Requirement Engineering Tools plant, kann diese Marktsegmentierung als erste Orientierung bei der Auswahl eines geeigneten Werkzeugs heranziehen. Eine Auswahl von in der Literatur und betrieblichen Praxis besonders relevanten Tools wurde anschließend mithilfe eines Kriterienkatalogs bewertet. Dieser wurde in der Seminararbeit eigens dafür entwickelt.

Die folgende Aufzählung zeigt die Realisierung der von Rupp gestellten Ansprüche an die Definition von Anforderungen⁵⁹ durch die in der Seminararbeit entworfene Vorgehensweise:

- **Vollständigkeit:** In der Seminararbeit wurden allgemeingültige und branchenunabhängige Anforderungen berücksichtigt. Der Kriterienkatalog kann jederzeit ohne großen Aufwand um unternehmensspezifische Anforderungen erweitert werden.
- **Korrektheit:** Um Fehler zu vermeiden, wurden die Anforderungsdefinitionen mehrfach überprüft.
- **Konsistenz:** Durch die Begründung einer Anforderung in der Karteikarte, wird die Zielsetzung einer Anforderung ersichtlich und Widersprüche zwischen den Anforderungen werden auf diese Weise vermieden.
- **Nachvollziehbarkeit:** Die Granularität der Abnahmekriterien und die Einsehbarkeit der Entscheidungsregeln sorgen dafür, dass der Erfüllungsgrad einer Anforderung jederzeit nachvollzogen wird.
- **Eindeutigkeit:** Die Kriterienkategorisierung und die detaillierte Definition der Abnahmekriterien ermöglichen die eindeutige Beschreibung jeder Anforderung.
- **Aktualität:** Durch die Anpassungsmöglichkeiten des Kriterienkatalogs kann jederzeit auf Änderungen und aktuelle Entwicklungen reagiert werden.
- **Bewertbarkeit:** Die individuelle Anpassbarkeit der Gewichtung einer Anforderung ermöglicht die Bewertung.

⁵⁹ Vgl. Rupp, C. (2007), S. 28 ff.

Die Aufzählung zeigt, dass die Vorgehensweise zur Erstellung der Marktübersicht die Anforderungen von Rupp besonders durch die entworfene Karteikarte in einem hohen Maß erfüllt. Für die effiziente Durchführung der Toolbewertungen wurde im Rahmen der Seminararbeit eine Excel-Datei entwickelt, die ein Unternehmen für die Auswahl eines Requirements Engineering Tools verwenden kann.

Eine Besonderheit des Kriterienkatalogs ist dessen Erweiterbarkeit um unternehmensspezifische Anforderungen und Abnahmekriterien, die ohne großen Aufwand in die Excel-Datei aufgenommen werden können. Daneben kann in der Excel-Datei die Gewichtung der Anforderungen individuell festgelegt werden. Nach der Anpassung der Excel-Datei kann ein Unternehmen eine auf seine Bedürfnisse abgestimmte Marktübersicht erstellen. Die Möglichkeiten der Anpassbarkeit des Kriterienkatalogs sollten von Unternehmen genutzt werden, da durch die Berücksichtigung individueller und unternehmensspezifischer Bedürfnisse die Ergebnisse der Marktübersicht zum Teil stark beeinflusst werden können. Um einen branchen- und unternehmensunabhängigen Überblick für das Marktsegment der Anforderungsmanagement-Tools zu erhalten, kann ein Unternehmen die Ergebnisübersicht aus Kapitel 4.3.1 heranziehen. Auch die Ergebnisübersicht lässt sich einfach um weitere Tools erweitern.

Zusammenfassend unterstützt die im Rahmen der Seminararbeit entwickelte Vorgehensweise Unternehmen bei der Erstellung spezifischer Marktübersichten, da sie individuelle Gewichtungen, sowie die Erweiterung des Kriterienkatalogs um Anforderungen, Abnahmekriterien und Tools zulässt. Gleichzeitig gibt sie einen Rahmen für die Bewertung vor, sodass u.a. spezifische Kriterien des Requirements Engineerings bereits automatisch berücksichtigt werden.

Die durchgeführte Marktanalyse zeigt, dass sich Open Source Tools für das Anforderungsmanagement aktuell überwiegend für einen Einsatz in einem kleinen Unternehmen eignen. Bei entsprechender Weiterentwicklung des Tools aNimble ist dessen Verwendung in einem mittleren Unternehmen denkbar. Für den Einsatz eines Open Source Requirements Engineering Tools in einem großen Unternehmen besteht aktuell noch großer Handlungsbedarf. Dies hängt weniger mit den funktionalen Defiziten der Tools zusammen, sondern vielmehr mit dem zu geringen Erfüllungsgrad der nichtfunktionalen Anforderungen bei den meisten Open Source Tools für das Anforderungsmanagement. So fehlen bspw. entsprechende Dienstleister, die ein großes Unternehmen bei der Einführung und Anpassung eines Open Source Tools unterstützen.

Aufgrund der hohen Bedeutung des Requirements Engineerings zur Vermeidung von Fehlern in der Phase der Produktplanung besteht für die Anbieter von Open Source Software in diesem Marktsegment viel Potenzial. Die Entwicklung eines leistungsstarken Open Source Anforderungsmanagement-Tools und das Anbieten ergänzender Dienstleistungen sind in diesem Wettbewerbsumfeld daher potenzielle Differenzierungsmöglichkeiten.

Anhang

Anhang 1: Ungewichtete Toolbewertungen	66
Anhang 2: Gewichtete Toolbewertungen.....	70
Anhang 3: Vergleich der Ergebnisübersichten (ungewichtet und gewichtet)	74

Anhang 1: Ungewichtete Toolbewertungen

Kriterien	Gewichtung des Kriteriums	Abnahme-kriterien	Blueprint	IBM Doors	Polarion	Caliber	Jama
Basisfunktionalitäten							
Basisdefinition einer Anforderung	1						
		1.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Änderung einer Anforderung	1						
		1.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kopieren einer Anforderung	1						
		1.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Löschen einer Anforderung	1						
		1	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kategorienbewertung			100%	100%	100%	100%	100%
Erweiterung der Basisfunktionalitäten							
Visuelle Detaillierung einer Anforderung	1						
		1.	J	J	J	J	N
		2.	J	J	J	J	N
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	nicht erfüllt
Anpassbarkeit der Anforderungsattribute	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	N	J	J	J
		4.	J	J	J	J	J
		5.	N	N	N	J	N
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Aufbereitung der Anforderungsdefinition	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	J	N	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	teilw. erfüllt	voll erfüllt	voll erfüllt
Kategorienbewertung			100%	100%	83%	100%	67%
Darstellung und Aufbereitung von Anforderungen							
Sortieren und Gruppieren von Anforderungen	1						
		1.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Filtern von Anforderungen	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Suchfunktion	1						
		1.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Erstellung von Reports	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	N	N	J	N
		4.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kategorienbewertung			100%	100%	100%	100%	100%
Unterstützung der Teamarbeit							
Verteilung von Anforderungen	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Ressourcenplanung	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	J	J	J	J
		4.	N	N	J	J	N
		5.	N	N	J	N	N
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Priorisierung von Anforderungen	1						
		1.	J	J	J	J	J
		2.	J	N	J	J	J
		3.	N	N	J	J	J
Erfüllungsgrad			voll erfüllt	teilw. erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kommentarfunktion	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	J	J	N	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Nachvollziehbarkeit des Fortschritts	1						
		1.	J	N	J	J	J
		2.	J	N	J	J	J
		3.	N	N	N	J	J
		4.	N	N	N	J	J
Erfüllungsgrad			teilw. erfüllt	nicht erfüllt	teilw. erfüllt	voll erfüllt	voll erfüllt

Kriterien	Gewichtung des Kriteriums	Abnahme-kriterien	Blueprint	IBM Doors	Polarion	Caliber	Jama
Basisfunktionalitäten							
Nachvollziehbarkeit einer Änderung	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	J	J	J	J
		4.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Beziehungsmanagement	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	N	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Berechtigungsmanagement Ebene Anforderung	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Berechtigungsmanagement Ebene Tool	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Mehrbenutzerfähigkeit	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kategorienbewertung			95%	85%	95%	100%	100%
Schnittstellenfunktionalitäten							
Import bestehender Anforderungen	1						
		1.	J	J	J	J	J
		2.	N	J	J	N	N
		3.	N	J	N	N	N
Erfüllungsgrad			nicht erfüllt	voll erfüllt	teilw. erfüllt	nicht erfüllt	nicht erfüllt
Export von Anforderungen	1						
		1.	N	J	J	N	N
		2.	J	J	J	J	J
		3.	J	J	N	N	N
Erfüllungsgrad			nicht erfüllt	voll erfüllt	teilw. erfüllt	nicht erfüllt	nicht erfüllt
Verknüpfung zu weiteren ALM-Aufgaben	1						
		1.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kategorienbewertung			33%	100%	67%	33%	33%
			90%	94%	92%	92%	88%
Bedienbarkeit							
Usability und Erlernbarkeit	1						
		1.	J	J	J	J	J
		2.	J	N	J	N	J
Erfüllungsgrad			voll erfüllt	teilw. erfüllt	voll erfüllt	teilw. erfüllt	voll erfüllt
Personalisierbarkeit	1						
		1.	N	N	J	N	N
		2.	J	J	J	J	J
Erfüllungsgrad			nicht erfüllt	nicht erfüllt	voll erfüllt	nicht erfüllt	nicht erfüllt
Kategorienbewertung			50%	25%	100%	25%	50%
Sicherheit							
Sicherheit der Daten	1						
		1.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Sicherung der Daten	1						
		1.	N	J	N	J	N
Erfüllungsgrad			nicht erfüllt	voll erfüllt	nicht erfüllt	voll erfüllt	nicht erfüllt
Kategorienbewertung			50%	100%	50%	100%	50%
Herstellereigenschaften							
Herstellerbewertung	1						
		1.	N	J	J	J	N
		2.	J	J	J	J	N
Erfüllungsgrad			nicht erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	nicht erfüllt
Support	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kategorienbewertung			50%	100%	100%	100%	50%
			50%	75%	83%	75%	50%

Kriterien	Gewichtung des Kriteriums	Abnahme-kriterien	Integrity	Jira	Mingle	Enterprise Architect	aNimble
Basisfunktionalitäten							
Basisdefinition einer Anforderung	1						
		1.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Änderung einer Anforderung	1						
		1.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kopieren einer Anforderung	1						
		1.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Löschen einer Anforderung	1						
		1	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kategorienbewertung			100%	100%	100%	100%	100%
Erweiterung der Basisfunktionalitäten							
Visuelle Detaillierung einer Anforderung	1						
		1.	N	J	J	J	J
		2.	N	J	J	J	N
Erfüllungsgrad			nicht erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	nicht erfüllt
Anpassbarkeit der Anforderungsattribute	1						
		1.	J	N	N	N	J
		2.	J	N	N	N	J
		3.	J	N	N	N	J
		4.	J	N	N	N	N
		5.	J	N	N	N	N
Erfüllungsgrad			voll erfüllt	nicht erfüllt	nicht erfüllt	nicht erfüllt	voll erfüllt
Aufbereitung der Anforderungsdefinition	1						
		1.	J	N	N	N	J
		2.	J	J	N	N	J
		3.	J	N	N	N	N
Erfüllungsgrad			voll erfüllt	teilw. erfüllt	nicht erfüllt	nicht erfüllt	teilw. erfüllt
Kategorienbewertung			67%	50%	33%	33%	50%
Darstellung und Aufbereitung von Anforderungen							
Sortieren und Gruppieren von Anforderungen	1						
		1.	J	J	J	J	N
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	nicht erfüllt
Filtern von Anforderungen	1						
		1.	J	J	J	N	J
		2.	J	J	J	N	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	nicht erfüllt	voll erfüllt
Suchfunktion	1						
		1.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Erstellung von Reports	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	J	J	J	J
		4.	J	J	N	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kategorienbewertung			100%	100%	100%	75%	75%
Unterstützung der Teamarbeit							
Verteilung von Anforderungen	1						
		1.	J	J	J	N	N
		2.	J	N	N	N	N
Erfüllungsgrad			voll erfüllt	teilw. erfüllt	teilw. erfüllt	nicht erfüllt	nicht erfüllt
Ressourcenplanung	1						
		1.	J	N	N	N	J
		2.	J	N	N	N	N
		3.	J	N	N	N	J
		4.	J	N	N	N	N
		5.	J	N	N	N	N
Erfüllungsgrad			voll erfüllt	nicht erfüllt	nicht erfüllt	nicht erfüllt	teilw. erfüllt
Priorisierung von Anforderungen	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	N	N	N	N
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kommentarfunktion	1						
		1.	J	J	J	J	J
		2.	N	J	J	N	J
		3.	N	N	J	N	N
Erfüllungsgrad			teilw. erfüllt	voll erfüllt	voll erfüllt	teilw. erfüllt	voll erfüllt
Nachvollziehbarkeit des Fortschritts	1						
		1.	J	J	J	J	N
		2.	J	J	J	J	N
		3.	J	N	N	N	N
		4.	J	N	N	N	N
Erfüllungsgrad			voll erfüllt	teilw. erfüllt	teilw. erfüllt	teilw. erfüllt	nicht erfüllt

Kriterien	Gewichtung des Kriteriums	Abnahme-kriterien	Integrity	Jira	Mingle	Enterprise Architect	aNimble
Basisfunktionalitäten							
Nachvollziehbarkeit einer Änderung	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	J	J	J	N
		4.	J	N	J	J	N
Erfüllungsgrad			voll erfüllt	teilw. erfüllt	voll erfüllt	voll erfüllt	nicht erfüllt
Beziehungsmanagement	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	N	N	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Berechtigungsmanagement Ebene Anforderung	1						
		1.	J	J	J	J	J
		2.	J	J	N	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	teilw. erfüllt	voll erfüllt	voll erfüllt
Berechtigungsmanagement Ebene Tool	1						
		1.	J	N	J	N	N
		2.	J	N	J	N	N
		3.	J	N	N	N	N
Erfüllungsgrad			voll erfüllt	nicht erfüllt	voll erfüllt	nicht erfüllt	nicht erfüllt
Mehrbenutzerfähigkeit	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kategorienbewertung			95%	65%	75%	60%	55%
Schnittstellenfunktionalitäten							
Import bestehender Anforderungen	1						
		1.	N	J	J	J	J
		2.	N	N	N	N	N
		3.	N	J	N	N	N
Erfüllungsgrad			nicht erfüllt	nicht erfüllt	nicht erfüllt	nicht erfüllt	nicht erfüllt
Export von Anforderungen	1						
		1.	N	N	J	J	N
		2.	N	J	N	N	J
		3.	N	J	N	N	N
Erfüllungsgrad			nicht erfüllt	nicht erfüllt	nicht erfüllt	nicht erfüllt	nicht erfüllt
Verknüpfung zu weiteren ALM-Aufgaben	1						
		1.	N	N	J	N	N
Erfüllungsgrad			nicht erfüllt	nicht erfüllt	voll erfüllt	nicht erfüllt	nicht erfüllt
Kategorienbewertung			0%	0%	33%	0%	0%
			81%	67%	73%	58%	58%
Bedienbarkeit							
Usability und Erlernbarkeit	1						
		1.	J	J	J	J	J
		2.	N	J	J	J	J
Erfüllungsgrad			teilw. erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Personalisierbarkeit	1						
		1.	N	N	J	N	J
		2.	J	J	J	J	J
Erfüllungsgrad			nicht erfüllt	nicht erfüllt	voll erfüllt	nicht erfüllt	voll erfüllt
Kategorienbewertung			25%	50%	100%	50%	100%
Sicherheit							
Sicherheit der Daten	1						
		1.	J	N	N	N	N
Erfüllungsgrad			voll erfüllt	nicht erfüllt	nicht erfüllt	nicht erfüllt	nicht erfüllt
Sicherung der Daten	1						
		1.	N	J	J	N	N
Erfüllungsgrad			nicht erfüllt	voll erfüllt	voll erfüllt	nicht erfüllt	nicht erfüllt
Kategorienbewertung			50%	50%	50%	0%	0%
Herstellereigenschaften							
Herstellerbewertung	1						
		1.	J	N	J	J	N
		2.	J	N	N	J	N
Erfüllungsgrad			voll erfüllt	nicht erfüllt	teilw. erfüllt	voll erfüllt	nicht erfüllt
Support	1						
		1.	J	J	J	J	J
		2.	J	J	N	J	N
		3.	J	J	N	J	N
Erfüllungsgrad			voll erfüllt	voll erfüllt	nicht erfüllt	voll erfüllt	nicht erfüllt
Kategorienbewertung			100%	50%	25%	100%	0%
			58%	50%	58%	50%	33%

Anhang 2: Gewichtete Toolbewertungen

Kriterien	Gewichtung des Kriteriums	Abnahme-kriterien	Blueprint	IBM Doors	Polarion	Caliber	Jama
Basisfunktionalitäten							
Basisdefinition einer Anforderung	1						
		1.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Änderung einer Anforderung	1						
		1.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kopieren einer Anforderung	1						
		1.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Löschen einer Anforderung	1						
		1	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kategorienbewertung			100%	100%	100%	100%	100%
Erweiterung der Basisfunktionalitäten							
Visuelle Detaillierung einer Anforderung	1						
		1.	J	J	J	J	N
		2.	J	J	J	J	N
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	nicht erfüllt
Anpassbarkeit der Anforderungsattribute	2						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	N	J	J	J
		4.	J	J	J	J	J
		5.	N	N	N	J	N
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Aufbereitung der Anforderungsdefinition	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	J	N	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	teilw. erfüllt	voll erfüllt	voll erfüllt
Kategorienbewertung			100%	100%	88%	100%	75%
Darstellung und Aufbereitung von Anforderungen							
Sortieren und Gruppieren von Anforderungen	2						
		1.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Filtern von Anforderungen	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Suchfunktion	1						
		1.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Erstellung von Reports	2						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	N	N	J	N
		4.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kategorienbewertung			100%	100%	100%	100%	100%
Unterstützung der Teamarbeit							
Verteilung von Anforderungen	3						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Ressourcenplanung	3						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	J	J	J	J
		4.	J	N	J	J	N
		5.	J	N	J	N	N
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Priorisierung von Anforderungen	3						
		1.	J	J	J	J	J
		2.	J	N	J	J	J
		3.	N	N	J	J	J
Erfüllungsgrad			voll erfüllt	teilw. erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kommentarfunktion	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	J	J	N	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Nachvollziehbarkeit des Fortschritts	3						
		1.	J	N	J	J	J
		2.	J	N	J	J	J
		3.	N	N	N	J	J
		4.	N	N	N	J	J
Erfüllungsgrad			teilw. erfüllt	nicht erfüllt	teilw. erfüllt	voll erfüllt	voll erfüllt

Kriterien	Gewichtung des Kriteriums	Abnahme-kriterien	Blueprint	IBM Doors	Polarion	Caliber	Jama
Basisfunktionalitäten							
Nachvollziehbarkeit einer Änderung	2						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	J	J	J	J
		4.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Beziehungsmanagement	3						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	N	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Berechtigungsmanagement Ebene Anforderung	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Berechtigungsmanagement Ebene Tool	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Mehrbenutzerfähigkeit	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kategorienbewertung			93%	79%	93%	100%	100%
Schnittstellenfunktionalitäten							
Import bestehender Anforderungen	1						
		1.	J	J	J	J	J
		2.	N	J	J	N	N
		3.	N	J	N	N	N
Erfüllungsgrad			nicht erfüllt	voll erfüllt	teilw. erfüllt	nicht erfüllt	nicht erfüllt
Export von Anforderungen	1						
		1.	N	J	J	N	N
		2.	J	J	J	J	J
		3.	J	J	N	N	N
Erfüllungsgrad			nicht erfüllt	voll erfüllt	teilw. erfüllt	nicht erfüllt	nicht erfüllt
Verknüpfung zu weiteren ALM-Aufgaben	1						
		1.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kategorienbewertung			33%	100%	67%	33%	33%
			91%	88%	92%	95%	92%
Bedienbarkeit							
Usability und Erlernbarkeit	1						
		1.	J	J	J	J	J
		2.	J	N	J	N	J
Erfüllungsgrad			voll erfüllt	teilw. erfüllt	voll erfüllt	teilw. erfüllt	voll erfüllt
Personalisierbarkeit	1						
		1.	N	N	J	N	N
		2.	J	J	J	J	J
Erfüllungsgrad			nicht erfüllt	nicht erfüllt	voll erfüllt	nicht erfüllt	nicht erfüllt
Kategorienbewertung			50%	25%	100%	25%	50%
Sicherheit							
Sicherheit der Daten	1						
		1.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Sicherung der Daten	1						
		1.	N	J	N	J	N
Erfüllungsgrad			nicht erfüllt	voll erfüllt	nicht erfüllt	voll erfüllt	nicht erfüllt
Kategorienbewertung			50%	100%	50%	100%	50%
Herstellereigenschaften							
Herstellerbewertung	1						
		1.	N	J	J	J	N
		2.	J	J	J	J	N
Erfüllungsgrad			nicht erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	nicht erfüllt
Support	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kategorienbewertung			50%	100%	100%	100%	50%
			50%	75%	83%	75%	50%

Kriterien	Gewichtung des Kriteriums	Abnahme-kriterien	Integrity	Jira	Mingle	Enterprise Architect	aNimble
Basisfunktionalitäten							
Basisdefinition einer Anforderung	1						
		1.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Änderung einer Anforderung	1						
		1.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kopieren einer Anforderung	1						
		1.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Löschen einer Anforderung	1						
		1	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kategorienbewertung			100%	100%	100%	100%	100%
Erweiterung der Basisfunktionalitäten							
Visuelle Detaillierung einer Anforderung	1						
		1.	N	J	J	J	J
		2.	N	J	J	J	N
Erfüllungsgrad			nicht erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	nicht erfüllt
Anpassbarkeit der Anforderungsattribute	2						
		1.	J	N	N	N	J
		2.	J	N	N	N	J
		3.	J	N	N	N	J
		4.	J	N	N	N	N
		5.	J	N	N	N	N
Erfüllungsgrad			voll erfüllt	nicht erfüllt	nicht erfüllt	nicht erfüllt	voll erfüllt
Aufbereitung der Anforderungsdefinition	1						
		1.	J	N	N	N	J
		2.	J	J	N	N	J
		3.	J	N	N	N	N
Erfüllungsgrad			voll erfüllt	teilw. erfüllt	nicht erfüllt	nicht erfüllt	teilw. erfüllt
Kategorienbewertung			75%	38%	25%	25%	63%
Darstellung und Aufbereitung von Anforderungen							
Sortieren und Gruppieren von Anforderungen	2						
		1.	J	J	J	J	N
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	nicht erfüllt
Filtern von Anforderungen	1						
		1.	J	J	J	N	J
		2.	J	J	J	N	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	nicht erfüllt	voll erfüllt
Suchfunktion	1						
		1.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Erstellung von Reports	2						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	J	J	J	J
		4.	J	J	N	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kategorienbewertung			100%	100%	100%	83%	67%
Unterstützung der Teamarbeit							
Verteilung von Anforderungen	3						
		1.	J	J	J	N	N
		2.	J	N	N	N	N
Erfüllungsgrad			voll erfüllt	teilw. erfüllt	teilw. erfüllt	nicht erfüllt	nicht erfüllt
Ressourcenplanung	3						
		1.	J	N	N	N	J
		2.	J	N	N	N	N
		3.	J	N	N	N	J
		4.	J	N	N	N	N
		5.	J	N	N	N	N
Erfüllungsgrad			voll erfüllt	nicht erfüllt	nicht erfüllt	nicht erfüllt	teilw. erfüllt
Priorisierung von Anforderungen	3						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	N	N	N	N
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kommentarfunktion	1						
		1.	J	J	J	J	J
		2.	N	J	J	N	J
		3.	N	N	J	N	N
Erfüllungsgrad			teilw. erfüllt	voll erfüllt	voll erfüllt	teilw. erfüllt	voll erfüllt
Nachvollziehbarkeit des Fortschritts	3						
		1.	J	J	J	J	N
		2.	J	J	J	J	N
		3.	J	N	N	N	N
		4.	J	N	N	N	N
Erfüllungsgrad			voll erfüllt	teilw. erfüllt	teilw. erfüllt	teilw. erfüllt	nicht erfüllt

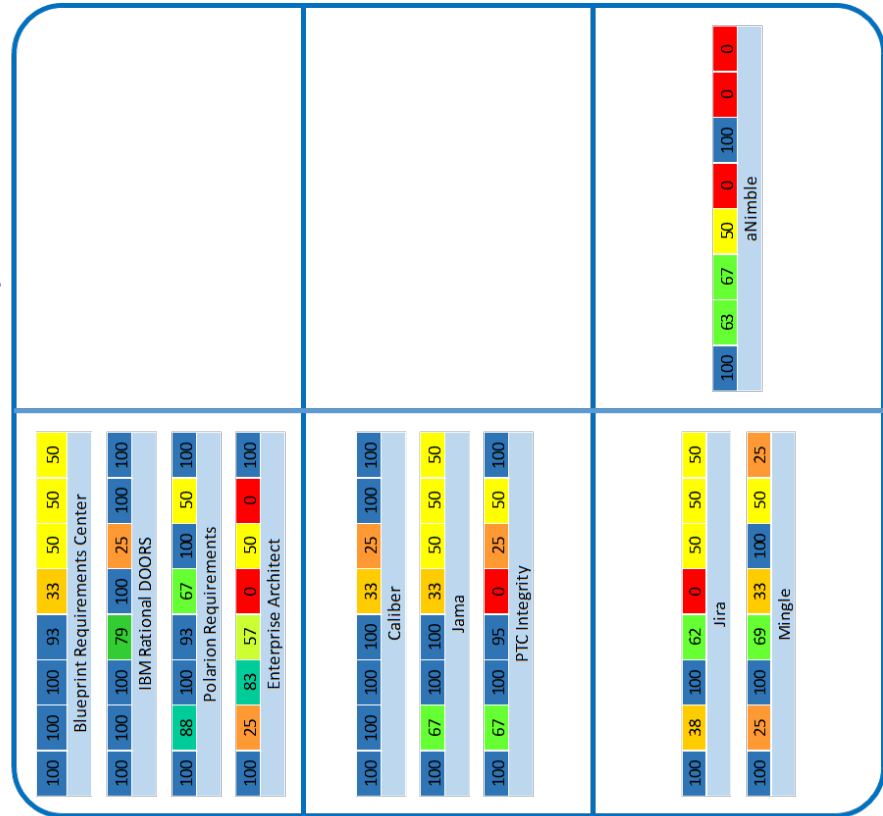
Kriterien	Gewichtung des Kriteriums	Abnahme-kriterien	Integrity	Jira	Mingle	Enterprise Architect	aNimble
Basisfunktionalitäten							
Nachvollziehbarkeit einer Änderung	2						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	J	J	J	N
		4.	J	N	J	J	N
Erfüllungsgrad			voll erfüllt	teilw. erfüllt	voll erfüllt	voll erfüllt	nicht erfüllt
Beziehungsmanagement	3						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
		3.	J	N	N	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Berechtigungsmanagement Ebene Anforderung	1						
		1.	J	J	J	J	J
		2.	J	J	N	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	teilw. erfüllt	voll erfüllt	voll erfüllt
Berechtigungsmanagement Ebene Tool	1						
		1.	J	N	J	N	N
		2.	J	N	J	N	N
		3.	J	N	N	N	N
Erfüllungsgrad			voll erfüllt	nicht erfüllt	voll erfüllt	nicht erfüllt	nicht erfüllt
Mehrbenutzerfähigkeit	1						
		1.	J	J	J	J	J
		2.	J	J	J	J	J
Erfüllungsgrad			voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Kategorienbewertung			98%	62%	69%	57%	50%
Schnittstellenfunktionalitäten							
Import bestehender Anforderungen	1						
		1.	N	J	J	J	J
		2.	N	N	N	N	N
		3.	N	J	N	N	N
Erfüllungsgrad			nicht erfüllt	nicht erfüllt	nicht erfüllt	nicht erfüllt	nicht erfüllt
Export von Anforderungen	1						
		1.	N	N	J	J	N
		2.	N	J	N	N	J
		3.	N	J	N	N	N
Erfüllungsgrad			nicht erfüllt	nicht erfüllt	nicht erfüllt	nicht erfüllt	nicht erfüllt
Verknüpfung zu weiteren ALM-Aufgaben	1						
		1.	N	N	J	N	N
Erfüllungsgrad			nicht erfüllt	nicht erfüllt	voll erfüllt	nicht erfüllt	nicht erfüllt
Kategorienbewertung			0%	0%	33%	0%	0%
			88%	64%	70%	58%	55%
Bedienbarkeit							
Usability und Erlernbarkeit	1						
		1.	J	J	J	J	J
		2.	N	J	J	J	J
Erfüllungsgrad			teilw. erfüllt	voll erfüllt	voll erfüllt	voll erfüllt	voll erfüllt
Personalisierbarkeit	1						
		1.	N	N	J	N	J
		2.	J	J	J	J	J
Erfüllungsgrad			nicht erfüllt	nicht erfüllt	voll erfüllt	nicht erfüllt	voll erfüllt
Kategorienbewertung			25%	50%	100%	50%	100%
Sicherheit							
Sicherheit der Daten	1						
		1.	J	N	N	N	N
Erfüllungsgrad			voll erfüllt	nicht erfüllt	nicht erfüllt	nicht erfüllt	nicht erfüllt
Sicherung der Daten	1						
		1.	N	J	J	N	N
Erfüllungsgrad			nicht erfüllt	voll erfüllt	voll erfüllt	nicht erfüllt	nicht erfüllt
Kategorienbewertung			50%	50%	50%	0%	0%
Herstellereigenschaften							
Herstellerbewertung	1						
		1.	J	N	J	J	N
		2.	J	N	N	J	N
Erfüllungsgrad			voll erfüllt	nicht erfüllt	teilw. erfüllt	voll erfüllt	nicht erfüllt
Support	1						
		1.	J	J	J	J	J
		2.	J	J	N	J	N
		3.	J	J	N	J	N
Erfüllungsgrad			voll erfüllt	voll erfüllt	nicht erfüllt	voll erfüllt	nicht erfüllt
Kategorienbewertung			100%	50%	25%	100%	0%
			58%	50%	58%	50%	33%

Anhang 3: Vergleich der Ergebnisübersichten (ungewichtet und gewichtet)

Gewichtet

Open Source

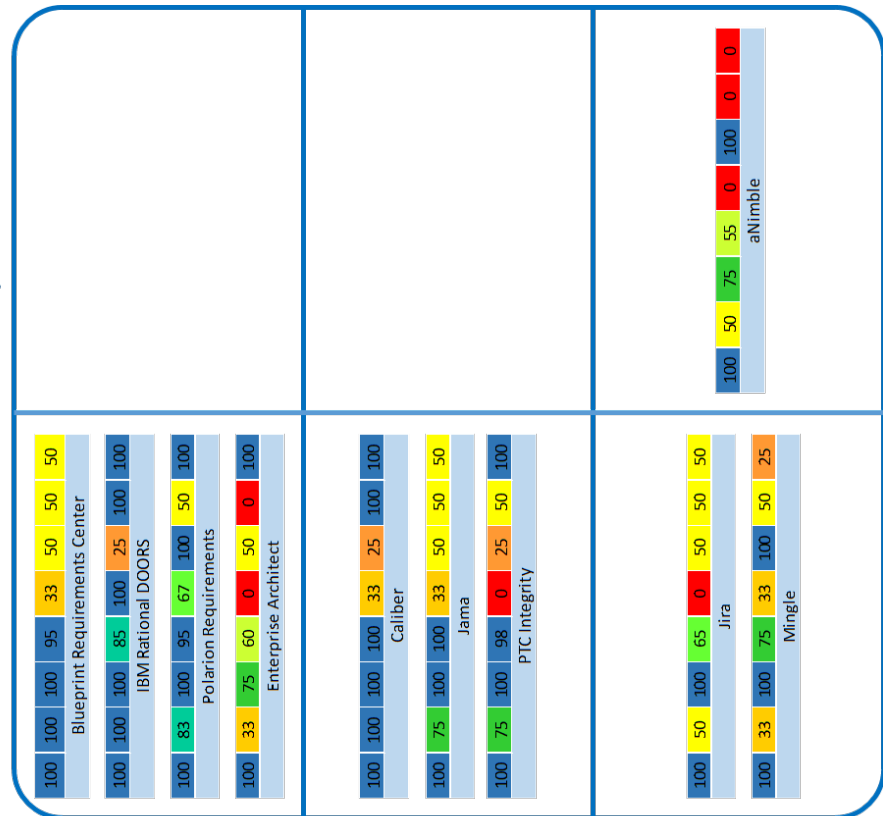
Kommerziell



Ungewichtet

Open Source

Kommerziell



Quellenverzeichnisse

Literaturverzeichnis

- Balzert, H. (2009): Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering, 3. Aufl., Heidelberg: Spektrum Akademischer Verlag
- Beatty, J. / Ferrari, R. (2011): How to Evaluate and Select a Requirements Management Tool, Austin: Seilevel
- Beatty, J. u.a. (o.J.): Seilevel's Evaluations of Requirements Management Tools: Summaries and Scores, Austin: Seilevel
- Ebert, C. (2005): Systematisches Requirements Management – Anforderungen ermitteln, spezifizieren, analysieren und verfolgen, Heidelberg: dpunkt Verlag
- Fahney, R./Herrmann, A. (2013): Requirements Engineering und Projektmanagement, (Hrsg.: Herrmann, A./Knauss, E./Weißbach, R.), Heidelberg: Springer Verlag
- Grande, M. (2011): 100 Minuten für Anforderungsmanagement – Kompaktes Wissen nicht nur für Projektleiter und Entwickler, Wiesbaden: Vieweg + Teubner Verlag
- Partsch, H. (1991): Requirements Engineering, München: Oldenbourg Verlag
- Partsch, H. (2010): Requirements-Engineering systematisch – Modellbildung für softwaregestützte Systeme, 2. Aufl., Heidelberg: Springer Verlag
- Polarion Software (2014): Polarion User Guide, Stuttgart / San Francisco: Polarion Software
- Rupp, C. (2007): Requirementsengineering und Management – Professionelle, iterative Anforderungsanalyse für die Praxis, 4. Aufl., München: Carl Hanser Verlag

Verzeichnis der Internet- und Intranet-Quellen

- Atlassian (2014): About Us,
<https://www.atlassian.com/company>,
Abruf: 29.06.2014
- Black, D. / Thomas, J. (2013): How Gartner Evaluates Vendors and Markets in Magic Quadrants and MarketScopes,
<https://www.gartner.com/doc/2560415>,
Abruf: 20.06.2014
- Blueprint (2011): Requirements Center 2010 FP3 Help,
http://files.blueprintsys.com/downloads/RC2010SR6/RC2010SR6_Help.pdf,
Abruf: 27.06.2014
- Blueprint (o.J.): Customers,
<http://www.blueprintsys.com/customers/>,
Abruf: 27.06.2014
- Borland (2004): User Guide, CaliberRM,
http://techpubs.borland.com/caliber/caliber_rm/2005/en/CaliberRMUserGuide.pdf,
Abruf: 27.06.2014
- EA AddOn (2014): Third Party Extensions for Enterprise Architect,
<http://www.sparxsystems.com/products/3rdparty.html>,
Abruf: 29.06.2014
- Forrester (o.J.): The Forrester Wave™ Methodology Guide,
<http://www.forrester.com/marketing/policies/forrester-wave-methodology.html>,
Abruf: 21.06.2014
- Gartner (o.J.): Magic Quadrants,
http://www.gartner.com/technology/research/methodologies/research_mq.jsp,
Abruf: 20.06.2014
- Glassdoor (o.J.): Blueprint Software Systems,
http://www.glassdoor.com/Overview/Working-at-Blueprint-Software-Systems-EI_IE465048.11,37.htm,
Abruf: 27.06.2014

- IBM (2010): IBM Rational DOORS Using Rational DOORS,
http://publib.boulder.ibm.com/infocenter/rsdp/v1r0m0/topic/com.ibm.help.download.doors.doc/pdf92/using_doors.pdf,
Abruf: 27.06.2014
- IdeaStub (2014): Company,
<http://www.ideastub.com/company.php>,
Abruf: 29.06.2014
- Inc. (o.J.): Jama Software,
<http://www.inc.com/profile/jama-software>,
Abruf: 27.06.2014
- inteGREAT (o.J.): Requirement Definition with inteGREAT,
<http://www.edevtech.com/videos/requirements-definition/Requirement-Definition-with-inteGREAT-TFS.htm>,
Abruf: 27.06.2014
- Jama Software (2014): Jama User Guide,
<http://help.jamasoftware.com/>,
Abruf: 27.06.2014
- MKS (2009): Integrity, User Guide,
<http://www.mks.com/support/productinfo/docs/2009IntegrityUserGuide.pdf>,
Abruf: 27.06.2014
- Object Management Group (o.J.): Documents associated with Requirements Interchange Format (ReqIF) 1.1,
<http://www.omg.org/spec/ReqIF/1.1/>,
Abruf: 24.06.2014
- PTC (2012): Die Integrity Plattform,
http://de.ptc.com/WCMS/files/138766/de/7217_Integrity_Plattform_DS_DE.pdf,
Abruf: 27.06.2014
- TDWG (2010): INCOSE Requirements Management Tools Survey,
<http://www.incose.org/productspubs/products/rmsurvey.aspx>,
Abruf: 15.06.2014

- The Making of Software (2014a): The Making of Software,
<http://makingofsoftware.com/>,
Abruf: 15.06.2014
- The Making of Software (2014b): List of Requirements Management Tools,
<http://makingofsoftware.com/resources/list-of-rm-tools>,
Abruf: 15.06.2014
- ThoughtWorks (2014): Wir sind ThoughtWorks,
<http://www.thoughtworks.com/de/about-us>,
Abruf: 29.06.2014
- VOLERE (2014): Requirements Tools,
<http://www.VOLERE.co.uk/tools.htm#>,
Abruf: 15.06.2014
- Wikipedia (2014a): Anforderungsmanagement-Software,
<http://de.wikipedia.org/wiki/Anforderungsmanagement-Software>,
Abruf: 15.06.2014
- Wikipedia (2014b): Parametric Technology Corporation,
http://de.wikipedia.org/wiki/Parametric_Technology_Corporation,
Abruf: 27.06.2014

Open Source Defect Management / Bug-Tracking Tools

Schriftliche Ausarbeitung
im Rahmen der Lehrveranstaltung „Integrationsseminar im 6. Semester“

Vorgelegt von

Marcel Dittkowski
Timm Walz
Tening Njie
Christian Heich

am 14.06.2014

Fakultät Wirtschaft
Studiengang Wirtschaftsinformatik
WWI2011V

Hinweis

Bei vorliegender Ausarbeitung handelt es sich um eine wissenschaftliche Arbeit weshalb, aus Gründen der besseren Lesbarkeit, auf eine geschlechtsspezifische Differenzierung verzichtet wurde. Im Sinne der Gleichbehandlung gelten die verwendeten Begriffe für beide Geschlechter.

Inhaltsverzeichnis

Abkürzungsverzeichnis	V
Abbildungsverzeichnis	VI
Tabellenverzeichnis	VII
1 Einleitung.....	1
2 Theoretische Grundlagen	2
2.1 Open Source.....	2
2.1.1 Definition	3
2.1.2 Lizenzmodelle.....	4
2.1.3 Proprietäre Software.....	6
2.2 Fehlerbegriff.....	6
2.2.1 Bug.....	7
2.2.2 Exploit.....	8
2.2.3 Bug-Tracking Tools	8
2.3 Referenzmodelle.....	10
2.3.1 Definition	11
2.3.2 ITIL V3.....	12
3 Soll-Zustand	16
3.1 Konzeption der Bug-Tracking Plattform	16
3.2 Pflicht- und Lastenheft	19
3.2.1 Pflichtenforderungen	19
3.2.2 Vergleichsanforderungen	22
4 Auswahl von Bug-Tracking Tools	26
4.1 Vorselektion	27
4.2 Vorstellung der selektierten Bug-Tracking Tools.....	28
4.2.1 Bugzilla.....	28
4.2.2 Mantis.....	28
4.2.3 Trac.....	29
4.2.4 Redmine.....	29
4.3 Vorgehensmuster bei der Bewertung.....	29
5 Test ausgewählter Bug-Tracking Tools	30
5.1 Trac.....	30
5.1.1 Installation	30

5.1.2	Pflichtkriterien.....	32
5.1.3	Vergleichskriterien.....	32
5.1.4	Gesamteindruck.....	33
5.2	Bugzilla.....	33
5.2.1	Installation.....	33
5.2.2	Pflichtkriterien.....	35
5.2.3	Vergleichskriterien.....	35
5.2.4	Gesamteindruck.....	36
5.3	Mantis.....	37
5.3.1	Installation.....	37
5.3.2	Pflichtkriterien.....	38
5.3.3	Vergleichskriterien.....	39
5.3.4	Gesamteindruck.....	40
5.4	Redmine.....	40
5.4.1	Installation.....	42
5.4.2	Pflichtkriterien.....	43
5.4.3	Vergleichskriterien.....	43
5.4.4	Gesamteindruck.....	44
5.5	Zusammenfassung der Bewertung.....	44
6	Vergleich: Open Source Bug-Tracking Tools mit kommerziellen Bug-Tracking Tools.....	46
7	Einführung eines Bug-Tracking-Tools im Unternehmen.....	53
7.1.1	Konzeption der Vorbereitungsmaßnahmen.....	53
7.1.2	Konzeption des infrastrukturellen Aufbaus.....	54
7.1.3	Konzeption der Rollenverteilung im Issue Tracking-System.....	57
7.1.4	Fortwährende Überlegungen.....	59
8	Fazit.....	61
9	Anhang.....	62
10	Quellenverzeichnisse.....	93

Abkürzungsverzeichnis

API	A pplication P rogramming I nterface
BSD	B erkeley S oftware D istribution
CCTA	C entral C omputer and T elecommunication A gency
CMMI	C apability M aturity M odel I ntegration
COBIT	C ontrol O bjectives for I nformation and R elated T echnology
DNS	D omain N ame S ystem
FAQ	F requently A s ked Q uestions
FSF	F ree S oftware F oundation
GPL	G eneral P ublic L icense
GUI	G raphical U ser I nterface
HP	H ewlett P ackard
IM	I ncident M anagement
IP	I nternet P rotocol
ITIL	I nformation T echnology I nfrast ructure L ibrary
ITSM	I nformation T echnology S ervice M anagement
KOS	K ompetenzcenter O pen S ource
LDAP	L ightweight D irectory A ccess P rotocol
LGPL	L esser G eneral P ublic L icense
MOF	M icrosoft O perations F ramework
OGC	O ffice of G overnment C ommerce
OSI	O pen S ource I nitiative
OSS	O pen S ource S oftware
W-LAN	W ireless- L ocal A rea N etwork

Abbildungsverzeichnis

Abb. 1: Logbuchseite inkl. der Motte von Grace Hopper.....	8
Abb. 2: Konzeptioneller Aufbau einer lauffähigen Bugtracking Umgebung	10
Abb. 3: ITIL Servicelifecycle	14
Abb. 4: Ranking von Bug-Tracking Tools.....	27
Abb. 5: Administrationsbereich in Trac.....	31
Abb. 6: Benutzerverwaltung in Bugzilla.....	34
Abb. 7: Mantis: Installationsmodus.....	38
Abb. 8: Redmine: Startmenü.....	41
Abb. 9:Gegenüberstellung der Vorteile von On-Demand und dedizierten Lösungen	48
Abb. 10: Produktübersicht für Jira	49
Abb. 11: Tarifierung des Jira-Produkts in der Cloud und auf dedizierten Servern.....	49
Abb. 12: Tarifierung von Adminitrac.....	51
Abb. 13: Darstellung des ePAVOS-Modells	54
Abb. 14: Konzeptioneller Aufbau der Netzwerkinfrastruktur	57
Abb. 15: Schrittweise Ablauf der System-Einführung.....	57

Tabellenverzeichnis

Tabelle 1: Trac : Anwendung der Pflichtkriterien.....	32
Tabelle 2: Trac: Anwendung der Vergleichskriterien.....	33
Tabelle 3: Bugzilla: Anwendung der Pflichtkriterien	35
Tabelle 4: Bugzilla: Anwendung der Vergleichskriterien	36
Tabelle 5: Mantis: Anwendung der Pflichtkriterien	39
Tabelle 6: Mantis: Anwendung der Vergleichskriterien	39
Tabelle 7: Redmine: Anwendung der Pflichtkriterien.....	43
Tabelle 8: Redmine: Anwendung der Vergleichskriterien	44
Tabelle 9: Zusammenfassung: Anwendung der Pflichtkriterien	45
Tabelle 10: Zusammenfassung: Anwendung der Vergleichskriterien.....	46
Tabelle 11: Gegenüberstellung von virtualisierten und dedizierten Systemen	55
Tabelle 12: Beispielhafte DNS-Konfiguration des Issue Tracking-Servers.....	56

1 Einleitung

Die Softwareentwicklung stellt einen komplexen Prozess dar, in dem zahlreiche Akteure beteiligt sind. Für diesen Prozess wird eine Kollaborationsplattform benötigt, über den die einzelnen Projektfortschritte dokumentiert werden können. Bei der Entwicklung und Benutzung von Software können grundsätzlich Fehler auftreten. Um den Lebenszyklus von Softwareprodukten zu begleiten und diese Fehler erfassen zu können, werden so genannte Bugtracking Tools in Unternehmen eingesetzt, die den Entwicklungsprozess von Software unterstützen. Diese sind sowohl aus dem kommerziellen, als auch aus dem Open Source-Umfeld erhältlich. Open Source-Software erfährt aufgrund der Eigenschaft als freie Software und der hohen Anpassbarkeit eine zunehmende Popularität in Unternehmen, welche durch die Initiierung des *Kompetenzcenter Open Source* (KOS)-Projekts bestätigt wird.

Innerhalb des theoretischen Teils dieser vorliegenden Ausarbeitung werden zunächst grundlegende Begriffe zum allgemeinen Verständnis der aufgeführten Themen aufgearbeitet. Im Zuge von steigender Relevanz informationstechnischer Prozesse gewinnen Referenzmodelle wie Information Technology Infrastructure Library (ITIL) zunehmend an Bedeutung.

Für die Auswahl von geeigneten Tools aus dem breiten Angebot werden unterschiedliche Plattformen herangezogen, die sich mit der Relevanz der einzelnen Softwareprodukte beschäftigen. Auf Grundlage der ITIL-Konformität und der Vorgaben innerhalb der Aufgabenstellung wird ein Kriterienkatalog erarbeitet, an dem die Tauglichkeit im Unternehmensumfeld der einzelnen Tools geprüft wird. Die einzelnen Open Source Produkte werden daraufhin mit ihren aussagekräftigsten Eigenschaften kurz vorgestellt.

Um den Funktionsumfang von Open Source-Tools einordnen zu können werden weitere Produkte aus dem kommerziellen Umfeld kurz vorgestellt. Hier wird ebenfalls auf die Lizenzbedingungen der einzelnen Produkte eingegangen.

Die Ausarbeitung wird von einem Fazit abgerundet, welches die zentralen Erkenntnisse der Forschungsarbeit erneut aufgreift und eine Empfehlung für ein Bug-Tracking Tool ausspricht.

2 Theoretische Grundlagen

In diesem Kapitel werden die, für diese Ausarbeitung benötigten, theoretischen Grundlagen ermittelt und erläutert.

2.1 Open Source

Die Weitergabe von Software, als Quelltext, ist auf die Hackerkultur der 1960er-Jahre zurückzuführen. Zu dieser Zeit wurde hauptsächlich im militärischen Bereich oder an Universitäten Software entwickelt. Aufgrund der Kommerzialisierung im Softwaremarkt wurde der Quelltext vermehrt als ein Geschäftsgeheimnis angesehen und dem Benutzer deswegen nur in Form von Binärcode zur Verfügung gestellt - der nicht verändert werden konnte. Als Gegenpol dazu gilt die von Richard Stallman in 1985 gegründete Free Software Foundation (FSF), sowie der Begriff „Freie Software“. Unter „Freie Software“ wird, laut der Free Software Definition der FSF, Software mit frei verfügbaren Quelltext verstanden. Dabei umfasst die Freiheit nicht nur die Verfügbarkeit des Quelltextes, sondern vielmehr auch das Recht diesen abzuändern, zu verbessern und anschließend zu veröffentlichen.¹

Nach dieser Definition war die Softwareentwicklungsumgebung GNU von Richard Stallman die erste Open Source Software (OSS), die darauf abzielte, ein alternatives und freies Betriebssystem zu erstellen. Insbesondere durch GNU/Linux von Linus Torvald hat OSS einen erheblichen Bekanntheitsgrad erreicht. Zusammen mit dem von Torvald entwickelten Linux-Kernel wurde das von Stallman initiierte GNU-Projekt zu einem vollständigen Betriebssystem, an dem mittlerweile viele, über die Welt verteilte, Entwickler arbeiten. Anfangs war GNU/Linux nur technikaffinen Personen bekannt, inzwischen gibt es aber zahlreiche Distributionen, die leicht installiert werden können und somit keine umfassenden Computerkenntnisse voraussetzen.²

Die Open Source Initiative (OSI) wurde in 1998 gegründet und liefert mit der Open Source Definition eine Alternative zur populären OSS-Lizenz GNU General Public License (GPL) der FSF. Die beiden Definitionen ähneln sich allerdings sehr.³

¹ Vgl. Keßler, S. (2013), S. 15

² Vgl. ebenda, S. 15 f.

³ Vgl. ebenda, S. 15; vgl. dazu auch Hüttenegger, G. (2006), S. 4

2.1.1 Definition

OSS bezeichnet im Wesentlichen dieselbe Software wie der Begriff „Freie Software“. Bei letzterem besteht allerdings ein Mehrdeutigkeitsproblem. Neben der beabsichtigten Bedeutung „Software die einem gewisse Freiheiten gibt“ kann auch „gratis Software“ in Freie Software interpretiert werden, was zu Akzeptanzprobleme im geschäftlichen Bereich führt. Die Absicht bei der Einführung des Begriffs „Open Source“ lag darin, dieses Problem zu lösen und einen Begriff für Freie Software zu erschaffen, der vor allem für den geschäftlichen Bereich akzeptabel ist.⁴

Von der Wortbedeutung ansich, bedeutet Open Source „quelloffen“ – der Quelltext darf eingesehen werden. Open Source bedeutet allerdings nicht nur quelloffen, sondern betrifft gleichzeitig die folgenden Kriterien:⁵

1. Free Redistribution (Freie Weitergabe)
2. Source Code (Das Programm muss den Quellcode beinhalten)
3. Derived Works (Abgeleitete Software)
4. Integrity of The Author's Source Code (Unversehrtheit des Quellcodes des Autors)
5. No Discrimination Against Persons or Groups (Keine Diskriminierung von einzelnen Personen oder Gruppen)
6. No Discrimination Against Fields of Endeavor (Keine Einschränkungen für bestimmte Anwendungsbereiche)
7. Distribution of License (Weitergabe der Lizenz)
8. License Must Not Be Specific to a Product (Die Lizenz darf nicht für ein bestimmtes Produkt gelten)
9. License Must Not Restrict Other Software (Die Lizenz darf andere Software nicht einschränken)
10. License Must Be Technology-Neutral (Die Lizenz muss technologisch neutral sein)

⁴ Vgl. Opensource (o. J.); vgl. dazu auch GNU (o. J.) und Henkel, J. (2007), S. 57 ff.

⁵ Vgl. Opensource (o. J.); vgl. dazu auch Henkel, J. (2007), S. 62 ff.

Zusammengefasst geht es in der engeren Definition von Open Source darum, die Modifizierbarkeit des Quelltextes zu garantieren. Nach Richard Stallmann beschränkt sich das Wesentliche auf die folgenden vier Punkte:⁶

1. Der Zugang zum Quellcode muss gewährleistet werden
2. Die Software darf kopiert und verteilt werden
3. Das Programm darf abgeändert werden
4. Das veränderte Programm muss wieder unter der GPL verteilt werden

Für Freie Software bzw. OSS sind Lizenzen, mit denen die vertraglichen Rechte und Pflichten der Urheber geregelt werden, von großer Bedeutung. Deswegen werden die Lizenzmodelle im nächsten Kapitel genauer vorgestellt.

2.1.2 Lizenzmodelle

Bevor auf einzelne Lizenzmodelle eingegangen wird, ist eine Abgrenzung von Freie Software bzw. OSS von den benachbarten Konzepten Freeware, Shareware und Public Domain-Software notwendig. Bei Shareware handelt es sich um copyright-geschützte Software. Diese ist kostenlos und mit der Bitte veröffentlicht, bei regelmäßiger Nutzung einen beliebig hohen, oftmals aber auch einen bestimmten, Geldbetrag dem Autor zukommen zu lassen. Ohne Quellcode und Veränderungserlaubnis, dafür aber kostenlos, ist Freeware hingegen frei weitergebbar. Oftmals enthält Freeware keine Lizenzbedingungen. Public Domain-Software ist nicht copyright-geschützt, da die Software gesetzlich nicht schützbar ist oder der Autor darauf verzichtet.⁷

Momentan gibt es ca. 45 verschiedene Lizenzen die von der OSI als Open-Source Lizenzen anerkannt werden und sich im Wesentlichen in zwei Merkmalen unterscheiden. Manche Lizenzen erlauben es die Software (oder veränderte Versionen) als proprietäre Software zu vermarkten – auf diese Art der Software wird in Kapitel 2.1.3 näher eingegangen. Andere Lizenzen wie bspw. die GPL schreiben dagegen vor, dass die Software (und alle Veränderungen dieser) unter der gleichen Lizenz veröffentlicht werden müssen. Diese Eigenschaft wird auch „viral“ oder „copyleft“ ge-

⁶ Vgl. Hüttenegger, G. (2006), S. 4; vgl. dazu auch Grassmuck, V. (2004), S. 233

⁷ Vgl. Grassmuck, V. (2004), S. 278 f.

nannt. Im Folgenden werden die drei wichtigsten Lizenzen Berkeley Software Distribution (BSD-Lizenz), GPL und Lesser General Public License (LGPL) vorgestellt.⁸

BSD-Lizenz

Die BSD-Lizenz enthält einen Copyright Vermerk und erlaubt die Verwendung, sowie die Weiterverbreitung der Software in Quell- und Binärform. Dies gilt zudem wenn Veränderungen an der Software vorgenommen wurden. Zusätzlich muss in allen Werbematerialien der Satz „Dieses Produkt beinhaltet Software, die von der Universität von Kalifornien in Berkeley und ihren Kontributoren entwickelt wurde“ verwendet werden.⁹

GPL

Die GPL hat die Absicht, die Freiheit sicherzustellen freie Software zu teilen und zu verändern. Die Freiheiten umfasst folgendes:¹⁰

- Das Programm darf für jeden Zweck ausgeführt werden
- Der Quellcode des Programms darf wörtlich kopiert und verbreitet werden sofern der Copyright Vermerk und die Lizenz mitverbreitet werden.
- Das Programm darf verändert und die veränderte Version darf ebenfalls kopiert und verbreitet werden. Dabei muss das veränderte Programm, wie das ursprüngliche, Angaben über die Änderungen enthalten und muss gebührenfrei unter denselben Lizenzbedingungen veröffentlicht werden.
- Die Software oder abgeleitete Versionen dürfen in Objektcode oder ausführbarer Form kopiert und verbreitet werden. Außerdem muss der dazugehörige maschinenlesbare Quellcode oder ein schriftliches mindestens drei Jahre gültiges Angebot, diesen Quellcode auf Anfrage bereitstellen.

„Um ein Programm mit Copyleft zu versehen, stellen wir zunächst fest, dass es dem Urheberrecht unterliegt, dann fügen wir als Rechtsinstrument Vertriebsbedingungen hinzu, die jedem das Recht einräumen, den Quellcode des Programms oder jedes davon abgeleitete Programm zu nutzen, zu modifizieren und weiterzuverbreiten, aber

⁸ Vgl. Henkel, J. (2007), S. 63 f.; vgl. dazu auch Keßler, S. (2013), S. 31

⁹ Vgl. Grassmuck, V. (2004), S. 279

¹⁰ Vgl. ebenda, S. 282 f.

nur, wenn die Vertriebsbedingungen unverändert bleiben. So werden der Quellcode und die Freiheiten rechtlich untrennbar miteinander verbunden.“¹¹

LGPL

Die GNU LGPL wurde vom FSF im Juni 1991 veröffentlicht und entspricht in der Grundintention der der GPL. Aller Freiheiten der GPL betreffen damit auch die LGPL, die Bibliothek muss frei kopier-, verbreit- und modifizierbar sein. Der Hauptunterschied zur GPL besteht darin, dass Programme, die die freie Bibliothek unter dieser Lizenz einlinken nicht selbst diesen Freiheiten unterstehen müssen.¹²

2.1.3 Proprietäre Software

Proprietäre Software wird umgangssprachlich auch als „unfreie Software“ bezeichnet, da sie im Gegensatz zu freier Software nicht verändert werden darf. Die Weiterverbreitung der Software ist je nach Lizenzvertrag teilweise erlaubt. Proprietäre Software kann durch Softwarepatente, das Urheberrecht oder durch die Behandlung des Quellcodes als Betriebsgeheimnis geschützt werden. Microsoft gilt als Pionier in diesem Bereich.¹³

2.2 Fehlerbegriff

Um das Verhalten eines Systems als fehlerhaft einstufen zu können, muss vorab festgelegt werden, wie die erwartete und korrekte Situation aussehen soll. Bei einem Fehler handelt es sich um die Nichterfüllung einer festgelegten Anforderung oder um Abweichungen zwischen dem Istverhalten und dem Sollverhalten des Systems. Dabei wird das Istverhalten während der Ausführung eines Systems festgestellt, wohingegen das Sollverhalten in sogenannte Spezifikationen oder Anforderungen festgelegt wird.¹⁴

Wird eine berechtigte Erwartung oder eine Anforderung nicht angemessen erfüllt, liegt ein Mangel vor. Dies ist auch dann der Fall, wenn die Verwendbarkeit eines Systems beeinträchtigt ist, obwohl dessen Funktionalität sichergestellt wird. Bei einem Softwaresystem entstehen Fehler nicht, wie bei physikalischen Systemen, durch

¹¹ GNU (2014)

¹² Vgl. Grassmuck, V. (2004), S. 289 f.

¹³ Vgl. Wikipedia (o. J.)

¹⁴ Vgl. Spillner A./Linz T. (2010), S. 7

Verschleiß oder Alterung. Fehler oder Mängel sind ab dem Zeitpunkt der Entwicklung in der Software vorhanden und werden oftmals erst bei der Ausführung registriert. Dieser Sachverhalt wird „Fehlerwirkung“ oder „failure“ genannt und hat seinen Ursprung in einem Fehlerzustand der Software. Unter „Bug“ wird ein solcher Fehlerzustand (Programm- oder Softwarefehler) verstanden.¹⁵

2.2.1 Bug

Ein Fehlerzustand bzw. ein Bug tritt auf, wenn der Programmierer einer Software eine bestimmte Spezifikation oder Festlegung nicht oder falsch umgesetzt hat oder die Laufzeitumgebung anders als erwartet arbeitet. Um die Ursachen für die aufgetretenen Bugs zu erkennen und zu analysieren werden sogenannte „Debugger“ eingesetzt, mit dem das Programm Schritt für Schritt ausgeführt und kontrolliert werden kann. Die anschließende Beseitigung des Bugs wird „bugfixing“ genannt. Das daraus resultierende Ergebnis wird in der Fachsprache als „Patch“ (Softwarepatch) oder als „Bugfix“ bezeichnet.¹⁶

„Bug“ ist Englisch und heißt übersetzt „Käfer“. Die Verwendung des Begriffs im Kontext von Computer geht möglicherweise auf die Computerpionierin Grace Hopper zurück, die im September 1945 eine Motte in einem Relais des Computers „Mark II Aiken Relay“ registrierte, welche zu einer Fehlfunktion führte. Diese Motte klebte sie daraufhin in ein Logbuch mit den Worten „First actual case of bug being found“. Das Logbuch ist in der folgenden Abbildung dargestellt.¹⁷

¹⁵ Vgl. ebenda, S. 7; vgl. dazu auch Wikipedia (2014a)

¹⁶ Vgl. Wikipedia (2014a)

¹⁷ Vgl. ebenda

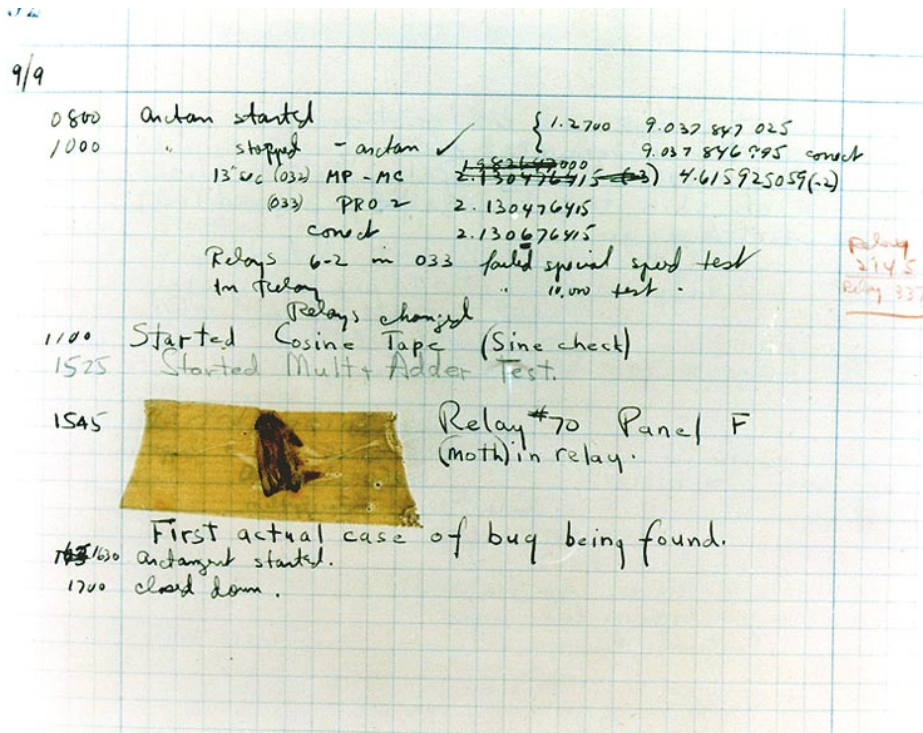


Abb. 1: Logbuchseite inkl. der Motte von Grace Hopper¹⁸

2.2.2 Exploit

Mit einem Exploit können innerhalb der elektronischen Datenverarbeitung Schwachstellen ausgenutzt werden, die bei der Entwicklung einer Software nicht berücksichtigt wurden. Das Ziel von Exploits ist es, dem Benutzer Zugang zu Ressourcen zu verschaffen oder Systeme zu beeinträchtigen, indem Sicherheitslücken oder Fehlfunktionen eines Systems oder einer Software verwendet werden.¹⁹

2.2.3 Bug-Tracking Tools

Ein Bug-Tracking Tool, oftmals auch Bugtracker genannt, ist ein Fallbearbeitungssystem, das in der Softwareentwicklung verwendet wird. Mit einem solchen Tool werden Programmfehler erfasst und dokumentiert, sowie Zustands- oder Feature-Berichte erstellt. Auf Basis dieser Dokumentationen versuchen die Softwareentwickler dann, die Bugs zu beseitigen. Durch diese Erfassung und Behebung der Programmfehler, wird mit einem Bugtracker die Kommunikation zwischen Anwendern und Softwareentwicklern verbessert. Aufgrund der umfassenden Dokumentation der

¹⁸ Enthalten in: Wikipedia (2014a)

¹⁹ Vgl. Wikipedia (2014b)

Bugs, wird zudem eine spätere Recherche bei bspw. ähnlichen Problemen ermöglicht. Des Weiteren bieten die meisten Bugtracker eine Funktion, Verbesserungsvorschläge und Wünsche der Benutzer aufzunehmen. Auf die bekannten Bugtracker Trac, Bugzilla, Mantis und Redmine wird im weiteren Verlauf dieser Ausarbeitung näher eingegangen.²⁰

Heutzutage bestehen die modernen Bugtracker aus mehreren Komponenten, um die Kollaboration zwischen Endanwender und den Softwareentwicklern zu optimieren.

Eine der soeben genannten Komponenten ist eine Datenbank. In dieser werden die aufgenommenen Fehler dokumentiert, verwaltet und anschließend archiviert. Mittels der Archivierung ist eine spätere Recherche nach alten, möglicherweise bereits gelösten Fehlern durchführbar. Insbesondere bei ähnlich auftretenden Problemen kann sich diese Funktion als nützlich erweisen. Die Anwender und deren Rechte in der Bug-Tracking Anwendung werden ebenso in der Datenbank abgelegt.

Der wichtigste Faktor für eine Kollaboration aller Teilnehmer ist die die Softwareunabhängigkeit zur Verwendung/Benutzung der Anwendung. Dieser Bestandteil wird in der Regel durch den Einsatz einer Schnittstelle zu einem Webserver realisiert. Der Zugriff auf die Anwendung erfolgt hierdurch fast ausschließlich über eine Weboberfläche mittels einem herkömmlichen Webbrowser. Der Benutzer kann anschließend über eine grafische Benutzeroberfläche mit dem System interagieren. Für gewöhnlich sind die Oberflächen solcher Systeme dermaßen konzeptioniert, sodass das Einsenden, Bearbeiten und Dokumentieren von Fehlerberichten mittels vordefinierten Rahmenbedingungen für den Endanwender erheblich erleichtert wird.

Damit ein ordnungsgemäßer Betrieb der Anwendung gewährleistet ist, ist eine weitere Komponente ein Anwendungsserver. Dieser stellt eine lauffähige Umgebung in Form von Betriebssystem und den notwendigen Ressourcen (Arbeitsspeicher, Prozessorleistung, Festplattenplatz) zur Verfügung.

Diese klassische Aufteilung der Komponenten in die Präsentationsschicht (Webserver), Datenhaltungsschicht (Datenbankserver) und Anwendungsschicht (Anwendungsserver) entspricht dem Client-Server Modell. In Anbetracht der drei Komponen-

²⁰ Vgl. Wikipedia (2013)

ten und dem Endanwender stellt ein Bugtracking System generell eine 4-tier Architektur dar. Dies wird in der folgenden Abb. 2 dargestellt.

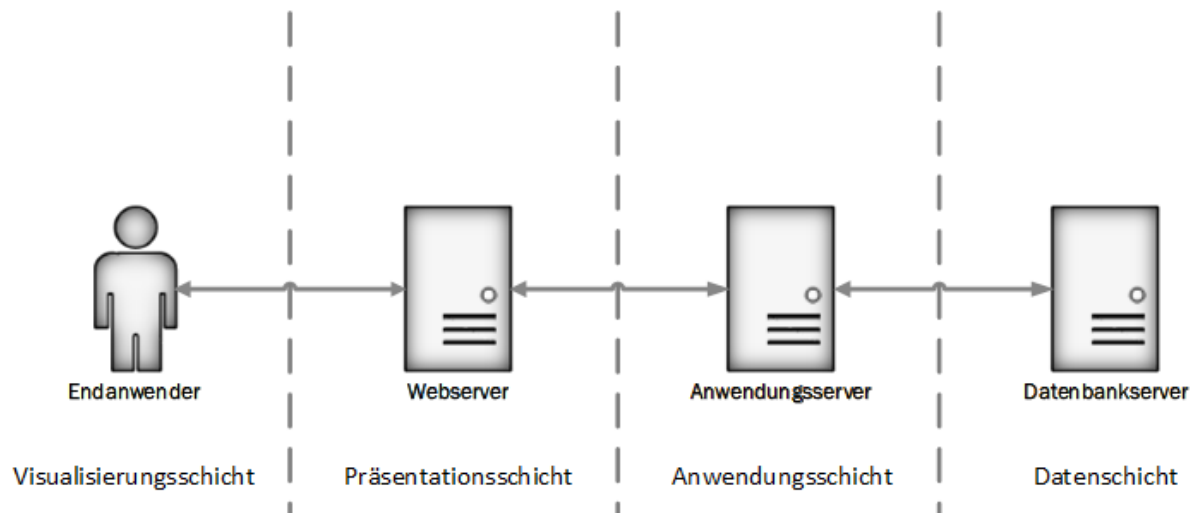


Abb. 2: Konzeptioneller Aufbau einer lauffähigen Bugtracking Umgebung

2.3 Referenzmodelle

Die rasante Entwicklung in der IT lassen die Anforderungen an die IT-Infrastrukturen der Unternehmen stetig steigen. Cloud-Computing, Smartphones und weitere neue Technologien haben die Branche über das letzte Jahrzehnt hinweg stark verändert. „Erhöhte Flexibilität und Adaptivität der IT-Abteilungen in ihren Technologien, Prozessen und Organisationsstrukturen, werden aufgrund neuer Geschäftsmodelle und der Erschließung neuer Märkte [stetig] gefordert.“²¹ Zur Bewältigung der Ansprüche sind Maßnahmen und Methoden mit dem Fokus hinsichtlich Serviceorientierung notwendig. Hierfür sind jedoch keine rein wissenschaftlichen Vorgehensweisen vorhanden, sodass sich im Laufe der Zeit sogenannte Best-Practice Referenzmodelle entwickelt haben. Im Zuge der verlangten Serviceorientierung wurden unterschiedliche Möglichkeiten generiert, IT Service Management (ITSM)-Praktiken umzusetzen. Die wohl bekannteste und am weit verbreiteste Best-Practice Lösung (engl. für bewährte Methoden) für die Umsetzung von ITSM ist das Information Technology Infrastructure Library.

²¹ Goeken, M. u.a. (o. J.)

Im folgenden Kapitel werden die grundlegenden Eigenschaften von Referenzmodellen erläutert. Dabei wird auf die Metamodellierung, als eine Möglichkeit Referenzmodelle in die Praxis umzusetzen, eingegangen. Anschließend wird das für diese Seminararbeit relevante Referenzmodell ITIL mit dem Aspekt für eine Einführung eines Bugtracking-Systems vorgestellt.

2.3.1 Definition

"Ein **Referenzmodell** ist ein für eine Branche oder einen ganzen Wirtschaftszweig erstelltes *Modell*, das *allgemeingültigen Charakter* haben soll. Es dient als Ausgangslösung zur Entwicklung unternehmensspezifischer Modelle (Becker/Schütte 1996)."²²

Ein fundamentales Charakteristikum von Referenzmodellen ist, dass die Entwicklung vorwiegend auf Praxiserfahrung der Autoren beruht. Deshalb stoßen Referenzmodelle auf immer mehr Akzeptanz und finden daher vermehrt Anwendung in der Praxis. Aus diesem Grund können sich Unternehmen, bzw. deren Mitarbeiter in diesen Referenzmodellen zertifizieren lassen. Das Unternehmen ist im Zuge dessen berechtigt, sich mit der angeeigneten Kompetenz in der Öffentlichkeit zu repräsentieren. Teilweise werden die Zertifizierungen als ein Qualitätsmerkmal in der Gesellschaft anerkannt. Infolgedessen werden offene als auch proprietäre Referenzmodelle angeboten. Unter den offenen Referenzmodellen befinden sich unter anderem ITIL, Control Objectives for Information and Related Technology (COBIT) oder Capability Maturity Model Integration (CMMI). Große Unternehmen wie beispielsweise Microsoft oder Hewlett Packard (HP) sind in der Lage, eigene, von der breiten Masse akzeptierte proprietäre Referenzmodelle zu erstellen. Bezeichnend hierfür existieren die Referenzmodelle Microsoft Operations Framework (MOF) oder das ITSM Reference Model von HP.²³

“Die unterschiedlichen Referenzmodelle legen ihren Schwerpunkt jeweils auf unterschiedliche Aspekte. So widmet sich ITIL dem IT-Servicemanagement, COBIT ist als

²² Krcmar, H. (2005), S. 107

²³ Vgl. Goeken, M. u.a. (o. J.)

Modell für IT-Governance positioniert, ohne dabei aber seine Wurzeln im Auditing zu verbergen.“²⁴

2.3.2 ITIL V3

Abhängig davon, in welcher Branche ein Unternehmen tätig ist, stellt die IT ein essenzielles Element in der Wertschöpfung des Betriebes dar. Dies ist beispielsweise bei Hard- und Softwareunternehmen der Fall. Größtenteils gilt sie jedoch als unterstützender Teil in einem Unternehmen und trägt zur Bewerkstelligung der Wertschöpfung bei. Daher sind Begrifflichkeiten wie Kunden- und Serviceorientierung im ITSM-Bereich hinsichtlich des Unternehmenserfolges nicht mehr wegzudenken. Generell verfolgt das ITSM drei Hauptziele:

- Gestaltung der IT Services gemäß den „Anforderungen des Unternehmens und seiner Kunden“²⁵
- Verbesserung der Qualität des geleisteten Services
- langfristige Reduzierung der Kosten für die Erbringung der Leistung

“Um diese Ziele zu erreichen, haben sich international anerkannte Best Practices für IT-Prozesse durchgesetzt. Diese sind in ITIL und COBIT dokumentiert.“²⁶

ITIL ist ein Akronym für IT Infrastructure Library. Entsprechend der Namensgebung handelt es sich hierbei um eine Sammlung von Werken, deren Inhalte die informationstechnischen Prozesse im Unternehmen optimieren sollen.²⁷ Die „strategische, taktische und operative Umsetzung von IT Services“²⁸ hat hierbei oberste Priorität.

Ursprünglich war ITIL ein Auftrag der britischen Regierung an die Central Computer and Telecommunication Agency (CCTA) – heute als Office of Government Commerce (OGC) bekannt – mit dem Ziel die IT Kosten zu senken und zugleich eine bessere Effektivität im staatlichen Bereich zu erzielen. Des Weiteren sollten dabei die

²⁴ Goeken, M. u.a. (o. J.)

²⁵ Buchsein, R. u. a. (2007), S.72

²⁶ Buchsein, R. u. a. (2007), S.72

²⁷ Vgl. Ebel, N. (2008), S. 21

²⁸ Ebel, N. (2008), S. 17

neuen technischen Innovationen nicht außer Acht gelassen werden. Die erste Version des ITIL Frameworks wurde Ende der 80er Jahre dann von der CCTA veröffentlicht. In den folgenden Jahren setzte sich das Framework zunehmend als Grundlage für die Erbringung von IT-Services durch. Wie zuvor bereits erwähnt besteht ITIL aus einer Sammlung von Werken, was zur damaligen Zeit aufgrund der vielen unterschiedlichen Autoren zu einem unabgestimmten und uneinheitlichen Framework führte. Dieser Problematik waren sich auch die Entwickler des ITIL Referenzmodells bewusst, sodass die OGC das Rahmenwerk in Kooperation mit dem Information Technology Service Management Forum (ITSMF) stetig weiterentwickelt. Zwischenzeitlich sind zwei größere Aktualisierungen des Rahmenwerks vorgenommen worden. Im Jahre 2006 erschien die Fassung in der Version zwei (ITIL V2). Diese umfasste insgesamt elf Werke, aber sah sich immer noch der Problematik, inhaltlich loser nebeneinander stehender Texte, ausgesetzt. Die derzeit aktuelle Fassung ist eine überarbeitete Auflage des ITIL Framework in der Version drei (ITIL V3) mit der Bezeichnung ITIL 2011. Es handelte sich dabei lediglich um Fehlerbehebungen aus der ersten Auflage von ITIL V3. Der Fokus von ITIL V3 liegt auf dem neuen Prozessmodell „Service Life Cycle“, welches sich an die Lebenszyklusphasen von IT-Services orientiert.

Das neue ITIL-Framework teilt seine Best-Practice-Ansätze für Servicemanagement-Prozesse in fünf Kernbereiche auf:

- Service Strategy
- Service Design
- Service Transition
- Service Operation
- Continual Service Improvement

„Die Module folgen einer Lebenszyklusperspektive auf IT-Services und schaffen die Grundlage für die verbesserte Verzahnung von IT-Servicemanagement und – Geschäftsstrategie“.²⁹ Die folgende **Fehler! Verweisquelle konnte nicht gefunden werden.** verschafft einen Überblick über den zuvor erwähnten Service Life Cycle und der fünf Kernbereiche. Der Service Life Cycle stellt die Abhängigkeiten der einzelnen Module und der äußeren Einflüsse auf die IT-Services dar. Das erste Kernbuch Ser-

²⁹ Böttcher, R. (2008)

vice Strategy gilt als „die Achse des Servicelebenszyklus“³⁰. Darin werden die Strategie, die Richtlinien und die Ziele, an denen sich die anderen Phasen Service Transition, Service Design und Service Operation orientieren, festgelegt. Darauf aufbauen befassen sich die soeben genannten Phasen jeweils mit der Anpassung und Veränderungen von IT-Services. Das letzte Modul Continual Service Improvement umfasst alle Prozesse und beschäftigt sich mit der stetigen Optimierung der IT Services.³¹



Abb. 3: ITIL Servicelife Cycle³²

Die wichtigen Kernelemente zur Metamodellierung eines Bugtracking Systems lassen sich vor allem im vierten Modul des ITIL Referenzmodells wiederfinden. Deshalb wird im Folgenden dieser Bereich etwas näher erläutert.

Service Operation

³⁰ Bon, J. van (2008), S. 22

³¹ Vgl. Bon, J. van (2008), S. 22

³² Enthalten in Ebel, N. (2008), S. 32

Diese Publikation beschreibt Herangehensweisen, welche für die Wertschöpfung des Unternehmens von relevanter Bedeutung sind. Die dort angesiedelten Prozesse und Funktionen befassen sich überwiegend mit der Sicherstellung des Betriebes und erzeugt den Nutzen für den Kunden.

Service Operation realisiert damit die strategischen Ziele des Unternehmens durch operative Dienstleistungen. Es wird ein Orientierungsrahmen zur Verfügung gestellt, um stabile Leistungen zu erbringen und diese kontinuierlich an die Bedürfnisse der Kunden und Nutzer anzupassen. Inhaltlicher Schwerpunkt ist unter anderem das Incident und Problem Management.³³ Denn bei Störungen im täglichen Betrieb, kann kein Nutzen erzielt werden, sowie keine Verbesserungen der bestehenden Prozesse vorgenommen werden.

2.3.2.1 Incident Management

Das Incident Management (IM) dient der schnellstmöglichen Wiederherstellung des normalen Service-Betriebs bei minimaler Störung des Geschäftsbetriebs.³⁴ Ein Incident stellt eine ungeplante Störung und der damit eingehenden Qualitätsminderung eines IT-Services dar.³⁵ Ziel des IM ist vorrangig die Beseitigung von Fehlern, die Sichtung und Behebung der Störungsursache fällt nicht unter das Incident Management. „Dabei werden erste Hilfestellungen geleistet und gegebenenfalls die weitere Bearbeitung in den nachgelagerten Support-Einheiten koordiniert.“³⁶ Diese Hilfestellungen liefern Methoden und Massnahmen zur Erfassung und Dokumentation von Störungen, deren Kategorisierung und Priorisierung und definiert die Eingangskanäle von Incidents. ITIL empfiehlt als ersten Eingangskanal einen Service Desk.³⁷ Der Service Desk entspricht einem sogenannten First-Level-Support (FLS), welcher nicht behebbare Störungen an nachgelagerte Stufen in der Hierarchie eskaliert. Die Ursachenforschung überschreitet den Aufgabenbereich des IM und wird dementsprechend an die zuständigen Support-Level des Problem Managements eskaliert.

³³ Vgl. Böttcher, R. (2008), S. 3

³⁴ Vgl. Ebel, N. (2008), S. 465

³⁵ Vgl. Ebel, N. (2008), S. 466

³⁶ Ebel, N (2008), S. 465

³⁷ Vgl. Böttcher, R. (2008), S. 134

2.3.2.2 Problem Management

Das Problem Management übernimmt die Aktivitäten, welche den Aufgabenbereich des Incident Management überschreiten. Im Wesentlichen betrifft dies die Ursachenforschung und alle Aktivitäten zur Beseitigung des Problems.³⁸ Ziel des Problem Management ist es nachhaltig wiederholt auftretende Störungen zu verhindern. Das Problem Management stellt somit eine Ergänzung des Incident Managements um ein proaktives Element dar, welche eine Minimierung von Störungen, Fehlern und Problemen im Vorfeld sicherstellt. Die beiden Prozesse Incident- und Problem Management wirken eng miteinander zusammen, weshalb Incidents und Probleme gemeinsam in einem Trouble-Ticket-System erfasst und bearbeitet werden.³⁹

3 Soll-Zustand

Damit eine Auswahl eines geeigneten Bugtracking-Tools stattfinden kann, muss vorab geklärt werden, welchen Anforderungen das Tool gerecht werden muss. In diesem Kapitel wird deshalb ein Kriterienkatalog zu erstellt, welcher die verschiedenen Voraussetzungen auflistet. Da nicht allen Kriterien die gleiche Bedeutung zukommt, werden diese einerseits in sogenannte Pflicht- und Vergleichskriterien unterschieden. Die Auflistung der Vergleichskriterien wird mittels einer Gewichtung stattfinden, welche die einzelnen Kriterien anhand ihrer Wichtigkeit beurteilt.

3.1 Konzeption der Bug-Tracking Plattform

Mittels einer Bug-Tracking-Lösung soll vorwiegend eine Kollaborationsplattform für alle am Projekt beteiligten Personen geschaffen werden. Insbesondere Störungen, Probleme und Fehler sollen auf einer solchen Plattform erfasst und bearbeitet werden, um einen ordnungsgemäßen Betrieb eines Unternehmens zu gewährleisten. Da das Referenzmodell ITIL sich zu einem defacto Standard entwickelt hat, kann es optimal für eine generelle Konzeption herangezogen werden. ITIL beschreibt für eine erfolgreiche Umsetzung von IT Services generelle Statements.

³⁸ Vgl. Böttcher, R. (2008), S. 145

³⁹ Vgl. Elsässer, W. (2006), S. 50

Kommunikation als Schlüsselfaktor

Wie bereits in der Einleitung erwähnt sind am Softwareentwicklungsprozess einige Akteure beteiligt. Für diesen Prozess ist eine adequate Kommunikation aller Beteiligten notwendig. Dementsprechend ist eine Kollaborationsplattform für einen transparenten und nachvollziehbaren Informationsaustausch erforderlich. Gemäß ITIL ist die Kommunikation zwischen den IT-Verantwortlichen und den Kunden der Schlüssel für einen erfolgreichen Service. Zur Unterstützung einer solchen Plattform spielen die Eigenschaften Verfügbarkeit, Mobilität und Softwareunabhängigkeit auf Seiten aller teilnehmenden Parteien eine essentielle Rolle. Eine mögliche Realisierung dieser Eigenschaften wird von ITIL nicht genau definiert, ist dennoch für gewöhnlich mittels eines Web-Services zu realisieren.

Dokumentation schafft optimale Rahmenbedingungen

Eine ordentliche Dokumentation schafft die optimale Grundlage für eine erfolgreiche Fehlerbearbeitung. Das ITIL Framework befasst sich unter anderem mit den notwendigen Informationen für eine ordnungsgemäße Bearbeitung von Störungsfällen. Für eine korrekte Fehlerdokumentation ist einerseits die korrekte Form und insbesondere der korrekte Inhalt ausschlaggebend. Eine schriftliche Erfassung einer Störungsursache muss möglichst transparent ausgelegt werden, sodass alle beteiligten Service-Gruppen die ergriffenen Maßnahmen nachvollziehen können. Für diesen Zweck werden einige Punkte von ITIL herausgehoben. Diese werden im folgenden aufgelistet.⁴⁰

- ID, die einem Fehler eindeutig zuordenbar ist
- Genaue Fehlerbeschreibung
- Zeitpunkt der Erfassung
- Priorität (Auswirkung + Dringlichkeit)
- Störungsbearbeiter
- Störungsstatus

⁴⁰ Vgl. Böttcher, R. (2008), S. 135

- Zeitpunkt der Störungsbeseitigung

Hierarchie in der Service-Organisation

Eine eindeutige Festlegung von Rollen und der damit verbundenen Verantwortlichkeiten führt zu klaren Strukturen in der Prozessorganisation. Insbesondere zur Steuerung von betrieblichen Abläufen ist dies sehr zuträglich. Das ITIL Referenzmodell beschreibt hierfür allgemeine Rollen, welche für gewöhnlich nicht eins zu eins übernommen werden können. Diese sind bei einer Überführung in die Unternehmenslandschaft entsprechend anzupassen. Im ITIL Framework ist im Allgemeinen von Service-Gruppen bzw. Support-Teams die Rede. Diese werden generell anhand ihrer Kompetenzen in die Folgenden Gruppen unterschieden:

- First-Level Support
- Second-Level Support
- N-Level Support
- Prozess-Manager

Das First-Level Support-Team dient als Eingangskanal einer Störungsmeldung und ist für die anschließende Eskalation des Problems zuständig. Im Second-Level Support befinden sich Akteure mit einem etwas breiteren und tieferen Expertenwissen als der First-Level-Support, sind aber dennoch für die weniger komplizierten Incidents zuständig.⁴¹ Alle nachgelagerten Eskalationsstufen in der Hierarchie werden als n Level-Support bezeichnet und zeichnen sich jeweils durch tiefer gehendes Wissen aus. Zudem ist ein Prozess-Manager, welcher für die ordnungsgemäße Durchführung des jeweiligen Prozesses zuständig ist, notwendig. Dies sind die generellen Rollen des ITIL Referenzmodells für die Störungsbeseitigung und setzen lediglich Richtlinien für einen möglichen Aufbau eines IT-Services.

Kategorisierung und Priorisierung

Das Incident Management ist verantwortlich Störungen von IT Services schnellstmöglich zu beheben. Dabei ist es das Ziel, „die Service-Qualität und –Verfügbarkeit

⁴¹ Vgl. Ebel, N. (2008), S. 471

auf dem höchstmöglichen Level zu halten.“⁴² Dabei ist es unumgänglich die auftretenden Störungen mittels einer Kategorisierungssystematik und einer entsprechenden Priorisierung zu versehen. Dies führt zu einer koordinierteren Vorgehensweise für die Störungsbeseitigung und dient einer schnelleren Bearbeitung der Störungsfälle. Diese Methoden und Massnahmen ermöglichen eine bessere Identifizierung des Incidents und eine entsprechende Weiterleitung an die richtigen Service-Gruppen. Die Priorität eines Incidents wird anhand der Dringlichkeit und der Auswirkung einer Störung definiert. Die Zuordnung von Prioritäten bestimmt die zeitliche Abarbeitung von Incidents.⁴³ „Eine First-In-First-Out Abarbeitung würde dazu führen, dass seine Störung, die einen Nutzer zu umständlichem Arbeiten zwingt [..], eher bearbeitet wird als seine Störung, die den Stillstand der Produktion zur Folge hat.“⁴⁴ Daher ist es wichtig wesentliche Störungen vor unwesentlichen zu bearbeiten. Aus diesem Grund ist die Definition von Prioritäten innerhalb einer IT-Infrastruktur notwendig.

3.2 Pflicht- und Lastenheft

Als Ausgangspunkt für eine Entscheidung hinsichtlich einer Softwareauswahl ist es empfehlenswert einen Anforderungskatalog bestehend aus Pflicht- und Vergleichsanforderungen zu erstellen. Dieser wird in den folgenden Kapiteln auf Basis der Konzeption erstellt. Jedoch beschreibt ITIL den konkreten Aufbau einer Lösungsdatenbank nicht genau, sodass ein Teil der Kriterien anhand einer Internetrecherche über die Funktionalitäten der unterschiedlichen Tools ermittelt wurde.

3.2.1 Pflichtenforderungen

Bei den Pflichtenforderungen handelt es sich um Kriterien, welche zwingend erfüllt werden müssen. Eine Nichterfüllung der Kriterien führt zu einem Ausschluss der Bugtracking Tools im Auswahlverfahren. Alternative Begrifflichkeiten sind auch „must-have“, Ausschluss- oder KO-Kriterien. Die im folgenden Abschnitt erläuterten Kriterien sind daher für einen möglichen Einsatz unerlässlich.

1. Webinterface

⁴² Ebel, N (2008), S. 467

⁴³ Vgl. Ebel, N. (2008), S. 468

⁴⁴ Böttcher, R. (2008), S. 136

Ein Kriterium welches ein Bugtracking-Tool erfüllen muss, betrifft das Webinterface. Hinter diesem Kriterium verbirgt sich die Notwendigkeit mittels einem herkömmlichen Webbrowser mit dem Bugtracking System interagieren zu können. Ein Webinterface bietet durch eine grafische Benutzeroberfläche einerseits den Vorteil der Plattform- und der Softwareunabhängigkeit auf Endanwenderseite. Somit entfällt eine lokale Installation der Anwendung auf dem Endgerät. Zusätzlich führt der Einsatz einer Webschnittstelle zu einer verbesserten Mobilität hinsichtlich der Zugriffsmöglichkeiten.

2. Definierbarer Workflow

Damit sich ein Bugtracking-Tool als geeignet erweist, muss eine Möglichkeit für eine individuelle Festlegung von Prozessabläufen gewährleistet sein. Die Verwendung einer Workflow Funktion unterstützt die Koordination aufeinander folgende oder gleichzeitig ablaufende Prozesse. Ein solcher Workflow (engl. für Arbeitsablauf) bildet für gewöhnlich die beteiligten Rollen, Aufgaben und Bedingungen ab. So kann beispielsweise die Lösung einer Aufgabe zu Folgeaufgaben und Benachrichtigungen bestimmter Beteiligten führen. Dies führt zu eindeutigen Zuständigkeiten und folglich zu einem geringeren Administrationsaufwand und Zeitersparnis.

3. Open Source

Für einen Einsatz von Open Source Software sprechen einige Gründe. Der Zugang zu offenen Standards, keine oder geringe Lizenzkosten als auch die Herstellerunabhängigkeit um nur ein paar Stärken von Open Source Software. Bei Bedarf kann der Quellcode der OSS durch individuelle Anforderungen erweitert und angepasst werden.

4. Benachrichtigungsfunktion

Ein weiteres essentielles Kriterium für den Einsatz ist eine automatische Benachrichtigungsfunktion, sodass alle beteiligten und zuständigen Personen bei einer Änderungen vom Bugtracking System automatisch generierte E-Mails erhalten. Diese Funktion ist sehr hilfreich für eine termingerechte Erfüllung/Abgabe von Aufgaben. In der Praxis sind Softwareentwickler oftmals in mehreren verschiedenen Projekten beteiligt, sodass der Überblick über die zu erledigenden Aufgaben verloren gehen kann. Ein Benachrichtigungsfeature unterstützt die Projektteilnehmer hinsichtlich des Zeitmanagements und die Organisation in einem Projekt.

5. Dateianhänge

Für ein erfolgreiches Fehlermanagement ist es unerlässlich, dass Dateianhänge in Form von Dokumenten oder Bilddateien eingefügt werden kann. Screenshots sind ein wichtiger Bestandteil von Fehlerberichten, da eine Visualisierung oftmals aussagekräftiger als eine schriftliche Form ist, sind schwer zu erklärende Sachverhalte einem Anwender deutlich einfacher zu vermitteln. Deshalb ist dieses Feature für einen unkomplizierten Informationsaustausch sehr hilfreich.

6. Klassifizierung

Ein weiteres Kriterium das ein Bugtracking Tool zu erfüllen hat ist eine Kategorisierung der eingehenden Fehler im System notwendig. Dies dient vorwiegend einer besseren Koordination und schnelleren Bearbeitung innerhalb eines Projektes. So können Fehler beispielsweise direkt an die richtigen Support-Gruppen weitergeleitet werden. „Dazu muss eine einheitliche Kategorisierungssystematik entwickelt (z.B. Hardware-, Software-, Netzstörungen) und verbindlich vorgegeben werden.“⁴⁵ Mittels diesem Feature kann die Bearbeitungszeit durch schnellere Zuordnungen minimiert werden.

7. Priorisierung

Das ITIL Referenzmodell empfiehlt für die Erfassung von Störungen eine Priorisierung vorzunehmen. Hierbei ist es wichtig etwas Wesentliches vom Unwesentlichen zu unterscheiden, da eine parallele Bearbeitung aller auftretenden Probleme nur schwer zu bewerkstelligen ist. Daher müssen schwerwiegende Probleme mit Vorrang behandelt werden.

8. Mandantenfähig

Ein Bugtracking Tool ist mandantenfähig, wenn es die Verwaltung mehrere Projekte unterstützt. Dieses Feature hat die Vorteile einer zentralen Installation und Verwaltung der Software. Zusätzlich ist die Einsichtnahme der Anwender auf die eigenen Daten ein weiterer essentieller Aspekt, sodass gewisse Projekte nur von berechtigten Benutzern eingesehen werden können. Dies ermöglicht den Projektleitern eine vereinfachte Administration der Anwendung.

⁴⁵ Vgl. Böttcher, R. (2008), S. 135

9. Such-/Sortierfunktion

Für die Auswahl eines geeigneten Bugtracking Tools ist für die Endanwender das Vorhandensein einer Such- mit integrierter Filterfunktion eine wichtige Rolle. Mittels dieser Funktion soll es dem Benutzer möglich sein, sich beispielsweise Fehlerlisten nach bestimmten Kriterien anzeigen zu lassen.

10. Benutzerverwaltung

Unter einer Benutzerverwaltung wird die Administration verstanden, damit die Benutzer die für Sie zuständigen Arbeiten erledigen können. Zugleich können unbefugte Tätigkeiten durch Steuerung der Rechte explizit eingestellt werden. Die Benutzerverwaltung soll zudem ein Rollenkonzept unterstützen, damit eine Aufwandsersparnis für den Administrator in Benutzerverwaltung erzielt wird.

11. Änderungsverlauf

Bei dem Änderungsverlauf handelt es sich um eine vom System angelegte Historie. Mit Hilfe dieser Funktion sollen alle Änderungen an einem Fehlerticket inklusive Zeitstempel und Benutzerkennung aufgezeichnet werden. Damit können bereits ergriffene Maßnahmen nachvollzogen und gegebenenfalls rückgängig gemacht werden. Der Änderungsverlauf stellt zudem eine unterstützende Funktion in der Koordination eines Projektes dar, indem die Revision einen Überblick verschafft und gegebenenfalls Doppelarbeiten vermieden werden können.

3.2.2 Vergleichsanforderungen

Für eine Evaluierung eines Produktes ist es zwingend erforderlich eine Basis für einen Vergleich zu schaffen. Deshalb werden im Folgenden die sogenannten Vergleichsanforderungen, auch „nice-to-have-“ oder Kann-Anforderungen genannt, vorgestellt, welche für einen Einsatz nicht zwingend notwendig sind. Daher führt eine Nichterfüllung dieser Kriterien zu keinem Ausschluss, trotzdem kommt diesen jeweils eine bestimmte Bedeutung zu.

Für einen solchen Vorgang bieten sich generell Punktesysteme an, um eine zahlenmäßige Auswertung vornehmen zu können. Infolgedessen wird den verschiedenen Kriterien jeweils ein Gewichtungsfaktor zugeordnet, welcher die Relevanz des Kriteriums zum Ausdruck bringt. Dieser wird zusammen mit einer Punktzahl zu einem Ge-

Samtergebnis verrechnet, das ein Bewerten der verschiedenen Tools ermöglicht. Die Punktzahl wird in Abhängigkeit des Erfüllungsgrades ermittelt. Für die Gewichtung wurden die Punkte wie folgt festgelegt:

- 5 Punkte für sehr wichtig
- 3 Punkte für wichtig
- 1 Punkt für unwichtig

1. Plattformunabhängigkeit

Die Plattformunabhängigkeit beschreibt die notwendigen Ressourcen einer Anwendung für eine lauffähige Umgebung. Damit ist den Unternehmen eine bedingte Flexibilität bei der Auswahl der Plattform gegeben. In Anbetracht des Open Source Pflichtkriteriums ist für sehr wahrscheinlich eine plattformübergreifende Nutzung der Tools auf den gängigen (Server-) Betriebssystemen wie beispielsweise Linux oder Windows gegeben. Angesichts dieses Aspekts erhält diese Eigenschaft lediglich eine geringe Gewichtung.

Gewichtung: 1

2. Usability

Des Weiteren ist die Usability ein wichtiger Aspekt für die Auswahl eines Bugtracking Tools. Dabei soll vorwiegend die Bedienung der Anwendung möglichst intuitiv sein. Der Hintergrundgedanke ist hierbei, dass keine zusätzliche Schulung für den Umgang mit dem Tool notwendig ist. Da in einer Bugtracking Software zahlreiche Akteure beteiligt sind, unter denen sich auch nicht technisch versierte Anwender befinden, daher ist eine einfache Handhabung der Anwendung sehr hilfreich, weshalb dieses Kriterium als sehr wichtig eingestuft wird.

Gewichtung: 1

3. Hilfefunktion

Die Hilfefunktion stellt eine Ergänzung zu der Usability dar. Dieses Feature soll den Anwendern bei Unwissen in der Bedienung der Anwendung weiterhelfen. Mögliche

Praktiken sind für gewöhnlich integrierte Dokumentationsplattformen. Diese können unter anderem in Form einer Wiki-Plattform oder als eine direkte Hilfefunktion realisiert werden und sollen dem Benutzer beispielsweise bei sogenannten Frequently Asked Questions (FAQ, engl. für häufig gestellte Fragen) weiterhelfen. Diese Funktion stellt daher eine unterstützende Funktion zu der Usability dar und wird deshalb als wichtig eingestuft.

Gewichtung: 3

4. Versionskonzept

Eine Gliederung von Softwareprojekten in einzelne Versionsnummern ist insbesondere bei der Fehlerverfolgung eine sehr hilfreiche Funktion. Auftretende Probleme und Fehler bei einem Releasewechsel können somit direkt einer bestimmten Version zugeordnet werden. Bezeichnend hierfür sind auftretende Fehler in einer neuen Version, welche in der Vorgängerversion nicht vorhanden waren. Infolgedessen liegt die Ursache des Fehlers für gewöhnlich in den Änderungen zwischen den beiden Versionen. Dieser Funktion ist eine wichtige Bedeutung beizumessen.

Gewichtung: 3

5. LDAP-Anbindung

Mittels dem Anwendungsprotokoll Lightweight Directory Access Protocol (LDAP) sind Abfragen oder Manipulation von Verzeichnisdiensten ausführbar. Dabei wird eine Anfrage gegenüber einer Datenbank des Verzeichnisdienstes durchgeführt. Entscheidender Prozess bei einer LDAP Abfrage ist die Authentifizierung. Diese überprüft ob der Client die Berechtigung auf einen Zugriff hat. Somit ist eine Benutzerauthentifizierung umsetzbar, welche auf zentral verwaltete Datenbank des Verzeichnisdienstes zurückgreift. Dadurch kann eine redundante Pflege der Benutzer in der Bugtracking Anwendung minimiert werden.

Gewichtung: 3

6. Support

Unter dem Kriterium Support sind zwei Faktoren zu betrachten. Einerseits ist darunter der Reifegrad und die Verbreitung der jeweiligen Software zu verstehen. Der Reifegrad kann unter anderem anhand des Datums des letzten Versionsstandes, gepflegten FAQs und regelmäßigen Releases in periodischen Abständen gemessen werden. Die Verbreitung kann mittels Statistiken von Online-Plattformen wie beispielsweise Ohloh ermittelt werden. Diese Eigenschaften lassen Rückschlüsse auf die Zukunftsaussicht von Open Source Projekten schließen und geben den Entscheidern somit Sicherheit. Des Weiteren ist unter diesem Punkt der Einsatz von kommerziellen Supportmöglichkeiten zu verstehen. Serviceverträge mit Partnern sollen schwerwiegende und unternehmenskritische Probleme mit einer zeitnahen und fachlichen Kundenbetreuung absichern. Aus diesen Gründen ist dieses Kriterium als sehr wichtig einzustufen.

Gewichtung: 5

7. Aufwandserfassung

Mit Hilfe einer Aufwandserfassung können die Bearbeitungszeiten eines Projektes genau festgehalten werden. Diese Funktionalität unterstützt einerseits eine Überprüfung der Auslastung der Mitarbeiter in einem Projekt und andererseits die Ermittlung der Arbeitskosten eines Projektes, was wiederum für die Preiskalkulation des Vertriebes eine wichtige Rolle spielt.

Gewichtung: 5

8. Statistikfunktion

Statistiken dienen dem Zweck aussagekräftige Daten zu generieren. Mittels dieser Funktion ermöglicht es Anwendern beispielsweise Angaben zu der Effektivität und Effizienz der Bearbeitung von Fehlern zu treffen. Damit übernimmt dieses Feature eine Controllingfunktion.

Gewichtung: 3

9. Individualisierbarkeit

Unter diesem Kriterium ist eine integrierte Funktionalität zur individuellen Anpassung an die eigenen Bedürfnisse zu verstehen. Beispielsweise soll bei der Erfassung eines Fehlertickets zusätzliche Datenfelder mittels einer integrierten Funktion und nicht durch Änderungen am Quellcode eingefügt werden.

Gewichtung: 3

10. Sprache des Interfaces

Dieses Kriterium stellt eine weitere Ergänzung zu der Usability dar. Die Sprache des Interfaces kann für das allgemeine Verständnis zur Bedienung der Anwendung sehr zuträglich sein. Die Bedienung wird unerfahrenen Nutzern dadurch erheblich erleichtert. Somit sind die sonst notwendigen Fremdsprachenkenntnisse nicht zwingend erforderlich. Unter diesem Kriterium werden die für Deutschland nativen Sprachen berücksichtigt. Eine Unterstützung der deutschen Sprachen führt daher zu einer vollständigen Erfüllung des Kriteriums, während eine Unterstützung der englischen Sprache als Teilerfüllung angesehen wird.

Gewichtung: 1

4 Auswahl von Bug-Tracking Tools

Durch eine Recherche auf der Online Plattform Ohloh.net, welche sich der Kategorisierung von Open Source Projekten annimmt, konnte früh festgestellt werden, dass derzeit über 70 Open Source Projekte existieren, die sich mit der Entwicklung von Bug-Tracking Tools beschäftigen. Da eine Betrachtung aller Bug-Tracking Tools dieser Menge im Rahmen dieser Projektarbeit nicht zu leisten und darüber hinaus auch nicht zielführend ist, findet zu Anfang dieses Kapitels eine Vorselektion statt. Die selektierten Tools werden im Anschluss in Bezug auf ihre Eigenschaften rudimentär vorgestellt. Nachfolgend wird erläutert, nach welchem Muster vorgegangen wird, um die einzelnen Bug-Tracking Tools in Kapitel 5 zu bewerten.

4.1 Vorselektion

Als Basis für die Vorselektion wurden neben der bereits erwähnten Online Plattform Ohloh.net, weitere drei Websites (softwaretestinghelp.com; webmasterpro.de; softwaretestingclass.com) zur Recherche herangezogen. Das Ziel war dabei die Auswahl von insgesamt vier Bug-Tracking Tools, um eine ausführliche Betrachtung im Rahmen dieser Projektarbeit zu ermöglichen. Ohloh.net, softwaretestinghelp.com, webmasterpro.de und softwaretestingclass.com stellen zu diesem Zweck jeweils ein Ranking zur Verfügung. In Bezug auf die Vorselektion wurde für die einzelnen Bug-Tracking Tools jeweils ein Punkt für die Erwähnung in der Top 4 einer Website vergeben. Das Ergebnis dieser Analyse ist in Abb. 4 dargestellt.

Erwähnungen in der Top 4

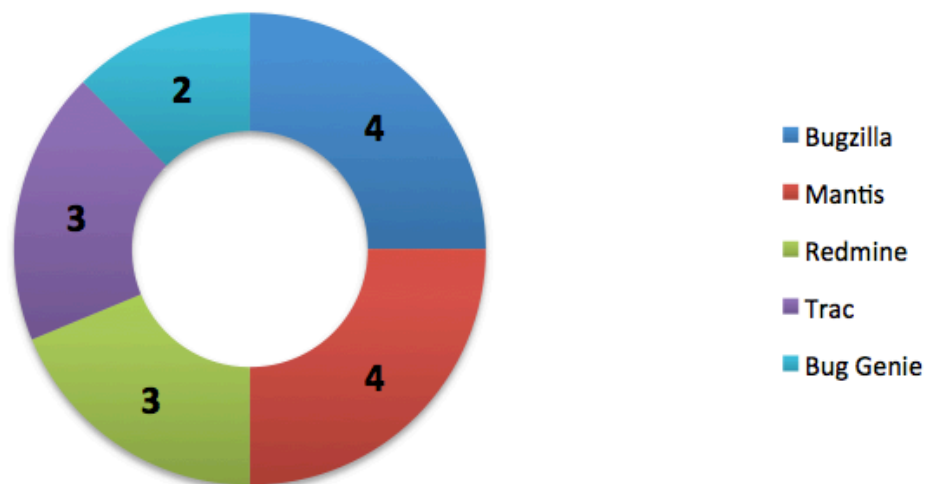


Abb. 4: Ranking von Bug-Tracking Tools

Abb. 4 veranschaulicht, dass im Top 4 Ranking der verschiedenen Websites insgesamt fünf unterschiedliche Bug-Tracking Tools vertreten sind. Bugzilla und Mantis werden in allen der vier betrachteten Rankings berücksichtigt und erhalten deshalb jeweils vier Punkte. Die Tools Redmine und Trac sind jeweils drei Mal in den Rankings vertreten. Lediglich zwei Top 4 Rankings enthalten das Tool Bug Genie. Für den weiteren Verlauf dieser Ausarbeitung hat die Analyse der Rankings zur Folge, dass in Kapitel 5 die Bug-Tracking Tools Bugzilla, Mantis, Trac und Redmine bezüglich der Bewertung berücksichtigt werden. An dieser Stelle wird hervorgehoben, dass

das Bug-Tracking Tool Bug Genie in dieser Projektarbeit nur aus Gründen des begrenzten Umfangs keine Berücksichtigung findet.

4.2 Vorstellung der selektierten Bug-Tracking Tools

Dieser Abschnitt dient dazu, die Bug-Tracking Tools, die aus der Vorselektion in Abschnitt 4.1 hervorgegangen sind, in Bezug auf ihrer Hintergrundinformationen grundlegend zu beschreiben. Dabei wird vor allem auf die Lizenzmodelle und die Systemvoraussetzungen der einzelnen Tools eingegangen.

4.2.1 Bugzilla

Bugzilla ist ein webbasiertes Bug-Tracking Tool welches auf einem Webserver mit CGI Unterstützung installiert werden kann. Eine weitere Voraussetzung für den Betrieb von Bugzilla ist eine Datenbank. Diesbezüglich ist Bugzilla mit MySQL, PostgreSQL und Oracle kompatibel. Ursprünglich wurde Bugzilla als firmeninternes Fehlerverwaltungstool bei Netscape Communications eingesetzt, bevor das Unternehmen den Quellcode freigab. Die freigegebene Version 2.0 wurde von der Mozilla Foundation in die Programmiersprache Perl portiert, in der das Tool bis heute weiterentwickelt wird. Bugzilla unterliegt derzeit den drei Lizenzen MPL, GPL und LGPL.⁴⁶

Namenhafte Open Source Projekte, die Bugzilla als Bug-Tracking Tool einsetzen sind beispielsweise Linux Kernel, Eclipse, GNOME, Libre Office und OpenSSH.⁴⁷

4.2.2 Mantis

Mantis ist ebenfalls ein webbasiertes Bug-Tracking Tool, welches die Verwendung eines Webserver voraussetzt. Unterstützte Server sind beispielsweise IIS und Apache. Als Datenbanken eignen sich für Mantis unter anderem MySQL, MS SQL und PostgreSQL. Das Bug-Tracking Tool wird in der Programmiersprache PHP entwickelt. Mantis unterliegt der GNU General Public License.⁴⁸

⁴⁶ Vgl. Webmasterpro (o.J.)

⁴⁷ Vgl. Bugzilla (2014)

⁴⁸ Vgl. Mantis BT (2014):

4.2.3 Trac

Das Open Source Projekt Trac ist ein Bug-Tracking Tool, welches neben dem klassischen Ticketsystem zusätzlich ein Wiki beinhaltet. Trac ist ebenfalls webbasiert und kann beispielsweise in Verbindung mit den Webservern IIS oder Apache eingesetzt werden. Kompatible Datenbanken für den Einsatz mit Trac sind MySQL, SQLite und PostgreSQL. Das Bug-Tracking Tool unterliegt der BSD Lizenz.

4.2.4 Redmine

Redmine ist, wie die drei bereits beschriebenen Bug-Tracking Tools, ebenfalls webbasiert. Ähnlich wie Trac stellt Redmine dem Benutzer neben seiner Funktionalität als Bug-Tracking Tool ein Wiki zur Verfügung. Das Bug-Tracking Tool Redmine wird in der Programmiersprache Ruby entwickelt. Hierbei wird das Framework Ruby on Rails verwendet. Das Tool unterliegt der GNU General Public License.

Redmine ist unter anderem bei den Open Source Projekten Ruby und TYPO3 im Einsatz.

4.3 Vorgehensmuster bei der Bewertung

Die Bewertung der selektierten Tools in Kapitel 5 erfolgt in zwei Schritten. Der erste Schritt der Bewertung beinhaltet die Prüfung der Bug-Tracking Tools auf sogenannte Pflichtkriterien. Diese wurden im Vorfeld im Kapitel 3.3 aus der ITIL abgeleitet. Bei den Pflichtkriterien handelt es sich dabei um Kriterien, die ein Bug-Tracking Tool zwingend erfüllen muss, um sich für ein Unternehmen zu eignen. Der zweite Schritt beinhaltet eine Bewertung der Bug-Tracking Tools auf der Basis von Vergleichskriterien. Diese wurden ebenfalls bereits in Kapitel 3.3 von ITIL abgeleitet. Für den Erfüllungsgrad von einzelnen Vergleichskriterien erhalten die Bug-Tracking Tools eine bestimmte Anzahl von Punkten. Die genaue Anzahl ergibt sich dabei aus dem Erfüllungsgrad in Kombination mit dem Gewichtungsfaktor des jeweiligen Vergleichskriteriums. Die Summe der Punkte bezüglich der einzelnen Vergleichskriterien ergibt eine Gesamtpunktzahl für jedes selektierte Bug-Tracking Tool. Auf Basis dieser Kennzahl wird eine Empfehlung für Unternehmen ausgesprochen.

5 Test ausgewählter Bug-Tracking Tools

Für den Vergleichprozess bei der Auswahl der Bug-Tracking Tools wurden für die Erfüllungsgrade des jeweiligen Kriteriums drei Möglichkeiten erstellt:

- ✓ = 10 bei erfüllt
- (✓) = 5 bei teilerfüllt
- ✗ = 0 bei nicht erfüllt

Die Punkte der Gewichtung und der Erfüllungsgrade wurden in dieser Ausarbeitung bewusst in dieser Skala definiert, um eine größere Streubreite zu erzeugen. Dadurch sollte sofern möglich ein eindeutiger Sieger ermittelt werden. Der Kriterienkatalog und das Bewertungsschema dienen lediglich als eine Richtlinie und können für einen Einsatz entsprechend den Bedürfnissen des Unternehmens angepasst werden.

5.1 Trac

5.1.1 Installation

Für das Bug-Tracking Tool Trac existieren mehrere Oneclick-Installer, welche von Drittanbietern bereitgestellt werden. Für eine unternehmensinterne Installation wird jedoch allgemein eine Installation über die Kommandozeile empfohlen. Voraussetzung für die Installation des Bug-Tracking Tools sind die Komponenten Datenbank, Webserver und Python.

Um die eigentliche Installation von Trac maßgeblich zu vereinfachen, wird empfohlen, die Setuptools der jeweiligen Python Version zu verwenden. Neben den Setup Tools wird des Weiteren die Python Bibliothek Genshi und das Toolkit Babel benötigt. Genshi ist unter anderem verantwortlich für das Parsen und Rendern von HTML-Dateien. Genshi ist eine Python-spezifische Bibliothek. Das Toolkit Babel ermöglicht eine deutsche Sprachunterstützung für Trac.⁴⁹

⁴⁹ Vgl. Sweo (2011)

Durch den Befehl `easy_install Trac` in der Python Shell wird Trac automatisiert installiert.⁵⁰ Die Konfiguration von Trac erfolgt ebenfalls über die Python Shell. Vereinzelte Konfigurationsmöglichkeiten sind über das Webinterface durchzuführen.

In der folgenden Abbildung sind die Konfigurationsmöglichkeiten von Trac in der Benutzeroberfläche dargestellt. Hierbei ist zu erwähnen, dass sich die Oberfläche als sehr intuitive gestaltet und durch eindeutige Reiternamen den Anwender schnell an die Bedienung der Anwendung gewöhnt. Allerdings kann nicht alles über die grafische Oberfläche verwaltet werden, sodass gelegentlich die Kommandozeile aushelfen muss.

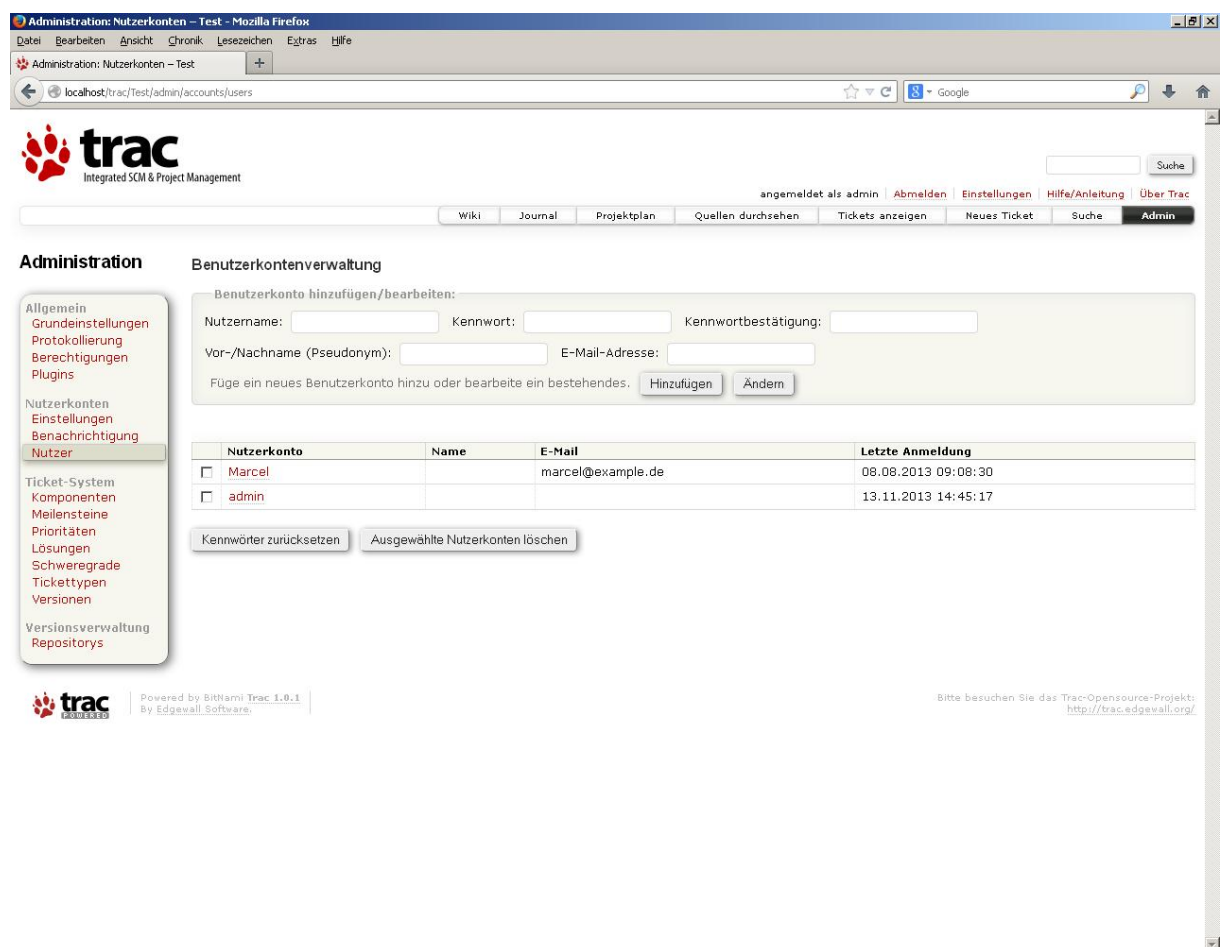


Abb. 5: Administrationsbereich in Trac

⁵⁰ Vgl. Trac (2014)

5.1.2 Pflichtkriterien

Pflichtkriterium	Erfüllungsgrad
1. Open Source	✓
2. Webinterface	✓
3. Definierbarer Workflow	✓
4. Klassifizierung	✓
5. Such-/Sortierfunktion	✓
6. Benachrichtigungsfunktion	✓
7. Änderungsverlauf	✓
8. Priorisierung	✓
9. Benutzerverwaltung	✓
10. Dateianhänge	✓
11. Mandantenfähig	✓
Ergebnis:	Vollständig erfüllt

Tabelle 1: Trac : Anwendung der Pflichtkriterien

5.1.3 Vergleichskriterien

Vergleichskriterium	Gewichtung	Erfüllungsgrad
1. Usability	5	X
2. Sprache des Interfaces	1	✓
3. Plattformunabhängigkeit	1	✓
4. Hilfefunktion	3	✓
5. Support	5	✓

6. Individualisierbarkeit	3	✓
7. LDAP-Anbindung	3	X
8. Versionskonzept	3	✓
9. Aufwandserfassung	5	X
10. Statistikfunktion	3	X
Erreichte Gesamtpunktzahl		160

Tabelle 2: Trac: Anwendung der Vergleichskriterien

5.1.4 Gesamteindruck

An dieser Stelle ist hervorzuheben, dass Trac kein konventionelles Bug-Tacking Tool ist. Durch die Darstellung der Benutzeroberfläche als Wiki verfolgt Trac das Ziel eine Art Projekt-Portal zu schaffen. Für Benutzer, die jedoch ausschließlich ein Tool für die Fehlerverwaltung suchen, ist die Oberfläche von Trac an vielen Stellen als umständlich zu betrachten. Ob die Verbindung von allgemeinen Projektinformationen mit Bug-Reports sinnvoll ist, muss unternehmensspezifisch entschieden werden. Für Unternehmen die eine Projektverwaltung möglichst zentral durch eine Software abdecken möchten lohnt sich ein Blick auf das Tool Trac.

5.2 Bugzilla

5.2.1 Installation

Die Installation von Bugzilla auf allen gängigen Plattformen ist auf der Website des Opensource Projekt in der Kategorie Docs ausführlich beschrieben. An gleicher Stelle wird ebenfalls auf die Voraussetzungen des Bug-Tracking Tools verwiesen, welche nachfolgend nochmals aufgelistet werden. Neben den beiden Standard Tools einer webbasierten Software (Webserver und Datenbank) erfordert Bugzilla die Programmiersprache Perl und einen Mail Client um die Benachrichtigungsfunktion des Tools gewährleisten zu können.⁵¹

⁵¹ Vgl. Bugzilla (o.J.)

In Bezug auf die Installationsanleitung von Bugzilla ist anzumerken, dass sich die Anleitung auf eine Installation auf einem Apache Webserver bezieht. Ob sich die Anleitung auch für die Installation des Bug-Tracking Tools auf einen anderen Webserver eignet muss im Vorhinein geprüft werden.

Sofern die vorausgesetzten Tools installiert und bezüglich Bugzilla konfiguriert wurden, kann die aktuelle Version von Bugzilla beispielsweise auf der Website des Opensource Projekts heruntergeladen werden. Alternativ kann Bugzilla direkt über einen Konsolenbefehl in ein Verzeichnis des Webserver geladen werden. Um Bugzilla lauffähig zu machen müssen in einem letzten Schritt diverse Perl Module installiert werden, die das Bug-Tracking Tool benötigt.⁵² Nach erfolgreicher Installation der Module kann Bugzilla in einem Browser über das jeweilige Webserververzeichnis aufgerufen werden. Über das Webinterface kann das Tool konfiguriert werden. Bezeichnend hierfür ist ein Ausschnitt der Benutzerkonfiguration in der folgenden Abbildung dargestellt. Bugzilla bietet einen hohen Grad der Anpassbarkeit ist jedoch für unerfahrene und nicht technisch versierte Anwender schwer zu handhaben.

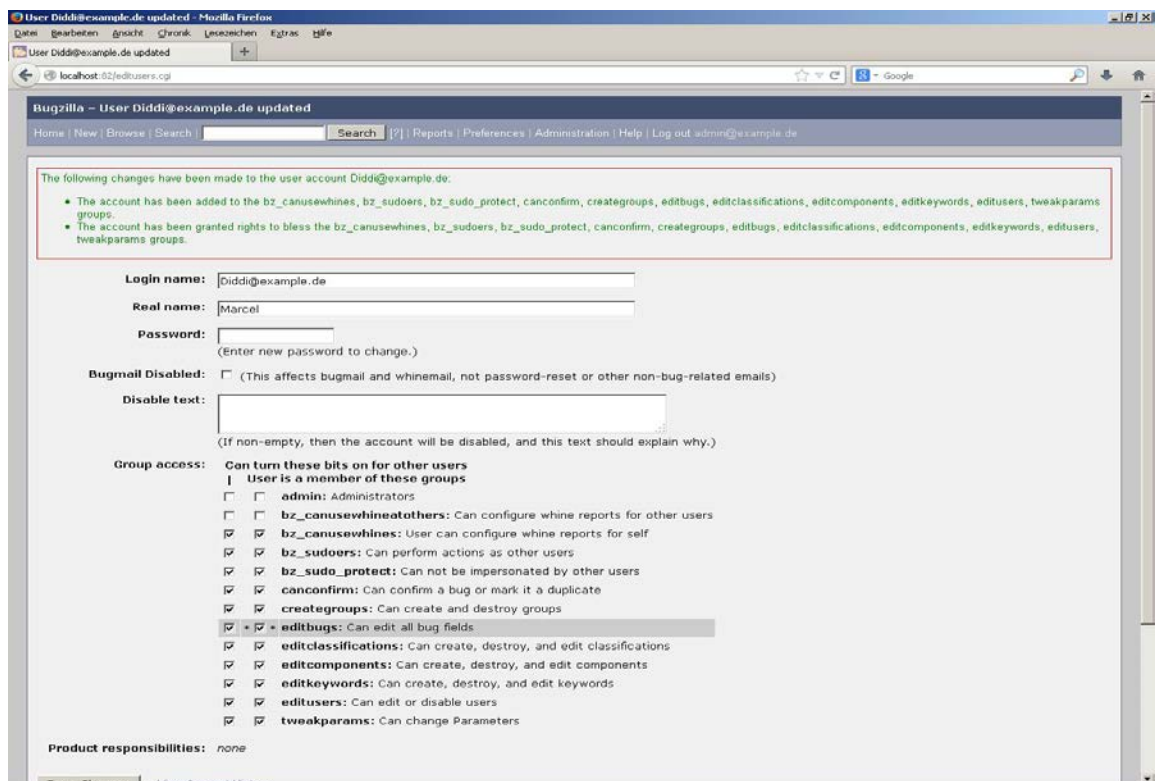


Abb. 6: Benutzerverwaltung in Bugzilla

⁵² Vgl. Bugzilla (o.J.)

5.2.2 Pflichtkriterien

Pflichtkriterium	Erfüllungsgrad
1. Open Source	✓
2. Webinterface	✓
3. Definierbarer Workflow	✓
4. Klassifizierung	✓
5. Such-/Sortierfunktion	✓
6. Benachrichtigungsfunktion	✓
7. Änderungsverlauf	✓
8. Priorisierung	✓
9. Benutzerverwaltung	✓
10. Dateianhänge	✓
11. Mandantenfähig	✓
Ergebnis:	Vollständig erfüllt

Tabelle 3: Bugzilla: Anwendung der Pflichtkriterien

5.2.3 Vergleichskriterien

Vergleichskriterium	Gewichtung	Erfüllungsgrad
1. Usability	5	X
2. Sprache des Interfaces	1	(✓)
3. Plattformunabhängigkeit	1	✓
4. Hilfefunktion	3	(✓)

5. Support	5	✓
6. Individualisierbarkeit	3	✓
7. LDAP-Anbindung	3	✓
8. Versionskonzept	3	✓
9. Aufwandserfassung	5	✓
10. Statistikfunktion	3	✓
Erreichte Gesamtpunktzahl		250

Tabelle 4: Bugzilla: Anwendung der Vergleichskriterien

5.2.4 Gesamteindruck

Der wichtigste Aspekt, der bei einer Gesamtbetrachtung von Bugzilla zwingend genannt werden muss, ist die schwierige Installation des Bug-Tracking Tools. Für Benutzer ohne Konsolen Kenntnisse ist die Installation von Bugzilla auch mit der Installationsanleitung des OpenSource Projektes nicht möglich. Von diversen Oneclick Installern für Bugzilla, die eine potentielle Erleichterung der Installation angeben, wird in der Community strikt abgeraten. Um Bugzilla ohne wiederkehrende Fehler in Betrieb zu setzen, sollte das Tool über die Konsole installiert werden.

Bugzilla ist insgesamt als Bug-Tracking Tool mit einem maximalen Grad der Anpassbarkeit zu definieren. Bugzilla eignet sich deshalb besonders für Unternehmen, die sich bei der Definition ihres Softwareentwicklungs-Workflows und ihrer Rollenkonzepte nicht einschränken lassen wollen. Bugzilla bietet zusätzlich die Möglichkeit benutzerdefinierte Fehlerstatus zu definieren. Auch die Definition von eigenen Datenfeldern für Fehlertickets wird von Bugzilla unterstützt.

Des Weiteren ist zu erwähnen, dass der hohe Grad der Anpassbarkeit von Bugzilla sich teilweise auf den Aspekt der Usability auswirkt. Für Benutzer wirken die Erfassungsmasken für die Fehlertickets oftmals überladen.

5.3 Mantis

5.3.1 Installation

Da Mantis ein plattformunabhängiges Bug-Tracking Tool ist, kann es unter allen gängigen Betriebssystemen wie etwa Linux, Mac OS X, Windows und sogar OS/2 eingesetzt werden. Die Installation von Mantis ist dabei sehr einfach und benutzerfreundlich gestaltet. Die Installation erfolgt wie die Bedienung des Bug-Tracking Tools ebenfalls über die Benutzeroberfläche. Dies trifft sowohl auf die clientseitige als auch auf die serverseitige Installation zu.

Für den Betrieb von Mantis sind als vorausgesetzte Komponenten eine Datenbank ein Webserver und PHP erforderlich. Vor der Installation des Bug-Tracking Tools müssen deshalb in einem ersten Schritt die genannten Komponenten installiert werden. Um Mantis zu installieren kann die aktuelle Version über Sourceforge.net heruntergeladen werden. Die erhaltene Archiv Datei wird nun in ein Verzeichnis des jeweiligen Webserver entpackt. Bereits zu diesem Zeitpunkt kann Mantis mit Hilfe eines Browsers über das gewählte Verzeichnis aufgerufen werden. Beim erstmaligen Aufrufen von Mantis startet der Installationsmodus welcher die Verknüpfung von Mantis mit der verwendeten Datenbank ermöglicht. Hierzu müssen die Datenbankparameter, der zuvor angelegten Datenbank angegeben werden.⁵³

⁵³ Vgl. Schoyswohl, P. (2011)

Checking Installation...	
Checking PHP version (your version is 5.3.3-7+squeeze1)	GOOD
Checking if safe mode is enabled for install script	GOOD

Installation Options	
Type of Database	MySQL (default)
Hostname (for Database Server)	localhost
Username (for Database)	mantis
Password (for Database)
Database name (for Database)	mantisbt
Admin Username (to create Database if required)	
Admin Password (to create Database if required)	
Print SQL Queries instead of Writing to the Database	<input type="checkbox"/>
Attempt Installation	Install/Upgrade Database

Abb. 7: Mantis: Installationsmodus

Nach einer erfolgreichen Installation von Mantis, erreicht man über das Mantis Verzeichnis zukünftig die Login Maske. Nach dem Login über die Administratorkennung kann Mantis konfiguriert werden.

5.3.2 Pflichtkriterien

Pflichtkriterium	Erfüllungsgrad
1. Open Source	✓
2. Webinterface	✓
3. Definierbarer Workflow	✓
4. Klassifizierung	✓
5. Such-/Sortierfunktion	✓

6. Benachrichtigungsfunktion	✓
7. Änderungsverlauf	✓
8. Priorisierung	✓
9. Benutzerverwaltung	✓
10. Dateianhänge	✓
11. Mandantenfähig	✓
Ergebnis:	Vollständig erfüllt

Tabelle 5: Mantis: Anwendung der Pflichtkriterien

5.3.3 Vergleichskriterien

Vergleichskriterium	Gewichtung	Erfüllungsgrad
1. Usability	5	✓
2. Sprache des Interfaces	1	✓
3. Plattformunabhängigkeit	1	✓
4. Hilfefunktion	3	✓
5. Support	5	✓
6. Individualisierbarkeit	3	✓
7. LDAP-Anbindung	3	(✓)
8. Versionskonzept	3	✓
9. Aufwandserfassung	5	✓
10. Statistikfunktion	3	✓
Erreichte Gesamtpunktzahl		305

Tabelle 6: Mantis: Anwendung der Vergleichskriterien

5.3.4 Gesamteindruck

Mantis ist ein Bug-Tracking Tool welches einen hohen Grad der Anpassbarkeit bietet. Interessant ist hierbei, dass die einzelnen Anpassungen benutzerfreundlich über das Webinterface durchzuführen sind. Hierzu sind keine Kenntnisse im Bereich der Kommandozeilenbefehle notwendig. Insgesamt ist Mantis sehr stark auf die Benutzerfreundlichkeit ausgerichtet. Die einzelnen Masken wirken klar strukturiert und sind nicht mit Informationen überladen. Besonders hervorzuheben ist die simple Installation bzw. Integration des Bug-Tracking Tools Mantis.

5.4 Redmine

Redmine ist ein webbasiertes Open Source-Tool, welches zur Unterstützung und Organisation von Projekten eingesetzt wird. Es handelt sich dabei um ein Paket, welches mehrere Module beinhaltet, die das Projektmanagement unterstützen. Primär ist Redmine durch das integrierte Issue-Tracking-System bekannt. Das Tool deckt durch die hohe Anzahl an Komponenten ein breites Spektrum für Softwareprojekte ab, da es nicht ausschließlich auf Issue-Tracking fokussiert ist. Durch die breite Open Source-Community hat das Tool eine weite Verbreitung aufzuweisen und wurde in mehr als 45 Sprachen übersetzt. Eine deutschsprachige Version ist ebenfalls erhältlich. Die grafische Nutzeroberfläche von Redmine ist in der folgenden Abbildung zu sehen.

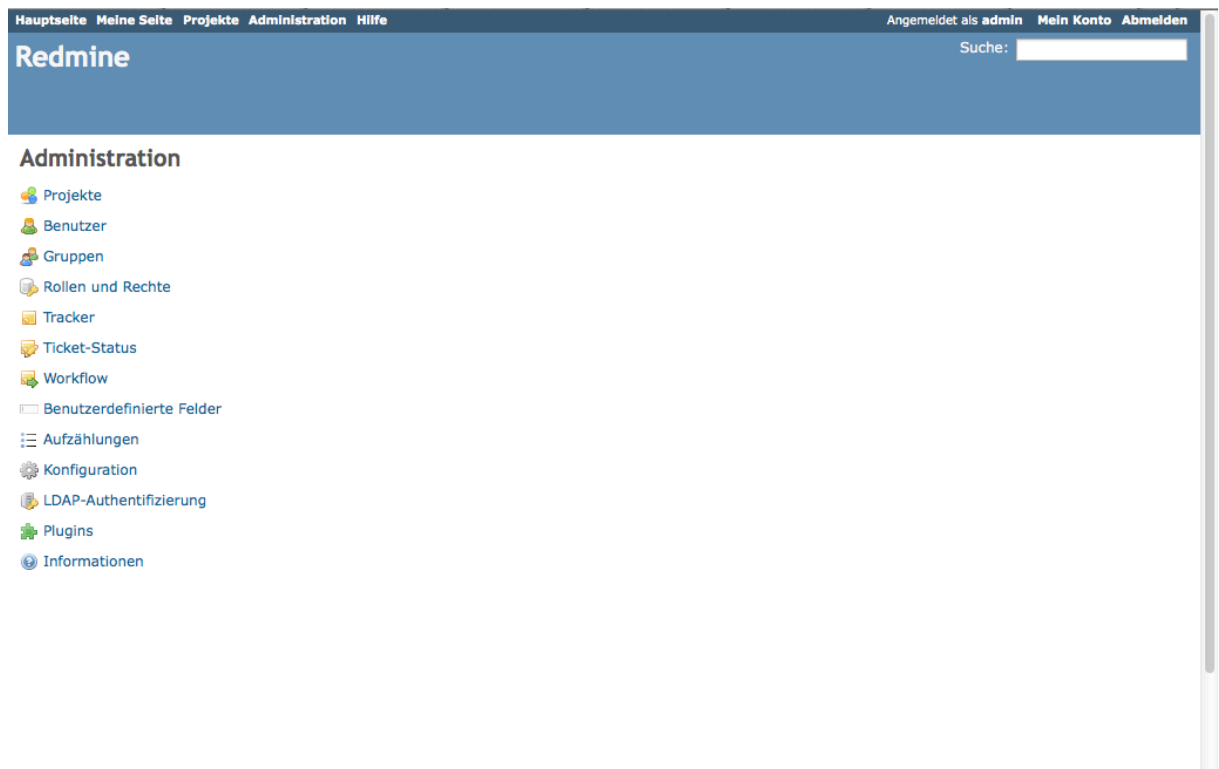


Abb. 8: Redmine: Startmenü

Der Funktionsumfang des Redmine-Basispakets wird nun im Folgenden aufgelistet.

- **Projektmanagement**
Redmine bietet die Möglichkeit, Aufgaben innerhalb des Tools zu verteilen und Termine festzusetzen. Verteilte Aufträge werden vom Werkzeug automatisch in den mitgelieferten Kalender geschrieben. Weitere grafische Unterstützung wird durch Balkendiagramme (z.B. Komplettierungsstatus) geliefert.
- **Issue-Tracking**
Die einzelnen Issues innerhalb von Projekten können individuell konfiguriert und angelegt werden.
- **Integriertes Wiki**
Für jedes Projekt kann ein gesondertes Wiki erstellt werden, welches als Diskussions- und Dokumentationsplattform dient
- **Mandantenfähigkeit**
Es können unterschiedliche Projekte innerhalb von Redmine angelegt werden. Jeder User kann dabei verschiedene Rollen in den einzelnen Projekten haben. Zudem kann ein User in mehreren Projekten involviert sein. Typische Rollen sind dabei Manager, Entwickler und Beobachter. Dem Administrator

steht es frei, weitere Rollen zu konfigurieren. So wird erreicht, dass Nutzer ausschließlich auf die Daten Einsicht erlangen, wofür sie berechtigt sind.

- **Persistenzdienste**
Die jeweiligen Datensätze werden in Datenbanken abgespeichert. Redmine unterstützt mannigfaltige Datenbanksysteme wie z.B. MySQL, SQL Server und PostgreSQL
- **LDAP-Fähigkeit**
Anbindung an einen bestehenden Verzeichnisdienst im Unternehmen. Über diese Funktion kann die Zugriffsberechtigung auf den Server für alle Mitarbeiter im Unternehmen geregelt werden. Bei entsprechender Konfiguration dürfen lediglich vorgesehene Nutzer über den Browser auf den Server zugreifen.
- **Benachrichtigungssystem**
Registrierte Nutzer erhalten beispielsweise Benachrichtigungen an ihre angegebene E-Mail-Adresse, sobald eine Änderung in einem Projekt vorliegt
- **Plugins**
Das Redmine Standard-Paket kann um weitere Plugins erweitert werden, die den Funktionsumfang des Softwarepakets weiter optimieren

Redmine obliegt der GNU GPLv2. Folglich steht es dem Unternehmen, welches das Tool im Einsatz hat frei, jegliche Änderungen im Quellcode vorzunehmen. Dieser ist auf der Webplattform redmine.org erhältlich. Die Anwendung kann somit uneingeschränkt abgeändert und speziell an die Gegebenheiten und Wünsche im Unternehmen angepasst werden. Zu beachten ist jedoch, dass eine eigens abgeänderte Version von Redmine ebenfalls unter der GNU GPL V2 zu veröffentlichen ist. Folglich müssen die eigens entwickelten Codezeilen der Open Source-Community zur freien Verfügung gestellt werden. Dies erweist sich im Unternehmensumfeld als kritisch, da gegebenenfalls Rückschlüsse über den infrastrukturellen Aufbau der IT gezogen werden können.

5.4.1 Installation

Redmine gilt als plattformunabhängiges Issue-Tracking-System. Es kann folglich auf Windows, Linux und Mac-Plattformen betrieben werden.

Das Tool kann auf unterschiedlichen Wegen installiert werden. Für jede der angegebenen Plattformen ist ein Bitnami Installer verfügbar, welcher alle notwendigen Komponenten vorinstalliert. Dazu zählen unter anderem Apache, ein MySQL-Server sowie einen Anwendungsserver. Ebenso ist es möglich, das System über die Kommandozeile zu implementieren. In diesem Fall sind die Anwendungen für den Web- und Applikationsserver jedoch gesondert zu installieren, da diese im Redmine-Basispaket nicht integriert sind.

5.4.2 Pflichtkriterien

Pflichtkriterium	Erfüllungsgrad
1. Open Source	✓
2. Webinterface	✓
3. Definierbarer Workflow	✓
4. Klassifizierung	✓
5. Such-/Sortierfunktion	✓
6. Benachrichtigungsfunktion	✓
7. Änderungsverlauf	✓
8. Priorisierung	✓
9. Benutzerverwaltung	✓
10. Dateianhänge	✓
11. Mandantenfähig	✓
Ergebnis:	Vollständig erfüllt

Tabelle 7: Redmine: Anwendung der Pflichtkriterien

5.4.3 Vergleichskriterien

Vergleichskriterium	Gewichtung	Erfüllungsgrad
---------------------	------------	----------------

1. Usability	5	✓
2. Sprache des Interfaces	1	✓
3. Plattformunabhängigkeit	1	✓
4. Hilfefunktion	3	✓
5. Support	5	✓
6. Individualisierbarkeit	3	✓
7. LDAP-Anbindung	3	✓
8. Versionskonzept	3	✓
9. Aufwandserfassung	5	✓
10. Statistikfunktion	3	✓
Erreichte Gesamtpunktzahl		320

Tabelle 8: Redmine: Anwendung der Vergleichskriterien

5.4.4 Gesamteindruck

Redmine bietet dem Nutzer eine intuitive grafische Oberfläche. So können jegliche Konfigurationen und Arbeitsschritte über die GUI im Browser durchgeführt werden. Es sind folglich keine tiefgehenden Kenntnisse wie z.B. die Benutzung der Kommandozeile notwendig. Dennoch weist das System einen hohen Funktionsumfang auf, welcher sich durch Plugins kontinuierlich erweitern und anpassen lässt. Eben durch dieses breite Spektrum ist die Einführung eines einheitlichen Systems möglich, da sämtliche für Softwareprojekte relevante Komponenten bereits integriert sind.

5.5 Zusammenfassung der Bewertung

In Tabelle 9 ist die gesamte Bewertungsmatrix mit den auf die einzelnen Produkte angewendeten Pflichtkriterien zu sehen. Diese dient zur verbesserten Übersicht im Vergleich zwischen den einzelnen Tools.

Pflichtkriterium	Mantis	Bugzilla	Trac	Redmine
1. Open Source	✓	✓	✓	✓
2. Webinterface	✓	✓	✓	✓
3. Definierbarer Workflow	✓	✓	✓	✓
4. Klassifizierung	✓	✓	✓	✓
5. Such-/Sortierfunktion	✓	✓	✓	✓
6. Benachrichtigungsfunktion	✓	✓	✓	✓
7. Änderungsverlauf	✓	✓	✓	✓
8. Priorisierung	✓	✓	✓	✓
9. Benutzerverwaltung	✓	✓	✓	✓
10. Dateianhänge	✓	✓	✓	✓
11. Mandantenfähig	✓	✓	✓	✓

Tabelle 9: Zusammenfassung: Anwendung der Pflichtkriterien

Analog zur obigen Übersicht ist in Tabelle 10 die Gegenüberstellung der einzelnen Tools in Bezug auf die jeweiligen Vergleichskriterien zu sehen.

Vergleichsanforderung	Gewichtung	Mantis	Bugzilla	Trac	Redmine
1. Usability	5	✓	✗	✗	✓
2. Sprache des Interfaces	1	✓	✓	✓	✓
3. Plattformunabhängigkeit	1	✓	✓	✓	✓

4. Hilfefunktion	3	✓	(✓)	✓	✓
5. Support	5	✓	✓	✓	✓
6. Individualisierbarkeit	3	✓	✓	✓	✓
7. LDAP-Anbindung	3	(✓)	✓	✗	✓
8. Versionskonzept	3	✓	✓	✓	✓
9. Aufwandserfassung	5	✓	✓	✗	✓
10. Statistikfunktion	3	✓	✓	✗	✓
Erreichte Gesamtpunktzahl von möglichen 320 Punkten		305	250	160	320

Tabelle 10: Zusammenfassung: Anwendung der Vergleichskriterien

6 Vergleich: Open Source Bug-Tracking Tools mit kommerziellen Bug-Tracking Tools

Neben den Open Source-Tools für das Issue-Tracking existieren weitere Produkte, die aus dem kommerziellen Umfeld stammen. In diesen Fällen ist der Lizenznehmer verpflichtet, eine Lizenzgebühr an das Softwareunternehmen zu entrichten. Des Weiteren ist es in der Regel nicht möglich, den Quellcode der Anwendung einzusehen und zu verändern, da das Urheberrecht beim Softwareunternehmen liegt. So sind gegebenenfalls notwendige Anpassungen an das unternehmensinterne Netz unter Umständen nur bedingt möglich. Es existieren dabei unterschiedliche Grundprinzipien hinsichtlich der Tarifierung derartiger Systeme. Die Tarifierung dieser Produkte kann je nach Anzahl der Nutzer, die für das Issue-Tracking-System vorgesehen sind, variieren. Eine weitere Möglichkeit ist das bedarfsgerechte Lizenzmodell. In diesem Fall wird nicht die tatsächliche Anzahl an Usern berechnet, sondern die Anzahl an Nutzern, die gleichzeitig das System verwenden. Grundsätzlich ist zu sagen, dass die Lizenzgebühr bei einer größeren Anzahl von potenziellen Nutzern wächst. Dieses Kapitel zeigt den Funktionsumfang von kommerzieller Software im Vergleich zu Open Source-Produkten auf. Die einzelnen Produkte werden dabei nicht ausführlich getestet, sondern lediglich vorgestellt, da der Fokus dieser Arbeit auf Open-Source-Tools gerichtet ist.

Durch den Einsatz von proprietärer Software kann das Unternehmen bei Problemfällen an den Softwarehersteller direkt wenden. So wird gezielte Unterstützung ermöglicht, da es sich bei den kommerziellen Produkten um standardisierte zu implementierende Anwendungen handelt. Der Service ist teilweise in den Lizenzkosten enthalten, jedoch kann eine gesonderte Gebühr verlangt werden.

Grundsätzlich besteht die Möglichkeit, auch beim Einsatz von Open Source-Tools einen Service von spezialisierten Dienstleistern in Anspruch zu nehmen. Den Unternehmen bleibt jedoch ein Spielraum, wie das Issue-Tracking-System aufgebaut wird. So erweisen sich nicht alle Servicedienstleister in Problemfällen als passend. Die breite Open Source-Community ermöglicht auf Internet-Plattformen den Austausch von Erfahrungen und die Aneignung von Know-How. Auf diesem Weg kann kostenfreie Hilfestellung beansprucht werden. So können sich Mitarbeiter auf diesen Portalen anmelden und aktiv beteiligen. Als kritischer Punkt ist in diesem Fall jedoch der Datenschutz aufzuführen. Es ist zu beachten, dass keinerlei unternehmensinterne Daten auf derartige Plattformen gelangen.

Im Zuge der Betrachtung von kommerziellen Lösungen werden folgende Produkte betrachtet und jeweils kurz vorgestellt:

- Jira
- Adminitrac
- Omnitacker

Die Auswahl dieser drei Produkte resultiert aus den Tendenzen des Google Trends-Services. Es handelt sich dabei um ein Analysetool von Google, welches die Häufigkeit von Suchbegriffen in der Suchmaschine auswertet. Folglich wurde in vergangener Zeit häufig nach eben diesen Produkten recherchiert.

Jira

Die Software Jira ist ein Issue-Tracking-System des Software-Unternehmens Atlassian. Atlassian bietet dabei unterschiedliche Grundprinzipien an, wie das Produkt Jira bereitgestellt wird. Das Unternehmen, welches die Software beansprucht, kann das Issue Tracking-System sowohl auf den unternehmenseigenen Servern hosten, als

auch on-demand auf den Servern von Atlassian. Die Entscheidung zwischen diesen beiden Prinzipien ist je nach Unternehmenspolitik zu treffen. Die Auslagerung von Systemen führt zwangsläufig dazu, dass interne Datenbestände in die Hände von Dritten gelangen. Dies stellt folglich ein datenschutzrechtliches Problem dar. Ein weiterer Faktor ist die Verfügbarkeit des Systems. Sofern das Unternehmen die on-demand-Lösung in Anspruch nimmt, ist die Verfügbarkeit sowohl vom bereitstellenden Dienstleister als auch von der Netzwerkverbindung zu diesen Servern abhängig. Die eigens gehostete Variante hingegen kann in der unternehmenseigenen Netzwerkinfrastruktur eingebunden und gegebenenfalls mit Hochverfügbarkeitskomponenten ausgestattet werden. In Abb. 9 ist eine Auflistung der jeweiligen Vorteile beider Varianten zu sehen.

Download oder OnDemand? Der Vergleich

☁️ **ONDEMAND VORTEILE**

- Gehostet in der Cloud.
- Sofort verfügbar.
- Abonnementpreise: monatlich kündbar.
- Erweiterungen & Remote Integrationen.
- Verteilte Zusammenarbeit.

📥 **DOWNLOAD VORTEILE**

- Läuft auf Ihren Servern.
- Extrem konfigurierbar.
- Unbefristete Lizenz: einmalige Zahlung.
- Erweiterungen verfügbar vom Atlassian Marketplace.
- Komplette Kontrolle über die Software.

Abb. 9: Gegenüberstellung der Vorteile von On-Demand und dedizierten Lösungen⁵⁴

Für alle Jira-Versionen bietet der Hersteller eine kostenlose Testversion an, um das System testweise im Unternehmen einzuführen. Somit ist Jira 30 Tage lang vollwertig einsetzbar. Dies gilt sowohl für die on-demand, als auch für die eigens zu implementierende Variante.

Jira ist in drei unterschiedlichen Paketen erhältlich. Das Kernprodukt, welches in allen drei Varianten enthalten ist, stellt das Issue-Tracking-Tool als solches dar. Als Zusätzliches Feature wird *Jira Agile* in der zweiten Variante angeboten. Hier steht ein

⁵⁴ Enthalten in Atlassian (2014)

agiles Projektmanagement im Fokus. Als dritte Variante gilt die Kombination aus dem *Jira Issue-Tracker* und dem *Jira Service Desk*.



Abb. 10: Produktübersicht für Jira⁵⁵

Die drei unterschiedlichen Typen werden jeweils unterschiedlich tarifiert. Innerhalb dieser Ausarbeitung wird der Fokus jedoch lediglich auf das Issue Tracking-Tool gesetzt. In der folgenden Abb. 11 sind die Tarife nach der Anzahl an Nutzern gestaffelt zu sehen. In der obigen Variante ist die auf den Atlassian-Servern gehostete Variante zu sehen, in der unteren die eigens im Unternehmen betriebene. Für die Cloud-Variante sind monatliche Beträge zu bezahlen, während die Server-Variante lediglich einmalige Anschaffungskosten verlangt.

Cloud	10 Benutzer	15 Benutzer	25 Benutzer	50 Benutzer	100 Benutzer	500 Benutzer	2.000 Benutzer
Server	\$10/Monat	\$50/Monat	\$100/Monat	200\$/Monat	\$300/Monat	\$500/Monat	\$1,000/Monat

Cloud	10 Benutzer	25 Benutzer	50 Benutzer	100 Benutzer	500 Benutzer	2000 Benutzer	10,000 Benutzer	10,000+
Server	\$10 Starter	\$1,200	\$2,200	\$4,000	\$12,000	\$16,000	\$20,000	\$24,000

Abb. 11: Tarifierung des Jira-Produkts in der Cloud und auf dedizierten Servern⁵⁶

Das Softwareprodukt lässt sich über einen integrierten Store beliebig erweitert. Anhand von Addon-Produkten kann beispielsweise die Funktion ergänzt werden, Screenshots in Tickets einzufügen. Dieser modulare Aufbau ermöglicht einen ressourcensparendes Grundkonzept, welches bedarfsgerecht erweitert werden kann.

Die speziellen Features der Anwendung werden im Folgenden aufgelistet. Die Kernelemente wie z.B. das Anlegen eines Projekts oder Tickets gelten als Grundvoraus-

⁵⁵ Enthalten in Atlassian (2014)

⁵⁶ Enthalten in Atlassian (2014)

setzung für ein Issue Tracking-System und werden deshalb bei den einzelnen Produkten nicht erneut aufgeführt.

- Erweiterbarkeit über Store
- Definition von Workflows
- Import der Projekte von anderen Systemen (u.a. Redmine, Excel, Trac, Bugzilla)
- Interface auch für mobile Endgeräte (Smartphones etc.)⁵⁷

Adminitrac

Adminitrac ist ebenfalls in Form von einer 30-tägigen Trial-Version für Testzwecke erhältlich. Im Gegensatz zu Jira ist Adminitrac hingegen vollständig webbasiert und wird auf den Systemen des Software-Unternehmens ausgeführt. Es gibt somit keine Variante, die auf den unternehmenseigenen Servern gehostet werden kann. Dementsprechend ist lediglich die Beanspruchung des Services notwendig und Implementierung des Systems im Netzwerk des Unternehmens. Als notwendig stellen sich folglich nur eine Internetverbindung sowie ein Web-Browser heraus. Das Unternehmen gibt folglich analog zu Jira (im Fall des Cloud-Produkts) die Herrschaft über die Infrastruktur an den Anbieter ab.

Im Webauftritt ist ein Kalkulator verfügbar, anhand dem eine genaue Kalkulation der Lizenzkosten vorgenommen werden kann. Als erstes Kriterium ist die Anzahl potentieller Benutzer anzugeben. Des Weiteren können unterschiedliche Bezahlungsperioden ausgewählt werden. Abhängig von der Frequenz erhält der Kunde einen Rabatt. Dabei gelten höhere Rabatte bei längeren Rechnungsperioden.

Der Support wird vom Hersteller selbst durchgeführt und wird mit durch die Lizenzkosten finanziert. Da sich die Systeme im Hause von Adminitrac befinden ist kein Support vor Ort beim Kunden notwendig. Im Fall von Adminitrac ist die jeweilige Philosophie des Unternehmens hinsichtlich der Auslagerung von Daten zu beachten.

Der Fokus des Produkts liegt auf dem Issue-Tracking.

⁵⁷ Vgl. Atlassian (2014)

Adminitrac bietet ein Rollenkonzept, welches unterschiedliche Berechtigungen auf die einzelnen Nutzer verteilt.

✓	Plan Name	Included Users ¹	Issues and Projects	Email Support & Upgrades	Attachment Space Included (MB) ¹	Base Monthly Fee	With 15% Discount ²
<input type="radio"/>	Silver 1	5	Unlimited	Included	1,000	\$59.95	\$50.96
<input type="radio"/>	Silver 2	15	Unlimited	Included	2,500	\$177	\$150.45
<input type="radio"/>	Silver 3	30	Unlimited	Included	5,000	\$327	\$277.95
<input type="radio"/>	Gold 1	60	Unlimited	Included	10,000	\$645	\$548.25
<input type="radio"/>	Gold 2	100	Unlimited	Included	20,000	\$895	\$760.75
<input type="radio"/>	Gold 3	150	Unlimited	Included	30,000	\$1,305	\$1,109.25
<input type="radio"/>	Platinum 1	250	Unlimited	Included	50,000	\$2,125	\$1,806.25
<input type="radio"/>	Platinum 2	500	Unlimited	Included	100,000	\$4,125	\$3,506.25
<input type="radio"/>	Platinum 3	1,000	Unlimited	Included	200,000	\$8,000	\$6,800.00
<input type="radio"/>	Titanium	2,500	Unlimited	Included	500,000	\$18,750	\$15,937.50

Abb. 12: Tarifierung von Adminitrac⁵⁸

Die speziellen Features von Adminitrac werden in der folgenden Aufzählung nochmals aufgelistet:

- Vollständig webbasiert in der Cloud
- Anhängen von Dokumenten
- Rollenbasierte Zugriffskontrolle
- E-Mail Benachrichtigungsservice
- Verfügbare Application Programming Interface (API) zur Eigenentwicklung⁵⁹

Omnitracker

Das Tool *Omnitracker* ist ein Issue Tracking-Werkzeug des Unternehmens Omninet.

Der Aufbau des Produkts orientiert sich an ITIL und deckt folglich alle essenziellen Bestandteile der Best-Practice-Lösung hinsichtlich Defect Management ab.

Omnitracker unterscheidet sich im Lizenzmodell grundlegend von den anderen beiden kommerziellen Lösungen. Während bei Jira und Adminitrac die tatsächliche Anzahl an Nutzern im gesamten System tarifiert wird, berechnet Omnitracker lediglich die Anzahl an Benutzern, die sich zur selben Zeit im System befinden. Folglich kann eine bedarfsgerechte Lizenzierung vorgenommen werden, die an den tatsächlichen Be-

⁵⁸ Enthalten in Adminitrac (2014)

⁵⁹ Vgl. Adminitrac (2014)

trieb angelehnt ist. Zu den genauen Kosten sind jedoch keine genaueren Angaben frei zugänglich. Diese sind auf Anfrage auszuhandeln.

Omnitracker bietet sich für unterschiedliche Bereiche im Unternehmen an. Dazu zählen beispielsweise „Helpdesks und Call Center, Vertriebsorganisationen, Entwicklungsteams und andere Serviceleistende Organisationen“⁶⁰. Es werden dabei unterschiedliche Schablonen angeboten, welche je nach Verwendungszweck beansprucht und ins System mit eingebaut werden kann. Diese gestaltet die Nutzeroberfläche der jeweiligen Nutzer. Dadurch resultiert eine hohe Anpassbarkeit und Flexibilität für unterschiedliche Verwendungszwecke im Unternehmen. So ist für jeden Bereich eine gesonderte Schablone vorhanden. Neben vom Hersteller vordefinierten Schablonen kann der Kunde ebenfalls eigene Vorlagen entwickeln.

- Erweiterbarkeit über Store
- Mandantenfähigkeit
- Flexible Lizenzierung
- Kunden können in Produktplanung aktiv mitwirken
- Vertraglich festgehaltene Kompatibilität bei neuen Releases

Das Unternehmen bietet zudem Schulungen für die Administratoren und Benutzer des Systems an.⁶¹

Abschließende Gegenüberstellung Open Source / Kommerzielle Software

Abschließend ist zu sagen, dass sich der Funktionsumfang von Open Source-Tools und kommerziellen Softwareprodukten nur marginal unterscheidet. So wird dem Kunden bei kommerziellen Produkten die technische Betreuung größtenteils abgenommen. Der Hersteller gibt Hilfestellung bei der Einführung des Produkts und steht fortwährend bei unterschiedlichen Anliegen zur Verfügung. Alle der drei eben vorgestellten Produkte sind in Form von beim Anbieter gehosteten Cloud-Diensten verfügbar. Die Entscheidung, ob sich der Server im eigenen Unternehmensnetz oder in einer Cloud befinden soll, ist je nach Unternehmen selbst zu treffen.

Die Kosten für das Produkt resultieren bei den Cloud-Anbietern nicht allein aus den Lizenzkosten für die Software als solche. Sie setzt sich des Weiteren aus Hosting- und Service-Gebühren zusammen. Durch die Lokalisierung der Software beim Her-

⁶⁰ Omninet (2013)

⁶¹ Vgl. Omninet (2013)

steller direkt kann eine schnelle Reaktionszeit bei Problemen geboten werden. Die Beanspruchung von Service beim Einsatz von Open Source-Tools kann hingegen nicht in diesem geringen Zeitraum erfolgen.

7 Einführung eines Bug-Tracking-Tools im Unternehmen

Um das zuvor ausgewählte Bugtracking-System im Unternehmensnetz integrieren zu können, sind diverse Vorbereitungsmaßnahmen zu treffen. In dieser Hinsicht sind nicht allein die technischen Realisierungsmöglichkeiten entscheidend, sondern auch der Einbezug von den zukünftigen Nutzern des Systems. Zur Veranschlichung der Einführungsphase dient das ePAVOS-Modell in Abb. 13. Es umfasst sämtliche Schritte, welche bei der Einführung einer neuen Software zu berücksichtigen sind.

7.1.1 Konzeption der Vorbereitungsmaßnahmen

Es gilt dabei zu prüfen, ob die Nutzer mit Hilfe des Tools ihren Aufgaben vollwertig nachkommen können. Dies ist in Abb. 13 im Schritt *Test* einzuordnen. Zu erfassen sind die Einschätzungen der Nutzer anhand eines Probelaufs auf einem Testsystem. In dieser Phase genügt die Implementierung des Tools auf einem Standalone-System, welches über einen lokalen Webserver verfügt (z.B. XAMPP). Dieser Schritt zeigt sich in Abb. 13 unter dem Pfeil *Implementierung*. Es handelt sich dabei jedoch nicht um die finale Implementierung, sondern lediglich die eines gesonderten Testsystems. Den Software-Test nach einer kompletten Implementierung im Unternehmensnetzwerk ist mit Hinblick auf ein mögliches schlechtes Feedback bezüglich des Tools riskant und nicht zielführend. Die Mitarbeiter werden beim Softwaretest beauftragt die essenziellen Funktionen des Produkts zu nutzen. Dazu zählt beispielsweise das Anlegen eines Projekts oder Tickets. Zu erfassen sind die Testergebnisse beispielsweise anhand eines mündlichen Feedbacks oder über einen Fragebogen. Abhängig davon wie das Feedback der Nutzer ausfällt sind gegebenenfalls Schulungsmaßnahmen einzuleiten. Diese Phase erweist sich als essenziell, da ansonsten die Produktivität und Motivation der Nutzer beim System-Launch stark beeinträchtigt werden kann.

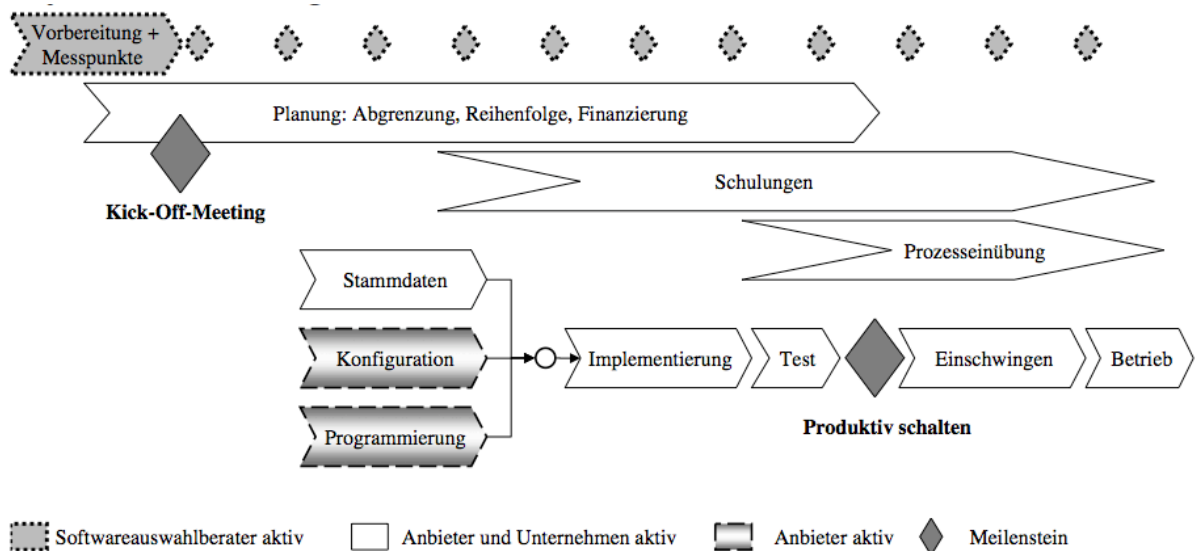


Abb. 13: Darstellung des ePAVOS-Modells⁶²

7.1.2 Konzeption des infrastrukturellen Aufbaus

Nachdem die Testphase erfolgreich abgeschlossen wurde sind die infrastrukturellen Gesichtspunkte der Produkteinführung zu betrachten.

Für das System wird ein Server benötigt, welcher den Dienst im Netzwerk anbietet und die jeweiligen Projekthinhalte speichert. Dies kann grundsätzlich auf unterschiedlichen Wegen realisiert werden. Als Beispiele hierfür sind die Installation auf dedizierter Serverhardware oder die Virtualisierung eines Serversystems auf einem Hypervisor. Durch den Betrieb auf einem gesonderten Server ist die Funktion des Systems von keiner weiteren Instanz abhängig, wie es bei einem Hypervisor der Fall ist. Folglich ist die ordnungsgemäße Verfügbarkeit des Issue-Tracking-Systems nur dann gewährleistet, solange der Hypervisor die virtualisierten Maschinen ausführen kann.

⁶² Enthalten in Teich, I./Kolbenschlag, W./Reiners, W. (2008), S.192

In Tabelle 11 werden die Vor- und Nachteile der beiden Herangehensweisen aufgelistet.

	Dedizierte Hardware	Virtualisierung
Vorteile	<ul style="list-style-type: none"> • Unabhängigkeit von anderen Systemen 	<ul style="list-style-type: none"> • Schnelle Bereitstellung • Skalierbarkeit • Effiziente Energienutzung
Nachteile	<ul style="list-style-type: none"> • Gesonderte Hardware muss beschafft werden • Erhöhter Energieverbrauch 	<ul style="list-style-type: none"> • Hohe Lizenzkosten für Hypervisor • Abhängigkeit vom Hypervisor

Tabelle 11: Gegenüberstellung von virtualisierten und dedizierten Systemen

Zusammenfassend ist zu sagen, dass die Entscheidung zwischen diesen beiden Herangehensweisen abhängig von der Philosophie eines jeden Unternehmens zu treffen ist. Sofern das Unternehmen bisher keine Vorkehrungen für die Virtualisierung von Servern besitzt, ist die Beschaffung eines solchen aufgrund des verhältnismäßig finanziellen hohen Aufwands bei der Ersteinrichtung nicht anzuraten. Beim Issue-Tracking-System handelt es sich nicht um ein kritisches System, dessen durchgängige Funktion essenziell für den Geschäftsbetrieb ist.

Für die Installation werden folgende Module benötigt:

- Webserver
- Datenbankserver
- Anwendungsserver

Mittels des Bitnami Redmine-Installers werden alle dieser relevanten Komponenten abgedeckt. Es handelt sich dabei um einen so genannten 1-Click-Installer. Durch die Verwendung dieses Assistenten ist sichergestellt, dass alle notwendigen Bestandteile vollständig auf das System portiert werden. So wird vermieden, dass bei der Erstinbetriebnahme einzelne Komponenten nicht oder nicht vollständig vorhanden sind. Dieser ist für Windows-, Linux- und Mac-Plattformen erhältlich. Alternativ kann die Anwendung über gesonderte Installationsprozesse auf dem Serversystem in Be-

trieb genommen werden. Diese können aus den technischen Dokumentationen auf den Homepages der jeweiligen Anbieter entnommen werden. Trotz des gesonderten Installationsprozesses der einzelnen Komponenten ist es möglich, diese auf einem einzigen Server zu implementieren.

Der Server benötigt für den Betrieb im Netzwerk eine eindeutige Internet Protocol (IP)-Adresse. Um einen vereinfachten Zugang zu ermöglichen, ist die Beanspruchung des Domain Name System (DNS)-Dienstes empfehlenswert. Auf diesem Weg kann die IP-Adresse des Servers mit einem einprägsamen Namen konfiguriert werden. Die Webadresse wird nach Anforderung über die Adresszeile im Browser mittels des DNS-Servers in die dazugehörige IP-Adresse aufgelöst.

Zur Verdeutlichung der eben erläuterten Annahme dient folgendes Konfigurationsbeispiel in Tabelle 12:

Beispiel-IP:	166.128.2.8
DNS-Name:	http://redmine.kos.dh

Tabelle 12: Beispielhafte DNS-Konfiguration des Issue Tracking-Servers

Da das System über sämtliche Internet-Browser erreichbar ist, ist keine Installation von gesonderter Software auf den Clients notwendig. Es genügt demzufolge die Implementierung des Issue-Tracking-Systems auf dem Server, sofern ein kompatibler Browser auf den Clients vorinstalliert ist. Als weitere Grundvoraussetzung gilt der Besitz von gültigen Login-Daten.

Für den Zugriff auf das Bugtracking-System empfiehlt sich neben den Login-Daten der einzelnen Nutzer die allgemeine Zugriffsberechtigung auf den Server. So können lediglich die Anwender auf die Login-Plattform gelangen, die dafür vorgesehen sind. Tiefergehende Einsichten sind zwar nur mit gültigen Zugangsdaten möglich, jedoch besteht innerhalb des Issue Tracking-Systems die Möglichkeit, öffentliche Projekte anzulegen. Diese sind auch im ausgeloggten Zustand einsehbar. Sofern alle Personen auf den Server zugreifen können, sind gegebenenfalls geheime Daten folglich frei einsehbar. Realisierbar ist dies mit einer Zugriffsberechtigung über LDAP, sofern das Unternehmen über derartige Einrichtungen verfügt. Unter anderem aus diesem Grund gilt die LDAP-Unterstützung des Issue-Tracking-Systems als Kriterium bei der

Produktauswahl. Versucht ein unberechtigter Nutzer das System nun über die IP-Adresse oder den DNS-Namen aufzurufen, so wird diesem kein Zugriff gewährt.

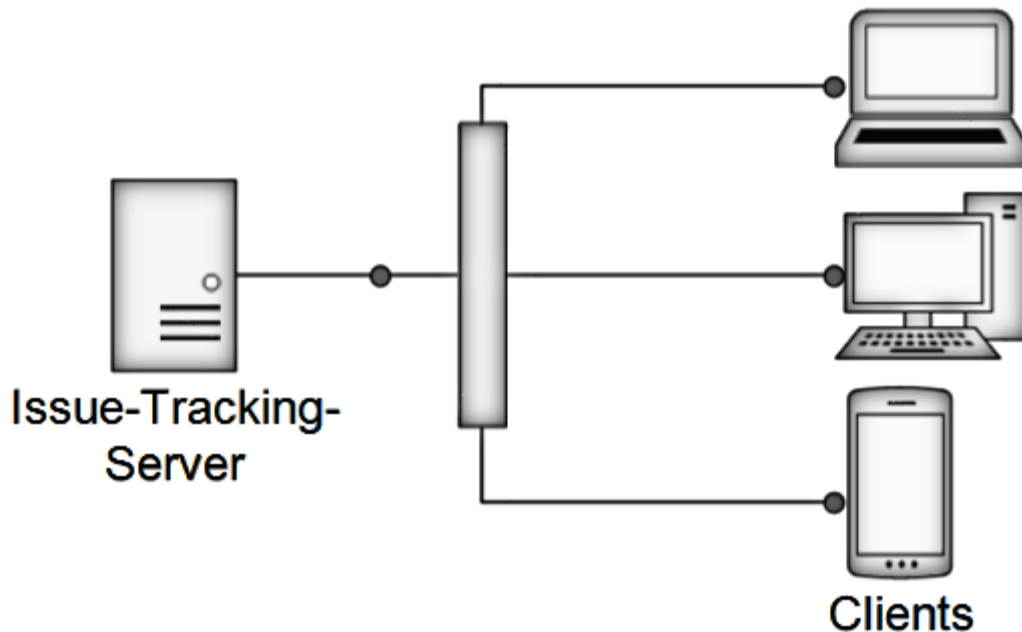


Abb. 14: Konzeptioneller Aufbau der Netzwerkinfrastruktur

Die Reihenfolge der zu tätigen Schritte wird in der folgenden Abbildung nochmals zusammenfassend dargestellt:



Abb. 15: Schrittweise Ablauf der System-Einführung

7.1.3 Konzeption der Rollenverteilung im Issue Tracking-System

Eine klare Zuordnung von Aufgaben schafft einen besseren Überblick in der Prozessstruktur eines Unternehmens. Wie zuvor bereits erwähnt müssen die Geschäftsprozess-Verantwortlichen diese festlegen, daher ist eine allgemeine Definition zur Verteilung der Verantwortlichkeiten nur schwer zu bewerkstelligen, da die Prozesslandschaften von Unternehmen zu Unternehmen jeweils verschieden sind. Daher sind die Rollen und die damit anfallenden Aufgaben von jedem Unternehmen individuell zu bestimmen. Die Unternehmenshierarchie kann hierfür als optimaler Anhaltspunkt herangezogen werden.

Für die Softwareentwicklung lassen sich jedoch allgemeine Richtlinien festlegen. Im Wesentlichen sind folgende Rollen zu definieren:

- Administrator
- Manager
- Entwickler
- Fachabteilung
- Leser

Zur vollständigen Verwaltung der Bug-Tracking Anwendung sind Administratoren notwendig. Der Administrator verfügt über erweiterte Benutzerrechte zur Erfüllung spezieller Aufgaben, hauptsächlich die Verwaltung und Durchsetzung von Nutzungsrichtlinien.

Während die Administratoren für die allgemeinen Verwaltungsaufgaben zuständig sind, bedarf es in der Softwareentwicklung sogenannte Manager. Diese übernehmen die Tätigkeiten als Projektleiter und sind für die ordnungsgemäße Durchführung für die Softwareprojekte verantwortlich. Die Rolle des Managers benötigt umfassende Rechte zur Organisation eines Projektes. Diese wären beispielsweise das Hinzufügen neuer Projektmitglieder, Festlegung der Start- und Endtermine etc.

Eine weitere unabdingbare Rolle ist der Softwareentwickler. Diese Anwendergruppe ist für die Entwicklung der Software und der Bearbeitung der eingehenden Bug-Tickets zuständig. Hierfür benötigt diese Anwendergruppe jegliche Rechte zur Erfassung neuer und Bearbeitung bestehender Tickets. Ergänzend zu den Entwicklern ist die Anwendergruppe der Fachabteilung erforderlich. Die Mitarbeiter der Fachabteilungen erstellen die Aufträge für die Entwickler und sind für die Überprüfung der jeweiligen Implementierung zuständig. Daher benötigen diese nahezu dieselben Rechte. Lediglich im Workflow bei der Bearbeitung von Bug-Tickets kann es zu unterschieden kommen.

Gegebenfalls sind Benutzergruppe mit ausschließlichem Leserecht sinnvoll. Dies betrifft Anwender, welche in irgendeiner Art und Weise mit einem Softwareprojekt in Verbindung stehen, jedoch keine Eingriffe im Bug-Tracking-System vornehmen kön-

nen sollen. Bezeichnend hierfür können Studenten des Unternehmens eine solche Benutzergruppe ausfüllen.

7.1.4 Fortwährende Überlegungen

Neben dem Zugriff von stationären PCs oder Notebooks bietet sich die Verwendung von mobilen Endgeräten wie Tablets oder Smartphones an. Im Zuge der Always-Online-Bewegungen gewinnt die Implementierung von derartigen Systemen auch auf diesen Plattformen zunehmend an Relevanz. Zwar verfügen die mobilen Endgeräte dieser Klasse ebenfalls über moderne Webbrowser, jedoch ist auf diesem unter Umständen nur eine erschwerte Bedienung möglich. Dabei wird von barrierefreier Bedienung gesprochen. Folglich ist die Nutzeroberfläche so zu konzipieren und zu entwickeln, dass eine ästhetische Bedienung auch auf kleinen Displays möglich ist. Es sind somit gesonderte Oberflächen für PC/Notebook und Smartphone/Tablet notwendig.⁶³ Meist decken diese nur die essenziellen Funktionen ab, die das System im Gesamten bietet. Dadurch wird die Übersichtlichkeit auf den kleineren Displays erhalten. Aufgrund des frei verfügbaren Quellcodes bei Open Source-Tools kann eine Lösung vom Unternehmen selbst entwickelt werden. Zwar kann auf diesem Weg eine an die Anforderungen angepasste Mobile Device-Lösung entwickelt werden, jedoch erfordert ein ausgereiftes Produkt einen verhältnismäßig hohen Aufwand. Für Redmine ist seit Mai 2011 bereits eine Lösung für das Apple iPhone im AppStore verfügbar. Die Anwendung greift über die offizielle Redmine-API auf die Projekte zu. Ende des Jahres 2011 folgte selbige App für das Google-Betriebssystem Android. Seit 2013 ist die Lösung auch für Windows Phone erhältlich. Aus dieser Entwicklung wird deutlich, dass der Zugriff auf Issue Tracking-Systeme auch von diesen Plattformen aus von Relevanz ist. Als Vorteil dieser Plattformen ist aufzuführen, dass die Vorrichtungen für eine Verbindung über ein mobiles Datennetz in Smartphones und Tablets in der Regel bereits integriert sind. So können auch Mitarbeiter, die sich nicht an ihrem Arbeitsplatz befinden, schnell auf Ereignisse reagieren. Der Datentransfer erweist sich als gering, da die grafischen Elemente auf dem Endgerät innerhalb der App gespeichert sind. Gesteuert wird die Anwendung auf dem Server anhand von Befehlen, welche über die API übermittelt werden. Größere Datenpakete fallen lediglich beim Download eines Screenshots oder sonstigen Anhängen an. Folglich ist die

⁶³ Vgl. Kersten, H./Klett, G. (2012), S.13ff

Verfügbarkeit eines Wireless-Local Area Networks (W-LAN) nicht zwingend erforderlich.

In folgenden Bereichen ist die Verwaltung über die Redmine-App möglich:

- Projekte
- Tickets
- Benutzer
- Kommentare

8 Fazit

Neben einer kurzen Betrachtung der theoretischen Grundlagen wurden in der vorliegenden Ausarbeitung die Softwareprodukte Trac, Bugzilla, Mantis und Redmine anhand eines erarbeiteten Kriterienkatalogs getestet und analysiert. Es hat sich herausgestellt, dass zahlreiche Open Source-Tools im Bug-Tracking Bereich auf dem Markt verfügbar sind. Diese bieten einen breiten Funktionsumfang, welcher sich nur marginal von kommerziellen Produkten unterscheidet. Als vorteilhaft an Open Source-Tools erweist sich der freie Zugang zum Quellcode und die daraus resultierende Anpassbarkeit der Anwendung. Im Internet ist eine große Community zu finden, die bei Problemfällen zu Rate gezogen werden kann. Des Weiteren bieten Servicedienstleister gezielten Support zu einzelnen Produkten an. Jedoch liegt der Vorteil hier grundsätzlich bei den kommerziellen Produkten, da Entwicklung und Service aus einer Hand stammen. Diese sind in Form von teilweise hohen Lizenzgebühren zu entlohnen.

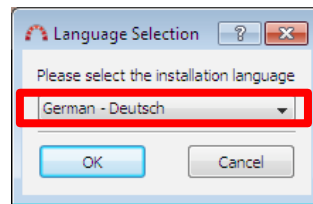
Generell wurden die aufgestellten Kriterien von allen Produkten nahezu erfüllt. Eine Empfehlung wird in dieser Ausarbeitung für das Produkt Redmine gesondert ausgesprochen. Das Tool konnte als einziges allen Kriterien gerecht werden. Zudem bietet es einen hohen Funktionsumfang, einen einfachen Installationsprozess und eine intuitive Bedienung. Die grafische Nutzeroberfläche ist dabei übersichtlich und simpel gehalten ohne auf Funktionen verzichten zu müssen. Das Tool *Trac* bietet einen detailreicheren Installationsprozess, in dem spezifische Parameter konfiguriert werden können. Zudem richtet sich die grafische Nutzeroberfläche primär an erfahrenere Nutzer. Bugzilla bietet einen hohen Funktionsumfang, konnte allerdings nicht mit der Usability im Vergleich zu Redmine mithalten. Dies schränkt unter Umständen die Produktivität von unerfahrenen Nutzern bei einer Einführung im Unternehmen ein. Der Bug-Tracker Mantis bietet ähnlich gute Werte wie Redmine, verzichtet jedoch auf Funktionalitäten wie beispielsweise die LDAP-Anbindung und das Versionskonzept.

9 Anhang

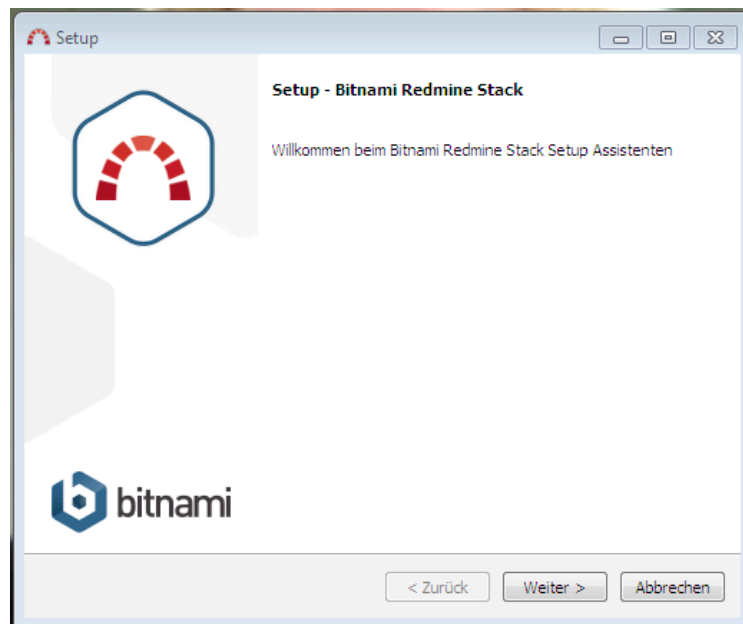
Anhangsverzeichnis

Anhang 1: Redmine: Installation	63
Anhang 2: Redmine: Start der Anwendung	70
Anhang 3: Redmine: Login.....	71
Anhang 4: Redmine: Übersicht	72
Anhang 5: Redmine: Anlegen eines Projekts.....	72
Anhang 6: Redmine: Versionierung	75
Anhang 7: Redmine: Darstellung der Roadmap.....	76
Anhang 8: Redmine: Darstellung des Kalenders.....	76
Anhang 9: Redmine: Erstellen eines neuen Benutzers	77
Anhang 10: Redmine: Rollenkonzept und Berechtigungen.....	79
Anhang 11: Redmine: Definition des Workflows	80
Anhang 12: Redmine: Ticket-Status.....	81
Anhang 13: Redmine: Tracker-Konfiguration	82
Anhang 14: Redmine: Gruppen-Konfiguration	82
Anhang 15: Redmine: Konfiguration von Feldern.....	83
Anhang 16: Redmine: Aufzählungen.....	84
Anhang 17: Redmine: Allgemeine Konfigurationen.....	85
Anhang 18: Redmine: Projektübersicht.....	85
Anhang 19: Redmine: Anlegen eines Tickets.....	86
Anhang 20: Redmine: Erstellung eines Unterprojekts.....	88
Anhang 21: Redmine: Erstellung von Widgets	90
Anhang 22: Redmine: Bearbeitung eines Tickets	91

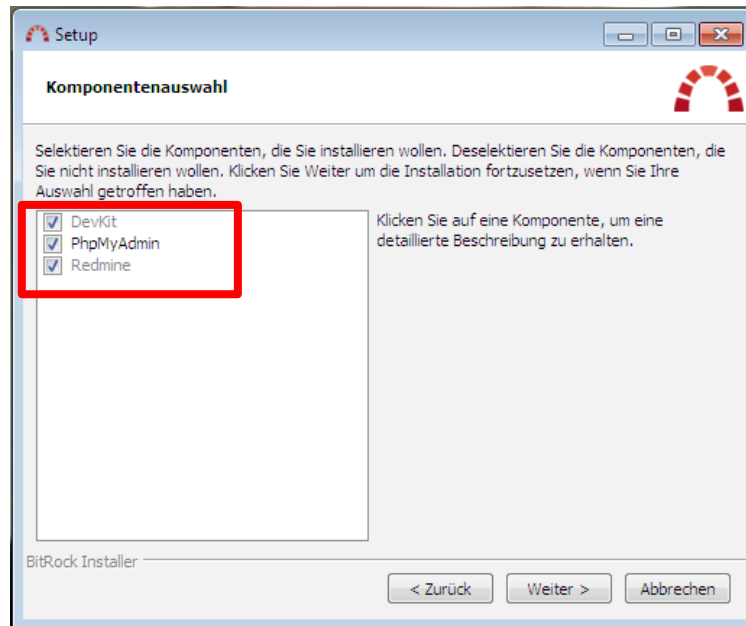
Anhang 1: Redmine: Installation



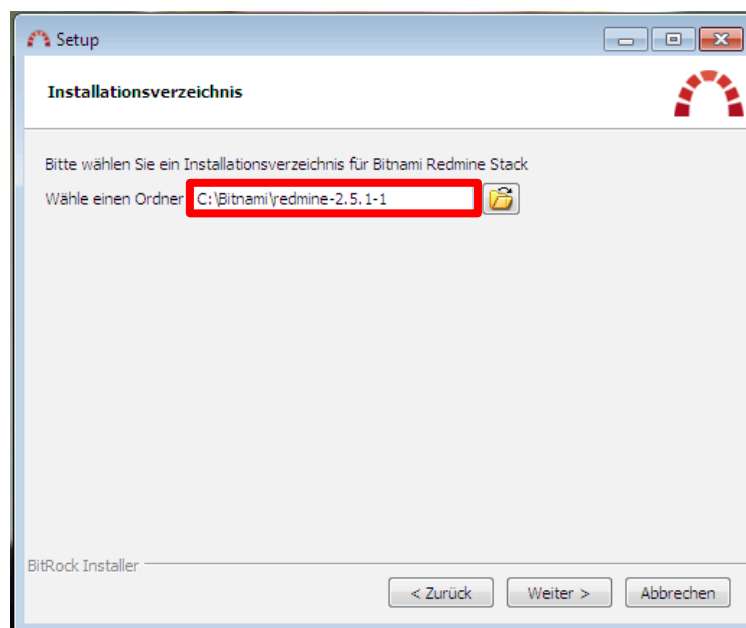
Nach dem Start des Installationsassistenten kann der Administrator zwischen unterschiedlichen Sprachen auswählen. Für Redmine existieren aufgrund der breiten Open Source-Community zahlreiche Übersetzungen. Die Sprachwahl gilt zunächst nur für den Installationsprozess.



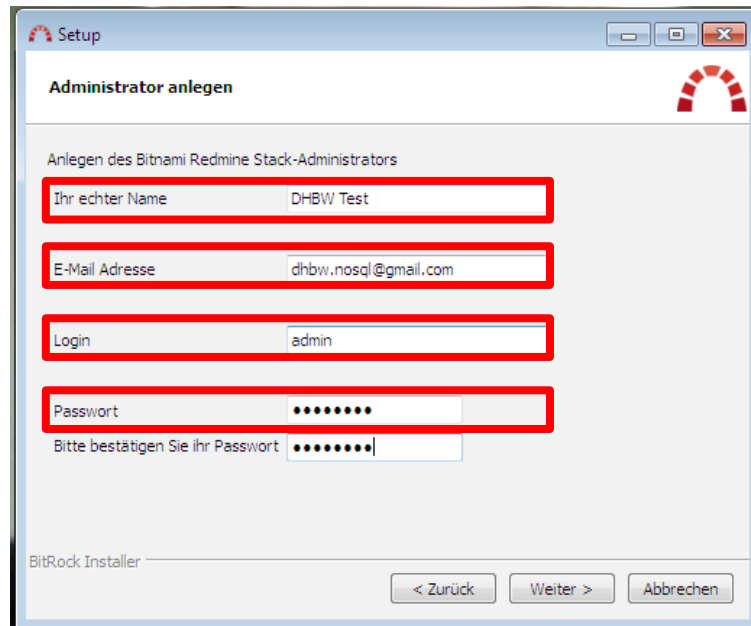
Nach der Sprachwahl ist das Setup vorzufinden.



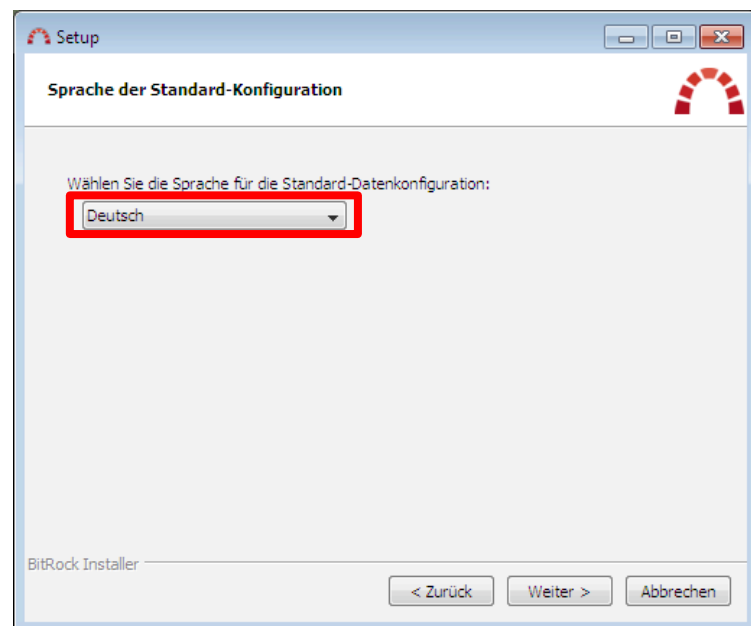
Es kann zwischen unterschiedlichen Komponenten gewählt werden, die im Zuge der Installation auf den Rechner kopiert werden. Um ein vollständiges und funktionsfähiges Softwarepaket auf dem System zu gewährleisten werden alle Komponenten ausgewählt.



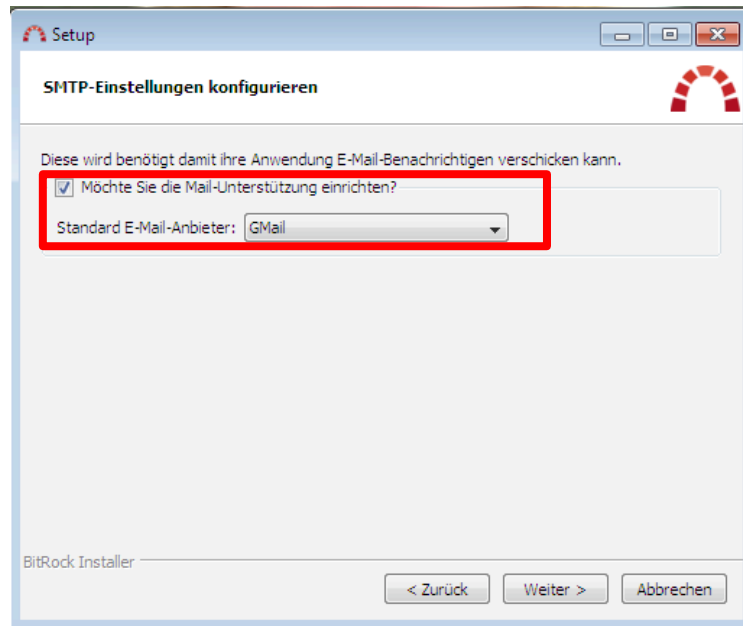
Das Installationsverzeichnis, in das die Daten von Redmine geschrieben werden, ist frei auszuwählen.



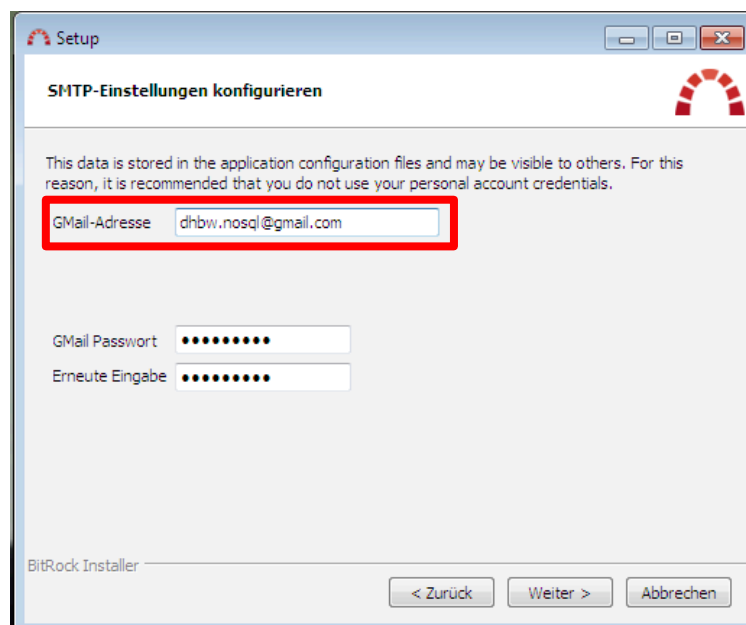
Als erster User im System wird ein Administrator-Account angelegt. Für den Account sind der echter Name des Benutzers (in der Praxis möglichst Vor- und Nachname), eine gültige E-Mail-Adresse, ein Login-Name sowie ein Passwort notwendig.



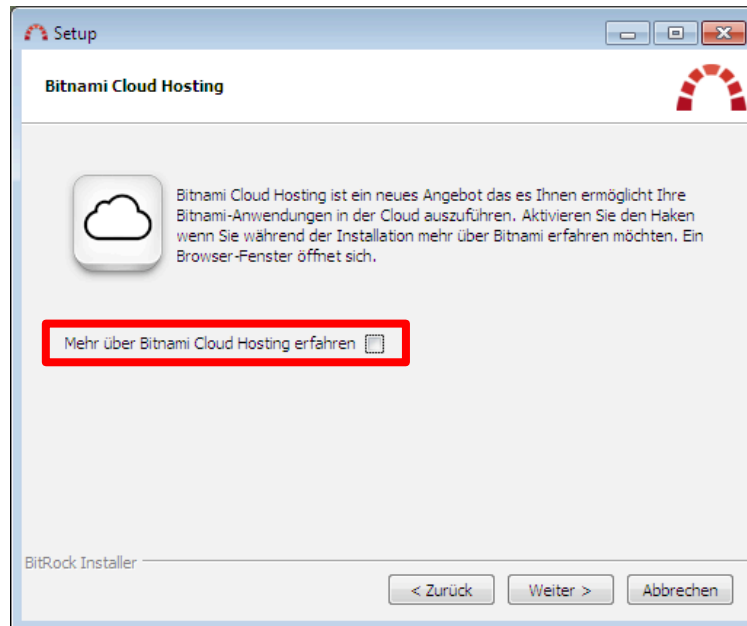
Im nächsten Schritt ist erneut die Systemsprache zu wählen.



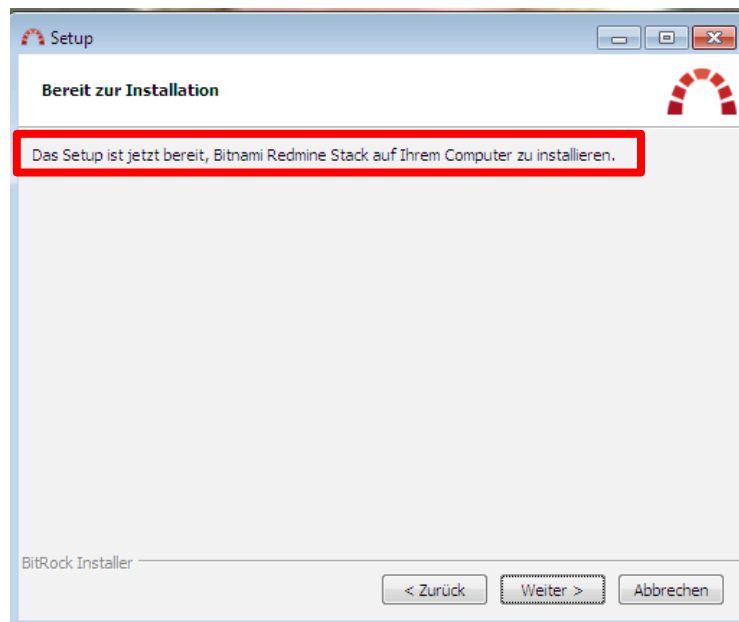
Das Issue-Tracking-System Redmine kann automatische Benachrichtigungen an einen vorgegebenen E-Mail-Account senden. Es existieren einige vordefinierte Mail-Dienste. Ist der gewünscht nicht in der Liste vorhanden, sind die einzelnen Daten manuell einzutragen.



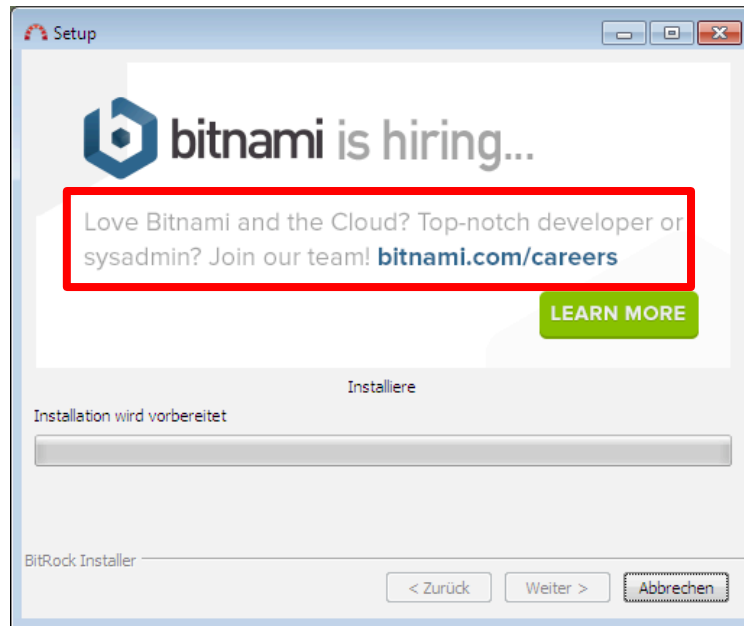
Im Rahmen des Tests wird ein Google-Mail-Account verwendet.



Neben dem Betrieb auf den unternehmenseigenen Servern kann Redmine auf den Bitnami-Servern gehostet werden. In diesem Schritt kann der Administrator wählen, ob er genauere Informationen einholen möchte. Folglich wird hier nur das Interesse des Kunden angeregt statt eine direkte Installation auf den Bitnami-Servern einzuleiten. Wünscht der Administrator Informationen über den Cloud-Dienst, so ist das obige Feld auszuwählen.



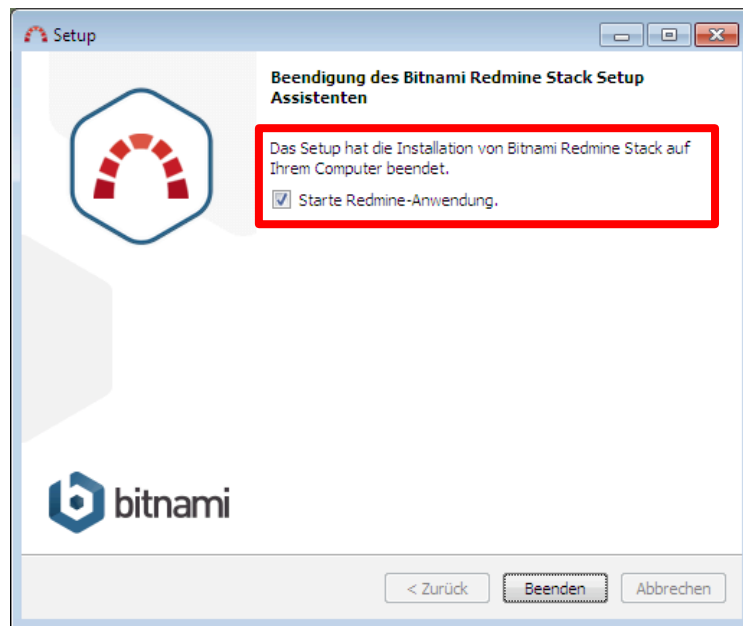
Der Assistent meldet nun, dass alle notwendigen Angaben getätigt wurden und das System nun bereit ist, Redmine auf dem Rechner zu installieren.



Während der Installation wirbt das Unternehmen bitnami in eigener Sache. Der Installationsfortschritt ist am obigen Balken zu erkennen, welcher zunehmend grün gefärbt wird.

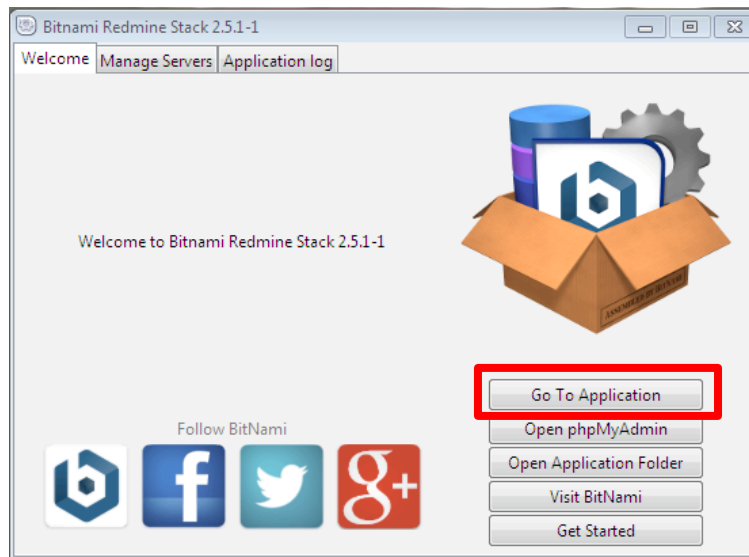


Im Laufe des Prozesses wird die Software *Redmine* nochmals kurz vorgestellt. Am grünen Balken ist zu erkennen, dass der Installationsprozess fortgeschritten ist.

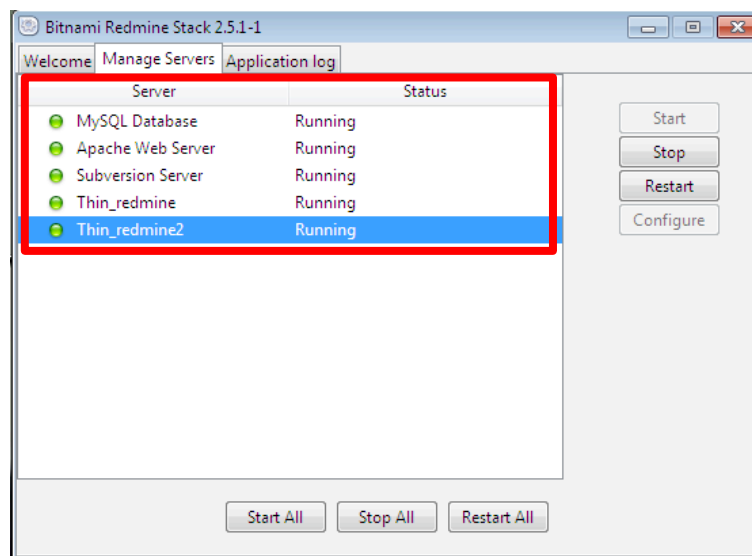


Der Assistent vermeldet nach erfolgreichem Abschluss, dass die Installation von Redmine beendet ist. Sofern der Administrator das obige Feld angewählt lässt, wird beim Beenden des Assistenten umgehend Redmine gestartet.

Anhang 2: Redmine: Start der Anwendung

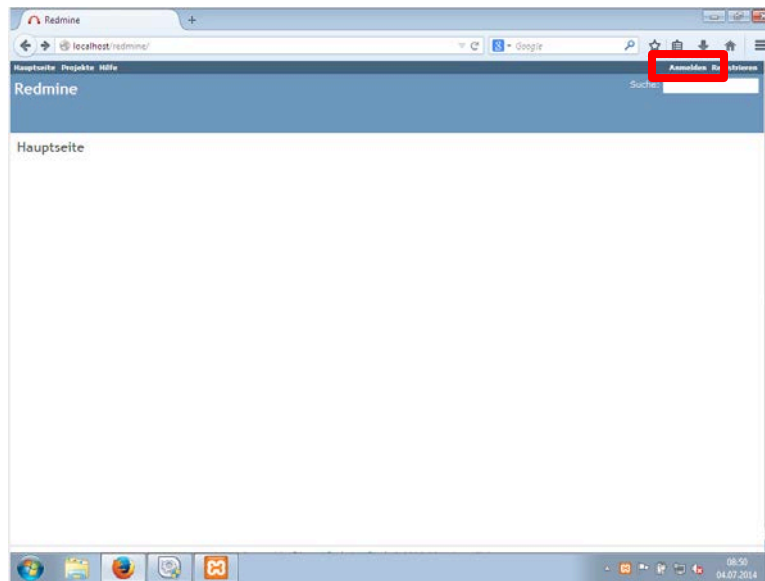


Nach dem Start von Redmine erhält der Administrator eine Übersicht über den Anwendungsinhalt. Durch Klick auf *Go to Application* wird die Anwendung im Browser gestartet.

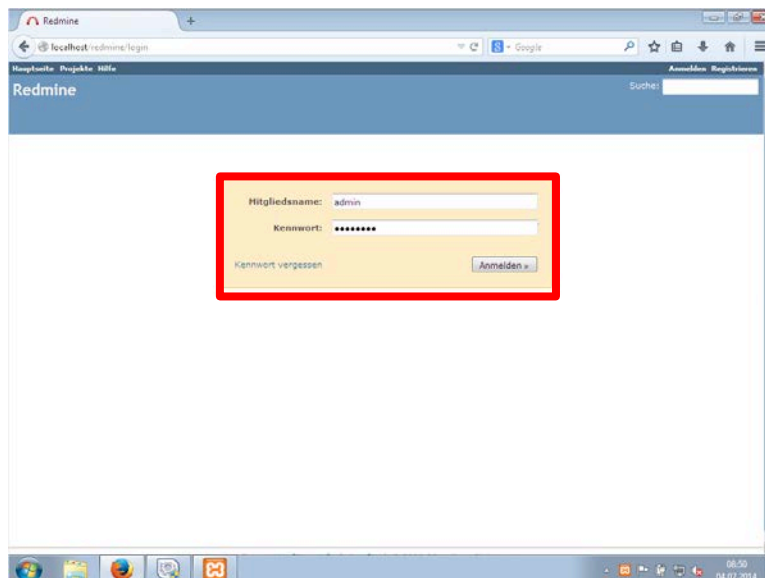


Wechselt der Administrator zur Registerkarte *Manage Servers*, werden alle im Zuge der Redmine-Installation eingerichteten Server aufgelistet. Durch Blick auf die Spalte *Status* wird der derzeitige Zustand von diesen Systemen aufgezeigt.

Anhang 3: Redmine: Login

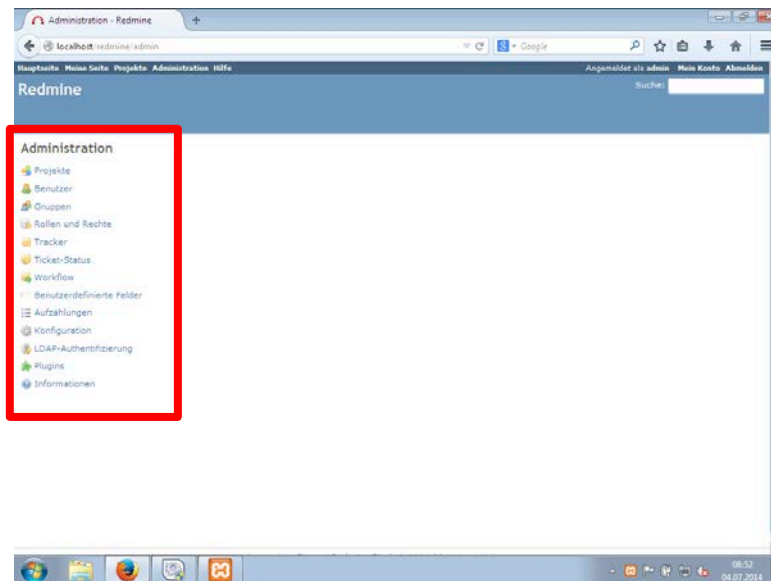


Nachdem die Anwendung gestartet wurde gelangt der Nutzer auf die Startseite von Redmine. Mit Klick auf Anmelden wird er auf das Login-Feld weitergeleitet. Zwar ist die Registrierung ebenfalls möglich, jedoch sollte dies innerhalb von Unternehmen lediglich durch Administratoren erfolgen.



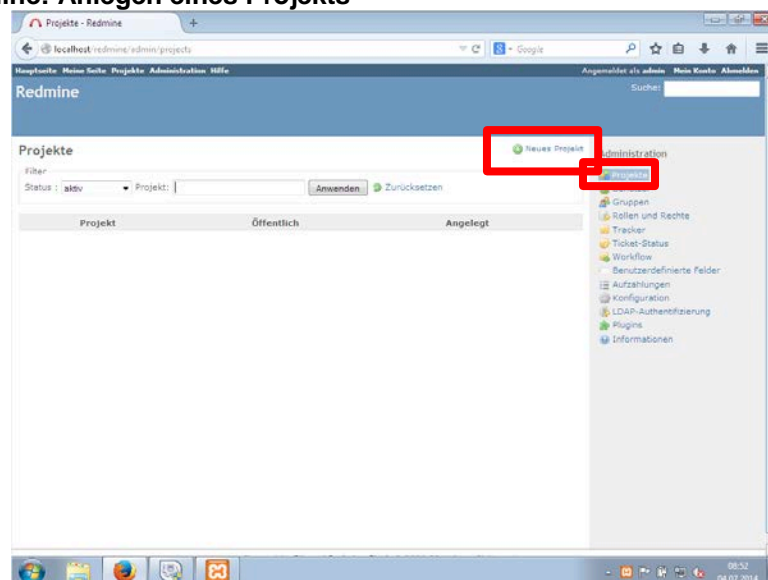
Der Login erfolgt anhand der vom Administrator vorgegebenen Daten. In diesem Fall loggt sich der Administrator selbst ein.

Anhang 4: Redmine: Übersicht

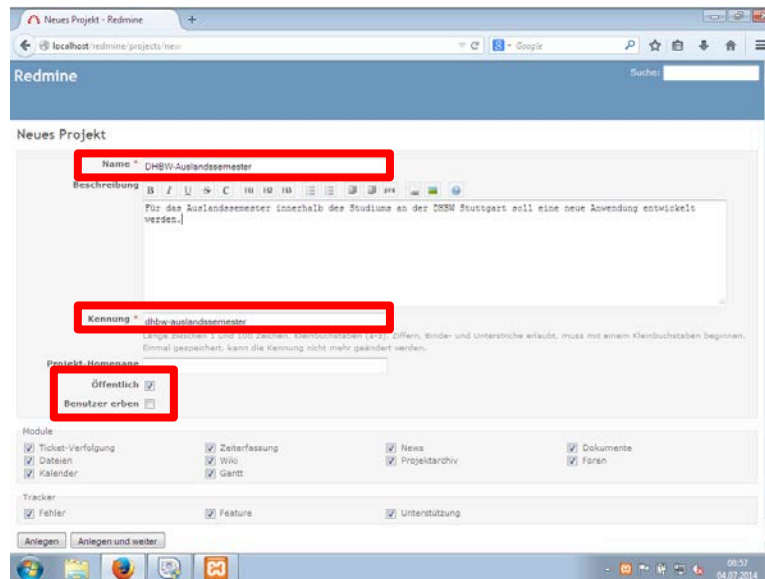


Nach dem erfolgreichen Login als Administrator wird dem Nutzer zunächst eine Übersicht über alle Hauptpunkte des Systems gegeben.

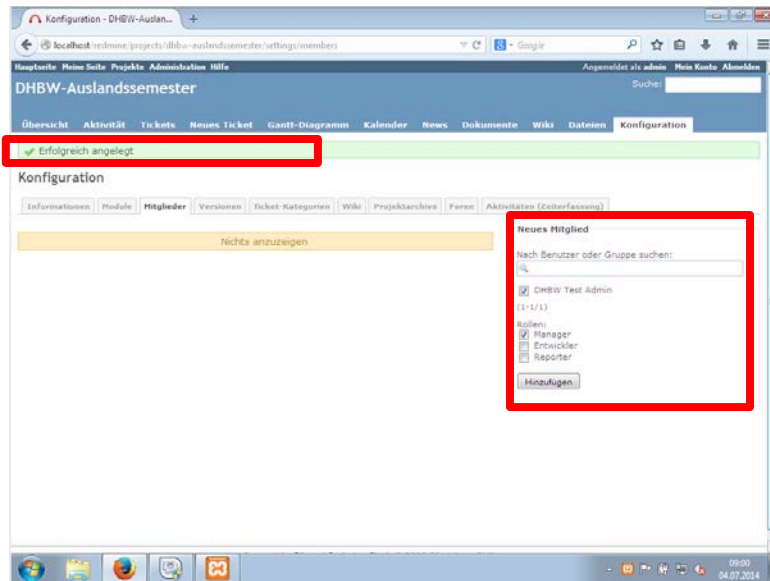
Anhang 5: Redmine: Anlegen eines Projekts



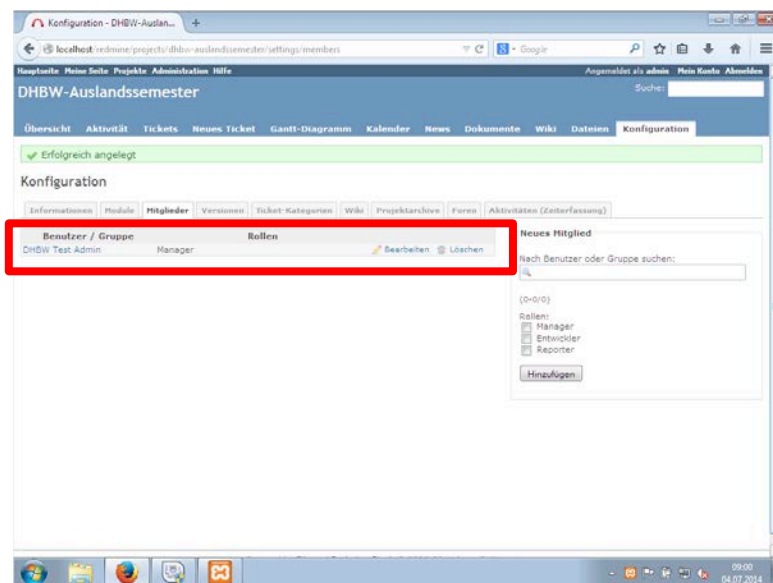
Nach dem Klick auf *Projekte* gelangt der Nutzer Projektübersicht von Redmine. Die Programmübersicht, welche sich zuvor in der Mitte befand, verschob sich nun nach Rechts und stellt eine Navigationsleiste dar. Derzeit sind noch keine Projekte angelegt, weshalb die Liste leer ist. Mit Klick auf *Neues Projekt* wird der Nutzer auf den Projektassistenten umgeleitet.



Bei der Erstellung eines neuen Projekts sind ein Name, eine Beschreibung und eine Kennung erforderlich. Die Kennung wird automatisch generiert und kann manuell abgeändert werden. Der Nutzer kann wählen, ob das Projekt auch für nicht-registrierte Benutzer öffentlich sein darf. Unter den Rubriken *Module* und *Tracker* können weitere Grundkonfigurationen vorgenommen werden.

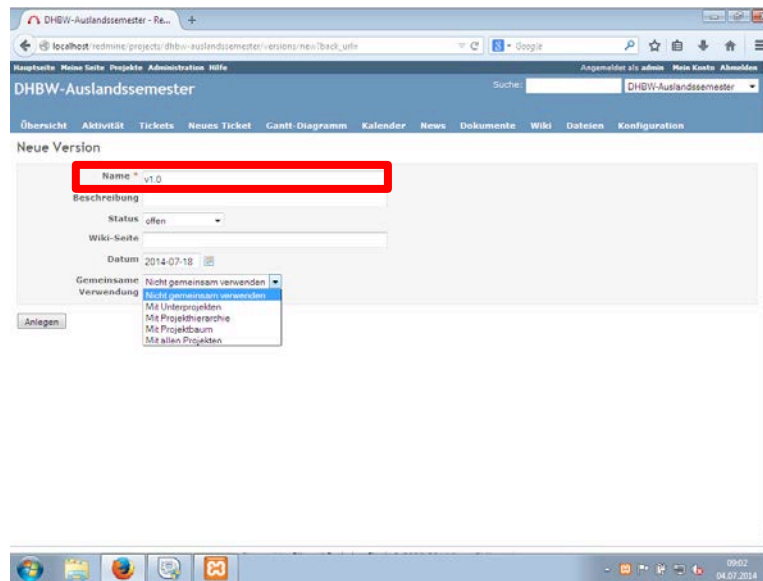


Nachdem ein Projekt angelegt wurde sind diesem Arbeitspaket Mitglieder zuzuordnen, die für die Teilnahme vorgesehen sind. Unter der Registerkarte *Mitglieder* eines jeden Projekts sind die jeweiligen Personen aufgelistet. Im rechten Teil des Fensters sind sämtliche User aufgelistet, die noch nicht Teil des Projekts sind. Möchte der Administrator einen dieser hinzufügen, so ist die Ticker-Box anzuklicken. Des Weiteren ist dem User eine Rolle im Projekt zuzuordnen. Mit Klick auf *Hinzufügen* wird der User in das Projekt eingebunden.

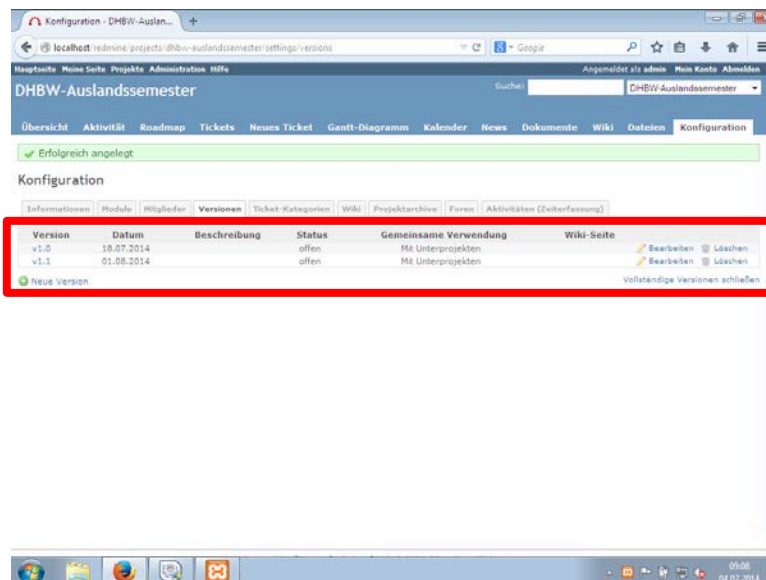


Nun ist zu sehen, dass *DHBW Test Admin* Teil des Projekts *DHBW-Auslandssemester* ist.

Anhang 6: Redmine: Versionierung

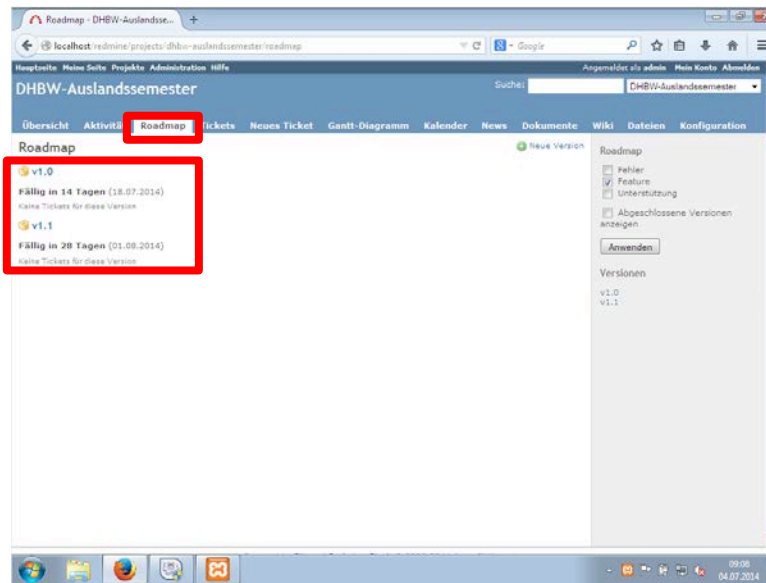


Um die Übersichtlichkeit im Verlauf des Projekts zu optimieren, können Versionsnummern angelegt werden. Es empfiehlt sich dabei eine einheitliche Notation zu verwenden. Neben dem Pflichtfeld *Versionsname* ist optional eine Beschreibung, der Status, ein Link zur Wiki-Seite, das Datum der spätesten Fertigstellung der Version sowie die Verwendung der Version ausgewählt werden.



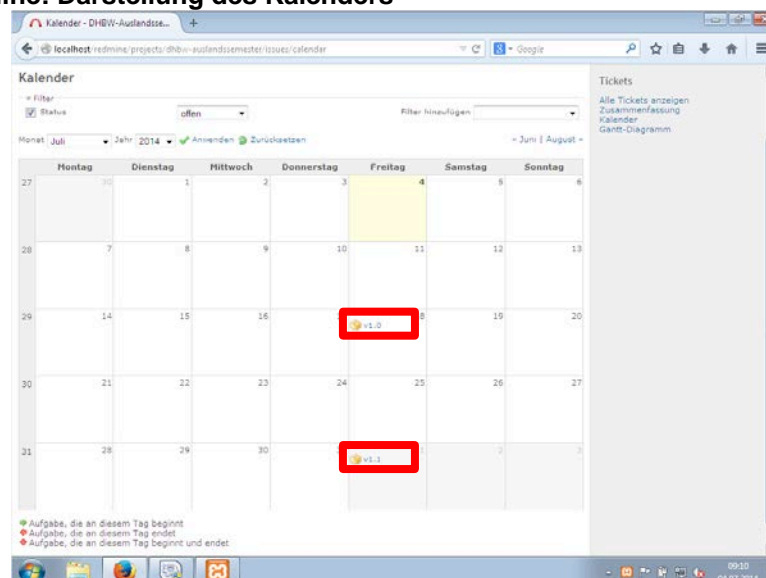
Wie in der obigen Abbildung zu sehen wurden zwei Versionen angelegt, die nach der Bezeichnung sortiert sind.

Anhang 7: Redmine: Darstellung der Roadmap



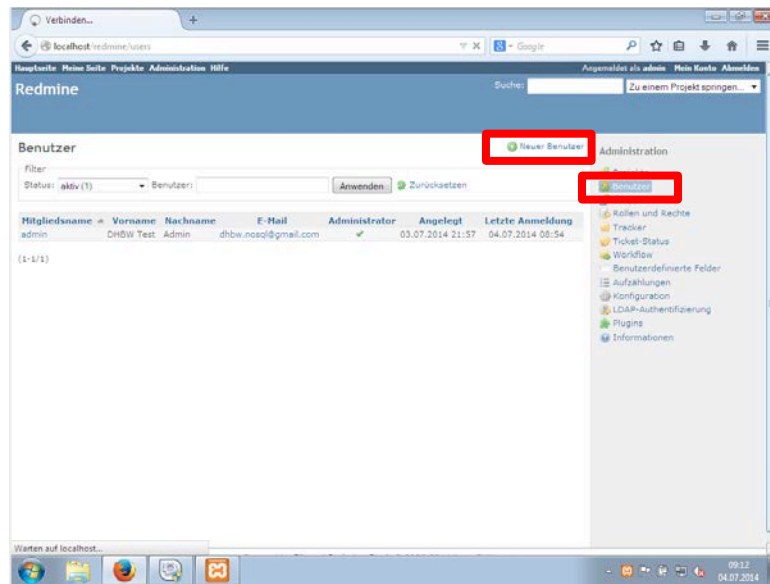
Beim Klick auf die Registerkarte *Roadmap* wird ein allgemeiner Verlauf des Projekts angezeigt. Auf dieser Seite sind die einzelnen Versionsnummern integriert. Des Weiteren werden die Fälligkeit der Version sowie gegebenenfalls vorhandene Tickets angezeigt.

Anhang 8: Redmine: Darstellung des Kalenders

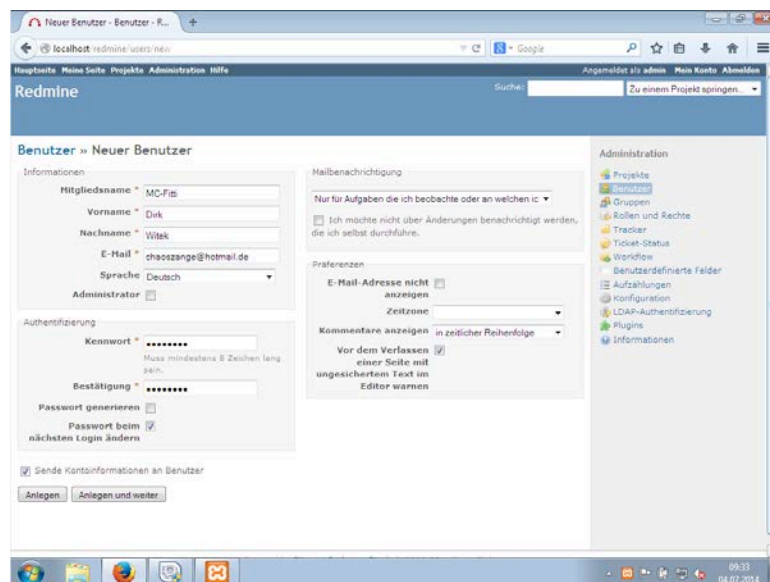


Im Register *Kalender* ist ein herkömmlicher Kalender vorzufinden, in den die einzelnen Fälligkeitstermine bzw. Versionsnummern automatisch eingetragen werden.

Anhang 9: Redmine: Erstellen eines neuen Benutzers



Um einen neuen Benutzer anzulegen ist ein Klick in der rechten Menüleiste notwendig. Auf dieser Übersicht sind die bisher vorhandenen User zu erkennen. Unter *Neuer Benutzer* wird der Administrator auf den Assistenten weitergeleitet.



Die erforderlichen Angaben beim Erstellen eines neuen Benutzers sind mit * gekennzeichnet. Bei Angabe einer E-Mail-Adresse erhält der Nutzer eine Mail mit seinen Zugangsdaten.

Redmine Kontoaktivierung



dhbw.nosql@gmail.com (dhw.nosql@gmail.com) Der Kontaktliste hinzufügen 09:35 Gruppen

Ich vertraue „dhw.nosql@gmail.com“. Inhalt immer anzeigen.

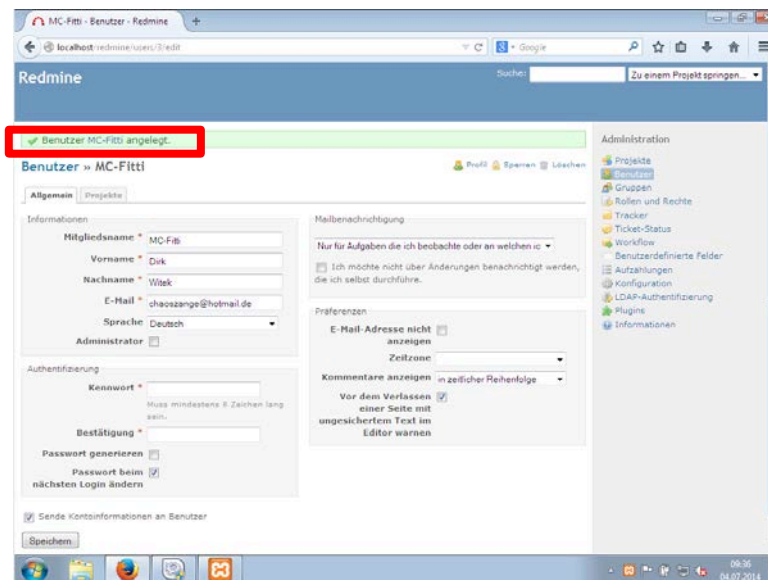
Ihre Konto-Informationen:

- Mitgliedsname: MC-Fitti
- Kennwort: qwertzui

Anmelden: <http://localhost:3000/login>

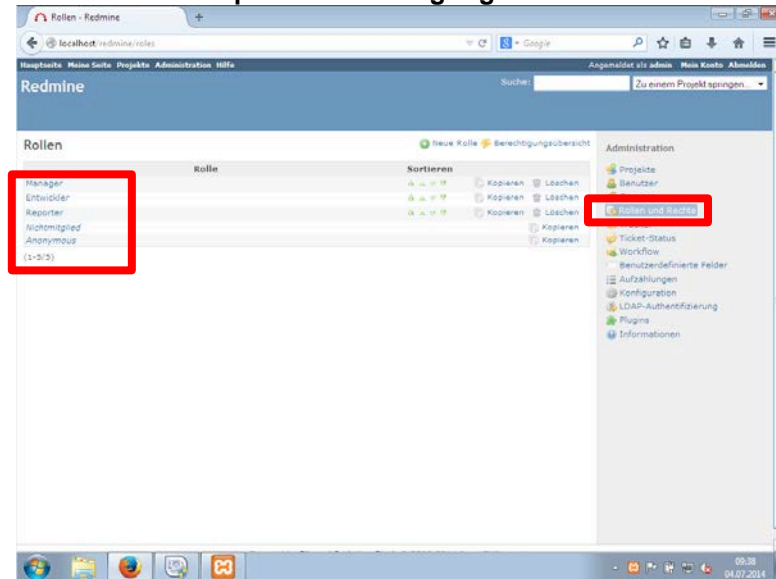
You have received this notification because you have either subscribed to it, or are involved in it.
To change your notification preferences, please click here: <http://hostname/my/account>

Der zuvor angelegte Benutzer *MC Fitti* hat eine E-Mail erhalten, dass für ihn ein Account angelegt wurde. Es ist somit zunächst keine Interaktion seitens des neuen Benutzers notwendig, wodurch ihm der Registrationsaufwand erspart bleibt.

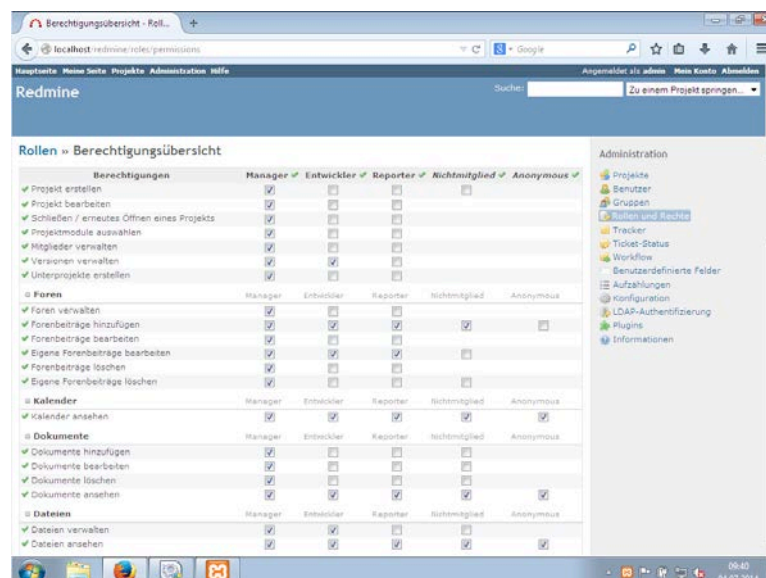


Sofern alle Eingaben korrekt waren erhält der Nutzer eine Bestätigung, dass ein neuer User angelegt wurde.

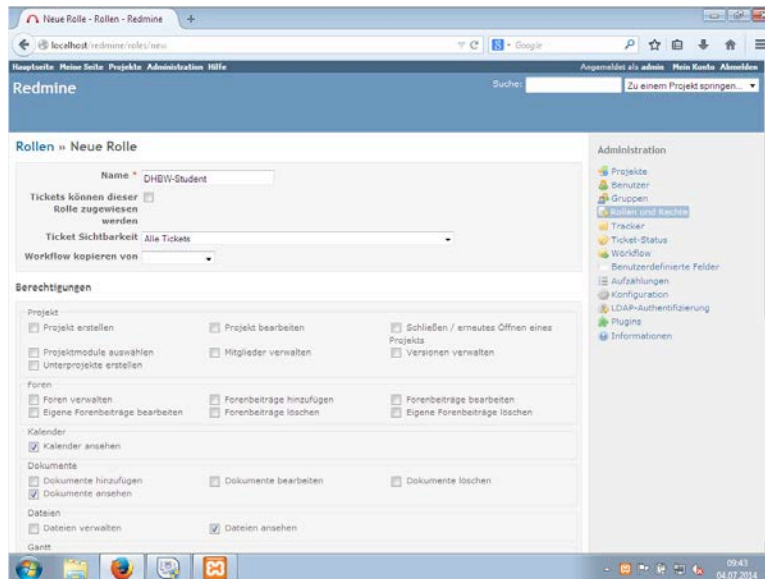
Anhang 10: Redmine: Rollenkonzept und Berechtigungen



Redmine bringt ein ausgereiftes Rollenkonzept mit. Standardmäßig sind die Rollen *Manager*, *Entwickler*, *Reporter*, *Nichtmitglied* und *Anonymus* vorhanden. Den einzelnen Usern kann jeweils eine Rolle zugewiesen werden, wodurch sie Berechtigungen in Redmine erhalten. Dies erspart den Aufwand bei der Rechteverwaltung wesentlich. Ebenfalls ist es möglich die Rollenberechtigungen nachträglich zu verändern, wodurch diese an die jeweiligen User weitervererbt werden. Beim Klick auf eine der Rollen kann die Konfiguration eingesehen werden.

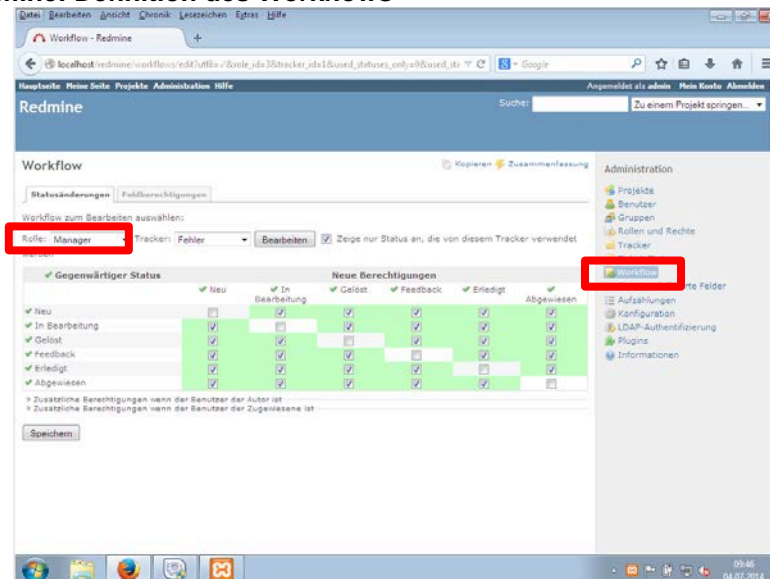


Die Kriterien sind für die Rollen jeweils identisch. Möchte der Administrator ein Recht konfigurieren, so muss die zugehörige Ticker-Box angewählt oder abgewählt werden.



Der Administrator hat die Möglichkeit, weitere Rollen anzulegen und zu konfigurieren. In diesem Fall wird die Rolle *DHBW-Student* erstellt. Hier ist vorgesehen, dass die mit dieser Rolle versehenen User alle Projekte betrachten, jedoch nichts verändern können.

Anhang 11: Redmine: Definition des Workflows

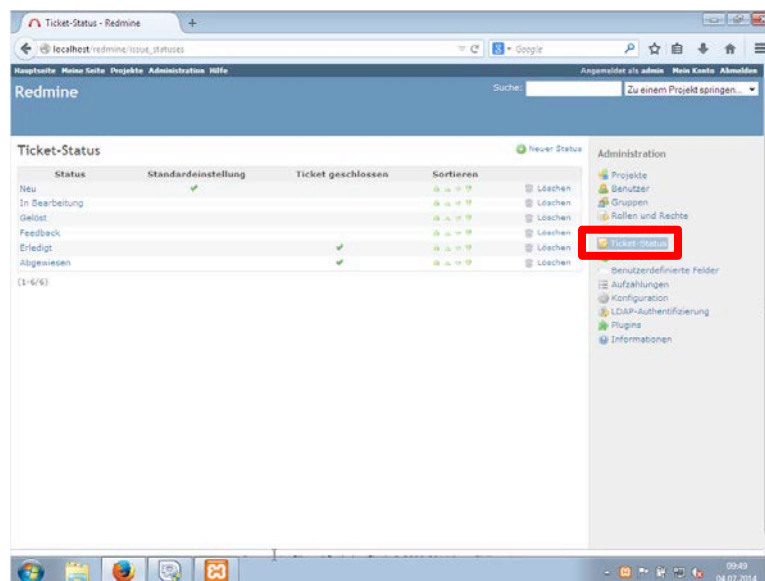


Beim Klick auf *Workflow* in der Navigationsleiste wird eine Oberfläche zur Konfiguration einzelner Berechtigungen im Workflow. Der Workflow beschreibt den Ablauf eines Prozesses, in dem ein Fehler zu beheben ist. Durch diese klare Strukturierung ist gegeben, dass die Folgeschritte in einem Prozess bereits vordefiniert sind. Zu jedem Zeitpunkt kommt dabei eine bestimmte Rolle zur Geltung. Dieser Mechanismus

kann als Unterstützung der Automatisierung in einem Prozess angesehen werden, da aufgrund der klaren Vorgabe keine Rücksprache unter Mitarbeitern notwendig ist.

Beispiel: befindet sich ein Ticket im Status *Neu*, so darf der Manager das Ticket in jeden anderen Status versetzen. Dies wird durch die grüne Markierung und den Ticker verdeutlicht.

Anhang 12: Redmine: Ticket-Status

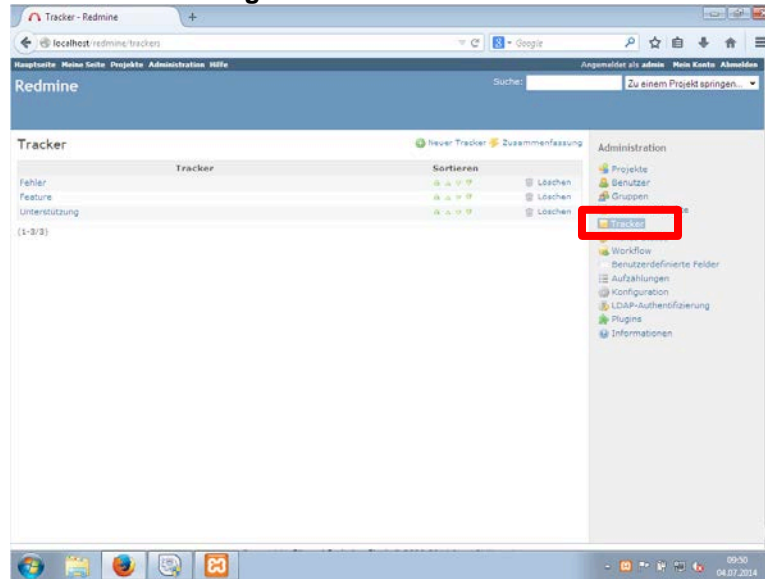


Innerhalb des Workflows durchschreitet ein Ticket unterschiedliche Status.

- In Bearbeitung → Problem wird gerade behoben/bearbeitet
- Gelöst → Fehler wurde identifiziert und behoben
- Feedback → Nutzer verlangt Feedback zu der Änderung
- Erledigt → Auftrag wurde ausgeführt. Wird gewählt falls kein Fehler sondern lediglich eine Änderung gewünscht war
- Abgewiesen → Annahme des zugeteilten Tickets wurde verweigert

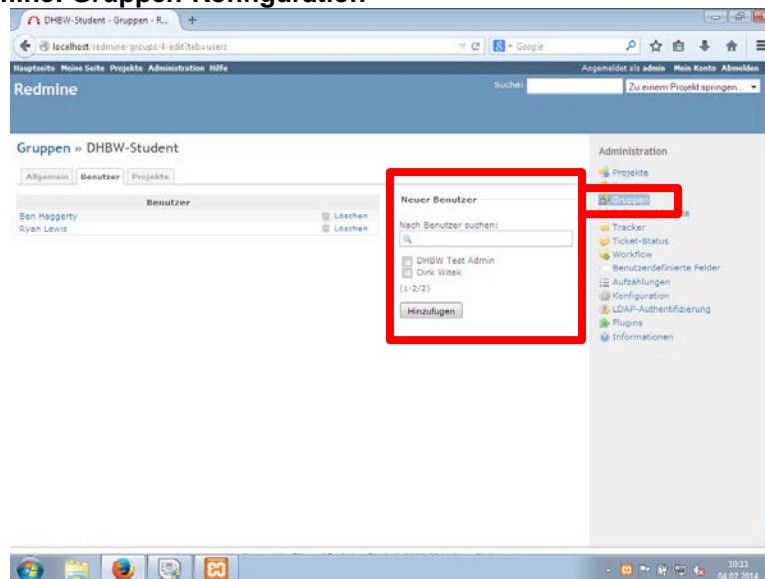
Nach der Bearbeitung eines Tickets ist jeweils der Status zu ändern. Der Administrator hat die Möglichkeit, weitere Ticket-Status zu erzeugen.

Anhang 13: Redmine: Tracker-Konfiguration



Ein Ticket kann unterschiedliche Tracker-Varianten aufweisen. Klickt der Nutzer in der rechten Menüleiste auf das Feld *Tracker*, so gelangt er zu dieser Übersicht. Dieser Parameter dient zur Einordnung des jeweiligen Tickets. Die standardmäßigen Tracker sind dabei *Fehler*, *Feature* und *Unterstützung*. Der Administrator kann weitere individuelle Tracker erstellen. Dieser Wert ist beim Erstellen des Tickets anzugeben.

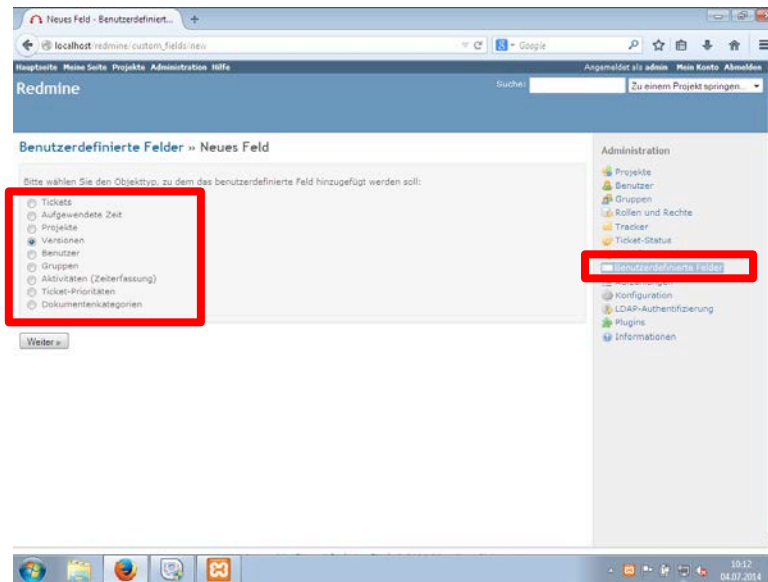
Anhang 14: Redmine: Gruppen-Konfiguration



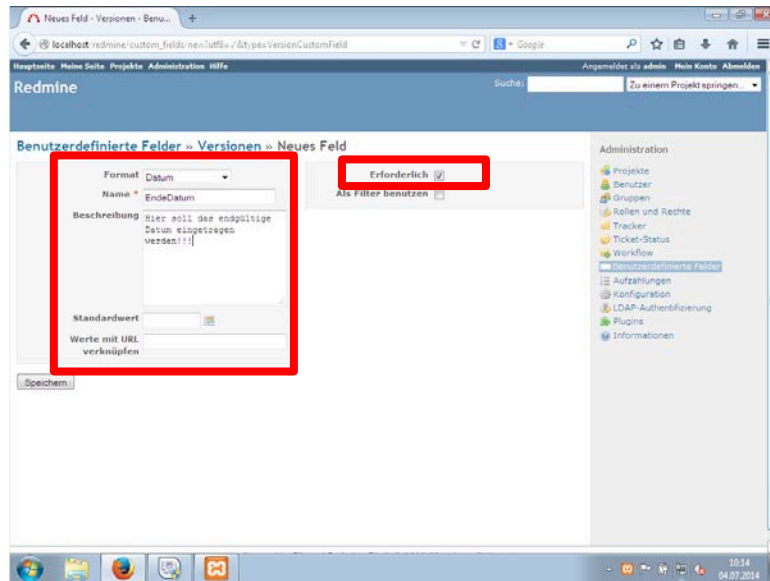
Redmine bietet die Möglichkeit Benutzergruppen anzulegen. In der Praxis können dies beispielsweise Teams oder Abteilungen im Unternehmen sein. In der Gruppe *DHBW-Student* befinden sich die Benutzer *Ben Haggerty* und *Ryan Lewis*. Im Be-

reich *Neuer Benutzer* können weitere Mitarbeiter der Gruppe zugeordnet werden. Sofern es wenige Benutzer im System gibt werden diese unter dem Suchfeld aufgelistet. Bei einer großen Anzahl empfiehlt es sich, die Suchfunktion zu benutzen.

Anhang 15: Redmine: Konfiguration von Feldern

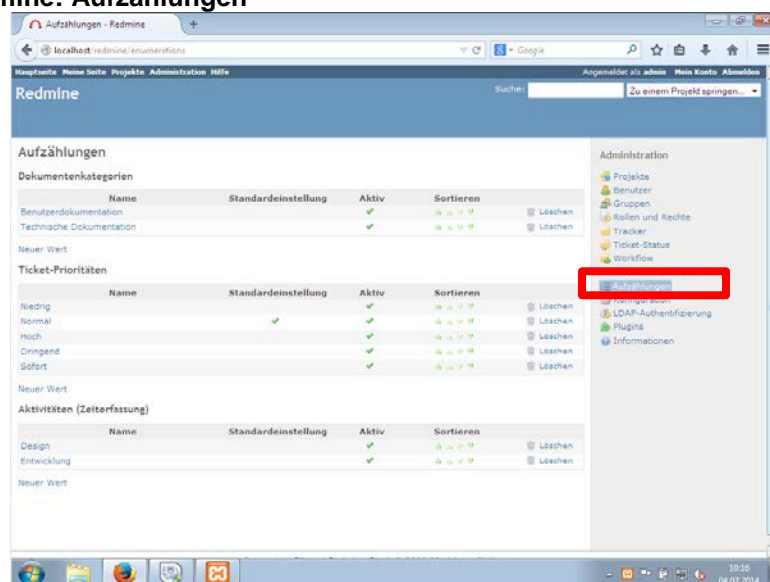


Redmine bietet die Möglichkeit, weitere Felder in den einzelnen Projekten und Tickets anzulegen. Das Menü ist über die Leiste am rechten Rand zu erreichen mit Klick auf *Benutzerdefinierte Felder*. In diesem Fall wird ein Feld für die einzelnen Versionen des Projekts erstellt.



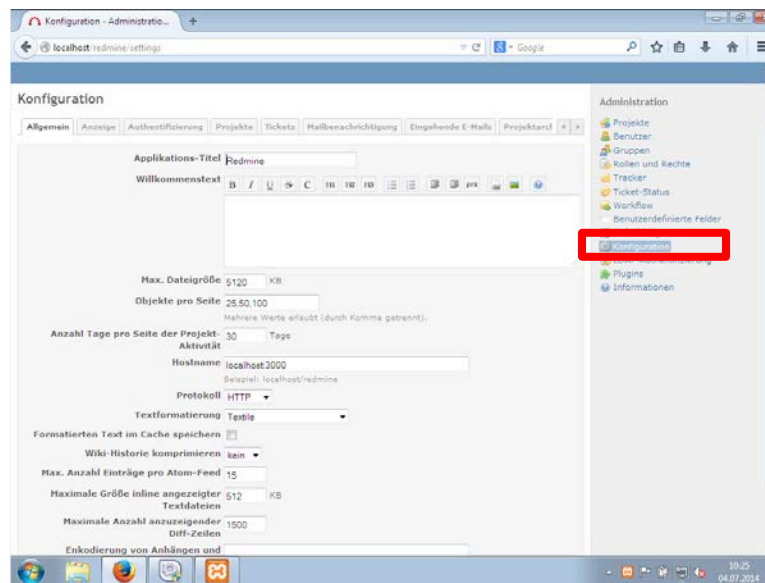
Im nächsten Schritt ist das Feld genauer zu konfigurieren. Es wird zunächst ein Datentyp eingestellt, den das neue Feld aufweisen soll. Das Feld erhält einen Namen und gegebenenfalls eine genauere Beschreibung. Zusätzlich kann vorgegeben werden, dass das neu Erstellte Feld ein Pflichtfeld darstellen soll. Dies ist mit dem Setzen eines Tickers bei *Erforderlich* zu tätigen. Mit Klick auf *Speichern* wird das Feld ins System eingebunden.

Anhang 16: Redmine: Aufzählungen



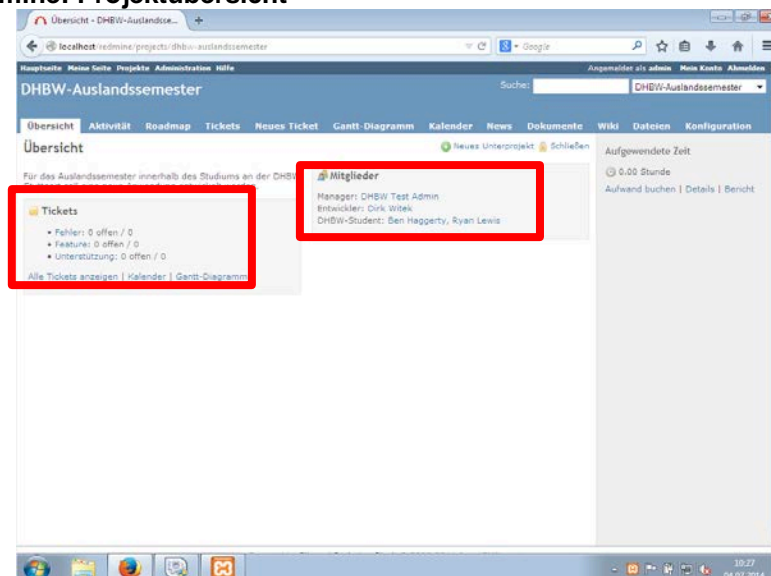
Innerhalb eines Projekts weisen die einzelnen Tickets unterschiedliche Prioritäten und Kategorien auf. Diese können unter dem Menüpunkt *Aufzählungen* konfiguriert werden. In diesem Fall sind die Rubriken *Dokumentenkategorien*, *Ticket-Prioritäten* und *Aktivitäten (Zeiterfassung)* vorhanden. Der Administrator kann weitere Aufzählungen anlegen und die Kategorisierung somit verfeinern.

Anhang 17: Redmine: Allgemeine Konfigurationen



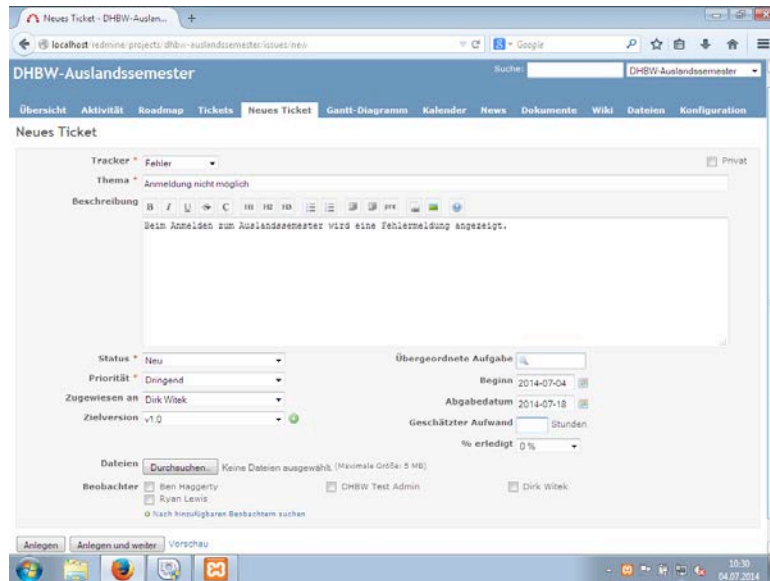
Unter *Konfiguration* können allgemeine Optionen für die Anwendung Redmine festgelegt werden. Diese Einstellungen gelten übergreifend für alle Projekte im System. Herauszuheben sind Werte wie beispielsweise der *Applikationstitel*, die *maximale Dateigröße* für Anhänge und der *Hostname*.

Anhang 18: Redmine: Projektübersicht

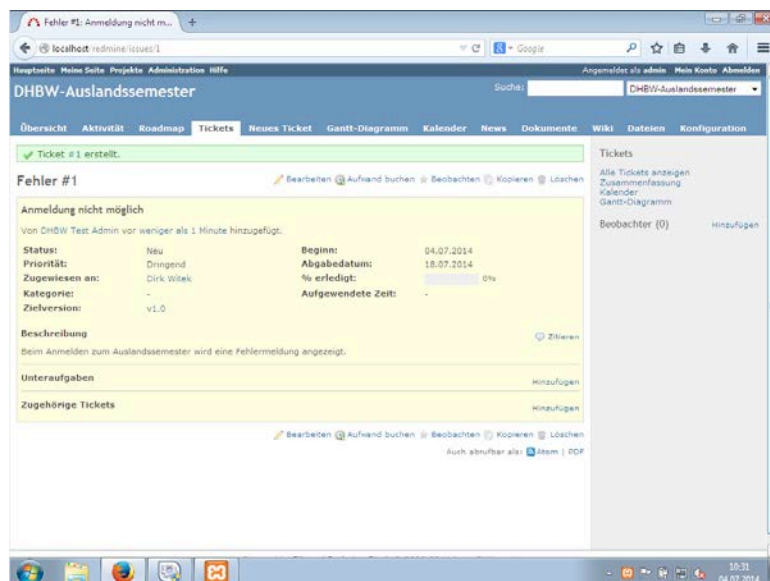


In der Übersicht des Projekts *DHBW-Auslandssemester* werden neben der kurzen Beschreibung auch die derzeitigen Mitglieder sowie die verfügbaren Tickets angezeigt. Des Weiteren ist in der rechten Leiste die bisher aufgewendete Zeit einzusehen, welche von den Nutzern jeweils manuell erfasst werden.

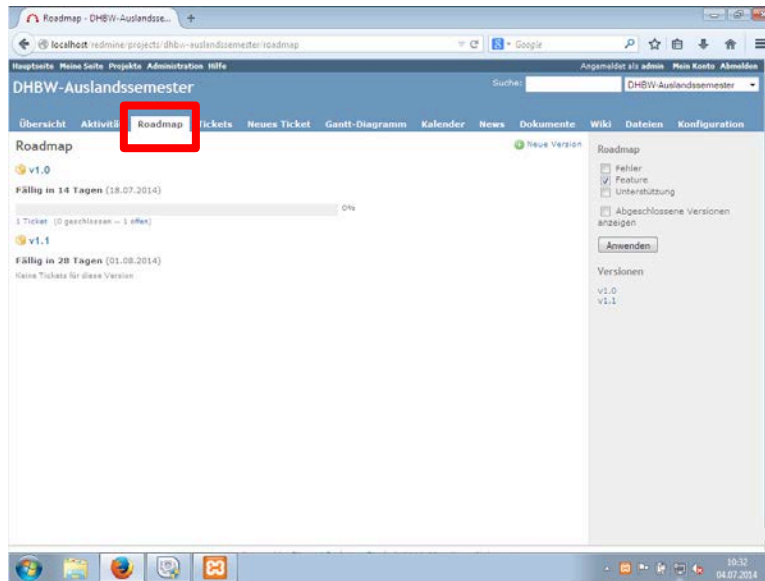
Anhang 19: Redmine: Anlegen eines Tickets



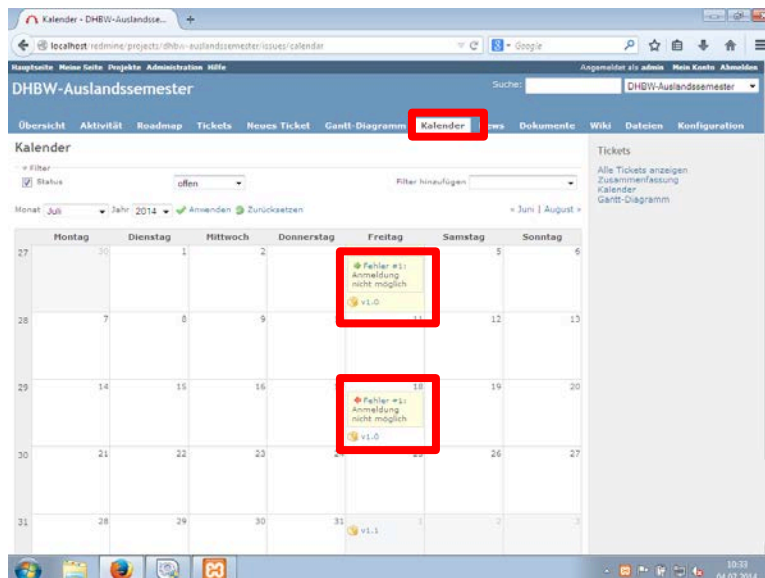
In der obigen Abbildung wird ein neues Ticket erstellt. Die mit einem * markierten Felder sind grundsätzlich aufgefüllt werden. Neben dem Tracker und dem Ticket-Thema, welches das Problem möglichst genau kategorisiert, kann eine ausführliche Beschreibung optional hinzugefügt werden. Diese kann gegebenenfalls mit einem Screenshot von einer Fehlermeldung versehen werden. Sofern klar ist, welche Person der Sachbearbeiter für das Ticket ist, wird dieser bereits in diesem Schritt dem User zugewiesen.



Das eben erstellte Ticket wird nun im Gesamten nochmals dargestellt.

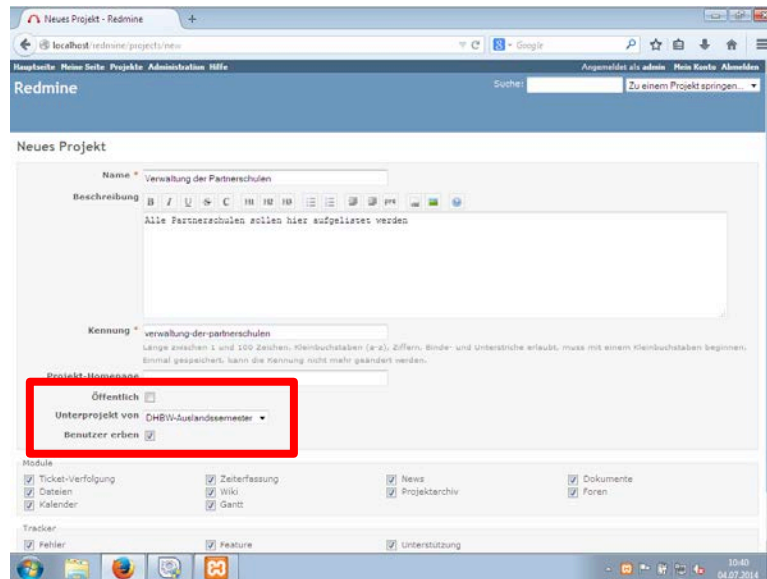


In der *Roadmap* ist unter Version 1.0 nun das Ticket vermerkt. Je nach Bearbeitungsfortschritt ist der Balken zwischen 0 und 100% ausgefüllt.

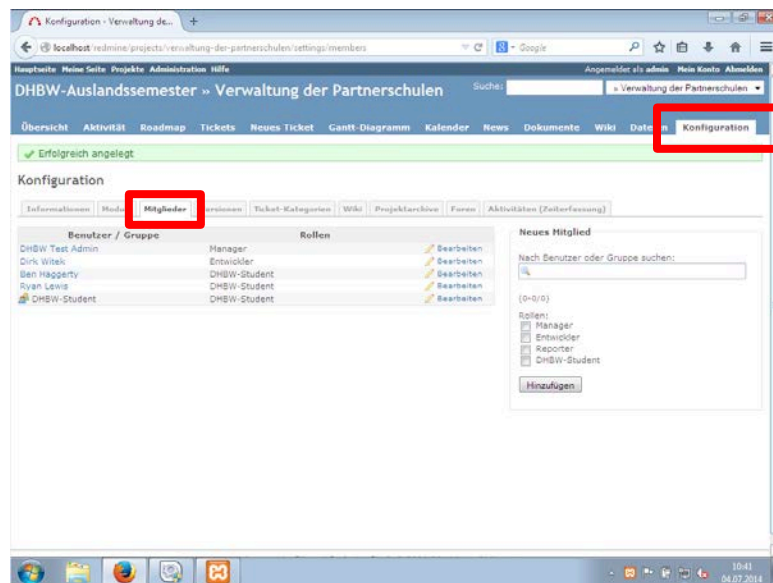


Im bereits vorgestellten Kalender ist nun auch automatisiert die Fehlermeldung eingetragen.

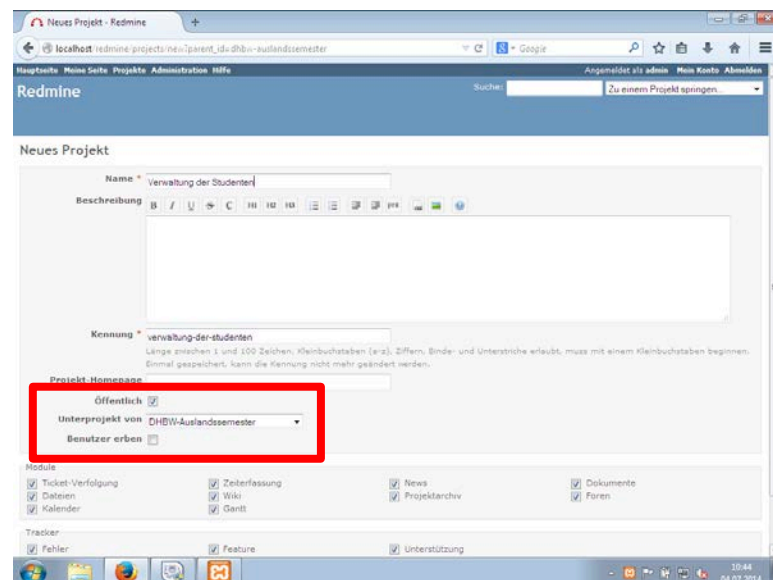
Anhang 20: Redmine: Erstellung eines Unterprojekts



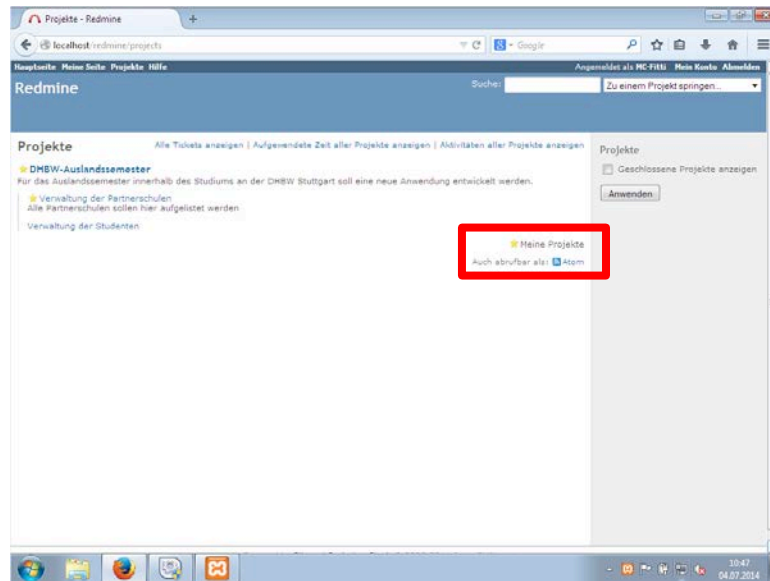
Ein Projekt kann mehrere Unterprojekte aufweisen. In diesem Schritt wird ein Unterprojekt des *DHBW-Auslandssemesters* namens *Verwaltung der Partnerschulen* erstellt. Dies ist im Dropdown-Feld auszuwählen, sofern dies gewünscht ist. Wird das Feld *Benutzer erben* gewählt, so übernimmt das Unterprojekt alle konfigurierten Nutzer des *DHBW-Auslandssemester-Projekts*. Das Feld *Öffentlich* wurde abgewählt. Dadurch ist das Unterprojekt für Nicht-Registrierte oder unberechtigte Personen nicht sichtbar.



Nachdem das Unterprojekt erfolgreich angelegt wurde, kann die User-Vererbung unter der Registerkarte *Mitglieder* im Bereich *Konfiguration* überprüft werden. Es ist zu erkennen, dass alle Benutzer erfolgreich eingetragen wurden.

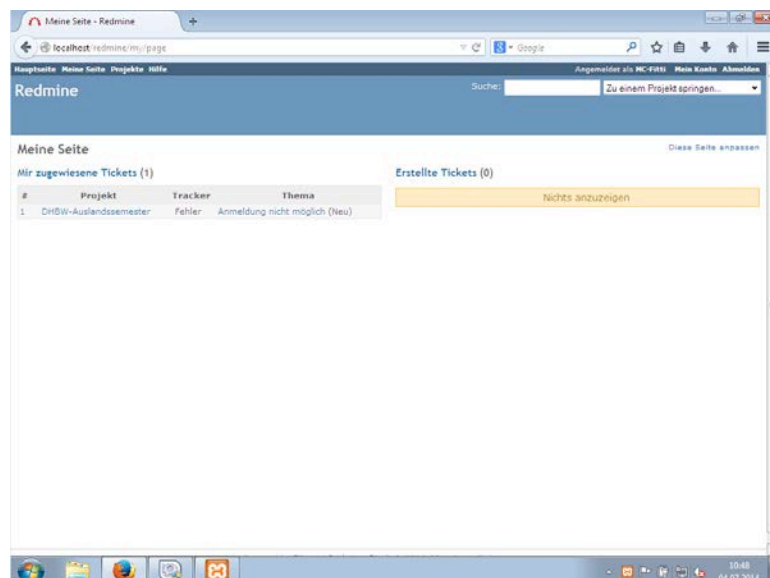


Ein weiteres Unterprojekt wird als *Öffentlich* konfiguriert. Folglich können auch Nicht-Registrierte Personen das Projekt beobachten. Zudem wird das Projekt keinem Nutzer explizit zugewiesen.

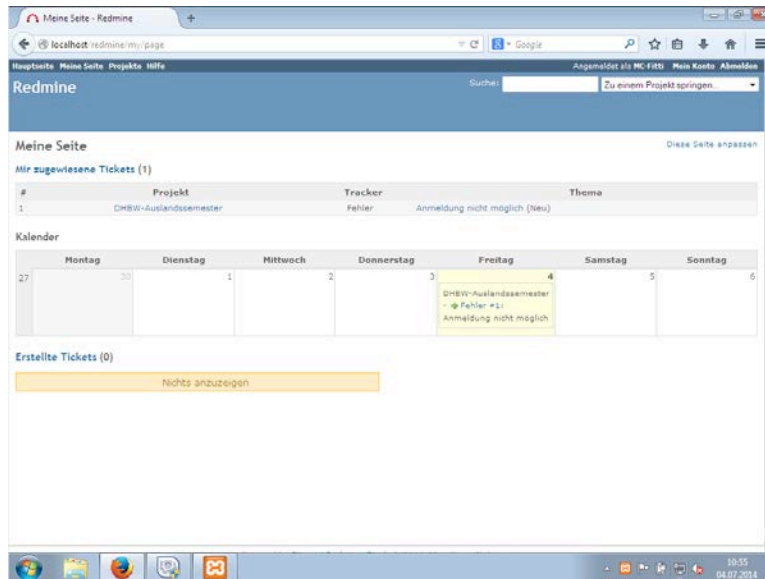


Bei der Projektübersicht ist zu erkennen, dass das Projekt *Verwaltung der Studenten* nicht mit einem gelben Stern versehen ist.

Anhang 21: Redmine: Erstellung von Widgets

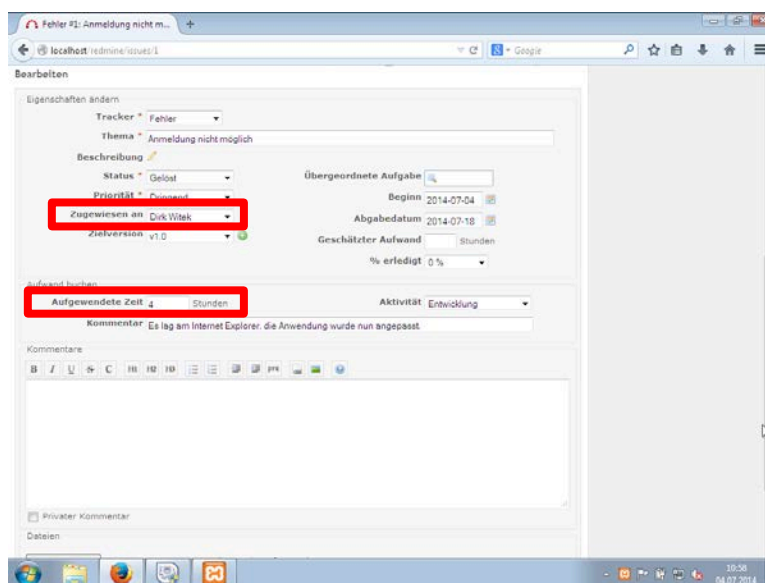


Die Nutzer haben die Möglichkeit, ihre persönliche Seite individuell mit Widgets zu versehen. So kann er seinen persönlichen Bereich mit den für ihn wichtigen Elementen kreieren.

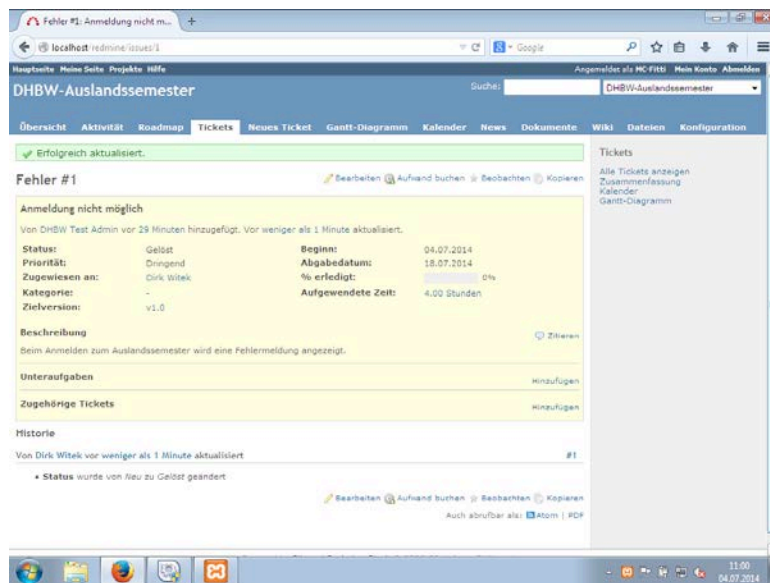


In diesem Fall wurde als Widget die Wochenansicht des Kalenders konfiguriert.

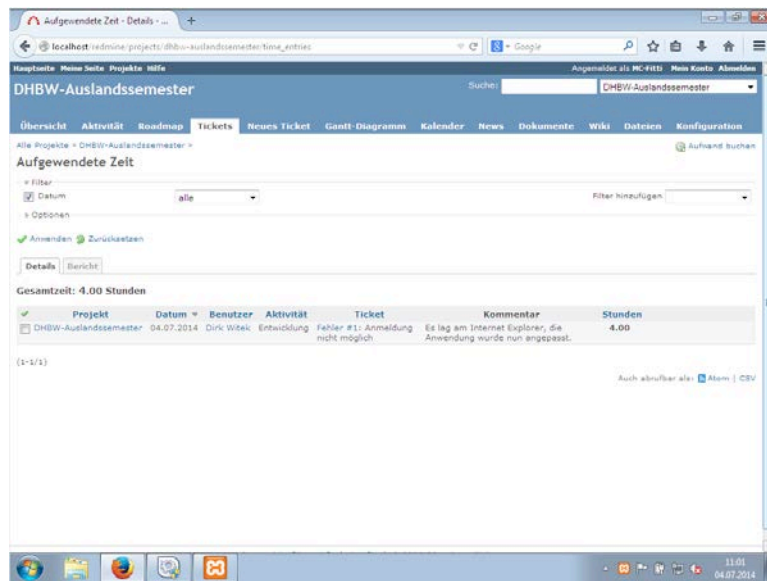
Anhang 22: Redmine: Bearbeitung eines Tickets



Sobald ein Ticket bearbeitet bzw. ein Fehler behoben wird, ist das jeweilige Ticket zu aktualisieren. *Dirk Witek* bearbeitet das Anliegen, setzt den Ticket-Status auf *Gelöst* und erfasst die *Aufgewendete Zeit*. Des Weiteren kann er das Ticket kurz kommentieren und auf diesem Wege angeben, was genau behoben wurde. Optional kann er einen ausführlichen Kommentar angeben, welches sich zusätzlich anhand eines Screenshots veranschaulichen lässt.



Nach der erfolgreichen Aktualisierung erhält Dirk Witek eine Bestätigung und eine erneute Übersicht über das geänderte Ticket.



In der allgemeinen Ticket-Übersicht des Projekts *DHBW-Auslandssemester* sind nochmals alle Tickets einzusehen. Bei einer großen Anzahl an Tickets kann der Nutzer die Filterfunktion nutzen, um das Spektrum einzuschränken. Das eben bearbeitete Ticket wird mitsamt dem verfassten Kommentar und der gebuchten angezeigt.

10 Quellenverzeichnisse

Literaturverzeichnis

Bon, J. van (2008): IT Service Management basierend auf ITIL V3, Das Taschenbuch, Zaltboomel: Van Haren Publishing

Böttcher, R. (2008): IT-Servicemanagement mit ITIL V3, Einführung, Zusammenfassung und Übersicht der elementaren Empfehlungen, Hannover: Heise Zeitschriften Verlag GmbH & Co KG

Buchsein, R. u. a. (2007): IT-Management mit ITIL® V3, Wiesbaden: Friedr. Vieweg & Sohn Verlag

Ebel, N. (2008): ITIL-V3-Basis-Zertifizierung, Grundlagenwissen und Zertifizierungsvorbereitung für die ITIL-Foundation-Prüfung, München: Addison Wesley

Elsässer, W. (2006): ITIL einführen und umsetzen, Leitfaden für ein effizientes IT-Management durch Prozessorientierung, 2. Auflage, München/Wien: Carls Hanser Verlag

Grassmuck, V. (2004): Freie Software, Zwischen Privat- und Gemeineigentum, 2. Auflage, Bonn: Bundeszentrale für politische Bildung

Henkel, J. (2007): Offene Innovationsprozesse, 1. Auflage, Wiesbaden: Deutscher Universitätsverlag

Hüttenegger, G. (2006): Open Source Knowledge Management, Heidelberg: Springer-Verlag

Keßler, S. (2013): Anpassung von Open-Source-Software in Anwenderunternehmen, Heidelberg: Springer-Verlag

Kersten, H./Klett, G. (2012): Mobile Device Management, Heidelberg/München/Landsberg/Frechen/Hamburg: mitp

Krcmar, H. (2005): Informationsmanagement, 4. überarbeitete und erweiterte Auflage, Heidelberg/Berlin: Springer-Verlag

Spillner, A./Linz T. (2010): Basiswissen Softwaretest, 4. Auflage, Heidelberg: dpunkt.verlag

Teich, I./Kolbenschlag, W./Reiners, W. (2008): Der richtige Weg zur Softwareauswahl, Berlin/Heidelberg: Springer-Verlag

Verzeichnis der Internet- und Intranet-Quellen

Adminitrack (2014): Product Info - Issue Tracking System for Professional Teams, <http://www.adminitrack.com/ProductInfo.aspx>, Abruf: 10.06.2014

Atlassian (2014): JIRA - Issue & Project Tracking Software, <https://www.atlassian.com/de/software/jira>, Abruf: 12.06.2014

Bugzilla (2014): Installation List, <http://www.bugzilla.org/installation-list/>, Abruf: 11.07.2014

Bugzilla (o.J.): The Bugzilla Guide, <http://www.bugzilla.org/docs/3.0/html/installation.html>, Abruf: 11.06.2014

GNU (2014): Was ist Copyleft, <http://www.gnu.org/copyleft/copyleft.html>, Abruf: 12.06.2014

GNU (o. J.): Open Source, <http://www.gnu.de/free-software/open-source.de.html>, Abruf: 11.06.2014

Goeken, M. u.a. (o. J.): Metamodelle von Referenzmodellen am Beispiel ITIL <http://subs.emis.de/LNI/Proceedings/Proceedings154/gi-proc-154-340.pdf>, Abruf: 02.07.2014

Mantis BT (2014): Mantis Bug Tracker, <https://www.mantisbt.org>, Abruf: 11.07.2014

Omninet (2013): OMNITRACKER / E-Tracking <http://www.omninet.de/index.php?id=138&L=0>, Abruf: 10.06.2014

Opensource (o. J.): The Open Source Definition, <http://opensource.org/osd>, Abruf: 11.06.2014

Schoyswohl, P. (2011): HowTo: Mantis – BugTracker Installation, <http://www.sysadminslife.com/linux/howto-mantis-bugtracker-installation-unter-debian>, Abruf: 11.07.2014

Sweo (2011): Trac installieren (Debian, Apache, modwsgi, Digest-Auth), <http://www.sweo.de/web-engineering/tools/trac-installieren-debian-apache-modwsgi-digest-auth>, Abruf: 11.07.2014

Trac (2014): Installing Trac, <http://trac.edgewall.org/wiki/TracInstall#InstallingTrac>, Abruf: 11.07.2014

Webmasterpro (o.J.): Bugtracker - Übersicht - Kostenlose und Kommerzielle Lösungen, <http://www.webmasterpro.de/coding/article/php-bugtracker-uebersicht.html>, Abruf: 11.07.2014

Wikipedia (2013): Bugtracker, <http://de.wikipedia.org/wiki/Bug-Tracker>, Abruf: 16.06.2014

Wikipedia (2014a): Programmfehler, <http://de.wikipedia.org/wiki/Programmfehler>,
Abruf: 14.06.2014

Wikipedia (2014b): Exploit, <http://de.wikipedia.org/wiki/Exploit>, Abruf: 14.06.2014

Wikipedia (o. J.): Proprietäre Software,
http://de.wikipedia.org/wiki/Propriet%C3%A4re_Software, Abruf: 13.06.2014

Vergleich von Open-Source-ETL-Tools und Erstellung eines Prototyps

Schriftliche Ausarbeitung
im Rahmen der Lehrveranstaltung „Integrationsseminar“
Kompetenzzentrum Open Source (KOS)

Vorgelegt von

Horlacher Marco,
Martins Pinheiro Johanna,
Meierhuber Felicitas,
Wanner Jonas

am 14.07.2014

Fakultät Wirtschaft
Studiengang Wirtschaftsinformatik
WWI2011V

Inhaltsverzeichnis

Abkürzungsverzeichnis	IV
Abbildungsverzeichnis.....	V
Tabellenverzeichnis.....	VI
1 Einleitung	1
1.1 Motivation.....	1
1.2 Gliederung und Ziel der Arbeit.....	2
2 Theoretische Grundlagen	3
2.1 Open-Source.....	3
2.2 ETL- Prozess	4
2.2.1 Extraktion.....	4
2.2.2 Transformation.....	5
2.2.3 Laden.....	5
3 Bewertung und Auswahl der ETL-Tools	6
3.1 Vorgehensweise.....	6
3.2 Marktanalyse.....	6
3.3 Kurzbeschreibung der zu bewertenden Tools	7
3.3.1 Open Talend	8
3.3.2 Smooks.....	8
3.3.3 Jaspersoft ETL.....	8
3.3.4 KETL.....	9
3.3.5 Apatar	9
3.4 Grobauswahl der ETL-Tools.....	9
3.4.1 Grundlegende Voraussetzungen zur Tauglichkeit der Tools	10
3.4.2 Grobkriterium 1 – Dokumentation und Informationsvielfalt	10
3.4.3 Grobkriterium 2 – Funktionalitätsumfang	12
3.4.4 Grobkriterium 3 – Downloadzahlen	14
3.4.5 Grobkriterium 4 – Nennenswerte Partner	15
3.4.6 Grobkriterium 5 – Ergonomie und Benutzerfreundlichkeit.....	16
3.4.7 Abschließende Beurteilung aller Grobkriterien	18
3.5 Feinauswahl der ETL-Tools	18
3.5.1 Erläuterung des Feinkriterienkatalogs	19
3.5.2 Produktprofil.....	20
3.5.2.1 Kategorie 1 – Ökosystem des Herstellers	20
3.5.2.2 Kategorie 2 – Funktionalität.....	22

3.5.2.3	Kategorie 3 – Leistung	25
3.5.2.4	Kategorie 4 – Bedienbarkeit/ Ergonomie/ Handhabung.....	27
3.5.2.5	Kategorie 5 – Performance/ Effizienz	30
3.5.2.6	Kategorie 6 – Daten- und Zukunftssicherheit.....	31
3.5.3	Anbieterprofil.....	33
3.5.3.1	Kategorie 1 – Marktakzeptanz.....	33
3.5.3.2	Kategorie 2 – Aktivität der Community	37
3.5.3.3	Kategorie 3 – Dokumentation.....	40
3.5.3.4	Kategorie 4 – Dienstleistungsangebot.....	41
3.5.3.5	Kategorie 5 – Finanzierbarkeit	43
3.5.4	Ergebnis der Feinauswahl.....	46
4	Prototypische Umsetzung mit Open Talend	47
4.1	Konzept zur Umsetzung der Transformation	47
4.2	Erstellung einer großen Datenbasis	48
4.3	Datenbank-Verbindung einrichten	49
4.4	Integration der Daten	50
5	Fazit	53
	Anhang.....	55
	Quellenverzeichnisse	61

Abkürzungsverzeichnis

BI	Business Intelligence
CRM	Customer Relationship Management
CSV	Comma-separated values
ER	Entity Relationship
ERP	Enterprise Resource Planning
ETL	Extraktion, Transformation, Laden
FAQ	Frequently Asked Questions
GPLv2	GNU General Public License version 2.0“
IDE	integrierte Entwicklungsumgebung
OEM	Original Equipment Manufacturer
OSI	Open-Source Initiative
TCO	Total Cost of Ownership
XML	Extensible Markup Language

Abbildungsverzeichnis

Abbildung 1: ETL-Prozess.....	4
Abbildung 2: Vorgehensweise zur Auswahl eines ETL-Tools	6
Abbildung 3: Marktübersicht der JAVA-basierten Open-Source-ETL-Tools	7
Abbildung 4: Ausdruckseditor des Open Talends	23
Abbildung 5: Talend Service Pakete.....	38
Abbildung 6: Leistungsumfang der angebotenen Talend Versionen	45
Abbildung 7: Konzept zur Umsetzung des Prototyps	47
Abbildung 8: ER-Modellierung der unterschiedlichen Datenbanktabellen	49
Abbildung 9: Job mit unterschiedlichen Komponenten	51
Abbildung 10: Transformation von Attributen.....	51

Tabellenverzeichnis

Tabelle 1: Vergleich in Bezug auf die Tauglichkeit der Tools.....	10
Tabelle 2: Vergleich der Tools im Hinblick auf die Dokumentation und Informationsvielfalt ..	12
Tabelle 3: Vergleich der Tools im Hinblick auf den Funktionalitätsumfang.....	14
Tabelle 4: Vergleich der Tools im Hinblick auf die Downloadzahlen	15
Tabelle 5: Vergleich der Tools im Hinblick auf nennenswerte Partner	16
Tabelle 6: Vergleich der Tools im Hinblick auf die Ergonomie und Benutzerfreundlichkeit....	18
Tabelle 7: Abschließende Beurteilung aller Grobkriterien	18
Tabelle 8: Beispiel zur Erklärung des Feinkriterienkatalogs.....	19
Tabelle 9: Beispiel zur Erklärung des Gesamtergebnisses	20
Tabelle 10: Beurteilung im Hinblick auf die Ökonomie des Herstellers	22
Tabelle 11: Beurteilung im Hinblick auf die Funktionalität	25
Tabelle 12: Ergebnis der Testläufe	27
Tabelle 13: Beurteilung im Hinblick auf die Leistung.....	27
Tabelle 14: Beurteilung im Hinblick auf die Bedienbarkeit/ Ergonomie/ Handhabung	30
Tabelle 15: Beurteilung im Hinblick auf die Performance/ Effizienz	31
Tabelle 16: Beurteilung im Hinblick auf die Daten-/ Zukunftssicherheit.....	33
Tabelle 17: Beurteilung im Hinblick auf die Marktakzeptanz	37
Tabelle 18: Beurteilung im Hinblick auf die Aktivität der Community.....	39
Tabelle 19: Beurteilung im Hinblick auf die Dokumentation	41
Tabelle 20: Beurteilung im Hinblick auf das Dienstleistungsangebot	43
Tabelle 21: Beurteilung im Hinblick auf die Aktivität der Community.....	46
Tabelle 22: Entgültiges Ergebnis der Feinauswahl	46
Tabelle 23: CSV Beispieldatei	49

1 Einleitung

"When it comes to software,
I much prefer open source software,
because I have very seldom seen a program that has worked well enough for my needs,
and having sources available can be a life-saver."

Linus Torvalds, Erfinder des Linux-Kernels

Im Jahre 1991 hat der Student Linus Trovalds einen entscheidenden Teil zu dem bekanntesten Open-Source-Produkt beigetragen, indem er den Linux-Kernel entwickelte. Laut Zitat bevorzugt der Erfinder des Linux-Kernels Open-Source-Software aus dem Grund, dass ein Programm meist nie hundertprozentig die Bedürfnisse des Anwenders abdeckt und somit angepasst werden muss, was die Verfügbarkeit des Quellcodes voraussetzt.

1.1 Motivation

Aus Sicht von Unternehmen gibt es neben dem genannten Grund von Herrn Trovalds noch einige weitere.

Der „Future of Open Source Survey“, welcher sich auf die Befragung von 1.240 Teilnehmern mit unterschiedlicher Funktion aus verschiedenen Unternehmen stützt, ergab, dass Unternehmen, die sich im Jahr 2013 für Open-Source-Produkte entschieden haben, dies wegen der Herstellerunabhängigkeit, den geringeren Kosten, der Sicherheit und der Qualität taten.¹ Jedoch muss beachtet werden, dass die Studie von Firmen gesponsert wurde, welche mit Open-Source Gewinne erzielen. Allerdings liefert eine unabhängige Studie des Fraunhofer Instituts aus dem Jahr 2005 ein ähnliches Ergebnis. Neben den bereits angesprochenen Gründen, wurden in dieser Unternehmensbefragung von 603 baden-württembergischen Unternehmen auch die Anpassungsfähigkeit sowie die Zuverlässigkeit als Vorteile von Open-Source genannt.²

Die Schwerpunkte des Open-Source Einsatzes liegen laut dem „Future of Open Source Survey“ mit über 50 Prozent Marktanteil von Open-Source-Produkten in den Bereichen Cloud/Virtualisierung, Content Management und Mobile. Im Analytics/Business Intelligence (BI) Bereich nutzen nur etwa ein Fünftel aller Umfrageteilnehmer Open-Source-Software.

Der Auftraggeber möchte beim Extrahieren, Transformieren und Laden der Daten für das Data Warehouse Optimierungen vornehmen. Der Grund für diese Entscheidung liegt darin, dass die Aktualität der Daten nicht ausreichend ist, da aufgrund von zeitaufwändigen Trans-

¹ Vgl. Black Duck (2014)

² Vgl. Haneke, U./u. a. (2010), S.54

aktionsverfahren die Daten nur monatsaktuell im Data Warehouse vorhanden sind. Nun überlegt die IT, ob es nicht sinnvoller ist, von der unternehmensinternen Eigenentwicklung auf ein bereits vorhandenes Open-Source-Produkt oder eventuell auch auf ein kommerzielles Produkt umzusteigen.³

1.2 Gliederung und Ziel der Arbeit

Wegen den vom Auftraggeber gemachten Überlegungen wird in dieser Seminararbeit, nachdem die theoretischen Grundlagen zu Open-Source und dem ETL-Prozess gelegt wurden, eine Marktanalyse durchgeführt, in welcher ein Überblick über die zur Verfügung stehenden ETL-Werkzeuge im Open-Source-Bereich geschaffen wird. Danach werden aus der Marktanalyse fünf Tools für die Grobauswahl übernommen. Die zwei am besten beurteilten Werkzeuge werden anschließend anhand eines Kriterienkatalogs detailliert beurteilt und das überzeugendere Werkzeug wird für die Erstellung eines Prototyps verwendet, um daraus eine konkrete Empfehlung für den Auftraggeber auszusprechen. Abgerundet wird die Arbeit durch ein Fazit, in welchem noch einmal die wichtigsten Ergebnisse festgehalten werden.

Das Ziel der Seminararbeit besteht somit darin, den Auftraggeber bei den Vorüberlegungen zur Einführung eines Open-Source-ETL-Tools zu unterstützen und durch den Prototypen die besondere Eignung eines bestimmten Werkzeugs für dieses Vorhaben zu verdeutlichen.

³ Vgl. Horlacher, M./u. a. (15.05.2014)

2 Theoretische Grundlagen

In den nachfolgenden Kapiteln wird zum einen auf den Begriff Open-Source und zum anderen auf den ETL-Prozess eingegangen.

2.1 Open-Source

Für Open-Source gibt es eine Vielzahl von Definitionen. Die anerkannteste stellte die Open-Source Initiative (OSI) auf, welche den Begriff wie folgt bestimmt: „Open source software is software that can be freely used, changed, and shared (in modified or unmodified form) by anyone. Open source software is made by many people, and distributed under licenses that comply with the Open Source Definition.“⁴

Neben dieser Kurzdefinition hat die OSI die nachfolgenden zehn Kriterien aufgestellt, die Aspekte der Begriffsbestimmung vertiefen und noch weitere beinhalten:

1. Freie Weitergabe
2. Programm muss Quellcode beinhalten
3. Abgeleitete Software
4. Unversehrtheit des Quellcodes des Autors
5. Keine Diskriminierung von einzelnen Personen oder Gruppen
6. Keine Einschränkung für bestimmte Anwendungsgebiete
7. Weitergabe der Lizenz
8. Lizenz darf nicht nur für ein bestimmtes Produkt gelten
9. Lizenz darf die Weitergabe nicht einschränken
10. Lizenz muss neutral in Bezug auf die Technologie sein

Nur bei Erfüllen all dieser Aspekte darf ein Produkt als Open-Source angeboten werden.⁵

Oft werden die beiden Begriffe Freie Software und Open-Source Software gleich gesetzt. Allerdings legen vor allem die Anhänger der Freien Software Bewegung großen Wert auf eine Differenzierung der beiden Begriffe. Laut diesen ist der signifikanteste Unterschied, dass „Open Source [...] eine Entwicklungsmethodik [...] [und] Freie Software [...] eine soziale Bewegung“⁶ ist.

⁴ Open Source Initiative (2014a)

⁵ Vgl. Open Source Initiative (2014b)

⁶ GNU/FSF (2014)

2.2 ETL- Prozess⁷

Die drei wesentlichen Schritte der Datenkonsolidierung für ein Data Warehouse oder ein Data Mart bestehen aus der Extraktion, der Transformation und dem Laden der Daten. Dieser ETL-Prozess „beschreibt den Vorgang, Daten aus bestehenden Datenquellen zu *extrahieren*, mittels geeigneter Transformationsregeln zu *homogenisieren*, nach bestimmten Vorschriften zu *bereinigen* und ggf. *anzureichern* und in einem separaten Ziel zu *speichern*“⁸. Somit bleiben die Quellsysteme unverändert bestehen. Alle Daten, die das Data Warehouse oder Data Mart benötigt, sind in eigenständigen Zielsystemen vorhanden. Der allgemeine ETL-Prozess ist in der nachfolgenden Abbildung 1 verdeutlicht.

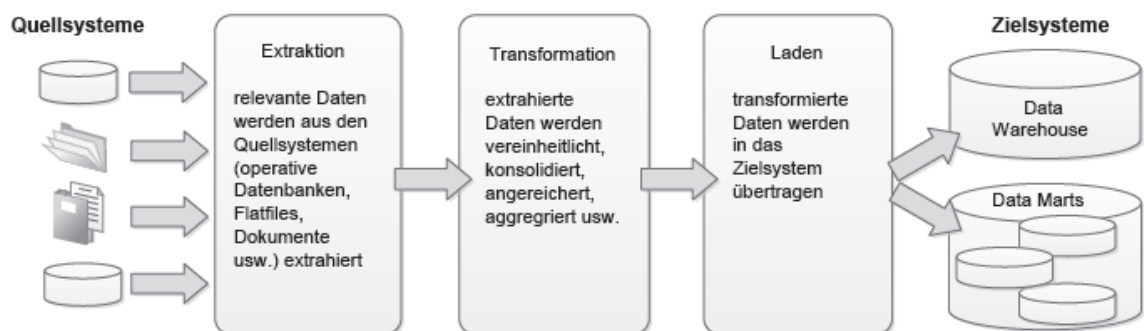


Abbildung 1: ETL-Prozess⁹

Der ETL-Prozess stellt die Grundlage für alle weiteren Analysen dar und ist ausschlaggebend für die Qualität des Data Warehouse oder des Data Marts.

2.2.1 Extraktion

Die Extraktion der Daten aus verschiedenen Systemen ist der erste Schritt des ETL-Prozesses. Zu solchen Vorsystemen zählen unter anderem ERP (Enterprise Resource Planning), CRM (Customer Relationship Management), andere Datenbanken oder auch nur Dateien. Da die Daten aus unterschiedlichen Quellsystemen stammen, sind diese meist in verschiedenen Formaten gespeichert, wie beispielsweise XML, Excel, CSV, und/oder variieren auch in ihrer Struktur. Hierbei wird zwischen strukturiert - zum Beispiel Tabellen -, semistrukturiert - zum Beispiel XML-Dateien - und unstrukturiert - zum Beispiel Briefe - unterschieden. Daher muss der Extraktionsprozess sowohl technische als auch syntaktische Unterschiede der Quellsysteme bewältigen.

Bei der Datenentnahme wird zwischen dem Push- und dem Pull-Verfahren unterschieden. Das Push-Verfahren ist vor allem für kleine Datenmengen geeignet, wohingegen das Pull-

⁷ Vgl. Rossak, I. (2013), S.37 ff.

⁸ Ebenda, S.37

⁹ Enthalten in: ebenda, S.37

Verfahren für Massendaten verwendet wird, da hier die Datenänderungen aus den Quellsystemen „gezogen“ werden und nicht „geschoben“ werden müssen.

Weiterhin wird die Extraktion je nach Zeitpunkt differenziert:

- Periodische Extraktion: Die Datenentnahme erfolgt zu festgelegten sich wiederholenden Intervallen
- Anfragegesteuerte Extraktion: Die Datenentnahme erfolgt je nach Bedarf
- Ereignisgesteuerte Extraktion: Die Datenentnahme wird durch ein Ereignis angestoßen
- Permanente Extraktion: Die Datenentnahme erfolgt ununterbrochen

Außerdem kann zu dem jeweiligen Zeitpunkt entweder der komplette Datenbestand übernommen werden oder nur die seit der letzten Datenentnahme durchgeführten Änderungen.

2.2.2 Transformation

Im zweiten Schritt werden die Daten transformiert. Das bedeutet, die extrahierten Daten werden konsolidiert und homogenisiert. Dabei werden die folgenden Schritte ausgeführt:

- Syntaktische Transformation: „Homogenisierung und Konsolidierung im Hinblick auf Zeichensätze, Datentypen, Datenformate usw.“¹⁰
- Strukturelle Transformation: Homogenisierung und Konsolidierung beispielsweise der Primär- und Fremdschlüssel oder der Modellierungsebenen
- Semantische Transformation: Homogenisierung und Konsolidierung der Semantik, zum Beispiel einheitliche Angabe der Währung mit „€“

ETL-Tools unterstützen durch bereits definierte Transformationskomponenten, welche der Anwender auswählen und individuell anpassen kann, die einzelnen Schritte der Transformation. Auch besteht die Möglichkeit, die extrahierten Daten mithilfe von Daten aus externen Quellen zu ergänzen, beispielsweise wenn Werte fehlen, oder die bestehenden Werte auf Plausibilität zu prüfen.

2.2.3 Laden

Zuletzt erfolgt der Ladeprozess, in dem die extrahierten und transformierten Daten in das Zielsystem gespeichert werden. Somit sind nun die Daten aus den Quellsystemen mit denen aus dem Zielsystem redundant, um eine schnelle Datenbereitstellung für den BI-Nutzer zu gewährleisten und Analysen über einen langen Zeitraum zu ermöglichen.

¹⁰ Rossak, I. (2013), S.39

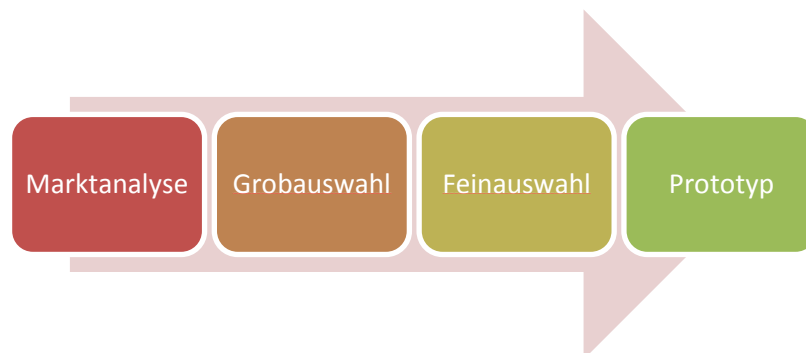
3 Bewertung und Auswahl der ETL-Tools

In den nachfolgenden Kapiteln wird zuerst die Vorgehensweise zur Auswahl und Bewertung der ETL-Tools beschrieben. Anschließend erfolgt die Marktanalyse, die Grobauswahl und die Feinauswahl.

3.1 Vorgehensweise

Für diese Studienarbeit wird zuerst eine Marktanalyse für Open-Source-ETL-Tools durchgeführt. Anschließend werden fünf Produkte in die Grobauswahl übernommen und nach Grobkriterien beurteilt. Die zwei besten Werkzeuge werden danach anhand eines Feinkriterienkatalogs bewertet. Zuletzt wird ein Prototyp mit dem Favorit des Auswahlprozesses umgesetzt. Grafisch veranschaulicht ist die Vorgehensweise in Abbildung 2.

Abbildung 2: Vorgehensweise zur Auswahl eines ETL-Tools



3.2 Marktanalyse

Für den ETL-Prozess gibt es eine Vielzahl an Tools. Die nachfolgende Marktanalyse konzentriert sich auf die Werkzeuge, welche mit der Programmiersprache JAVA implementiert wurden. Der Grund hierfür liegt in der Popularität der Programmiersprache und darin, dass im Rahmen dieser Studienarbeit für die Umsetzung des Prototyps in einer anderen Programmiersprache die Ressourcen fehlen würden.

Das Ergebnis der Marktanalyse ist in Abbildung 3 auf der nachfolgenden Seite festgehalten.



Abbildung 3: Marktübersicht der JAVA-basierten Open-Source-ETL-Tools¹¹

Bei vielen der in Abbildung 3 genannten Werkzeuge ist zwar der Sourcecode verfügbar, allerdings existiert keine explizite Internetseite. Weiterhin fällt es bei einigen Werkzeugen schwer, überhaupt passende Informationen zu finden. Zur Untersuchung eignen sich daher nach einer ersten Analyse lediglich: Open Talend, KETL, Apatar, Jaspersoft ETL, Smooks, Kettle Penthao, Adeptia, Jitterbit, Xineo, CloverETL und Data Pipeline. Diese Tools werden in die Grobauswahl übernommen.

3.3 Kurzbeschreibung der zu bewertenden Tools

In der Grobauswahl dieser Seminararbeit werden nur fünf der zuvor aufgezählten Werkzeuge bewertet: Open Talend, KETL, Apatar, Jaspersoft ETL und Smooks. Die restlichen Tools werden von einer zweiten Gruppe in einer separaten Arbeit bewertet.

Nachstehend erfolgt eine kurze Vorstellung der fünf zu bearbeitenden Tools. Dies dient dazu, einen Gesamtüberblick zu vermitteln und das Verständnis für die späteren Kriterienbeurteilungen zu erleichtern.

¹¹ In Anlehnung an: Manageability (2004)

3.3.1 Open Talend

Das Open-Source-Tool Talend, auch Open Talend, nur Talend, Talend Open Studio oder exakt Talend Open Studio for Data Integration genannt, stellt eine komplette Plattform im Open-Source-Bereich für den Umgang mit elektronischen Daten zur Verfügung. Diese ermöglicht es, der IT alle Daten zu liefern, die für den geschäftlichen Erfolg eines Unternehmens wichtig sind und dies egal wo, wann oder wie. Die Grundlage dafür stellt ein modularer Systemaufbau dar, der durch unterschiedlichste Softwarelösungen auf die eigenen Bedürfnisse abgestimmt werden kann.¹²

Durch den angebotenen Funktionsumfang und einer starken Wachstumsrate bietet das Unternehmen Talend eine zukunftsorientierte Investition im Bereich Big Data und der Datenintegration, welche durch die eigene Weiterentwicklung stetig auf dem aktuellsten Stand gehalten wird.¹³ Das im Jahr 2006 gegründete Unternehmen umfasst bereits heute über 400 Mitarbeiter und wächst dank immer neuer Investoren in einem rasanten Tempo weiter.¹⁴

3.3.2 Smooks

Smooks ist ein Open-Source-Framework im Bereich der Transformationstechnologie. Neben der Möglichkeit Daten aus beliebiger Quelle und beliebigem Format in das gewünschte Zielformat zu überführen, wird auch das Routing von Nachrichten unterstützt. Dies wird als „Structured Data Event Stream Processor“ bezeichnet. Andere Lösungen wie DOM oder SAX sind vergleichbar, können aber nicht die gebotene Vielfalt an möglichen Umwandlungstypen unterstützen, sind langsamer und zusätzlich aufwendiger in der Konfiguration.¹⁵

3.3.3 Jaspersoft ETL

Jaspersoft bietet eine vollständige BI-Lösung an, in welcher Jaspersoft ETL integriert ist. Allerdings ist es auch möglich, Jaspersoft ETL separat herunterzuladen. Insgesamt gibt es fünf verschiedene Softwarepakete: Community, Express, Aws, Professional und Enterprise.¹⁶ Allerdings ist nur die Community Edition als Open-Source verfügbar. Speziell für Jaspersoft ETL bietet die kommerzielle Version gegenüber der Community Version noch einige zusätzliche Funktionalitäten wie zum Beispiel einen „job coordinator“ oder einen „data viewer“.¹⁷ Im

¹² Vgl. Talend (2014a)

¹³ Vgl. ebenda

¹⁴ Vgl. ebenda; Vgl. dazu auch Talend (2013)

¹⁵ Vgl. Demolsky, M. (2009)

¹⁶ Vgl. Jaspersoft (2014a)

¹⁷ Vgl. Jaspersoft (2014b)

ETL-Bereich kooperiert Jaspersoft mit Talend und daher sind die Funktionalitäten zwischen Jaspersoft ETL, auch JasperETL genannt, und dem Open Talend identisch.¹⁸

3.3.4 KETL

Das ETL-Tool KETL, welches von Kinetic Networks stammt, bietet Unternehmen eine komplexe Lösung zur Datenmanipulation und -verwaltung an. Es stellt dafür eine metadatengetriebene Lösung bereit, die sich – nach Angaben des Herstellers – insbesondere durch ihre Robustheit, ihre Funktionalitäten und ihre gute Skalierbarkeit auszeichnet.

KETL eignet sich jedoch nicht für alle Anwender, da es über keine grafische Oberfläche verfügt, sondern eine reine Konsolenanwendung ist, in welcher die ETL-Jobs über XML Job Definition Language gesteuert werden. Somit setzt es einige Kenntnisse in diesen Bereichen voraus.¹⁹

3.3.5 Apatar

Apatar Data Integration/ETL, wie der vollständige Name lautet, ist eine kostenlose und schnell zu installierende Datenintegrations-Software, die dazu entwickelt wurde, um sowohl Anwender als auch Entwickler beim ETL-Prozess zu unterstützen. Hierzu unterstützt Apatar zahlreiche Datenquellen und -formate. Das Tool greift dafür die Bedürfnisse der Anwender auf, dies ohne Codierungsaufwand durchführen zu können. So können selbst komplexe ETL-Prozesse über die grafische Anwendungsoberfläche gelöst werden.²⁰

3.4 Grobauswahl der ETL-Tools

Im Folgenden wird zunächst geprüft, ob einige grundlegende Voraussetzungen zur Tauglichkeit des Tools gegeben sind () oder nicht () beziehungsweise ob trotz ausgiebiger Suche keine verlässlichen Angaben dazu gefunden werden konnten (). Anschließend werden die fünf verwendeten Grobkriterien kurz erläutert, wobei im direkten Anschluss an jede Erläuterung eine kurze Beurteilung der einzelnen Tools zum beschriebenen Kriterium folgt. Die Beurteilung der Grobkriterien wird jeweils durch eine Bewertungsübersicht über alle Tools abgeschlossen, in welcher nach dem Punktesystem 0-5 Punkte (sehr schlecht bis sehr gut) pro Tool und Kriterium vergeben werden können. Nach Abschluss aller Grobkriterien folgt eine weitere Bewertungsübersicht, welche die Summierung aller Grobkriterien enthält und als Basis für die anschließende Feinauswahl genutzt wird.

¹⁸ Vgl. Jaspersoft (2008)

¹⁹ Vgl. KETL (2014a)

²⁰ Vgl. Apatar (2014a)

3.4.1 Grundlegende Voraussetzungen zur Tauglichkeit der Tools

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 und hierarchische Datenbanken unterstützt werden. Des Weiteren soll die Angemessenheit des Produktes gegeben sein, worunter die Eignung für die Nutzung und Ausführung spezieller Aufgaben verstanden wird. Die Erfüllung der genannten Kriterien ist in der Tabelle 1 für jedes Tool veranschaulicht.

	Open Talend	Smooks	Jasper-soft ETL	KETL	Apatar
Plattform - Windows	✓	✓	✓	✓	✓
Plattform - Linux	✓	✓	✓	✓	✓
Format - CSV	✓	✓	✓	□	✓
Format - XML	✓	✓	✓	✓	✓
Format - Excel	✓	□	✓	□	✓
Format - Relationale DB	✓	✓	✓	✓	✓
Format - Hierarchische DB	✓	□	✓	□	□
Angemessenheit	✓	✓	✓	✓	✓
Tauglichkeit	✓	✓	✓	□	✓

Tabelle 1: Vergleich in Bezug auf die Tauglichkeit der Tools

Aus Tabelle 1 wird ersichtlich, dass zu dem Tool KETL zu drei Bereichen keine Informationen verfügbar sind und sich das Werkzeug daher wahrscheinlich nicht für den Auftraggeber eignen wird. Da aber kein Kriterium vollkommen ausgeschlossen werden kann, sondern nur die Informationen dazu fehlen, wird KETL trotzdem in der weiteren Grobauswahl untersucht.

3.4.2 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.

Open Talend

Das Tool Open Talend überzeugt durch einen breiten Umfang an unterschiedlichen Informationsangeboten. Neben den vom Unternehmen selbst zur Verfügung gestellten Benutzer-

handbüchern, Quellcodedokumentationen, Tutorials und Webinaren besteht eine große, aktive Community. Der Problem- und Wissensaustausch wird dabei primär auf einer von der Firma Talend zur Verfügung gestellten Internetplattform durchgeführt.²¹ Es finden sich allerdings auch zahlreiche Drittanbieter, beziehungsweise Nutzer, welche entsprechende Tutorials oder Videoanleitungen auf ihrer Internetseite kostenlos anbieten.

Smooks

Für das Framework besteht allein durch den Anbieter eine große Fülle an Dokumentations- und Informationsmaterial. Zu jeder Version werden ein umfangreiches Benutzerhandbuch, eine Java-Dokumentation und vorgefertigte Beispiele zum Download angeboten.²² Dieses Angebot wird durch ein eigenes Wiki und die Möglichkeit von Entwickler-Trainings vervollständigt.²³ Auch auf externen Seiten findet sich Informationsmaterial.²⁴ Als Möglichkeit für den Nutzeraustausch bei Problemen und Fragen wird in der Regel das JBoss Forum verwendet, eine eigene Plattform konnte nicht gefunden werden.²⁵

Jaspersoft ETL

Für Informationen über Jaspersoft ETL wird stets auf die Dokumentation des Open Talends verwiesen. Nach mehreren Recherchen findet sich im Internet eine Dokumentation über die Komponenten des Tools. Diese ist allerdings von einer sehr alten Version und daher schon veraltet. Insgesamt ist das Informationsangebot speziell zu Jaspersoft ETL sehr begrenzt beziehungsweise gar nicht vorhanden.

KETL

Die Dokumentationsauswahl für KETL ist sehr gering. Lediglich auf der Produkthomepage finden sich Dokumentationsunterlagen. Hierbei ist die Auswahl jedoch auf ein Installationshandbuch, einen Administration Guide und eine Präsentation mit Basistipps beschränkt. Ein Vermerk im Installationshandbuch, welcher besagt, dass es sich um ein Handbuch für UNIX oder Linux handelt und auf eine separate Anleitung für Windows verweist, führt ins Leere.²⁶

Apatar

Für das Tool Apatar werden zahlreiche Hilfestellungen geboten, wie beispielsweise Administrations- oder User-Guides, sowie ein Quick-Start-Guide, welcher auch als Web Demo verfügbar ist und schrittweise durch den Installations- und Einrichtungsprozess führt. Zudem

²¹ Vgl. Talendforge (2014)

²² Vgl. Smooks (2014a)

²³ Vgl. Redpill Linpro (2014) (2014); Vgl. dazu auch Smooks (2014b)

²⁴ Vgl. Redhat (o.J.)

²⁵ Vgl. JBoss (2008)

²⁶ Vgl. KETL (2014b)

wird ein großes Angebot an Beispielpunkten bereitgestellt, sodass die Einarbeitung immens erleichtert wird.²⁷

Bewertungsübersicht

Die Tabelle 2 liefert einen Vergleich der Tools im Hinblick auf die Dokumentation und die Informationsvielfalt, wobei Jaspersoft ETL am schlechtesten beurteilt wird und Apatar sowie Open Talend die Höchstpunktzahl erlangt haben.

Dokumentation und Informationsvielfalt	Open Talend	Smooks	Jaspersoft ETL	KETL	Apatar
Bewertung	5	4	0	1	5

Tabelle 2: Vergleich der Tools im Hinblick auf die Dokumentation und Informationsvielfalt

3.4.3 Grobkriterium 2 – Funktionalitätsumfang

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

Open Talend

Das Tool Open Talend ist mehr als eine einzelne Anwendung, es umschreibt vielmehr eine komplette Plattform. Beim Aufbau des Softwaresystems wurde dazu ein modularer Ansatz verfolgt, sodass je nach eigenen Bedürfnissen eine Erweiterung um andere Funktionsbereiche möglich ist. Jeder Funktionsbereich, wie beispielsweise „Big Data“ oder der hier relevanten „Data Integration“, kann zunächst als eigenständiges Softwareprodukt gesehen werden und umfasst die entsprechenden Funktionalitäten des ausgewählten Bereichs. Bei einer Erweiterung um eine andere Komponente, wird diese in die Plattform integriert und ermöglicht vordefinierte Schnittstellen und abgestimmte Prozesse zwischen den installierten Teilprodukten.²⁸

Für den zu untersuchenden „Data Integration“-Baustein steht eine immense Anzahl unterschiedlichster Schnittstellen bereit. Durch diese hohe Interoperabilität lässt sich das Produkt auch ohne größere Probleme in ein bestehendes IT-Netzwerk einbinden und harmonisiert mit notwendigen Drittsystemen – wie einer MySQL oder DB2 Datenbank.²⁹ Unterschiedlichste Dateiformate – sowohl in der Eingabe, als auch in der Ausgabe – sind, neben der Möglich-

²⁷ Vgl. Apatar Forge (2014a)

²⁸ Vgl. Talend (2014b)

²⁹ Vgl. Talend (2014c)

keit eigene Anpassungen vorzunehmen, kein Problem, sodass auch hier eine hohe Flexibilität gegeben ist.

Smooks

Die unterschiedlichen Funktionalitäten, welche vom Framework bereitgestellt werden, sind die „Transformation“, das „Routing“ und die Möglichkeit der „Persistenz“. Durch eine entsprechende Kombination der drei genannten Funktionalitäten wird es dem Nutzer möglich, Smooks als ETL-Tool einzusetzen.³⁰ Damit kann die Voraussetzung erfüllt werden. Durch den Einsatz einer entsprechenden Entwicklungsumgebung, welche das Smooks Framework einbinden kann, ist auch eine Kombination mit weiteren Frameworks – wie beispielsweise das von Apache angebotene Camel Framework – möglich und bietet einen entsprechenden Mehrwert.

Jaspersoft ETL

Bei diesem Kriterium ist Jaspersoft ETL sehr gut aufgestellt. Es beinhaltet alle Funktionen von Open Talend. Allerdings sind neue Funktionen zuerst in Open Talend integriert und etwas zeitversetzt im Jaspersoft ETL-Tool. Daher wird Jaspersoft ETL bei diesem Kriterium ein Punkt im Vergleich zu Open Talend abgezogen.

KETL

Im Hinblick auf den Funktionalitätsumfang ist die Informationsdichte zu gering, als dass dazu konkrete Angaben gemacht werden können. Auch auf Grund der Nutzung des Tools als eine Konsolenanwendung, konnten keine Funktionalitäten ermittelt werden, welche den eigentlichen ETL-Prozess übersteigen.

Apatar

Die Hauptfunktionalität bei Apatar ist der ETL-Prozess, dennoch bietet Apatar Funktionalitäten, wie Data Warehousing und Business Intelligence.³¹ Auch durch die Unterstützung mannigfaltiger Datenquellen, Applikationen, Dateiformaten und Protokollen³² hebt sich Apatar gegenüber alternativen ETL-Tools ab.

Bewertungsübersicht

Die Beurteilung aller Tools für das Grobkriterium des Funktionalitätsumfangs ist in der Tabelle 3 verdeutlicht.

Funktionalitäts- umfang	Open Talend	Smooks	Jaspersoft ETL	KETL	Apatar
----------------------------	----------------	--------	-------------------	------	--------

³⁰ Vgl. Smooks (2014c)

³¹ Vgl. Apatar (2014b)

³² Vgl. Apatar (2014c)

Bewertung	5	4	4	1	4
------------------	---	---	---	---	---

Tabelle 3: Vergleich der Tools im Hinblick auf den Funktionalitätsumfang

3.4.4 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-Projekte und dient im Rahmen dieser Arbeit als Informationsquelle für das Grobkriterium ‚Downloadzahlen‘.

Open Talend

2011: 23.259 2012: 36.724 2013: 12.578 2014: 6.229³³

→ Sehr hohe Downloadzahlen zu Beginn mit weiterer Steigerung, in den letzten Jahren ist dennoch ein Rückgang zu verzeichnen

Smooks

2011: 7.738 2012: 6.054 2013: 5.261 2014: 1.974³⁴

→ Mittelmäßige Downloadzahlen, die mit den Jahren deutlich rückläufig waren

Jaspersoft ETL

2011: 22.390 2012: 14.882 2013: 2.789 2014: 999³⁵

→ Sehr hohe Downloadzahlen zu Beginn, starker Rückgang in den letzten Jahren

KETL

2011: 1.228 2012: 1.059 2013: 1.699 2014: 518³⁶

→ Schwankende Werte, verhältnismäßig aber sehr geringe Downloadzahlen

Apatar

2011: 4.711 2012: 3.960 2013: 3.276 2014: 1.342³⁷

→ Downloadzahlen zeigen einen leichten aber konstanten Rückgang bei vergleichsweise eher geringer Verbreitung

³³ Vgl. Sourceforge (2014a)

³⁴ Vgl. Sourceforge (2014b)

³⁵ Vgl. Sourceforge (2014c)

³⁶ Vgl. Sourceforge (2014d)

³⁷ Vgl. Sourceforge (2014e)

Bewertungsübersicht

Eine Übersicht über die Bewertung aller fünf Tools im Hinblick auf die Downloadzahlen liefert Tabelle 4.

Downloadzahlen	Open Talend	Smooks	Jaspersoft ETL	KETL	Apatar
Bewertung	5	3	2	1	3

Tabelle 4: Vergleich der Tools im Hinblick auf die Downloadzahlen

3.4.5 Grobkriterium 4 – Nennenswerte Partner

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

Open Talend

Der Hersteller Talend bietet auf seiner Homepage einen eigenen Partnerbereich an, auf welchem die unterschiedlichen Beratungs-, Technologie- und OEM-Partner aufgelistet werden.³⁸ Im Technologiebereich wird mit zahlreichen Marktführern auf deren jeweiligem Gebiet kooperiert und entsprechende Schnittstellen und Funktionalitäten angeboten. Exemplarisch können Unternehmen wie EnterpriseDB, Google, Intalio, Jaspersoft, MySQL oder auch Sage genannt werden.³⁹ Durch die enge Zusammenarbeit kann eine effiziente Eingliederung eingesetzter Softwarelösungen erreicht werden, die beispielsweise mit MySQL eine schnelle Möglichkeit der Datenbankanbindung für Open Talend bietet.

Smooks

Die Entstehung von Smooks ist der JBoss Gemeinde zu verdanken. Diese stellt noch heute den wichtigsten Partner dar, auf deren Grundlage und Austausch Smooks weiterentwickelt wird. Für den praxisbezogenen Einsatz wird dabei oftmals die Eclipse-Plattform verwendet, welche deshalb ebenfalls einen nennenswerten Partner darstellt. Hier kann das Framework in Kombination mit anderen Eclipse-Erweiterungen genutzt werden, um entsprechende Mehrwerte bei der Entwicklung zu bieten – wie beispielsweise durch ein integriertes Apache Camel Framework, das für die Kommunikation zwischen den einzelnen Netzwerkkomponenten beziehungsweise für die Verteilung der elektronischen Daten sorgt.⁴⁰

³⁸ Vgl. Talend (2014d)

³⁹ Vgl. Talend (2014e)

⁴⁰ Vgl. Demolsky, M. (2009), Vgl. dazu auch Smooks (2014d)

Jaspersoft ETL

Jaspersoft und somit auch Jaspersoft ETL hat eine Vielzahl von nennenswerten Partnern. Beispielsweise: sun, Talend, MySQL, redhat. und JBoss⁴¹. Die wohl entscheidenste Kooperation stellt die mit Talend dar, da das Ergebnis eine nahezu identische ETL-Anwendung darstellt.

KETL

Für das ETL-Tool KETL konnten keine nennenswerten Partner ermittelt werden, weshalb hier mit einer Null die schlechteste Bewertung vergeben wird.

Apatar

Ein nennenswerter Partner von Apatar ist „Salesforce.com“, ein Marktführer in den Bereichen CRM und Enterprise Cloud Computing.⁴² So bietet Apatar beispielsweise die Möglichkeit, über das ETL-Tool direkt auf die in Salesforce gespeicherten Daten zuzugreifen.

Bewertungsübersicht

In Tabelle 5 ist zu jedem Tool die Bewertung für die Kategorie ‚Nennenswerte Partner‘ aufgeführt.

Nennenswerte Partner	Open Talend	Smooks	Jaspersoft ETL	KETL	Apatar
Bewertung	5	3	5	0	4

Tabelle 5: Vergleich der Tools im Hinblick auf nennenswerte Partner

3.4.6 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 in wie weit der Anwender Hilfestellung durch das Tool erhält.

Open Talend

Die Installation von Open Talend ist sehr einfach und kann durch Entpacken von der hierfür angebotenen Zip-Datei oder mittels eines Installationsassistenten – .exe Datei – durchgeführt werden. Im praxisbezogenen Einsatz bietet das Tool eine optisch ansprechende grafische Benutzeroberfläche. Durch die angebotenen Tutorials und Beispielprojekte wird der ergonomische Aufbau sehr gut erklärt, der sich in unterschiedliche Sichtbereiche unterteilt

⁴¹ Vgl. Jaspersoft (2014c)

⁴² Vgl. Salesforce (2014)

und eine übersichtliche Navigationsleiste beinhaltet. Mit Hilfe von integrierten Assistenten bei der Funktionswahl und einem unterstützenden Fehlerassistenten lassen sich ausgelassene Schritte oder bestehende Fehler erkennen, sodass dem Benutzer stets eine Hilfestellung durch das Tool gegeben wird.

Smooks

Das Framework Smooks besitzt keine eigene Systemplattform, sondern wird in eine integrierte Entwicklungsumgebung (IDE) gepackt. Als Standard wird hier die Eclipse IDE verwendet, für welche das Framework optimiert wurde.⁴³ Leider besteht keine grafische Oberfläche, sodass Nutzer hier auch gleichzeitig eine hohe IT-Affinität besitzen sollten. Innerhalb von Eclipse wird aber durch einen bestehenden Editor für Smooks die anfallende Arbeit deutlich erleichtert. Diese liegt in der gewünschten Konfiguration für die Umwandlung, was bedeutet, dass hier festgelegt werden muss, welche Quelle gewählt werden soll, welches Zielformat gewünscht wird, wie die Transformation durchzuführen ist und wohin „geroutet“ werden muss. Auch die Möglichkeit einer Anbindung von bestehenden Datenbanken lässt sich relativ einfach bewerkstelligen.⁴⁴

Jaspersoft ETL

Mit einer grafischen Benutzeroberfläche, die nahezu identisch mit der des Open Talends ist, und einem übersichtlichen Aufbau ist Jaspersoft ETL überaus benutzerfreundlich.

KETL

Das als Konsolenanwendung betriebene Tool KETL erfordert deutlich höhere Vorkenntnisse, als dies bei Tools mit einer grafischen Oberfläche der Fall ist. Daher fällt die Bewertung der allgemeinen Benutzerfreundlichkeit niedrig aus, denn auch eine intuitive Bedienung wird durch das reine Codieren größtenteils verhindert.

Apatar

Aufgrund seiner grafischen Oberfläche und einer übersichtlichen Aufteilung, lässt sich Apatar bereits nach kurzer Orientierungsphase flüssig und intuitiv bedienen. Durch die Tatsache, dass für jeden Arbeitsschritt, wie beispielsweise beim Hinterlegen einer CSV-Datei, eine schrittweise Menüführung erfolgt, wird der Einarbeitungsprozess zusätzlich erleichtert.

Bewertungsübersicht

Die Tabelle 6 hält die Ergebnisse aus der Beurteilung der Tools in Bezug auf die Ergonomie und Benutzerfreundlichkeit fest, wobei sogar drei Werkzeuge – Open Talend, Jaspersoft ETL und Apatar – die Höchstpunktzahl erreichen konnten.

⁴³ Vgl. Fennelly, T. (2010); Vgl. dazu auch Haneke, U./u. a. (2010)

⁴⁴ Vgl. Smooks (2014b); Vgl. dazu auch Straube, C. (2011)

Ergonomie/ Benutzer- freundlichkeit	Open Talend	Smooks	Jaspersoft ETL	KETL	Apatar
Bewertung	5	2	5	1	5

Tabelle 6: Vergleich der Tools im Hinblick auf die Ergonomie und Benutzerfreundlichkeit

3.4.7 Abschließende Beurteilung aller Grobkriterien

Die Tabelle 7 weist die aufsummierten Beurteilungen aller Grobkriterien für jedes Tool aus. Eine Tabelle, in welcher noch einmal die Bewertung der ETL-Werkzeuge für alle Grobkriterien aufgelistet wird, befindet sich in Anhang 1.

	Open Talend	Smooks	Jaspersoft ETL	KETL	Apatar
Bewertung	25	16	16	4	21

Tabelle 7: Abschließende Beurteilung aller Grobkriterien

Am schlechtesten schneidet das Tool KETL ab, da es sich zum einen um eine Konsolenanwendung handelt und zum anderen kaum verwertbare Informationen für dieses Tool bereitgestellt werden. Somit kann dieses Werkzeug bereits nach kurzer Einarbeitungs- und Orientierungsphase für einen Einsatz beim Auftraggeber ausgeschlossen werden.

Im Mittelfeld liegen Smooks, welches nicht besonders benutzerfreundlich ist, und Jaspersoft ETL, zu dem leider keinerlei eigenständige Dokumentationen verfügbar sind.

Die beiden Sieger aus der Grobauswahl sind Apatar und Open Talend, wobei Open Talend in allen Bereichen die Höchstpunktzahl erreichen konnte.

3.5 Feinauswahl der ETL-Tools

Resultierend aus der Grobauswahl der ETL-Tools wurden die Favoriten-Tools – Apatar und Open Talend – in die Feinauswahl übernommen. In dieser werden sie nach weiteren Kriterien untersucht. Die Erläuterung des Feinkriterienkatalogs erfolgt im nachfolgenden Gliederungspunkt.

3.5.1 Erläuterung des Feinkriterienkatalogs

Der Feinkriterienkatalog wurde in Zusammenarbeit beider Studentengruppen erstellt. Ausschlaggebend für diese Zusammenarbeit war, dass ein einheitlicher Kriterienkatalog einen höheren Mehrwert für das Unternehmen liefert, da eine bessere Vergleichbarkeit der Tools gegeben ist. Durch die zunächst selbstständige Erarbeitung eines Katalogs beider Gruppen und der anschließenden Zusammenführung, konnten Synergieeffekte erzielt werden, welche sich positiv auf die Qualität des Kriterienkatalogs auswirken.

Die Kriterien wurden nach zwei unterschiedlichen Prinzipien gewählt. Zum einen nach der DIN EN ISO 9241, einem internationalen Standard zur „Ergonomie der Mensch-System-Interaktion“. Die Norm besteht aus mehreren Teilen und berücksichtigt dabei sowohl den Bereich Hardware-Ergonomie als auch den Bereich Software-Ergonomie.⁴⁵ Zum anderen beruhen die gewählten Kriterien auf Internet- und Literaturrecherchen zum Thema „Evaluation von Open-Source-Anwendungen“.

Der Feinkriterienkatalog unterteilt sich in zwei Oberkategorien: das Produktprofil, welches mit einer blauen Farbe hervorgehoben ist, und das Anbieterprofil, das in Grün gehalten ist. Zu jedem Profil gehören wiederum einzelne Kategorien. Diese sind für das Produktprofil: ‚Ökosystem des Herstellers‘, ‚Funktionalität‘, ‚Leistung‘, ‚Bedienbarkeit/ Ergonomie/ Handhabung‘, ‚Performance/ Effizienz‘ sowie ‚Datensicherheit/ Zukunftssicherheit‘. Für das Anbieterprofil sind folgende Kriterien relevant: ‚Marktakzeptanz‘, ‚Aktivität der Community‘, ‚Dokumentation‘, ‚Dienstleistungsangebot‘ und ‚Finanzierbarkeit‘. Alle Kategorien unterteilen sich weiterhin in Kriterien. Ein Beispiel einer Kategorie des Produktprofils ist in Tabelle 8 dargestellt.

Oberkategorie	Kategorie	Gewichtung Kategorie	Kriterium	Gewichtung Kriterium	Produkt 1	Produkt 2
	Ökosystem des Herstellers	10%	Drittanbieter für Beratung und externe Dienstleitung	30%	0	5
			Partner zur Bereitstellung von Schnittstellen	30%	4	5
			Stakeholder/ Kunden	40%	2	4
			<i>Zwischenergebnis</i>	100%	2,0	4,6

Tabelle 8: Beispiel zur Erklärung des Feinkriterienkatalogs

Zudem wird für jede Kategorie und für jedes Kriterium eine Gewichtung angegeben. Die Summierung der Gewichtung der Kriterien ergibt pro Kategorie 100 Prozent. Ebenso beträgt die Summe der Gewichtung aller Kategorien für ein Oberkriterium auch 100 Prozent. Für die Produkte wird für jedes Kriterium eine Kennzahl von 0 bis 5 eingetragen, wobei 0 für sehr schlecht beziehungsweise nicht vorhanden und 5 für sehr gut steht. Anschließend werden die Kennzahlen mit der Gewichtung des Kriteriums multipliziert sowie für alle Kriterien aufsummiert und die verrechnete Bewertung als Zwischenergebnis ausgewiesen. Die Zwischenergebnisse für jede Kategorie werden für das Gesamtergebnis der Oberkategorie mit der Gewichtung der Kategorie verrechnet und für alle Kategorien, welche zur Oberkategorie

⁴⁵ Vgl. Technische Uni Berlin (2014)

gehören, aufsummiert und anschließend in einer separaten Zeile aufgeführt. Weiterhin werden in einer gesonderten Übersicht, siehe Tabelle 9, die Gesamtergebnisse der Oberkategorien nochmals ausgewiesen und miteinander verrechnet, wobei die Gewichtung 60 Prozent für das Produktprofil und 40 Prozent für das Anbieterprofil beträgt.

Ergebnisse in der Übersicht		Gewichtung des Profils	Produkt 1	Produkt 2
Produktprofil		60%	3,7	4,3
Anbieterprofil		40%	2,5	4,4
Gesamtergebnis des Kriterienkatalogs		100%	3,2	4,4

Tabelle 9: Beispiel zur Erklärung des Gesamtergebnisses

Ein ausgefüllter Feinkriterienkatalog in seiner Gesamtheit befindet sich in Anhang 2.

3.5.2 Produktprofil

Die ausgewählten Kriterien im ersten Teil des Feinkriterienkatalogs sollen helfen, das Produktprofil besser einzuschätzen und zu bewerten.

3.5.2.1 Kategorie 1 – Ökosystem des Herstellers

Nach Bosch besteht „ein Software Ökosystem aus einer 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.“⁴⁶

Kriterienbeschreibung

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 und teilweise auch im Wettbewerb mit dem Hersteller.⁴⁷

Diese Kategorie beschreibt also, in welcher Intensität alle Akteure zusammenarbeiten und in wie weit nennenswerte Partner in Kooperation mit dem Hersteller stehen.

Toolbewertung Open Talend

Das Ökosystem für die Plattform von Talend und ihren hierfür angebotenen Softwareprodukten ist sehr vielfältig. Viele namenhafte Hersteller von IT-Lösungen, die ihren Schwerpunkt auf unterschiedlichste IT-Produkte wie Datenbanken, Cloud-Lösungen oder IT-

⁴⁶ Bosch, J. (2009), S.111ff.

⁴⁷ Vgl. Slinger, J. (2011)

Servicedienstleistungen legen, arbeiten eng mit Talend zusammen.⁴⁸ Mit vielen dieser Hersteller wurden direkte Partnerschaften vereinbart, welche für alle Beteiligten und somit auch letztlich für den Kunden einen echten Mehrwert bieten sollen.⁴⁹

Zudem bietet Talend für Drittanbieter von Beratungs- und Schulungsdienstleistungen über Talend-Produkte die Möglichkeit für Partnerschaften. Damit können entsprechende Zertifizierungen und eine professionelle Aus- beziehungsweise Weiterbildung der Kooperationspartner ermöglicht werden, mit dem letztlichen Ziel, ihren Kunden ein breites Angebot an Ansprechpartnern für notwendige Unterstützungen zu bieten.⁵⁰

Aufgrund der hohen Nachfrage und der großen Akzeptanz für die Softwarelösungen von Talend, und damit auch für das zu untersuchende Teilgebiet der „Data Integration“-Lösung, besteht bereits heute ein sehr breites Netzwerk an beteiligten Firmen. Der Wettbewerb zwischen den Drittanbietern und dem Hersteller selbst lässt sich bisher jedoch primär auf die Beratungsleistung beschränken – was allerdings auch von Talend so gewollt ist. Der Hintergrund ist, dass die Produkte auch aktuell noch größtenteils selbst durch die zahlreichen Mitarbeiter entwickelt werden.

Toolbewertung Apatar

Für das ETL-Tool Apatar sind keine Drittanbieter verzeichnet, die eine Unternehmens- oder Kundenberatung anbieten. Im Hinblick auf externe Dienstleistungen, wie beispielsweise die Einrichtung des Tools innerhalb des Unternehmens, ist es ebenfalls so, dass keine konkreten Angebote dazu aufzufinden sind.

Auch im Aspekt Stakeholder, welcher die Kunden – und in gewissem Maße auch die freien Entwickler – umfasst, schneidet Apatar bei genauer Betrachtung der Foren schlecht ab. So fand zwar in der Anfangszeit des Tools ein reger Austausch zwischen festangestellten Entwicklern, freien Entwicklern und Anwendern statt, der jedoch mittlerweile stark zurückgegangen ist.⁵¹ Über eine weitere Zusammenarbeit zwischen Kunden und dem Unternehmen Apatar sind keine Informationen erhältlich.

Als Partner von Apatar, welche durch die Bereitstellung von Schnittstellen zur besseren Verbreitung oder zur breiteren Einsatzmöglichkeit des Tools beitragen, sind auf der Produkthomepage Salesforce und QuickBooks angegeben.⁵² Bei Salesforce handelt es sich um

⁴⁸ Vgl. Talend (2014d)

⁴⁹ Vgl. Talend (2014e)

⁵⁰ Vgl. Talend (2014o)

⁵¹ Vgl. Apatar Forge (2014b)

⁵² Vgl. Apatar (2014d)

einen Marktführer in den Bereichen CRM und Enterprise Cloud Computing⁵³ und bei QuickBooks um eine Bilanzierungssoftware.⁵⁴ Es besteht also die Möglichkeit, weitere Systeme an Apatar anzubinden oder Apatar an bestehende Systeme anzubinden und so einen weiteren und breiter aufgestellten Mehrwert beim Einsatz des Tools zu erzielen.

Bewertungsübersicht

In der Tabelle 10 werden für die Kategorie ‚Ökosystem des Herstellers‘ die Punkte für die beiden Tools und das daraus ermittelte Zwischenergebnis ausgewiesen. Hierbei zeigt sich, dass Talend in dieser Kategorie eindeutig besser abschneidet. Der ausschlaggebende Grund ist, dass Apatar keinerlei Drittanbieter für Beratung und externe Dienstleistung besitzt.

Kategorie	Gewichtung	Kriterium	Gewichtung	Apatar	Talend
Ökosystem des Herstellers	10%	Drittanbieter für Beratung und externe Dienstleistung	30%	0	5
		Partner zur Bereitstellung von Schnittstellen	30%	4	5
		Stakeholder/ Kunden	40%	2	4
	<i>Zwischenergebnis</i>		100%	2,0	4,6

Tabelle 10: Beurteilung im Hinblick auf die Ökonomie des Herstellers

3.5.2.2 Kategorie 2 – 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.

Kriterienbeschreibung

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 mit Hilfe einer Programmiersprache wie beispielsweise JAVA.

Die ‚Richtigkeit‘ des Tools beschreibt die Korrektheit der 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.

⁵³ Vgl. Salesforce (2014)

⁵⁴ Vgl. QuickBooks (2014)

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.⁵⁵

Toolbewertung Open Talend

Die Software Open Talend bietet ein vielseitiges Spektrum an Funktionalitäten für die regelbasierte Transformation der Daten. Der Anwender wird durch den Prozess der Transformation Schritt für Schritt geführt. Durch dieses schrittweise Vorgehen bietet Open Talend unterschiedliche Komponenten für die Transformation von Daten. Einerseits ist es möglich, die Daten einer Typkonvertierung zu unterziehen, andererseits können einzelne Werte verändert werden, wie beispielsweise das Hochzählen eines Indexes. Die grafische Aufbereitung und die genaue Vorgehensweise bei der Transformation werden beim Prototyp von Bedeutung sein, siehe Kapitel 4.4.

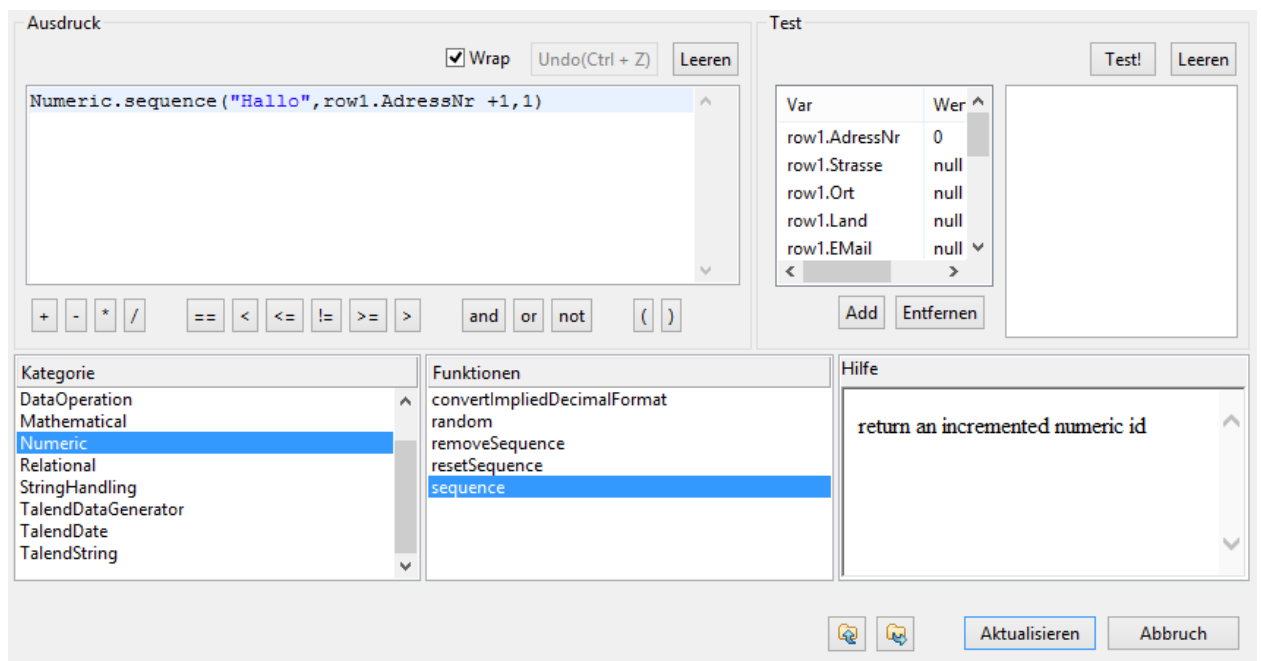


Abbildung 4: Ausdruckseditor des Open Talends

Die Abbildung 4 zeigt den Ausdruckseditor, in dem verschiedene Werte transformiert und „gemappt“ werden können. Im Bereich „Kategorie“, unten links in Abbildung 4, werden dem Anwender bereits viele verschiedene Funktionalitäten zur Verfügung gestellt. Beispielhaft kann die Kategorie „Numeric“ mit der Funktion „sequence“ gewählt werden. Jeder neu eingelesene Wert wird in diesem Fall um jeweils 1 hochgezählt.

Das nächste Kriterium in Bezug auf die Funktionalität ist die ‚Richtigkeit‘ des Tools. Anhand verschiedener Testläufe und unterschiedlicher Eingabeoperationen konnte jeweils die Kor-

⁵⁵ Vgl. Springer Gabler Verlag (2014a)

rektheit für die zu prüfenden Daten bestätigt werden. Diese Tests können repräsentativ für die verschiedenen Bereiche verwendet werden. Open Talend bietet dabei schon vorab die Unterstützung mit Warnungen und Hilfetexten, um die Korrektheit der Daten zu gewährleisten und unnötige falsche Transaktionen zu vermeiden. Bei Open Talend handelt es sich um einen grafisch aufbereiteten Integrationsprozess, der ausführenden Java-Code erzeugt.

In Bezug auf die ‚Interoperabilität‘ der Software ist zu sagen, dass Open Talend für verschiedene Betriebssysteme erhältlich ist. Neben dem breiten Einsatzgebiet sind bei der Interaktion mit kommerziellen Datenbanken und anderen Produkten keine Probleme bekannt.⁵⁶ Das Werkzeug bietet für fast jede Datenbank – zum Beispiel MySQL, DB2, MS SQL, PostgreSQL und Access – bestehende Komponenten für Lese- und Schreibzugriffe an. Neben Datenbanken können auch einige Dateien – zum Beispiel Excel-, JSON-, XML- und CSV-Dateien – als In- und Output verwendet werden.

Das bereits genannte Kriterium der ‚Modularität‘ kann bei Open Studio von Talend als sehr ausgereift bewertet werden. Grund dafür liegt im Aufbau als Plattform, welche durch verschiedene Bereiche ergänzt werden kann, ohne dafür einen enormen Abstimmungsaufwand durchführen zu müssen. Hier wird schlichtweg das gewünschte „Add-on“ installiert und gliedert sich in die Gesamtfunktionalität ein.

Toolbewertung Apatar

Die regelbasierte Transformation erfolgt bei Apatar über eine grafische Oberfläche, welche in der Kategorie 4 – ‚Bedienbarkeit/ Ergonomie/ Handhabung‘ – genauer beschrieben wird. Es besteht die Möglichkeit, ETL-Projekte abzuspeichern, wie beispielsweise die Zusammenführung eines CSV-Dokuments mit einem XML-Dokument. Diese Funktionalität kann genutzt werden, sofern der Vorgang mit identischen Dateien und Transformationsregeln wiederholt werden soll. Ebenso können Vorlagen erstellt werden, die als Grundlage für gängig Transformationen, wie die Selektion bestimmter Spalten aus einer relationalen Datenbank, verwendet werden können.

Die Richtigkeit des Tools, in Form der Korrektheit der gelieferten Ergebnisse, war bei den durchgeführten Tests stets gegeben. Jedoch beschränkten sich diese Tests auf kleinere Transformationen, wie beispielsweise die Umwandlung eines Datumformats – von TT.MM.JJJJ nach JJJJ-MM-TT – oder der Zusammenführung von zwei Tabellen über eine gemeinsame Identifikationsnummer. Für komplexere oder mehrschichtige Transformationen kann daher keine Aussage getroffen werden.

⁵⁶ Vgl. Rossak, I. (2013), S. 25 ff.

Inwiefern eine Interoperabilität mit anderen Systemen, wie beispielsweise „Salesforce“, über eine vordefinierte Schnittstelle möglich ist, konnte im Rahmen dieser Arbeit nicht getestet werden, weshalb ein Punkt von der Wertung abgezogen wird. Die Bewertung dieses Kriteriums bezieht sich daher ausschließlich auf die Interoperabilität mit verschiedenen Quell- und Zieldaten. Dies war bei allen getesteten Datenarten – CSV, XML, relationale Datenbanken und Excel – problemlos möglich. Sowohl bei der separaten Verarbeitung dieser Daten, als auch bei der Verknüpfung der verschiedenen Datenarten.

Bezüglich der Modularität kann zu Apatar nichts gesagt werden, da es nur eine Version des Tools gibt und somit eine individuelle Zusammenstellung der einzelnen Komponenten durch den Anwender nicht möglich ist.

Bewertungsübersicht

In der Tabelle 11 wird die Beurteilung der beiden Tools im Hinblick auf die Funktionalität dargestellt. Das ausgewiesene Zwischenergebnis zeigt, dass Apatar aufgrund der schlechten Bewertung im Kriterium ‚Modularität‘ hinter Open Talend liegt.

Kategorie	Gewichtung	Kriterium	Gewichtung	Apatar	Talend
Funktionalität	20%	Regelbasierte Transformation der Daten	25%	5	5
		Richtigkeit	25%	5	5
		Interoperabilität	25%	4	3
		Modularität	25%	0	5
<i>Zwischenergebnis</i>			100%	3,5	4,5

Tabelle 11: Beurteilung im Hinblick auf die Funktionalität

3.5.2.3 Kategorie 3 – 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.

Kriterienbeschreibung

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.⁵⁷

⁵⁷ Vgl. Springer Gabler Verlag (2014b)

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 der 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.⁵⁸

Toolbewertung Open Talend

Die Leistung einer Software ist eines der ausschlaggebendsten Kriterien für die Wahl eines neuen Softwareprodukts.

Open Talend ist derzeit in der Version 5 verfügbar. Durch mehrjährige Erfahrung und eine große Entwickler-Community kann ein stabiles Produkt mit geringer Fehlerwahrscheinlichkeit angeboten werden, woraus eine hohe Zuverlässigkeit resultiert.

Im Umfeld von Open-Source-Integrationssoftware gehört Open Talend zu den weltweit führenden Anbietern. Es verfügt über eine hohe Zufriedenheit bei den Anwendern, da die Verfügbarkeit 24/7 gewährleistet ist. Bei der eigenen Arbeit mit der Software konnte eine ständige Verfügbarkeit festgestellt werden.

Open Talend ist eine zuverlässige und gut skalierbare Anwendung und kann sich an unterschiedliche Umgebungen beziehungsweise Situationen anpassen. Dabei gestattet Talend, welches aus einer modularen Architektur besteht, eine unkomplizierte und skalierbare Erweiterung.⁵⁹

Toolbewertung Apatar

Die Kriterien ‚Zuverlässigkeit‘ und ‚Verfügbarkeit‘ können für die Beurteilung gemeinsam beschrieben werden. Soweit nach Durchführung der Tests hierzu eine Aussage getroffen werden kann, wurden beide Kriterien zu jeder Zeit sehr gut erfüllt. Es traten keine unerwarteten Fehler auf, es gab keine Programmabstürze, ein Zugriff auf hinterlegte Daten konnte stets problemlos erfolgen und Datenverlust wurde durch Warnmeldungen des Programms vermieden, die vor dem Beenden auf die Notwendigkeit der Speicherung aufmerksam machten. Optimaler wäre hierbei lediglich die Möglichkeit, eine automatische Speicherung zu definieren, welche die geöffneten Projekte in regelmäßigen Abständen sichert.

⁵⁸ Vgl. fremdwort.de (2014)

⁵⁹ Vgl. Talend (2014f)

Um eine Aussage über die Skalierbarkeit treffen zu können, wurden drei Testläufe bei gleicher Systemeinstellung durchgeführt. Dabei wurden aus CSV-Dateien vorgegebene Werte gelesen und in eine Tabelle geschrieben. Dies erfolgte für drei unterschiedlich große CSV-Dateien, welche einen Umfang von 10.000 bis 20.000 Zeilen aufwiesen. Jedoch muss dabei beachtet werden, dass es bei Durchführung dieses Tests auf einem anderen Rechner, zu starken Abweichungen der Testergebnisse kommen kann. Eine aussagekräftigere Einschätzung der Skalierbarkeit ist jedoch im Rahmen dieser Arbeit nicht durchführbar. Die Testergebnisse, welche den Durchschnitt von jeweils fünf Durchläufen repräsentieren, sind in der Tabelle 12 dargestellt.

Datenzeilen	Gesamtdauer	Verhältnis von Datenzeilen und Dauer
10.000	6,22 Sekunde	1.608/s
15.000	9,83 Sekunden	1.526/s
20.000	11,61 Sekunden	1.722/s

Tabelle 12: Ergebnis der Testläufe

Da das Verhältnis zwischen Datenzeilen und Verarbeitungsdauer bei diesen Tests ziemlich konstant war, wird die Skalierbarkeit als gut eingestuft.

Bewertungsübersicht

Die Tabelle 13 zeigt die Beurteilung der beiden Tools im Hinblick auf die Leistung. Die Werte liegen sehr eng beieinander, daher kann man lediglich sagen, dass Open Talend eine leichte Favoritenrolle in dieser Kategorie einnimmt.

Kategorie	Gewichtung	Kriterium	Gewichtung	Apatar	Talend
Leistung	20%	Zuverlässigkeit	40%	4	5
		Verfügbarkeit	30%	4	4
		Skalierbarkeit	30%	4	4
<i>Zwischenergebnis</i>			100%	4,0	4,4

Tabelle 13: Beurteilung im Hinblick auf die Leistung

3.5.2.4 Kategorie 4 – Bedienbarkeit/ Ergonomie/ Handhabung

Die Kategorie ‚Bedienbarkeit/ Ergonomie/ Handhabung‘ bezieht sich auf die Schnittstelle zwischen ETL-Tool und Anwender und lässt sich durch die Kriterien ‚Verständlichkeit/ Erlernbarkeit‘, ‚Grafische Benutzeroberfläche‘, ‚Fehlertoleranz‘ und ‚Komplexität‘ beschreiben. Dies sind wichtige Bestandteile, um die Zukunftssicherheit der Anwendung gewährleisten zu können und die Kundenzufriedenheit hochzuhalten.

Kriterienbeschreibung

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 Dateieindung oder ähnlichem, sollte das System entweder den Fehler aufzeigen oder ihn – sofern möglich – 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.

Toolbewertung Open Talend

Zum Kriterium ‚Verständlichkeit/ Erlernbarkeit‘ von Open Talend lässt sich sagen, dass der Nutzer schon nach wenigen Stunden mit dem Tool aktiv arbeiten und die ersten Transformationen durchführen kann. Durch viele Online-Tutorials und den kostenlosen Support über die Community auf „www.talendforge.org“ können die Funktionen von Talend schnell erlernt werden.

Open Talend bietet dem Nutzer eine übersichtliche und grafische Benutzeroberfläche. Intuitiv können dort per „Drag-and-Drop“ verschiedene Komponenten dem Arbeitsbereich zugeteilt werden.

Neben einer einfachen Bedienung der Anwendung wird dem Nutzer ein Error-Handling zur Seite gestellt. Falsche Eingaben werden vom System frühzeitig erkannt und können durch Hilfestellungen schnell behoben werden.

Im Hinblick auf die Komplexität Talends kann gesagt werden, dass die Strukturierung und der Aufbau von Projekten übersichtlich gestaltet sind und mit einer Baumstruktur dargestellt werden. Diese Darstellungsform ist bei Projekten üblich und spart somit Einarbeitungszeit.

Toolbewertung Apatar

Apatar lässt sich durch seine strukturiert aufgebaute und grafische Anwendungsoberfläche bereits nach einer kurzen Orientierungsphase weitestgehend problemlos bedienen. Bei Unsicherheiten stehen dem Anwender zahlreiche Hilfsmittel wie Web-Demos oder User-Guides zur Verfügung. Als größte Herausforderung stellte sich die erste Inbetriebnahme des Tools heraus, da eine Fehlermeldung erschien, mit dem Hinweis, dass Apatar bereits laufe und nicht mehrere Instanzen zur selben Zeit ausgeführt werden könnten. Nach kurzer Suche konnte im „Quick-Start-Guide“ eine Lösung dazu gefunden werden – Installation bei Vista/Windows 7 NICHT im Default-Pfad durchführen.⁶⁰

Damit schneidet Apatar sowohl bei der Verständlichkeit/ Erlernbarkeit, als auch bei der grafischen Benutzeroberfläche sehr gut ab. Ein Screenshot der Anwendungsoberfläche ist in Anhang 3 zu finden.

Im Hinblick auf die Fehlertoleranz des Tools kann ebenfalls ein positives Fazit gezogen werden. So werden beispielsweise bei der Eingabe des Datentyps zahlreiche Typen vorgegeben, die dann ausgewählt werden können. Damit können Flüchtigkeitsfehler, wie Tippfehler, verhindert werden. Auch durch die automatische Übernahme von Spalten- oder Tag-Namen aus den Quelldateien werden derartige Fehler unterbunden. Sofern dennoch eine fehlerhafte Angabe erfolgt, wird der Anwender in einigen Fällen darauf aufmerksam gemacht.

Für das Kriterium ‚Administrationskomplexität‘ lässt sich sagen, dass hierbei eine Verwaltung von Daten und Projekten völlig unproblematisch über eine auf Windows beruhende Ordnerstruktur möglich ist. Informationen darüber, inwieweit eine Erteilung von Zugriffsrechten möglich ist, konnten nicht ermittelt werden.

Bewertungsübersicht

Die Tabelle 14 gibt Aufschluss darüber, wie die gerade beschriebenen Kriterien sich auf die Bewertung auswirken. In der Kategorie ‚Bedienbarkeit/ Ergonomie/ Handhabung‘ schließen sowohl Apatar, als auch Open Talent mit einem Sehr Gut ab und liegen beide knapp unter der Höchstpunktzahl. Da Apatar im Bezug auf die Fehlertoleranz minimal besser beurteilt wird, hat es ein etwas höheres Zwischenergebnis.

Kategorie	Gewichtung	Kriterium	Gewichtung	Apatar	Talend
Bedienbarkeit/ Ergonomie/ Handhabung	20%	Verständlichkeit und Erlernbarkeit	50%	5	5
		Grafische Benutzeroberfläche	10%	5	5
		Fehlertoleranz	30%	5	4
		Komplexität der Administration	10%	4	4
<i>Zwischenergebnis</i>			100%	4,9	4,6

⁶⁰ Vgl. Apatar (2014b)

Tabelle 14: Beurteilung im Hinblick auf die Bedienbarkeit/ Ergonomie/ Handhabung

3.5.2.5 Kategorie 5 – 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 geantwortet wird.

Kriterienbeschreibung

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 und wie schnell Dateien generiert werden. Dabei ist der Datendurchsatz eine Messgröße für die Leistungsfähigkeit eines Übertragungssystems.⁶¹ 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.

Toolbewertung Open Talend

Die Performance für einfache Transformationen wird von der Open-Source-Variante abgedeckt. Die Geschwindigkeit und der Datendurchsatz können anhand des erstellten Prototyps gemessen werden. Auf die genauen Performancewerte wird in Kapitel 4.4 näher eingegangen. Die Performancetests wurden auf einem lokalen Rechner mit begrenzter Leistung durchgeführt. Diese Werte werden sich bei leistungsstärkeren Rechner und Servern in großen Unternehmen deutlich verbessern.

Sind komplexe Transformationen von riesigen Datenmengen notwendig, so bietet sich die kostenpflichtige „Big Data Edition“ an. Diese Erweiterung kann komplexere und aufwändigere Aufgaben bewältigen.

⁶¹ Vgl. It-Wissen (2014)

Der Speicherplatz und die CPU-Leistung, die von dem ETL-Tool in Anspruch genommen wird, sind minimal und können auf jedem Standardrechner ausgeführt werden. Eine entsprechende Erweiterung der Hardwareressourcen ist daher zunächst nicht notwendig.

Toolbewertung Apatar

Eine manuelle und individuelle Anpassung an die Hardware kann über eine Konfigurierungs-XML erfolgen. Allerdings bietet diese Datei dazu lediglich in geringem Maße Möglichkeiten.

Insgesamt lässt sich zur Performance sagen, dass – selbst auf den mittlerweile etwas veralteten Testgeräten – unverzüglich auf Benutzereingaben reagiert wird und auch bei größeren Datenmengen ein robustes Toolverhalten vorliegt.

Konkrete Werte hinsichtlich des Ressourcenverbrauchs unter Ausführung des Tools sind nur schwer zu treffen. Jedoch werden vom Anbieter einige Rahmenbedingungen vorgegeben, welche heutzutage durch jeden handelsüblichen Rechner abgedeckt werden: 100 MB Speicherplatz, 512 MB Arbeitsspeicher und einen 1GHz Prozessor.⁶²

Bewertungsübersicht

Insgesamt fällt die Bewertung, wie in Tabelle 15 dargestellt, aus. Wiederum liegen beide Tools sehr eng beieinander. Der einzige Unterschied liegt in dem Kriterium ‚Ressourcenverbrauch‘, bei dem Apatar mit einem Punkt besser abschneidet wie Open Talend. Somit fällt das Zwischenergebnis für Apatar in dieser Kategorie etwas höher aus.

Kategorie	Gewichtung	Kriterium	Gewichtung	Apatar	Talend
Performance/ Effizienz	15%	Individuelle Anpassung an die jeweilige Hardware	10%	3	3
		Geschwindigkeit und Datendurchsatz	30%	4	4
		Ressourcenverbrauch	30%	5	4
		Reaktionszeit auf Benutzereingaben	30%	5	5
	<i>Zwischenergebnis</i>		100%	4,5	4,2

Tabelle 15: Beurteilung im Hinblick auf die Performance/ Effizienz

3.5.2.6 Kategorie 6 – 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‘.

⁶² Vgl. Khizhnyak, A. (2007)

Kriterienbeschreibung

In Bezug auf die Datensicherheit ist die Sicherung in Form von Backups ein großes Thema. Die Datensicherung dient zur Sicherstellung von vorhandenen Datenbeständen und schützt vor dem 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. Solche Fehler 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 und dadurch unerlaubte Zugriffe zu verhindern.⁶³

Für den Sicherheitsaspekt ‚Zukunftssicherheit‘ sind insbesondere Releases von Bedeutung, da diese zeigen, dass eine permanente 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, welche 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 Problemen zuständig. Releases sind unter anderem auch wichtig für Sicherheitsaktualisierungen und Wartungsaktualisierungen sowie für die Behebung schwerwiegender Fehler.⁶⁴

Toolbewertung Open Talend

Die Datensicherheit wird von Talend insoweit erhöht, als dass Fehler und Hinweise über einen „Bug Tracker“ gemeldet werden können. Durch dieses Verfahren kann die Richtigkeit der Daten durch die Nutzer selbst sichergestellt werden.⁶⁵

Talend bietet die Möglichkeit, mit Hilfe eines Backups die Daten dauerhaft und konsistent zu sichern. Bei Verlust der vorhandenen Datenbestände können diese über die Backup-Funktion wieder geladen werden.

Das komplette Talend-Projekt befindet sich im stetigen Wachstum. Aufgrund der kostenpflichtigen Enterprise Edition mit einer Vielzahl an Entwicklern ist zu erwarten, dass auch Open Talend in den nächsten Jahren weiterentwickelt und verbessert wird. Ein weiterer Aspekt ist, dass Talend eine gute Platzierung im Gartner Bericht hat. Dadurch ist zudem mit

⁶³ Vgl. Bundesamt für Sicherheit in der Informationstechnik (o.J.)

⁶⁴ Vgl. Schmidt, O. (o.J.)

⁶⁵ Vgl. Rossak, I. (2013), S. 73

einer Steigerung des Interesses zu rechnen und so wird sich die Entwickler Community weiter vergrößern.⁶⁶

Toolbewertung Apatar

Apatar bietet keine integrierte Möglichkeit, die Daten oder Projekte explizit zu sichern. Da jedoch auf die Projekte auch außerhalb des Tools über den Windows Explorer zugegriffen werden kann, besteht über diesen Weg dennoch die Möglichkeit für den Anwender, für diese Daten ein Backup zu erstellen. Gleichermaßen gestaltet sich die Situation für die zugrundeliegenden Daten.

Die Zukunftssicherheit von Apatar schneidet schlecht ab, denn es ist deutlich zu sehen, dass die Aktivität von Seiten des Herstellers sehr niedrig bis kaum noch vorhanden ist. So ist beispielsweise der aktuellste Pressebericht, welcher auf der Herstellerseite zu lesen ist, bereits über drei Jahre alt.⁶⁷ Auch das aktuellste Release von Apatar, welches über „Sourceforge“ erhältlich ist, ist bereits über drei Jahre alt.⁶⁸

Bewertungsübersicht

Die Ergebnisse für die Kategorie ‚Daten-/ Zukunftssicherheit‘ befinden sich in Tabelle 16. Insgesamt zeigt sich, dass Open Talend hierbei deutlich besser abschneidet, vor allem auf Grund der häufigeren Releases.

Kategorie	Gewichtung	Kriterium	Gewichtung	Apatar	Talend
Daten-/ Zukunfts- sicherheit	15%	Backup	40%	3	3
		Manipulierbarkeit	30%	2	4
		Releases	30%	1	4
		<i>Zwischenergebnis</i>	100%	2,1	3,6

Tabelle 16: Beurteilung im Hinblick auf die Daten-/ Zukunftssicherheit

3.5.3 Anbieterprofil

In den folgenden Unterkapiteln 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.

3.5.3.1 Kategorie 1 – Marktakzeptanz

Die *Marktakzeptanz* der ausgewählten Open Source Software stellt ein zentrales Auswahlkriterium dar, welches den allgemeinen Produktstatus innerhalb des fokussierten ETL-

⁶⁶ Vgl. Rossak, I. (2013), S. 73

⁶⁷ Vgl. Apatar (2014e)

⁶⁸ Vgl. Sourceforge (2014f)

Softwarebereichs repräsentieren soll. Inhaltlich werden hierfür die Punkte ‚Google Trends‘, ‚Auszeichnungen/Zertifizierungen‘ und ‚Verbreitung (Entwickler, Anwender, Downloads)‘ untersucht und gewichtet.

Kriterienbeschreibung

Mit Hilfe von „Google Trends“ soll geprüft werden, inwiefern das Open-Source-ETL-Tool von den Internetnutzern nachgefragt wird – basierend auf historischen Statistiken entsprechender Suchanfragen. 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.⁶⁹

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

Im letzten Punkt dieser Kategorie wird das Kriterium ‚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 für den Austausch von Entwicklern, die den aktuellen Quellcode beziehen. Aufgrund dessen kann dieser Punkt als Repräsentation für den Fortbestand und die Weiterentwicklung des Produkts gesehen werden.

Toolbewertung Open Talend

Nach der Trendanalyse von Google sind die drei häufigsten Suchbegriffe für das ETL-Tool von Talend, welche jeweils mit einem Trendrating von x abschließen:⁷⁰

- Open Talend (78)
- Talend Open Studio (69)
- Talend ETL (95)

An den Werten lässt sich eine hohe relative Nachfrage nach den genannten Suchanfragen erkennen. Laut dem Graphen kann von einem allgemeinen Aufwärtstrend der Nachfrage

⁶⁹ Vgl. Google Trends (2014a)

⁷⁰ Vgl. Google Trends (2014b)

ausgegangen werden, sodass hier eine positive Rezession gegeben ist. Auffällig waren neben diesem Trendgewinn vor allem die Verteilungsregionen. Laut „Google Trends“ wird das Tool am stärksten in Frankreich nachgefragt, gefolgt von Indien, Deutschland und den USA. Für den deutschen IT-Bereich in größeren Unternehmungen sind dies wichtige Regionen, sodass auch hier eine erfreuliche Entwicklung vorliegt, welche sich in der Bewertung widerspiegelt.

Bereits kurze Zeit nach Gründung der Unternehmung erhielt Talend einige Auszeichnungen. Ende 2011 wurde der „Best of Open Source Software Award“ vom Fachmagazin InfoWorld bereits zum dritten Mal in Folge an Talend verliehen. Die Lösung Open Talend gewann hier zusätzlich in der Kategorie „Best Open Source Data Center and Cloud Software“. Grund dafür sei eine deutliche Effizienzsteigerung der Anwender bei der Datenintegration, eine schnelle Realisierung, reduzierte Wartungskosten und die Unterstützung für alle Arten von Datenintegration, -migration sowie -synchronisation.⁷¹

Auch in diesem und im letzten Jahr konnte Talend einige Awards gewinnen. Die zwei bekanntesten Preisverleihungen im Bereich der Datenintegration waren:

- 2014 DBTA 100
“Database Trends and Applications DBTA 100 is the list of the companies that matters most in data.”⁷²
- 2013 Gartner’s Magic Quadrant for Data Integration Tools
“[...] Talend was positioned as a Visionary in Gartner’s Magic Quadrant for Data Integration Tools.”⁷³

Im Bewertungspunkt ‚Verbreitung (Entwickler, Anwender, Downloads)‘ schneidet das Datenintegrationstool Open Studio, mit den bereits im Grobkriterium ‚Downloadzahlen‘ aufgeführten Werten (2012: 36.724, 2013: 12.578 und 2014 (Stand 24.06.): 6.629)⁷⁴ sehr gut ab, siehe Kapitel 3.4.4.

Diese Zahlen repräsentieren allerdings nur einen Teil der Entwicklung und Nutzung. Inzwischen hat Talend bereits knapp 400 eigene Entwickler angestellt, um seine Produkte zu optimieren und innovativ zu halten – Tendenz steigend. Allein zum Ende des Jahres 2014 sollen über 150 neue Entwickler angestellt werden.⁷⁵ Laut veröffentlichtem Pressebericht konnte, hier allerdings als eine allgemeine Aussage zur Plattform und nicht spezifisch für die Da-

⁷¹ Vgl. Talend (2011); Vgl. dazu auch Dineley, D./Borck, J./Wayner, P. (2011)

⁷² Vgl. Talend (2014g)

⁷³ Vgl. ebenda

⁷⁴ Vgl. Sourceforge (2014a)

⁷⁵ Vgl. Talend (2014h)

tenintegration, allein im vierten Quartal des Jahres 2013 ein Zuwachs der Anwenderzahl um 81 Prozent verzeichnet werden.⁷⁶

Toolbewertung Apatar

„Google Trend“ gibt Aufschluss darüber, dass die Nachfrage nach Apatar überwiegend aus dem indonesischen Raum stammt, gefolgt von den Vereinigten Staaten.⁷⁷ Der Trendanalyse Googles zufolge war das Interesse nach Apatar seit dessen Registrierung im Jahr 2007 stets stark schwankend, bis es Mitte 2011 seinen Höhepunkt erreichte. Seitdem ist jedoch ein deutlicher und stetiger Abwärtstrend erkennbar.⁷⁸

Diese Entwicklung spiegelt sich auch in den Downloadzahlen wieder, die bereits in der Grobauswahl betrachtet wurden, verwiesen sei auf Kapitel 3.4.4, die bis heute von durchschnittlich 600 Downloads pro Monat im Jahr 2010 bis auf durchschnittlich 220 Downloads pro Monat gesunken sind.⁷⁹ Dies zeigt, dass die Verbreitung des Tools zunächst einen guten Start hatte, jedoch bereits nach kürzerer Zeit von Konkurrenzprodukten überholt und verdrängt wurde.

Konkrete Angaben über Entwicklerzahlen sind aus Mangel an Informationen nicht möglich ebenso wenig, wie über genaue Anwenderzahlen. Jedoch kann auf der Webseite von Apatar eine Auflistung größerer Anwender eingesehen werden, zu welchen beispielsweise auch die „World Bank Group“ gehört.⁸⁰

Im Hinblick auf Zertifizierungen oder Auszeichnungen sind keine nennenswerten Awards seitens Apatar zu verzeichnen. Es gibt lediglich einige kleinere Awards, die 2008 von Apatar selbst an die Community vergeben wurden.⁸¹

Bewertungsübersicht

Die Tabelle 17 zeigt, dass Open Talend gegenüber Apatar vor allem im Kriterium ‚Auszeichnungen/Zertifizierungen‘ punkten kann. Insgesamt befindet sich Open Talend in der Kategorie ‚Marktakzeptanz‘ schon sehr nahe an der Höchstpunktzahl. Apatar hingegen hat ein vergleichsweise schlechtes Ergebnis erzielt.

⁷⁶ Vgl. Talend (2014i)

⁷⁷ Vgl. Google Trends (2014c)

⁷⁸ Vgl. ebenda

⁷⁹ Vgl. Sourceforge (2014e)

⁸⁰ Vgl. Apatar (2014f)

⁸¹ Vgl. Apatar (2014g)

Kategorie	Gewichtung	Kriterium	Gewichtung	Apatar	Talend
Markt- akzeptanz	25%	Google Trends	10%	3	4
		Auszeichnungen/Zertifizierungen	30%	0	5
		Verbreitung (Entwickler, Anwender, Downloads)	60%	3	4
	<i>Zwischenergebnis</i>	100%	2,1	4,3	

Tabelle 17: Beurteilung im Hinblick auf die Marktakzeptanz

3.5.3.2 Kategorie 2 – Aktivität der Community

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

Kriterienbeschreibung

Unter der ‚Forenaktivität‘ wird die Möglichkeit der Softwarenutzer verstanden, sich bei Problemen gegenseitig austauschen und einander 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 ziehen.

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

Als ‚FAQ‘ werden die bestehenden Antworten auf die gängigsten Fragen zum Produkt verstanden.⁸² 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.

Toolbewertung Open Talend

Für die Talend-Plattform und somit auch für das prüfende Datenintegrationstool bestehen mehrere Foren, Blogs und Videos. Das Hauptforum wird vom Unternehmen selbst geleitet. Mit etwa 180.000 registrierten Usern und über 130.000 Posts können anfallende Probleme oftmals im Kollektiv gelöst werden. Die Aktivität unterstreichen tägliche Posts und schnelle Antworten. Ein möglicher Nachteil für deutsche Nutzer, welcher allerdings damit auch eine

⁸² Vgl. Duden (2013)

viel breitere Masse an potenziellen Helfern anspricht, ist die stringente Verwendung der englischen Sprache innerhalb des Forums.⁸³

Wie bereits erwähnt wurde, hat Talend neben einer großen Community auch viele eigene Entwickler. Erfüllung offener Wünsche, wie individuelle Erweiterungen oder die Beseitigung gravierende technische Probleme, sind gegen einen gewissen Aufpreis möglich. Der Anbieter selbst bietet hierfür ein breites Support-Spektrum an, welches individuell nach den eigenen Bedürfnissen gewählt werden kann und multilingual zur Verfügung steht, siehe Abbildung 5.

	Silver Support	Gold Support	Platinum Support
Zugang zu zertifizierten Versionen	✓	✓	✓
Automatische Upgrades	✓	✓	✓
Bug Tracking	✓	✓	✓
Support Forum	✓	✓	✓
Garantierte Reaktionszeiten	✓	✓	✓
Garantierte Diagnose- und Antwortzeiten		✓	✓
Unbegrenzte Zugriffe		✓	✓
Web Support	✓	✓	✓
Email Support		✓	✓
Telefon Support			✓
24 Stunden Support		✓	✓

Abbildung 5: Talend Service Pakete⁸⁴

Die bestehenden FAQs des Anbieters werden in einer eigenen Hilfeseite unter der Rubrik „Talend Help Center“ geführt und sind sehr umfangreich. Diese sind sowohl für die unterschiedlichen Versionen, als auch in unterschiedlichen Sprachen verfügbar.⁸⁵ Auch extern, auf anderen Foren und Blogs, können FAQs zu den Produkten von Talend gefunden werden.⁸⁶

⁸³ Vgl. Talendforge (2014)

⁸⁴ Vgl. Talend (2014j)

⁸⁵ Vgl. Talend (2014k)

⁸⁶ Vgl. Developpez (2013)

Toolbewertung Apatar

Über die Entwicklerhomepage von Apatar – „ApatarForge.com“ – erhält man Zugriff auf ein Forum, welches 5.361 Artikel enthält und über 26.900 registrierte User verzeichnet. Jedoch sind zu manchen Themenbereichen, wie beispielsweise „Development“ oder „DataMaps“, die aktuellsten Beiträge bereits im Jahr 2012 verfasst worden. Zu anderen Themenbereichen hingegen, wie „Help“ oder „Installation and Upgrade“, sind Beiträge vorhanden, die weniger als eine Woche alt sind.⁸⁷ Ebenfalls auf der Entwicklerhomepage ist ein Reiter vorhanden, welcher auf Mailing-Listen verweist, die den Anwender zusätzlich unterstützen und auf dem Laufenden halten können. Allerdings führt dieser Verweis lediglich auf eine leere Seite und kann daher nicht genutzt werden.⁸⁸

Auf der eigentlichen Homepage von Apatar – „Apatar.com“ – erreicht man über den Reiter „Community“ eine Auflistung von FAQs, die nach einer kurzen Beantwortung der Frage teilweise direkt auf ausführlichere Artikel verweisen und so den Anwender bei der Antwortsuche gut unterstützen.⁸⁹

Apatar bietet zudem die Möglichkeit, direkten Kontakt über ein Kontaktformular aufzunehmen. Über Angabemöglichkeiten zum Problem- oder Interessenbereich des Anwenders kann eine Differenzierung der Anfragen vorgenommen und der Rückmeldungsprozess beschleunigt werden.⁹⁰

Bewertungsübersicht

In der Tabelle 18 werden die Bewertungen der beiden Tools in Hinblick auf die Aktivität der Community ausgewiesen. Hierbei schneidet Talend im Zwischenergebnis mit der Höchstpunktzahl ab und Apatar lediglich mit einem Befriedigend.

Kategorie	Gewichtung	Kriterium	Gewichtung	Apatar	Talend
Aktivität der Community	25%	Forenaktivität	40%	3	5
		Direkte Ansprechpartner	40%	3	5
		FAQ	20%	3	5
	Zwischenergebnis		100%	3,0	5,0

Tabelle 18: Beurteilung im Hinblick auf die Aktivität der Community

⁸⁷ Vgl. Apatar Forge (2014b)

⁸⁸ Vgl. Apatar Forge (2014c)

⁸⁹ Vgl. Apatar (2014h)

⁹⁰ Vgl. Apatar (2014i)

3.5.3.3 Kategorie 3 – Dokumentation

Innerhalb der Kategorie ‚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.

Kriterienbeschreibung

Über das Kriterium ‚Code Dokumentation‘ wird die Dokumentierung des Quellcodes untersucht. Im Rahmen dieser Arbeit kann allerdings keine tiefgreifende Untersuchung durchgeführt werden, deshalb wird sich auf die verfügbare Code-Dokumentation, deren Aufbaustruktur und die ersten Seiten als Leseprobe beschränkt, welche repräsentativ für die restliche Quellcodedokumentation zählt.

Bei der Überprüfung des ‚Benutzerhandbuchs‘ wird zunächst die generelle Verfügbarkeit dieser Art der Hilfestellung ermittelt. Besteht eine 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.

Toolbewertung Open Talend

Für die Open-Source-Software Open Talend steht innerhalb des bereits genannten „Help Centers“ eine umfangreiche Dokumentation zur Verfügung. Einerseits gibt es eine Quellcode-Dokumentation, andererseits ein Benutzerhandbuch für jedes einzelne Produkt, jede Version und in unterschiedlichen Sprachausgaben – Englisch, Deutsch und Chinesisch.⁹¹

Die Dokumentation selbst ist sehr ausführlich gehalten und durch eine Unterteilung in Kapitel, Unterkapitel sowie der Möglichkeit einer Stichwortsuche sinnvoll aufgebaut. Dem Nutzer wird eine gute Unterstützung und Einführung gewährleistet. Die ersten Schritte sind sowohl textuell, als auch durch Screenshots erläutert. Von der Installation bis hin zur generellen Nutzung sollten durch diese Hilfestellung keine größeren Probleme bestehen bleiben.

Toolbewertung Apatar

Auf der Entwicklerhomepage von Apatar werden dem Anwender unter dem Reiter „Documentation/ Wiki“ umfangreiche Informationen bereitgestellt. Es handelt sich dabei nicht um ein Benutzerhandbuch im klassischen Sinne, da es sich nicht um ein einzelnes Dokument handelt, welches alle Angaben beinhaltet. Stattdessen soll zunächst ein Themengebiet gewählt werden. Jedoch ist diese Vorgehensweise nicht von Nachteil. So wird der Nutzer davor geschützt, von einer Informationsflut überrollt zu werden. Mögliche Themengebiete sind un-

⁹¹ Vgl. Talend (2014)

ter anderem „Administration Guides“, „User Guides“, „Connector Guides“ oder auch „Web Demos“. Hinter den Guides verstecken sich weitere Unterthemen, die dem Anwender bildgestützt vermittelt werden, indem zahlreiche Screenshots zur Übermittlung der Informationen eingesetzt werden. Hinter den Web-Demos verbergen sich kleinere Videos, die den Anwender schrittweise durch den erfragten Prozess führen und ihn so bestmöglich unterstützen. Die Dokumentationsprache ist dabei durchgängig Englisch, was zum einen der gängige Standard für Open-Source-Dokumentationen ist und zum anderen auch aufgrund der starken grafischen Inhaltsvermittlung kein Hindernis darstellt, selbst wenn keine professionellen Sprachkenntnisse vorhanden sind.⁹²

Eine reine Code-Dokumentation ist für das Tool Apatar jedoch nicht erhältlich.

Bewertungsübersicht

Die Tabelle 19 liefert eine Übersicht über die Bewertung der beiden Tools im Hinblick auf die Dokumentation. Wegen der fehlenden Code-Dokumentation schneidet Apatar in dieser Kategorie schlechter ab als Talend.

Kategorie	Gewichtung	Kriterium	Gewichtung	Apatar	Talend
Dokumentation	20%	Code Dokumentation	40%	0	4
		Benutzerhandbuch	40%	5	5
		Sprache der Dokumentation	20%	4	4
	<i>Zwischenergebnis</i>		100%	2,8	4,4

Tabelle 19: Beurteilung im Hinblick auf die Dokumentation

3.5.3.4 Kategorie 4 – Dienstleistungsangebot

Die Kategorie ‚Dienstleistungsangebot‘ 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“)⁹³ verstanden werden.

Kriterienbeschreibung

Unter dem Punkt ‚Training‘ werden angebotene Kurse und Weiterbildungsmaßnahmen untersucht, welche die eigenen Mitarbeiter auf einen möglichst effizienten Umgang mit dem letztlich eingesetzten Softwareprodukt schulen. Für die ‚Beratung‘ werden die angebotenen technischen Unterstützungen geprüft. Diese beinhalten individuelle Erweiterungen, um neue

⁹² Vgl. Apatar Forge (2014d)

⁹³ Enzyklo (o.J.)

Funktionalitäten oder Anpassungen nach spezielle Gegebenheiten und Wünsche auszurichten.

Toolbewertung Open Talend

Die Unterstützung durch den Anbieter ist auch hier für Open Talend sehr gut, sowohl im Beratungsumfeld als auch im Angebot der Schulungs- beziehungsweise Trainingsmöglichkeiten. Für eine optimale Beratungsleistung stehen unterschiedliche Pakete zur Verfügung, welche entsprechende Dienstleistungen beinhalten, wie beispielsweise eine technische Hilfestellung, Optimierung, Anpassung oder auch Anwenderschulungen. Speziell im Bereich Training bietet Talend ein breites Spektrum an Schulungsmöglichkeiten. Es kann sowohl online, mittels Videokursen oder individuellen Tutorials und Unterlagen, als auch durch ausgebildetes Personal geschult werden.⁹⁴

Neben den eigenen Dienstleistungsangeboten bestehen inzwischen viele externe Möglichkeiten, welche den Mitarbeitern einen effizienten Umgang mit dem eingesetzten Tool lehren oder eine technische Unterstützung, wie beispielsweise eine funktionale Erweiterung, bieten. Beeindruckend ist hierbei die breite Masse an externen Anbietern und Dienstleistungen, sodass von einem intensiven Wettbewerb und einer anwenderbezogenen Optimierung des Schulungs- beziehungsweise Beratungsangebot auszugehen ist. Im Umkehrschluss können für die eigene Unternehmung auch hohe Kosten eingespart werden.

Toolbewertung Apatar

Die Auswahl des Trainings- und Beratungsangebots von Apatar ist eher gering. So konnten trotz ausgiebiger Suche auch keine offiziellen Drittanbieter ermittelt werden, die diesbezügliche Dienstleistungen anbieten. Allerdings besteht die Möglichkeit, sich bei Fragen über die Seite „Sourceforge“ direkt an Apatar-Entwickler zu wenden, oder sich dort in eine Mailing-Liste eintragen zu lassen.⁹⁵ Da dies aber im engeren Sinne nicht unter Drittanbieter fällt, wirkt es sich nicht auf die Bewertung des Kriteriums aus.

Von Seiten des Anbieters sind ebenfalls keine konkreten Angaben über Trainings für Apatar Data Integration vorhanden. Lediglich ein Vermerk, dass Trainings sowohl innerhalb als auch außerhalb des Unternehmens angeboten werden und diese von Experten für Apatar Data Integration und Beratern aus dem praktischen Umfeld durchgeführt werden.⁹⁶

Über Beratungs- und Supportangebot werden – im Hinblick auf die Kosten – genauere Informationen bereitgestellt. Dabei wird nach zwei Supportmodellen unterschieden. Zum einen der Stunden-basierte Support, bei dem eine bestimmte Anzahl an Supportstunden erworben

⁹⁴ Vgl. Talend (2014m)

⁹⁵ Vgl. Sourceforge (2014g)

⁹⁶ Vgl. Apatar (2014j)

und beliebig verwendet werden kann und zum anderen der Fall-basierte Support, welcher eine bestimmte Anzahl an Incidents beinhaltet, die ohne zeitliche Begrenzung bis zu ihrer Behebung unterstützt und bearbeitet werden. Die Preisspannen liegen dabei für den Stunden-basierten Support bei 495 - 1.495\$ und beim Fall-basierten Support zwischen 1.095 und 5.095\$.⁹⁷

Bewertungsübersicht

Auch in der Kategorie ‚Dienstleistungsangebot‘ ist Talend im Vergleich zu Apatar deutlich besser aufgestellt, wie aus Tabelle 20 ersichtlich wird. Talend kann sogar die Höchstnote in allen Kriterien und somit auch im Zwischenergebnis erzielen, wohingegen Apatar bei zwei Kriterien sogar mit der niedrigsten Punktzahl bewertet wird.

Kategorie	Gewichtung	Kriterium	Gewichtung	Apatar	Talend
Dienstleistungsangebot	15%	Training durch den Anbieter	30%	2	5
		Beratung durch den Anbieter	20%	4	5
		Training durch Drittanbieter	30%	0	5
		Beratung durch Drittanbieter	20%	0	5
	Zwischenergebnis		100%	1,4	5,0

Tabelle 20: Beurteilung im Hinblick auf das Dienstleistungsangebot

3.5.3.5 Kategorie 5 – Finanzierbarkeit

Das letzte Auswahlkriterium beinhaltet die allgemeine Finanzierbarkeit. Diese bezieht sich sowohl auf die Kosten für die Einführung des neuen ETL-Tools, als auch auf zu erwartenden Folgekosten im Unterhalt und den Mitarbeiterschulungen. Zentrale Punkte, welche hierfür untersucht werden, sind die ‚Lizenz‘, die angebotenen ‚Versionen‘ sowie die zu erwartenden Gesamtkosten als ‚Total Cost of Ownership‘ (TCO).

Kriterienbeschreibung

Der Punkt ‚Lizenz‘ beinhaltet die Bewertung der Lizenz, unter welcher die Open-Source-Software steht. Die Relevanz ergibt sich hier 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 geprüft, inwiefern trotz der Fokussierung auf den Open-Source-Bereich eine direkte Vertragsvereinbarung – in Verbindung mit den fixen Kosten – einzugehen ist, um den gewünschten Funktionsumfang auch tatsächlich zu erhalten.

⁹⁷ Vgl. Apatar (2014k)

Für eine Unternehmung sind letztlich primär die Gesamtkosten einer geplanten Tätigkeit im Verhältnis zu dessen erwartenden Mehrwert relevant. Durch eine Prognose der Gesamtkosten im Sinne der TCO kann ein erster Richtwert, welcher primär das Verhältnis der untersuchten Tools widerspiegelt, gegeben werden. Unter dem Konzept der TCO ist eine Nebenrechnung zur Vollkostenrechnung zu verstehen. Diese unterstützt die Beschaffung bei der kostenmäßigen Bewertung von Vergabealternativen. So können gewichtige Kostenfaktoren wie Logistikkosten, Kosten zur Absicherung von Währungsschwankungen, Kosten des Änderungsmanagements und Komplexitätskosten antizipativ bei Vergabeentscheidungen berücksichtigt werden.⁹⁸ Da eine fundierte TCO-Analyse im Rahmen dieser Arbeit nicht durchführbar ist, wird dieses Kriterium lediglich in der Kriterienbeschreibung aufgegriffen und nicht in die finale Bewertung der Kategorie ‚Finanzierbarkeit‘ übernommen.

Toolbewertung Open Talend

Das Tool Open Talend oder exakt Talend Open Studio for Data Integration ist ein kostenloses Open-Source-Entwicklungswerkzeug und steht unter der „GNU General Public License version 2.0“ (GPLv2) zur Verfügung.⁹⁹ Durch diese Lizenz fallen für den Einsatz innerhalb der eigenen Unternehmung keine direkten Kosten bei der Verwendung des Softwaresystems an. Eigene Erweiterungen sind durch den mitgelieferten Programmcode ohne Probleme möglich, müssen aber aufgrund der zugrundeliegenden Lizenz ebenfalls unter jener veröffentlicht werden. Zweiter Nachteil besteht in einem gekürzten Leistungsumfang für diese Programmausgabe.

Wie aus der nachfolgenden Abbildung 6 erkennbar ist, fehlen für die Standardversion unternehmenswichtige Komponenten, wie ein gemeinsames Repository oder die Möglichkeit von Management- und Monitoring-Tools. Auch der Support ist bei der Open-Source-Version nicht enthalten, was eine schnelle Reaktion auf Probleme allein von der Community oder entsprechenden internen Spezialisten abhängig werden lässt.¹⁰⁰

⁹⁸ Vgl. TCW (o.J.)

⁹⁹ Vgl. Sourceforge (2014a)

¹⁰⁰ Vgl. Talend (2014o)

Funktionen	Talend Open Studio for Data Integration	Talend Enterprise Data Integration	Talend Platform for Data Management
Job Designer	✓	✓	✓
Business Modeler	✓	✓	✓
Über 800 Konnektoren	✓	✓	✓
Versionierung	✓	✓	✓
Gemeinsames Repository		✓	✓
Assistenten		✓	✓
Management und Monitoring Tools		✓	✓
Datenqualität			✓
Clustering			✓
Haftungsausschluss/Garantie und Talend Support		✓	✓
Lizenz	Open Source	Subskription	Subskription

Abbildung 6: Leistungsumfang der angebotenen Talend Versionen¹⁰¹

Aufgrund der genannten Nachteile empfiehlt es sich für die Unternehmung eine der beiden umfangreicheren Versionen zu wählen, welche allerdings mit einer kostenpflichtigen Lizenz verbunden sind. Diese stehen unter der „Subskription“-Lizenz. Die anfallenden Kosten belaufen sich, laut Hersteller, auf 20-50% gegenüber denen eines proprietären Produkts und lassen eine viel flexiblere Anpassung an aktuelle Notwendigkeiten zu – damit sind auch zeitliche „Downgrades“, die parallele Nutzung unterschiedlicher Altversionen oder beispielsweise die Nutzung an einem Heim-PC möglich. Für beide kostenpflichtigen Versionen ist bereits ein entsprechender Support integriert und eine Gewährleistung durch den Hersteller gegeben.¹⁰²

Toolbewertung Apatar

Apatar verwendet ein ‚Dual Licensing‘-Geschäftsmodell. Nach diesem Modell können Benutzer bei der Entscheidung für ein Apatar Produkte wählen, ob sie es unter der freien Software/ Open-Source GNU GPL oder unter einer kommerziellen Lizenz erwerben und verwenden möchten. Vorteil der kommerziellen Lizenz ist, dass Apatar damit auch zu bestimmten Supportleistungen verpflichtet ist. Der Nachteil ist aber, dass da-

¹⁰¹ Vgl. Talend (2014n)

¹⁰² Vgl. ebenda

mit automatisch höhere Kosten auf den Nutzer zukommen.¹⁰³ Aufgrund des dualen Lizenzmodells kann jedes Unternehmen die Lizenzfrage entsprechend seiner individuellen Ansprüche beantworten.

Angebote über differenzierte Versionen, wie eine Prime-Version oder eine Standard-Version sind für Apatar nicht erhältlich.

Bewertungsübersicht

In Tabelle 21 findet sich eine Gesamtübersicht für die Kategorie ‚Finanzierbarkeit‘. Trotz unterschiedlicher Ergebnisse der beiden Tools in den Kriterien, sind die Zwischenergebnisse identisch.

Kategorie	Gewichtung	Kriterium	Gewichtung	Apatar	Talend
Finanzierbarkeit	15%	Lizenz	60%	5	3
		Version	40%	0	3
	<i>Zwischenergebnis</i>			100%	3,0

Tabelle 21: Beurteilung im Hinblick auf die Aktivität der Community

3.5.4 Ergebnis der Feinauswahl

In der Feinauswahl wurden die beiden Sieger der Grobauswahl, siehe Kapitel 3.4, miteinander verglichen. Das Ergebnis ist in Tabelle 22 veranschaulicht. Der gesamte ausgefüllte Kriterienkatalog befindet sich in Anhang 2.

	Gewichtung des Profils	Apatar	Talend
Produktprofil	60%	3,7	4,3
Anbieterprofil	40%	2,5	4,4
Gesamtergebnis des Kriterienkatalogs	100%	3,2	4,4

Tabelle 22: Entgeltiges Ergebnis der Feinauswahl

Es zeigt sich, dass Open Talend in beiden Oberkategorien ähnlich abschneidet und als Gesamtergebnis 4,4 von 5 möglichen Punkten erhält. Apatar hingegen erzielt vor allem im Anbieterprofil ein eher schlechtes Ergebnis und kommt insgesamt nur auf 3,2 der möglichen 5 Punkte. Somit ist eindeutig, dass das Tool Open Talend, welches auch schon in der Grobauswahl am besten abgeschnitten hat, als Sieger aus der Feinauswahl hervorgeht und für die prototypische Umsetzung herangezogen wird.

¹⁰³ Vgl. Apatar (2014l)

4 Prototypische Umsetzung mit Open Talend

Die prototypische Umsetzung ermöglicht die Bewertung des Open-Source-Tools Open Talend in der Praxis. Im Folgenden wird der Ablaufprozess von der Erstellung einer Datenbank bis zur Erzeugung eines Output Files prototypisch aufgezeigt. Die Daten werden dabei aus mehreren Datenbanktabellen gelesen und in eine zusammengefasste Tabelle transformiert. Im Folgenden wird ein Konzept für die Umsetzung des Prototyps erstellt und auf die einzelnen Schritte näher eingegangen. Hierbei ist zu erwähnen, dass für die Generierung der Datengrundlage ebenfalls eine Zusammenarbeit beider Projektgruppen erfolgte, um eine verbesserte Vergleichbarkeit zu ermöglichen.

4.1 Konzept zur Umsetzung der Transformation

Als Grundlage und Anhaltspunkt für die spätere prototypische Umsetzung wird ein Konzept zur Transformation benötigt. Hierbei soll eine Reduktion der Komplexität durch das Aufteilen des Gesamtprozesses der Transformation in einzelne, voneinander unabhängigen Komponenten erreicht werden. Das Konzept befasst sich mit dem kompletten Ablaufprozess von der Erstellung einer Datenbank, der Verbindung mit Open Talend und der Transformation in ein beliebiges Format. Hierbei werden die vielseitigen Möglichkeiten, die mit dem Tool umgesetzt werden können, aufgezeigt.

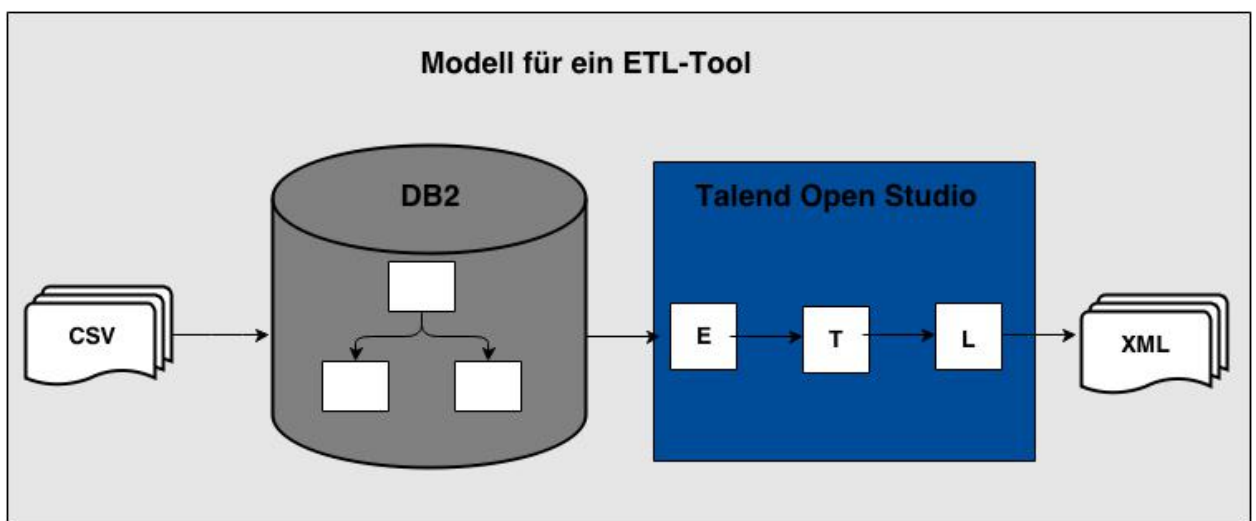


Abbildung 7: Konzept zur Umsetzung des Prototyps

Das grundsätzliche Konzept zur Erstellung des Prototyps ist in der Abbildung 7 dargestellt. Die Abbildung dient als Gesamtübersicht für die Transformation und es wird in den darauffolgenden Kapiteln 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 im Unternehmen simuliert werden kann. Der große Datenbestand ermöglicht das Testen des Tools Open Talend. Dabei können beliebige Dateien wie beispielsweise CSV oder XML erzeugt werden.
- Der zweite Schritt befasst sich mit der **Einrichtung der Datenbank-Verbindung**. Es stehen verschiedene Datenbank-Systeme zur Auswahl: relationale DB – zum Beispiel DB2 von IBM – und hierarchische DB – zum Beispiel DL1. In dieser Arbeit wird der Fokus nach Wunsch des Auftraggebers auf die relationalen Datenbanken gelegt. Anschließend müssen hier die automatisch generierten Tabellen und Daten aus Schritt eins eingebunden werden.
- Der dritte Schritt befasst sich mit der **Datenintegration** und der Erstellung eines beliebigen Ausgabeformats. In Abbildung 7 wird als Ausgabeformat XML erstellt. Nach dem ETL-Prozess kann jedes beliebige Dateiformat erzeugt werden.

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

4.2 Erstellung einer großen Datenbasis

Dieses Kapitel stützt sich auf den ersten Schritt der Abbildung 7. Die Struktur und der Inhalt der Daten sind von großer Bedeutung, um die verschiedenen Daten mit dem ETL-Tool zu transformieren. Auch ist es aufgrund der besseren Vergleichbarkeit entscheidend, dass die zweite Gruppe, welche das gleiche Thema bearbeitet, mit der identischen Datenbasis arbeitet, weshalb die Datenbasis in Zusammenarbeit erstellt wurde.

In dieser Arbeit wird der Fokus auf das Eingabeformat CSV gelegt. Dieses Format dient der Speicherung beziehungsweise dem Austausch von einfach strukturierten Daten. Das CSV-Format wird dabei oft von Unternehmen eingesetzt, um Daten zwischen unterschiedlichen Computersystemen auszutauschen. Ein Merkmal dieses Formats ist, dass die Werte durch Semikolons voneinander getrennt sind. Ein kleiner Ausschnitt aus der erzeugten CSV-Datei ist in Tabelle 23 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

Tabelle 23: 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 Tabelle 23 zeigt eine Tabelle mit unterschiedlichen Spalten und den dazugehörigen Werten. Die generierte CSV-Datei beinhaltet insgesamt 500 unterschiedliche Datensätze.

4.3 Datenbank-Verbindung einrichten

Um die Daten in einer Datenbank zu speichern und diese zugänglich zu machen, muss eine Datenbank erstellt werden. In dieser Arbeit wird dabei der Fokus auf die relationalen Datenbanken gelegt, die mit der Open-Source-Software MySQL realisiert wird. Dabei lassen sich mit MySQL-Daten erstellen, bearbeiten und verwalten.

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

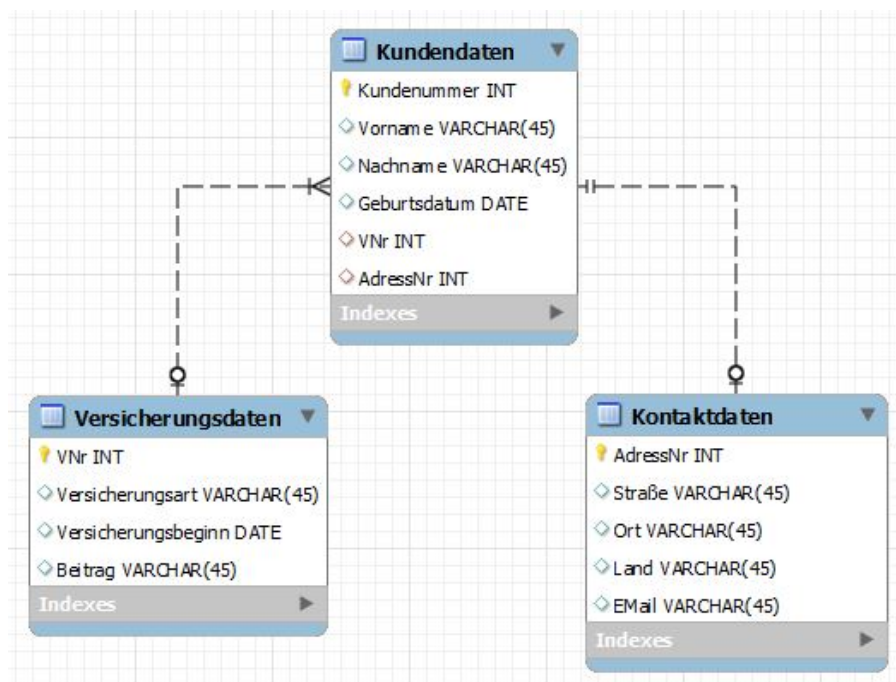


Abbildung 8: ER-Modellierung der unterschiedlichen Datenbanktabellen

Das in Abbildung 8 dargestellte 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 Tabelle 23 ist beispielhaft ein Ausschnitt aus der Tabelle „Kundendaten“ dargestellt. Anhand dieser Struktur wurde das ER-Modell erstellt. Anhand 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 sequentiell in die Tabelle „Kundendaten“ geschrieben. Jedem Attribut werden die Werte aus der CSV-Datei „KundenDB.csv“ zugewiesen. Insgesamt werden jeweils 500 Datensätze in die drei Tabellen geschrieben.

4.4 Integration der Daten

Nachdem die vorangegangenen Abschnitte einen ersten Einblick in die Erstellung einer Datenbank gegeben haben, wird nun die Integration der Daten mittels Open Talend näher beschrieben. Dieses Kapitel bezieht sich auf Schritt drei der Abbildung 7.

Um Daten aus der Datenbank zu beziehen oder in diese schreiben zu können, muss eine Verbindung zwischen der Anwendung Open Talend und der MySQL-Datenbank bestehen. Die Abbildung im Anhang 4 zeigt die grafische Oberfläche von Open Talend. Im linken Abschnitt der Anwendung kann die neue Datenbankverbindung über „Meta-Daten>Db Connections“ erstellt werden. Eine Verbindung zum MySQL-Server mit der Datenbank „IntegrationMySQL 0.1“ wurde beispielhaft hergestellt. Nachdem Open Talend mit der MySQL-Datenbank verbunden wurde, ist ein Zugriff auf die einzelnen Tabellen der Datenbank möglich.

Die eigentliche Erstellung für die Datenintegration erfolgt mit Hilfe eines „JobDesigners“ auf einer grafischen Oberfläche. Durch einen Job werden eine oder mehrere Komponenten miteinander verbunden. Die entwickelten Prozesse werden in Code übersetzt und zur Ausführung gebracht. Im Weiteren wird anhand des Jobs in Abbildung 9 der Prototyp näher be-

schrieben. Der Prototyp hat zum Ziel, die Daten aus den Relationen „kundendaten“ und „kontaktdaten“ in die Relation aus der Datenbank „kunden_all“ zurück zu schreiben.

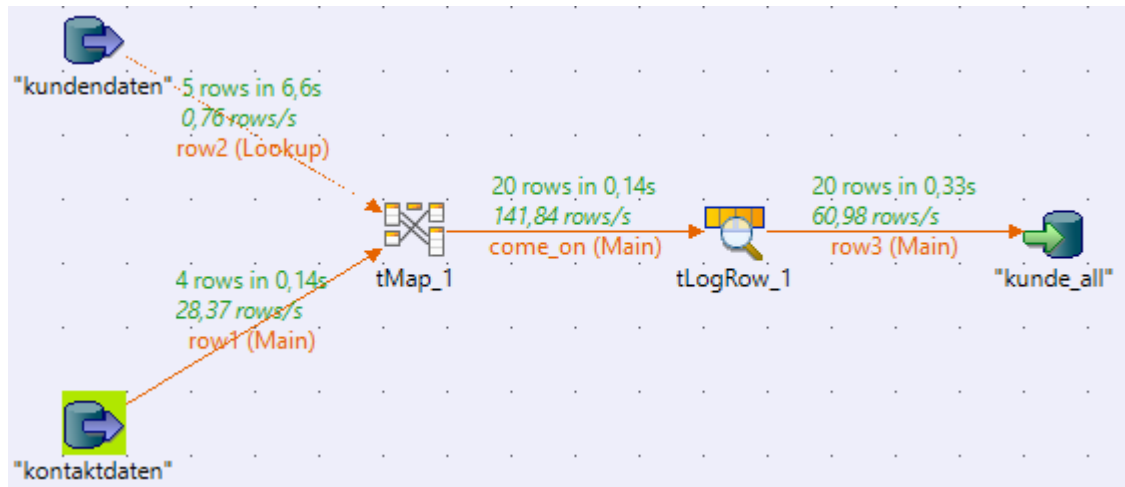


Abbildung 9: Job mit unterschiedlichen Komponenten

In Abbildung 9 ist ein Job mit unterschiedlichen Komponenten dargestellt. In diesem Abschnitt werden die wichtigsten Funktionsweisen der verwendeten Komponenten erläutert.

Open Talend bietet mit der Komponente „tMysqlInput“ die Möglichkeit zur Anbindung an die bestehenden Relationen aus der MySQL-Datenbank. Die Daten können somit aus dem Quellsystem bezogen werden. Die beiden Relationen „kundendaten“ und „kontaktdaten“ werden mit dieser Komponente angebunden. Neben dem Einlesen der Daten ist es mit der Komponente „tMap“ möglich, Input und Output Attribute aufeinander zu mappen und Transformationen vorzunehmen. Für den Prototyp sieht diese Transformation wie in Abbildung 10 nachfolgend dargestellt ist aus.

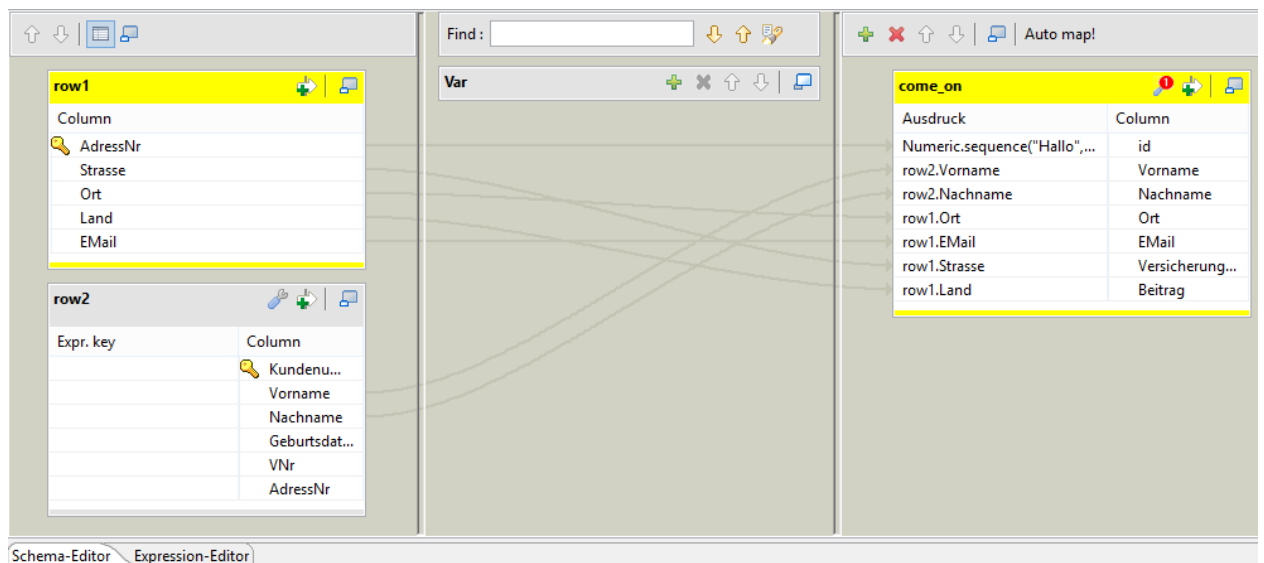


Abbildung 10: Transformation von Attributen

Der linke Bereich zeigt alle eingehenden Attribute der Relationen an. Diese stammen aus den zwei Tabellen „kundendaten“ und „kontaktdaten“. Der Bereich auf der rechten Seite stellt alle verfügbaren Attribute für die Ausgabe bereit. In der Spalte Ausdruck können Eingabespalten zugeordnet und Transformationen durchgeführt werden. Die zwei Bereiche sind mit Zuordnungspfeilen verbunden, die zeigen, welche Eingaben für welche Ausgaben genutzt werden. In der vorangegangenen Abbildung 10 wird die Tabelle „come_on“ mit den Attributen aus beiden Tabellen gefüllt und zu einer Tabelle zusammengefasst. Falls bei einem Attribut Transformationen durchzuführen sind, können diese im Ausgabestrom erstellt werden.

```
Numeric.sequence("Hallo",row1.AdressNr +1,1)
```

Das Code Beispiel zeigt die Transformation der id, die von der Tabelle AdressNr übergeben wird. Das Ziel ist es, den Index bei jedem Datensatz um den Wert '1' zu erhöhen. Auf diese Art und Weise können beliebig viele Transformationen durchgeführt werden.

Nachdem mit der „tMap“ Komponente die Attribute gemappt wurden, wird die Komponente „tLogRow“ für die Anzeige der bearbeiteten Datensätze in der Konsole benötigt. Im Anhang 4 ist im unteren Bereich die Konsolenansicht mit den durchgelaufenen Datensätzen zu sehen.

Als letzter Schritt müssen die transformierten Attribute mit der Komponente „tMySQLOutput“ in die Tabelle „kunde_all“ geschrieben werden. Die Datenbank-Relation „kunde_all“ wird nun mit den Werten aus den beiden Quellrelationen befüllt.

Dieser Transformationsschritt kann für beliebig viele Tabellen durchgeführt werden. Es können beispielsweise auch drei oder mehr Tabellen zu einer zusammengefasst werden. In diesem Beispiel wurden die veränderten Daten in die relationale Datenbank zurückgeschrieben. Es ist möglich, die Daten in andere Dateiformate zu schreiben, wie zum Beispiel CSV oder XML. Die Transformationen können die Komponenten der BI und Big Data enthalten. Je nach Anwendungsfall muss anhand dieser Komponenten der Job erstellt werden.

Nach Ablauf des Jobs werden die einzelnen Performancewerte direkt bei den Verbindungen zwischen den einzelnen Jobs erstellt und angezeigt. Für die einzelnen Einlese- und Schreibzugriffe können somit die Zeiten bewertet werden. In Abbildung 10 wurden beispielhaft fünf Datensätzen aus der Tabelle Kundendaten ausgelesen. Die Geschwindigkeit für das Auslesen beläuft sich auf 6,6 Sekunden und einem Datendurchsatz von 0,76 rows/second. Die Transformation und Komprimierung der zwei Tabellen in eine Tabelle beansprucht insgesamt 0,14 Sekunden mit einem Datendurchsatz von 141,84 rows/second. Die schnellen Durchlaufzeiten sind in diesem Prototypen auf die einfachen Transformationen der Attribute zurückzuführen. Bei komplexeren und aufwändigeren Transformationen werden die Perfor-

mance und der Datendurchsatz mehr beansprucht und die Durchlaufzeiten werden steigen, was jedoch durch leistungsstärkere Rechner ausgeglichen werden kann.

Open Talend stellt eine Vielzahl an Eingabe- und Ausgabeformate bereit, wie beispielsweise JSON, XML; CSV. Diese Komponenten können mit drag and drop in die Job-Erstellung hinzugefügt werden und mit Funktionalitäten versehen werden. Die Daten Integration kann mit unterschiedlichen Komponenten von Open Talend realisiert werden.

5 Fazit

Zuerst erfolgte in dieser Seminararbeit eine Marktanalyse, in welcher alle JAVA-basierten am Markt befindlichen Open-Source-ETL-Tools aufgelistet werden. Nachdem allerdings nicht alle Werkzeuge zur weiteren Untersuchung in Frage kamen und zwei Gruppen das gleiche Thema bearbeiteten, spezialisiert sich die Grobauswahl dieser Arbeit auf die Folgenden fünf Tools: Open Talend, Smooks, Jaspersoft ETL, KETL und Apatar. In der Grobauswahl wurden die ausgewählten Tools auf fünf Grobkriterien untersucht: ‚Dokumentation und Informationsvielfalt‘, ‚Funktionalitätsumfang‘, ‚Downloadzahlen‘, ‚Nennenswerte Partner‘ sowie ‚Ergonomie und Benutzerfreundlichkeit‘. Die Grobauswahl ergab, dass sich Open Talend und Apatar für eine nähere Untersuchung eignen.

Für die Feinauswahl wurde ein Kriterienkatalog erstellt und anhand dessen die beiden Tools detaillierter bewertet. Insgesamt schnitt Open Talend sowohl im Produktprofil, als auch im Anbieterprofil besser ab als Apatar. Der Unterschied war mit einer Differenz von 1,2 Bewertungseinheiten – 4,4 Bewertungseinheiten für Open Talend und 3,2 für Apatar – sehr deutlich zu sehen.

Für die Umsetzung des Prototyps wurde das ETL-Tool Open Talend verwendet, welches sowohl aus der Grob- als auch aus der Feinauswahl als bestes Tool hervorging: Als Fazit kann zusammengefasst werden, dass es ein starkes Tool für einfache wie auch komplexe Transformationen ist. Eine einfache Bedienbarkeit und hilfreiche Unterstützungen ermöglichen eine einfache Einarbeitung und eine Entwicklung verschiedener Aufgaben. Des Weiteren ist die Performance für die durchgeführten Transformationen und der Datendurchsatz sehr hoch.

Es lässt sich festhalten, dass sich zahlreiche Open-Source-ETL-Tools auf dem Markt befinden. Dabei ist allerdings nicht jedes für den Auftraggeber empfehlenswert. Nach tiefgehenden Analysen und dem praktischen Einsatz kann Open Talend als Favorit genannt werden. Allerdings muss der Auftraggeber vor der Einführung noch einmal genauere Untersuchungen

durchführen, da es aufgrund fehlender Ressourcen – zum Beispiel Rechnerleistung – nicht möglich war, das Umfeld des Krankenversicherers eins zu eins nachzustellen.

Anhang

Anhangverzeichnis

Anhang 1: Gesamtübersicht der Grobauswahl	56
Anhang 2: Ausgefüllter Feinkriterienkatalog	57
Anhang 3: Screenshot des Tools Apatar	60
Anhang 4: Screenshot der Konsole des Open Talends	60

Anhang 1: Gesamtübersicht der Grobauswahl

	Open Talend	Smooks	Jaspersoft ETL	KETL	Apatar
Dokumentation und Informationsvielfalt	5	4	0	1	5
Funktionalitätsumfang	5	4	4	1	4
Downloadzahlen	5	3	2	1	3
Nennenswerte Partner	5	3	5	0	4
Ergonomie/ Benutzerfreundlichkeit	5	2	5	1	5
Summierung	25	16	16	4	21

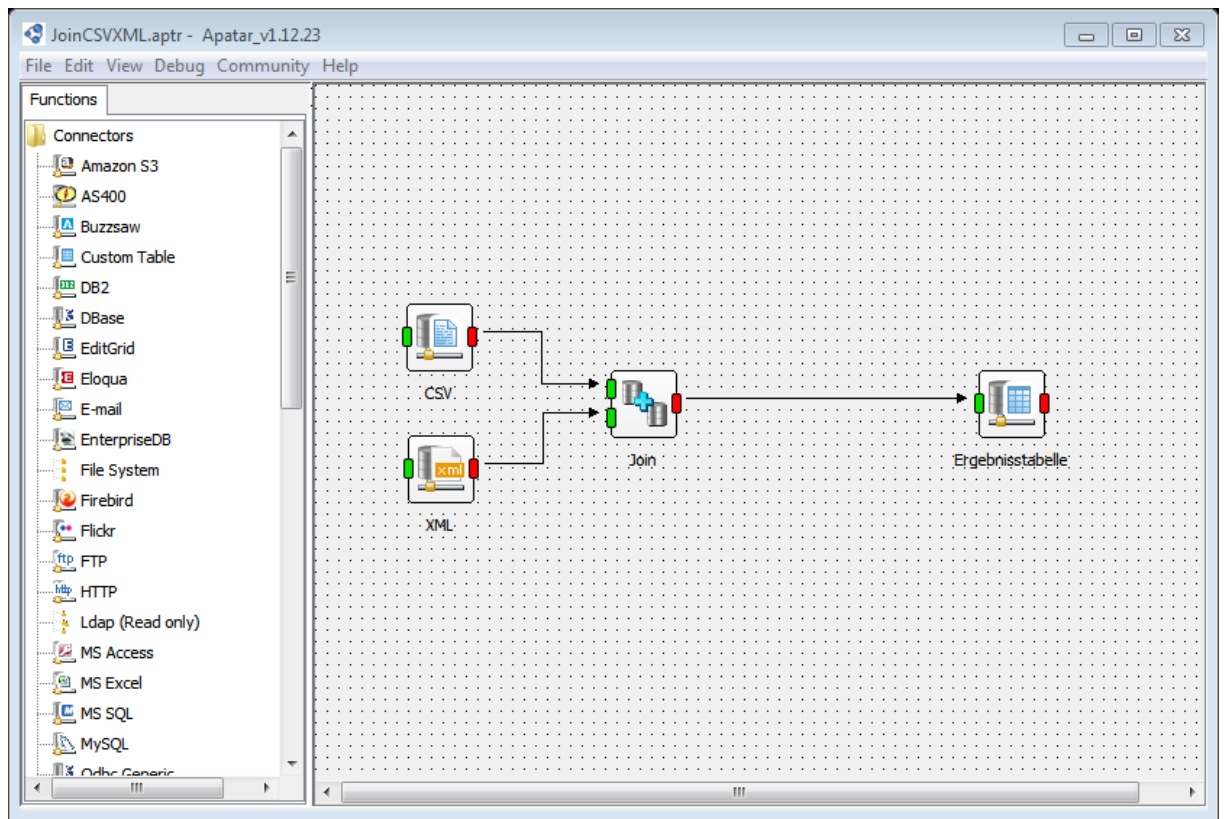
Anhang 2: Ausgefüllter Feinkriterienkatalog

Oberkategorie	Kategorie	Gewichtung	Kategorie	Kriterium	Gewichtung Kriterium	Apapar	Talend			
PRODUKTPROFIL	Ökosystem des Herstellers	10%		Drittanbieter für Beratung und externe Dienstleistung	30%	0	5			
				Partner zur Bereitstellung von Schnittstellen				30%	4	5
				Stakeholder/ Kunden				40%	2	4
		Zwischenergebnis				100%	2,0	4,6		
	Funktionalität	20%			Regelbasierte Transformation der Daten	25%	5	5		
					Richtigkeit	25%	5	5		
					Interoperabilität	25%	4	3		
					Modularität	25%	0	5		
		Zwischenergebnis				100%	3,5	4,5		
	Leistung	20%			Zuverlässigkeit	40%	4	5		
					Verfügbarkeit	30%	4	4		
					Skalierbarkeit	30%	4	4		
		Zwischenergebnis				100%	4,0	4,4		
	Bedienbarkeit/ Ergonomie/ Handhabung	20%			Verständlichkeit und Erlernbarkeit	50%	5	5		
					Graphische Benutzeroberfläche	10%	5	5		
					Fehlertoleranz	30%	5	4		
					Komplexität der Administration	10%	4	4		
		Zwischenergebnis				100%	4,9	4,6		
	Performance/ Effizienz	15%			Individuelle Anpassung an die jeweilige Hardware	10%	3	3		
					Geschwindigkeit und Datendurchsatz	30%	4	4		
Ressourcenverbrauch					30%	5	4			
Reaktionszeit auf Benutzereingaben					30%	5	5			
	Zwischenergebnis				100%	4,5	4,2			
Datensicherheit/ Zukunftssicherheit	15%			Backup	40%	3	3			
				Manipulierbarkeit	30%	2	4			
				Releases	30%	1	4			
	Zwischenergebnis				100%	2,1	3,6			
	Gesamtergebnis des Produktprofils					3,7	4,3			

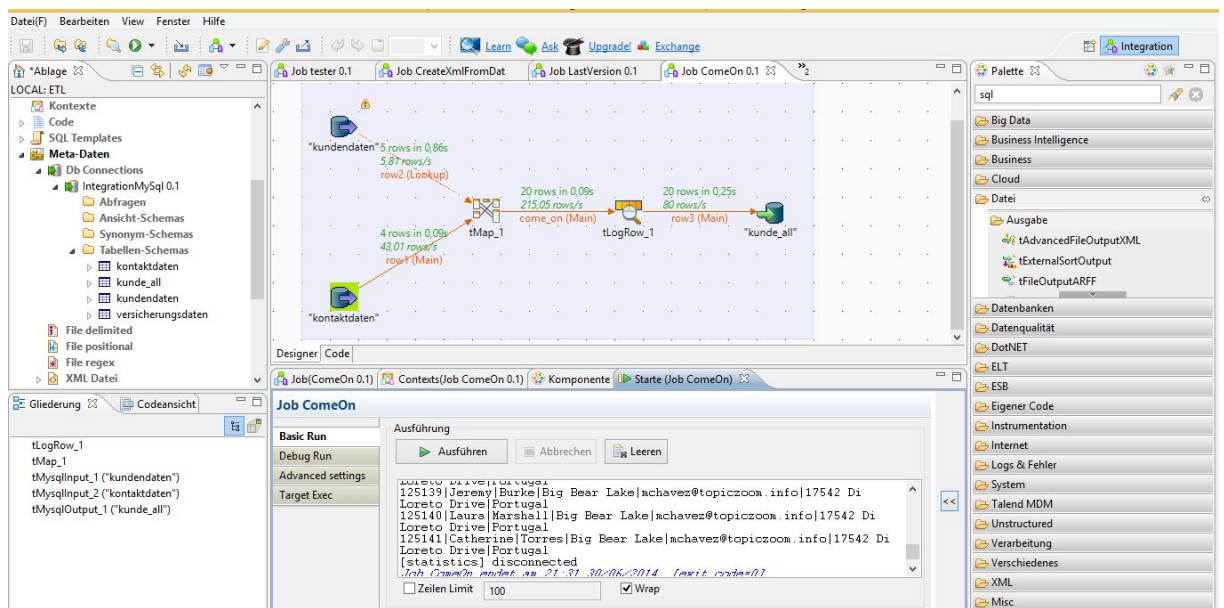
Oberkategorie	Kategorie	Gewichtung Kategorie	Kriterium	Gewichtung Kriterium	Apatar	Talend
ANBIETERPROFIL	Marktakzeptanz	25%	Google Trends	10%	3	4
			Auszeichnungen/ Zertifizierungen	30%	0	5
			Verbreitung (Entwickler, Anwender, Downloads)	60%	3	4
			Zwischenergebnis	100%	2,1	4,3
	Aktivität der Community	25%	Forenaktivität	40%	3	5
			Direkte Ansprechpartner	40%	3	5
			FAQ	20%	3	5
	Zwischenergebnis	100%	3,0	5,0		
	Dokumentation	20%	Code Dokumentation	40%	0	4
			Benutzerhandbuch	40%	5	5
			Sprache der Dokumentation	20%	4	4
	Zwischenergebnis	100%	2,8	4,4		
	Dienstleistungsangebot	15%	Training durch den Anbieter	30%	2	5
			Beratung durch den Anbieter	20%	4	5
Training durch Drittanbieter			30%	0	5	
Beratung durch Drittanbieter			20%	0	5	
Zwischenergebnis	100%	1,4	5,0			
Finanzierbarkeit	15%	Lizenz	60%	5	3	
		Version	40%	0	3	
Zwischenergebnis	100%	3,0	3,0			
Gesamtergebnis des Anbieterprofils		100%			2,5	4,4

Ergebnisse in der Übersicht		Gewichtung des Profils	Apatar	Talend
Produktprofil		60%	3,7	4,3
Anbieterprofil		40%	2,5	4,4
Gesamtergebnis des Kriterienkatalogs		100%	3,2	4,4

Anhang 3: Screenshot des Tools Apatar



Anhang 4: Screenshot der Konsole des Open Talends



Quellenverzeichnisse

Literaturverzeichnis

- Bosch, J. (2009):** From software product lines to software ecosystems, ProceedingSPLC09 of the 13th International Software Product Line ConferenceCarnegie Mellon University Pittsburgh, 2009. Jg., S. 111–119
- Haneke, U./u. a. (2010):** OPEN SOURCE BUSINESS INTELLIGENCE, 1. Aufl, München/Wien: Hanser
- Rossak, I. (2013):** Datenintegration, München: Hanser

Verzeichnis der Internet- und Intranet-Quellen

- Apatar (2014a):** Product,
<http://apatar.com/product.html>,
Abruf: 01.07.2014
- Apatar (2014b):** For Data Warehousing and Analytics,
http://www.apatar.com/for_data_warehousing_and_analytics.html,
Abruf: 28.06.2014
- Apatar (2014c):** Apatar Quick Start User Guide,
http://www.apatar.com/pdf/Apatar_Quick_Start_Guide.pdf,
Abruf: 28.06.2014
- Apatar (2014d):** On-demand synchronization of Salesforce accounts and QuickBooks customers,
http://apatar.com/sfdc_and_quickbooks_data_synchronization.html,
Abruf: 01.07.2014
- Apatar (2014e):** News/ Press Releases,
http://apatar.com/news_press_releases.html,
Abruf: 01.07.2014
- Apatar (2014f):** Customers,
<http://www.apatar.com/customers.html>,
Abruf: 28.06.2014
- Apatar (2014g):** Apatar Community Awards 2008,
http://www.apatar.com/aca_2008.html,
Abruf: 28.06.2014
- Apatar (2014h):** FAQ,
http://www.apatar.com/community_faq.html,
Abruf: 28.06.2014
- Apatar (2014i):** Contact Us,
http://www.apatar.com/contact_us.html,
Abruf: 28.06.2014
- Apatar (2014j):** Training,
<http://www.apatar.com/training.html>,
Abruf: 28.06.2014
- Apatar (2014k):** Support Packages,
http://apatar.com/buy_support_packages_online.html,
Abruf: 28.06.2014
- Apatar (2014l):** Licensing FAQ,
http://www.apatar.com/licensing_faq.html,
Abruf: 28.06.2014
- Apatar Forge (2014a):** Where do I start?,
<http://www.apatarforge.org/wiki/dashboard.action>,
Abruf: 27.06.2014

- Apatar Forge (2014b):** Forum,
<http://www.apatarforge.org/forums/>,
 Abruf: 28.06.2014
- Apatar Forge (2014c):** Mailing List,
<http://www.apatarforge.org/apatarforge-mailing-lists.html>,
 Abruf: 28.06.2014
- Apatar Forge (2014d):** Apatar Wiki,
<http://www.apatarforge.org/wiki/dashboard.action>,
 Abruf: 28.06.2014
- Black Duck (2014):** Eighth Annual Future of Open Source Survey Finds OSS Powering New Technologies, Reaching New People, and Creating New Economics,
<http://www.blackducksoftware.com/news/releases/2014-future-open-source-survey-results-revealed>,
 Abruf: 30.06.2014
- Bundesamt für Sicherheit in der Informationstechnik (o.J.):** Glossar und Begriffsdefinitionen,
https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/Glossar/glossar_node.html,
 Abruf: 01.07.2014
- Demolsky, M. (2009):** Open-Source-Framework Smooks: Transformer,
<http://jaxenter.de/artikel/Open-Source-Framework-Smooks-Transformers>,
 Abruf: 29.06.2014
- Developpez (2013):** La FAQ Talend,
<http://talend.developpez.com/faq/?page=1>,
 Abruf: 29.06.2014
- Dineley, D./Borck, J./Wayner, P. (2011):** Bossie Awards 2011: The best open source data center and cloud software,
<http://www.infoworld.com/d/open-source-software/bossie-awards-2011-the-best-open-source-data-center-and-cloud-software-171813-1¤t=4&last=1#slideshowTop>,
 Abruf: 28.06.2014
- 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
- Fennelly, T. (2010):** UN/EDIFACT Editor for Eclipse... Ecore/XMI Models etc,
<http://blog.smooks.org/2010/09/18/unedifact-editor-for-eclipse/>,
 Abruf: 29.06.2014
- fremdwort.de (2014):** Skalierbarkeit,
<http://www.fremdwort.de/suchen/bedeutung/Skalierbarkeit>,
 Abruf: 01.07.2014

- GNU/FSF (2014):** Warum Freie Software besser ist als Open-Source-Software, <https://www.gnu.org/philosophy/free-software-for-freedom.de.html>,
Abruf: 30.06.2014
- Google Trends (2014a):** Trends, <http://www.google.de/trends/>,
Abruf: 24.06.2014
- Google Trends (2014b):** Talend, <http://www.google.de/trends/explore?q=Talend#q=Talend&cmpt=q>,
Abruf: 28.06.2014
- Google Trends (2014c):** Apatar, <http://www.google.de/trends/explore#q=Apatar&cmpt=q>,
Abruf: 28.06.2014
- It-Wissen (2014):** Datendurchsatz data throughput, <http://www.itwissen.info/definition/lexikon/Durchsatz-throughput.html>,
Abruf: 01.07.2014
- Jaspersoft (2008):** JASPERSOFT UND TALEND ARBEITEN AN DATENINTEGRATIONSPACK ZUSAMMEN, DASS FÜR BI KUNDEN SPEZIFISCH ANGEPASST WIRD, <http://www.jaspersoft.com/de/press/jaspersoft-und-talend-arbeiten-datenintegrationspack-zusammen-dass-f%C3%BCr-bi-kunden-spezifisch-an>,
Abruf: 30.06.2014
- Jaspersoft (2014a):** EDITIONS, <https://www.jaspersoft.com/de/editions-de>,
Abruf: 30.06.2014
- Jaspersoft (2014b):** Jaspersoft ETL, <http://community.jaspersoft.com/>,
Abruf: 30.06.2014
- Jaspersoft (2014c):** PARTNERS, <http://www.jaspersoft.com/de/partners>,
Abruf: 30.06.2014
- JBoss (2008):** Smooks Dev Tools Reference Guide, http://docs.jboss.org/tools/OLD/3.0.1.GA/en/jboss_smooks_plugin_ref_guide/html_single/index.html,
Abruf: 28.06.2014
- KETL (2014a):** What is KETL? KETL Features, <http://www.ketl.org/products/overview.shtml>,
Abruf: 30.06.2014
- KETL (2014b):** Documentation, <http://www.ketl.org/documentation/overview.shtml>,
Abruf: 30.06.2014

- Khizhnyak, A. (2007):** Installation Guide for Apatar Data Integration,
http://www.apatar.com/pdf/Apatar_Installation.pdf,
 Abruf: 01.07.2014
- Manageability (2004):** Open Source ETL (Extraction, Transform, Load) Written in Java,
<http://www.manageability.org/blog/stuff/open-source-etl/view>,
 Abruf: 30.06.2014
- Open Source Initiative (2014a):** The Open Source Initiative,
<http://opensource.org/>,
 Abruf: 30.06.2014
- Open Source Initiative (2014b):** The Open Source Definition,
<http://opensource.org/osd>
 Abruf: 01.07.2014
- QuickBooks (2014):** Accounting Software,
<http://www.quickbooks.co.za/>,
 Abruf: 01.07.2014
- Redhat (o.J.):** What is Smooks?,
https://access.redhat.com/site/documentation/en-US/Red_Hat_JBoss_Developer_Studio/4.1/html/Smooks_Developer_Tools_Reference_Guide/ch01s02.html,
 Abruf: 28.06.2014
- Redpill Linpro (2014) (2014):** Redpill Linpro Training,
<https://training.redpill-linpro.com/category/java/course/smooks-java-integration-developers/>,
 Abruf: 27.06.2014
- Salesforce (2014):** Über Salesforce,
<http://www.salesforce.com/de/company/>,
 Abruf: 27.06.2014
- Schmidt, O. (o.J.):** Aktuelle Themen der Informatik - IT Infrastructure Library Release Management,
http://webuser.hs-furtwangen.de/~kasper/seminar0405/ReleaseManagement_ausarbeitung.pdf,
 Abruf: 32.06.2014
- Slinger, J. (2011):** SOFTWARE-ÖKOSYSTEME,
http://it-ra-dar.eu/serendipity/uploads/transkripte/Zusammenfassung_Interview_Jansen.pdf,
 Abruf: 01.07.2014
- Smooks (2014a):** User Documentation,
http://www.smooks.org/mediawiki/index.php?title=User_Documentation,
 Abruf: 29.09.2014

- Smooks (2014b):** V1.4: Smooks v1.4 User Guide,
http://www.smooks.org/mediawiki/index.php?title=V1.4:Smooks_v1.4_User_Guide#Getting_Started,
 Abruf: 27.06.2014
- Smooks (2014c):** Main Page,
http://www.smooks.org/mediawiki/index.php?title=Main_Page,
 Abruf: 29.09.2014
- Smooks (2014d):** Eclipse IDE,
http://www.smooks.org/mediawiki/index.php?title=Eclipse_IDE,
 ,Abruf: 29.09.2014
- Sourceforge (2014a):** Talend Open Studio for Data Integration,
<http://sourceforge.net/projects/talend-studio/files/stats/timeline>,
 Abruf: 24.06.2014
- Sourceforge (2014b):** Smooks,
<http://sourceforge.net/projects/smooks/files/stats/timeline>,
 Abruf: 24.06.2014
- Sourceforge (2014c):** Jaspersoft ETL,
<http://sourceforge.net/projects/jasperetl/files/stats/timeline>,
 Abruf: 27.06.2014
- Sourceforge (2014d):** KETL,
<http://sourceforge.net/projects/ketl/files/KETL/stats/timeline>,
 Abruf: 27.06.2014
- Sourceforge (2014e):** Apatar Data Integration/ ETL,
http://sourceforge.net/projects/apatar/files/Apatar%20Data%20Integration_ETL/stats/timeline,
 Abruf: 27.06.2014
- Sourceforge (2014f):** Files – Apatar Data Integration/ ETL,
http://sourceforge.net/projects/apatar/files/Apatar%20Data%20Integration_ETL,
 Abruf: 01.07.2014
- Sourceforge (2014g):** Apatar Data Integration/ ETL,
http://sourceforge.net/p/apatar/_members/,
 Abruf: 28.06.2014
- Springer Gabler Verlag (2014a):** Gabler Wirtschaftslexikon,
<http://wirtschaftslexikon.gabler.de/Archiv/77765/modularitaet-v7.html>,
 Abruf: 01.07.2014
- Springer Gabler Verlag (2014b):** Gabler Wirtschaftslexikon,
<http://wirtschaftslexikon.gabler.de/Archiv/77319/zuverlaessigkeit-v7.html>,
 Abruf: 01.07.2014

- Straube, C. (2011):** Transforming and Splitting huge EDI files with Smooks,
<http://www.catify.com/2011/03/29/transforming-and-splitting-huge-edi-files-with-smooks/>,
Abruf: 29.06.2014
- Talend (2011):** Talend im dritten Jahr in Folge mit InfoWorld Bossie Awards ausgezeichnet,
<http://de.talend.com/about-us/press/talend-im-dritten-jahr-folge-mit-infoworld-bossie-awards-ausgezeichnet>,
Abruf: 27.06.2014
- Talend (2013):** Talend erhält 40 Millionen Dollar in strategischer Finanzierungsrunde,
<http://de.talend.com/about-us/press/talend-closes-40-million-strategic-funding>,
Abruf: 29.09.2014
- Talend (2014a):** Wer wir sind,
<http://de.talend.com/about-us>,
Abruf: 28.06.2014
- Talend (2014b):** Talend Open Studio,
<http://de.talend.com/products/talend-open-studio>,
Abruf: 29.06.2014
- Talend (2014c):** Technische Details: Datenintegration,
<http://de.talend.com/products/specifications-data-integration>,
Abruf: 27.06.2014
- Talend (2014d):** Partners,
<http://de.talend.com/ecosystem/partners>,
Abruf: 28.06.2014
- Talend (2014e):** Ecosystem,
http://de.talend.com/ecosystem/partners-technology?field_company_type_tid_1=All&page=4,
Abruf: 29.06.2014
- Talend (2014f):** ESB,
<https://de.talend.com/products/esb>,
Abruf: 01.07.2014
- Talend (2014g):** About Us,
<http://www.talend.com/about-us/press-room/awards>,
Abruf: 28.06.2014
- Talend (2014h):** Über Uns,
<http://de.talend.com/about-us/careers>,
Abruf: 28.06.2014
- Talend (2014i):** Big-Data-Nachfrage sorgt für Rekordwachstum bei Talend,
<http://de.talend.com/about-us/press/demand-for-big-data-fuels-record-growth-for-talend>,
Abruf: 27.08.2014

- Talend (2014j):** Professional Support,
<http://de.talend.com/services/technical-support>,
Abruf: 01.07.2014
- Talend (2014k):** Talend Help Center,
<https://help.talend.com/display/TalendEnterpriseDataIntegrationStudioInstallationGuide51bEN/A.+%09%09%09FAQ>,
Abruf: 29.06.2014
- Talend (2014l):** Talend Help Center,
<https://help.talend.com/display/HOME/Talend+Open+Studio>,
Abruf: 29.06.2014
- Talend (2014m):** Services,
<http://de.talend.com/services/consulting>,
Abruf: 28.06.2014
- Talend (2014n):** Data Integration,
<http://de.talend.com/products/data-integration>,
Abruf: 30.06.2014
- Talend (2014o):** Data Integration,
<http://de.talend.com/products/data-integration/matrix>,
Abruf: 30.06.2014
- Talendforge (2014):** Talend Forum,
<http://www.talendforge.org/>,
Abruf: 28.09.2014
- TCW (o.J.):** Total Cost of Ownership Analyse (TCO),
<http://www.tcw.de/management-consulting/servicemanagement/total-cost-of-ownership-tco-analyse-213>,
Abruf: 24.06.2014
- Technische Uni Berlin (2014):** DIN EN ISO 9241,
www.uselab.tu-berlin.de/wiki/index.php/DIN_EN_ISO_9241,
Abruf: 27.06.2014

Gesprächsverzeichnis

Horlacher, M./u. a. (15.05.2014): Besprechung der Aufgabenstellung, Sitzungsraum
Auftraggeber

Erweiterungen relationaler Datenbanken in Data-Warehouse-Systemen

Schriftliche Ausarbeitung
im Rahmen der Lehrveranstaltung „Integrationsseminar“
Kompetenzzentrum Open Source (KOS)

Vorgelegt von

Dominique Gallus, Bernd Graef,
Marc Sparn

am 14.07.2014

Fakultät Wirtschaft
Studiengang Wirtschaftsinformatik
WWI2011V

Inhaltsverzeichnis

Abkürzungsverzeichnis	IV
Abbildungsverzeichnis.....	V
Tabellenverzeichnis.....	VI
1 Einleitung	1
2 Einführung und Motivation.....	2
2.1 Grundlagen zu Data-Warehouse-Systemen	2
2.2 Star- und Snowflake-Schema.....	4
2.3 Relevanz von Data-Warehouse-Systemen.....	5
3 Erweiterung relationaler Datenbanken	6
3.1 Partitionierung.....	6
3.1.1 Horizontale Partitionierung	7
3.1.2 Vertikale Partitionierung	8
3.2 Indizierung	9
3.2.1 B-Bäume und Varianten	10
3.2.2 Bitmap-Index.....	10
3.2.3 Verbundindexe.....	13
3.2.4 Mehrdimensionale Indexstrukturen	14
3.3 Materialisierte Sichten	16
3.4 Analytisches SQL.....	18
3.4.1 Star-Join	19
3.4.2 Der CUBE-Operator	20
3.4.3 Die GROUPING-Funktion	21
3.4.4 Sampling.....	21
3.4.5 Sequenzorientierte Analyse	22
3.4.6 Rangfolge	22
3.4.7 Dynamische Fenster	23
3.4.8 Bloomfilter.....	24
3.5 Hardware	24
3.5.1 Skalierbarkeit	24
3.5.2 Parallelverarbeitung	25
3.5.3 Clustering.....	25
3.6 Physische Organisation.....	26
3.6.1 Spaltenorientierte Datenhaltung.....	26
3.6.2 Hauptspeicherdatenbanken	27
3.7 Einfügen von Daten.....	28

4	Kriterienkatalog	28
4.1	Auswahl geeigneter relationaler Datenbanksysteme	29
4.2	Untersuchung der ausgewählten Systeme	30
4.2.1	Partitionierung	31
4.2.2	Indizierung	32
4.2.3	Materialisierte Sichten	33
4.2.4	Analytisches SQL	34
4.2.5	Hardware	35
4.2.6	Physische Organisation	36
4.2.7	Einfügen von Daten	37
5	Anwendungsbeispiele	38
5.1	Star Join, CUBE und GROUPING anhand des DWH „Star*Kauf“	38
5.2	Sequenzbasierte Analyse, Rangfolge und dynamische Fenster anhand des TCP-D DWH 41	
6	Beurteilung	45
6.1	Theoretische Erweiterungen	45
6.2	Vergleich Open Source Systeme mit kommerziellen Systemen	46
7	Zusammenfassung und Ausblick	47
	Quellenverzeichnisse	50

Abkürzungsverzeichnis

DBMS	Datenbankmanagementsystem
DWH	Data-Warehouse
MKBKBMI	Mehrkomponenten-Bereichskodierte-Bitmap-Index
MOLAP	Multidimensionales OLAP
OLAP	On Line Analytical Processing
RDBMS	Relationales Datenbankmanagementsystem
ROLAP	Relationales OLAP
SQL	Structured Query Language

Abbildungsverzeichnis

Abbildung 1: Architektur eines Data-Warehouse-Systems	3
Abbildung 2: Schematische Darstellung eines Star Schemas	4
Abbildung 3: Darstellung horizontaler und vertikaler Partitionierung	6
Abbildung 4: Beispiel für einen R-Baum	15
Abbildung 5: Spalten- und Zeilenorientierte Speicherung	26
Abbildung 6: DB-Engines Ranking der relationalen DBMS	29
Abbildung 7: Gartner Magic Quadrant 2014 zum Thema Data-Warehouse	30
Abbildung 8: Tabellendiagramm des Star*Kauf DWH	39
Abbildung 10: TPC-D-Datenbankschema	42
Abbildung 11: Beispiel zu dynamischen Fenstern anhand einer Glättung von Werten	44

Tabellenverzeichnis

Tabelle 1: Beispiel für einen Bitmap-Index	10
Tabelle 2: Mehrkomponenten-Bereichskodierter-Bitmap-Index	12
Tabelle 3: Verkauf	13
Tabelle 4: Geographie	13
Tabelle 5: Ausprägung für einen Verbundindex	14
Tabelle 6: Beispiel zu Aggregationsoperationen bei der Rangfolge	23
Tabelle 7: Kriterienkatalog zur Partitionierung	31
Tabelle 8: Kriterienkatalog zur Indizierung.....	32
Tabelle 9: Kriterienkatalog zu materialisierten Sichten	33
Tabelle 10: Kriterienkatalog zu analytischem SQL	34
Tabelle 11: Kriterienkatalog zur Hardware.....	35
Tabelle 12: Kriterienkatalog zur physischen Organisation	36
Tabelle 13: Kriterienkatalog zum Einfügen von Daten	38
Tabelle 14: Ergebnis der Beispielabfrage mit (und ohne) CUBE Operator	41
Tabelle 15: Ergebnisausschnitt der Rangfolgen Analyse mit RANK() Operator	43

1 Einleitung

Diese Arbeit beschäftigt sich mit den Erweiterungen von relationalen Datenbanken, die in Data-Warehouse-Systemen zum Einsatz kommen. Im Grunde stellen Data-Warehouses (DWH) große Datenbanken dar, die in vielen Bereichen als Grundlage für die Datenanalyse und darauf aufbauend systemgestützter unternehmerischer Entscheidungen stehen. Eine Zuordnung von DWHs in die Systemlandschaft großer Wirtschaftsunternehmen lässt sich im Bereich entscheidungsunterstützender Informationssysteme vornehmen. Hier stehen diese in Kombination mit Begriffen wie Business Intelligence sehr im Fokus und sind elementarer Bestandteil. Die Datenbanken, die den Data Warehouses zugrunde liegen, bestehen heutzutage überwiegend auf relationaler Technologie. Diese seit geraumer Zeit im Bereich der Datenbanken eingesetzte Technologie wurde für die Verwendung im Bereich DWH um einige Konzepte und Features erweitert. Ziel dieser Erweiterungen ist es, relationale Datenbanken an die neuen Herausforderungen aus dem Bereich DWH, wie steigendes Datenvolumen, komplexe Entscheidungen und vor allem Geschwindigkeit, anzupassen. Beispiele für diese Erweiterungen und Features sind unter anderem Möglichkeiten zur Partitionierung (Kap. 3.1), spezielle Indextypen (Kap. 3.2) oder Verbesserungen im Bereich der Datenabfrage unter dem Stichpunkt analytisches SQL (Kap. 3.4).

Diese Arbeit untersucht die im Einsatz befindlichen Erweiterungen relationaler Systeme zu Data-Warehouse-Systemen. Dazu wird zunächst eine Motivation zum Thema DWH gegeben und einige grundlegende Begriffe geklärt sowie Abgrenzungen vorgenommen (Kap. 2). Anschließend werden die recherchierten Konzepte und Features detailliert beschrieben (Kap. 3) und daran anknüpfend deren Einsatz in ausgewählten kommerziellen, sowie Open Source Data-Warehouses untersucht (Kap. 4). Hierzu werden sechs Systeme und deren Erweiterungen anhand eines Kriterienkatalogs dargestellt. Darauf basierend werden ausgewählte Konzepte anhand von Codebeispielen illustriert (Kap. 5). Im Zuge einer Beurteilung wird ein Vergleich zwischen Open Source DWH und kommerziellen DWH vorgenommen, sowie speziell auf Erweiterungen in der Theorie eingegangen, welche bisher nicht, beziehungsweise rudimentär im Einsatz sind (Kap. 6). Abschließend werden eine kurze Zusammenfassung zum Thema und ein Ausblick auf zukünftige Entwicklungen gegeben (Kap. 7).

2 Einführung und Motivation

Diese Kapitel stellt zunächst die Grundlagen zum Thema Data-Warehouse dar, die für das weitere Verständnis dieser Arbeit benötigt werden. Hierfür wird der Begriff Data-Warehouse in ein Gesamtkonzept eingeordnet und definiert (Kap. 2.1). Weiterhin werden technikbezogene Abgrenzungen des großen Themenfeldes Data-Warehouse für diese Forschungsarbeit vorgenommen. Vor allem Begriffe im Zusammenhang mit dem sogenannten Star Schema, welche im Verlauf dieser Arbeit vorausgesetzt werden, werden hier erläutert (Kap. 2.2). Abschließend wird eine Motivation zum Thema DWH gegeben, welche die Relevanz dieses Themengebietes anhand aktueller Zahlen und Kenntnisse verdeutlichen soll (Kap. 2.3).

2.1 Grundlagen zu Data-Warehouse-Systemen

In Abgrenzung zu transaktionsorientierten, operativen Datenbanksystemen, welche als „On Line Transaction Processing-Systeme“ bezeichnet werden, handelt es sich bei Data-Warehouses eher um analyseorientierte Systeme. ¹ Um im Sinne der Informationsaufbereitung den Weg der Daten aus den operativen Datensystemen als Datenquelle hin zum Informationsnutzer zu unterstützen, haben sich als gesamtheitlicher Ansatz sogenannte Data-Warehouse-Systeme als sinnvoll erwiesen. ² Eine etwas vereinfachte Darstellung solcher Data-Warehouse-Systeme zeigt folgende Abbildung.

¹ vgl. Gabriel, R. / Gluchowski, P. / Pastwa, A. (2009), S. V

² vgl. Farkisch, K. (2011), S. 7

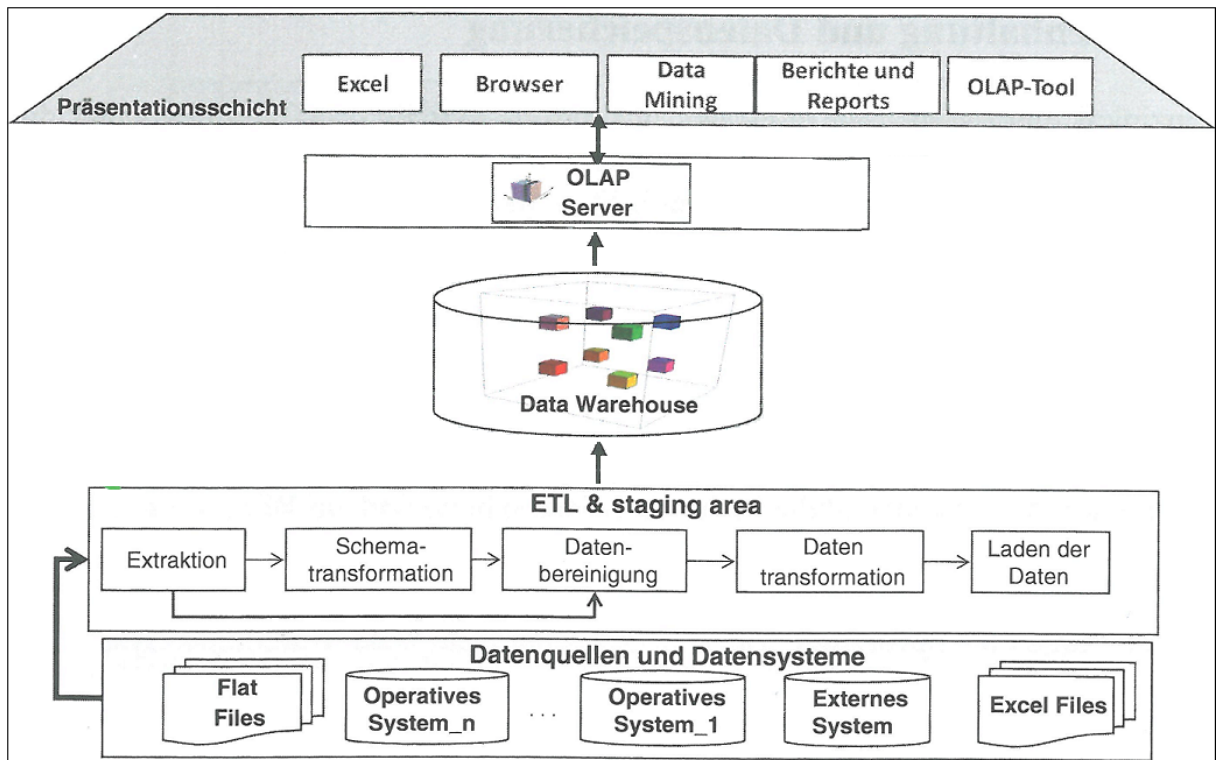


Abbildung 1: Architektur eines Data-Warehouse-Systems ³

Zentrale Komponente dieser oft auch als Referenzarchitektur bezeichneten Darstellung ist das in der Mitte der Abbildung 1 befindliche Data-Warehouse, welches auch das zentrale Thema dieser Arbeit ist. Auf Erläuterungen zu den anderen Bereichen dieser Abbildung wird aufgrund der Aufgabenstellung dieser Forschungsarbeit verzichtet.

Im Grunde stellt ein Data-Warehouse eine zentrale Datenbank dar, welche als Basis für betriebliche Informationssysteme dient. ⁴ Eine erste Definition zu diesem Thema, die auch heute noch oft verwendet wird, gibt der als Vater des Data Warehouse-Konzepts geltende W.H. Inmon. Er definiert ein Data-Warehouse wie folgt: „a data warehouse is a subject oriented, integrated, non-volatile, and time variant collection of data in support of management decisions“ ⁵. In Bezug auf die weitere Verarbeitung der Daten eines Data-Warehouses wird häufig eine als On Line Analytical Processing (OLAP) bezeichnete Technologie verwendet. Hierauf ist demnach auch die Datenhaltung in einem Data-Warehouse abgestimmt. ⁶ Für die OLAP Technologie haben sich zwei grundlegende Aspekte zur Speicherung von Daten herauskristallisiert. Es wird in eine relationale Datenhaltung, welches als relationales OLAP (ROLAP) bezeichnet wird und eine multidimensionale Datenhaltung, als multidimensionales

³ mit Änderung entnommen aus: Farkisch, K. (2011), S. 67

⁴ vgl. Tegel, T. (2005), S. 70

⁵ Farkisch, K. (2011), S. 5

⁶ vgl. Gabriel, R. / Gluchowski, P. / Pastwa, A. (2009), S. V

OLAP (MOLAP) bezeichnet, unterschieden.⁷ Wie bereits in der Einleitung dieser Arbeit erwähnt, wird vornehmlich die relationale Datenbanktechnologie als Grundlage für Data-Warehouses verwendet. Dementsprechend beschränkt sich auch diese Arbeit auf die Betrachtung der relationalen Speicherung von Daten im DWH. Um die Daten effizient und gegebenenfalls aus verschiedenen Sichtweisen (Dimensionen) zu betrachten, haben sich für deren Speicherung spezielle Schemata entwickelt. Als wichtige Grundlage für den Verlauf dieser Arbeit werden diese als Star- und Snowflake-Schema bezeichneten Speicherformen der Daten in folgendem Abschnitt separat näher betrachtet.

2.2 Star- und Snowflake-Schema

Ziel der beiden Schemata ist es, multidimensionale Datenstrukturen auf Basis von relationalen Tabellenstrukturen ohne Datenverlust darzustellen. Hierfür werden die Datenbanktabellen nach einer bestimmten Struktur aufgebaut.

Beim Star-Schema werden zunächst die Tabellen in denormalisierter Form gespeichert. Hierfür werden die Tabellen in die zweite Normalform gebracht.⁸ Weiterhin wird beim Star Schema zwischen zwei Arten von Tabellen unterschieden. Den Dimensionstabellen und den Faktentabellen. Diese werden speziell angeordnet, woraus sich auch die Bezeichnung „Star“ (engl. Stern) ergibt, wie folgende Abbildung zeigt.⁹

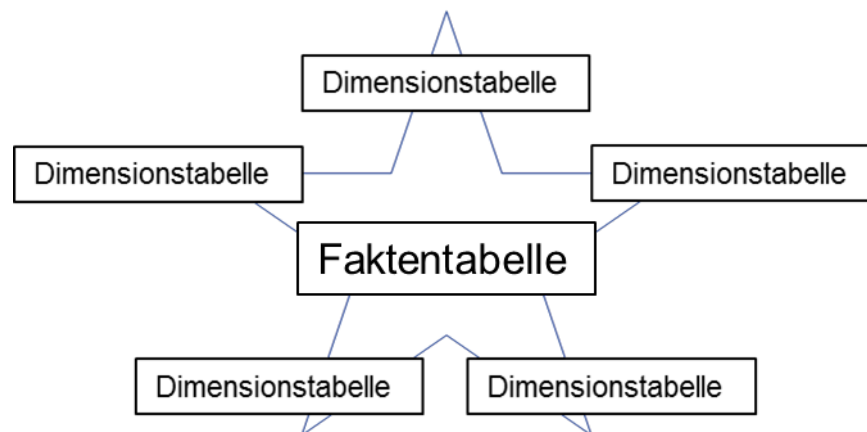


Abbildung 2: Schematische Darstellung eines Star Schemas¹⁰

Um eine zentrale Faktentabelle ordnen sich sternförmig mehrere Dimensionstabellen an, wobei zwischen der Faktentabelle und den Dimensionstabellen Primär-Fremdschlüssel Be-

⁷ vgl. Lehner, W. (2003), S. 55

⁸ vgl. Farkisch, K. (2011), S. 27

⁹ vgl. ebenda, S. 27

¹⁰ mit Änderungen entnommen aus: ebenda, S. 28

ziehungen bestehen. Die Faktentabelle eines Star Schemas beinhaltet neben den Fremdschlüsseln der Dimensionstabelle die Rechenwerte, auch als Measures bezeichnet, welche den zentralen Teil der Auswertungen darstellen.¹¹ Die Dimensionstabellen hingegen stellen hierarchisch organisiert die verschiedenen Betrachtungsweisen zu den auszuwertenden Measures dar. Über die Beziehung zwischen Dimensions- und Faktentabelle erfolgt eine direkte Zuordnung der Measures zu den einzelnen Dimensionen und deren Hierarchiestufen, über welche eine Aggregation der Daten möglich ist.

Das Snowflake-Schema kann als eine Variante des Star-Schemas gesehen werden, bei der die Daten in der dritten Normalform vorliegen und sich somit durch Auslagerungen weitere, sogenannte Satellitentabellen entstehen.¹²

2.3 Relevanz von Data-Warehouse-Systemen

Für viele Unternehmen ist aufgrund der steigenden Datenflut eine effiziente und standardisierte Datenhaltung, aufgrund derer fundierte Analysen durchgeführt werden können, essenziell. Dass für diese Anforderungen aufgrund der Geschwindigkeit der Auswertungen, der einheitlichen, soliden Informationsbasis und der Verfügbarkeit von historischen Daten passende Konzept des Data-Warehouse ist daher für die meisten Unternehmen unverzichtbar geworden. Nach einer Studie des renommierten Forschungsinstituts BARC (Business Application Research Center) setzten bereits im Jahr 2011 etwa 86% aller Unternehmen auf die Data-Warehouse Technologie.¹³ Galten Data-Warehouses anfangs noch für viele mittelständische Unternehmen als zu teuer, sind diese heutzutage aufgrund der immer günstiger werdenden Standardkomponenten wie CPUs und Speicher auch für kleinere Unternehmen erschwinglich.¹⁴ Zudem haben sich mittlerweile neben den großen kommerziellen Anbietern von Datenbanken wie Oracle, IBM und Microsoft auch viele kleinere Unternehmen, sowie Open Source Anbieter im Bereich Data-Warehouse etabliert und haben konkurrenzfähige Systeme auf den Markt gebracht. Gerade dieser Aspekt wird auch im weiteren Verlauf dieser Arbeit im Zuge einer Beurteilung zu kommerziellen und Open Source Data-Warehouses aufgegriffen.

Die Erweiterungen, welche die relationalen Datenbanksysteme hin zu Data-Warehouses gebracht haben und auf welche im folgenden Kapitel im Einzelnen eingegangen wird, werden zudem immer weiterentwickelt und optimiert. Alles in allem belegen die genannten As-

¹¹ vgl. Azevedo, P. (u.a.) (2009), S. 51

¹² vgl. Farkisch, K. (2011), S. 30

¹³ vgl. Kurzlechner, W. (2011), S. 1

¹⁴ vgl. Gfaller, H. (2010), S. 1

pekte die hohe Relevanz von Data-Warehouses und lassen vermuten, dass die bereits im Jahr 2011 sehr hohe Verbreitung weiter wachsen wird.

3 Erweiterung relationaler Datenbanken

Dieses Kapitel beschreibt die Erweiterungen relationaler Datenbanken in Data-Warehouse-Systeme, welche in dieser Arbeit untersucht werden. Die verschiedenen Aspekte werden dazu anhand von Oberbegriffen in verschiedene Bereiche unterteilt. Demnach werden in den folgenden Abschnitten detailliert die Bereiche Partitionierung (Kap. 3.1), Indizierung (Kap. 3.2), materialisierte Sichten (Kap. 3.3), analytisches SQL (Kap. 3.4), Hardware (Kap. 3.5), physische Organisation (Kap. 3.6) und das Einfügen von Daten (Kap. 3.7) betrachtet.

3.1 Partitionierung

Die Idee der Partitionierung stammt ursprünglich aus dem Bereich verteilter, paralleler Datenbanksysteme, wo zur Lastverteilung die Relationen auf verschiedene Rechnerknoten verteilt wurden. Im Allgemeinen versteht man unter Partitionierung die Aufteilung umfangreicher Tabellen, auch Relationen genannt, in kleine Teilrelationen. Diese werden dann als Partitionen bezeichnet. Partitionen können einzeln gelesen und geschrieben werden, weshalb Transaktionen auf einzelnen Partitionen andere Partitionen nicht betreffen. Ein Vorteil davon ist, dass die Verwaltung von Partitionen effizienter ist als die von umfangreichen Relationen. Beispielsweise eignen sich vor allem Faktentabellen sowie große Dimensionstabellen für die Partitionierung. Prinzipiell wird bei der Partitionierung in eine horizontale und eine vertikale Partitionierung unterschieden, dies wird in Abbildung 3 dargestellt.¹⁵

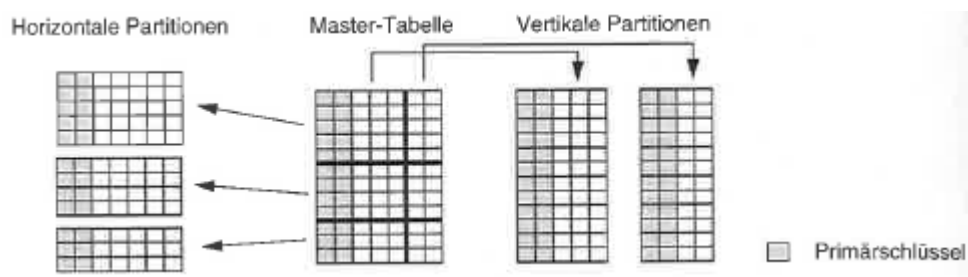


Abbildung 3: Darstellung horizontaler und vertikaler Partitionierung¹⁶

¹⁵ vgl. Bauer, A. / Günzel, H. (2013), S. 314

¹⁶ Entnommen aus: Bauer, A. / Günzel, H. (2013), S. 314

In den folgenden Abschnitten werden der horizontalen und vertikalen Partitionierung verschiedene Konzepte zugeordnet und in Zusammenhang mit Data-Warehouse-Systemen erläutert.

3.1.1 Horizontale Partitionierung

Im Bereich der Data-Warehouse-Systeme kommt vor allem die horizontale Partitionierung zum Einsatz. Dabei wird im Bereich der Dimensionstabellen in der Regel nach Attributen wie der Zeit oder dem Ort partitioniert. Bei der horizontalen Partitionierung werden Mengen von Zeilen (Tupelmengen) einer Datenbank aus der Master-Relation in paarweise disjunkte Teiltabellen, die Partitionen, aufgeteilt. Dabei sind die Attribute der Partitionen mit denen der Mastertabelle identisch. Für die Festlegung, wie die Tupelmengen ausgelagert werden, gibt es verschiedene Arten horizontaler Partitionierung. Dazu gehören insbesondere Range-, List- und Hash-Partitionierung. Weiterhin gibt es noch einen Spezialfall namens Round Robin.¹⁷

Als gebräuchlichste Art der Partitionierung gilt die sogenannte Range-Partitionierung, welche auch ohne Unterstützung des Datenbanksystems einsetzbar ist. Hierbei werden Werte bzw. Wertebereiche eines Attributs einer Tabelle als Teilungskriterium eingesetzt. Ziel der Range-Partitionierung ist die Ausnutzung der Zugriffslokalität, d. h. es wird lediglich die Teilrelation durchsucht, in deren Wertebereich die betreffenden Daten vorhanden sind. Somit muss nicht die gesamte Master-Tabelle durchsucht und gegebenenfalls für weitere Transaktionen gesperrt werden. Beispielhaft für eine Range-Partitionierung ist die Aufteilung eines Zeitattributs in Monate oder Quartale, sodass ein Datumswert lediglich in der entsprechenden Partition gesucht werden muss.¹⁸

Eine weitere Möglichkeit horizontaler Partitionierung ist die List-Partitionierung. Hier werden die Partitionen durch diskrete Wertelisten definiert. Dies bedeutet, dass eigenständig syntaktisch nicht zusammenhängende Werte zu Partitionen zusammengefasst werden können. Die Partitionen bilden auch hier, wie bei der Range-Partitionierung, Teilmengen der Master-Relation ab. Zur Absicherung wird eine Default-Partition angelegt, so dass Datensätze, welcher keiner Partition zugeordnet werden können, keine Fehler erzeugen. Ein großer Vorteil der List-Partitionierung ist, dass nicht zusammenhängende Werte in einer Partition zusammengefasst werden können. Ein Beispiel für die List-Partitionierung ist die Zusammenfas-

¹⁷ vgl. Bauer, A. / Günzel, H. (2013), S. 315

¹⁸ vgl. ebenda, S. 315

sung augenscheinlich beliebiger Städte in einer Partition. Dies ist gerade dann hilfreich, wenn Auswertungen über verschiedene Standorte durchgeführt werden müssen.¹⁹

Im Gegensatz zur Range- und List-Partitionierung wird bei der Hash-Partitionierung die Trennung nicht nach der inhaltlichen Bedeutung der Attribute, sondern anhand einer Hash-Funktion vorgenommen. Jede Zeile der Master-Tabelle wird einer Hash-Funktion übergeben, welche anschließend die Zuordnung der Attribute zu den jeweiligen Partitionen vornimmt. Vor allem in Kombination mit der Parallelverarbeitung (Kap. 3.5.2) wird diese Art der Partitionierung eingesetzt und führt zu einer gleichmäßigen Verteilung der Daten, und somit der Last, auf die verschiedenen Rechnerknoten.²⁰

Ein Spezialfall der horizontalen Partitionierung stellt das sogenannte Round-Robin-Verfahren dar. Hierbei werden die Datensätze einer Tabelle gleichmäßig auf verschiedene Platten verteilt, sodass jede Platte die gleiche Anzahl an Datensätzen enthält. Die Datensätze haben wie bei der Hash-Partitionierung keine inhaltliche Ordnung. Beim Round-Robin-Verfahren wird jedoch keine Hash-Funktion genutzt, sondern der Datenbestand sequentiell durchlaufen und anhand der Reihenfolge auf die verfügbaren Platten verteilt. Nachteil dieses Verfahrens ist, dass Punkt- und Bereichsabfragen schwer möglich sind, da aufgrund der fehlenden Ordnung jede Partition durchsucht werden muss. Bei sequentiellen Abfragen auf den gesamten Datenbestand hat dieses Verfahren jedoch extreme Performancevorteile.²¹

Als Fazit zur horizontalen Partitionierung kann gesagt werden, dass im Bereich DWH vor allem Range- und Hash-Partitionierung effizient eingesetzt werden können, wenn diese auf häufig gestellte Anfragen.²²

3.1.2 Vertikale Partitionierung

Im Gegensatz zur horizontalen Partitionierung, wo zeilenweise Datensätze in Relationen ausgelagert werden, werden bei der vertikalen Partitionierung einzelne Attribute aus Datensätzen ausgelagert. Hierdurch können beispielsweise die nur selten abgefragt werden, in Teiltabellen ausgelagert werden. Bei der vertikalen Partitionierung sind die Partitionen über Projektionen der Masterrelationen definiert. Somit haben die Partitionen denselben Primär-

¹⁹ vgl. Bauer, A. / Günzel, H. (2013), S. 315 f.

²⁰ vgl. ebenda, S. 316

²¹ vgl. Singh, S. (2009), S. 536

²² vgl. Bauer, A. / Günzel, H. (2013), S. 316

schlüssel wie die Masterrelation und die Kardinalität aller Teiltabellen zur Mastertabelle ist 1:1.²³

Im Bereich Data-Warehouse spielt die vertikale Partitionierung im Gegensatz zur horizontalen Partitionierung eine untergeordnete Rolle. Ein Spezialfall vertikaler Partitionierung stellen sogenannte Minidimensionen (original: „demographic mini dimension“) dar. Diese kommen bei extrem großen Dimensionstabellen, auch „Monster Dimensionstabellen“ genannt, zum Einsatz. Das Problem bei Monster Dimensionstabellen mit vielen Attributen ist, dass viele Attribute selten abgefragt werden. Da trotzdem bei jeder Abfrage ein Verbund zwischen der Fakten- und der Dimensionstabelle hergestellt werden muss, sind die Kosten der Abfrage hoch. Weiterhin ändern sich viele häufig abgefragte Attribute, wie beispielsweise der Nachname oder die Adresse, oft. Dies führt dazu, dass die Dimensionstabelle immer weiter wächst. Die beiden beschriebenen Probleme werden dadurch gelöst, dass Attribute, welche häufig geändert werden, aus der ursprünglichen Dimensionstabelle in eine extra Tabelle ausgelagert werden. Die Besonderheit hierbei ist, dass nur verschiedene Kombinationen der Attributwerte abgespeichert werden und dafür einen eigenen Primärschlüssel erhalten. Somit bleiben die erstellten Extratabellen relativ klein und auch das Datenvolumen der Dimensionstabellen bleibt geringer. Die Extratabellen werden fortan als Minidimension bezeichnet und direkt an die Faktentabelle angeknüpft. Ein Kritikpunkt an dieser Vorgehensweise ist, dass die logische Unabhängigkeit der Dimensionen verletzt wird.²⁴

3.2 Indizierung

Indexe sind Datenstrukturen, welche den Zugriff auf die Daten einer Tabelle beschleunigen. Durch einen Suchbaum lässt sich z.B. der Zugriff auf bestimmte Werte mit logarithmischem Aufwand ermöglichen, ohne dabei eine komplette Datei durchsuchen zu müssen. Ohne Index ist in RDBMS ein Full-Table-Scan nötig, d. h. eine Tabelle wird vollständig durchlaufen. Eine Problematik bei Data-Warehouse-Systemen ist, dass Tabellen, vor allem die Faktentabellen, oft sehr groß sind, sodass ein Full-Table-Scan nicht effizient ist. Darüber hinaus betreffen typische Anfragen in einem DWH nur einen relativ kleinen Anteil der Daten. Folglich ist die Anwendung von Indexstrukturen in diesem Bereich sinnvoll, um die zu lesenden Daten zu verringern.²⁵

Die folgenden Kapitel beschreiben die für den Bereich Data-Warehouse-Systeme relevanten Varianten von Indexstrukturen.

²³ vgl. Bauer, A./ Günzel, H. (2013), S. 316

²⁴ vgl. ebenda, S. 317 f.

²⁵ vgl. Köppen, V. / Saake, G. / Sattler, K. (2012), S. 181

3.2.1 B-Bäume und Varianten

B-Bäume sind eindimensionale Baumstrukturen und demnach für mehrdimensionale Abfragen eher ungeeignet. Sie sind insbesondere für eindimensionale Bereichsanfragen zu verwenden. In den meisten Systemen sind sie bereits vorimplementiert. Im Vergleich zu mehrdimensionalen Indexstrukturen, die neu zu implementieren sind, sind B-Bäume dadurch sehr konkurrenzfähig. Die Vorteile eines B-Baums sind seine robuste und generische Datenstruktur, die unabhängig vom konkreten Datentyp verwendet werden kann. Es ist lediglich eine Ordnung auf den Datenwerten erforderlich. Zudem bietet er effiziente Aktualisierungsalgorithmen an und ist kompakt speicherbar. Im Bereich der Data-Warehouse-Systeme existieren jedoch zwei große Nachteile. Zum einen verursachen Attribute mit einer geringen Kardinalität Datenstrukturen, die die Daten zwar korrekt speichern jedoch nicht mehr die Vorteile einer Baumstruktur aufzeigen. Man spricht dabei von degenerierten B-Bäumen. Zum anderen sind zusammengesetzte Indexe mit B-Bäumen möglich, allerdings ist die Reihenfolge der Attribute wichtig für die Performance der Abfragen. Aufgrund dieser Schwachstellen kommen bei Data-Warehouse-Systemen andere Indexstrukturen vor, die im Weiteren beschrieben werden.²⁶

3.2.2 Bitmap-Index

Bitmap-Indexe sind insbesondere bei Data-Warehouses sinnvoll, da sie auch bei kleinen Attributkardinalitäten performant einsetzbar sind. Der Bitmap-Index basiert auf einem Bit-Array. Für jede mögliche Ausprägung eines Attributes wird ein Bit-Vektor angelegt. Dabei entspricht jedes Tupel einem Bit. Dieses Bit wird auf 1 gesetzt wenn das indexierte Attribut des Tupels mit dem Referenzwert des Bit-Vektors übereinstimmt.²⁷

TID	B	F	O
TID1	0	0	1
TID2	0	1	0
TID3	0	0	1
TID4	1	0	0
TID5	1	0	0
...

Tabelle 1: Beispiel für einen Bitmap-Index²⁸

²⁶ vgl. Köppen, V. / Saake, G. / Sattler, K. (2012), S. 184 f.

²⁷ vgl. ebenda, S. 190

²⁸ Mit Änderungen entnommen aus: ebenda, S. 190

Tabelle 1 zeigt einen Bitmap-Index für das Attribut „Bestellstatus“ einer Tabelle „Bestellung“ für die die Werte B = in Bearbeitung, F = Fertig und O = Offen möglich sind. Zudem sieht man, dass die Anzahl der Bit-Vektoren für eine Dimension der Kardinalität des Attributes entspricht. Vorteil dieser Technologie ist die Reduzierung des Speicherbedarfes, sowie eine effiziente Auswertung von Abfragen. Im Vergleich zu baumbasierten Strukturen lassen sich die degenerierten B-Bäume vermeiden, zudem bietet der Bitmap-Index bessere Unterstützung bei einer hohen Anzahl von Dimensionen, da diese vorteilhaft verknüpft werden können. Die Kosten für die Aktualisierung sind zwar höher, dieser Punkt ist aber bei Data-Warehouse-Systemen zu vernachlässigen, da hauptsächlich Lesezugriffe getätigt werden. Insbesondere bei der Star-Transformation kann der Bitmap-Index nutzbringend eingesetzt werden.²⁹

Die Variante, die am häufigsten im Einsatz ist, ist der Standard-Bitmap-Index. Hier wird in einem Bitmap-Index jede Dimension getrennt voneinander abgespeichert, d. h. für jeden möglichen Attributwert wird ein Bit-Vektor erzeugt. Die Auswahl der Ergebnisse erfolgt durch die Verknüpfung entsprechender Bit-Vektoren. Beispielhaft werden für die Anfrage „alle im März geborenen Frauen anzeigen“, alle Datensätze ausgewählt, die im Bit-Vektor für das Attribut „Geschlecht“ und im Bit-Vektor für „Geburtsmonat“ jeweils den Wert 1 für das Bit weiblich und das Bit März haben.³⁰

Eine weitere Möglichkeit, einen Bitmap-Index zu realisieren ist der Mehrkomponenten-Bitmap-Index. Dieser ist in der Praxis jedoch noch nicht im Einsatz. Besonders bei Attributen mit hoher Kardinalität, die viele Bit-Vektoren verursachen, könnte der Mehrkomponenten-Bitmap-Index gewinnbringend eingesetzt werden. Als gängiges Beispiel gelten hier die Postleitzahlen: beim Standard-Bitmap-Index kämen hierfür 100.000 Bitmap-Vektoren zum Einsatz. Sofern man eine Einteilung in die ersten beiden und die letzten drei Ziffern vornimmt reduziert sich dieser Speicheraufwand auf 1.100 Vektoren. Der Speicheraufwand reduziert sich somit von $100 \cdot 1.000$ auf $100 + 1.000$. Bei einer Aufteilung in fünf verschiedene Bereiche benötigt man sogar nur 50 Bit-Vektoren.³¹

Der Standard- und der Mehrkomponenten-Bitmap-Index sind insbesondere für Punktanfragen anzuwenden. Bei Bereichsanfragen haben sie jedoch den Nachteil, dass viele Bit-Vektoren miteinander verknüpft werden müssen. Aufgrund dessen besteht die Idee der Bereichskodierten-Bitmap-Indexe. Im Bit-Vektor wird ein Bit auf 1 gesetzt, wenn der Wert des Attributs kleiner oder gleich dem gegebenen Wert ist. Beispielhaft wird hier das Attribut „Mo-

²⁹ vgl. Köppen, V. / Saake, G. / Sattler, K. (2012), S. 190 f.

³⁰ vgl. ebenda, S. 192

³¹ vgl. ebenda, S. 193 f.

nat., verwendet. Für den September erhalten die Bits von September bis Dezember den Wert 1, die restlichen den Wert 0. Eine Bereichsanfrage benötigt somit nur noch zwei Bit-Vektoren. ³²

Eine Kombination der letzten beiden Varianten ist der Mehrkomponenten-Bereichskodierte-Bitmap-Index, kurz MKBKBI. Dieser besteht zunächst aus einem Mehrkomponenten-Bitmap-Index. Auf jede Gruppe von Bitmap-Vektoren wird dann Bereichskodierung angewendet. Vorteile dieser Technologie sind zum einen der geringere Speicherbedarf aufgrund der Mehrkomponenten-Technik und zum anderen können Bereichsanfragen durch die Bereichskodierung besser umgesetzt werden.

Tabelle 2 zeigt einen Auszug aus einem MKBKBI für das Attribut „Monat“. Die Spalte B-3-0 ist nicht notwendig, da sie immer den Wert 1 enthält. ³³

Monat	B-1-1`	B-0-1`	B-2-0`	B-1-0`	B-0-0`
Juni - 5	1	0	1	1	0
April - 3	1	1	0	0	0
Januar - 0	1	1	1	1	1
Dez. - 11	0	0	0	0	0

Tabelle 2: Mehrkomponenten-Bereichskodierter-Bitmap-Index ³⁴

Eine weitere Möglichkeit ist der Intervallkodierte-Bitmap-Index. Dabei steht jeder Bit-Vektor für ein fest definiertes Intervall. Zum Beispiel das Intervall 0 enthält die Monate Januar bis Juni und das Intervall 1 Februar bis Juli bis hin zum Intervall 6 welches die Werte Juli bis Dezember enthält. ³⁵

Zusammenfassend für die Eignung der verschiedenen Bitmap-Indexstrukturen und deren Effizienz gilt folgendes: ³⁶

- Standard-Bitmap-Index: schnelle und effiziente Implementierung, aber großer Bedarf an Speicher bei hoher Kardinalität des indexierten Attributes
- Mehrkomponenten-Bitmap-Index: optimal bei der Reduzierung des Speicherbedarfes bei Punktabfragen und Anzahl der benötigten Leseoperationen
- Bereichskodierter-Bitmap-Index: optimal für einseitig begrenzte Bereichsanfragen
- Intervallkodierter Bitmap-Index: optimal für zweiseitig begrenzte Bereichsanfragen, sofern typische Intervallgröße bekannt ist, z.B. ein Quartal

³² vgl. Köppen, V. / Saake, G. / Sattler, K. (2012), S. 194

³³ vgl. ebenda, S. 194 f.

³⁴ Mit Änderungen entnommen aus: ebenda, S. 195

³⁵ vgl. ebenda, S. 196

³⁶ vgl. ebenda, S. 197 f.

3.2.3 Verbundindexe

Im Gegensatz zu klassischen Indizes beschleunigt der Verbundindex nicht den Zugriff auf eine gespeicherte Tabelle, sondern auf den Verbund zweier Tabellen. Er kommt vor allem in RDBMS, aber auch in objektorientierten und XML Datenbanken zum Einsatz. ³⁷

In Data-Warehouse-Systemen kommt er insbesondere zum Verbund von Fakten- und Dimensionstabellen, bspw. über ein Fremdschlüsselattribut, vor. Ein mögliches Einsatzszenario ist die Gruppierung von Verkaufsfakten für eine Filiale. Die folgenden Tabellen zeigen eine beispielhafte Anwendung eines Verbundindexes. ³⁸

V_ROWID	GeoID	ZeitID	Verkäufe	...
0x001	101	11	200	
0x002	101	11	210	
0x003	102	11	190	
0x004	102	11	195	

Tabelle 3: Verkauf ³⁹

Tabelle 3 zeigt eine beispielhafte Verkaufstabelle, die als Faktentabelle in einem DWH dient.

G_ROWID	GeoID	Filiale	Stadt	...
0x100	101	Allee-Center	Magdeburg	
0x101	102	Bördepark	Magdeburg	
0x102	103	Anger	Erfurt	

Tabelle 4: Geographie ⁴⁰

Tabelle 4 zeigt die dazugehörige Dimensionstabelle zur geographischen Lage der Filialen.

0x100: {0x001, 0x002, ...}

³⁷ vgl. Köppen, V. / Saake, G. / Sattler, K. (2012), S. 198

³⁸ vgl. ebenda, S. 198

³⁹ Mit Änderungen entnommen aus: ebenda, S. 199

⁴⁰ Mit Änderungen entnommen aus: ebenda, S. 199

0x101: {0x003, 0x004, ...}
0x102: {...}

Tabelle 5: Ausprägung für einen Verbundindex ⁴¹

Tabelle 5 bildet den daraus abgeleiteten Verbundindex ab.

Eine besondere Variante des Verbundindexes ist der Bitmap-Verbundindex. Dabei handelt es sich um eine Kombination eines Bitmap-Indexes mit einem Verbundindex. Dies erspart die Berechnung verschiedener Verbunde. Im vorhergehenden Beispiel ermöglicht die Realisierung als Bitmap-Index außerdem die Verknüpfung mit anderen Bitmap-Indexes der Tabelle „Verkauf“. ⁴²

3.2.4 Mehrdimensionale Indexstrukturen

Eine Technologie mehrdimensionaler Indexstrukturen ist der R-Baum. Er entsteht aus dem eindimensionalen B-Baum, ist aber angepasst an die Anwendung auf mehrdimensionale Bereiche. Mit einem R-Baum können Punkte in einem mehrdimensionalen Raum indiziert werden. Abbildung 4 zeigt einen R-Baum mit Datensätzen, die als Rechtecke dargestellt werden. Grundsätzlich bietet der R-Baum die gleichen Funktionen wie der B+-Baum, jedoch gilt es einige Aspekte zu beachten. Beim Suchen können sich verschiedene Regionen der Rechtecke überlappen, so dass mehrere Nachfolger benötigt werden. Beim Einfügen von Daten soll das Rechteck gefunden werden, welches nicht erweitert werden muss. Falls dies nicht möglich ist, soll das Rechteck genommen werden, das am wenigsten angepasst werden muss. Das Löschen von Daten spielt für Data-Warehouse-Systeme keine Rolle. ⁴³

⁴¹ Mit Änderungen entnommen aus: vgl. Köppen, V. / Saake, G. / Sattler, K. (2012), S. 199

⁴² vgl. ebenda, S. 199

⁴³ vgl. ebenda, S. 204 ff.

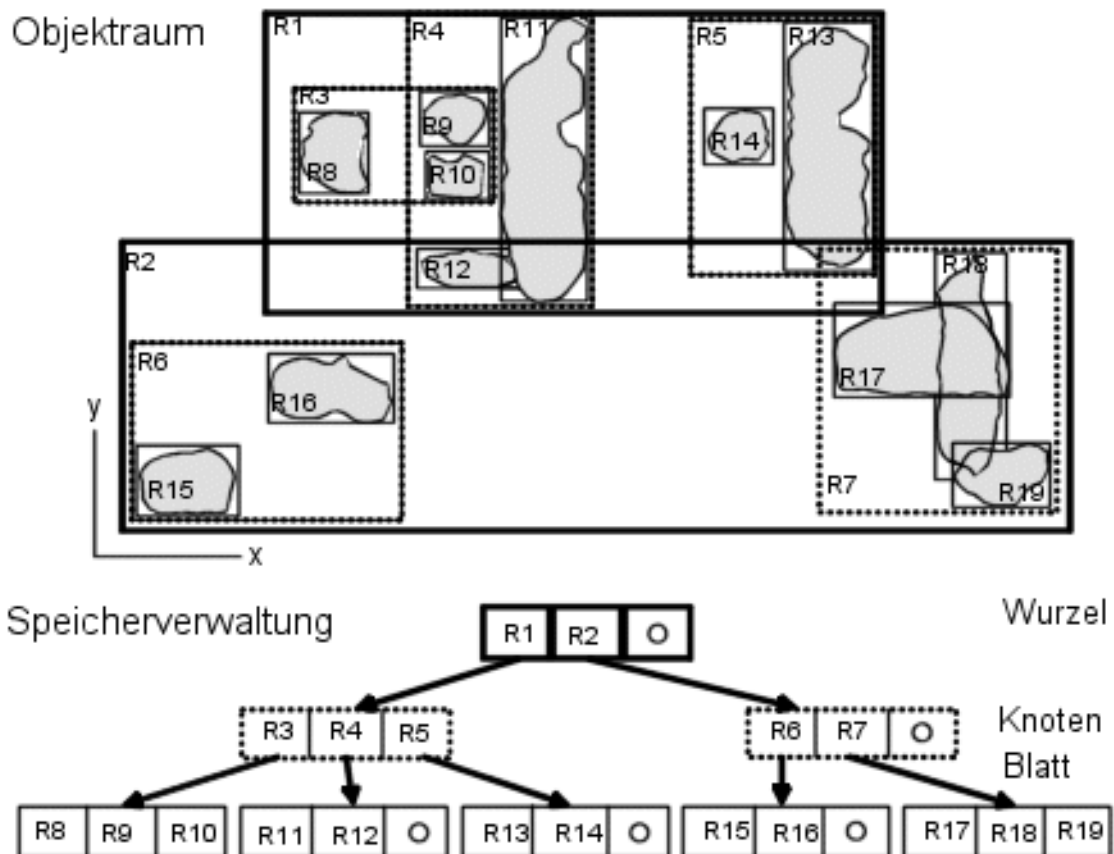


Abbildung 4: Beispiel für einen R-Baum ⁴⁴

Eine weitere Variante der R-Bäume ist der R+-Baum. Er vermeidet überlappende Rechteckregionen. Kommt es zu einer Überlappung werden die Rechtecke aufgeteilt, man spricht dabei vom Clipping. Die Konsequenz, die sich daraus ergibt ist die Notwendigkeit von mehr Rechtecken als bei R-Bäumen. Dadurch benötigt die Punktsuche nur noch einen Pfad, da Überlappungen nicht mehr möglich sind. Weitere Varianten von R-Bäumen sind der R*-Baum, der durch spezielle Einfügemöglichkeiten die Aufteilung der Rechtecke optimiert. Und der R*a -Baum, welcher für den Einsatz in der Datenanalyse zugeschnitten ist. Hiermit sollen aggregierte Daten innerhalb von Baumknoten mit abgespeichert werden. Dabei hält der innere Knoten die aggregierten Daten der Unterpunkte. ⁴⁵

Neben R-Bäumen existieren auch die sogenannten UB-Bäume. Hierbei wird eine eindimensionale Indexstruktur (siehe B-Baum) als Speicherstruktur für die mehrdimensionalen Daten verwendet. Somit findet eine Umwandlung der mehrdimensionalen Daten in eindimensionale Daten statt. Der Datenraum wird durch eine Kurve in disjunkte Teilräume aufgeteilt. Hierfür wird zumeist die sogenannte Z-Kurve eingesetzt. Jeder Punkt des Raumes, der die zu inde-

⁴⁴ enthalten in: Universität Rostock (2001)

⁴⁵ vgl. Köppen, V. / Saake, G. / Sattler, K. (2012), S. 206 f.

xierenden Werte enthält, wird auf einen skalaren Wert abgebildet (Z-Wert). Diese Z-Werte fungieren dann als Schlüssel im B+-Baum. ⁴⁶

3.3 Materialisierte Sichten

Bei den materialisierten Sichten werden oft durchgeführte Anfragen berechnet und zur Beantwortung weiterer Anfragen herangezogen. ⁴⁷ Die Ergebnisse der Berechnung sind dauerhaft in einer Relation abgespeichert. Diese wird beim Anlegen bzw. vor dem ersten Zugriff gefüllt. Die Anfragen müssen im Anschluss so geändert werden, dass sie auf die materialisierten Sichten zugreifen können. Dabei gibt es zwei Probleme: einerseits das Query Containment, das prüft, ob die Anfrage auf die Sicht zugreift und andererseits Query Rewriting, welches die eigentlich Umformulierung der Anfrage ist. Das bedeutet, dass die Sicht für die Anfragebearbeitung genutzt wird. ⁴⁸ Dies wird gemacht, um die Anfrage der Anwender nicht ändern zu müssen. Dabei kann die materialisierte Sicht auch nur für einen Teil der Anfrage genutzt werden und für die restliche Anfrage die Detaildaten. Man spricht hierbei von Multi-blockanfragen. ⁴⁹

Eine Voraussetzung für die Wiederverwendung der Berechnung ist die Additivität der Aggregatsfunktionen. Das bedeutet, es muss eine Zusammenfassung von Werten auf einen Wert möglich sein. Dies ist z. B. mit SUM möglich. COUNT ist nur semiadditiv, weil die Kardinalitäten hier aufsummiert werden. Mit AVG indirekt-additiv möglich, da eine Division durch den Kardinalitätswert durchgeführt wird. ⁵⁰

Bei der Auswahl der Sichten kann ein so genanntes Aggregationsgitter genutzt werden, aus dem die benötigten Sichten abgeleitet werden. Das Gitter zeigt über Gruppierungen, welche Attributskombinationen direkt oder indirekt abgeleitet werden können. Dafür gibt es zwei verschiedene Verfahren: ⁵¹

- Statisch: Menge zum Zeitpunkt x wird durch eine Person oder einen Algorithmus festgelegt und bleibt dann bis zur nächsten Aktualisierung erhalten; Problem hierbei ist, dass nur das alte Anfrageverhalten für die Bestimmung der Menge herangezogen wird; ein Beispiel für dieses Verfahren ist das Auswahlverfahren nach Harinarayan, Rajaraman und Ullman: hier wird über einen Algorithmus bei vorgegebenem Aggre-

⁴⁶ vgl. Köppen, V. / Saake, G. / Sattler, K. (2012), S. 207 f.

⁴⁷ vgl. Bauer, A. / Günzel, H. (2013), S. 322

⁴⁸ vgl. FH Köln (2010)

⁴⁹ vgl. Bauer, A. / Günzel, H. (2013), S. 323 f.

⁵⁰ vgl. ebenda, S. 325

⁵¹ vgl. ebenda, S. 326

gationsgitter und maximalen Speicherplatz die Menge der zu materialisierenden Aggregationsknoten mit dem höchsten Nutzen berechnet, dabei muss auch der Mehrspeicheraufwand für die materialisierten Sichten berücksichtigt werden ⁵²

- Dynamisch: Aufgrund des hohen Aktualisierungsaufwands, des historischen Anfrageverhaltens und der fehlenden Einbindung von Adhoc-Anfragen beim statischen Verfahren ist es sinnvoll, einen Teilbereich des Speichers für die erneute Verwendung von Anfrageergebnissen vorzusehen; bspw. weil Anwender oft Bezug auf ihre eigene Anfrage nehmen und diese dann bereit steht; ein Beispiel für dieses Verfahren ist das semantische Caching: dabei werden Sekundärspeicherblöcke in Caches abgelegt, um einen beschleunigten Zugriff auf die häufig benötigten Daten zu erreichen; die Administration der Caches wird anhand von formalen Gesichtspunkten und nicht anhand des Inhaltes durchgeführt, da der Inhalt nicht bekannt ist; Dies ist der Vorteil dieses Verfahrens: die Verdrängungsentscheidung wird datenabhängig getroffen. Durch den Einblick in den Anwendungsbereich der Daten ist dieses Verfahren insbesondere bei OLAP anwendbar ⁵³

Die Aktualisierung der Sichten kann auf zwei Weisen erfolgen. Bei der inkrementellen Aktualisierung wird nach der Bearbeitung der Daten über eine Formel und der Art der Änderung die neue Sicht berechnet. Die Rematerialisierung nimmt eine vollständige Löschung und Neuberechnung der geänderten Sicht vor. Darüber hinaus existieren Aktualisierungsstrategien, die aufgrund der Granularität oder des Zeitpunkts bestimmt werden. Bei der Granularität muss aufgrund der Datenkonsistenz häufig das gesamte DWH aktualisiert werden, was zeitaufwändig und unflexibel ist. Für den Zeitpunkt gibt es drei unterschiedliche Ansätze der Aktualisierung:

- Sofortige Aktualisierung: synchron zur Datenaktualisierung
- Verzögerte Aktualisierung: bei Zugriff auf die Sicht; Nachteil: evtl. lange Wartezeiten
- Snapshot: asynchrone Aktualisierung, festgelegt durch Benutzer; Nachteil: Zugriff auf alte Daten möglich ⁵⁴

Das hauptsächliche Problem der Aktualisierung ist Wahrung der Konsistenz der Daten. Dabei sind verschiedene Aspekte zu beachten.

- Anwenderdefinierte Aktualität: Daten müssen genau aktuell sein, es können aber auch zeitliche, wertmäßige oder versionsbezogene Vorgaben angegeben werden, bspw. die Sicht darf nur zwei Stunden älter sein als die Basistabelle

⁵² vgl. Bauer, A. / Günzel, H. (2013), S. 326 f.

⁵³ vgl. ebenda, S. 329 f.

⁵⁴ vgl. ebenda S. 330 f.

- Anfragekonsistenz: Anfragen an die materialisierte Sicht und die originale Datenbasis müssen das gleiche Resultat liefern
- Sitzungskonsistenz: aufeinander aufbauende Anfragen werden als eine Sitzung bezeichnet, innerhalb der die Konsistenz sichergestellt sein muss
- Aktualisierungsgranulate: Daten, die unabhängig voneinander aktualisiert werden. Das gesamte DWH zu aktualisieren, ist die einfachste Strategie, einzelne Granulate die flexibelste, da die Aktualisierungsstrategie je nach Granulat gesondert bestimmt werden kann. Die Folge ist ein höherer Aktualisierungs- und Verwaltungsaufwand führen kann. ⁵⁵

Für den Bereich der materialisierten Sichten bestehen vier Hauptproblemzonen. Zum einen die Verwendung, da sie keinen Einfluss auf die Erstellung der Anfragen haben darf. Zum anderen die Anfrageersetzung, d. h. die Anfragen müssen so aufgebaut sein, dass materialisierte eingesetzt werden können. Außerdem spielt die Auswahl der Sichten eine Rolle, dabei werden die Kosten für die Speicherung und Verwaltung dem Nutzen gegenüber gestellt. Der letzte Aspekt ist die Wartung. Hierbei ist es notwendig, die Änderungen in den Relationen in die materialisierten Sichten einzufügen. ⁵⁶

Die Funktionalität der materialisierten Sichten ist im SQL-Standard nicht enthalten, sondern muss durch den Hersteller des Datenbanksystems implementiert werden. Der Vorteil dieser Technologie ist eine Performancesteigerung, vor allem bei häufig wiederkehrenden Anfragen. Als Nachteile gelten eine redundante Aufbewahrung der Daten und die daraus entstehenden Konsistenzprobleme. ⁵⁷

3.4 Analytisches SQL

Unter den Bereich analytisches SQL fallen Erweiterungen bezüglich der Abfragesprache SQL, welche sich im Zuge der Erweiterung von relationalen Datenbanken zu Data-Warehouses ergeben haben. Hierunter fallen beispielsweise die Aspekte Star-Join sowie die Operatoren CUBE, GROUP und SAMPLING. Weiterhin gehören die Aspekte sequenzorientierte Analyse, Rangfolgen, dynamische Fenster und als Spezialfall der sogenannte Bloomfilter zu dieser Kategorie.

⁵⁵ vgl. Bauer, A. / Günzel, H. (2013), S. 331 f.

⁵⁶ vgl. FH Köln (2010)

⁵⁷ vgl. ebenda

3.4.1 Star-Join

Aufgrund der Problematik, dass bei multidimensional formulierten Abfragen mittels SQL in der Regel viele verschiedene Dimensionstabellen mit der Faktentabelle verknüpft werden müssen, was bei einfachen Joins zu Performanceproblemen führt, wurde für den Bereich Data-Warehouse ein spezielles Konzept entwickelt. In Anlehnung an das zugrundeliegende Star-Schema wird in diesem Zusammenhang ein solcher Mehrfachverbund von Dimensions- und Faktentabellen als Star-Query bezeichnet. Multidimensionale Anfragen werden hierbei über ein sogenanntes Star-Join-Anfragemuster durchgeführt.⁵⁸ Der Aufbau des Star-Joins besteht aus:

- SELECT-Klausel: diese enthält die aggregierten Kenngrößen (Measures) aus der Faktentabelle und die gewünschte Ergebnisgranularität, welche später in der GROUP BY-Klausel angegeben wird
- FROM-Klausel: hier werden alle an der Abfrage beteiligten Dimensions- und Faktentabellen angegeben. Die Dimensionen werden sowohl als Einschränkung als auch zur Aggregation verwendet.
- WHERE-Klausel: diese enthält die Verbundbedingungen für die Verknüpfung der verschiedenen Tabellen und Restriktionsbedingungen in Bezug auf die Dimensionsattribute.
- GROUP BY-Klausel: hier wird die gewünschte Ergebnisgranularität verwendeter Dimensionen angegeben.⁵⁹

Der Star-Join kann darüber hinaus optimiert werden. Die Optimierung des Star-Joins steht dabei in Zusammenhang mit den Indextechniken. Ein Problem hierbei ist, dass viele relationale Datenbanksysteme keinen Mehrfachverbund, sondern nur einen paarweisen Verbund ermöglichen. Ein $n+1$ Wegeverbund wird dabei in eine Sequenz paarweiser Verbunde aufgeteilt, wodurch sich $(n+1)!$ mögliche Verbundreihenfolgen ergeben. Die Aufgabe von Optimierern ist es daher, die günstigste Verbundreihenfolge herauszufinden.⁶⁰

Hierfür wurde bei IBM der sogenannte Semiverbund von Dimensionstabellen als Optimierungstechnik entwickelt. Das Prinzip hierbei ist, dass auf der Faktentabelle für jede Dimension ein einfacher B*-Baumindex angelegt wird. Dann werden durch einen Semiverbund der Dimensionstabellen und deren jeweiligen Index die Tupelidentifizier der Verbundkandidaten in der Faktentabelle ermittelt. Die Schnittmenge der Kandidaten, welche über ein Hash-

⁵⁸ vgl. Bauer, A. / Günzel, H. (2013), S. 254

⁵⁹ vgl. ebenda, S. 254

⁶⁰ vgl. ebenda, S. 319 f.

Verfahren ermittelbar sind, ergibt anschließend die benötigten Tupelidentifizierer der Faktentabelle. Für die ermittelten Tupel wird die paarweise Verbundmethode eingesetzt, wodurch nicht die komplette Faktentabelle in den Verbund eingeht, sondern nur die tatsächlich angefragten Tupel.⁶¹

Bei Oracle wird die Optimierung des Star-Joins über die sogenannte Star-Transformation vorgenommen. Das Ziel hierbei ist, wie beim Semiverbund von Dimensionstabellen, dass die Verbundoperationen nicht mit der kompletten Faktentabelle durchgeführt werden sollen. Voraussetzung hierfür ist, dass jede Fremdschlüsselspalte der Faktentabelle mit einem Bitmap-Index versehen sein muss. Die Anfragen werden dann nach folgendem Schema bearbeitet.

1.

- relevante Tupel in der Faktentabelle werden ermittelt
- die einzelnen Bitmap-Vektoren die mit der jeweiligen Dimensionstabelle korrespondieren, werden erzeugt
- der Bitmap-Vektor repräsentiert die Menge an Tupeln, die die jeweilige Bedingung der Dimensionstabelle erfüllt. Anschließend werden die Bitmap-Vektoren mit Hilfe einer AND-Operation zu einem Ergebnisvektor zusammengefasst, welcher die Menge an Tupeln, die zur Ergebnismenge gehören, repräsentiert
- über diesen Ergebnisvektor werden die eigentlichen Datensätze aus der Faktentabelle ermittelt.

2.

- für die Tupel der Ergebnismenge werden klassische Verbundmethoden eingesetzt, wobei der Optimierer der Datenbank die jeweils beste Methode, z.B. einen Hash-Verbund, auswählt.
- die Folge davon ist, dass, wie beim Semiverbund von Dimensionstabellen bei IBM, nur die tatsächlich angefragten Tupel in den Verbund eingeht.⁶²

3.4.2 Der CUBE-Operator

Ziel des CUBE-Operators ist es, die Problematik mühsamer, komplexer und oft auch ineffizienter analytischer Abfragen mittels SQL-Statements zu vereinfachen. Dazu bildet der CUBE-

⁶¹ vgl. Bauer, A. / Günzel, H. (2013), S. 321

⁶² vgl. ebenda, S. 321 f.

Operator aus einer Menge von gegebenen Gruppierungsattributen alle möglichen Gruppierungskombinationen. Verwendet wird der CUBE-Operator bei der GROUP BY-Klausel. Der Aufbau des CUBE-Operators beginnt nach der SELECT FROM WHERE Anweisung mit der GROUP BY-Klausel. Die Syntax lautet dabei: GROUP BY CUBE (Gruppierungsattribute). Im Allgemeinen hat der CUBE-Operator interdimensionalen Charakter, d. h. er kann auf Attribute verschiedener Dimensionen angewendet werden.⁶³

3.4.3 Die GROUPING-Funktion

Die GROUPING-Funktion schränkt die Teilsummen im Ergebnis des CUBE-Operators ein. Dabei wird die GROUPING-Funktion bei der HAVING-Klausel verwendet und bekommt ein oder mehrere Attribute als Parameter. Die Funktion liefert 1, wenn über das in der HAVING-Klausel angegebene Attribut aggregiert wurde. Andernfalls, wenn über das in der HAVING-Klausel angegebene Attribut gruppiert wurde, liefert die Funktion 0. Der Aufbau der GROUPING-Funktion, basierend auf dem CUBE-Operator, lautet: HAVING/HAVING NOT (GROUPING (Attribut1) = 1/0 AND GROUPING (Attribut n) = 1/0)^{64 65}

3.4.4 Sampling

Mittels Sampling soll schnell ein Überblick über die Daten gegeben werden, sodass ein erster Eindruck gewonnen werden kann. Dafür sollen nicht alle Detailwerte der Faktentabelle betrachtet werden, da dies in der Regel sehr viele (mehrere Millionen) sind. Daher wird eine Beispielmenge, bzw. Stichprobe, von Werten betrachtet, wodurch eine kürzere Antwortzeit und ein trotzdem annähernd genaues Ergebnis erzielt wird. Hierfür wird die Analyse auf einem relativ hohen Verdichtungslevel durchgeführt. Dies wird dadurch erreicht, dass Daten niedriger Granularität in temporären Tabellen, Sample-Tabellen genannt, als Stichprobenmengen abgelegt werden. Zur Verfeinerung der Ergebnisse kann anschließend auf höherer Granularität eine vollständige Analyse der Daten der Faktentabelle vorgenommen werden. In der Regel liefern jedoch bereits die Sample-Tabellen über 90% genaue Ergebnisse.⁶⁶

⁶³ vgl. Farkisch, K. (2011), S. 42 f.

⁶⁴ Der Schrägstrich zeigt die alternativen Eingabemöglichkeiten an

⁶⁵ vgl. Farkisch, K. (2011), S. 43 f

⁶⁶ vgl. Mucksch, H. / Behme, W. (2000), S. 235 f.

3.4.5 Sequenzorientierte Analyse

Die sequenzorientierte Analyse wird beispielsweise zur Ermittlung laufender Summen, gleitender Durchschnitte oder Ranking-Analysen durchgeführt. Sie basiert auf der OLAP-Funktion und kann als Erweiterung der Aggregationsfunktionen gesehen werden. Die Aggregationsfunktionen werden hierbei auf lokal generierte Partitionen angewendet. Am Ende einer Aggregationsoperation werden dann die OLAP-Funktionen durch eine OVER-Klausel angehängt. Diese OVER-Klausel wird bei der sequenzorientierten Analyse um eine ORDER BY-Subklausel erweitert und kann optional auch auf Partitionen angewendet werden. Somit wird zunächst der Datenbestand der OLAP-Funktion über die OVER-Klausel sortiert und davor optional eine Partitionierung vorgenommen. Das Ergebnis kann dann beispielsweise aufsummiert werden. Der beispielhafte Aufbau anhand einer Summen-Funktion sieht folgendermaßen aus: `SELECT SUM (Attribut) OVER (PARTITION BY Attribut ORDER BY Attribut) AS Bezeichnung FROM Tabelle.`⁶⁷

3.4.6 Rangfolge

Die Rangfolge gehört zum Bereich der sequenzorientierten Analysen und besteht aus der OVER-Klausel (Kap. 3.4.5.). Bei der Ranking-Analyse wird der Datenbestand nach dem Gegenstand des Rankings, das Attribut, welches in eine Reihenfolge gebracht werden soll, sortiert. Der COUNT-Operator liefert dabei den gewünschten Rang. Die generelle Syntax für RANKING-Analysen lautet: `SELECT Attribut, COUNT (Attribut/e) OVER (ORDER BY Attribut) AS Bezeichnung des Ranks FROM Tabelle.` Da Ranking-Analysen sehr häufig sind, wurden hierfür einige parameterfreie Aggregationsoperatoren eingeführt. Diese lauten `RANK()`, `DENSRANK()` und `ROWNUMBER()`. Sie ersetzen in der obengenannten Syntax den COUNT-Operator inklusive des zugehörigen Attributes. Der Vorteil der RANK-Funktion ist eine exakt definierte Semantik bei Duplikaten. Äquivalente Tupel bekommen hier den gleichen Rang und Duplikate führen zu Lücken zur nächsten Platzierung im Ranking. Die DENSRANK-Funktion vergibt Ränge bei äquivalenten Tupeln mehrfach, bildet jedoch keine Lücken. Die ROWNUMBER-Funktion liefert eine eindeutige Nummerierung der Tupel, wobei mehrfach Platzierungen bei äquivalenten Tupeln nichtdeterministisch aufgelöst werden.

Tabelle 6 zeigt beispielhaft das Vorgehen bei der Rangfolgenbildung der drei Aggregationsoperatoren `RANK()`, `DENSRANK()` und `ROWNUMBER()`.⁶⁸

⁶⁷ vgl. Lehner, W. (2003), S. 113, S. 115

⁶⁸ vgl. ebenda, S. 116 f.

RANK	DENSRANK	ROWNUMBER
1	1	1
1	1	2
3	2	3
4	3	4
4	3	5

Tabelle 6: Beispiel zu Aggregationsoperationen bei der Rangfolge ⁶⁹

3.4.7 Dynamische Fenster

Dynamische Fenster werden durch eine dritte optionale Komponente, neben PARTITION BY und ORDER BY, innerhalb der OVER-Klausel ermöglicht. Hierdurch wird innerhalb einer Partition eine dynamische Gruppe gebildet. Für die Festlegung der Größe einer Gruppe, welche als Aggregatfenster bezeichnet wird, kann entweder über den Operator ROWS die Anzahl der Tupel oder über den Operator RANGE die Anzahl wertemäßig unterschiedlicher Tupel verwendet werden. Für die Ermittlung von Größen dynamischer Fenster gibt es zwei Alternativen.

1. Von einem frei wählbaren Startpunkt bis zum aktuellen Tupel.

Hierbei wird der Startpunkt über UNBOUND PRECEDING, womit das erste Tupel der Partition ausgewählt wird, festgelegt. Alternativ kann über <n>PRECEDING der n-te Vorgänger relativ zur aktuellen Position ausgewählt werden. Eine weitere Möglichkeit stellt CURRENT ROW dar, womit das aktuelle Tupel als Startposition festgelegt wird.

2. Frei wählbare Ober- und Unterschranken des Fensters

Die Schranken werden mit BETWEEN <lower bound> AND <upper bound> festgelegt, wobei die obere Grenze immer eine höhere Position als die untere Grenze haben muss. Alternativ können die Grenzen auch analog zu Alternative 1 über UNBOUND PRECEDING und UNBOUND FOLLOWING oder über <n>PRECEDING und <n>FOLLOWING oder CURRENT ROW festgelegt werden. ⁷⁰

⁶⁹ mit Änderungen entnommen aus: ebenda, S. 117

⁷⁰ vgl. Lehner, W. (2003), S. 118 f.

3.4.8 Bloomfilter

Ein Bloomfilter besteht aus einem Bit-Array und verschiedenen Hash-Funktionen und stellt eine spezielle Abfragetechnik dar, mittels der schnell festgestellt werden kann, ob Daten in einem Datenstrom schon einmal vorgekommen sind, oder zum ersten Mal auftreten. Hierzu wird mit einer Hash-Funktion ein Fingerabdruck der gelesenen Datensätze in Hash-Tabellen gespeichert. Die Problematik beim Bloomfilter ist, dass er nicht komplett exakt ist, sondern falsche positive Ergebnisse möglich sind.⁷¹

3.5 Hardware

Auch die Hardware spielt beim DWH eine entscheidende Rolle, um die Datenbank darauf auszurichten, dass die Anfragen und Analysen effektiv und effizient durchgeführt werden können. Die mitunter sehr großen Datenmengen benötigen eine Vielzahl von Ressourcen, sodass es ein erhebliches Kriterium ist, das Datenbanksystem hardwareseitig optimieren zu können. Die folgenden Abschnitte beschreiben im Wesentlichen die Punkte Skalierung, Parallelverarbeitung und Clustering.

3.5.1 Skalierbarkeit

DWHs werden oft auf Basis der Client/Server-Architektur aufgebaut. Diese Art der Architektur führt zum sogenannten Flaschenhals: der Server muss die Anfragen verarbeiten und das auch oft innerhalb einer bestimmten Zeit. Die Anfragen kommen aber nicht von einem einzelnen Anwender, es sind mehrere Anwender, die auf einmal Anfragen an den Server schicken. Zusätzlich wachsen die Daten durch das Tagesgeschäft ständig. Daher ist es wichtig, die Architektur erweiterbar, also skalierbar, zu gestalten, insbesondere auf die Menge der Daten und die Anzahl der Benutzer des Systems. Wichtig sind dabei vor allem ein geringer Aufwand und wenig Kosten, um die Skalierung umzusetzen.⁷²

Dabei gibt es zwei mögliche Arten der Skalierung:

- horizontale Skalierung, auch Scale-Out: hier werden die Server der Datenbank erweitert, Vorteile sind die Ausfalltoleranz und die Verfügbarkeit

⁷¹ vgl. Wikipedia (o.J.)

⁷² vgl. Bauer, A. / Günzel, H. (2013), S. 490 f.

- vertikale Skalierung, auch Scale-Up: hier werden die Server mit hochwertigeren (z.B. schnelleren) Komponenten (z.B. Hauptspeicher) ausgestattet ⁷³

3.5.2 Parallelverarbeitung

Bei der Parallelverarbeitung werden die Anfragen auf verschiedene Serverprozesse aufgliedert. Dabei gibt es drei verschiedene Möglichkeiten, die auch oft von den Datenbanken unterstützt werden.

- Dedicated Server: jeder Anfrage wird ein eigener Prozess zugewiesen. Die Anzahl der Prozesse ist nicht beschränkt, was zum Erreichen und auch Übersteigen der Hardwarekapazität führen kann
- Multithreaded Server: der Server übernimmt die Administration eines Zusammenschlusses von Prozessen. Solange der Zusammenschluss freie Prozesse hat, werden Anfragen zu den Prozessen zugewiesen.
- Paralleler Datenbankserver: es gibt einen Verbund von mehreren selbstständigen Serversystemen. Die Aufteilung der Anfragen erfolgt unter Beachtung der jeweiligen Serverkapazität (Load-Balancing). Für die Wahrung der Integrität und Konsistenz gibt es spezielle Prozesse. ⁷⁴

3.5.3 Clustering

Beim Clustering sind mehrere Architekturen aneinander gekoppelt. Es handelt sich demnach um einen Zusammenschluss von Komponenten (z.B. Servern), die wie eine Einheit agieren. Dabei sind die verschiedenen Komponenten meist über ein Netzwerk verbunden, um die Datenadministration durchzuführen. Dabei können verschiedene Implementierungstechniken, wie z.B. ein gemeinsamer Datenspeicher (shared-memory), verwendet werden. Die Vorteile des Clustering sind die geringen Kosten und die simple Umsetzung. Die Umsetzung ist besonders deshalb einfach, da die neuen Komponenten einfach angeschlossen werden können und nicht das gesamte Netzwerk geändert werden muss. Dies unterstützt ebenfalls eine einfache Skalierung der Datenbank. Nachteilig ist, dass die Sicherheit und die

⁷³ vgl. FH Köln (2012)

⁷⁴ vgl. Bauer, A. / Günzel, H. (2013), S. 553

Stabilität stark an die Konfiguration des Systems und die Kenntnisse der Administratoren gebunden sind.⁷⁵

3.6 Physische Organisation

Dieser Abschnitt widmet sich der physischen Organisation von Datenbanken. Betrachtet wird hierbei das zum einen die spaltenorientierte Datenhaltung und ihr bevorzugter Einsatz in analytischen Systemen sowie die Nutzung von Hauptspeicherdatenbanken in Zusammenhang mit der Komprimierung von Daten.

3.6.1 Spaltenorientierte Datenhaltung

In RDBMS kommt überwiegend die zeilenorientierte Speicherung der Daten zum Einsatz. Das bedeutet, dass jede Zeile einer Tabelle zusammenhängend gespeichert wird. Die zeilenorientierte Datenspeicherung bietet Vorteile, wenn bei Anfragen und Änderungen ganze Zeilen bearbeitet werden. Bereits in den 70er Jahren existierte allerdings die Idee der spaltenorientierten Datenhaltung, insbesondere für analytische Systeme. Bei der Spaltenorientierung erfolgt die Zusammenfassung der Daten nicht zeilen-, sondern spaltenweise nach Attributen. Abbildung 5 zeigt die Unterschiede der beiden Speicherarten.⁷⁶

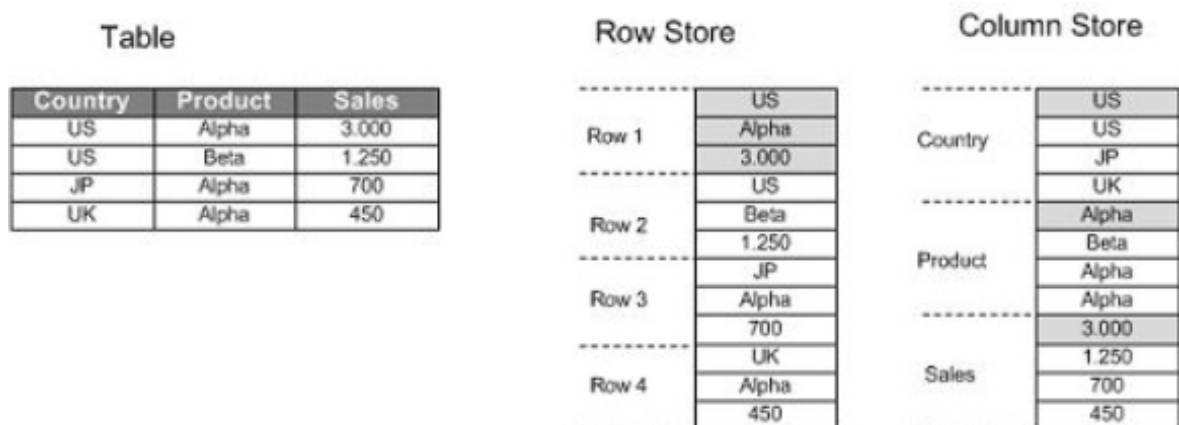


Abbildung 5: Spalten- und Zeilenorientierte Speicherung⁷⁷

Der Vorteil der spaltenorientierten Datenhaltung bei Data-Warehouse-Systemen liegt darin, dass Analyseanfragen in DWH aus Bereichsselektion und Aggregationen von Kennzahlen

⁷⁵ vgl. Bauer, A. / Günzel, H. (2013), S. 496

⁷⁶ vgl. Köppen, V. / Saake, G. / Sattler, K. (2012), S. 162

⁷⁷ enthalten in: FH Köln (2013)

über mehrere Zeilen hinweg bestehen. Der Beschleunigung der Zugriffe steht zwar ein höherer Aufwand beim Einfügen von Zeilen gegenüber. Dies ist jedoch für das Data-Warehouse-System irrelevant. Ein weiterer Vorteil ist die bessere Komprimierbarkeit der Daten, da Spalten, bezogen auf Datentyp und Wert, homogener als Zeilen sind. Die sequentielle Speicherung der Daten unterstützt die Durchführung effizienter Scans. Aus diesem Grund verzichten einige Systeme sogar vollständig auf Indexstrukturen. Ein wesentlicher Aspekt der Spaltenorientierung ist es, die Wiederherstellung der Zeilen zu gewährleisten. Hierfür bestehen zwei verschiedene Möglichkeiten. Einerseits kann eine feste Zeilenreihenfolge genutzt werden. Andererseits ist es auch möglich, die Rekonstruktion über ein Identifikationsattribut sicherzustellen.⁷⁸

3.6.2 Hauptspeicherdatenbanken

Die persistente Speicherung von Daten in einer Datenbank erfolgt auf dem Hintergrundspeicher, da der Hauptspeicher nicht alle Daten aufnehmen kann. Der Hintergrundspeicher ist aber um den Faktor 10^5 langsamer beim Datenzugriff. Der Zugriff erfolgt blockweise. Die Idee der Hauptspeicherdatenbanken sieht vor, die Datenbank vollständig im Hauptspeicher zu speichern, sofern dieser groß genug ist, um effiziente Hauptspeicherdatenstrukturen zu nutzen. Beispielsweise entspricht beim B-Baum die Seitengröße der Blockgröße des Hintergrundspeichers, wodurch die Knoten im Allgemeinen nicht ganz gefüllt sind. Durch eine Hauptspeicherdatenstruktur wird eine bessere Auslastung erreicht. Hauptspeicherdatenbanken bieten somit die Möglichkeit, Algorithmen und Datenstrukturen zu nutzen, die bisher nicht im RDBMS eingesetzt wurden. Beispielsweise ist es möglich, Bäume mit kleinen Knoten oder variierender Knotengröße zu verwenden, aber auch Komprimierungsalgorithmen, die nicht auf sequentiellen Durchlauf basieren. Allerdings ist der Hauptspeicher flüchtig und daher eher für analytische Systeme geeignet, da die Daten nicht geändert werden müssen. Ein Datenverlust ist somit nicht entscheidend. Die spaltenorientierte Speicherung und die Hauptspeichertechnologie werden insbesondere für Echtzeitanalysen verwendet. Durch die spaltenorientierte Speicherung können die Daten besser komprimiert werden, so dass sie eher in den Hauptspeicher passen. Die Kompressionsrate bei zeilenorientierter Speicherung beträgt 1:3, bei spaltenorientierter Speicherung 1:10.⁷⁹ Zudem ist die Blockgröße der Datensätze ein Thema, da diese in Data-Warehouse-Systemen in der Regel höher ist als in

⁷⁸ vgl. Köppen, V. / Saake, G. / Sattler, K. (2012), S. 162 ff.

⁷⁹ vgl. Köppen, V. / Saake, G. / Sattler, K. (2012), S. 171 ff.

normalen Datenbanken. Dadurch werden die Kompression und der Lesezugriff zusätzlich unterstützt.⁸⁰

3.7 Einfügen von Daten

Als weiteres Kriterium spielt das Einfügen von Daten eine große Rolle bei Data Warehouses. Hierbei existieren verschiedene Möglichkeiten. Das Einlesen von Datensätzen aus einer kompletten externen Datei mit Hilfe eines BulkLoaders⁸¹ sowie das Einfügen über das SQL-Statement MERGE INTO.

Beim MERGE INTO ist der Hauptanwendungsgrund, automatisch ein Update, Insert oder Delete durchzuführen. Dies wird daran entschieden, ob der Datensatz vorhanden ist oder nicht. Sofern der Datensatz vorhanden ist, bestehen die Optionen, diesen zu aktualisieren oder zu löschen. Falls der Datensatz nicht vorhanden ist, wird er neu in die Datenbank eingefügt. Im Weiteren wird der Aufbau eines MERGE INTO-Statements erläutert.

- MERGE INTO: gibt das Ziel an, in der Regel eine Tabelle
- USING: Quelle (Tabelle, View oder Select-Anfrage)
- ON: Kriterium, welche Datensätze der Suchanfrage entsprechen
- WHEN MATCHED: Aktion, die ausgeführt wird, falls Datensatz vorhanden
- WHEN NOT MATCHED: Aktion, die ausgeführt wird, falls Datensatz nicht vorhanden

Innerhalb der MATCHED-Operatoren können mittlerweile zudem Where-Klauseln eingefügt werden.⁸² Bei Data Warehouses wird durch diese Technik ein aufeinander folgendes Update von Fakten und sich wenig verändernden Dimensionen.⁸³

4 Kriterienkatalog

In diesem Abschnitt werden die in Kapitel 3 erläuterten Erweiterungen relationaler Datenbanken auf deren Verwendung in sechs verschiedenen Systemen untersucht. Dazu wird ein Kriterienkatalog erstellt, anhand dessen die zu untersuchenden Bereiche separat betrachtet

⁸⁰ vgl. Hilker, S. (2013)

⁸¹ vgl. Stokes, A. (o.J.)

⁸² vgl. Oracle (2011)

⁸³ vgl. Codeproject (2009)

werden. Bevor der Kriterienkatalog in Kapitel 4.2 beschrieben wird, werden im folgenden Abschnitt zunächst sechs geeignete Systeme für die Untersuchung ausgewählt.

4.1 Auswahl geeigneter relationaler Datenbanksysteme

Auf Basis der Aufgabenstellung dieser Forschungsarbeit sollen sowohl gängige kommerzielle, als auch Open Source Systeme auf deren Erweiterungen in Data-Warehouses untersucht werden.

In Abstimmung mit dem Auftraggeber dieser Forschungsarbeit wurde sich dabei auf die Auswahl von den vier kommerziellen Anbietern von Datenbanksystemen Oracle, Microsoft, IBM und SAP, sowie der beiden Open Source Lösungen MySQL und PostgreSQL geeinigt. Bei dieser Auswahl wurde sich zum einen an dem Popularitätsranking von DB-Engines ⁸⁴, welches in folgender Abbildung dargestellt wird orientiert. Hier stehen fünf der sechs ausgewählten Systeme auf den vorderen Rängen.

83 Systeme im Ranking, Mai 2014					
Rang	Vormonat	DBMS	Datenbankmodell	Punkte	Änderung
1.	1.	Oracle	Relational DBMS	1502,74	-11,34
2.	2.	MySQL	Relational DBMS	1309,10	+16,43
3.	3.	Microsoft SQL Server	Relational DBMS	1207,80	-2,63
4.	4.	PostgreSQL	Relational DBMS	240,64	+10,41
5.	5.	DB2	Relational DBMS	186,47	+1,89
...					
24.	↓	23. Sybase IQ	Relational DBMS	4,92	+0,16

Abbildung 6: DB-Engines Ranking der relationalen DBMS ⁸⁵

Weiterhin wird diese Auswahl durch den Magic Quadrant der Gartner Group zum Thema Data-Warehouse bestätigt. In diesem werden die Systeme der vier ausgewählten kommerziellen Anbieter im Bereich des Leader Quadranten angesiedelt, wie in Abbildung 7 rot dargestellt. Weiterhin basieren zwei weitere in dieser Abbildung orange dargestellte Anbieter auf MySQL Datenbanken.

⁸⁴ Die Berechnungsmethode für das Ranking befindet sich unter:
http://db-engines.com/de/ranking_definition

⁸⁵ mit Änderungen entnommen aus: DB-Engines (2014)



Abbildung 7: Gartner Magic Quadrant 2014 zum Thema Data-Warehouse ⁸⁶

Die Untersuchung der Systeme der sechs Data-Warehouse Anbieter Oracle, IBM, Microsoft, SAP, MySQL und PostgreSQL findet in folgendem Abschnitt statt.

4.2 Untersuchung der ausgewählten Systeme

Die im vorherigen Abschnitt festgelegten Systeme werden nun anhand eines Kriterienkataloges auf die sieben in Kapitel 3 beschriebenen Aspekte und deren Unterpunkte untersucht. Da-

⁸⁶ enthalten in: Beyer, M. / Edjlali, R. (2014), S. 3

zu wird zu jedem dieser sieben Punkte ein Ausschnitt des Kriterienkataloges in Tabellenform kurz dargestellt und dieser anschließend weiter beschrieben und analysiert.

4.2.1 Partitionierung

Im Bereich der Partitionierung wurden die sechs ausgewählten Systeme auf die drei Hauptvarianten Range, List und Hash untersucht. Weiterhin wurde die Spezialvariante Round Robin in die Untersuchung mit aufgenommen, sowie spezielle Aspekte der einzelnen Systeme in dem Bereich "Sonstige" aufgeführt. Tabelle 7 zeigt die gesammelten Ergebnisse zu diesem Themengebiet.

	Kommerziell				Open Source	
	Oracle DB 12c	IBM DB2	MS SQL Server	Sybase	MySQL	PostgreSQL
Partitionierung						
Range	ja	ja	ja	ja	ja	ja
List	ja	nein	nein	nein	ja	ja
Hash	ja	ja	nein	ja	ja	nein
Round Robin	ja	ja	ja	ja		ja
Sonstige	Composite Partitioning *	Multidimensionales Clustering			Composite Partitioning, Key Partitioning *	Composite Partitioning

Tabelle 7: Kriterienkatalog zur Partitionierung

Die vier kommerziellen Systeme von Oracle, IBM, Microsoft und Sybase beherrschen in der Regel alle der aufgeführten Partitionierungsvarianten. Eine Ausnahme hierzu stellt der SQL Server von Microsoft dar, welcher sich im Bereich Partitionierung auf die Range-Partitionierung spezialisiert hat und dafür auf List- und Hash-Partitionierung verzichtet. Weiterhin wird die List-Partitionierung auch von DB2 und Sybase IQ nicht unterstützt. Spezialfälle im Bereich Partitionierung bieten Oracle mit dem sogenannten Composite Partitioning und IBM mit dem multidimensionalen Clustering. Das Composite Partitioning bei Oracle ermöglicht eine Kombination von zwei der drei Standardarten der Partitionierung Range, List und Hash.

Die beiden Open Source Systeme MySQL und PostgreSQL stellen ebenfalls vielfältige Partitionierungsmöglichkeiten bereit. Lediglich die Hash-Partitionierung ist bei PostgreSQL nicht realisiert. Zudem bieten beide Systeme das Composite Partitioning an. MySQL stellt darüber hinaus ein sogenanntes Key-Partitioning zur Verfügung, unterstützt allerdings im Gegensatz zu allen anderen untersuchten Systemen keine Datumspartitionierung.

Zusammenfassend ist auf dem Gebiet der Partitionierung zu erkennen, dass, bis auf geringfügige Ausnahmen, alle betrachteten Anbieter die gängigen Technologien zur Verfügung stellen. Lediglich beim SQL Server von Microsoft ist eine Spezialisierung auf die Range-

Partitionierung festzustellen. Vergleicht man insbesondere die kommerziellen Produkte mit den Open-Source-Implementierungen, kann man jedoch keine relevanten Unterschiede in den angebotenen Funktionalitäten nachweisen.

4.2.2 Indizierung

Für die Indizierung wurden die verschiedenen Systeme vor allem dahingehend untersucht, ob sie die Technologie des Bitmap-Indexes bereitstellen. Darüber hinaus wurde zudem geprüft, ob Verfahren der multidimensionalen Indexierung durch die Anbieter unterstützt werden. Tabelle 8 bildet die Ergebnisse der Untersuchungen zu den verschiedenen Technologien zur Indexierung ab, die in den Systemen bereitgestellt werden.

	Kommerziell				Open Source	
	Oracle DB 12c	IBM DB2	MS SQL Server	Sybase	MySQL	PostgreSQL
Indizierung						
Standard-Bitmap-Index	ja	ja	nein, aber Bitmap-Filter	ja, bitweise funktionierende Indizes	nein, ähnliche Funktionalität durch "index_merge"	ja
R-Baum	ja	ja	nein	nein	ja	ja, in Form von GIS-Index*
UB-Baum	ja	ja	nein	nein	nein	nein
Sonstige	ja, Bitmap-Join-Index	Dynamic Bitmap Index				GIN/Hash-Index für Volltextsuche

Tabelle 8: Kriterienkatalog zur Indizierung

Auf der Seite der kommerziellen Anbieter ist zu erkennen, dass Oracle und IBM den Standard-Bitmap-Index zur Verfügung stellen. Microsoft und SAP bieten in ihren Systemen diesen zwar nicht, greifen aber auf Verfahren zurück, die eine ähnliche Funktionalität aufweisen. Neben dem Standard-Bitmap-Index bietet Oracle außerdem noch die Möglichkeit des Bitmap-Verbundindex an, während IBM seinen Nutzern einen Dynamic Bitmap Index bereitstellt. Im Bereich der mehrdimensionalen Indexierung ist ebenfalls ersichtlich, dass Oracle und IBM bereits die R- und UB-Bäume unterstützen. Die Systeme von Microsoft und SAP bieten diese noch nicht an.

Die beiden Open-Source-Vertreter MySQL und PostgreSQL sind im Bereich des Standard-Bitmap-Index ähnlich aufgestellt wie die kommerziellen Anbieter. Dabei verfügt PostgreSQL über diese Technologie, während MySQL mit „index_merge“ eine ähnliche Funktionalität anbietet. Neben dem Standard-Bitmap-Index verfügen beide Systeme auch über die Möglichkeit des R-Baums. Im Falle von PostgreSQL ist dieser durch den so genannten Geo-Informationssystem(GIS)-Index realisiert, welcher den R-Baum in späteren Versionen ersetzt

hat. Der UB-Baum ist in beiden Systemen nicht implementiert. Zusätzlich dazu bietet PostgreSQL noch einen GIN-/Hash-Index für die Volltextsuche an.

Das Thema Indizierung wird insbesondere von den Anbietern Oracle und IBM angeführt, die über alle der im Vordergrund betrachteten Technologien verfügen und darüber hinaus sogar noch Zusatzfeatures, wie den Bitmap-Verbundindex bei Oracle, zur Verfügung stellen. Der Standard-Bitmap-Index wird noch durch alle untersuchten Systeme direkt oder durch ähnliche Funktionalitäten unterstützt. Gerade bei den mehrdimensionalen Indexstrukturen lassen sich jedoch größere Unterschiede aufzeigen, da Microsoft und SAP hier keine der gesuchten Funktionen, und die beiden Open-Source-Vertreter lediglich die R-Bäume, anbieten. Der UB-Baum wird nur durch Oracle und IBM unterstützt.

4.2.3 Materialisierte Sichten

In diesem Abschnitt wird dargestellt, wie die Technik der materialisierten Sichten durch die sechs Anbieter implementiert ist. Dabei wird zum einen darauf geachtet, ob das System die Möglichkeit, materialisierte Sichten zu nutzen, generell anbietet und falls ja, die vorhandene Aktualisierungsstrategie gekennzeichnet. Tabelle 9 stellt eine kompakte Übersicht der Ergebnisse zu diesem Themenbereich dar.

	Kommerziell				Open Source	
	Oracle DB 12c	IBM DB2	MS SQL Server	Sybase	MySQL	PostgreSQL
Materialisierte Sichten						
Möglichkeit	ja	ja, Materialized Query Tables	ja, Indexed Views	ja	nein, nur mit Workaround	Ja
Aktualisierungsstrategie	automatisch, manuell	automatisch, manuell	automatisch	u.a. Exclusive oder Snapshot		manuell

Tabelle 9: Kriterienkatalog zu materialisierten Sichten

Für die vier kommerziellen Anbieter ist ein sehr einheitliches Resultat festzustellen. Alle vier bieten die Möglichkeit, materialisierte Sichten zu erstellen und zu nutzen, an. Im Falle von IBM DB2 und Microsoft SQL Server wird die Technologie zwar anders bezeichnet, bietet jedoch die gleiche Funktionalität. Die Aktualisierungsstrategie kann man insbesondere in die Varianten manuell und automatisch unterscheiden. Während Oracle und IBM beide Möglichkeiten unterstützen, ist beim Microsoft SQL Server lediglich eine automatische Aktualisierung vorgesehen. Auch SAP bietet für Sybase IQ verschiedene Strategien an. Dabei kommen vor allem die Snapshot- und die Exclusive-Technik zum Einsatz.

Ein abweichendes Bild zeigt die Betrachtung der beiden Open-Source-Systeme. PostgreSQL bietet zwar noch die Möglichkeit, materialisierte Sichten einzusetzen, diese müssen aber

manuell aktualisiert werden. Eine automatische Aktualisierungsstrategie wird nicht unterstützt. Am wenigsten ausgestattet ist in diesem Bereich MySQL, da hier materialisierte Sichten nicht angeboten werden und nur durch ein Workaround realisiert werden können.

Zusammenfassend sind hier erste Vorteile auf der Seite der kommerziellen Anbieter ersichtlich, die allesamt diese Funktionalität anbieten und bis auf Microsoft auch flexiblere Aktualisierungsstrategien ermöglichen. PostgreSQL kann dabei eher mit diesen Systemen konkurrieren, da es zumindest die Funktionalität bereitstellt. Im Bereich der Aktualisierungsstrategien ist es jedoch noch nicht so ausgereift wie die kostenpflichtigen Systeme.

4.2.4 Analytisches SQL

Der nächste Abschnitt des Kriterienkataloges beschäftigt sich mit dem Thema analytisches SQL. Dabei wird insbesondere untersucht, ob die Systeme SQL-Operationen unterstützen, die bei der Analyse von Daten im Data-Warehouse-System hilfreich sind. Die gesammelten Ergebnisse sind Tabelle 10 zu entnehmen.

	Kommerziell				Open Source	
	Oracle DB 12c	IBM DB2	MS SQL Server	Sybase	MySQL	PostgreSQL
Analytisches SQL						
Star-Join	ja	ja	ja	ja	nein	ja
Star Query Optimierung	ja, Startransformation	ja, Semiverbund von Dimensionstabellen	ja, Star Query Optimization	Query Optimizer optimiert alle Anfragen	nein	nein
CUBE	ja	ja	ja	ja	nein	ja
GROUPING	ja	ja, GROUPING SETS	ja, GROUPING SETS	ja	nein	ja, GROUPING SETS
SAMPLING	ja, Bernoulli und System	ja, Bernoulli und System	ja, Repeatable und System	nein	nein	nein
Sequenzorientierte Analyse	ja	ja	ja	ja	nein	ja
Rangfolge	ja	ja	ja	ja	nein	ja
Dynamische Fenster	ja	ja	ja	ja	nein	ja
Räumliche Abfragen	ja	ja, spatial index	ja, spatial index	ja	ja, spatial index	ja
Bloomfilter	ja	ja		ja	nein	ja

Tabelle 10: Kriterienkatalog zu analytischem SQL

Anhand des Kriterienkataloges wird deutlich, dass die kommerziellen Anbieter eine breite Palette an analytischen SQL-Operationen unterstützen und somit die Analyse komplexer Daten erleichtern. Weitgehend alle für die Untersuchung relevanten Funktionen sind hier implementiert. Die einzige Ausnahme bildet das System SAP Sybase IQ, welches das Sampling nicht ermöglicht. Darüber hinaus sind noch Unterschiede im Einsatz der Star Query Optimierung aufzuweisen. Während Oracle auf die Startransformation setzt, kommt bei IBM der Semiverbund von Dimensionstabellen zum Einsatz. Aber auch die anderen beiden Anbieter nutzen ähnliche Technologien zur Star Query Optimierung.

Bei den Open-Source-Systemen kann sofort erkennen, dass MySQL keinerlei analytische SQL-Funktionen bereitstellt. Lediglich räumliche Abfragen werden über einen spatial index unterstützt. PostgreSQL kann dagegen weitaus mehr Analyseoperationen aufweisen. Lediglich die Möglichkeit der Star Query Optimierung und des Samplings werden in diesem System nicht angeboten.

Zusammenfassend lassen sich in diesem Bereich weitere gravierende Unterschiede zwischen den kommerziellen und den Open-Source-Systemen nachweisen. Während die kostenpflichtigen Systeme weitgehend alle untersuchten Funktionen abdecken, bietet MySQL keine Unterstützung der Analysefunktionen an. PostgreSQL stellt zwar die meisten Operationen zur Verfügung, kann aber auch nicht vollständig mit den anderen vier Systemen mithalten.

4.2.5 Hardware

Der folgende Abschnitt beschäftigt sich mit den Möglichkeiten, die die Systeme hardwareseitig anbieten. Dabei werden speziell die Punkte Parallelverarbeitung, Clustering und Skalierung betrachtet. Tabelle 11 zeigt den Teil des Kriterienkatalogs, der die Ergebnisse zum Thema Hardware enthält.

	Kommerziell				Open Source	
	Oracle DB 12c	IBM DB2	MS SQL Server	Sybase	MySQL	PostgreSQL
Hardware						
Parallelverarbeitung	ja	ja	ja	ja	ja, mit Shard Query	ja
Clustering	Oracle RAC	ja, shared-nothing	ja, Failover Cluster Instance (FCI) *	ja	ja, shared-nothing	ja
Skalierung	Oracle RAC	Pure Scale	ja	ja	Auto-Sharding	ja (linear bis zu 64 CPU-Kerne)

Tabelle 11: Kriterienkatalog zur Hardware

Die Betrachtung der Möglichkeiten im Hardwarebereich liefert ein sehr ausgeglichenes Bild. Die kostenpflichtigen Systeme sind alle dazu in der Lage, Daten parallel zu verarbeiten. Auch das Clustering und die Skalierung werden jeweils unterstützt, dabei bedarf es jedoch mitunter eines Zusatztools. Für Oracle ist dies Oracle Real Application Clusters, das sowohl Clustering und Skalierung möglich macht. Bei IBM DB2 erfolgt das Clustering in einer Shared-Nothing-Architektur, das heißt die Rechnerknoten agieren unabhängig voneinander. Zur Skalierung verwendet IBM das Werkzeug Pure Scale. Beim Microsoft SQL Server ist das Thema Clustering in der Version 2014 mittlerweile durch die Failover Cluster Instance abgedeckt.

In diesem Themenbereich bieten auch die Open-Source-Produkte wieder einen ähnlichen Leistungsumfang wie die kommerziellen Systeme. MySQL unterstützt die Parallelverarbeitung durch Shard Query. Wie bei IBM erfolgt das Clustering in einer Shared-Nothing-Architektur. Die Skalierung wird ebenfalls unterstützt und wendet die Technik des Auto-Shardings an, das heißt die Tabellen werden automatisch auf Standarddatenknoten verteilt. Auch PostgreSQL bietet Möglichkeiten der Parallelverarbeitung und des Clusterings an. Ebenfalls möglich ist die Skalierung, die linear auf bis zu 64 CPU-Kerne ausgedehnt werden kann.

Das Thema Hardware ist sowohl durch die kommerziellen Systeme als auch durch die Open-Source-Produkte weitgehend abgedeckt, die Umsetzungen weisen allerdings diverse Unterschiede auf. Für manche Systeme werden zusätzliche Tools benötigt, die von den Anbietern zur Verfügung gestellt werden. Diverse Anbieter verfügen über Komplettlösungen, die die Kriterien für ein optimal eingerichtetes Data Warehouse auf der Hardwareseite erfüllen. Im Fall von Oracle handelt es sich dabei z. B. um Oracle ExaData.

4.2.6 Physische Organisation

Als weiterer Punkt des Kriterienkataloges fungiert der Bereich physische Organisation. Hierzu zählen insbesondere Techniken, die die effiziente und platzsparende Speicherung der Daten betrachten sowie ein vereinfachtes Abfragen ermöglichen. Ein wichtiger Punkt ist dabei die spaltenorientierte Datenhaltung, aber auch die Umsetzung der Komprimierung und der mögliche Einsatz als Hauptspeicherdatenbank werden betrachtet. Ferner werden auch mögliche Blockgrößen untersucht. Tabelle 12 zeigt die Resultate der Recherche zum Bereich der physischen Organisation.

	Kommerziell				Open Source	
	Oracle DB 12c	IBM DB2	MS SQL Server	Sybase	MySQL	PostgreSQL
Physische Organisation						
Spaltenorientierte Datenhaltung	ja, mit In-Memory Option	ja, mit BLU Acceleration	ja, mit Column Store Index	ja, Pionier	ja, InfoBright	ja, cstore_fdw
Komprimierung	Advanced Row Compression	Advanced Row Compression	ja	ja	Compressed Row Format	ja, TableCompression
Hauptspeicherdatenstruktur	ja, mit IMDB-Cache	ja, BLU Acceleration	ja	ja	Speicher-Engine Memory	Foreign Data Wrapper
Blockgröße	innerhalb einer DB bis zu fünf verschiedene Blockgrößen: 2 - 32 K möglich	Seitengröße begrenzt Größe eines Tablespaces: 4 - 32 K möglich	nicht anpassbare Blockgrößen, Default: 8KB	Blockgröße ergibt sich aus festzulegender Seitengröße, nur bei Neuanlage der DB anpassbar	Default: 16K, möglich: 8 - 64K	Default: 8K, möglich: 8 - 32K

Tabelle 12: Kriterienkatalog zur physischen Organisation

Die kommerziellen Produkte bieten durchweg Möglichkeiten zur spaltenorientierten Datenhaltung. Der Pionier auf diesem Gebiet ist SAP mit Sybase IQ. Die weiteren Anbieter zogen in den letzten Jahren nach und bieten inzwischen Erweiterungen an, die diese Technologie unterstützen. Für die Oracle DB benötigt man hierfür bspw. die In-Memory Option, bei IBM funktioniert die spaltenorientierte Datenhaltung mit BLU Acceleration. Auch die Komprimierung wird durch die verschiedenen Systeme ermöglicht. Insbesondere bei Oracle und IBM steht hier die Variante der Advanced Row Compression im Vordergrund, bei der alle Vorteile der Komprimierung zum Tragen kommen. Neben der spaltenorientierten Datenhaltung und der Komprimierung ist auch der Einsatz als Hauptspeicherdatenbank mit allen vier Systemen möglich. Für Oracle wird hier mit IMDB-Cache eine weitere Erweiterung benötigt, während bei IBM die spaltenorientierten Datenhaltung BLU Acceleration verwendet wird. Bezüglich der Blockgröße ist festzustellen, dass Oracle und IBM flexible Blockgrößen bis zur 32 K ermöglichen. Microsoft und SAP sind bei diesem Thema weniger flexibel, da die Blockgröße entweder gar nicht oder nur bei Neuanlage der Datenbank anpassbar ist. Der Microsoft SQL Server bietet einen Default-Wert von 8 K an, bei Sybase IQ ergibt sich die Blockgröße aus der Seitengröße, die vom Benutzer festzulegen ist.

Die Open-Source-Produkte weisen eine ähnliche Situation vor. Die spaltenorientierte Datenhaltung bei MySQL wird unter Verwendung von InfoBright unterstützt, während bei PostgreSQL erst seit kurzer Zeit durch die Erweiterung `cstore_fdw` eine spaltenorientierte Datenhaltung möglich ist. Im Rahmen der Komprimierung bieten beide Systeme eine Lösung an. MySQL verwendet das Compressed Row Format und PostgreSQL die TableCompression. Auch der Einsatz als Hauptspeicherdatenbank ist bei beiden Produkten durch verschiedene Erweiterungen möglich. Bei der Blockgröße bietet insbesondere MySQL mehr Möglichkeiten als die restlichen Systeme, da lediglich hier die Blockgröße auf bis zu 64 K gesetzt werden kann. Aber auch PostgreSQL kann diesbezüglich mit den Systemen von Oracle und IBM mithalten.

Zusammenfassend bieten alle Systeme vielseitige Unterstützungen der physischen Organisation an. Dabei kommen bei allen Systemen viele Erweiterungen zum Einsatz.

4.2.7 Einfügen von Daten

Der letzte Abschnitt des Kriterienkataloges beschäftigt sich mit dem Einfügen von Daten. Dabei spielt insbesondere der SQL-Operator `MERGE INTO`, das Vorteile beim Einfügen, Löschen und Aktualisieren von Daten bietet. Darüber hinaus wird untersucht, ob die Systeme einen BulkLoader beinhalten, der das Laden mehrerer Datensätze in eine Tabelle ermöglicht.

	Kommerziell				Open Source	
	Oracle DB 12c	IBM DB2	MS SQL Server	Sybase	MySQL	PostgreSQL
Einfügen von Daten						
MERGE INTO	ja	ja	ja		ja	nein
BulkLoader	ja	ja	ja	ja	ja	ja

Tabelle 13: Kriterienkatalog zum Einfügen von Daten

Dieses Thema ist sowohl von den kostenpflichtigen als auch von den Open-Source-Produkten sehr gut abgedeckt. Die Funktion des Operators MERGE INTO wird weitgehend unterstützt, lediglich PostgreSQL verfügt nicht über die Option. Ein BulkLoader wird zudem von allen sechs Anbietern eingesetzt.

5 Anwendungsbeispiele

Dieses Kapitel soll anhand von Codebeispielen typische Anfragen an Data Warehouses in Bezug auf die in Kapitel 3 erläuterten Konzepte veranschaulichen. Hierzu werden vor allem die im Kapitel 3.4 unter dem Begriff „analytisches SQL“ beschriebenen Operatoren, Funktionen und Konzepte herangezogen. Diese werden im Folgenden anhand von Beispielabfragen aus der Literatur betrachtet.

5.1 Star Join und CUBE anhand des DWH „Star*Kauf“

Zunächst werden der Star Join und dessen Anfragemuster (Kap. 3.4.1) sowie der CUBE Operator (Kap. 3.4.2) anhand des Beispiels DWH „Star*Kauf“ dargestellt. Das Schema des zugrundeliegenden DWH wird dazu zunächst in folgender Abbildung dargestellt.

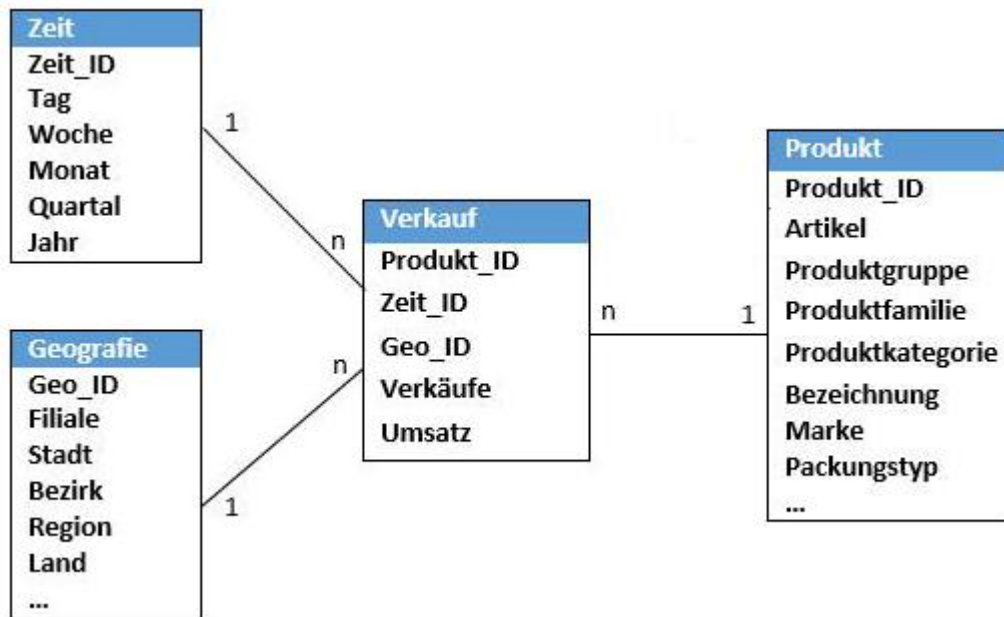


Abbildung 8: Tabellendiagramm des Star*Kauf DWH ⁸⁷

Auf Basis der Tabellen aus Abbildung 8 wird im Folgenden eine beispielhafte Abfrage nach dem Schema des Star Join Anfragemusters durchgeführt. Die der Abfrage zugrunde liegende Fragestellung „Wie viele Artikel der Produktfamilie Waschgeräte wurden 2011 pro Monat in den unterschiedlichen Regionen verkauft?“ ⁸⁸ enthält dabei unter anderem aggregierte Kenngrößen, verschiedene Granularitäten der einzelnen Dimensionen sowie Restriktionen bezüglich der Dimensionsattribute.

```

SELECT    Geografie.Region, Zeit.Monat, SUM(Verkäufe)
FROM      Verkauf, Zeit, Produkt, Geografie
WHERE     Verkauf.Produkt_ID = Produkt.Produkt_ID AND
            Verkauf.Zeit_ID = Zeit.Zeit_ID AND
            Verkauf.Geo_ID = Geografie.Geo_ID AND
            Produkt.Produktfamilie = "Waschgeräte" AND
            Zeit.Jahr = 2011 AND
            Geografie.Land = "Deutschland"
GROUP BY Geografie.Region, Zeit.Monat; 89

```

Über komplexere Gruppierungsattribute aus dem Bereich analytisches SQL können Anfragen, wie die oben dargestellte, erheblich vereinfacht werden. Vor allem der CUBE Operator, welcher in folgender Beispielanfrage dargestellt wird, spielt hierbei eine wichtige Rolle.

⁸⁷ mit Änderungen entnommen aus: Bauer, A./Günzel, H. (2013), S.245

⁸⁸ ebenda, S.254

⁸⁹ vgl. ebenda, S.254

```

SELECT    REGION, PRODFAMILIE, JAHR,
            GROUPING(REGION), GROUPING(PRODFAMILIE), GROUPING(JAHR),
            SUM(VERKÄUFE) AS VERKÄUFE

FROM ...
GROUP BY CUBE(REGION, PRODFAMILIE, JAHR) 90

```

Den Nutzen, den der CUBE Operator hier hat, wird deutlich, wenn die Abfrage ohne den CUBE Operator dargestellt wird. Hierfür müssen dann alle möglichen Gruppierungsmöglichkeiten mittels UNION und der damit einhergehenden Auswertung mehrerer Teilanfragen realisiert werden. Grob ausformuliert sieht diese Abfrage ohne CUBE-Operator in etwas abgekürzter Form wie folgt aus:

```

(SELECT    REGION, PRODFAMILIE, JAHR,
            SUM(VERKÄUFE) AS VERKÄUFE

FROM ...
GROUP BY REGION, PRODFAMILIE, JAHR)
UNION
(SELECT    NULL AS PRODFAMILIE, REGION, JAHR,
            SUM(VERKÄUFE) AS VERKÄUFE

FROM ...
GROUP BY REGION, JAHR)
UNION
            ... //(PRODFAMILIE, REGION), (PRODFAMILIE, JAHR),
            //(REGION), (JAHR)

UNION
(SELECT    PRODFAMILIE, NULL AS REGION, NULL AS JAHR,
            SUM(VERKÄUFE) AS VERKÄUFE

FROM ...
GROUP BY PRODFAMILIE)
UNION
(SELECT    NULL AS PRODFAMILIE, NULL AS REGON, NULL AS JAHR,
            SUM(VERKÄUFE) AS VERKÄUFE

FROM ... ) 91

```

Das Ergebnis der beiden Anfragen, einmal mit und einmal ohne CUBE Operator, wird in Tabelle 14 dargestellt.

⁹⁰ vgl. Bauer, A./Günzel, H. (2013), S.257

⁹¹ vgl. ebenda, S.256

Verkäufe (Region	ProdFamilie	Jahr	Verkäufe
	Bayern	Video	2005	12
	Bayern	Video	2006	48
	Bayern	Video	2007	58
	Bayern	Video	NULL	118
	Bayern	Audio	2009	31
	Bayern	Audio	2010	67
	Bayern	Audio	2011	66
	Bayern	Audio	NULL	164
	Bayern	TV	2009	15
	Bayern	TV	2010	55
	Bayern	TV	2011	51
	Bayern	TV	NULL	121
	Bayern	NULL	NULL	403
	Hessen	Video	2009	22
	...			
	Hessen	NULL	NULL	501
	NULL	Video	2009	34
	NULL	Video	2010	98
	NULL	Video	2011	155
	NULL	Video	NULL	257
	NULL	Audio	2009	82
	...			
	NULL	NULL	2009	172
	NULL	NULL	2010	382
	NULL	NULL	2011	350
	NULL	NULL	NULL	806

Tabelle 14: Ergebnis der Beispielabfrage mit (und ohne) CUBE Operator ⁹²

Weitere Beispielabfragen aus dem Bereich analytisches SQL werden im folgenden Abschnitt anhand eines anderen exemplarischen Datenmodells dargestellt.

5.2 Sequenzbasierte Analyse, Rangfolge und dynamische Fenster anhand des TCP-D DWH

Dieses Kapitel behandelt SQL Abfragen zur sequenzbasierten Analyse (Kap. 3.4.5), Rangfolgen (Kap. 3.4.6) und dynamischen Fenstern (Kap. 3.4.7) anhand des TPC-D-Datenbankschemas, welches in folgender Abbildung dargestellt wird.

⁹² mit Änderungen entnommen aus: Bauer, A./Günzel, H. (2013), S.255

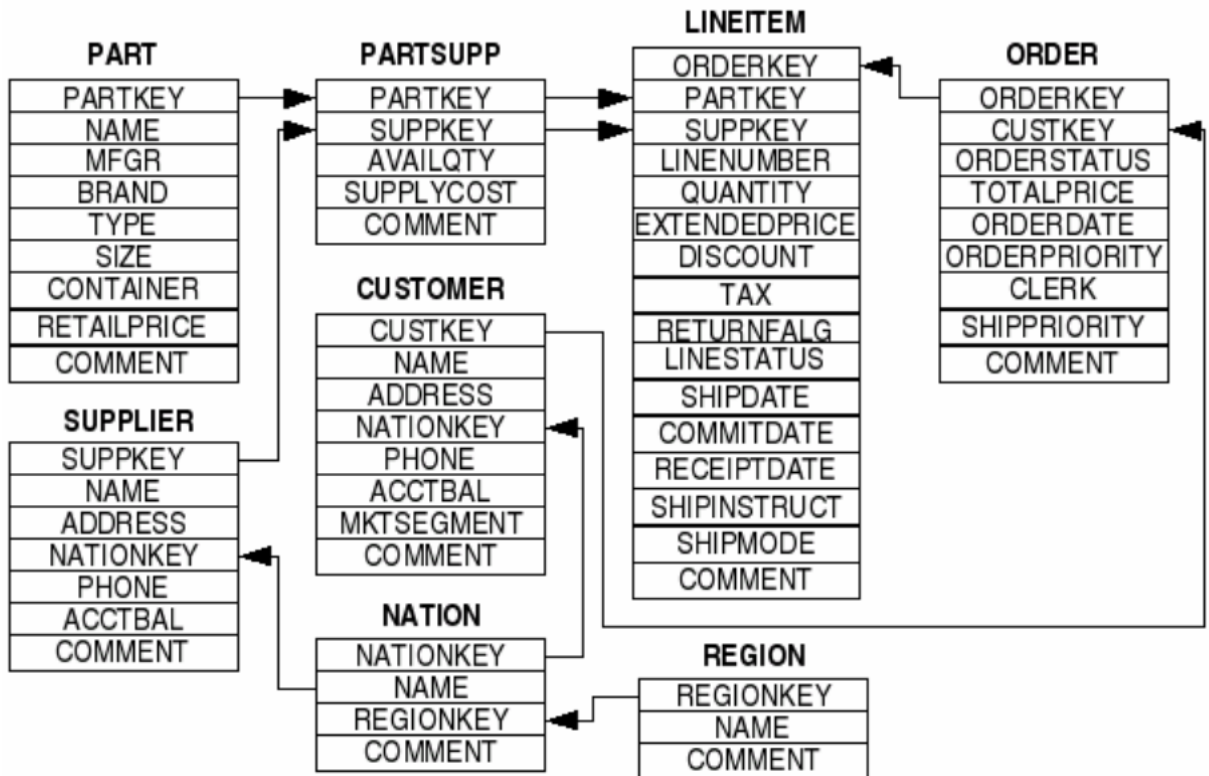


Abbildung 9: TPC-D-Datenbankschema ⁹³

Auf Basis der in Abbildung 9 dargestellten Tabellen des TCP-D Datenbankschemas lautet eine Beispielabfrage für eine sequenzbasierte Analyse folgendermaßen:

```

SELECT    L_SHIPDATE, L_SHIPMODE,
            SUM (L_QUANTITY) OVER (ORDER BY L_SHIPDATE) AS SUM_GLOBAL,
            SUM (L_QUANTITY) OVER (PARTITION BY L_SHIPMODE
                                     ORDER BY L_SHIPDATE) AS SUM_PER_SHIPMODE,
            SUM (L_QUANTITY) OVER (PARTITION BY MONTH (L_SHIPDATE)
                                     ORDER BY L_SHIPDATE) AS SUM_PER_MONTH
FROM TPCD.SD_LINEITEM; 94

```

Dabei werden die Daten in dieser Abfrage nach dem Lieferdatum geordnet (L_SHIPDATE), die kumulierten Lieferzahlen (L_QUANTITY) über den gesamten Zeitraum (SUM_GLOBAL), nach dem Auslieferungsmodus (SUM_PER_SHIPMODE) und nach dem Monat der Lieferung (SUM_PER_MONTH) ausgegeben.

Ein Beispiel einer Ranking Analyse, basierend auf den Tabellen aus Abbildung 9, könnte mit einem einfachen COUNT-Operator folgendermaßen aussehen:

⁹³ in Anlehnung an: Lehner, W. (2003), S.85

⁹⁴ vgl. ebenda, S.116


```
SELECT L_SHIPDATE, COUNT(*) OVER (ORDER BY L_QUANTITY) AS RANK_QUAN
FROM TPCD.LINEITEM; 95
```

Mit den bereits in Kapitel 3.4.6 erläuterten Aggregationsoperatoren RANK(), DENS_RANK() und ROW_NUMBER() sieht eine Beispielabfrage in Bezug auf Abbildung 9 wie folgt aus:

```
SELECT X.L_SHIPDATE, X.MTH_TOP_QUAN, X.ALL_TOP_QUAN
FROM (SELECT L_SHIPDATE,
           RANK() OVER (ORDER BY L_QUANTITY DESC) AS ALL_TOP_QUAN,
           RANK() OVER (PARTITION BY MONTH (L_SHIPDATE)
                       ORDER BY L_QUANTITY DESC) AS MTH_TOP_QUAN
        FROM TPCD.SD_LINEITEM) X
WHERE X.MTH_TOP_QUAN <= 5
ORDER BY X.ALL_TOP_QUAN DESC, X.MTH_TOP_QUAN DESC; 96
```

Ein Ergebnisausschnitt dieser Abfrage wird in Tabelle 15 dargestellt.

L_SHIPDATE	MTH_TOP_QUAN	ALL_TOP_QUAN
1998-06-19	1	1
1994-09-14	1	2
1995-06-27	2	3

Tabelle 15: Ergebnisausschnitt der Rangfolgen Analyse mit RANK() Operator ⁹⁷

Die Abfrage mit DENS_RANK() und ROW_NUMBER() würde analog zu der Zählweise in Tabelle 6 ein entsprechend anderes Ergebnis liefern.

Die Bildung dynamischer Fenster in SQL Abfragen wird in folgendem Beispiel gezeigt. Dabei werden dynamische Fenster dazu verwendet, gleitende Durchschnitte über Lieferzeiten zu bilden.

⁹⁵ vgl. Lehner, W. (2003), S.116

⁹⁶ vgl. ebenda, S.117

⁹⁷ mit Änderungen entnommen aus: ebenda, S.117

```

SELECT L_SHIPDATE,
       AVG (L_QUANTITY) OVER (PARTITION BY MONTH (L_SHIPDATE)
                              ORDER BY L_SHIPDATE
                              ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING) AS AVG_3_QUAN,
       AVG (L_QUANTITY) OVER (PARTITION BY YEAR (L_SHIPDATE)
                              ORDER BY L_SHIPDATE
                              ROWS BETWEEN 29 PRECEDING AND CURRENT ROW)
       AS AVG_29L_QUAN
FROM TPCD.SD_LINEITEM; 98

```

Das Ergebnis dieser Abfrage wird in Abbildung 10 verdeutlicht.

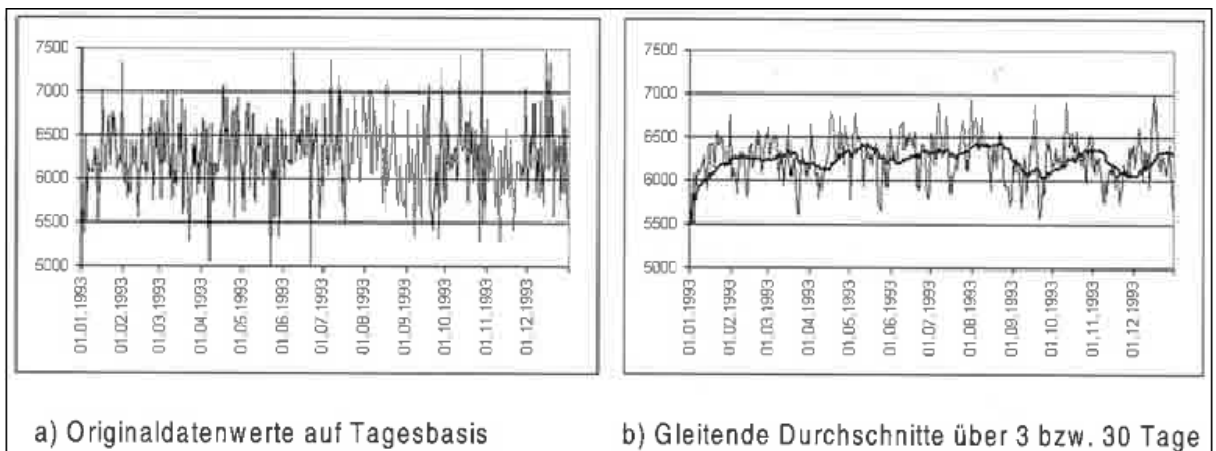


Abbildung 10: Beispiel zu dynamischen Fenstern anhand einer Glättung von Werten ⁹⁹

Hier werden im linken Bereich die Originaldaten dargestellt und im rechten Bereich die gleitenden Durchschnitte über die, in der Abfrage gebildeten, 3 und 30 Tage Zeiträume.

Folgendes Kapitel gibt nun einen Überblick über theoretische Erweiterungen im Bereich DWH und vergleicht Open Source Systeme mit kommerziellen Produkten aus diesem Gebiet.

⁹⁸ vgl. Lehner, W. (2003), S.119

⁹⁹ enthalten in: ebenda, S.119

6 Beurteilung

In diesem Abschnitt soll eine Beurteilung der ermittelten Ergebnisse aus dem Kriterienkatalog vorgenommen werden. Zunächst geht es darum, Technologien zu erwähnen, die zwar in der Theorie bereits dargelegt sind, zu denen es allerdings noch keine praktische Umsetzung gibt. Anschließend werden die sechs untersuchten Systeme verglichen. Insbesondere wird dabei auf die Unterschiede zwischen den kommerziellen Systemen und den Open-Source-Produkten eingegangen.

6.1 Theoretische Erweiterungen

Die meisten der in Kap. 3 erwähnten Technologien finden bereits erste Umsetzungen in verschiedenen RDBMS. Dennoch existieren einige Erweiterungen, die noch keinem der untersuchten Systeme zuzuordnen sind.

Zum einen wären dies viele der erweiterten Indexierungsarten. Zwar ist der Standard-Bitmap-Index in der reinen Form oder durch Methoden mit ähnlichen Funktionalitäten in allen Produkten enthalten, die erweiterten Formen, die in der Theorie dargestellt wurden, jedoch nicht. Dabei handelt es sich um den Mehrkomponenten-Bitmap-Index sowie um den bereichskodierten Bitmap-Index. Ebenfalls darunter fällt die Kombination dieser beiden Bitmap-Index-Varianten, der MKBKBMI. Des Weiteren fällt auf, dass Verbundindexierung und mehrdimensionale Indexierungsarten bisher nur bei einem kleinen Teil der betrachteten Systeme zum Vorschein kommen. Es bleibt offen, wann und in welcher Form diese Technologien in den gängigen RDBMS erscheinen und genutzt werden können.

Neben noch nicht verfügbaren Indexierungsarten bestehen darüber hinaus auch Lücken beim analytischen SQL. So werden bspw. räumliche Abfragen bereits durch einige der untersuchten Systeme unterstützt, die Implementierung erfolgte bisher jedoch nur rudimentär, sodass auf diesem Gebiet in der nahen Zukunft noch einige technologische Sprünge möglich sind. Eine ähnliche Situation zeigt sich bei den materialisierten Sichten, die zwar von den meisten Produkten ermöglicht werden, deren Einsatz jedoch noch zahlreiche Probleme mit sich bringt. Insbesondere in den Bereichen der Verwendung, Auswahl und Wartung, die in Kap. 3.3 ausführlicher skizziert wurden, zeigt sich diese Problematik. Dadurch ist auch in diesem Umfeld noch mit einigen technischen Verbesserungen zu rechnen.

Davon abgesehen sind die betrachteten Erweiterungen relativ flächendeckend im praktischen Einsatz, jedoch nicht immer bereits in allen der begutachteten Systeme enthalten. Daher ist auf dem Gebiet des Data Warehousings in der Zukunft mit einigen technologischen Neuerungen und Verbesserungen zu rechnen.

6.2 Vergleich Open Source Systeme mit kommerziellen Systemen

Die einzelnen Kategorien der Erweiterungen wurden bereits im Rahmen des Kap. 4.2 erläutert und dabei jeweils auch ein Vergleich der kommerziellen und der Open-Source-Systeme vorgenommen. In diesem Abschnitt sollen diese gesammelten Erkenntnisse nochmals kompakt präsentiert werden, um eine abschließende Beurteilung fällen zu können. In diese Beurteilung fließen lediglich Kenntnisse, die im Rahmen dieser Arbeit gewonnen wurden, ein. Bezüglich der Effizienz der Implementierungen verschiedener Erweiterungen in den untersuchten Systemen werden daher keine Aussagen vorgenommen.

Im Bereich der Partitionierung sind sowohl die kostenpflichtigen als auch die freien Systeme technologisch sehr breit aufgestellt. Die gängige Range-Partitionierung wird von allen Produkten angeboten. Bei den weiteren Partitionierungsarten kommt es dann teilweise zu Abweichungen, da nicht jeder Anbieter alle Varianten unterstützt. Es sind jedoch keine wesentlichen Unterschiede zwischen den Open-Source- und den kommerziellen Produkten zu erkennen. Auffällig ist lediglich, dass Oracle sämtliche Partitionierungstechniken implementiert hat.

Ein ähnliches Bild zeigt sich auch bei der Indizierung: wie bereits erwähnt wird der Standard-Bitmap-Index oder eine gleichartige Technologie von allen Produkten bereitgestellt. Darüber hinaus weisen insbesondere IBM und Oracle zusätzliche Implementierungen im Bereich der mehrdimensionalen Indexierung auf. Bei Oracle existiert dabei auch noch die Möglichkeit, einen Bitmap-Verbundindex zu nutzen. Während die beiden weiteren kommerziellen Anbieter die Technologien der mehrdimensionalen Indexierung, die in dieser Ausarbeitung untersucht wurden, nicht unterstützen, können die Open-Source-Produkte zumindest die R-Baum-Technik vorweisen.

Die größten Differenzen zwischen Open-Source- und kommerziellen Systemen lassen sich im Bereich des analytischen SQLs aufweisen. Während hier die kostenpflichtigen Produkte in der Regel alle Operationen bereitstellen, gibt es für MySQL überhaupt keine analytischen SQL-Funktionen, die genutzt werden können. Bei PostgreSQL sieht dies zwar deutlich vielfältiger aus, jedoch sind auch hier die Funktionen des Samplings und die Star Query Optimierung nicht implementiert. Für den Einsatz als Data-Warehouse-Systeme ergeben sich daraus zahlreiche Vorteile für die kommerziellen Anbieter bei der Analyse von Daten.

Dies gilt außerdem für den Einsatz materialisierter Sichten, die zwar, wie in Kap. 6.1 erwähnt, nur rudimentär in der Praxis vorkommen, aber in allen kommerziellen Systemen bereits angeboten werden. Für MySQL sind materialisierte Sichten nur mit einem Workaround zu erreichen, während PostgreSQL diese Technologie zwar anbietet, jedoch ausschließlich die manuelle Aktualisierung der Sichten ermöglicht.

Die Bereiche Hardware und physische Organisation werden von allen Anbietern ähnlich gut abgedeckt. Hier sind nur wenig nennenswerte Abweichungen zu erkennen, sodass kein großer Unterschied beim Einsatz als Data-Warehouse-System zu vermuten ist.

Zusammenfassend bieten die Systeme allesamt sehr viele der untersuchten Erweiterungen bereits an. Bei vielen Technologien können jedoch Vorteile zugunsten der kommerziellen Systeme ausgemacht werden. Insbesondere Oracle und IBM bieten hier deutlich mehr Funktionalitäten an als die untersuchten Konkurrenten. Die größte Schwäche der Open-Source-Systeme offenbart sich allerdings im Bereich des analytischen SQLs, der in den kostenpflichtigen Produkten einen ausgeprägten Eindruck macht, während PostgreSQL hier noch einige Defizite aufweist und MySQL ganz auf diese Komponente verzichtet.

Insofern lassen sich für den Einsatz im DWH vor allem die Systeme von Oracle und IBM empfehlen, die dem Nutzer alle bisher in der Praxis umgesetzten und in dieser Arbeit untersuchten Implementierungen bieten können. Die Wahrscheinlichkeit, dass sie auch bei kommenden Erweiterungen die Pionierstellung einnehmen, ist dadurch durchaus hoch. Von den beiden Open-Source-Vertretern dieser Forschungsarbeit ist vor allem PostgreSQL empfehlenswert, da hier nicht vollständig ohne analytische SQL-Anfragen ausgekommen werden muss und zudem materialisierte Sichten bereits verfügbar sind. Das System von MySQL verfügt zwar technisch bereits über viele Voraussetzungen für den Einsatz als Data-Warehouse-System, muss dafür aber vor allem beim analytischen SQL Fortschritte machen, um eine empfehlenswerte Alternative zu werden.

7 Zusammenfassung und Ausblick

Im abschließenden Kapitel werden nochmals die Vorgehensweise und Ergebnisse dieser Forschungsarbeit zusammengefasst sowie ein Ausblick auf zukünftige Entwicklungen im Umfeld der Data-Warehouse-Systeme getätigt.

Zu Beginn wurde eine Einführung in das Thema gegeben und dabei auch die Relevanz von DWHs in der heutigen Zeit erläutert. In diesem Rahmen wurden erst Grundlagen zum Thema DWH geklärt sowie die Begriffe ROLAP und MOLAP abgegrenzt. Darüber hinaus wurde die Betrachtungsweise auf ROLAP begrenzt. Zusätzlich wurden die beiden wichtigsten Schemata im DWH-Umfeld beschrieben. Es handelt sich dabei um das Star- und um das Snowflake Schema. Den Abschluss dieses Kapitels liefern dann verschiedene Studien und Statistiken, die Relevanz und Einsatzhäufigkeit der Data-Warehouse-Systeme in Unternehmen unterstreichen.

Anschließend wurden dann die Erweiterungen dargestellt, die in RDBMS eingesetzt werden, um den Betrieb als Data-Warehouse-System zu unterstützen. Große Punkte waren dabei vor allem die Partitionierung, die Indizierung und auch das analytische SQL. Aber auch kleinere Aspekte wie die materialisierten Sichten, die Rahmenbedingungen durch die Hardware, das Einfügen großer Datenmengen und die physische Organisation der RDBMS spielen eine Rolle. Diese verschiedenen Bereiche werden mit ihren wichtigsten theoretischen Gesichtspunkten und Vorteilen für Data-Warehouse-Systeme dargestellt.

Anhand dieser untersuchten technologischen Erweiterungen wurde dann ein Kriterienkatalog aufgesetzt, der verschiedene Techniken zu diesen Oberpunkten beinhalten sollte. Dazu ist zunächst ausgewählt worden, welche Systeme im Rahmen dieser Forschungsarbeit untersucht werden sollen. Die Entscheidung fiel dabei auf vier kommerzielle Systeme von Oracle, IBM, Microsoft und SAP sowie auf die zwei Open-Source-Produkte MySQL und PostgreSQL. Nachdem diese Auswahl feststand, wurden die Punkte des Kriterienkatalogs dahin gehend überprüft, ob die relevanten Erweiterungen in den Systemen implementiert sind. Die ausführliche Darstellung der gesammelten Ergebnisse befindet sich, aufgeschlüsselt nach den verschiedenen Kategorien, in Kapitel 4.2.

Im Anschluss an die Präsentation des gefüllten Kriterienkatalogs werden in Kapitel 5 Anwendungsbeispiele mit Hilfe von Code-Fragmenten dargestellt. Der Fokus liegt hierbei insbesondere auf dem analytischen SQL. Es finden sich Beispiele zum Star Join, dem Operator CUBE, zur sequenzorientierten Analyse, der Rangfolge und auch zu den dynamischen Fenstern. Hierfür wurden verschiedene Beispiel-Datenmodelle für DWHs aus der Literatur herangezogen, auf denen anschließend die entsprechenden SQL-Anfragen durchgeführt wurden.

Nach der Durchführung dieser Arbeiten wurde in Kap. 6 eine abschließende Beurteilung der vorhandenen Technologien in den untersuchten Systemen getätigt. Zunächst wurde dafür aufgezeigt, welche Erweiterungen in der Praxis noch nicht zum Einsatz kommen. Speziell komplexere Bitmap-Indexierungsarten wie der Mehrkomponenten-Bitmap-Index und der bereichskodierte Bitmap-Index sind noch nicht in den Systemen integriert. Darüber hinaus wird noch verglichen, welche Unterschiede sich zwischen den kommerziellen Systemen und den Open-Source-Produkten festzustellen sind.

Anhand der gesammelten Ergebnisse können folgende Aussagen für die Zukunft getroffen werden. Da DWHs in nahezu jedem Unternehmen im Einsatz sind, ist von allen Systemen zu erwarten, dass insbesondere die Technologien, die bisher noch keine praktische Implementierung aufweisen, in naher Zukunft in die verschiedenen Systeme integriert werden sollten. Außerdem wird es darum gehen, bereits implementierte Erweiterungen zu optimieren. Hierbei ist es noch unklar, wie technisch fortgeschritten die vorhandenen Technologien in den

Systemen eingebaut sind. Dies konnte im zeitlichen Rahmen dieser Forschungsarbeit nicht ausreichend untersucht werden. Folgeprojekte können daher einerseits untersuchen, ob bisher nur theoretisch bekannte Techniken wie die verschiedenen erweiterten Bitmap-Indizierungen inzwischen implementiert oder geplant sind. Darüber hinaus kann begutachtet werden, wie ausgereift die bereits integrierten Erweiterungen der RDBMS für Data-Warehouse-Systeme sind und ob dort Optimierungen seitens der Anbieter durchgeführt werden.

Zukünftige Entwicklungen im Bereich Data-Warehouse werden sich zudem mit den aufkommenden Themengebieten „unstrukturierte Daten“, worunter beispielsweise Bilder, Videos und Texte fallen, und dem bereits viel diskutierten Thema „Big Data“ beschäftigen. Hierfür werden geeignete Techniken benötigt, deren Entwicklung in nächster Zeit vorangetrieben werden muss.¹⁰⁰

Eine Möglichkeit zur Bewältigung der Big Data Problematik könnte beispielsweise in der Bildung mehrerer Data Warehouses anstatt der bisher vorwiegenden Vorgehensweise, eines zentralen, unternehmensweiten Data Warehouses liegen. Ansatzpunkte für technologische Entwicklungen bestehen dann vor allem im Bereich der optimalen Verknüpfung der verschiedenen Data Warehouses.

¹⁰⁰ vgl. Bayer, M. (2011)

Quellenverzeichnis

Literaturverzeichnis

- Azevedo, P. (u.a.) (2009): Business Intelligence und Reporting mit Microsoft SQL Server 2008, Unterschleißheim: Microsoft Press Deutschland
- Bauer, A. / Günzel, H. (2013): Data Warehouse Systeme – Architektur, Entwicklung, Anwendung, 4. Aufl., Hedelberg: dpunkt-Verlag
- Beyer, M. / Edjlali, R. (2014): Magic Quadrant for Data Warehouse Database Management Systems
- Farkisch, K. (2011): Data-Warehouse-Systeme kompakt – Aufbau, Architektur, Grundfunktionen, Heidelberg/Dordrecht/London/New York: Springer Verlag
- Gabriel, R. / Gluchowski, P. / Pastwa, A. (2009):
Data Warehouse & Data Mining, Herdecke/Witten: W3L GmbH
- Köppen, V. / Saake, G. / Sattler, K. (2012):
Data Warehouse Technologien, Heidelberg/München/Landsberg/Frechen/Hamburg: mitp Verlag
- Lehner, W (2003): Datenbanktechnologie für Data-Warehouse-Systeme – Konzepte und Methoden, Heidelberg: dpunkt-Verlag
- Mucksch, H. / Behme, K. (2000): Das Data Warehouse Konzept – Architektur – Datenmodelle – Anwendungen, 4. Aufl., Wiesbaden: Gabler Verlag
- Singh, S. (2009): Database Systems – Concepts, Design and Applications, Pearson Education India
- Tegel, T. (2005): Multidimensionale Konzepte zur Controllingunterstützung in kleinen und mittleren Unternehmen, Wiesbaden: Deutscher Universitäts-Verlag

Verzeichnis der Internet- und Intranet-Quellen

- Bayer, M. (2011): Business Intelligence – Es wird chaotisch – neue BI Ansätze sind gefragt, <http://www.computerwoche.de/a/business-intelligence-es-wird-chaotisch,2502157,2>, Abruf: 03.07.2014
- Codeproject (2009): Merge Statement in SQL Server 2008, <http://www.codeproject.com/Articles/37172/Merge-Statement-in-SQL-Server>, Abruf: 01.07.2014
- DB-Engines (2014): DB-Engines Ranking von Relational DBMS, <http://db-engines.com/de/ranking/relational+dbms>, Abruf: 26.05.2014
- FH Köln (2010): Materialisierte Sicht, http://wikis.gm.fh-koeln.de/wiki_db/Datenbanken/Materialisierte-Sicht, Abruf: 30.06.2014
- FH Köln (2012): Skalierbarkeit, http://wikis.gm.fh-koeln.de/wiki_db/Datenbanken/Skalierbarkeit, Abruf: 01.07.2014
- FH Köln (2013): SAP-HANA, http://wikis.gm.fh-koeln.de/wiki_db/Datenbanken/SAP-HANA, Abruf: 30.06.2014
- Gfaller, H. (2010): Data Warehouse Appliances – Trends und neue Techniken, <http://www.computerwoche.de/a/data-warehouse-appliances-trends-und-neue-techniken,1934843>, Abruf: 01.07.2014
- Hilker, S. (2013): Increase the Database Block Size, <http://www.toadworld.com/platforms/oracle/w/wiki/1373.increase-the-database-block-size.aspx>, Abruf: 30.06.14
- Kurzlechner, W. (2011): Massive Zweifel am Data Warehouse, <http://www.cio.de/knowledgecenter/bi/2287235/>, Abruf: 01.07.2014

- Oracle (2011): SQL Insert, Update und Delete in einem: Das MERGE-Kommando,
<http://www.oracle.com/webfolder/technetwork/de/community/apex/tipps/sql-merge/index.html>, Abruf: 01.07.2014
- Stokes, A. (o.J.) Definition of Bulk Loading,
http://www.ehow.com/about_5618573_definition-bulk-loading.html, Abruf 01.07.2014
- Universität Rostock (2001): R-Baum, <http://www.geoinformatik.uni-rostock.de/einzel.asp?ID=1466>, Abruf: 30.06.2014
- Wikipedia (o.J.): Bloomfilter, <http://de.wikipedia.org/wiki/Bloomfilter>, Abruf: 30.06.2014

Objektorientierte Datenverwaltung mit relationalen Datenbanken

Schriftliche Ausarbeitung
im Rahmen der Lehrveranstaltung „Integrationsseminar“

vorgelegt von

Daniel Weiß,
Diana Schery Acosta,
Matthias Koschar

am 14.07.2014

Fakultät Wirtschaft
Studiengang Wirtschaftsinformatik
WWI2011V

Inhaltsverzeichnis

Abkürzungsverzeichnis	IV
Abbildungsverzeichnis.....	V
Tabellenverzeichnis.....	VI
1 Einleitung.....	1
2 Grundlagen.....	2
2.1 Relationale Datenbanken.....	2
2.2 SQL.....	3
2.3 Objektrelationale Datenbanken	5
2.4 Objektrelationale Datenbanksysteme	6
2.4.1 Oracle.....	6
2.4.2 DB2	7
2.4.3 PostgreSQL.....	7
2.5 Objektrelationale Statements nach dem SQL Standard.....	8
2.5.1 Benutzerdefinierte Datentypen	8
2.5.1.1 Distinct-Datentyp	9
2.5.1.2 Strukturierte Datentypen.....	10
2.5.1.2.1 Strukturierte Attribute.....	10
2.5.1.2.2 Typisierte Tabellen.....	11
2.5.1.2.3 Vererbung in Typ- und Tabellenhierarchien.....	12
2.5.1.3 ROW-Datentyp	18
2.5.1.4 Array-Datentyp	18
2.5.1.5 CLOB- und BLOB-Datentypen.....	20
2.5.2 Benutzerdefinierte Routinen (Methoden)	20
2.5.3 XML-Dokumente.....	22
3 Gang der Untersuchung der SQL-Statements.....	24
3.1 Systeminstallation und Ersteinrichtung.....	24
3.1.1 Oracle - Installation und Inbetriebnahme	24
3.1.2 DB2 - Installation und Inbetriebnahme	26
3.1.3 PostgreSQL - Installation und Inbetriebnahme.....	27
3.2 Vorgehen bei der Statementuntersuchung	29
3.3 Statementuntersuchung Oracle.....	30
3.3.1 Benutzerdefinierte Datentypen	30
3.3.1.1 Distinct-Datentyp	30
3.3.1.2 Strukturierter Datentyp.....	31
3.3.1.2.1 Strukturierte Attribute.....	30

3.3.1.2.2	Vererbung in Typ- und Tabellenhierarchien.....	32
3.3.2	XML-Dokumente.....	34
3.4	Statementuntersuchung DB2	35
3.4.1	Benutzerdefinierte Datentypen	35
3.4.1.1	Distinct-Datentyp	35
3.4.1.2	Strukturierter Datentyp.....	36
3.4.1.2.1	Strukturierte Attribute.....	36
3.4.1.2.2	Typisierte Tabellen.....	37
3.4.1.2.3	Vererbung in Typ- und Tabellenhierarchien.....	38
3.4.1.3	CLOB- und BLOB-Datentyp.....	41
3.4.2	XML-Dokumente.....	41
3.5	Statementuntersuchung PostgreSQL.....	43
3.5.1	Benutzerdefinierte Datentypen	43
3.5.1.1	Distinct-Datentyp	43
3.5.1.2	Strukturierter Datentyp.....	44
3.5.1.2.1	Strukturierte Attribute.....	43
3.5.1.2.2	Typisierte Tabellen.....	45
3.5.1.3	Array-Datentyp	47
3.5.1.4	CLOB- und BLOB-Datentyp.....	48
3.5.2	XML-Dokumente.....	48
4	Schlussbetrachtung und Ausblick	50
	Anhang.....	52
	Quellenverzeichnisse	74

Abkürzungsverzeichnis

ANSI	American N ational S tandards I nstitute
BLOB	B inary L arge O bject
CLOB	C haracter L arge O bject
DBMS	D aten b ank m anagement s ystem
DBS	D aten b ank s ystem
DCL	D ata C ontrol L anguage
DDL	D ata D efinition L anguage
DML	D ata M anipulation L anguage
IBM	International B usiness M achines
ISO	International O rganization for S tandardization
OID	O bjekt i dentität
PSM	P ersistent S tored M odules
SQL	S tructured Q uery L anguage
XML	E xtensible M arkup L anguage

Abbildungsverzeichnis

Abb. 1: Zusammensetzung objektrelationaler Datenbanken.....	5
Abb. 2: Beispielhaftes XML-Dokument	22
Abb. 3: Get Started in Oracle	25
Abb. 4: Neue Datenbankverbindung anlegen in Oracle	26
Abb. 5: Ersteinrichtung im IBM Data Studio für DB2.....	27
Abb. 6: Verbindungsaufbau zum PostgreSQL-Server.....	28
Abb. 7: Vorgehen der Statementuntersuchung.....	29
Abb. 8: Vorgehensweise des Selbststudiums	29

Tabellenverzeichnis

Tab. 1: SQL-Anweisungsgruppen.....	3
Tab. 2: Vergleichsoperatoren in SQL	4

1 Einleitung

Seit vielen Jahren werden zur Datenverwaltung in Unternehmen standardmäßig relationale Datenbanksysteme (DBS) eingesetzt. Diese eignen sich für die Verarbeitung von einfachen strukturierten Datensätzen, sind aber nicht primär auf die Verarbeitung von objektorientierten Daten ausgelegt. Es ist schwierig eine objektorientierte Programmiersprache, wie z. B. Java mit einem relationalen DBS zu verknüpfen. Daher wären objektorientierte Eigenschaften, wie beispielsweise benutzerdefinierte Datentypen, Routinen oder Verwaltung von Extensible Markup Language (XML)-Dokumenten wünschenswert. Diese sind im Structured Query Language (SQL)-Standard definiert und lassen sich grundsätzlich auf einigen relationalen DBS umsetzen.

Problemstellung

Dafür gilt es jedoch einige Dinge zu beachten, da in keinem der DBS die im SQL-Standard definierten Anweisungen genau in der gleichen Syntax umgesetzt wurden. Es erfordert einen hohen Aufwand herauszufinden, welche Eigenschaften überhaupt unterstützt werden und über welche Syntax dies umgesetzt werden kann.

Zielsetzung

Das Ziel dieser Arbeit ist es drei repräsentative relationale DBS (Oracle, DB2 und PostgreSQL) zu installieren und die Abweichungen der Syntax dieser Systeme zum SQL-Standard aufzuzeigen. Jede Abweichung der Syntax soll dabei dokumentiert werden.

Nutzen

Den Nutzen dieser Arbeit beziehen in erster Linie alle Personen und Unternehmen, die die objektorientierten Eigenschaften in einem relationalen DBS nutzen möchten, aber nicht wissen, mit welcher Syntax zu arbeiten ist. Die vorliegende Arbeit ist somit als eine Hilfestellung zu verstehen, die dem SQL-Benutzer an die Hand gegeben werden kann.

Aufbaustruktur

Zu Beginn werden die theoretischen Grundlagen erläutert, die für das weitere Verständnis wichtig sind. Dafür wird ein Überblick über relationale DBS gegeben, bevor auf die dafür verwendete Abfragesprache eingegangen wird. Es folgt die Darstellung des Grundgedankens der objektorientierten Datenverwaltung. Dabei werden relationale DBS aufgeführt, die die objektorientierten Eigenschaften unterstützen. Nachdem die hierfür im Standard definierten objektrelationalen Statements aufgeführt und erläutert wurden, beginnt der praktische Teil der Arbeit. Die Installation und Einrichtung der relationalen DBS wird beschrieben, woraufhin die Untersuchung vorgenommen wird. Nach DBS getrennt werden die erzielten Resultate erläutert. Eine Zusammenfassung der wesentlichen Ergebnisse, eine abschließende Bewertung, sowie ein Ausblick in die Zukunft runden diese Arbeit ab.

2 Grundlagen

2.1 Relationale Datenbanken

Relationale DBS basieren auf einem relationalen Datenmodell. Dieses wurde erstmals im Jahre 1970 von E. F. Codd publiziert. Unter einer Relation wird die „Eigenschaft einer Datenbank [verstanden], bei der sich Datenbestände durch logische Beziehungen miteinander verknüpfen lassen.“¹ Bei relationalen Datenbanken werden die Daten in Tabellen gespeichert. Eine Tabelle stellt eine Datensammlung dar, welche in Spalten und Zeilen gegliedert ist.² Sie zeichnen sich dabei unter anderem durch folgende Merkmale aus:

- Tabellennamen
- Feste Anzahl von Spalten
- Beliebige Anzahl von Zeilen
- Spaltenname im Kopf jeder Spalte
- Jede Zeile der Tabelle entspricht einem Datensatz³

Ein DBS besteht aus einem Datenbankmanagementsystem (DBMS) sowie aus einer oder mehreren Datenbanken. Eine Datenbank stellt hierbei eine Sammlung von strukturierten, inhaltlich zusammengehörenden Daten dar.⁴ Ein DBMS beinhaltet mehrere Programme zur Abfrage, Änderung, Löschung und zum Aufbau von Daten einer Datenbank. Relationale DBS wie bspw. Oracle, DB2 und PostgreSQL unterstützen das Relationenmodell. Die Anwender eines solchen relationalen DBS können über die Abfragesprache SQL direkt mit einer Datenbank kommunizieren. Ein Kommandoprozessor interpretiert die gestellten SQL-Anfragen eines Anwenders an die Datenbank. Der Kommandoprozessor verfügt hierzu über Funktionen, welche die SQL-Anfragen verarbeiten und durchführen. Die relationalen DBS halten häufig verwendete Daten im Hauptspeicher, um die Festplattenzugriffe möglichst gering zu halten. Zur Gewährleistung konsistenter Daten werden die Datenbankzugriffe stets überwacht und protokolliert. Dies ermöglicht die Wiederherstellung der Daten selbst nach Programmabbrüchen oder sonstigen Störungen.⁵

DBS müssen insbesondere bestimmte Anforderungen erfüllen, diese werden in Anhang 1 aufgelistet. Darüber hinaus werden in Anhang 2 die Anforderungen an ein relationales DBS dargestellt.

¹ Webhostings.ch (2011)

² Vgl. IBM (2000)

³ Vgl. Abts, D./Mülder, W. (2010), S. 605

⁴ Vgl. ebenda, S. 614

⁵ Vgl. ebenda

2.2 SQL

Die Abkürzung SQL steht für Structured Query Language und stellt die am weitesten verbreitete Datenbanksprache für relationale Datenbanken dar.⁶ Sie entstand aus der 1975 von International Business Machines (IBM) entwickelten Datenbank-Abfragesprache Structured English Query Language. Seit 1986 wurde SQL durch das ANSI (American National Standards Institute) standardisiert. Im Jahre 1987 wurde dieser Standard ebenfalls von der International Organization for Standardization (ISO) aufgegriffen.⁷ Seitdem ist der Sprachumfang von SQL einer kontinuierlichen Weiterentwicklung und Standardisierung unterworfen. Mit dem Standard SQL-1999 gibt es einen ersten Standard für objektrelationale Datenbanken. Die Integration von XML in SQL sieht der Standard SQL-2003 vor.⁸ Der aktuelle Standard stammt aus dem Jahr 2011 und stellt den ANSI/ISO SQL-2011 Standard dar.⁹ Alle großen DBS, die sich derzeit auf dem Markt befinden, realisieren die im Standard definierten Konzepte größtenteils.¹⁰

SQL ist nicht nur eine Abfragesprache, sondern sie umfasst neben der klassischen SELECT-Anweisung insgesamt drei Anweisungsgruppen, welche in Tab. 1 dargestellt sind.

Anweisungsgruppen in SQL	Beschreibung
DML (Data Manipulation Language)	Wird für die Änderung von Daten, sowie zum Einfügen, Ändern und Löschung von Daten verwendet. DML umfasst Befehle, wie bspw. SELECT, Insert, Update und Delete.
DDL (Data Definition Language)	Wird für die Verwaltung von Datenbankobjekten wie bspw. Tabellen, Sichten, Indizes aber auch für Datenbanken selbst verwendet. DDL beinhaltet diverse Befehle, wie bspw. Create, Alter und Drop.
DCL (Data Control Language)	Wird für die Verwaltung von Benutzern, Rollen und Zugriffsrechten verwendet. DCL bietet Befehle, wie bspw. Grant und Deny.

Tab. 1: SQL-Anweisungsgruppen¹¹

⁶ Vgl. Uni Freiburg (2005)

⁷ Vgl. Panther, R. (2012), S. 113

⁸ Vgl. Türker, Can (2003), S.1

⁹ Vgl. Cordts, S./Blakowski, G./Brosius, G. (2011), S. 97 f.

¹⁰ Vgl. Uni Freiburg (2005)

¹¹ Vgl. Panther, R. (2012), S. 114;
vgl. dazu auch Throll, M./Bartosch, O. (2007);

Bevor eine Tabelle durch die DDL in einer Datenbank erstellt werden kann, muss für jede Spalte dieser Tabelle ein Name sowie ein Datentyp definiert werden, wie bspw. Char, Var-char, Integer, Decimal und Date. Für die Erstellung einer Tabelle wird die Anweisung ‚Create Table‘ verwendet. Außerdem muss der Primärschlüssel und eventuelle Fremdschlüssel festgelegt werden. Die Anweisung ‚Alter Table‘ ändert die Definition der Tabelle, während ‚Drop Table‘ die Tabelle mit den zugehörigen Daten aus der Datenbank entfernt. Darüber hinaus gibt es in der DML die Anweisungen ‚SELECT‘, ‚Insert‘, ‚Update‘ und ‚Delete‘. Mit SELECT werden Abfragen durchgeführt. Durch Insert werden neue Daten in eine Tabelle eingefügt, welche sich durch die Update-Anweisung jederzeit verändern lassen. Mit der Delete-Anweisung können einzelne Datensätze gelöscht werden.¹² Überdies gibt es noch die DCL mit Hilfe derer sich Benutzer, Rollen und Zugriffsrechte verwalten lassen. Die WHERE-Bedingung spielt bei SQL-Abfragen eine sehr wichtige Rolle. Sie gibt Bedingungen an, welche Zeilen einer Tabelle selektiert werden sollen. Dabei dienen für die Formulierung der Auswahlbedingung unterschiedliche Operatoren.¹³ Mögliche Vergleichsoperatoren sind in nachfolgender Tab. 2 **Fehler! Verweisquelle konnte nicht gefunden werden.** dargestellt.

Befehl	Bedeutung	Befehl	Bedeutung
=	Ist gleich	LIKE	Ist einem angegebenen Wert ähnlich
<	Ist kleiner als	IS NULL	Ist nicht gefüllt
>	Ist größer als	IS NOT NULL	Ist gefüllt (nicht ungefüllt)
<=	Ist kleiner oder gleich	IN	Ist in einer Liste von Werten
>=	Ist größer oder gleich	BETWEEN	Bereichsvergleich
<>	Ist ungleich		

Tab. 2: Vergleichsoperatoren in SQL

vgl. dazu auch Geisler, F. (2011), S. 216
¹² Vgl. Panther, R. (2012), S. 620 f.
¹³ Vgl. Abst, , D./Mülder, W. (2010), S. 623

2.3 Objektrelationale Datenbanken

Die relationale DBS haben einen hohen Entwicklungszustand und werden daher als ‚State of the Art‘ bezeichnet. Es stellte sich heraus, dass die sehr einfache Strukturierung der Daten in flachen Relationen in administrativen Anwendungsbereichen benutzerfreundlich ist.¹⁴ Allerdings wurden in den 80er Jahre die Unzulänglichkeiten des relationalen Datenmodells für komplexe Anwendungsbereiche, wie bspw. ingenieurwissenschaftliche Entwurfsanwendungen entdeckt.¹⁵ Für diese Anwendungsbereiche galt es eine geeignete Lösung zu finden.

Eine objektrelationale Datenbank stellt eine Kombination von einer relationalen Datenbank mit Elementen der objektorientierten Betrachtungsweise dar. Dies ist in Abb. 1 vereinfacht dargestellt.

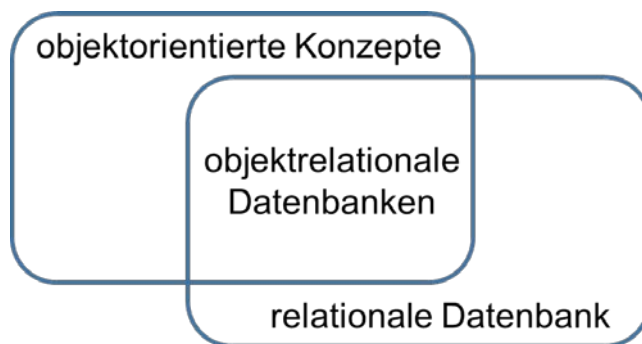


Abb. 1: Zusammensetzung objektrelationaler Datenbanken

Das Ziel der Entwicklung objektrelationaler Datenbanken war, die Vorteile zur Speicherung komplexer Objekte von objektorientierter Datenbanken unter gleichzeitiger Verwendung der standardisierten Abfragesprache SQL zu nutzen. Durch objektrelationale Datenbanken können somit Datenobjekte und Operationen höherer Komplexität unterstützt und wesentlich schneller und effizienter bearbeitet werden, da die Verknüpfungen der Datenbanken in Form von Objekten erfolgen. Unter Datenobjekte werden hierbei vor allem geographische Karten sowie Text-, Bild- und Audiodateien verstanden, deren Verwaltung bei relationalen Datenbanken entweder überhaupt nicht oder nur mit erhöhtem Aufwand funktioniert. Der Grund für die Realisierung von objektrelationalen DBS waren die gestiegenen Ansprüche an die Verwaltung großer Datenmengen. „Im objektrelationalen Modell werden die Daten als Relationen von komplexen Objekten gespeichert. Wiederum können über Referenzen Objekte verkettet werden und auch Relationen in Untermengenbeziehungen zueinander stehen.“¹⁶ Hierfür werden in objektrelationalen Datenbanken Operationen benötigt, mit deren Hilfe es möglich ist, Relationshierarchien bearbeiten zu können. Zu der höheren strukturellen Komplexität kommt somit zugleich eine erhöhte funktionelle Komplexität hinzu. Um komplexe Sachver-

¹⁴ Vgl. Kemper, A./Eickler, A. (2011), S. 381

¹⁵ Vgl. ebenda

¹⁶ ITWissen (2014)

halte auch abbilden zu können, müssen aufbauend auf die elementaren Datentypen relationaler Datenbanken Datentypen für komplexe Daten definiert werden. Hierfür werden Typ-Konstrukte wie bspw. Tupel, Mengen und Listen verwendet. In objektrelationalen Datenmodellen stellen die strukturierten Datentypen ein zentrales Konzept dar. „Die strukturierten Typen erfüllen die Anforderungen der Objektorientierung beispielsweise hinsichtlich der Strukturierung der Daten, der Vererbung von Attributen und Methoden oder des Prinzips der Kapselung.“¹⁷ Klassen werden dagegen durch typisierte Tabellen abgebildet, die zur Speicherung von Daten dienen und wiederum auf strukturierten Datentypen basieren.¹⁸ Das hinzuzufügende Schlüsselattribut in einer relationalen Modellierung wird durch die Objektidentität (OID) identifiziert. Hierbei kann jedes Objekt über seine Identität eindeutig referenziert werden.¹⁹

2.4 Objektrelationale Datenbanksysteme

Für die objektrelationale Datenverwaltung existieren spezielle DBS. Im Rahmen dieser Seminararbeit liegt der Fokus auf den drei objektrelationalen DBS Oracle, DB2 und PostgreSQL, da sich diese im Bereich der objektrelationalen Erweiterungen besonders auszeichnen sowie vom Auftragsgeber vorgegeben wurden. Diese Systeme werden anschließend näher beschrieben.

2.4.1 Oracle

„Der Ursprung des Oracle DBS geht auf die Entwicklung des Datenbank-Prototypensystem R von Edgar F. Codd im Jahre 1970 zurück. Larry Ellison entwickelte diesen Prototypen weiter zur Datenbank mit dem Namen Oracle.“²⁰

Eine Oracle-Datenbank setzt sich aus diversen Datenbankdateien zusammen, welche sich auf einem Datenbankserver befinden. Hierbei wird eine Instanz erstellt, die zur Interaktion mit der Datenbank dient.²¹ Oracle gibt es in verschiedenen Versionen. Jede Edition bietet hierbei einen unterschiedlichen Funktionsumfang an.²² Das Oracle DBS wird auch Oracle Database Server genannt. Hierbei handelt es sich um eine DBMS-Software des Unternehmens Oracle. Das DBS kann in Form einer Express-Edition kostenlos verwendet werden. Hier besteht allerdings der Nachteil, dass die Datenbank das Laden von Java-Klassen nicht unterstützt. Oracle ermöglicht Zugriff auf ein Datenvolumen bis zu einer Größe von 40 Exbi-

¹⁷ ITWissen (2014)

¹⁸ Vgl. ebenda

¹⁹ Vgl. Kemper, A./Eickler, A. (2011), S. 385 f.

²⁰ Extrememarkup.com (o. J.)

²¹ Vgl. Adar, M. (2013), S. 9

²² Vgl. ebenda, S. 172

byte, sowie die Speicherung von XML-Datenstrukturen und nichtrelationalen Daten wie bspw. Videos, Musik und Dokumente.²³

Das DBMS von Oracle besteht nicht nur aus einer bzw. mehrerer Oracle-Instanzen, sondern auch aus einem bzw. mehreren Listener-Prozessen, sowie aus einem oder mehreren Oracle-Server-Prozessen. Ein Listener-Prozess dient u.a. der Verbindungsaufnahme zu einer Datenbank. Die Oracle-Server-Prozesse nehmen entweder Datenbankabfragen oder Datenmanipulationsanweisungen von den Datenbank-Clients entgegen, führen diese anschließend als Aufträge aus und liefern den Clients die Ergebnisdaten zurück.²⁴

2.4.2 DB2

Das relationale DBS DB2 wurde ebenfalls in den 80er Jahren entwickelt. Das damalige Forschungsprojekt unter dem Namen „System R“, welches auf die von Edgar F. Codd publizierten Inhalte zu relationalen Datenmodellen aufbaute, entwickelte sich zum ersten relationalen DBS. Dieses wurde ab 1982 unter dem Namen ‚SQL/DS‘ von der IBM vertrieben. Zahlreiche Weiterentwicklungen führten zum Entstehen des heutigen DB2. Ursprünglich erhielt das System den Namen Database 2, bis 1993 die Vermarktung von DB2 begonnen wurde. Heutzutage ist DB2 ein System, welches unter den gängigen Betriebssystemen läuft.²⁵

Das DBS IBM DB2 Express Edition ist ein budgetfreundliches System, welches daher besonders für den Einstieg geeignet ist. Die Anwendung ist in einer Vielzahl von Lizenzierungsmodellen erhältlich, so gibt es auch eine kostenfreie Version, welche hauptsächlich für den Privatgebrauch vorgesehen ist.²⁶

2.4.3 PostgreSQL

„PostgreSQL ist ein lizenzkostenfrei verfügbares [objekt]relationales open source DBS“²⁷ und kann über die Webseite ‚<http://www.postgresql.org>‘ heruntergeladen werden. Der Ursprung von PostgreSQL lag in einem universitären Projekt an der University of California at Berkeley Computer Science Department in den 80er Jahren. Das DBS wurde unter dem Namen POSTGRES entwickelt. Seit 1996 wird das System von einer Open Source Community unter dem Namen PostgreSQL weiterentwickelt.²⁸

²³ Vgl. Extrememarkup.com (o. J.)

²⁴ Vgl. ebenda

²⁵ Vgl. Orhanovic, J./Grodtko, I./Tiefenbacher, M. (2007), S. 25

²⁶ Vgl. IBM (o. J.)

²⁷ Kleinschmidt, P./Rank, C. (2005), S.205

²⁸ Vgl. PostgreSQL (o. J.)

Als objektrelationales DBS sind in PostgreSQL die Speicherung von nicht atomarer Daten, die Vererbung und die Objektidentitäten implementiert. Zudem können die Anwender eigene Datentypen, Funktionen und Operatoren definieren.²⁹

PostgreSQL ist nach dem Prinzip der Client-Server-Architektur aufgebaut. „Ein Serverprozess verwaltet die Datenbankdateien sowie die Verbindungen, die von Client-Programmen zum Server aufgebaut werden und bearbeitet die Anfragen, die von diesen Client-Programmen gestellt wurden.“³⁰ Bei PostgreSQL werden die Aufgaben von dem Serverprogramm ‚postmaster‘ übernommen. Die Kommunikation mit dem postmaster kann durch die Verwendung von unterschiedlichen Client-Programmen geführt werden. Ein solches Programm stellt die Verbindung zum Datenbankserver her und führt anschließend Aktionen in der Datenbank aus.³¹

2.5 Objektrelationale Statements nach dem SQL Standard

In der Praxis existieren deutliche Abweichungen bei der Umsetzung der objektrelationalen Datenbankerweiterungen nach dem SQL-Standard auf den erwähnten DBS. Im Folgenden werden die objektrelationale Erweiterungen anhand von beispielhaften SQL-Statements aufgeführt.³² Diese Statements wurden nach dem aktuellen SQL-Standard erstellt und lassen sich wie in Anhang 3 visualisiert gliedern.

2.5.1 Benutzerdefinierte Datentypen

Benutzerdefinierte Datentypen lassen sich in insgesamt fünf Datentypen (Distinct, strukturiert, ROW, Array und CLOB/BLOB) unterteilen. Die einzelnen Datentypen werden nachfolgend im Detail erläutert, wobei dazu beispielhafte SQL-Statements aufgezeigt werden.

²⁹ Vgl. PostgreSQL (o. J.)

³⁰ Ebenda

³¹ Vgl. ebenda

³² Vgl. Preiß, N. (2014)

2.5.1.1 Distinct-Datentyp

Distinct-Datentypen basieren auf bereits definierten Typen und stellen im Grunde genommen lediglich eine Umbenennung dar.

```
CREATE TYPE LebAlterTyp AS INTEGER FINAL;
```

Bei diesem Statement wird ein benutzerdefinierter Distinct-Datentyp vom Typ Integer mit der Bezeichnung ‚LebAlterTyp‘ angelegt. Die Angabe der Klausel ‚FINAL‘ ist hierbei obligatorisch. Diese sagt aus, dass ein Distinct-Datentyp keine Subtypen besitzen kann.

```
CREATE TYPE GewichtTyp AS INTEGER FINAL;
```

Im Anschluss daran wird noch ein zweiter Datentyp ‚GewichtTyp‘ angelegt. Dieser ist ebenfalls vom Typ Integer.

```
CREATE TABLE Personen1  
( PersID INTEGER PRIMARY KEY,  
  LebAlter LebAlterTyp,  
  Gewicht GewichtTyp);
```

Nachdem die beiden Distinct-Datentypen angelegt wurden, wird eine Tabelle mit der Bezeichnung ‚Personen1‘ erstellt. Die Tabelle hat insgesamt drei Attribute. Eine ‚PersID‘ vom Typ Integer, welche zugleich den Primärschlüssel der Tabelle darstellt, ein ‚LebAlter‘ vom Typ LebAlterTyp und ein ‚Gewicht‘ vom Typ GewichtTyp.

```
SELECT *  
FROM Personen1  
WHERE Gewicht > LebAlter;
```

In die zuvor angelegte Tabelle Personen1 wurden inzwischen Beispieldatensätze eingefügt, sodass in einem nächsten Schritt alle Datensätze der Tabelle ‚Personen1‘ ausgegeben werden, bei denen das Gewicht größer ist als das Lebensalter. Der Vergleich der beiden Attribute geschieht hierbei über einen Vergleichsoperator. Laut Standard ist ein solcher Vergleich aufgrund unterschiedlicher Datentypen allerdings nicht zulässig und dürfte somit nicht funktionieren.

```
SELECT *  
FROM Personen1  
WHERE Gewicht > CAST (LebAlter AS GewichtTyp);
```

Dieses Statement soll letztendlich das Ergebnis des vorherigen Vergleichs liefern. Der Unterschied zum vorherigen Statement besteht darin, dass der Typ des Attributs ‚LebAlter‘ in den Typ GewichtTyp konvertiert wird, um beide miteinander vergleichen zu können.

2.5.1.2 Strukturierte Datentypen

Strukturierte Datentypen stellen eine Struktur dar und bestehen aus mehreren Attributen eines vordefinierten oder benutzerdefinierten Typs. Zudem sind diese Datentypen stets gekapselt. Das bedeutet, dass der Zugriff auf die jeweiligen Attribute lediglich über spezielle Methoden möglich ist.

2.5.1.2.1 Strukturierte Attribute

```
CREATE TYPE GeldTyp AS DECIMAL (10,2) FINAL;
```

Zunächst wird erneut ein benutzerdefinierter Distinct-Datentyp vom Typ Decimal mit der Bezeichnung ‚Geldtyp‘ angelegt. Der Wertebereich reicht hierbei von -99.999.999,99 bis + 99.999.999,99.

```
CREATE TYPE AdressTyp AS  
( Straße VARCHAR (30),  
  HausNr INTEGER,  
  PLZ DECIMAL (5), Stadt VARCHAR (20) DEFAULT ‚Mannheim‘ )  
NOT FINAL;
```

Anschließend wird ein strukturierter Datentyp mit der Bezeichnung ‚AdressTyp‘ angelegt. Der Datentyp besteht aus insgesamt vier Attributen. Einer ‚Straße‘ vom Typ Varchar mit maximal 30 Zeichen, eine ‚HausNr‘ vom Typ Integer, einer ‚PLZ‘ vom Typ Decimal mit maximal 5 Zahlen und einer ‚Stadt‘ mit maximal 20 Zeichen. Das Attribut Stadt soll zudem mit dem Standardwert Mannheim vorbelegt sein. Die Angabe der Klausel ‚NOT FINAL‘ ist hierbei obligatorisch. Sie sagt aus, dass ein strukturierter Datentyp stets Subtypen besitzen kann.

```
CREATE TABLE Angestellte1  
( PersNr INTEGER PRIMARY KEY,  
  Name VARCHAR (40),  
  Adresse AdressTyp,  
  Gehalt GeldTyp );
```

Nach der Erstellung zweier Datentypen wird eine Tabelle mit der Bezeichnung ‚Angestellte1‘ angelegt. Die Tabelle besitzt dabei insgesamt vier Attribute. Zum einen das Attribut ‚PersNr‘ vom Typ Integer, welches zugleich der Primärschlüssel darstellt. Außerdem ein Attribut mit der Bezeichnung ‚Name‘ vom Datentyp Varchar und einer maximalen Länge von 40 Zeichen, sowie die Attribute ‚Adresse‘ vom Typ Adresstyp und ‚Gehalt‘ vom Typ Geldtyp.

```

INSERT INTO Angestellte1
VALUES
( 3333,
'Franz Groß',
AdressTyp ().Straße ('Weitweg').HausNr (23).PLZ (68165),
CAST (8650 AS GeldTyp));

```

Die zuvor angelegte Tabelle Angestellte1 soll nun mit Werten befüllt werden. Die PersNr soll ,3333' sein und der Name ,Franz Groß'. Bei der Adresse müssen alle Attribute des strukturierten Datentyps AdressTyp angegeben werden. Die Straße soll den Wert ,Weitweg', die HausNr ,23' und die PLZ den Wert ,68165' zugewiesen werden. Die Stadt entspricht hierbei dem Default-Wert ,Mannheim'. Das Gehalt soll den Wert ,8650' bekommen. Dieser numerische Wert soll allerdings zunächst in den Datentyp GeldTyp konvertiert werden.

```

SELECT Adresse
FROM Angestellte1
WHERE PersNr = 3333;

```

Nachdem die Tabelle mit Inhalten befüllt wurde, soll eine Abfrage durchgeführt werden. Es soll der Datensatz ausgegeben werden, bei dem die Personalnummer den Wert ,3333' entspricht.

```

SELECT Adresse.PLZ ()
FROM Angestellte1
WHERE PersNr = 3333;

```

Bei dieser Abfrage ist lediglich die Postleitzahl des Datensatzes auszugeben, bei welcher das Attribut ,PersNr' den Wert ,3333' enthält.

```

UPDATE Angestellte1
SET Adresse = Adresse.HausNr (77)
WHERE PersNr = 3333;

```

Zuletzt soll der Datensatz mit der Personalnummer ,3333' aktualisiert werden. Dabei erfolgt eine Abänderung der Hausnummer auf den Wert ,77'.

2.5.1.2.2 Typisierte Tabellen

Unter einer typisierten Tabelle wird eine Tabelle verstanden, „deren Schema durch einen strukturierten Typ festgelegt wird“³³ Jedes Attribut des zugrundeliegenden strukturierten Datentyps spiegelt eine Spalte der Tabelle wieder. Um einen Datensatz innerhalb einer Tabelle eindeutig identifizieren zu können, gibt es das Attribut ,OID'.

³³ Preiß, N. (2014)

Laut dem SQL-Standard gelten folgende Eigenschaften für eine solche OID-Spalte:

- Unveränderbar
- Der Wert darf nicht NULL sein
- Der Wert muss UNIQUE (eindeutig) sein

```
CREATE TYPE AbtTyp AS
( AbtNr INTEGER,
  AbtName VARCHAR (20),
  Budget GeldTyp)
NOT FINAL
REF IS SYSTEM GENERATED;
```

Bei diesem Statement wird ein strukturierter Datentyp mit der Bezeichnung ‚AbtTyp‘ erstellt. Der Datentyp besteht hierbei aus den Attributen AbtNr (Integer), AbtName (Varchar) und Budget welchem der Distinct-Datentyp GeldTyp zugewiesen ist. Die Ergänzung ‚REF IS SYSTEM GENERATED‘ gibt an, dass die OID vom System erzeugt wird. Die OID ist nur innerhalb der Tabellenhierarchie eindeutig.³⁴

```
CREATE TYPE AngestTyp AS
( PersNr INTEGER,
  PersName VARCHAR (40),
  Gehalt GeldTyp,
  AbtRef REF (AbtTyp))
NOT FINAL
REF USING INTEGER;
```

Anschließend wird noch ein weiterer strukturierter Datentyp mit der Bezeichnung ‚AngestTyp‘ definiert, welcher die Attribute PersNr (Integer), PersName (Varchar), Gehalt (GeldTyp) und AbtRef enthält. AbtRef stellt einen Pointer bzw. Verweis auf den strukturierten Datentyp AbtTyp dar. Bei diesem Datentyp wird die OID-Spalte allerdings nicht vom System generiert, sondern durch den Benutzer festgelegt. Die Werte der OIDs sind hierbei vom Typ Integer. Die OID-Werte müssen vom Benutzer beim Anlegen eines Objektes angegeben werden.

```
CREATE TABLE Abteilungen OF AbtTyp
( PRIMARY KEY (AbtNr),
  REF IS AbtOID SYSTEM GENERATED);
```

Nach Erstellung der beiden Datentypen wird eine Tabelle mit dem Namen ‚Abteilungen‘ vom Typ AbtTyp angelegt. Als Primärschlüssel wird das Attribut AbtNr festgelegt. Die OID-Spalte wird vom System generiert.

³⁴ Vgl. Datenbanken Online Lexikon (2010)

```
CREATE TABLE Angestellte OF AngestTyp
( PersNr WITH OPTIONS PRIMARY KEY,
  REF IS AngOID USER GENERATED,
  AbtRef WITH OPTIONS SCOPE Abteilungen );
```

Darüber hinaus wird noch eine weitere Tabelle ‚Angestellte‘ vom Typ AngestTyp erstellt. Die PersNr wird als Primärschlüssel festgelegt. Die OID-Werte müssen hierbei wieder manuell vom Nutzer vergeben werden. Das Attribut AbtRef darf zudem nur Abteilungsobjekte referenzieren.

```
INSERT INTO Abteilungen (AbtNr, AbtName, Budget)
VALUES
(124, ‚Forschung‘, CAST(5000000 AS GeldTyp));
```

Anschließend wird ein Datensatz in die Tabelle Abteilungen eingefügt. Der Wert für das Budget muss allerdings zunächst in den Typ GeldTyp umgewandelt werden bzgl. der Typenverträglichkeit. Die OID muss nicht angegeben werden, da diese systemgeneriert erzeugt wird.

```
INSERT INTO Angestellte (AngOID, PersNr, PersName, Gehalt, AbtRef)
VALUES
( CAST (314159 AS REF (AngestTyp)),
  1112,
  ‚Willi Wurm‘,
  CAST (8300 AS GeldTyp),
  ( SELECT AbtOID
    FROM Abteilungen
    WHERE AbtNr = 124));
```

Im Anschluss daran wird ebenfalls ein Datensatz in die Tabelle Angestellte hinzugefügt. Da bei dieser Tabelle die OID benutzergeneriert vergeben werden muss, ist eine Angabe des Wertes durch den Benutzer zwangsläufig notwendig. Der Wert muss hierbei in den Datentyp AngestTyp umgewandelt werden. Zur Ermittlung des Wertes für das Attribut AbtRef ist eine Unterabfrage auf die OID der Abteilung mit der Abteilungsnummer 124 notwendig.

```
SELECT *
FROM Abteilungen
WHERE AbtNr = 124;
```

Nachdem die Datensätze in die jeweiligen Tabellen eingefügt wurden, erfolgt eine Abfrage auf alle Attribute der Tabelle Abteilungen, die die Abteilungsnummer 124 haben.

```
SELECT Deref (AbtRef)
FROM Angestellte
WHERE PersNr = 1112;
```

Bei einer weiteren Abfrage wird ein komplettes Abteilungsobjekt mit der PersNr 1112 der Tabelle Angestellte zurückgeliefert. Die Funktion Deref bewirkt hierbei für die Dereferenzierung einer Referenz.

```
SELECT Deref (AbtRef).AbtName
FROM Angestellte
WHERE PersNr = 1112;
```

Bei dieser Abfrage wird lediglich der Abteilungsname des zurückgelieferten Abteilungsobjekts abgefragt.

```
SELECT PersName
FROM Angestellte
WHERE AbtRef->AbtNr = 124;
```

In einer weiteren Abfrage wird der Personennamenname der Tabelle Angestellte abgefragt, welcher der Abteilung 124 zugeordnet ist. Hierfür ist eine Dereferenzierung notwendig um auf die AbtNr des Abteilungsobjekts abfragen zu können. Anders als bei der vorletzten Abfrage wird hier nun nicht die Funktion Deref verwendet, sondern eine alternative Schreibweise über das Symbol ,->‘.

```
UPDATE Angestellte
SET AbtRef =
( SELECT AbtOID
  FROM Abteilungen
  WHERE AbtNr = 123 )
WHERE PersNr = 1112;
```

Nachdem diverse Abfragen durchgeführt wurden, soll die Abteilungszuordnung für den Angestellten mit der Personalnummer 1112 auf den Wert ,123‘ abgeändert werden. Dies geschieht hierbei über eine Unterabfrage.

```
DELETE FROM Angestellte
WHERE AbtRef -> AbtNr = 123;
```

Zuletzt sollen alle Datensätze, welche die Abteilungsnummer 123 enthalten, gelöscht werden. Hierfür ist erneut eine Dereferenzierung notwendig um die Abteilungsnummer des Abteilungsobjekts ausfindig zu machen.

2.5.1.2.3 Vererbung in Typ- und Tabellenhierarchien

Eine Vererbung von Eigenschaften kann zwischen Typ- und Tabellenhierarchien vorkommen. Hierbei erben die Subtypen/Subtabellen stets von den Supertypen/Supertabellen. Subtypen lassen sich von vorhandenen strukturierten Datentypen ableiten und erben neben den Attributen auch die Operationen des Supertyps. Subtabellen werden aus bestehenden typisierten Tabellen abgeleitet.


```
CREATE TYPE PersonTyp AS
( PersNr INTEGER,
  Name VARCHAR (40),
  Adresse AdressTyp )
NOT FINAL
INSTANTIABLE;
```

Zunächst soll ein neuer strukturierter Datentyp mit der Bezeichnung ‚PersonTyp‘ angelegt werden. Der Datentyp soll aus den Attributen PersNr (Integer), Name (Varchar) und Adresse (AdressTyp) bestehen. ‚INSTANTIABLE‘ gibt an, dass es möglich ist, Instanzen von diesem Datentyp zu bilden. Abstrakte Klassen würden somit über das Schlüsselwort ‚NOT INSTANTIABLE‘ angelegt.

```
CREATE TYPE AngTyp UNDER PersonTyp AS
( Gehalt GeldTyp,
  Abteilung VARCHAR (30) )
NOT FINAL
INSTANTIABLE;
```

Anschließend wird ein Subtyp ‚AngTyp‘ erstellt, welcher von dem Supertyp ‚PersonTyp‘ abgeleitet wird. Zusätzlich zu den Attributen einer Person werden die Attribute Gehalt (GeldTyp) und Abteilung (Varchar) festgelegt. Zudem ist es möglich Instanzen von diesem Typ zu bilden

```
CREATE TYPE ManagerTyp UNDER AngTyp AS
( HandyNr VARCHAR(15) )
NOT FINAL
INSTANTIABLE;
```

Es wird noch ein zweiter Subtyp mit der Bezeichnung ‚ManagerTyp‘ angelegt. Der Subtyp wird wiederum von dem eben erstellten Typ ‚AngTyp‘ abgeleitet und erhält zusätzlich das Attribut HandyNr (Varchar). Auch hier ist das bilden weiterer Instanzen möglich.

```
CREATE TABLE Personen OF PersonTyp
( REF IS PersOID SYSTEM GENERATED,
  PRIMARY KEY (PersNr) );
```

Bei diesem Statement wird eine typisierte Tabelle vom Typ PersonTyp erstellt. Das Attribut ‚PersNr‘ stellt den Primärschlüssel der Tabelle dar und die OID-Spalte wird vom System generiert.

```
CREATE TABLE Angestellte2 OF AngTyp UNDER Personen
( Gehalt WITH OPTIONS CHECK (Gehalt > CAST (2000 AS GeldTyp) ) );
```

Danach wird eine Subtabelle ‚Angestellte2‘ vom Typ AngTyp angelegt, welche von der Supertabelle Personen abgeleitet wird. Die Gehälter der Angestellten müssen jeweils über 2000 € liegen.


```
CREATE TABLE Manager OF ManagerTyp UNDER Angestellte2;
```

Zudem wird noch eine weitere Subtabelle ‚Manger‘ vom Typ Managertyp angelegt, welche von der Supertabelle ‚Angestellte2‘ abgeleitet ist.

```
SELECT *  
FROM Personen;
```

Bei dieser Abfrage werden alle Datensätze der Tabelle Personen ausgegeben.

```
SELECT *  
FROM ONLY (Personen);
```

Das Schlüsselwort ‚ONLY‘ bewirkt, dass in Abfragen Datensätze von Subtabellen ausgeschlossen werden. Somit werden nur Personen ausgegeben, die keine Angestellten oder Manager sind.

```
INSERT INTO Angestellte2 (PersNr, Name, Adresse, Gehalt, Abteilung)  
VALUES  
( 3333,  
  'Hans Klein',  
  Adresstyp ().Strasse ('Weitweg').HausNr (23).PLZ (68165),  
  CAST (8650 AS GeldTyp),  
  'Forschung' );
```

Anschließend wird in die Tabelle ‚Angestellte2‘ ein neuer Datensatz eingefügt.

```
UPDATE Angestellte2  
SET Gehalt = Gehalt * 1.1  
WHERE Abteilung = ‚Forschung‘;
```

Im Anschluss daran bekommen alle Angestellte der Forschungsabteilung eine Gehaltserhöhung von 10%. Hierfür müssen die Datensätze der Tabelle ‚Angestellte2‘ aktualisiert werden.

```
DELETE FROM Angestellte2  
WHERE Gehalt < CAST (2000 AS GeldTyp);
```

Abschließend werden die Datensätze der Angestellten aus der Tabelle ‚Angestellte2‘ gelöscht, die weniger als 2000 € verdienen. Allerdings dürften hierbei keine Datensätze gelöscht werden, da laut Tabellendefinition die Gehälter nicht kleiner als 2000 € sein dürften.

2.5.1.3 ROW-Datentyp

Neben der Definition eines strukturierten Datentyps, welche eine explizite Deklaration darstellt, gibt es die Möglichkeit einen ROW-Datentyp zu verwenden. Bei der Verwendung des ROW-Datentyps handelt es sich um eine implizite Deklaration und somit bleibt eine vorherige explizite Typ-Definition erspart.

```
CREATE TABLE Angestellte3
( PersNr INTEGER PRIMARY KEY,
  Name VARCHAR (40),
  Adresse ROW ( Straße VARCHAR (30),
               HausNr INTEGER,
               PLZ INTEGER,
               Stadt VARCHAR (20) ),
  Gehalt GeldTyp );
```

Zunächst wird eine neue Tabelle mit dem Namen ‚Angestellte3‘ erstellt. Die Tabelle besteht aus einer PersNr (Integer), welche den Primärschlüssel darstellt, einem Namen (Varchar), einer Adresse und aus einem Gehalt (GeldTyp). Der große Unterschied zu der Tabelle ‚Angestellte1‘ besteht darin, dass die Adresse nicht vom Typ Adresstyp ist, sondern die Attribute impliziert durch den ROW-Datentyp definiert werden.

```
SELECT Adresse.PLZ
FROM Angestellte3;
```

```
Anschließend wird auf die PLZ abgefragt.
UPDATE Angestellte3
SET Adresse = ROW ('Weitweg', 23, 12345, 'Kleinheim')
WHERE PersNr = 3333;
```

Zuletzt wird die Adresse des Angestellten mit der PersNr 3333 aktualisiert. Dies geschieht wiederum mit dem ROW-Datentyp.

2.5.1.4 Array-Datentyp

Listen lassen sich in SQL in Form von Arrays abbilden. Diese Array-Typen können für Tabellenspalten oder als Typ-Attribut verwendet werden.

```
CREATE TABLE Entleihungen
( Entleiher VARCHAR (40) PRIMARY KEY,
  BuchListe VARCHAR (50) ARRAY [20]);
```

Durch diese Anweisung wird eine Tabelle mit der Bezeichnung ‚Entleihungen‘ erzeugt. Die Tabelle hat insgesamt zwei Spalten. Der Entleiher (Varchar) stellt hierbei den Primärschlüssel dar. Die Buchliste (Varchar) wird vom Typ Array definiert. In dem Array lassen sich insgesamt 20 Buchtitel pro Entleiher speichern.

```
INSERT INTO Entleihungen
VALUES
( 'Fritz Klein',
  ARRAY ['Der Schakal', 'Die Reblaus', 'Nur Du allein'] );
```

Anschließend wird ein Datensatz in die Tabelle eingefügt. Der Entleiher Fritz Klein hat hierbei insgesamt drei Bücher ausgeliehen.

```
UPDATE Entleihungen
SET BuchListe = BuchListe || ARRAY ['Dick und Doof', 'Das Boot']
WHERE Entleiher = 'Fritz Klein';
```

Der zuvor angelegte Datensatz wird nun aktualisiert. Zu den drei entliehenen Büchern werden zwei neue Titel hinzugefügt. Das Anhängen der neu entliehenen Titeln an die bereits existierende Liste wird mit dem Konkatenations-Operator ,||' realisiert.

```
SELECT BuchListe [4]
FROM Entleihungen
WHERE CARDINALITY (BuchListe) >= 5;
```

Bei dieser Abfrage werden die Titel der entliehenen Bücher ausgegeben, die sich jeweils an vierter Stelle der jeweiligen Liste befindet. Zudem werden nur Bücher von Entleihungen angezeigt, bei denen mindestens fünf Bücher ausgeliehen wurden. Diese Bedingung wird über ,CARDINALITY' angegeben.

```
UPDATE Entleihungen
SET BuchListe [4] = 'Schlank in 7 Tagen'
WHERE Entleiher = 'Fritz Klein';
```

Hierbei wird bei dem Entleiher Fritz Klein das Buch, welches sich an vierter Stelle der Leihliste befindet, durch den Buchtitel ,Schlank in 7 Tagen' ersetzt.

```
SELECT Entleiher
FROM Entleihungen, UNNEST (BuchListe) AS Buch (Titel)
WHERE Titel = 'Das Boot';
```

Bei dieser Abfrage werden alle Entleiher ermittelt, die das Buch mit dem Titel ,Das Boot' geliehen haben. Um diese Abfrage ausführen zu können, muss das Array in eine einspaltige Tabelle, hier mit dem Spaltennamen Titel, umgewandelt werden. Die Angabe des Wortes ,AS' ist hierbei optional.

```
SELECT Entleiher, Nr  
FROM Entleihungen, UNNEST (BuchListe) AS Buch (Titel, Nr) WITH ORDINALITY  
WHERE Titel = 'Das Boot';
```

Diese Abfrage unterscheidet sich lediglich in der Ergänzung, dass neben dem Titel des Buches ebenfalls die Stelle angegeben werden soll, an der sich das Buch in der Entleihungsliste befindet.

2.5.1.5 CLOB- und BLOB-Datentypen

Für die Verwaltung großer Objekte gibt es zwei Datentypen. Zum einen das BLOB (Binary Large Object) für die Verwaltung von großen binären Objekten wie bspw. Bild- und Audiodateien und das CLOB (Character Large Object) für die Verwaltung von langen Zeichenketten wie bspw. eBooks. Bei der Definition eines Attributes kann die maximale Speichergröße des Datentyps durch Angabe von ‚K‘, ‚M‘ oder ‚G‘ festgelegt werden. ‚K‘ steht hierbei für Kilobyte, ‚M‘ für Megabyte und ‚G‘ für Gigabyte.

```
CREATE TABLE Buecher  
( BuchID INTEGER PRIMARY KEY,  
  Titel VARCHAR (200),  
  Zusammenfassung CLOB (32 K),  
  BuchText CLOB (20 M),  
  FilmZumBuch BLOB (2 G) );
```

Durch diese SQL-Anweisung wird eine Tabelle mit der Bezeichnung ‚Buecher‘ erstellt. Die Tabelle besitzt eine BuchID (Integer), die zugleich den Primärschlüssel darstellt, einen Titel (Varchar), eine Zusammenfassung (CLOB), einen Buchtext (CLOB) und einen FilmZumBuch (BLOB). In Klammern der Datentypen wird jeweils angegeben, wie groß das jeweilige Objekt maximal sein darf.

2.5.2 Benutzerdefinierte Routinen (Methoden)

Eine objektrelationale Datenverwaltung benötigt entsprechende Operationen. Daher sollte eine Datenbank nicht nur als Speicherort von Datensätzen fungieren, sondern darüber hinaus die Ausführung von Operationen erlauben. Diese Möglichkeit zur Erstellung von SQL-Routinen wird im SQL-Teilstandard SQL/PSM (Persistant Stored Modules) definiert. Insgesamt gibt es in SQL drei Arten von benutzerdefinierten Routinen, Prozeduren, Funktionen und Methoden. Der hauptsächliche Unterschied zwischen diesen Routinen besteht darin, dass Prozeduren keinen Rückgabewert liefern sondern in der Regel direkt Änderungen an der Datenbank vornehmen. Funktionen liefern hingegen einen Rückgabewert und ändern daher zumeist nicht direkt die Daten der Datenbank. Methoden stellen spezielle Funktionen dar, die sowohl überschreibbar als auch überladbar sind.

```

CREATE TYPE AngestTyp2 AS
( PersNr INTEGER,
  Name VARCHAR (40),
  Gehalt GeldTyp )
NOT FINAL
METHOD Gehaltserhoehung ()
RETURNS GeldTyp
LANGUAGE SQL
DETERMINISTIC;

```

Bei diesem Statement wird zunächst ein strukturierter Datentyp mit der Bezeichnung ‚Angesttyp2‘ zusammen mit einer Methodensignatur für Gehaltserhöhungen definiert. Die Tabelle besteht aus den Attributen PersNr (Integer), Name (Varchar) und Gehalt (Gehaltstyp). Die Methode liefert als Rückgabeparameter einen Wert vom Typ Geldtyp zurück. ‚DETERMINISTIC‘ gibt an, dass stets ein eindeutiger Wert zurückgeliefert wird.

```

CREATE TYPE ManagerTyp2 UNDER AngestTyp2 AS
( HandyNr VARCHAR (15) )
NOT FINAL
OVERRIDING METHOD Gehaltserhoehung ()
RETURNS GeldTyp
LANGUAGE SQL
DETERMINISTIC;

```

Anschließend wird ein Subtyp ‚ManagerTyp2‘ definiert, der von dem Supertyp ‚AngestTyp2‘ abgeleitet wird. Der strukturierte Datentyp besitzt als zusätzliches Attribut noch eine HandyNr (Varchar). Zudem wird die Methodensignatur einer überschreibenden Methode definiert.

```

CREATE METHOD Gehaltserhoehung ()
RETURNS GeldTyp
FOR AngestTyp2
BEGIN
  RETURN SELF.Gehalt * 0.2
END;

```

In der eigentlichen Methode ‚Gehaltserhoehung‘ wird das momentane Gehalt mit dem Faktor 0.2 multipliziert und das Ergebnis als Rückgabewert zurückgeliefert.

```

CREATE METHOD Gehaltserhoehung ()
RETURNS GeldTyp
FOR ManagerTyp2
BEGIN
  RETURN SELF.Gehalt * 0.1
END;

```

Durch dieses SQL-Statement wird die bereits vorhandene Methode zur Gehaltserhöhung überschrieben. So wird der Faktor durch den Wert 0.1 ersetzt.

```
CREATE TABLE Angestellte4 OF AngestTyp2
( REF IS AngestOid SYSTEM GENERATED,
  PRIMARY KEY (PersNr) );
```

Danach wird eine typisierte Tabelle ‚Angestellte4‘ vom Typ AngestTyp2 erstellt. Die OID-Spalte wird dabei systemgeneriert und das Attribut ‚PersNr‘ stellt den Primärschlüssel der Tabelle dar.

```
UPDATE Angestellte4
SET Gehalt = Gehalt + Deref(AngestOid).Gehaltserhoehung;
```

Im Anschluss daran wird das aktuelle Gehalt eines Angestellten durch den Rückgabewert der Methode ‚Gehaltserhoehung‘ ersetzt.

```
SELECT Name
FROM Angestellte4
WHERE Deref(AngestOid).Gehaltserhoehung > CAST (500 AS GeldTyp);
```

Zuletzt werden die Namen der Angestellten ausgegeben, deren Gehaltserhöhung größer als 500 ist.

2.5.3 XML-Dokumente

XML stellt eine erweiterbare Auszeichnungssprache dar. Es stellt die Kerntechnologie beim Datenaustausch zwischen Anwendungen über das Internet dar. Ein XML-Dokument enthält neben den eigentlichen Daten auch Strukturinformationen, sogenannte Metadaten. In nachfolgender Abb. 2 ist ein beispielhaftes XML-Dokument mit den unterschiedlichen Bestandteilen dargestellt.

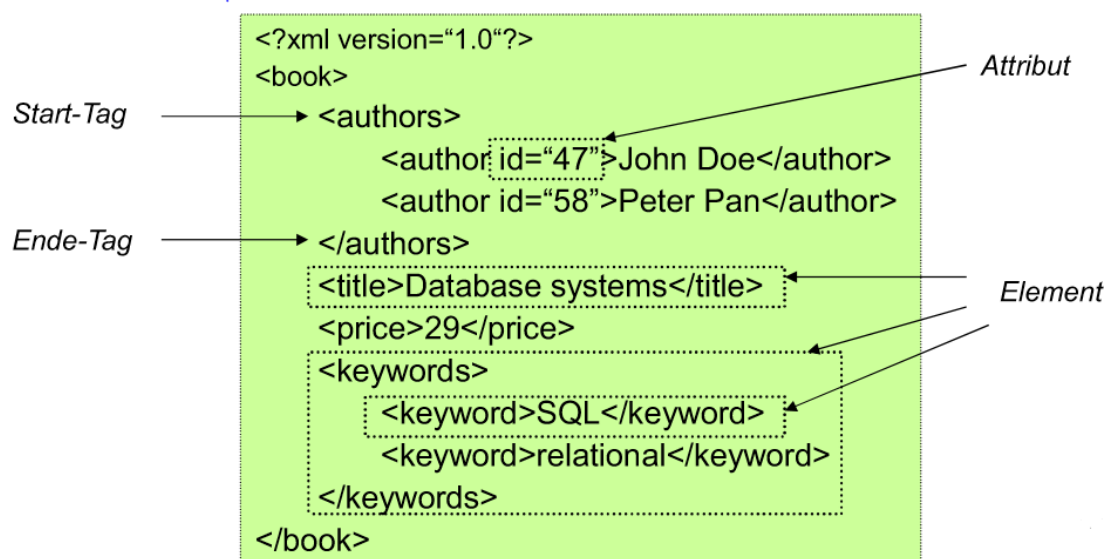


Abb. 2: Beispielhaftes XML-Dokument³⁵

³⁵ Enthalten in: Preiß, N. (2014)

Nachfolgend werden hierfür repräsentative SQL-Statements aufgeführt, die konform dem SQL-Standard sind.

```
CREATE TABLE Abteilung
( AbtNr INTEGER PRIMARY KEY,
  AbtName VARCHAR (20),
  AbtDoku XML);
```

Zunächst wird eine Tabelle ‚Abteilung‘ mit den Attributen AbtNr (Integer), welches zugleich der Primärschlüssel ist, AbtName (Varchar) und AbtDoku (XML) angelegt.

```
INSERT INTO Abteilung
VALUES
( 111,
  'Informatik',
  '<Abteilung Name="Informatik">
  <Budget>1.000.000</Budget>
  <Leiter>Karl Schreier</Leiter>
  <Mitarbeiter>Bernd Diener</Mitarbeiter>
  <Mitarbeiter>Ute Contra</Mitarbeiter>
  </Abteilung>'
);
```

Im nächsten Schritt werden Daten in die zuvor erstellte Tabelle eingefügt. Wie zu sehen ist, wird dem Attribut ‚AbtDoku‘ ein komplettes XML-Dokument zugewiesen.

```
CREATE TABLE Berichte
( ID INTEGER PRIMARY KEY,
  Inhalt XML);
```

Im Anschluss daran wird noch eine weitere Tabelle mit dem Namen ‚Berichte‘ erstellt. Die Tabelle erhält das Primärschlüssel-Attribut ID (Integer) und das Attribut Inhalt (XML).

Weitere XML-Statements des SQL-Standards sind in Anhang 4 beigefügt. Diese konnten aufgrund des begrenzten Umfangs nicht untersucht werden. Die Abweichungen der einzelnen DBS zu dem bis hierhin beschriebenen SQL-Standard werden im folgenden Teil untersucht und einzeln aufgeführt.

3 Gang der Untersuchung der SQL-Statements

Die im Rahmen der Untersuchung erarbeiteten Ergebnisse repräsentieren nicht den vollständigen Umfang der in Kapitel 2.5 dargestellten und beschriebenen SQL-Statements. In den einzelnen DBS konnte lediglich ein Substrakt erfolgreich umgesetzt werden. Die exakten Ergebnisse sowie mögliche Abweichungen vom SQL-Standard werden im Rahmen dieses Kapitels ausführlich dargestellt. Bevor die beschriebene Untersuchung stattfinden kann müssen zuerst die drei DBS installiert und eingerichtet werden.

3.1 Systeminstallation und Ersteinrichtung

Im Folgenden wird ein Überblick über Installation und Einrichtung von Oracle, DB2 und PostgreSQL gegeben, bevor die eigentliche Untersuchung dargestellt wird.

3.1.1 Oracle - Installation und Inbetriebnahme

Als erster Schritt muss das DBS heruntergeladen und installiert werden. Eine exakte Anleitung ist hierzu in Anhang 5 vorzufinden. Nach abgeschlossener Installation von Oracle, wurden zunächst einige Einstellungen vorgenommen, bevor mit dem DBS gearbeitet werden konnte. Zunächst wurde die Bedienoberfläche von Oracle über ‚Startmenü → Oracle Database 11g Express Edition → Get Started‘ aufgerufen. Daraufhin wird in einem Browser-Fenster folgendes, in Abb. 3 dargestelltes, Auswahlmenü geöffnet.

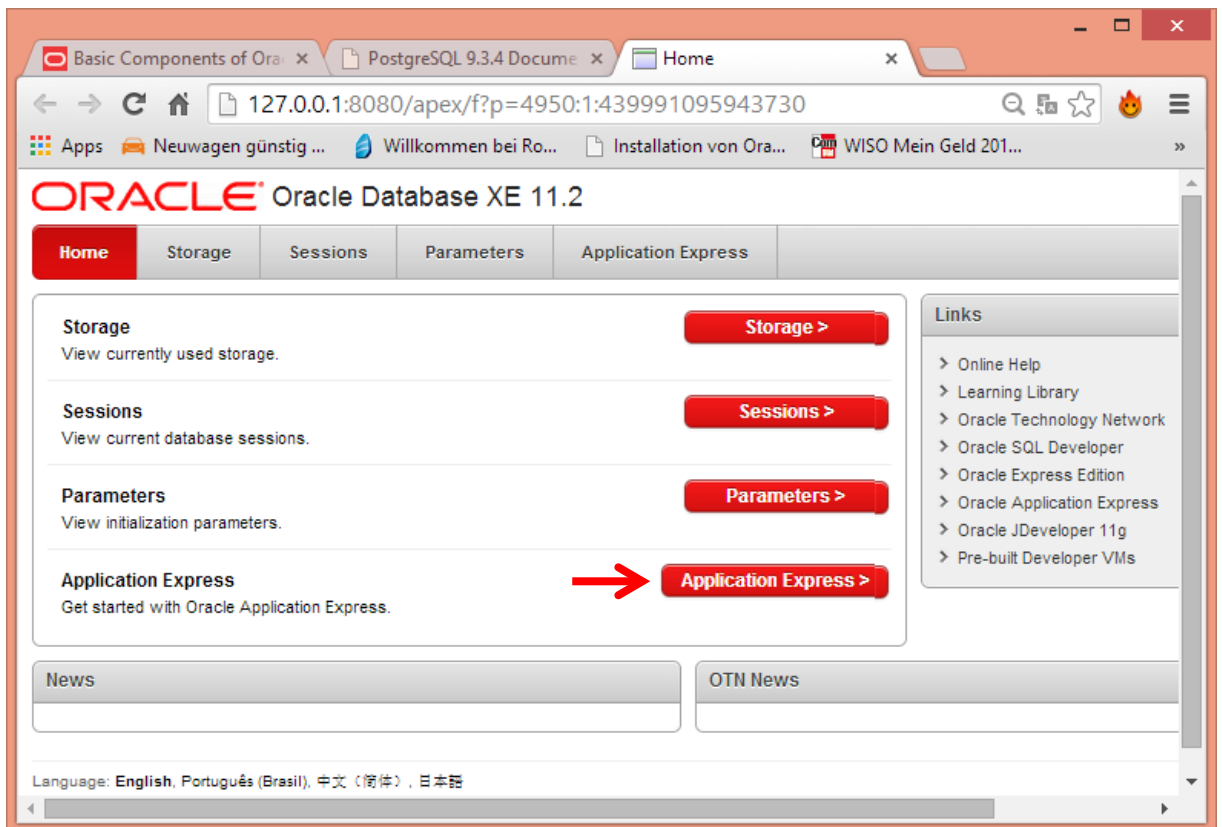


Abb. 3: Get Started in Oracle

Um eine Beispiel-Datenbank anzulegen, muss in dem Menü der Punkt ‚Application Express‘ ausgewählt werden. In der sich daraufhin öffnenden Maske ist zunächst eine Authentifizierung notwendig. Hierbei muss das Passwort angegeben werden, welches bei der Installation vergeben wurde. Nach erfolgter Anmeldung besteht die Möglichkeit einen neuen Workspace, sprich eine neue Datenbank, anzulegen. Hierfür muss ein Datenbankname, sowie ein Anwender und ein Passwort angegeben werden. Nachdem die Datenbank erfolgreich angelegt wurde, besteht die Möglichkeit sich an dieser anzumelden und anschließend mit ihr arbeiten zu können. Allerdings lassen sich mit dem SQLDeveloper komfortabler Tabellen anlegen und Beziehungen zueinander erstellen. Daher bietet es sich für die weiteren Schritte an, den SQLDeveloper zu starten. Nach Aufruf des SQLDevelopers muss zunächst eine Verbindung zu der zuvor angelegten Datenbank hergestellt werden. Über das Menü lässt sich eine neue Datenbankverbindung erstellen. In der sich daraufhin öffnenden Eingabemaske (Siehe Abb. 4), muss als erster Schritt ein beliebiger Verbindungsname angegeben werden. Anschließend muss der Name der zuvor angelegten Datenbank und das dazugehörige Passwort angegeben werden. Zudem besteht die Möglichkeit das Kennwort für die Verbindung zu speichern, sodass dieses nicht bei jedem Start abgefragt wird.

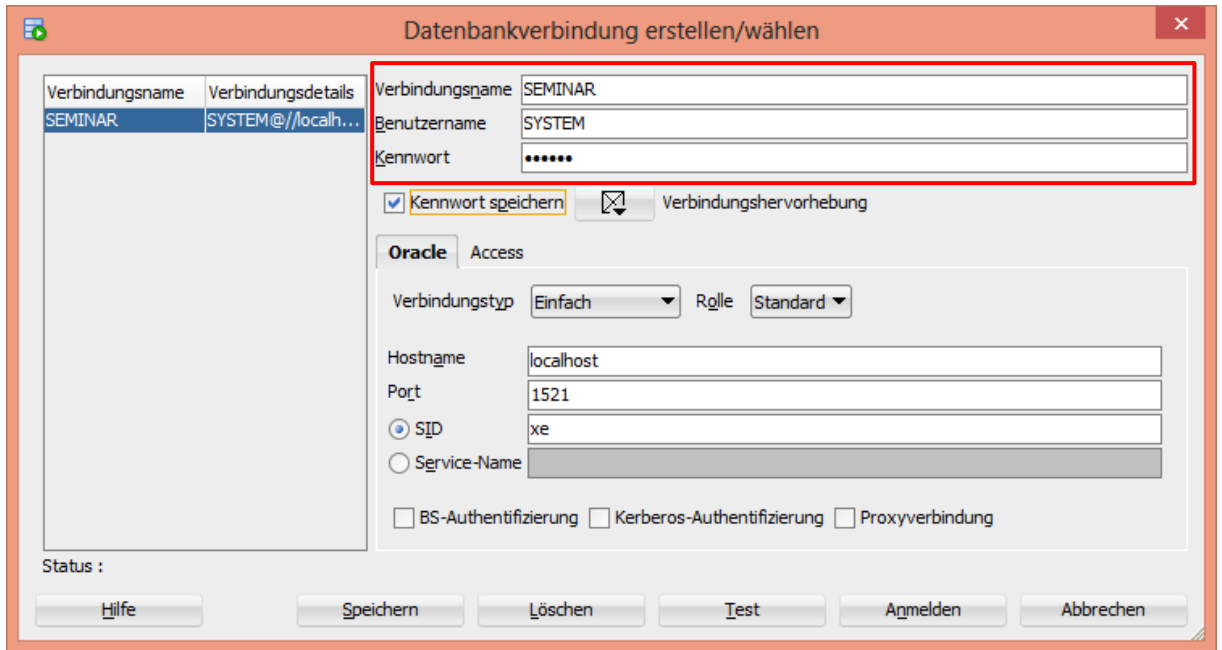


Abb. 4: Neue Datenbankverbindung anlegen in Oracle

Nach erfolgreicher Verbindung mit der Datenbank kann mit dieser bearbeitet werden und ist somit bereit zur Überprüfung der objektrelationalen SQL-Statements.

3.1.2 DB2 - Installation und Inbetriebnahme

Eine genaue Schritt-für-Schritt Anleitung zur Installation von DB2 ist in Anhang 6 beigefügt. Ist diese abgeschlossen können die ersten Einstellungen vorgenommen werden, um DB2 in Betrieb zu nehmen. Dazu muss im Frontend IBM Data Studio eine Datenbankverbindung angelegt werden. Im Fenster ‚Datenquellen-Explorer‘, welches sich links unten befindet, muss in der oberen Menüleiste ‚Neues SQL-Script‘ gewählt werden. Dieser Button ist in Abb. 5 mit einer roten ‚1‘ markiert. Daraufhin erscheint im Bereich rechts oben der Reiter ‚Script.sql‘, in welchem SQL-Statements eingegeben werden können. Bevor jedoch die Eingabe der SQL-Syntax erfolgen kann muss im Datenquellen-Explorer der Button für ein „Neues Verbindungsprofil“ angeklickt werden. Die betreffende Stelle ist in Abb. 5 mit einer roten ‚2‘ markiert.

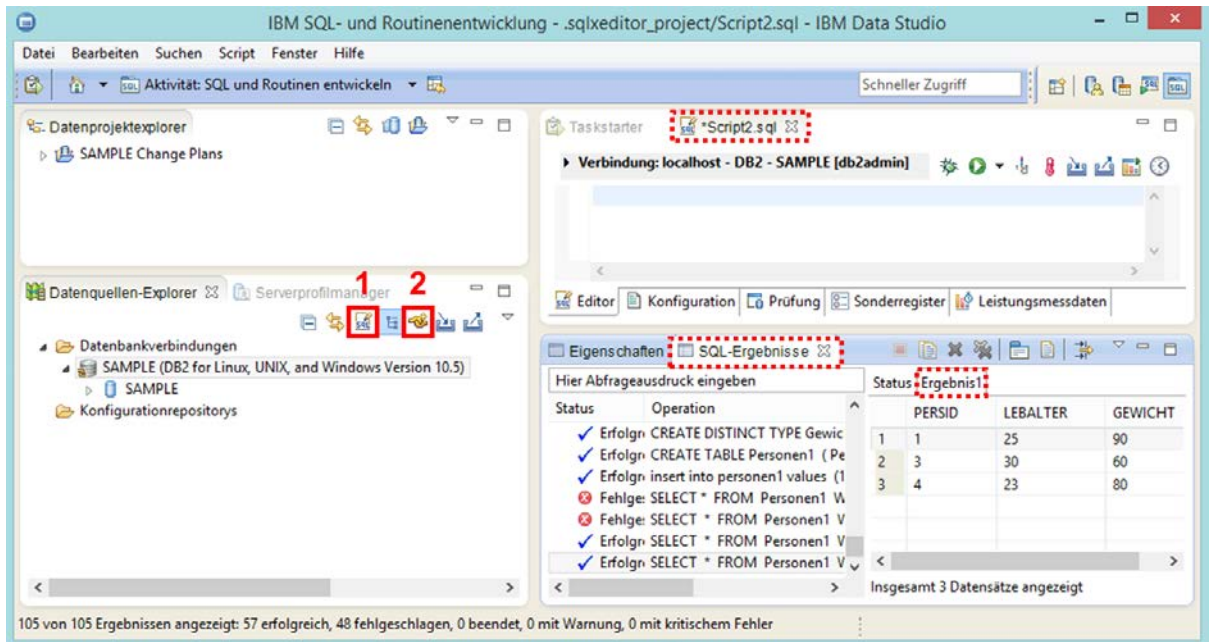


Abb. 5: Ersteinrichtung im IBM Data Studio für DB2

In einem sich daraufhin öffnenden Fenster ist der Datenbankname ‚SAMPLE‘, der bei der Installation angegebene Benutzername und das Kennwort einzugeben. Mit einem Klick auf ‚Fertig stellen‘ wird eine Verbindung zu der bei der Installation erzeugten Beispieldatenbank erzeugt. Somit können durch betätigen des Buttons mit dem grünen Pfeil im Reiter oben rechts die SQL-Anweisungen ausgeführt werden. Im darunter liegenden Bildschirmbereich werden Fehler- oder Erfolgsmeldungen bei der Ausführung angezeigt. Die Ergebnisse einer SELECT-Anweisung sind unter dem Reiter ‚Ergebnis1‘ zu finden. Diese Bereiche sind in Abb. 5 mit einer roten gestrichelten Linie markiert.

3.1.3 PostgreSQL - Installation und Inbetriebnahme

Zunächst ist die Installation des DBS notwendig. Eine ausführliche Installationsanleitung ist hierfür in Anhang 7 vorzufinden. Nach erfolgreicher Installation des Systems PostgreSQL muss dieses ebenfalls zunächst konfiguriert werden. Hierzu ist zunächst eine Verbindung mit dem PostgreSQL-Server notwendig. Diese kann am schnellsten über den Objektbrowser hergestellt werden. Dafür genügt ein Doppelklick auf anfangs unverbundenen PostgreSQL-Server. Daraufhin öffnet sich ein Popup-Fenster mit einer Passwortaufforderung, welches in folgender Abb. 6 dargestellt ist.

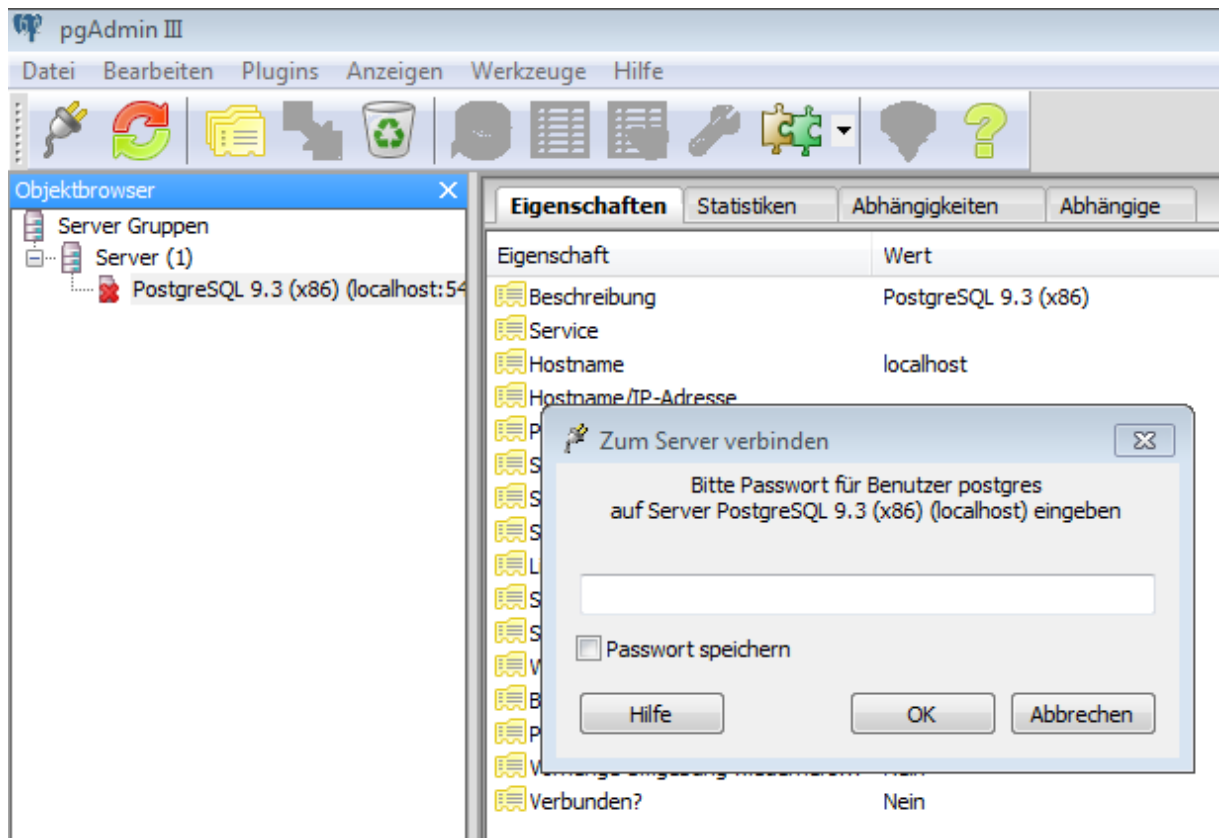


Abb. 6: Verbindungsaufbau zum PostgreSQL-Server

Um die Verbindung herstellen zu können, muss das Passwort angegeben werden, welches bei der Installation angegeben wurde. Bei erfolgreicher Verbindung wurde als nächster Schritt eine neue Datenbank angelegt. Hierfür muss im Objektbrowser zunächst auf den Eintrag ‚Datenbanken‘ mit der rechten Maustaste geklickt und anschließend der Menüpunkt ‚Neue Datenbank‘ ausgewählt werden. In der darauf folgenden Eingabemaske kann der Name der neu zu erstellenden Datenbank festgelegt werden. Nach Bestätigung durch Klick auf den OK-Button wird die Datenbank angelegt. Als weitere Einstellung sollte unter ‚Datei → Optionen → Browser → Anzeige‘ ein Haken bei dem Eintrag ‚Typen‘ gesetzt werden. Dadurch werden im Objektbrowser alle individuell erstellten Datentypen angezeigt. Anschließend sind die notwendigen Einstellungen abgeschlossen und es kann mit der Datenbank gearbeitet und somit die objektrelationalen SQL-Statements getestet werden.

3.2 Vorgehen bei der Statementuntersuchung

Die Vorgehensweise, die für die Untersuchung der SQL-Statements aus dem Standard vorgesehen war, ist Abb. 7 zu entnehmen.



Abb. 7: Vorgehen der Statementuntersuchung

Im ersten Schritt hat das Projektteam im Internet und in den jeweiligen Handbüchern nach Angaben zu den objektrelationalen Erweiterungen der einzelnen DBS recherchiert. Die Angaben dabei sind sehr zurückhaltend und allgemein gehalten. Daher wurde in einem weiteren Schritt an die Entwicklerfirmen sowie die Ausbildungsunternehmen des Projektteams von Oracle und DB2 herangetreten. Für PostgreSQL entfiel das Herantreten an die Entwicklerfirma, da es sich bei diesem DBS um ein Open Source-System handelt. Bei den Entwicklern wurde nach entsprechenden Ansprechpartnern zu der vorliegenden Problemstellung gefragt. Zusätzlich wurde um Dokumentationen der jeweiligen DBS gebeten, aus welchen eventuell Angaben und Syntaxbeispiele zu entnehmen wären. Als Rückmeldung wurden einige Hinweise auf veröffentlichte Artikel zu den objektrelationalen Features gesendet. Die Ansprechpartner konnten innerhalb des zur Verfügung stehenden Zeitraums leider keine umfassende Kooperation anbieten, um die verschiedenen SQL-Statements in der Syntax des jeweiligen Systems zu prüfen, obwohl zumindest seitens IBM die Bereitschaft dazu vermittelt wurde. Daher verblieb im letzten Schritt das Selbststudium, indem die standardisierten SQL-Statements in die Syntax des jeweiligen DBS zu übertragen waren. Die beiden Aufgaben, in welche sich das Selbststudium zur Untersuchung der SQL-Statements untergliedert sind in Abb. 8 dargestellt.



Abb. 8: Vorgehensweise des Selbststudiums

Zunächst wird versucht die Syntax der Statements dementsprechend abzuändern, dass das jeweilige DBS die Eingabe akzeptiert und ausführt. Im darauf folgenden Schritt ist die Semantik zu prüfen. Dabei muss untersucht werden, ob mit dem durchgeführten Statement auch die gewünschte Aktion im DBS ausgeführt und dabei die entsprechenden Daten angelegt wurden. Diese Vorgehensweise wurde gewählt, um in der kurzen zur Verfügung stehenden Zeit so viele Statements wie möglich zu prüfen und primär herauszufinden, welche Syntaxänderungen vorgenommen werden müssen.

3.3 Statementuntersuchung Oracle

In diesem Kapitel werden die Ergebnisse der Untersuchung aufgeführt, die für das DBS Oracle herausgefunden wurden.

3.3.1 Benutzerdefinierte Datentypen

Nachfolgend werden die Abweichungen zu den benutzerdefinierten Datentypen aufgeführt.

3.3.1.1 Distinct-Datentyp

Statement nach SQL-Standard	Statement unter Oracle
CREATE TYPE LebAlterTyp AS INTEGER FINAL;	CREATE TYPE LebAlterTyp AS OBJECT (LebAlter INTEGER);
In Oracle ist bei der Erstellung eines Distinct-Datentyps die Angabe des Schlüsselwortes ‚OBJECT‘ notwendig. Der Objektname steht mit Angabe des Datentyps in Klammern dahinter. Die Angabe der Klausel ‚FINAL‘ kann hierbei erfolgen oder weggelassen werden und ist somit nicht zwingend notwendig.	
CREATE TYPE GewichtTyp AS INTEGER FINAL;	CREATE TYPE GewichtTyp AS OBJECT (Gewicht INTEGER);
Wie bei der Erstellung der vorherigen Datentyps muss erneut die Angabe des Schlüsselwortes ‚OBJECT‘ erfolgen. Der Objektname steht hierbei ebenfalls mit Angabe des Datentyps in Klammern dahinter.	
CREATE TABLE Personen1 (PersID INTEGER PRIMARY KEY, LebAlter LebAlterTyp, Gewicht GewichtTyp);	CREATE TABLE Personen1 (PersID INTEGER PRIMARY KEY, LebAlter LebAlterTyp, Gewicht GewichtTyp);
<u>Ergänzung:</u> Einfügen von beispielhaften Datensätzen	INSERT INTO personen1 VALUES (1, LebAlterTyp (25), GewichtTyp(90)); INSERT INTO personen1 VALUES (2, LebAlterTyp (80), GewichtTyp(65)); INSERT INTO personen1 VALUES (3, LebAlterTyp (30), GewichtTyp(60)); INSERT INTO personen1 VALUES (4, LebAlterTyp (23), GewichtTyp(80)); INSERT INTO personen1 VALUES (5, LebAlterTyp (90), GewichtTyp(63)); INSERT INTO personen1 VALUES (6, LebAlterTyp (50), GewichtTyp(50));

Statement nach SQL-Standard	Statement unter Oracle
<p>Nachdem die Tabelle ‚Personen1‘ angelegt wurde, mussten zunächst Datensätze zu der Tabelle hinzugefügt werden, um bei den nachfolgenden Abfragen ein sinnvolles Ergebnis zu bekommen. Um die benutzerdefinierten Attribute zu befüllen, muss über den Datentyp des Attributs der Wert gesetzt werden.</p>	
<pre>SELECT * FROM Personen1 WHERE Gewicht > LebAlter;</pre>	<p>→ nicht funktionsfähig</p>
<p>Es war nicht möglich einen Vergleich zwischen zwei benutzerdefinierten Datentypen unter Oracle zu realisieren. Daher konnten aufgrund der geringen zur Verfügung stehenden Zeit keine tiefergehenden Untersuchungen durchgeführt werden, die hierfür notwendig sind.</p>	
<pre>SELECT * FROM Personen1 WHERE Gewicht > CAST (LebAlter AS GewichtTyp);</pre>	<p>→ nicht funktionsfähig</p>
<p>Beim Versuch eine Datentypumwandlung durchzuführen ist stets die Fehlermeldung ‚invalid CAST to a type that is not a nested table or VARRAY‘ aufgetreten. Daher konnten die beiden Attribute nicht miteinander verglichen werden.</p>	

3.3.1.2 Strukturierter Datentyp

In diesem Kapitel werden SQL-Statements zu strukturierten Datentypen untersucht.

3.3.1.2.1 Strukturierte Attribute

Statement nach SQL-Standard	Statement unter Oracle
<pre>CREATE TYPE GeldTyp AS DECIMAL (10,2) FINAL;</pre>	<pre>CREATE TYPE GeldTyp AS OBJECT (Geld DECIMAL (10,2));</pre>
<p>Wie bei jedem benutzerdefinierten Datentyp ist auch hier wieder die Angabe von ‚OBJECT‘ notwendig. Die Bezeichnung des Objekts und der Datentyp stehen wieder in Klammern dahinter.</p>	

Statement nach SQL-Standard	Statement unter Oracle
<pre>CREATE TYPE AdresTyp AS (Straße VARCHAR (30), HausNr INTEGER, PLZ DECIMAL (5), Stadt VARCHAR (20) DEFAULT 'Mannheim') NOT FINAL;</pre>	<pre>CREATE TYPE AdresTyp AS OBJECT (Straße VARCHAR (30), HausNr INTEGER, PLZ DECIMAL (5), Stadt VARCHAR (20));</pre>
<p>Bei der Definition eines strukturierten Datentyps werden die einzelnen Attribute wie beim Standard durch Kommas getrennt aufgelistet. Das Festlegen eines Default-Wertes ist bei dem DBS Oracle nicht möglich. Die Angabe der Klausel ‚NOT FINAL‘ kann hierbei erfolgen oder weggelassen werden und ist somit nicht zwingend notwendig.</p>	
<pre>CREATE TABLE Angestellte1 (PersNr INTEGER PRIMARY KEY, Name VARCHAR (40), Adresse AdresTyp, Gehalt GeldTyp);</pre>	<pre>CREATE TABLE Angestellte1 (PersNr INTEGER PRIMARY KEY, Name VARCHAR (40), Adresse AdresTyp, Gehalt GeldTyp);</pre>
<p>Eine Tabelle lässt sich unter Oracle standardkonform anlegen.</p>	
<pre>INSERT INTO Angestellte1 VALUES (3333, 'Franz Groß', AdresTyp ().Straße ('Weitweg').HausNr (23).PLZ (68165), CAST (8650 AS GeldTyp));</pre>	<pre>INSERT INTO ANGESTELLTE1 VALUES ('3333', 'Franz Groß', AdresTyp('Weitweg', 23, 68165, 'Mannheim'), GeldTyp(8650));</pre>
<p>Um einem strukturierten Datentyp konkrete Werte zuzuweisen, ist die Angabe des Typs notwendig. In Klammern stehen anschließend die konkreten Werte für die einzelnen Attribute.</p>	
<pre>SELECT Adresse FROM Angestellte1 WHERE PersNr = 3333;</pre>	<pre>SELECT Adresse FROM Angestellte1 WHERE PersNr = 3333;</pre>
<p>Die SQL-Abfrage ist erneut Abweichungen vom SQL-Standard realisierbar.</p>	
<pre>SELECT Adresse.PLZ () FROM Angestellte1 WHERE PersNr = 3333;</pre>	<pre>SELECT e.Adresse.PLZ FROM Angestellte1 e WHERE e.PersNr = 3333;</pre>
<p>Um ein einzelnes Attribut eines strukturierten Datentyps abfragen zu können, ist die Vergabe einer Art Referenz, in diesem Beispiel durch ‚e‘ repräsentiert, notwendig.</p>	

<pre>UPDATE Angestellte1 SET Adresse = Adresse.HausNr (77) WHERE PersNr = 3333;</pre>	<pre>UPDATE Angestellte1 SET Adresse = AdressTyp('Weitweg', 77, 68165, 'Mannheim') WHERE PersNr = 3333;</pre>
<p>Es ist nicht möglich einen einzelnen Wert des strukturierten Datentyps zu aktualisieren, da ständig die Fehlermeldung ‚invalid identifier‘ auftritt. Aus diesem Grund werden bei der Aktualisierung die restlichen Werte ebenfalls angegeben.</p>	

3.3.1.2.2 Vererbung in Typ- und Tabellen-Hierarchien

Statement nach SQL-Standard	Statement unter Oracle
<pre>CREATE TYPE PersonTyp AS (PersNr INTEGER, Name VARCHAR (40), Adresse AdressTyp) NOT FINAL INSTANTIABLE;</pre>	<pre>CREATE TYPE PersonTyp AS OBJECT (PersNr INTEGER, Name VARCHAR (40), Adresse AdressTyp) INSTANTIABLE;</pre>
<p>Das Anlegen eines strukturierten Datentyps ist wie gewohnt möglich.</p>	
<pre>CREATE TYPE AngTyp UNDER PersonTyp AS (Gehalt GeldTyp, Abteilung VARCHAR (30)) NOT FINAL INSTANTIABLE;</pre>	<pre>CREATE TYPE AngTyp UNDER PersonTyp (Gehalt GeldTyp, Abteilung VARCHAR (30)) NOT FINAL INSTANTIABLE;</pre>
<p>Die Definition eines Sub-Datentyps ist standardkonform möglich. Hierbei sind keine Änderungen notwendig. Die Angabe von AS ist hierbei wieder optional.</p>	
<pre>CREATE TYPE ManagerTyp UNDER AngTyp AS (HandyNr VARCHAR(15)) NOT FINAL INSTANTIABLE;</pre>	<pre>CREATE TYPE ManagerTyp UNDER AngTyp (HandyNr VARCHAR(15)) NOT FINAL INSTANTIABLE;</pre>
<p>Auch hier ist die das Erstellen eines Sub-Datentyps ohne Probleme möglich.</p>	

3.3.2 XML-Dokumente

In diesem Unterkapitel werden die Abweichungen zu XML-Dokumente dargestellt.

Statement nach SQL-Standard	Statement unter Oracle
<pre>CREATE TABLE Abteilung (AbtNr INTEGER PRIMARY KEY, AbtName VARCHAR (20), AbtDoku XML);</pre>	<pre>CREATE TABLE Abteilung (AbtNr INTEGER PRIMARY KEY, AbtName VARCHAR (20), AbtDoku XMLTYPE);</pre>
<p>Unter Oracle ist bei der Erstellung eines XML-Datentyps die Bezeichnung ‚XMLTYPE‘ notwendig.</p>	
<pre>INSERT INTO Abteilung VALUES (111, 'Informatik', '<Abteilung Name="Informatik"> <Budget>1.000.000</Budget> <Leiter>Karl Schreier</Leiter> <Mitarbeiter>Bernd Diener</Mitarbeiter> <Mitarbeiter>Ute Contra</Mitarbeiter> </Abteilung>');</pre>	<pre>INSERT INTO Abteilung VALUES (111, 'Informatik', xmltype.createxml ('<Abteilung Name = "Informatik"> <Budget>1.000.000</Budget> <Leiter>Karl Schreier</Leiter> <Mitarbeiter>Bernd Diener</Mitarbeiter> <Mitarbeiter>Ute Contra</Mitarbeiter> </Abteilung>');</pre>
<p>Beim Zuweisen von konkreten Werten an ein XML-Attribut ist die Angabe ‚xmltype.createxml‘ notwendig. Der konkrete Wert wird nachfolgend in Klammern angegeben.</p>	
<pre>CREATE TABLE Berichte (ID INTEGER PRIMARY KEY, Inhalt XML);</pre>	<pre>CREATE TABLE Berichte (ID INTEGER PRIMARY KEY, Inhalt XMLTYPE);</pre>
<p>Hier ist ebenfalls die Definition eines XML-Attributes unter Angabe von ‚XMLTYPE‘ ohne Probleme möglich.</p>	

Im Rahmen der Untersuchung der durch den Auftraggeber vorgegebenen Statements, konnten unter Oracle nicht zu allen Themenbereichen Lösungen gefunden werden. Aufgrund der Komplexität, des sehr begrenzten Zeitfensters und vor allem der Abweichungen vom SQL-Standard konnten die Themenbereiche typisierte Tabellen, ROW-Datentyp, Array-Datentyp, CLOB-/BLOB-Datentyp, benutzerdefinierte Routinen und XML-Dokumente weitestgehend nicht ergebnisführend bearbeitet werden.

Das Arbeiten mit dem Oracle SQL Developer erwies sich als sehr intuitiv. Die Usability der Anwendung war auf Grund der grafischen Benutzeroberfläche und vieler nützlicher Funktionen sehr gut. Negativ fiel hingegen um umfangreiche und langwierige Installation sowie die sehr nichts aussagenden Fehlermeldungen auf. Zudem sind die vielen Systemgenerierten Tabellen bei der Anlage einer neuen Datenbank nervig und führen zu Unübersichtlichkeit.

3.4 Statementuntersuchung DB2

Die Ergebnisse der Untersuchung, die für das DBS DB2 herausgefunden wurden, werden in diesem Kapitel aufgeführt.

3.4.1 Benutzerdefinierte Datentypen

Im Folgenden werden die Abweichungen zu den benutzerdefinierten Datentypen aufgeführt.

3.4.1.1 Distinct-Datentyp

Statement nach SQL-Standard	Statement unter DB2
<pre>CREATE TYPE LebAlterTyp AS INTEGER FINAL;</pre>	<pre>CREATE TYPE LebAlterTyp AS INTEGER;</pre> <p>Alternativ:</p> <pre>CREATE DISTINCT TYPE LebAlterTyp2 AS INTEGER WITH COMPARISONS;</pre>
<p>Beim Anlegen eines benutzerdefinierten Distinct-Typs in DB2 ist keine Angabe von ‚FINAL‘ erforderlich. Prinzipiell kann in DB2 keine Angabe von ‚FINAL‘ und ‚NOT FINAL‘ erfolgen. Eine weitere Alternative stellt die Angabe des Schlüsselwortes ‚DISTINCT‘ dar, ergänzt um ‚WITH COMPARISONS‘. Damit lässt sich ebenfalls ein Distinct-Typ definieren. Bei näherer Untersuchung beider Varianten konnte kein Unterschied im Umgang festgestellt werden.</p>	
<pre>CREATE TYPE GewichtTyp AS INTEGER FINAL;</pre>	<pre>CREATE TYPE GewichtTyp AS INTEGER;</pre>
<p>Die beschriebenen Veränderungen sind bei jeder Definition eines Distinct-Typs erforderlich, wie dem aufgeführten Statement zu entnehmen ist.</p>	

Statement nach SQL-Standard	Statement unter DB2
<pre>CREATE TABLE Personen1 (PersID INTEGER PRIMARY KEY, LebAlter LebAlterTyp, Gewicht GewichtTyp);</pre>	<pre>CREATE TABLE Personen1 (PersID INTEGER NOT NULL PRIMARY KEY, LebAlter LebAlterTyp, Gewicht GewichtTyp);</pre>
<p><u>Ergänzung:</u> Einfügen von beispielhaften Datensätzen</p>	<pre>INSERT INTO Personen1 VALUES (1, LebAlterTyp (25), GewichtTyp(90), 2, LebAlterTyp (80), GewichtTyp (65), 3, LebAlterTyp (30), GewichtTyp (60), 4, LebAlterTyp (23), GewichtTyp (80), 5, LebAlterTyp (90), GewichtTyp (63), 6, LebAlterTyp (50), GewichtTyp (50));</pre>
<p>Das Anlegen der Tabelle, können die Distinct-Typen ohne Änderungen der Syntax verwendet werden. Beim Anlegen eines Primärschlüssels muss jedoch ein syntaktischer Zusatz vorgenommen werden, da DB2 verlangt noch einmal explizit anzugeben, dass der Primärschlüssel keine leeren Werte enthalten darf. Daher ist ‚NOT NULL‘ zu ergänzen. Diese Angabe muss bei jeder Definition eines Primärschlüssels erfolgen und wird daher im weiteren Verlauf noch häufiger verwendet.</p>	
<pre>SELECT* FROM Personen1 WHERE Gewicht > LebAlter;</pre>	<p style="text-align: center;">→ nicht funktionsfähig</p>
<p>Dieses Statement ist, wie auch im Standard, nicht ausführbar, da die Datentypen, die durch den Operanden verbunden werden, nicht zueinander kompatibel sind. Nur gleiche Datentypen können in DB2 miteinander verglichen, addiert, subtrahiert, multipliziert und dividiert werden.</p>	
<pre>SELECT * FROM Personen1 WHERE Gewicht > CAST (LebAlter AS GewichtTyp);</pre>	<pre>SELECT * FROM Personen1 WHERE CAST (Gewicht AS INTEGER) > CAST (LebAlter AS INTEGER);</pre>
<p>Die Funktion einen benutzerdefinierten Datentyp in einen anderen umzuwandeln wird nicht unterstützt. Deshalb muss jeweils der benutzerdefinierte Datentyp in einen Standarddatentyp umgewandelt werden, wie dem Statement zu entnehmen ist.</p>	

3.4.1.2 Strukturierter Datentyp

In diesem Kapitel werden SQL-Statements zu strukturierten Datentypen untersucht.

3.4.1.2.1 Strukturierte Attribute

Statement nach SQL-Standard	Statement unter DB2
CREATE TYPE GeldTyp AS DECIMAL (10,2) FINAL;	CREATE TYPE GeldTyp AS DECIMAL (10,2);
Wie weiter oben bereits aufgeführt darf auch hier die Angabe von ‚FINAL‘ nicht erfolgen.	
CREATE TYPE AdressTyp AS (Straße VARCHAR (30), HausNr INTEGER, PLZ DECIMAL (5), Stadt VARCHAR (20) DEFAULT 'Mannheim') NOT FINAL;	CREATE TYPE AdressTyp AS (Straße VARCHAR (30), HausNr INTEGER, PLZ DECIMAL (5), Stadt VARCHAR (20)) MODE DB2SQL;
In DB2 ist die Angabe von ‚Mode DB2SQL‘ beim Erzeugen strukturierter Typen zwingend erforderlich. Aus der Dokumentation sowie der Recherche im Internet geht nicht hervor welche Aktion durch diesen Zusatz durchgeführt wird, es wird lediglich erwähnt, dass die Angabe notwendig ist. Die Angabe eines Defaultwertes ist bei der Definition eines Typs nicht möglich.	
CREATE TABLE Angestellte1 (PersNr INTEGER PRIMARY KEY, Name VARCHAR (40), Adresse AdressTyp, Gehalt GeldTyp);	CREATE TABLE Angestellte1 (PersNr INTEGER NOT NULL PRIMARY KEY, Name VARCHAR (40), Adresse AdressTyp, Gehalt GeldTyp);
Tabellen von strukturierten Typen können gemäß dem SQL-Standard erzeugt werden, wobei beim Primärschlüssel wieder ‚NOT NULL‘ zu berücksichtigen ist.	
INSERT INTO Angestellte1 VALUES (3333, 'Franz Groß', AdressTyp ().Straße ('Weitweg').HausNr (23).PLZ (68165), CAST (8650 AS GeldTyp));	INSERT INTO Angestellte1 VALUES (3333, 'Franz Groß', AdressTyp () ..Straße ('Weitweg') ..HausNr (23) PLZ (68165) ..Stadt ('Mannheim'), CAST (8650 AS GeldTyp));
In DB2 müssen Methoden und Objektkomponenten mit dem Doppelpunkt-Operator aufgerufen werden. Wie dem Statement zu entnehmen ist, erfolgt damit beispielsweise das Befüllen des strukturierten Datentyps.	

Statement nach SQL-Standard	Statement unter DB2
<pre>SELECT Adresse FROM Angestellte1 WHERE PersNr = 3333;</pre>	<p>→ nicht funktionsfähig</p>
<p>Dieses Statement funktioniert nicht, da die Fehlermeldung erscheint „Eine Umsetzungsgruppe ist für den Datentyp nicht definiert“. Über das Statement können die Daten nicht am Stück ausgelesen werden. Die Inhalte der strukturierten Attribute werden bei der Betrachtung der Daten in der Tabelle mit dem Wert ‚NULL‘ angezeigt.</p>	
<pre>SELECT Adresse.PLZ () FROM Angestellte1 WHERE PersNr = 3333;</pre>	<pre>SELECT Adresse..PLZ () FROM Angestellte1 WHERE PersNr = 3333;</pre>
<p>Die Inhalte des strukturierten Datentyps müssen, wie dieses Statement zeigt, einzeln angesprochen werden. Dadurch wird ein Auslesen ermöglicht.</p>	
<pre>UPDATE Angestellte1 SET Adresse = Adresse.HausNr (77) WHERE PersNr = 3333;</pre>	<pre>UPDATE Angestellte1 SET Adresse = Adresse..HausNr (77) WHERE PersNr = 3333;</pre>
<p>Das Updaten der Inhalte ist wie im Standard definiert möglich, es muss lediglich der Doppelpunkt-Operator verwendet werden.</p>	

3.4.1.2.2 Typisierte Tabellen

Statement nach SQL-Standard	Statement unter DB2
<pre>CREATE TYPE AbtTyp AS (AbtNr INTEGER, AbtName VARCHAR (20), Budget GeldTyp) NOT FINAL REF IS SYSTEM GENERATED;</pre>	<pre>CREATE TYPE AbtTyp2 AS (AbtNr INTEGER, AbtName VARCHAR (20), Budget GeldTyp) INSTANTIABLE MODE DB2SQL;</pre>
<p>Der spezielle Zusatz ‚MODE DB2SQL‘ in DB2 muss ebenso bei typisierten Typen angegeben werden. Die Erzeugung der Referenz wird dabei standardmäßig vom System generiert. Wie im folgenden Statement zu sehen ist gibt, es die weitere Möglichkeit die OID-Werte anzugeben.</p>	

Statement nach SQL-Standard	Statement unter DB2
<pre>CREATE TYPE AngestTyp AS (PersNr INTEGER, PersName VARCHAR (40), Gehalt GeldTyp, AbtRef REF (AbtTyp)) NOT FINAL REF USING INTEGER;</pre>	<pre>CREATE TYPE AngestTyp3 AS (PersNr INTEGER, PersName VARCHAR (40), Gehalt GeldTyp, AbtRef REF (AbtTyp2)) REF USING INTEGER MODE DB2SQL;</pre>
<p>Wie in diesem Statement zu sehen ist wird ein Verweis auf den benutzerdefinierten Datentyp ‚AbtTyp2‘ erstellt. Über diese angegebene Syntax ‚REF USING INTEGER‘ muss der Benutzer beim Anlegen eines Objekts explizit eine OID-Wert angeben. Veränderungen der Syntax sind in DB2 gegenüber dem Standard nicht erforderlich.</p>	
<pre>CREATE TABLE Abteilungen OF AbtTyp (PRIMARY KEY (AbtNr), REF IS AbtOID SYSTEM GENERATED);</pre>	<p>→ nicht funktionsfähig</p>
<p>Beim Anlegen der typisierten Tabelle treten Probleme bei der Festlegung des Primärschlüssels auf. Das System erwartet die Angabe von ‚NOT NULL‘. Wird die Syntax jedoch um diese Angabe ergänzt, meldet die Syntaxhilfe einen syntaktischen Fehler. Es entsteht ein Deadlock. Daher konnte diese Tabelle nicht angelegt werden. In diese Problemstellung muss weiterer Zeitaufwand investiert werden, um eine Lösung zu finden.</p>	

3.4.1.2.3 Vererbung in Typ- und Tabellen-Hierarchien

Statement nach SQL-Standard	Statement unter DB2
<pre>CREATE TYPE PersonTyp AS (PersNr INTEGER, Name VARCHAR (40), Adresse AdressTyp) NOT FINAL INSTANTIABLE;</pre>	<pre>CREATE TYPE PersonTyp AS (PersNr INTEGER, Name VARCHAR (40), Adresse AdressTyp) MODE DB2SQL;</pre>
<p>Das Anlegen eines benutzerdefinierten Datentyps muss hier ebenfalls, wie weiter oben bereits beschrieben, lediglich die Angabe von ‚MODE DB2SQL‘ vorgenommen werden.</p>	

Statement nach SQL-Standard	Statement unter DB2
<pre>CREATE TYPE AngTyp UNDER PersonTyp AS (Gehalt GeldTyp, Abteilung VARCHAR (30)) NOT FINAL INSTANTIABLE;</pre>	<pre>CREATE TYPE AngTyp UNDER PersonTyp AS (Gehalt GeldTyp, Abteilung VARCHAR (30)) INSTANTIABLE MODE DB2SQL;</pre>
<p>Auch bei der Vererbung sind keine Änderungen in der Syntax vorzunehmen. Der Typ ‚PersonTyp‘ wird wie im Standard angelegt, indem er vom Typ ‚AngTyp‘ erbt.</p>	
<pre>CREATE TYPE ManagerTyp UNDER AngTyp AS (HandyNr VARCHAR(15)) NOT FINAL INSTANTIABLE;</pre>	<pre>CREATE TYPE ManagerTyp UNDER AngTyp AS (HandyNr VARCHAR(15)) INSTANTIABLE MODE DB2SQL;</pre>
<p>Dieses Statement kann ebenso ohne zusätzliche Veränderungen übernommen werden.</p>	
<pre>CREATE TABLE Personen OF PersonTyp (REF IS PersOID SYSTEM GENERATED, PRIMARY KEY (PersNr));</pre>	<p>→ nicht funktionsfähig</p>
<p>Bei diesem Statement tritt derselbe Fehler auf, wie unter Fehler! Verweisquelle konnte nicht gefunden werden. erläutert wurde. Die Angaben der Syntaxhilfe für die Definition des Primärschlüssels sind kontrovers. Zum einen Verlangt er die Angabe von ‚NOT NULL‘ zum anderen erwartet er jedoch einen anderen Wert an dieser Stelle, wenn dies angegeben wird. Aufgrund dieser Problemstellung wurden die weiteren Statements zur Vererbung nicht untersucht.</p>	

3.4.1.3 CLOB- und BLOB-Datentyp

Statement nach SQL-Standard	Statement unter DB2
<pre>CREATE TABLE Buecher (BuchID INTEGER PRIMARY KEY, Titel VARCHAR (200), Zusammenfassung CLOB (32 K), BuchText CLOB (20 M), FilmZumBuch BLOB (2 G));</pre>	<pre>CREATE TABLE Buecher (BuchID INTEGER PRIMARY KEY NOT NULL, Titel VARCHAR (200), Zusammenfassung CLOB (32 K), BuchText CLOB (20 M), FilmZumBuch BLOB (2 G));</pre>
<p>Das Anlegen eines CLOB und BLOB-Datentyps entspricht in DB2 dem Standard. Es müssen keine Änderungen der Syntax vorgenommen werden, abgesehen von der bereits in Fehler! Verweisquelle konnte nicht gefunden werden. beschriebenen Ergänzung durch ‚NOT NULL‘ beim Primärschlüssel. Die Größenangabe von CLOB und BLOB ist wie im Standard in Kilobyte (K), Megabyte (M) und Gigabyte (G) möglich.</p>	

3.4.2 XML-Dokumente

Die Abweichungen zu XML-Dokumente werden in diesem Unterkapitel dargestellt.

Statement nach SQL-Standard	Statement unter DB2
<pre>CREATE TABLE Abteilung (AbtNr INTEGER PRIMARY KEY, AbtName VARCHAR (20), AbtDoku XML);</pre>	<pre>CREATE TABLE Abteilung (AbtNr INTEGER PRIMARY KEY NOT NULL, AbtName VARCHAR (20), AbtDoku XML);</pre>
<p>Das Anlegen von ‚AbtDoku‘ vom Datentyp XML kann dem Standard entsprechend vorgenommen werden.</p>	

Statement nach SQL-Standard	Statement unter PostgreSQL
<pre>INSERT INTO Abteilung VALUES (111, 'Informatik', '<Abteilung Name="Informatik"> <Budget>1.000.000</Budget> <Leiter>Karl Schreier</Leiter> <Mitarbeiter>Bernd Diener</Mitarbeiter> <Mitarbeiter>Ute Contra</Mitarbeiter> </Abteilung>');</pre>	<p>→ nicht funktionsfähig</p>
<p>Bei der Übernahme des Standards gibt DB2 einen XML-Syntaxfehler aus. Dabei erscheint die Fehlermeldung „Attribute Value wurde nicht gefunden“. Weitere Prüfungen der Syntax blieben erfolglos.</p>	
<pre>CREATE TABLE Berichte (ID INTEGER PRIMARY KEY, Inhalt XML);</pre>	<pre>CREATE TABLE Berichte (ID INTEGER PRIMARY KEY NOT NULL, Inhalt XML);</pre>
<p>Wie im ersten Statement aufgeführt, funktioniert die Anlage von Attributen vom Typ XML problemlos. Aufgrund des eben beschriebenen Syntaxfehlers können die weiteren Statements, innerhalb der zur Verfügung stehenden Zeit, nicht untersucht werden.</p>	

Während der Untersuchung konnten keine Erkenntnisse zu ROW- und ARRAY-Datentypen gewonnen werden. Nach dem derzeitigen Stand der Untersuchung wird daher vermutet, dass DB2 beide Konstrukte nicht kennt. Zusätzlich bleibt zu erwähnen, dass der Datentyp BOOLEAN von DB2 nicht unterstützt wird, da keine Attribute von diesem Typ angelegt werden konnten. Für die benutzerdefinierten Routinen konnte, aufgrund der Komplexität sowie der starken Abweichung vom Standard in DB2, keine Umsetzung erarbeitet werden.

Abschließend lässt sich sagen, dass die Arbeit mit DB2 intuitiv und benutzerfreundlich war, abgesehen von der aufwendigen Installation. Dennoch ist eine Einarbeitungszeit notwendig, bis die möglichen Funktionen im Frontend Data Studio gefunden sind und eingesetzt werden können. Vor Allem die im Editor vorhandene Syntaxhilfe erwies sich als eine sehr positive Unterstützung für die Arbeit.

3.5 Statementuntersuchung PostgreSQL

In diesem Kapitel werden die Ergebnisse der Untersuchung aufgeführt, die für das DBS PostgreSQL herausgefunden wurden.

3.5.1 Benutzerdefinierte Datentypen

Nachfolgend werden die Abweichungen zu den benutzerdefinierten Datentypen aufgeführt.

3.5.1.1 Distinct-Datentyp

Statement nach SQL-Standard	Statement unter PostgreSQL
CREATE TYPE LebAlterTyp AS INTEGER FINAL;	CREATE TYPE LebAlterTyp AS (LebAlter INTEGER);
Entgegen dem Standard muss bei PostgreSQL bei der Definition eines neuen Datentyps neben einer Bezeichnung für den Datentyp auch ein Attributname vergeben werden, bei welchem der Datentyp angegeben wird. In PostgreSQL ist die Angabe der Klausel ‚FINAL‘ nicht möglich.	
CREATE TYPE GewichtTyp AS INTEGER FINAL;	CREATE TYPE GewichtTyp AS (Gewicht INTEGER);
Die Abweichung vom Standard betrifft hierbei jede Definition eines benutzerdefinierten Datentyps.	
CREATE TABLE Personen1 (PersID INTEGER PRIMARY KEY, LebAlter LebAlterTyp, Gewicht GewichtTyp);	CREATE TABLE Personen1 (PersID INTEGER PRIMARY KEY, LebAlter LebAlterTyp, Gewicht GewichtTyp);
Ergänzung: Einfügen von beispielhaften Datensätzen	INSERT INTO Personen1 VALUES (1, ROW(25), ROW(90)), (2, ROW(80), ROW(65)), (3, ROW(30), ROW(60)), (4, ROW(23), ROW(80)), (5, ROW(90), ROW(63)), (6, ROW(50), ROW(50));

Statement nach SQL-Standard	Statement unter PostgreSQL
<p>Als Zusatz wurde nach erfolgreicher Erstellung der Tabelle ‚Personen1‘ insgesamt sechs Datensätze zu der Tabelle hinzugefügt, um die semantische Richtigkeit der nachfolgenden Abfragen sicherstellen zu können. Um den Attributen eines benutzerdefinierten Datentyps Werte zuweisen zu können, ist die explizite Angabe des Schlüsselwortes ‚ROW‘ notwendig. Die Angabe des konkreten Wertes erfolgt anschließend in Klammern.</p>	
<pre>SELECT * FROM Personen1 WHERE Gewicht > LebAlter;</pre>	<pre>SELECT * FROM Personen1 WHERE Gewicht > LebAlter;</pre>
<p>Der Vergleich zweier unterschiedlicher benutzerdefinierter Datentypen ist unter PostgreSQL ohne Probleme möglich. Das System liefert die semantisch korrekten Abfrageergebnisse, ohne dass ein Cast notwendig ist.</p>	
<pre>SELECT * FROM Personen1 WHERE Gewicht > CAST (LebAlter AS GewichtTyp);</pre>	<p style="color: red; text-align: center;">→ nicht notwendig</p>
<p>Da bei PostgreSQL keine Datentypenumwandlung notwendig ist, wird keine explizite Durchführung eines Casts benötigt.</p>	

3.5.1.2 Strukturierter Datentyp

In diesem Kapitel werden SQL-Statements zu strukturierten Datentypen untersucht.

3.5.1.2.1 Strukturierte Attribute

Statement nach SQL-Standard	Statement unter PostgreSQL
<pre>CREATE TYPE GeldTyp AS DECIMAL (10,2) FINAL;</pre>	<pre>CREATE TYPE GeldTyp AS (Geld DOUBLE PRECISION);</pre>
<p>Für die Definition einer Gleitkommazahl ist in PostgreSQL die Angabe des Datentyps ‚DOUBLE PRECISION‘ notwendig.</p>	

Statement nach SQL-Standard	Statement unter PostgreSQL
<pre>CREATE TYPE AdressTyp AS (Straße VARCHAR (30), HausNr INTEGER, PLZ DECIMAL (5), Stadt VARCHAR (20) DEFAULT 'Mannheim') NOT FINAL;</pre>	<pre>CREATE TYPE AdressTyp AS (Straße Varchar(30), HausNr INTEGER, PLZ DECIMAL(5), Stadt VARCHAR(20));</pre>
<p>Unter PostgreSQL ist die Angabe eines Default-Wertes für ein Attribut eines benutzerdefinierten Datentyps sowie die Angabe der Klausel ‚FINAL‘ nicht möglich.</p>	
<pre>CREATE TABLE Angestellte1 (PersNr INTEGER PRIMARY KEY, Name VARCHAR (40), Adresse AdressTyp, Gehalt GeldTyp);</pre>	<pre>CREATE TABLE Angestellte1 (PersNr INTEGER PRIMARY KEY, Name VARCHAR (40), Adresse AdressTyp, Gehalt GeldTyp);</pre>
<p>Das Erstellen der Tabelle ‚Angestellte1‘ war standardkonform möglich.</p>	
<pre>INSERT INTO Angestellte1 VALUES (3333, 'Franz Groß', AdressTyp ().Straße ('Weitweg').HausNr(23). PLZ (68165), CAST (8650 AS GeldTyp));</pre>	<pre>INSERT INTO Angestellte1 VALUES (4444, 'Franz Groß', ROW('Weitweg', 23, 68165, 'Mannheim'), ROW(8650));</pre>
<p>Auch bei dieser Insert-Anweisung erfolgt die Zuweisung konkreter Werte an ein Attribut eines benutzerdefinierten Datentyps über Angabe von ‚ROW‘. Bei strukturierten Datentypen erfolgt die Angabe der einzelnen Werte durch Trennung eines Kommas.</p>	
<pre>SELECT Adresse FROM Angestellte1 WHERE PersNr = 3333;</pre>	<pre>SELECT Adresse FROM Angestellte1 WHERE PersNr = 3333;</pre>
<p>Die Abfrage war standardkonform möglich.</p>	
<pre>SELECT Adresse.PLZ () FROM Angestellte1 WHERE PersNr = 3333;</pre>	<pre>SELECT (Adresse).PLZ FROM Angestellte1 WHERE PersNr = 3333;</pre>
<p>Um den Wert eines einzelnen Attributes eines strukturierten Datentyps abfragen zu können, muss das strukturierte Attribut in Klammern stehen, gefolgt von einem Punkt und dem Namen des Attributes, auf das zugegriffen werden soll.</p>	

Statement nach SQL-Standard	Statement unter PostgreSQL
<pre>UPDATE Angestellte1 SET Adresse = Adresse.HausNr (77) WHERE PersNr = 3333;</pre>	<pre>UPDATE Angestellte1 SET Adresse.HausNr = 77 WHERE PersNr = 3333;</pre>
<p>Um einen Wert eines Attributs mit strukturiertem Datentyp ändern zu können, genügt entgegen der Umsetzung bei SELECT-Abfragen die folgende Angabe ‚<i>strukturiertes Attribut.Attributname</i>‘. Die im Standard notwendigen Klammern können weggelassen werden.</p>	

3.5.1.2.2 Typisierte Tabellen

Bei der Realisierung typisierter Tabellen kam es zu großen Problemen bei der Angabe der Festlegung wie die OID-Spalte angelegt werden soll. Um dennoch an einem stark vereinfachten Beispiel zeigen zu können, wie sich typisierte Tabellen unter PostgreSQL erstellen lassen, wurden zwei SQL-Statements abgewandelt und nachfolgend dargestellt.

Statement nach SQL-Standard	Statement unter PostgreSQL
<pre>CREATE TYPE AbtTyp AS (AbtNr INTEGER, AbtName VARCHAR (20), Budget GeldTyp) NOT FINAL REF IS SYSTEM GENERATED;</pre>	<pre>CREATE TYPE AbtTyp AS (AbtNr INTEGER, AbtName VARCHAR (20), Budget GeldTyp);</pre>
<p>Bei der Erstellung des vereinfachten Datentyps wird auf die Angabe, dass die OID-Spalte systemgeneriert ist, verzichtet.</p>	
<pre>CREATE TABLE Abteilungen OF AbtTyp (PRIMARY KEY (AbtNr), REF IS AbtOID SYSTEM GENERATED);</pre>	<pre>CREATE TABLE Abteilungen OF AbtTyp (PRIMARY KEY (AbtNr));</pre>
<p>Die Erstellung der typisierten Tabelle ‚Abteilungen‘ erfolgt anschließend ebenfalls ohne Angabe einer systemgenerierten OID-Spalte.</p>	

3.5.1.3 Array-Datentyp

Statement nach SQL-Standard	Statement unter PostgreSQL
<pre>CREATE TABLE Entleihungen (Entleiher VARCHAR (40) PRIMARY KEY, BuchListe VARCHAR (50) ARRAY [20]);</pre>	<pre>CREATE TABLE Entleihungen (Entleiher VARCHAR (40) PRIMARY KEY, BuchListe VARCHAR (50) ARRAY [20]);</pre>
Ein Array lässt sich standardkonform ohne Änderungen des SQL-Statements anlegen.	
<pre>INSERT INTO Entleihungen VALUES ('Fritz Klein', ARRAY ['Der Schakal', 'Die Reblaus', 'Nur Du allein']);</pre>	<pre>INSERT INTO Entleihungen VALUES ('Fritz Klein', ARRAY ['Der Schakal', 'Die Reblaus', 'Nur Du allein']);</pre>
Auch bei der Wertzuweisung an das Array sind keine Änderungen des Statements nach SQL-Standards notwendig.	
<pre>UPDATE Entleihungen SET BuchListe = BuchListe ARRAY ['Dick und Doof', 'Das Boot'] WHERE Entleiher = 'Fritz Klein';</pre>	→ nicht funktionsfähig
Es ist nicht gelungen den Konkatenationsoperator unter PostgreSQL zu realisieren	
<pre>SELECT BuchListe [4] FROM Entleihungen WHERE CARDINALITY (BuchListe) >= 5;</pre>	<pre>SELECT BuchListe [2] FROM Entleihungen;</pre>
Da der Update-Befehl nicht durchgeführt werden kann, wird die Abfrage der Arraywerte vereinfacht realisiert. Hierbei muss auf die Angabe des ‚CARDINALITY‘-Operators verzichtet werden.	

3.5.1.4 CLOB- und BLOB-Datentyp

Statement nach SQL-Standard	Statement unter PostgreSQL
<pre>CREATE TABLE Buecher (BuchID INTEGER PRIMARY KEY, Titel VARCHAR (200), Zusammenfassung CLOB (32 K), BuchText CLOB (20 M), FilmZumBuch BLOB (2 G));</pre>	<pre>CREATE TABLE Buecher (BuchID INTEGER PRIMARY KEY, Titel VARCHAR (200), Zusammenfassung TEXT, BuchText TEXT, FilmZumBuch BYTEA);</pre>
<p>Über PostgreSQL lassen sich auch große Datentypen verwalten. Allerdings werden die Typen ‚BLOB‘ und ‚CLOB‘ nicht unterstützt. Ein äquivalenter Datentyp zu BLOB stellt der Typ ‚BYTEA‘ (Byte Array) dar. Eine Alternative zu CLOB stellt der Datentyp TEXT dar. Über diesen Datentyp lassen sich Datenmengen bis zu einem Gigabyte verwalten.</p>	

3.5.2 XML-Dokumente

In diesem Unterkapitel werden die Abweichungen zu XML-Dokumente dargestellt.

Statement nach SQL-Standard	Statement unter PostgreSQL
<pre>CREATE TABLE Abteilung (AbtNr INTEGER PRIMARY KEY, AbtName VARCHAR (20), AbtDoku XML);</pre>	<pre>CREATE TABLE Abteilung (AbtNr INTEGER PRIMARY KEY, AbtName VARCHAR (20), AbtDoku XML);</pre>
<p>Der XML-Datentyp lässt sich unter PostgreSQL standardkonform umsetzen.</p>	
<pre>INSERT INTO Abteilung VALUES (111, 'Informatik', '<Abteilung Name="Informatik"> <Budget>1.000.000</Budget> <Leiter>Karl Schreier</Leiter> <Mitarbeiter>Bernd Diener</Mitarbeiter> <Mitarbeiter>Ute Contra</Mitarbeiter> </Abteilung>');</pre>	<pre>INSERT INTO Abteilung VALUES (111, 'Informatik', '<Abteilung name ="Informatik"> <Budget>1.000.000</Budget> <Leiter>Karl Schreier</Leiter> <Mitarbeiter>Bernd Diener</Mitarbeiter> <Mitarbeiter>Ute Contra</Mitarbeiter> </Abteilung>');</pre>
<p>Auch bei dem Einfügen von konkreten Werten müssen keine Änderungen des Statements vorgenommen werden.</p>	

Statement nach SQL-Standard	Statement unter PostgreSQL
<pre>CREATE TABLE Berichte (ID INTEGER PRIMARY KEY, Inhalt XML);</pre>	<pre>CREATE TABLE Berichte (ID INTEGER PRIMARY KEY, Inhalt XML);</pre>
<p>Auch diese Tabellenerzeugung ist konform des SQL-Standards.</p>	

Wie bei den beiden anderen DBS konnte auch bei PostgreSQL nicht alle Themenbereiche ergebnisführend bearbeitet werden. Begründet liegt das ebenfalls an einem streng limitierten Zeitfenster und ebenfalls an großen Abweichungen vom SQL-Standard. Die Bereiche Vererbung in Typ- und Tabellen-Hierarchien, ROW-Datentyp, Benutzerdefinierte Routinen und XML-Dokumente wurden daher weitestgehend nicht bearbeitet.

Bei PostgreSQL fielen die sehr einfache und intuitive Systeminstallation sowie die Ersteinrichtung sehr positiv auf. Die Benutzeroberfläche ist äußerst intuitiv und übersichtlich gestaltet. Lediglich der Objektbrowser wirkt anfangs etwas überladen. Ein weiterer Kritikpunkt ist, dass in den Standardeinstellungen keine benutzerdefinierten Datentypen über den Objektbrowser dargestellt werden. Diese müssen zunächst über die Optionen angezeigt werden.

4 Schlussbetrachtung und Ausblick

Das Ziel dieser Seminararbeit bestand in der Installation dreier repräsentativer relationalen DBS (Oracle, DB2 und PostgreSQL) und der anschließenden Untersuchung, inwiefern die Syntax dieser Systeme vom SQL-Standard abweicht. Die Erkenntnisse sollten hierbei dokumentiert werden, um die Nachvollziehbarkeit zu gewährleisten.

Zur Erreichung dieser Zielsetzung wurden die Systeme zunächst heruntergeladen, installiert und eingerichtet. Dieser Schritt gestaltete sich unerwartet komplex und zeitintensiv. Zudem konnten bereits in diesem primären Arbeitsschritt große Unterschiede zwischen den drei DBS ausgemacht werden. Nachdem jedes System funktionsfähig war, übernahm jedes Teammitglied ein DBS und überprüfte dieses zunächst auf syntaktische Abweichungen der vom Auftraggeber erhaltenen SQL-Statements. Schnell zeigte sich, dass alle untersuchten Systeme vom Standard abweichen und nur wenige SQL-Statements standardkonform ausgeführt werden können. Die Recherche nach den systemspezifischen Abweichungen gestaltete sich zudem sehr aufwendig. Die Handbücher der jeweiligen Systeme gaben einen ersten Anhaltspunkt, allerdings waren diese für die zu untersuchenden Statements keine erfolgreiche Unterstützung. Aus diesem Grund wurden für die DBS Oracle und DB2 die Hersteller kontaktiert und um hilfreiche Informationen gebeten. Parallel dazu erfolgte der Besuch der Ausbildungsunternehmen der Teammitglieder, um Hinweise zu syntaktischen Besonderheiten der DBS zu erhalten. Letzteres erwies sich allerdings als weniger hilfreich, da die jeweiligen Unternehmen keine objektrelationalen Erweiterungen verwenden. Die Hersteller der DBS Oracle und DB2 boten ihre Kooperation an, welche allerdings aufgrund des beschränkten Zeitfensters nur ansatzweise genutzt werden konnte.

Als Ergebnis der Untersuchungen konnten die aufgeführten Erkenntnisse über die einzelnen Themenbereiche erlangt werden. Allerdings war es nicht möglich mit jedem System die SQL-Statements vollumfassend zu untersuchen, sodass vor allem die Untersuchungen in dem Bereich der benutzerdefinierten Routinen zu keinem vorzeigbaren Ergebnis führte.

Bewertung

An dieser Stelle soll noch ein Vergleich der drei untersuchten DBS erfolgen, um die gemachten Erfahrungen in den Bereichen Installation/Ersteinrichtung, Benutzeroberfläche/Usability und den Abweichungen vom Standard darzustellen.

Die Installation der Systeme Oracle (Oracle SQL Developer) und DB2 (IBM Data Studio) gestalteten sich sehr kompliziert und zeitaufwendig. Der Grund hierfür ist vor allem der Umstand, dass zunächst das DBMS installiert werden muss und anschließend eine Frontend-Anwendung zur visuellen Darstellung und Verwaltung. Die Installation des Systems PostgreSQL

(pgAdmin III) gestaltet sich hingegen viel komfortabler, da das Installationspaket bereits alle notwendigen Komponenten beinhaltet und somit unmittelbar nach der Installation mit dem DBS gearbeitet werden kann. Die Ersteinrichtung ist bei allen drei Systemen ähnlich komfortabel gestaltet, sodass hierbei kaum Unterschiede auszumachen sind. Daher stellt PostgreSQL bezüglich der Installation und Ersteinrichtung das komfortabelste System dar. Oracle und DB2 folgen auf einem geteilten zweiten Rang.

Alle verwendeten DBS lassen äußerst intuitiv und benutzerfreundlich bedienen. Die Menüs sind übersichtlich gestaltet und die Aufteilung der einzelnen Oberflächenelementen logisch angeordnet. Die Softwareergonomie ist somit bei jedem einzelnen System äußerst positiv zu bewerten. Keines der Systeme ist hierbei besonders negativ im Vergleich zu den anderen aufgefallen.

Bei der Überprüfung der SQL-Statements auf Standardkonformität wurde festgestellt, dass jedes System an vielen Stellen vom Standard abweicht. Dennoch existieren Statements, die ohne Änderungen standardkonform ausgeführt werden können. Daher fällt es schwer ein System besonders zu empfehlen, da der Rechercheaufwand sowohl bei Oracle als auch bei DB2 und PostgreSQL ähnlich aufwändig ist. Zudem sind die Abweichungen in den einzelnen DBS abhängig vom jeweiligen Thema. Die Ergebnisse dieser Seminararbeit verdeutlichen diesen Umstand. Somit muss sich jede Interessierte Person bzw. jedes Interessierte Unternehmen selbst ein Bild vom Abweichungsgrad des Standards machen. Abhängig von den Themenstellungen, die umgesetzt werden sollen, muss entschieden werden welches DBS sich eignet.

Ausblick

Die mit dieser Arbeit gewonnen Erkenntnisse dienen als Grundlage, um vertiefende Folgearbeiten durchzuführen zu können. Es bietet sich an, die Kooperationsangebote von Oracle und IBM anzunehmen und den Kontakt zu intensivieren, um künftige Arbeiten an dieser Thematik vertieft fortführen zu können. Hierbei sollten die aus Komplexitäts- und Zeitgründen nicht bearbeiteten, in Anhang 4 beigefügten, XML-Statements aufgegriffen und verfolgt werden. Mit Blick auf die herausgearbeiteten Ergebnisse wird deutlich, dass bestimmte Themenbereiche, aufgrund des begrenzten Zeitumfangs, nicht in einem für das Projektteam zufriedenstellenden Ausmaß bearbeitet werden konnte. Somit ist ein hohes Potential für zukünftige, auf diese Arbeit aufbauende Seminararbeiten vorhanden. Es bleibt festzuhalten, dass durch diese Seminararbeit wertvolle Impulse für die Dokumentation der Syntaxabweichung in DBS zu objektrelationalen Erweiterungen gegeben wurden, um die Nutzung in der Praxis zu ermöglichen.

Anhang

Anhangverzeichnis

Anhang 1: Anforderungen eines Datenbanksystems	53
Anhang 2: Anforderungen an ein relationales DBS.....	54
Anhang 3: Übersicht Statementbestandteile zu objektrelationale DBS	55
Anhang 4: Weitere SQL-Statements zu XML.....	56
Anhang 5: Installationsanleitung Oracle	58
Anhang 6: Installationsanleitung DB2.....	63
Anhang 7: Installationsanleitung PostgreSQL.....	70

Anhang 1: Anforderungen eines Datenbanksystems³⁶

Datenunabhängigkeit	Die Daten können unabhängig von der Anwendung bearbeitet werden. Das bedeutet, dass wenn die Daten von einem Programm erzeugt wurden, können diese dann von anderen Programmen wieder gelesen werden. Es gibt aber auch die physische Datenunabhängigkeit und die logische Datenunabhängigkeit. Bei der physischen geht es bspw. um die mögliche Änderung der Speicherungsformen, ohne dass Anwendungsprogrammen geändert werden müssen und bei der logischen geht es um die Änderung der Datenstruktur, hierbei dürfen solche Änderungen nicht dazu führen, dass bisherige Anwendungsprogramme geändert werden müssen.
Mehrbenutzerbetrieb	Es soll vermieden werden, dass bei konkurrierenden Zugriffen durch mehrere Benutzer Probleme entstehen. Es muss deshalb die Sicherheit gewährleistet werden, dass Programme, die bestimmte Teile der Datenbank parallel benutzen, nicht fehlerhafte Ausgaben oder Änderungen bewirken.
Datenintegrität	Alle Daten müssen korrekt und vollständig sein, hierbei müssen bestimmte Plausibilitäten berücksichtigt werden.
Datensicherheit	Die Wiederherstellung eines korrekten Datenbankzustandes muss nach Störungen durch Hardware- oder Softwarefehler gewährleistet werden. Programme für die Datensicherung und die Zugriffsprotokollierung müssen vorhanden sein.
Datenschutz	Alle Daten müssen vor unbefugtem Zugriff geschützt sein. Für diese Sicherheit müssen die Berechtigungen für jeden Benutzer für den Zugriff auf Daten festgelegt sein und in der Datenbank archiviert werden können.
Effizienz	Die Gewährleistung eines guten Antwortzeitverhaltens bei einem DBS muss bei Abfragen und Änderungen der Datenbank durch optimierte Zugriffen vorhanden sein.

Tab. A- 1: Anforderungen an ein DBS

In Tab. A- 1 werden die Anforderungen eines DBS mit den jeweiligen Erläuterungen aufgezeigt

³⁶ Vgl. Abst, D./Mülder, W. (2010), S. 614

Anhang 2: Anforderungen an ein relationales DBS³⁷

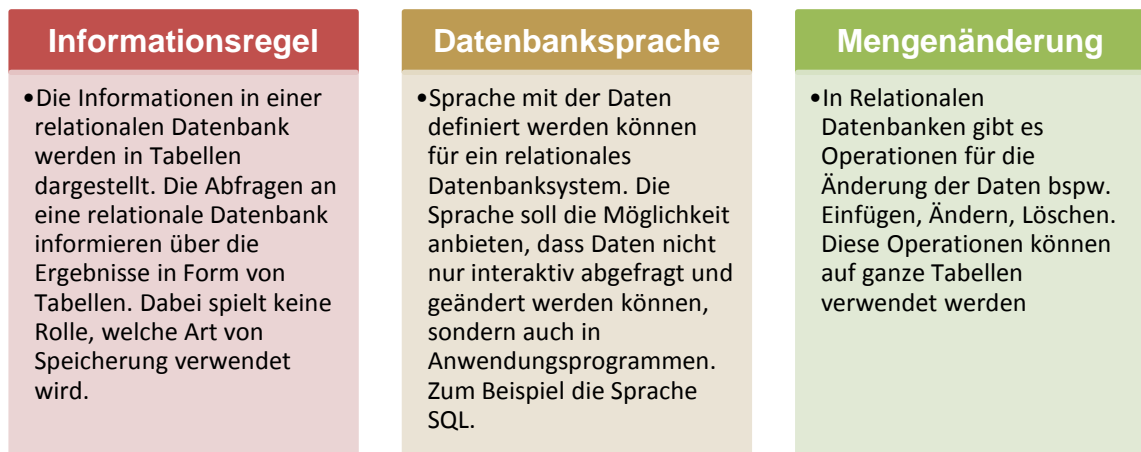


Abb. A- 1: Drei Anforderungen an relationale DBS³⁸

In Abb. A- 1 sind die drei Anforderungen an ein relationales DBS abgebildet.

³⁷ Vgl. Abts, D./Mülder, W. (2010), S. 617

³⁸ Vgl. ebenda

Anhang 3: Übersicht Statementbestandteile zu objektrelationale DBS

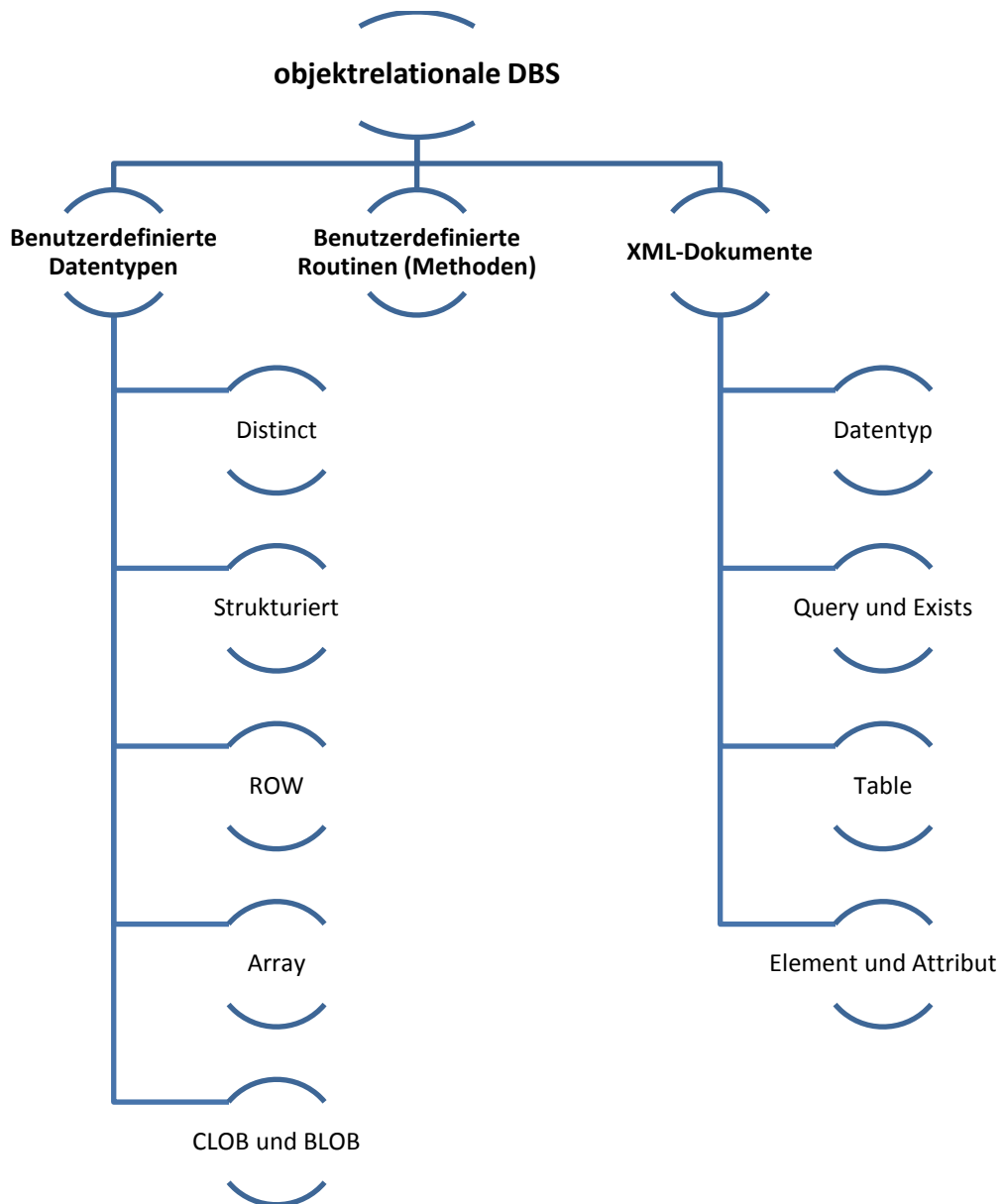


Abb. A- 2: Gliederung der standardisierten SQL-Statements

In Abb. A- 2 ist abgebildet, wie sich die zu untersuchenden SQL-Statements gliedern lassen.

Anhang 4: Weitere SQL-Statements zu XML

XMLQUERY und XMLEXISTS

```
SELECT ID,  
XMLQUERY ( '$Inh//Autor' PASSING Inhalt AS "Inh" ) AS Autorennamen  
FROM Berichte;
```

```
CREATE TABLE dept  
( deptid CHAR (8) PRIMARY KEY,  
  Deptdoc XML);
```

```
SELECT deptid,  
XMLQUERY ( '$doc/dept[@bldg="101"]/employee' PASSING deptdoc AS "doc") AS empinfo  
FROM dept;
```

```
SELECT deptid,  
XMLQUERY ( '$d/dept/name' PASSING deptdoc AS "d" ) AS empinfo  
FROM dept  
WHERE deptid LIKE 'PR%'  
AND XMLEXISTS ( '$dd/dept[@bldg = "101"]' PASSING deptdoc AS "dd" );
```

XMLTABLE

```
SELECT X.*  
FROM dept,  
XMLTABLE ( '$d/dept/employee' PASSING deptdoc AS "d"  
  COLUMNS  
    empID INTEGER PATH '@id',  
    firstname VARCHAR(30) PATH 'name/first',  
    lastname VARCHAR(30) PATH 'name/last',  
    office INTEGER PATH 'office' ) AS X;
```

```
CREATE TYPE VerlagTyp AS  
( Name VARCHAR (40),  
  Ort VARCHAR (40) )  
NOT FINAL;
```

```
CREATE TYPE PreisTyp AS  
( Betrag DECIMAL (5,2),  
  Waehrung CHAR (3) CHECK (VALUE IN ("CHF","EUR","USD")) )  
NOT FINAL;
```

```
CREATE TYPE BuchTyp AS  
( Autor VARCHAR (40) ARRAY[10] NOT NULL,  
  Titel VARCHAR (40) NOT NULL,  
  Verlag VerlagTyp NOT NULL,  
  Auflage INTEGER,  
  Preis PreisTyp,  
  ISBN VARCHAR (40) NOT NULL,  
  Jahr DATE NOT NULL )  
NOT FINAL  
REF IS SYSTEM GENERATED;
```



```
CREATE TABLE Buch OF BuchTyp
( REF IS BuchOid SYSTEM GENERATED,
  PRIMARY KEY (ISBN) );
```

```
INSERT INTO Buch (Autor, Titel, Verlag, Auflage, Preis, ISBN, Jahr)
SELECT Doku.*
FROM XML-Dokumente,
XMLTABLE ( '$doc/Buecherliste/Buch' PASSING Dokument AS "doc"
  COLUMNS
  Autor VARCHAR (40) ARRAY[10] PATH 'Autor',
  Titel VARCHAR (40) PATH 'Titel',
  Verlag VerlagTyp PATH 'Verlag | Verlag/@Ort',
  Auflage INTEGER PATH 'Auflage',
  Preis PreisTyp PATH 'Preis/*',
  ISBN VARCHAR(40) PATH '@ISBN',
  Jahr DATE PATH '@Jahr' )
AS Doku;
```

XMLELEMENT und XMLATTRIBUTES

```
CREATE TABLE Mitarbeiter
( Name VARCHAR (30) PRIMARY KEY,
  Gehalt DECIMAL (5,2));
```

```
SELECT XMLEMENT
( NAME "Mitarbeiter",
  XMLATTRIBUTES (Gehalt),
  Name ) AS Element
FROM Mitarbeiter;
```

```
CREATE TABLE Schauspieler
( PNr INTEGER PRIMARY KEY,
  Manager VARCHAR (30) );
```

```
CREATE TABLE Person
( PNr INTEGER PRIMARY KEY,
  Vorname VARCHAR (15),
  Nachname VARCHAR (15) );
```

```
SELECT p.PNr AS Nr,
XMLEMENT
( NAME „Schauspieler“,
  XMLEMENT (NAME „Nachname“, Nachname),
  XMLEMENT (NAME „Vorname“, Vorname),
  XMLEMENT (NAME „Auszeichnung“),
  XMLEMENT (NAME „Manager“, Manager) ) AS SchauspielerDaten
FROM Schauspieler s, Person p
WHERE p.PNr = s.PNr;
```

```
SELECT p.PNr AS Nr,
XMLEMENT (NAME "Schauspieler",
XMLATTRIBUTES ( Nachname, Vorname, Manager AS "Manager" ) ) AS SchauspielerDaten
FROM Schauspieler s, Person p
WHERE p.PNr = s.PNr;
```

Anhang 5: Installationsanleitung Oracle

Die Installation des DBS Oracle besteht aus insgesamt zwei Arbeitsschritten. Zunächst muss das Backend installiert werden bevor anschließend das Frontend als visuelle Administrationsoberfläche installiert werden kann.

Schritt 1: Download Oracle Database Express Edition 11g Release 2:

Zunächst muss das Backend des DBS heruntergeladen werden. Bevor die Datei heruntergeladen werden kann, ist zunächst eine Anmeldung bei Oracle erforderlich. Der Download erfolgte über die Internetseite Oracle (<http://www.oracle.com/technetwork/database/database-technologies/express-edition/downloads/index.html>). Hierbei wurde die Version ‚Oracle for Windows x32‘ gewählt, wie in Abb. A- 3 zu sehen ist.

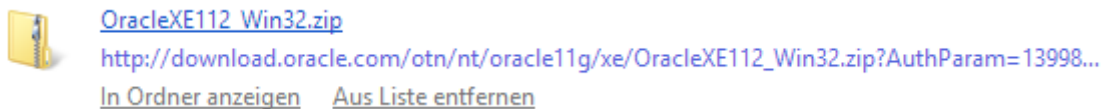


Abb. A- 3: Download Oracle Database Express Edition 11g Release 2

Schritt 2: Installation Oracle Database Express Edition 11g:

Nach erfolgreichem Download muss die heruntergeladene Datei zunächst entpackt werden. Anschließend muss in dem Ordner ‚DISK1‘ die Datei ‚setup.exe‘ ausgeführt werden um die Installation zu starten (Siehe Abb. A- 3).

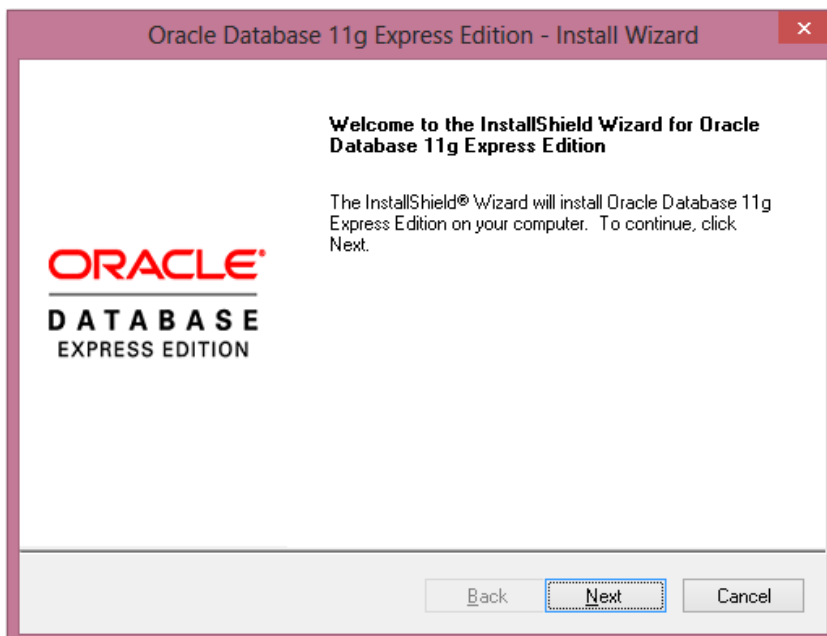


Abb. A- 4: Installationsmaske Oracle

Nach einem Klick auf die Schaltfläche ‚Next‘ muss dem in Abb. A- 5 abgebildeten Lizenzabkommen zugestimmt werden, um mit der Installation fortfahren zu können.

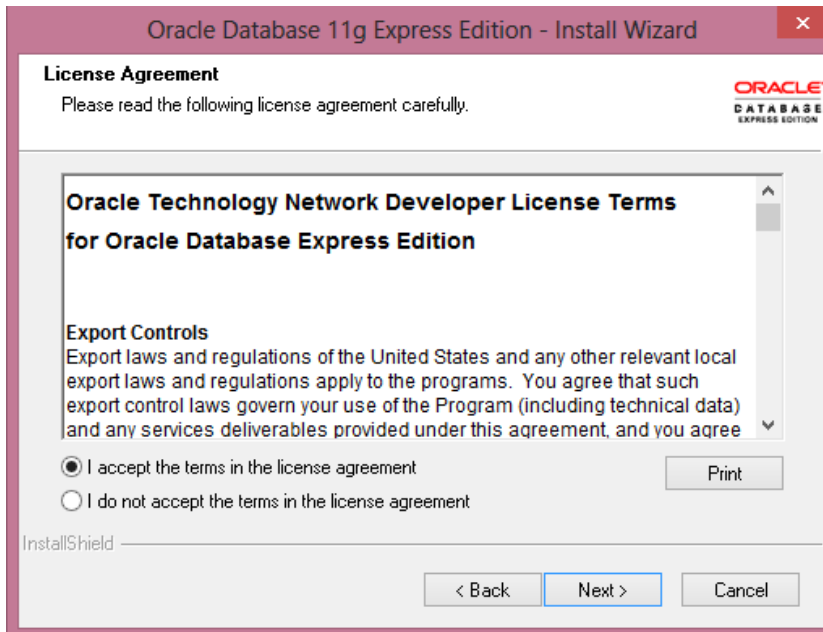


Abb. A- 5: Zustimmung des Lizenzabkommens Oracle

Anschließend besteht die Möglichkeit den Installationspfad festzulegen. Wie in Abb. A- 6 zu sehen ist, wurde hierbei die Default-Einstellung beibehalten.

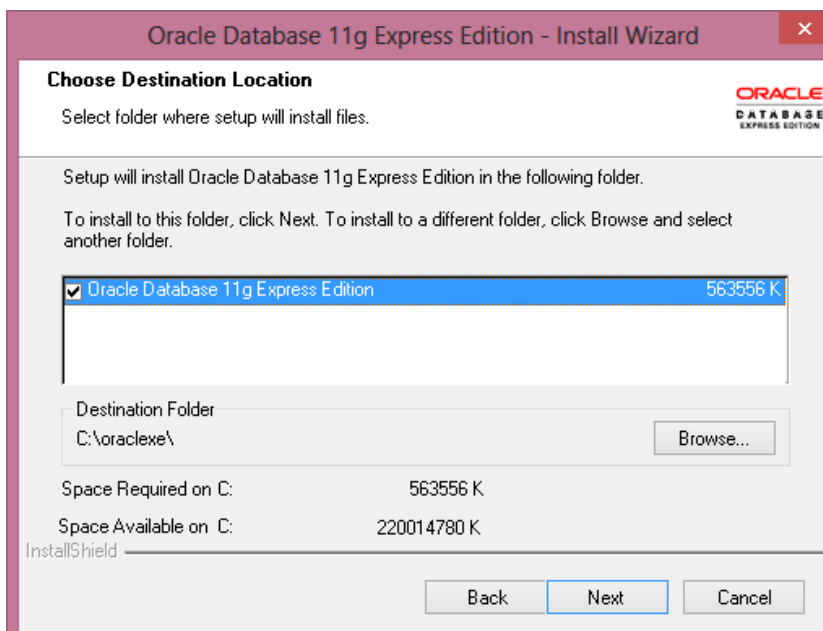


Abb. A- 6: Bestimmung des Installationspfads Oracle

Im Anschluss daran muss ein Passwort für die Datenbank vergeben werden. Dieser Vorgang ist in Abb. A- 7 dargestellt. Dieses Passwort sollte gut aufbewahrt werden, da dieses für die Verbindung mit der Datenbank benötigt wird.

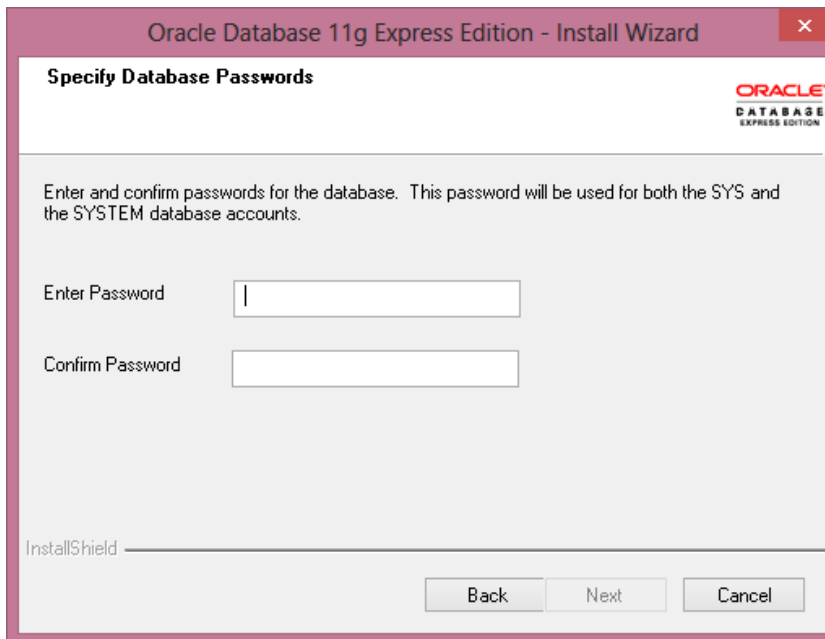


Abb. A- 7: Bestimmung des Datenbank-Passworts Oracle

Nachdem alle notwendigen Angaben gemacht wurden, ist das System bereit zur Installation. Der Installationsprozess wird hierbei, wie in Abb. A- 8 dargestellt, über den Button ‚Install‘ gestartet.

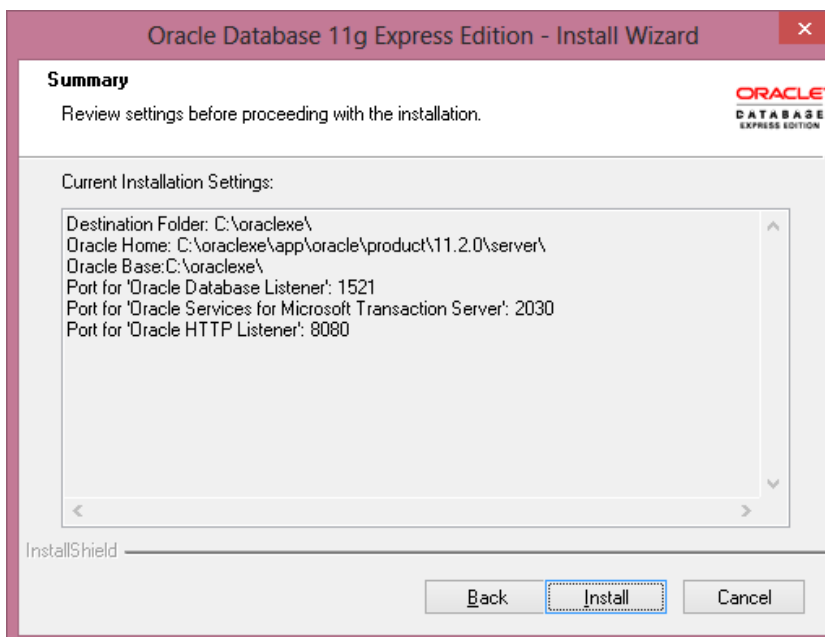


Abb. A- 8: Start des Installationsprozesses Oracle

Während der Installation trat folgende, in Abb. A- 9 dargestellte, Fehlermeldung auf, welche allerdings auf den weiteren Verlauf der Installation und Inbetriebnahme keine Auswirkungen zeigte.

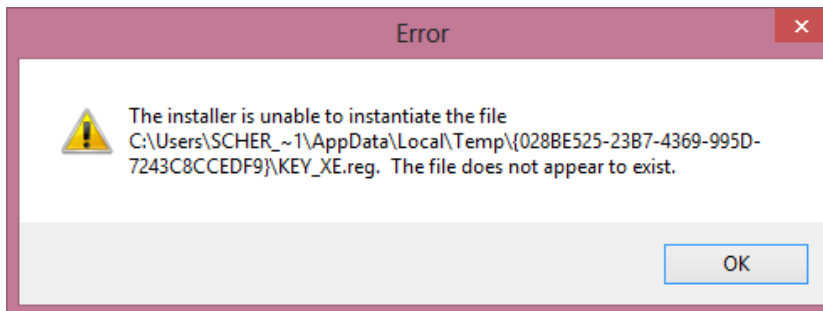


Abb. A- 9: Fehlermeldung bei der Installation von Oracle

Nach Bestätigung der Fehlermeldung durch ‚OK‘ wird die Datenbank konfiguriert und das Setup abgeschlossen. (Siehe Abb. A- 10)

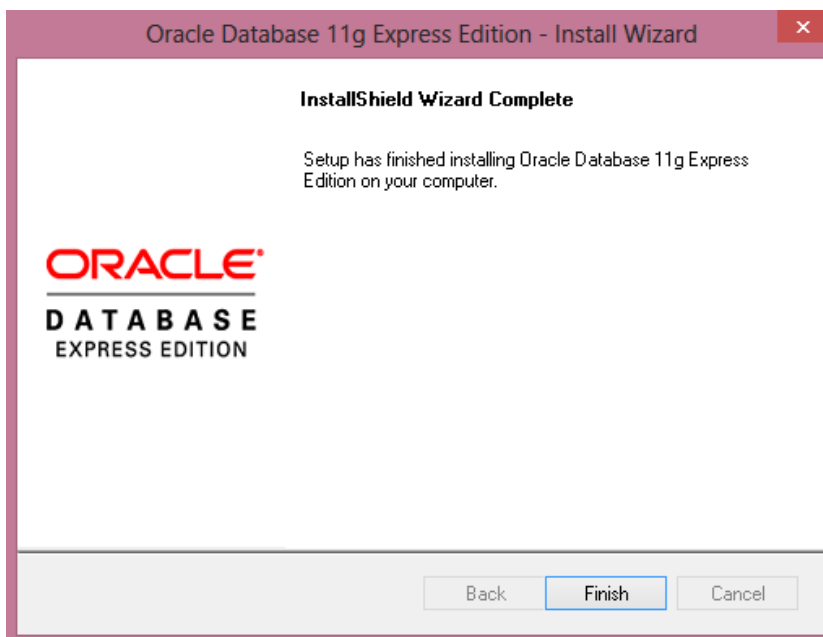


Abb. A- 10: Abschluss der Oracle-Installation

Schritt 3: Download Oracle SQL Developer 4.0.2.15.21:

Nach Download des Backends des DBS muss als nächster Schritt ein Tool zum Erstellen und Bearbeiten von Datenbanken heruntergeladen werden. Das Tool stellt hierbei der Oracle SQL Developer dar. Der Download erfolgte über die Internetseite Oracle (<http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html>). Hierbei wurde die Version ‚Oracle SQL Developer Windows 32/64-bit‘ gewählt, wie in Abb. A- 11 zu sehen ist.

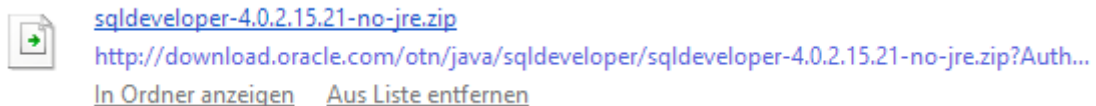


Abb. A- 11: Download Oracle SQL Developer 4.0.2.15.21

Schritt 4: Installation Oracle SQL Developer 4.0.2.15.21:

Nach erfolgtem Download muss die heruntergeladene Datei zunächst entpackt werden. Eine Installation ist hierbei nicht notwendig. Anschließend muss das ‚Java Platform Standard Edition Development Kit‘ (JDK) in der Version 7 oder höher heruntergeladen werden, um den SQL Developer ausführen zu können. Der Download der 32Bit Version erfolgte über die Internetseite Oracle (<http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>). Hierbei wurde die Version JDK7 gewählt, wie in Abb. A- 12 zu sehen ist.



Abb. A- 12: Download JDK 7

Nach Installation des JDK 7 muss der entpackte Ordner des SQL Developer lediglich in den Installationspfad der Oracle Database verschoben werden. Durch einen Doppelklick auf die Datei sqldeveloper.exe lässt sich die Anwendung öffnen.

Anhang 6: Installationsanleitung DB2

Um das DBS DB2 zu installieren sind zwei Arbeitsschritte notwendig. Zu Beginn muss das Backend „DB2 Express-C“ installiert werden, bevor anschließend das Frontend „IBM Data Studio“ als visuelle Administrationsoberfläche installiert werden kann. Beide Schritte sind mit Admin-Berechtigungen durchzuführen und werden im Folgenden beschrieben.

Schritt 1: Download DB2 Express-C Version 10.5.1

Zunächst muss der Download von DB2 Express-C erfolgen. Dies geschieht direkt über die Internetseite des Entwicklers IBM (<http://www-01.ibm.com/software/data/db2/express-c/download.html>). Hierbei wurde die 64-bit Variante in der aktuellsten Version 10.5.1 ausgewählt. Nach dem erforderlichen Anmeldeprozess, der für den Download notwendig ist, muss die heruntergeladene Datei entpackt werden.

Schritt 2: Installation DB2 Express-C Version 10.5.1

Durch einen Doppelklick auf die Datei „v10.5fp1_winx64_expc“ beginnt der Installationsprozess, indem sich das Launchpad öffnet. Unter dem Punkt „Produkt installieren“ sollte der Button „Neue Installation“ gedrückt werden, wie es in Abb. A- 13 dargestellt ist.



Abb. A- 13: DB2 Express-C Installationssetup

Anschließend sind im Folgebildschirm die Nutzungsbedingungen zu akzeptieren, bevor der Installationstyp gewählt werden muss. Hierbei wird die Standardvariante gewählt. Daraufhin kann der Dateispeicherort festgelegt werden, an welchem eine Antwortdatei erzeugt wird, in welcher alle Installationseinstellungen geschrieben werden. Es folgt die Angabe des Installationsverzeichnis von DB2 sowie die Einstellung zum Autostart. Hierbei wurde für diese Installation gewählt, dass das Programm nicht automatisch bei jedem Systemstart geöffnet

wird. Im Folgeschritt sind wie Abb. A- 14 zu entnehmen ist, die Benutzerinformationen anzugeben. Ein Benutzername und ein Kennwort sind zu vergeben. Diese sollten unbedingt notiert werden, da sie später bei der Einrichtung der Datenbankverbindung wieder benötigt werden.

Benutzerinformationen für den DB2-Verwaltungsserver festlegen

Der DB2-Verwaltungsserver (DB2 Administration Server, DAS) wird auf Ihrem Computer ausgeführt, um die für die DB2-Tools erforderliche Unterstützung bereitzustellen. Geben Sie die erforderlichen Benutzerinformationen für den DAS an.

Es wird dringend empfohlen, dass Sie ein Konto für einen lokalen Benutzer oder Domänenbenutzer anstatt des Kontos für das lokale System verwenden. Weitere Details hierzu können Sie durch Anklicken von 'Hilfe' abrufen.

Konto für lokalen Benutzer oder Domänenbenutzer

Konto für lokales System

Dasselbe Konto für die übrigen DB2-Services verwenden

InstallShield

< Zurück Weiter > Abbrechen Hilfe

Abb. A- 14: Festlegung der Benutzerinformationen

Mit einem Klick auf ‚Weiter‘ können die Datenbankinstanzen ausgewählt werden, die installiert werden sollen. Die automatische Auswahl von DB2 wird beibehalten. Im darauf folgenden Bildschirm wird, wie in Abb. A- 15 dargestellt, eine Übersicht aller Installationseinstellungen gegeben, die wie bereits erwähnt in einer Antwortdatei gespeichert werden.

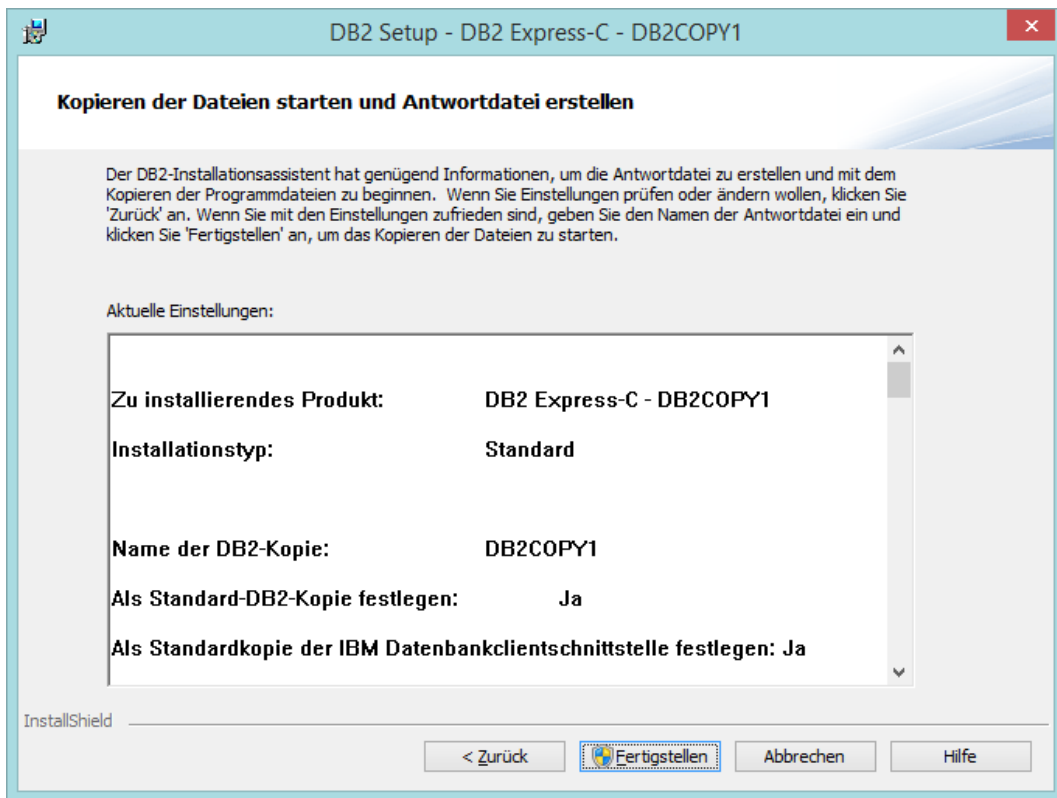


Abb. A- 15: Abschlussbildschirm der Installation von DB2 Express-C

Mit einem Klick auf „Fertigstellen“ beginnt die Installation, wonach das Backend von DB2 erfolgreich installiert wurde. Nach dem erfolgreichen Abschluss kann daraufhin die Installation des Frontend erfolgen.

Schritt 3: Beispieldatenbank erzeugen

Um im weiteren Verlauf auf einer Datenbank arbeiten zu können, muss eine Datenbank in DB2 angelegt werden. Hierfür gibt es eine spezielle Funktion, die „DB2 – Erste Schritte“ bezeichnet wurde. Diese ist aufzurufen und der in Abb. A- 16 dargestellte Button „Beispieldatenbank erstellen“ muss angeklickt werden.

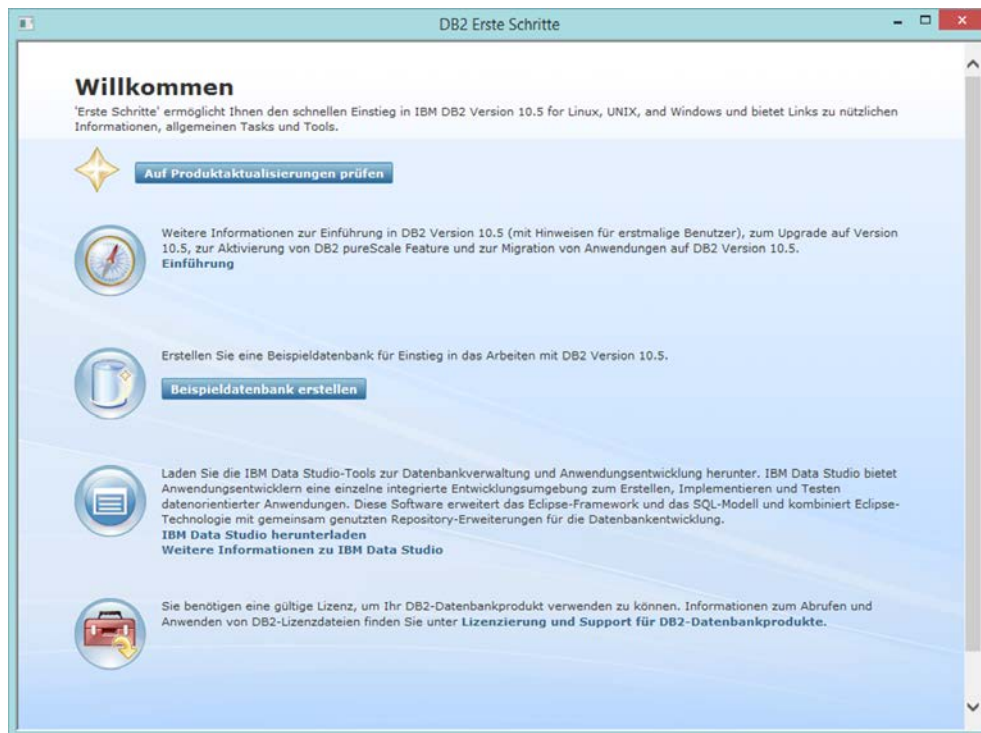


Abb. A- 16: Erste Schritte zur Anlage der Beispieldatenbank

Nachdem sich somit eine beispielhafte Datenbank erzeugt hat wird noch eine grafische Oberfläche für die Verwaltung benötigt. Hierfür wird IBM Data Studio zur Verfügung gestellt.

Schritt 4: Download IBM Data Studio 4.1.0.1

Das Client-System zur Verwaltung und Administration wird ebenfalls direkt von IBM auf der Internetseite <http://www.ibm.com/developerworks/downloads/im/data/> heruntergeladen. Die aktuellste Version 4.1.0.1 von IBM Data Studio wurde ausgewählt. Die damit heruntergeladene Datei „ibm_ds4101_win“ ist zu entpacken.

Schritt 5: Installation IBM Data Studio 4.1.0.1

Die in den entpackten Dateien enthaltene Setup-Datei ist auszuführen. Damit startet sich der IBM Data Studio Client. Unter dem Punkt „Produkt installieren oder aktualisieren“ muss „Verwaltungsinstallation oder –aktualisierung“ ausgewählt werden, wie Abb. A- 17 zu entnehmen ist.

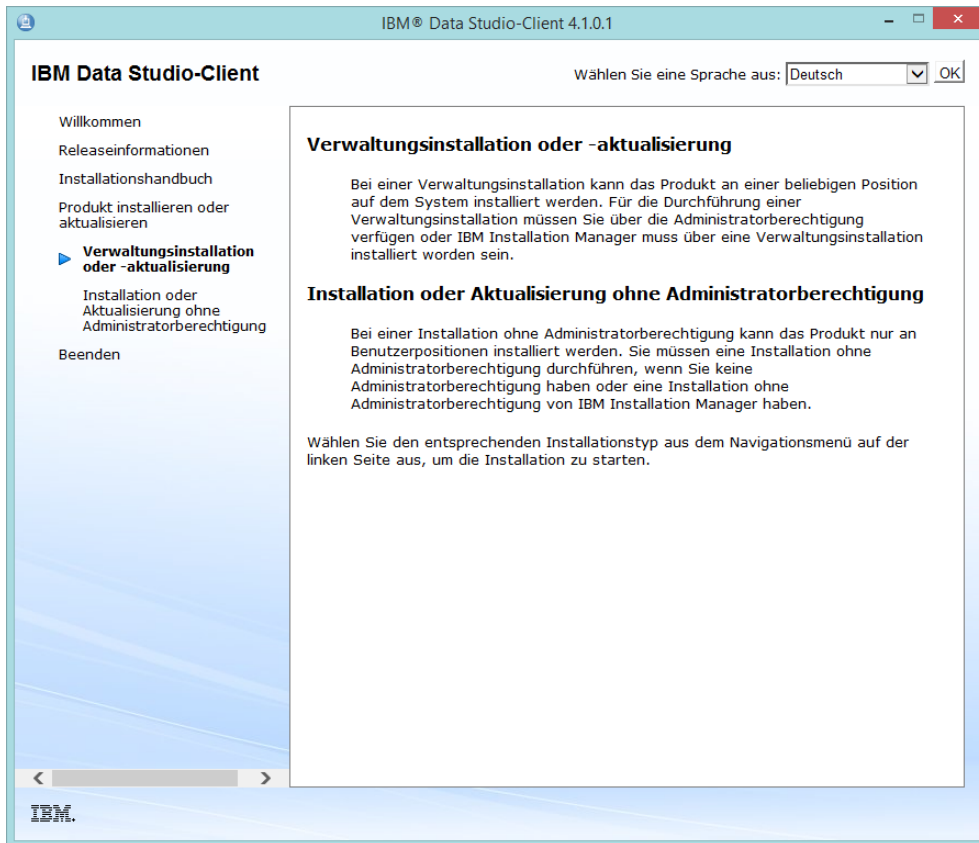


Abb. A- 17: Auszuwählende Option zur Installation des Installation Managers

Daraufhin startet sich das Setup des Installation Managers, welcher vor dem eigentlichen Frontend IBM Data Studio installiert werden muss. Hierbei können alle Standardeinstellungen zur Version, der Lizenzbedingungen und des Speicherortes beibehalten werden. Ist die Installation abgeschlossen öffnet sich der in Abb. A- 18 dargestellte Bildschirm zur Installation von IBM Data Studio.



Abb. A- 18: Startbildschirm des Installation Manager

Durch das Anklicken des Buttons „Installieren“ öffnet sich der Installationsprozess, in welchem Standardangaben zur Version, den Lizenzbedingungen und des Dateispeicherortes bestätigt werden. In Abb. A- 19 ist zu sehen, dass die 64-Bit Architektur gewählt wurde, wie bei der Backendinstallation. Dies liegt daran, dass ein Rechner der Autoren diese Architektur besitzt.

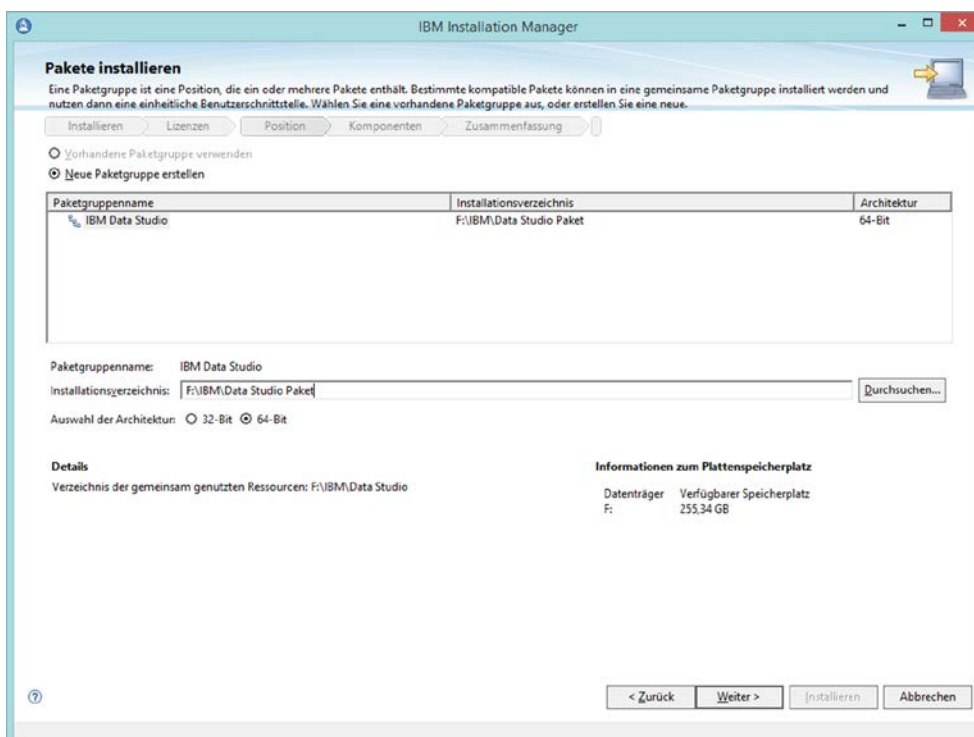


Abb. A- 19: Angabe des Speicherortes und der Architektur von IBM Data Studio

Im anschließenden Bildschirm wird die deutsche Sprache für die Installation ausgewählt und die zu installierenden Komponenten werden ausgewählt. Hier muss der Haken bei „SQL-Ausführung“ gesetzt werden, damit die entsprechenden Tools mit installiert werden. Für den Zugriff auf das Hilfesystem wird wie standardmäßig angegeben, ein Internetzugriff erlaubt. Alle gemachten Angaben erscheinen daraufhin im Zusammenfassungsbildschirm. Nun muss letztendlich nur noch ein Klick auf „Installation“ erfolgen. Nach der erfolgreichen Installation von IBM Data Studio als Frontends öffnet sich das Programm, indem der in Abb. A- 20 dargestellte Bildschirm erscheint.

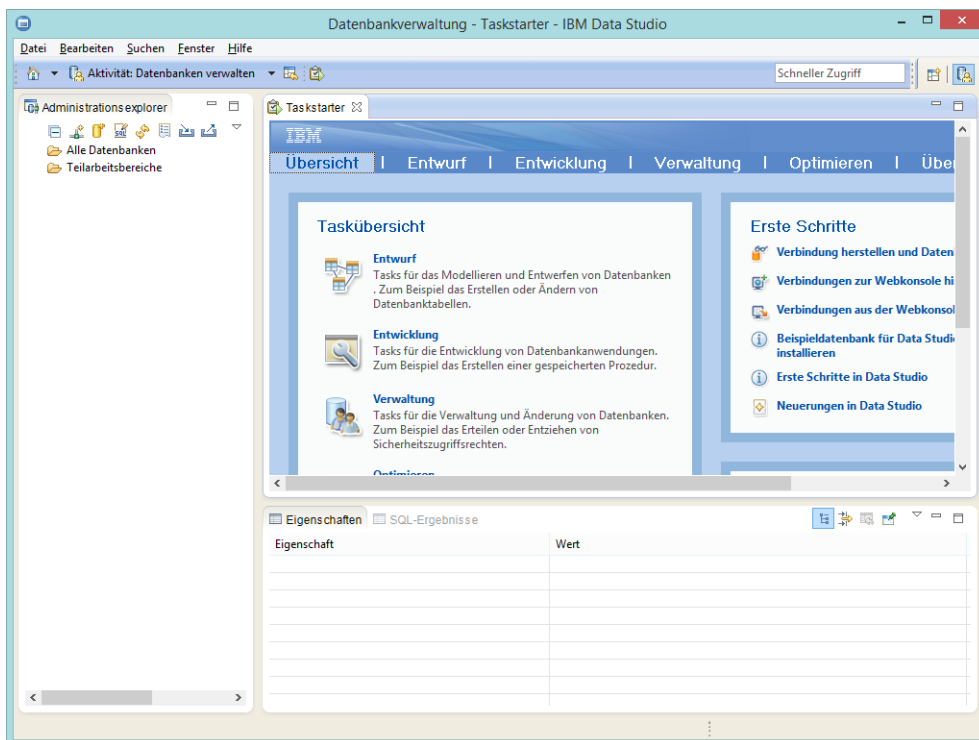


Abb. A- 20: Programmeinstieg des Frontends IBM Data Studio für DB2

Anhang 7: Installationsanleitung PostgreSQL

Schritt 1: Download PostgreSQL 9.3.4

Zunächst muss das DBS heruntergeladen werden. Der Download erfolgte über die Internetseite ENTERPRISEDB (<http://www.enterprisedb.com/products-services-training/pgdownload#windows>). Hierbei wurde die Version ‚WIN x86-32‘ gewählt, wie in Abb. A- 21 zu sehen ist.



Abb. A- 21: Download PostgreSQL 9.3.4

Schritt 2: Installation PostgreSQL 9.3.4

Nach erfolgreichem Download wird per Doppelklick auf die heruntergeladene Datei die Installation gestartet (Siehe Abb. A- 22).

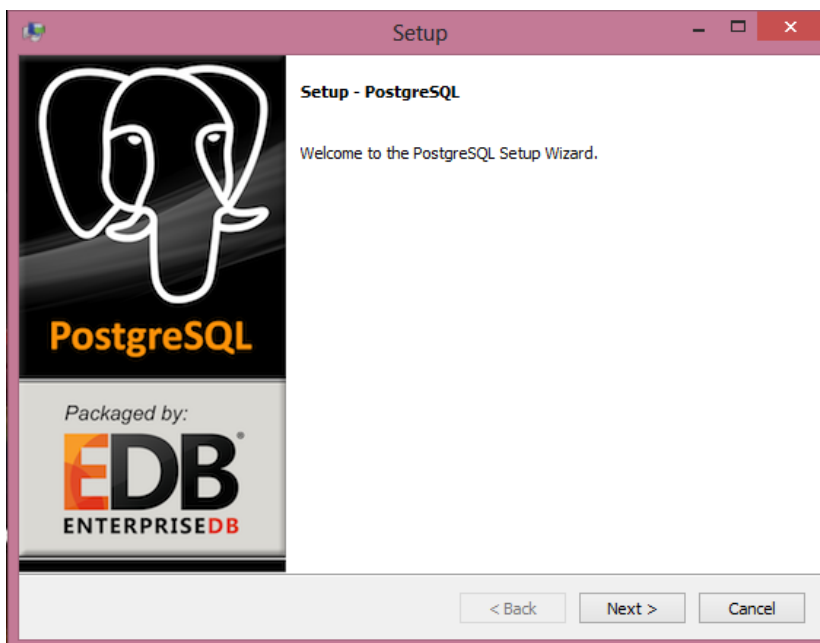


Abb. A- 22: Installationsmaske PostgreSQL

Nach einem Klick auf den Button ‚Next‘ besteht die Möglichkeit den Installationspfad festzulegen. Wie in Abb. A- 23Abb. A- 23 zu sehen ist, wurde auch bei dieser Installation der Default-Installationspfad beibehalten.

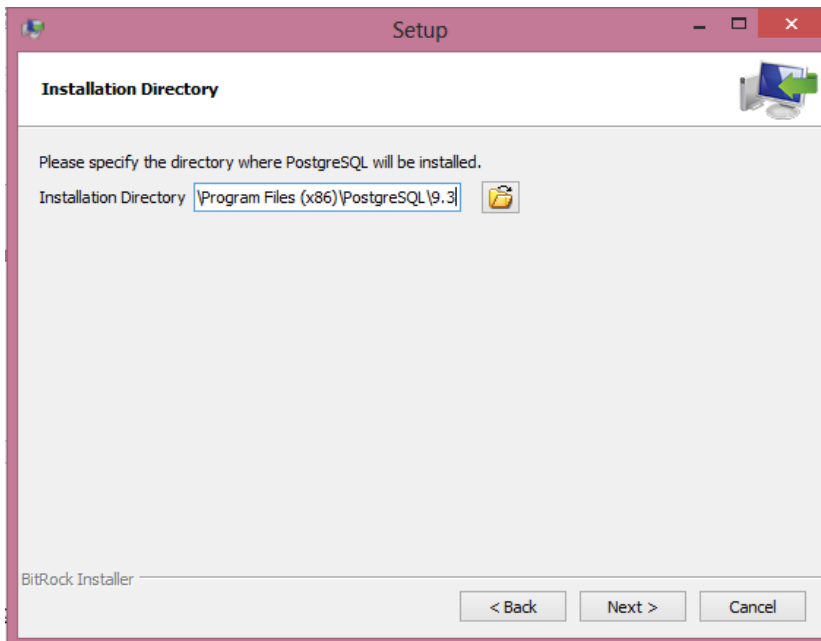


Abb. A- 23: Bestimmung des Installationspfads PostgreSQL

Im Anschluss daran muss ein Password für den sogenannten ‚database superuser‘ festgelegt werden. (Siehe Abb. A- 24) Unter einem database superuser ist der Datenbank-Administrator zu verstehen.

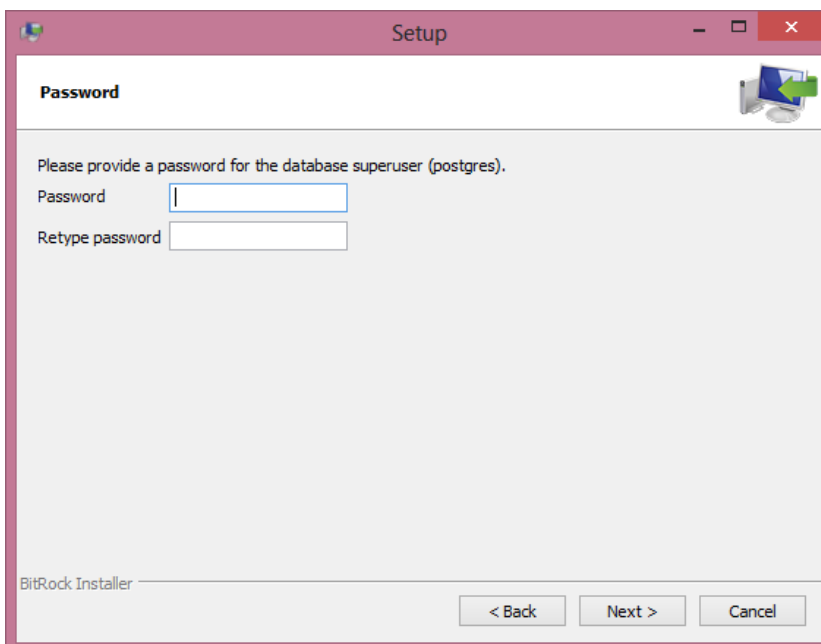


Abb. A- 24: Bestimmung des superuser-passworts PostgreSQL

Anschließend kann der Default-Port, über den der Datenbank-Server zu erreichen ist, geändert werden. Dieser ist standardmäßig auf den Wert ‚5432‘ festgelegt. (Siehe Abb. A- 25)

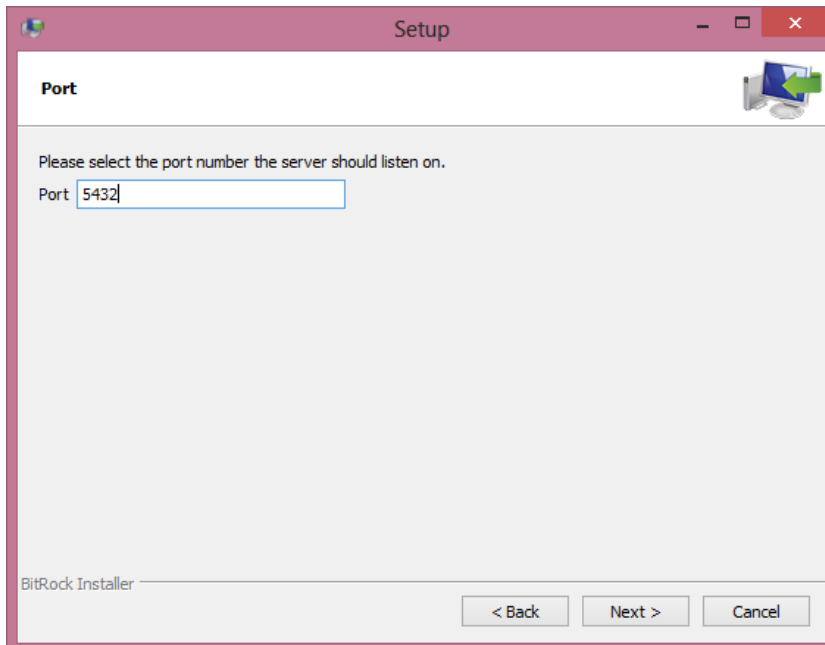


Abb. A- 25: Festlegung des Datenbank-Ports PostgreSQL

Danach lässt sich die Datenbankcluster-Umgebung bestimmen. Dieser wurde, wie in Abb. A- 26 abgebildet, ebenfalls auf den voreingestellten Wert ‚Default locale‘ belassen.

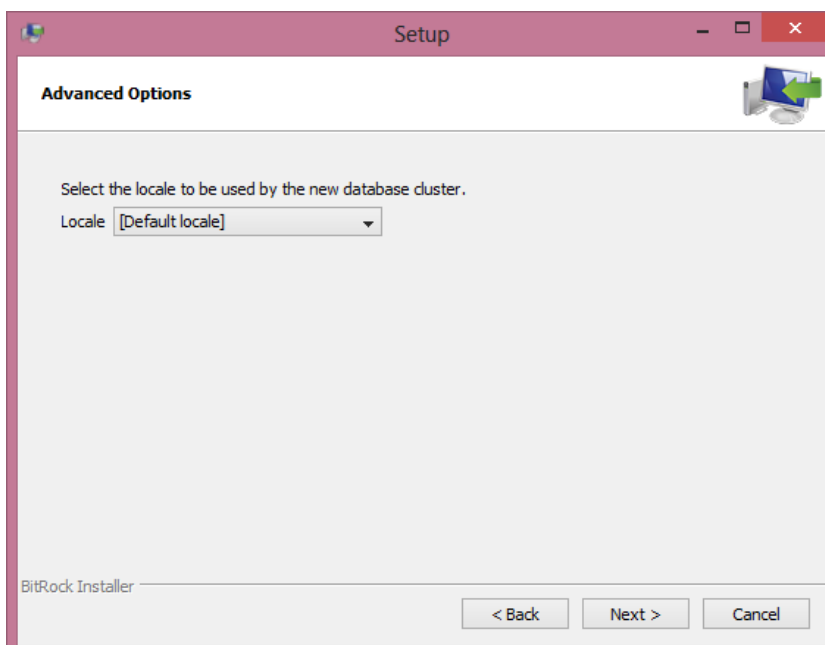


Abb. A- 26: Bestimmung der Datenbankcluster-Umgebung

Nachdem alle notwendigen Angaben gemacht wurden, ist das System bereit zur Installation und der in Abb. A- 27 abgebildete Bildschirm erscheint.

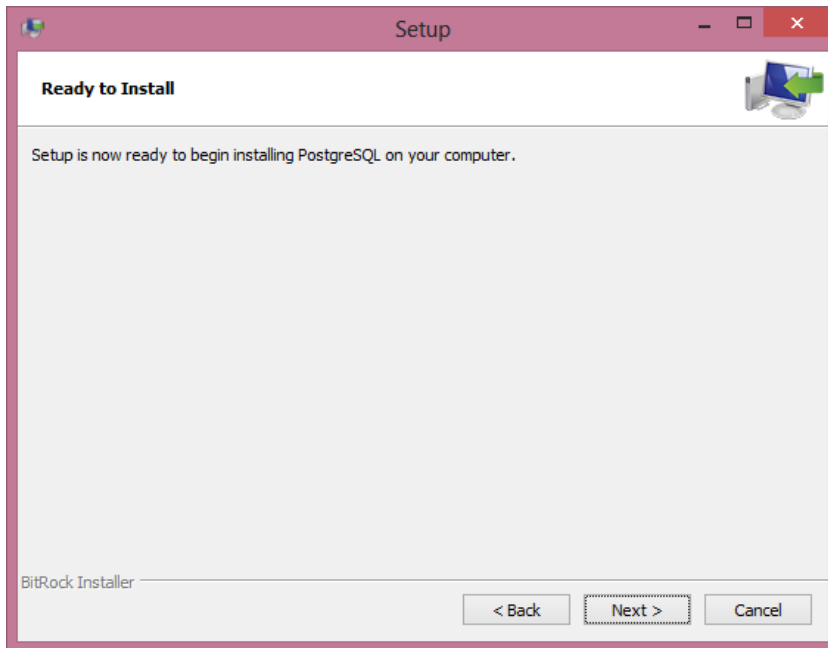


Abb. A- 27: Start des Installationsprozesses PostgreSQL

Nach Abschluss der Installation besteht die Möglichkeit die Anwendung ‚Stack Builder‘ zu installieren. Die Anwendung vereinfacht die Installation additionaler Software. Im Rahmen dieser Seminararbeit wurde allerdings auf die Installation der Anwendung ‚Stack Builder‘ verzichtet. Nachfolgende Abb. A- 28: Installation Stack Builder zeigt die Installationsmaske des ‚Stack Builder‘

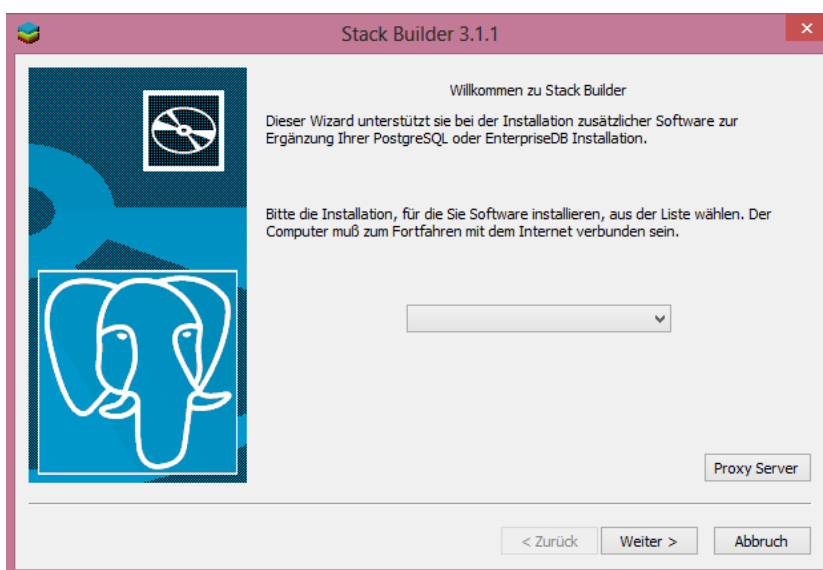


Abb. A- 28: Installation Stack Builder

Um die Installation abzuschließen, muss das System abschließend neu gestartet werden.

Quellenverzeichnisse

Literaturverzeichnis

- Abts, D. / Mülder, W. (2010):** Masterkurs Wirtschaftsinformatik, Kompakt, praxisnah, verständlich – 12 Lern- und Arbeitsmodule, Wiesbaden: Vieweg + Teubner
- Adar, M. (2013):** Das große Oracle Datenbank Einsteigerbuch, Ein strukturierter Einstieg in die Oracle Datenbankwelt, Norderstedt: Adar-Consult
- Cordts.S./**
- Blakowski, G./Brosius, G.(2011):** Datenbanken für Wirtschaftsinformatiker, Nach dem aktuellen Standard SQL:2008, Wiesbaden: Vieweg+Teubner Verlag
- Geisler, F. (2011):** Datenbanken: Grundlagen und Design, 4. Aufl., Heidelberg: Hüthig Jehle Rehm GmbH
- Kemper, A./Eickler, A. (2011):** Datenbanksysteme, Eine Einführung, 8. Aufl., München: Oldenbourg
- Kleinschmidt, P./Rank, C. (2005):** Relationale Datenbanksysteme, Eine praktische Einführung; mit zahlreichen Beispielen und Übungsaufgaben, 3. Aufl., Berlin, Heidelberg: Springer-Verlag
- Orhanovic, J./Grodtko, I./
Tiefenbacher, M. (2007):** DB2 Administration, Einführung, Handbuch und Referenz, München: Addison Wesley

- Panther, R. (2012):** Richtig einsteigen: Datenbanken entwickeln mit SQL Server 2012, Köln: O'Reilly Verlag GmbH & Co. KG
- Preiß, N. (2014):** Vorlesung Objektrelationale Datenbanken
- Throll, M./Bartosch, O. (2007):** Einstieg in SQL, 2. Aufl., Bonn: o. V.
- Türker, Can (2003):** SQL:1999 & SQL:2003, Objektrelationales SQL, SQLJ & SQL/XML, Heidelberg: Dpunkt-verl.

Verzeichnis der Internet- und Intranet-Quellen

- Datenbanken Online Lexikon (2010):** Create Type-Anweisung SQL, http://wikis.gm.fh-koeln.de/wiki_db/Datenbanken/CREATE-TYPE, Abruf: 09.06.2014
- Extrememarkup.com (o. J.):** Oracle Datenbanksystem, <http://www.extrememarkup.com/software/datenbanken/oracle-datenbanksystem>, Abruf: 08.06.2014
- IBM (o. J.):** DB2 Express Server Edition, <http://www-03.ibm.com/software/products/de/db2-express-server-edition>, Abruf: 28.06.2014
- IBM (2000):** SQL Erste te, <ftp://ftp.software.ibm.com/ps/products/db2/info/vr7/pdf/letter/nlv/db2y0g70.pdf>, Abruf: 06.06.2014
- ITWissen (2014):** Objektrelationale bank, <http://www.itwissen.info/definition/lexikon/Objektrelationale-Datenbank-object-relational-database.html>, Abruf: 04.06.2014
- PostgreSQL (o. J.):** Das freie objektrelationale Open Source Datenbanksystem, <http://www.postgresql.de/>, Abruf: 27.06.2014
- Uni Freiburg (2005):** SQL-Standard, <http://dbis.informatik.uni-freiburg.de/content/DBBuch/Folien/kapitel03.pdf>, Abruf: 15.06.2014

Webhostings.ch (2011):

Profi-Webhosting für Unternehmen, Glossar,
<http://www.webhostings.ch/webhosting/features/glossar/index.html>, Abruf: 27.06.2014

Konzepte und Einsatzszenarien von Wide-Column-Datenbanken

Wissenschaftliche Arbeit

Vorgelegt am
14.07.2014

Heike Binner
Alexandra Höger
Needa Montazeri

Fakultät Wirtschaft
Studiengang Wirtschaftsinformatik
WWI2011V

Inhaltsverzeichnis

1. Einleitung	1
1.1 Motivation und Problemstellung.....	1
1.1.1 Aufbau.....	1
2. Grundlagen von Wide-Column-Datenbanken	2
2.1. NoSQL Datenbanken	2
2.1.1 Definition	2
2.1.2 Funktionen.....	3
2.2 Wide-Column-Datenbanken	4
2.2.1 Hintergrund und Überblick	5
2.2.2 Definition	5
3. Marktübersicht Wide-Column-Datenbanken	8
3.1. Markt.....	8
3.2. Hadoop/HBase.....	10
3.3. Cassandra	12
3.4. Hypertable	14
3.5. Appach Accumulo	16
3.6 Amazon SimpleDB.....	17
4. Kriterienkatalog	21
4.1 Definition und Merkmale	21
4.2 Kriterien	23
4.3 Bewertung.....	28
5. Prototyp.....	30
5.2 Testdaten.....	30
5.2. Benutzeroberfläche	32
5.1. Aufbau der Datenbank	33
6. Fazit	35
Anhang 1.....	37

Abkürzungsverzeichnis

SQL	=	Structured Query Language
NoSQL	=	Not only SQL
OLAP	=	Online Analytical Processing
CQL	=	Cassandra Query Language
HDFS	=	Hadoop Distributed File System
HQL	=	Hypertable Query Language
API	=	Application Programming Interface
REST	=	Representational State Transfer
SSL	=	Secure Sockets Layer

Abbildungsverzeichnis

Abbildung 1: Aufbau Column Family.....	5
Abbildung 2: Marktforschung: Ablauf der Studie	9
Abbildung 3: Column Family HBASE/Hadoop.....	11
Abbildung 4: Beispiel Column Family Cassandra	13
Abbildung 5: Beispiel Super Column Family Cassandra	13
Abbildung 6: Beispiel Datenmodell Amazon SimpleDB.....	18
Abbildung 7: Ranking Wide Column Stores solidIT.....	29
Abbildung 10: MySQL Workbench.....	32
Abbildung 8: Aufbau der MySQL-DB	33
Abbildung 9: Benutzeroberfläche Cassandra.....	34

Tabellenverzeichnis

Tabelle 1: Differenzierung relationales und spaltenorientiertes Datenmodell	6
Tabelle 2: Beispiel zeilenorientiertes Datenbanksystem	6
Tabelle 3: Beispiel spaltenorientiertes Datenbanksystem	6
Tabelle 4: Basiseigenschaften Apache Hadoop.....	10
Tabelle 5: Basiseigenschaften Cassandra.....	12
Tabelle 6: Basiseigenschaften Hypertable.....	15
Tabelle 7: Basiseigenschaften Apache Accumulo	16
Tabelle 8: Basiseigenschaften Amazon SimpleDB	18
Tabelle 10: Kriterienkatalog HBase/Hadoop / Cassandra	24
Tabelle 11: Kriterienkatalog Hypertable / Accumulo	26
Tabelle 12: Kriterienkatalog AmazonSimpleDB	27
Tabelle 13: Bewertung Tools	28

1. Einleitung

In einem Unternehmen entsteht jeden Tag eine große Anzahl von Informationen, so die Computerwoche.¹ Um diese Datenflut zu bewältigen und die Informationen schnell verfügbar zu halten, ist eine Not Only (NoSQL)-Datenbank eine geeignete Lösung.

1.1 Motivation und Problemstellung

Die NoSQL-Bewegung führte zu verschiedenen Lösungen, die dem Anwender kostenlos zur Verfügung gestellt wurden. Dabei wird NoSQL als Sammelbegriff verstanden und umfasst ca. 150 verschiedene Systeme und Anwendungen. Sie unterscheiden sich primär in Größe, Reifegrad und Ausprägung. Diese Arbeit beschäftigt sich speziell mit den Open Source Wide-Column-Datenbanken.

Im Rahmen dieser Arbeit soll zunächst eine Auswahl eines geeigneten Tools getroffen werden, welches im Anschluss einer MySQL Datenbank gegenübergestellt wird. Das Ziel dabei ist es, die Wide-Column-Datenbanken zu beschreiben, zu klassifizieren und dann Vergleichbar zu machen. Zur Erreichung des Ziels werden verschiedene Aufgaben vorausgesetzt. Dazu zählt unter anderem ein Kriterienkatalog, welcher die Funktionen der Datenbanken enthalten soll. Der detaillierte Aufbau wird in dem folgenden Unterkapitel beschrieben.

1.1. Aufbau

Zu Beginn werden die Grundlagen von NoSQL, speziell der Bereich Wide-Column-Datenbanken, beleuchtet. Dazu zählen unter anderem die Definition, Anwendungs- sowie Themengebiete. Anschließend werden die einzelnen Tools beschrieben.

Die dabei betrachteten Datenbanken sind unter anderem Cassandra, Hadoop oder Accumulo. Die Auswahlkriterien sind zum einen, dass es eine spaltenorientiert Datenbank ist und zum anderen unter einer OpenSource Lizenz stehen muss. Die Beschreibung der Tools findet dabei immer nach einem festgelegten Schema statt. Zuerst wird der Hintergrund beleuchtet, anschließend gibt es eine Tabelle, die die grundlegenden Daten wie Hersteller, Lizenzmodell oder aktuelle Version aufführt. Weiterhin werden die Funktionen, sowie Vor- und Nachteile betrachtet.

¹ Wolfgang Klimt (26.06.2013)

Mit den dort gesammelten Daten wird ein Kriterienkatalog erstellt und gefüllt. Damit ist auf einen Blick erkennbar, welche Tools sich für welche Einsatzgebiete besonders eignen beziehungsweise die meisten Funktionen enthalten. Daraufhin entscheiden sich die Autoren für eine der Datenbanken mit welcher sie im fünften Kapitel ein Prototyp erstellen.

Dieser wird einer relationalen Datenbank gegenüber gestellt. Die Datensätze sind in beiden Anwendungen die Gleichen. Die Möglichkeiten der Abfragen werden kurz beschrieben und anschließend auf einem gemeinsamen System getestet. Dabei werden die unterschiedlichen Zeiten gemessen und dokumentiert. Durch dieses Vorgehen können die Vor- beziehungsweise Nachteile einer spaltenorientierten gegenüber einer relationalen Datenbank deutlich gemacht werden. Zum Abschluss wird ein Fazit mit den, in diesem Projekt erarbeiteten Ergebnissen, geschrieben.

2. Grundlagen von Wide-Column-Datenbanken

Was ist ein Spaltenorientiertes Datenbanksystem? Welche Funktionen beinhalten solche Systeme? Wie lautet die genaue Definition? Diese und weitere Fragen werden in dem folgenden Kapitel detailliert erläutert.

2.1. NoSQL Datenbanken

Die NoSQL Datenbanken sind in die Kapitel geschichtlicher Hintergrund, Definition sowie Funktionen unterteilt. Letzteres ist wieder in die Hauptfunktionen untergliedert.

2.1.1 Definition

Unter dem Begriff NoSQL Datenbanken werden alle Datenbankmodelle zusammengefasst, die das gängige relationale Datenbankmodell infrage stellen.² NoSQL steht dabei jedoch für „Not only SQL“ und nicht „No SQL“. Weitere Merkmale dieser Datenbanken sind, dass es Open Source Systeme sind, es ist schemafrei und bietet eine einfache Schnittstelle. Zudem liegen diesen einfache Architekturen zugrunde, weshalb Datenreplikationen unkompliziert erstellt werden können. Ziel der Entwicklung von NoSQL Datenbanken ist es, moderne web-scale Datenbanken zu erstellen, die große Datenmengen aufnehmen können und einfach damit umgegangen werden kann.

²DJ Walker-Morgan (2014)

Die Merkmale sind allerdings nicht zu strikt zu sehen, da jeder unter einer NoSQL Datenbank etwas anderes versteht und kein Gremium oder Organisation existiert, die diesen Begriff einheitlich definiert.³

Der Begriff NoSQL kam um ca. 1998 auf und wurde zunächst für ein relationales Datenbankmodell verwendet, das keine Structured Query Language (SQL) Schnittstelle bietet. Die klassischen NoSQL Systeme wurden zwischen 2006 und 2009 entwickelt. Erst seit 2009 hat die Bewegung eine Formation erstellt, die einen Kontrast zu den RDBMS bilden.⁴ Grundsätzlich kann bei NoSQL Datenbanken zwischen Key/Value Systeme, Column-Family-Systeme, Document Stores und Graphdatenbanken unterschieden werden. Darüber hinaus gibt es weitere, die aber weniger weit verbreitet sind. In dieser Arbeit werden lediglich die Column-Family Systeme (Wide Column Datenbanken) beleuchtet.⁵

2.1.2 Funktionen

Bei NoSQL Datenbanken werden beispielsweise Protokolle und Algorithmen eingesetzt, die aus dem relationalen Umfeld nicht bekannt sind. Die wichtigsten davon sind Map/Reduce, CAP-Theorem/EventuallyConsistent sowie bestimmte APIs. Diese Komponenten, werden nachfolgend näher erläutert.⁶

Map/Reduce

Map/Reduce ist ein Framework mit alternativem Algorithmus, welches für die Verarbeitung und Durchsuchung von verteilten Daten bei paralleler Ausführung entwickelt wurde. Es übernimmt die Abfrage von Datenbankeinträgen. Damit stellt es eine zentrale Funktion in NoSQL Datenbanken dar. Dieses wird vor allem bei Datenmengen von Terabytes bis zu mehreren Petabytes relevant. Das Framework macht eine effiziente nebenläufige Berechnung möglich. Dabei werden, vereinfacht dargestellt, die Daten parallel zunächst in einem map Prozess in Zwischenergebnisse und dann in einem reduce() in Ergebnissen zusammengefasst und beendet. Damit lassen sich die Aufgaben der Parallelisierung, Speicherung von Zwischen- und Endergebnissen sowie von Datentransfer realisieren. Darüber hinaus ist Monitoring und eine effiziente Lastverteilung möglich. Zudem hat dieses Tool eine gewisse Fehlertoleranz.⁷

³ Edlich, S./Friedland, A./al., Jens Hampe et (2011), S. 2 ff.

⁴ Edlich, S./Friedland, A./al., Jens Hampe et (2011), S. 1 ff.

⁵ Edlich, S./Friedland, A./al., Jens Hampe et (2011), S. 7

⁶ Edlich, S./Friedland, A./al., Jens Hampe et (2011), S. 11

⁷ Edlich, S./Friedland, A./al., Jens Hampe et (2011), S. 12 ff.

CAP und EventuallyConsistent

CAP ist bezeichnend für das CAP Theorem. Dieses lieferte den Nachweis, dass für das Web 2.0 und wachsenden Datenmengen das Konsistenzmodell der relationalen Datenbanken (ACID) unvorteilhaft ist. Das Problem liegt in der Vereinbarkeit von Konsistenz, Verfügbarkeit und Ausfalltoleranz. Bei verteilten Datensystemen können höchstens zwei dieser Attribute erreicht werden. Die Lösung für dieses Problem bildet das Modell BASE (BasicallyAvailable, Soft State, EventuallyConsistent). Im Gegensatz zum ACID steht nicht die Konsistenz, sondern die Verfügbarkeit im Vordergrund.

Die Bezeichnung EventuallyConsistent folgt daraus, dass der Zustand der Konsistenz irgendwann erreicht wird, jedoch nicht nach jeder Transaktion. Ein weiterer Nachteil des BASE Modells ist, dass bei NoSQL Datenbanken jeder Hersteller in Hinblick auf dieses seinen eigenen Weg geht und daher sehr viele verschiedene existieren.⁸

APIs

http/REST

Mit REST, REpresentational State Transfer, wird ein Entwurfsmuster bezeichnet. Dieses nimmt Einfluss auf die Struktur des Systems, zumeist verteilter Anwendungen, für die dieses Modell entwickelt wurde. Dabei bietet dieses eine breite Skalierbarkeit und bindet meist http ein. REST besteht aus drei Komponenten: Ressourcen, Operationen und Links.⁹

Thrift

Thrift ist ein Framework, mit dem Services entwickelt werden können. Dabei stellt dieses einen Generator bereit, der Codevorlagen für Client- sowie Serverseite erstellen kann. Dies ist in vielen Sprachen wie beispielsweise Java möglich. Damit kann Thriftprogrammiersprachenübergreifend eingesetzt werden. Implementiert ist das Framework in ThriftIDL.¹⁰

2.2 Wide-Column-Datenbanken

In diesem Kapitel werden die Details zu den sogenannten Wide-Column-Datenbanken beschrieben. Sie zählen zu einem Teilgebiet der im letzten dargestellten NoSQL Systemen.

⁸ Edlich, S./Friedland, A./al., Jens Hampe et (2011), S. 30 ff.

⁹ Edlich, S./Friedland, A./al., Jens Hampe et (2011), S. 51 ff.

¹⁰ Edlich, S./Friedland, A./al., Jens Hampe et (2011), S. 74

2.2.1 Hintergrund und Überblick

Bereits seit den 80er Jahren besteht der Gedanke ein spaltenorientiertes Datenbankmodell zu generieren und dessen Vorteile zu nutzen. Besonders das Zusammenfügen von Spalten zu sogenannten Column-Families birgt unter anderem in Sachen Leistung oder auch Organisation gegenüber relationalen Modellen je nach Einsatzgebiet der Datenbank eine Möglichkeit der Verbesserung. So können innerhalb Column-Families beispielsweise Daten gruppiert werden, welche in den meisten Fällen zusammenhängend sind und bei Aufrufen häufig in Verbindung zueinander stehen. Eine Column-Family kann wiederum eine Anzahl von Rows haben, die aus einem RowKey (Primärschlüssel) und mehreren Columns (Spalten) besteht.¹¹

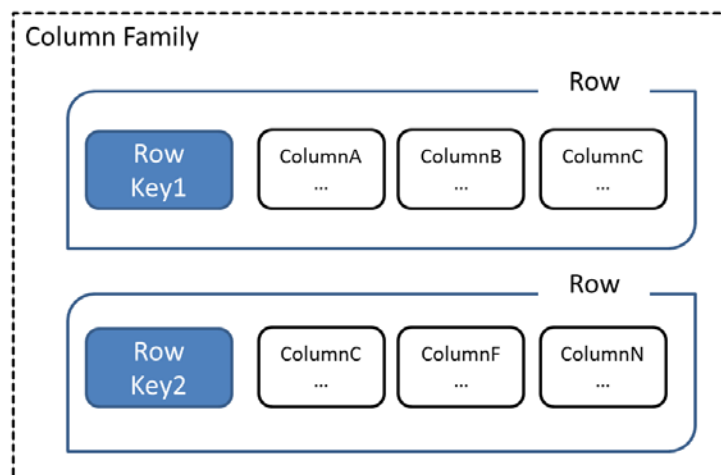


Abbildung 1: Aufbau Column Family¹²

Die Abbildung 1 zeigt den zuvor beschriebenen Aufbau einer Wide-Column-Datenbank auf. Zu erkennen ist die Gliederung in eine Column Family und die darin enthaltenen Rows mit den jeweiligen RowKeys und Columns. Im nachfolgenden Kapitel wird näher auf die Datenstrukturierung und den Unterschied zu relationalen Datenbankmodellen eingegangen, sowie in welchen Bereichen Vorteile oder auch Nachteile in der Nutzung entstehen.

2.2.2 Definition

Unter Wide-Column-Store werden spaltenorientierte Datenbanken verstanden. Im Gegensatz zu relationalen Modellen, werden hierbei die Spaltenattribute innerhalb einer Tabelle *nacheinander* gespeichert. Der Unterschied der physischen Datenorganisation ist anhand der Tabelle 1 ersichtlich.

¹¹ Vgl. Fowler und Sadalage 2012, S. 99

¹² Mit Änderungen entnommen aus: Fowler und Sadalage 2012, S. 100

Die erste Spalte führt beispielhafte Datensätze eines relationalen Datenmodells auf, wohingegen die zweite Spalte die Attribute eines Wide-Column-Stores aufzeigt.¹³

<i>Relationales Datenmodell</i>	<i>Spaltenorientiertes Datenmodell</i>
Erika Müller, 70567, Stuttgart, Max Schmidt, 80995, München, Elke Schulze, 72160, Horb	Erika Müller, Max Schmidt, Elke Schulze, 70567, 80995, 72160, Stuttgart, München, Horb

Tabelle 1: Differenzierung relationales und spaltenorientiertes Datenmodell

Zu erkennen ist die unterschiedliche Speicherung der Daten. Bei einem zeilenorientierten Datenbanksystem werden **alle** Attribute **einer** Zeile angesprochen.

<i>Name</i>	<i>PLZ</i>	<i>Ort</i>
Erika Müller	70567	Stuttgart

Tabelle 2: Beispiel zeilenorientiertes Datenbanksystem

Innerhalb eines spaltenorientierten Datenbanksystems wird wiederum **ein** Attribut **aller** Zeilen bearbeitet.

<i>Name</i>	<i>PLZ</i>	<i>Ort</i>
Erika Müller		
Max Schmidt		
Elke Schulze		

Tabelle 3: Beispiel spaltenorientiertes Datenbanksystem

Eine spaltenorientierte Speicherung von Datensätzen ist zum einen vorteilhafter (gegenüber zeilenorientierter Datenbanken) wenn innerhalb einer Tabelle viele Zeilen bestehen, aber eine begrenzte Anzahl von Spalten notwendig ist.

Beispiel: `SELECT COUNT(MA-Nummer) FROM Mitarbeiter;`

¹³ Vgl. Edlich, S./u. a. (2010), S. 53 f.

Zum anderen ist das Einfügen von Datensätzen effektiver, sobald dieser Wert für alle Zeilen, zum Beispiel abgeändert werden muss und dieser nur eine Spalte betrifft.

*Beispiel: UPDATE Mitarbeiter SET Gehalt=Gehalt*1,01;*

Werden hingegen viele beziehungsweise mehrere Spalteninhalte eines Datensatzes benötigt, ist eine Wide-Column-Datenbank nicht effizient, da viele Spalten angesprochen werden müssen.

*Beispiel: SELECT * FROM Mitarbeiter;*

Ein weiterer Nachteil bezüglich der Nutzung von Wide-Column-Stores besteht, sobald Zeilen eingefügt werden, bei welchen alle Attribute befüllt werden müssen. Das heißt alle Daten zu demselben Zeitpunkt bekannt sind und eingefügt werden können.

*Beispiel: INSERT INTO Mitarbeiter (MA-Nummer, Nachname, Vorname, Gehalt)
VALUES (156, Müller, Heinz, 2500);*

Die oben aufgeführten Vorteile lassen auf eine effektive Nutzung von Wide-Column-Datenbanken schließen, wenn die Analyse, die Aggregation und die Verfeinerung von Daten im Fokus stehen. Aus diesem Grund werden spaltenorientierte Datenbanken in der Praxis häufig im Bereich Online Analytical Processing (OLAP)-Anwendungen und Data-Warehouse-Umgebungen verwendet.¹⁴

¹⁴ Vgl. Böswetter D. (2010)

3. Marktübersicht Wide-Column-Datenbanken

Um eine Marktübersicht zu erhalten, müssen verschiedene Tools lokalisiert und getestet werden. Das folgende Unterkapitel beschreibt das theoretische Vorgehen einer Marktanalyse. Im Anschluss werden die einzelnen Anwendungen aufgezeigt, die in dieser wissenschaftlichen Arbeit betrachtet wurden.

3.1. Markt

Bevor eine Marktanalyse durchgeführt werden kann, werden grundlegende Informationen recherchiert. Beispiel dafür ist unter anderem die allgemeine Marktsituation. Diese Sammlung von Informationen wird zusammengefasst, aufbereitet und ausgewertet. Grundsätzlich wird in unternehmensinterne und -externe Informationsbereiche unterschieden.

Fehlentscheidungen können durch unternehmensinterne Informationen verhindert werden. Die dazu benötigten Daten werden im Unternehmen für tägliche Entscheidungen verwendet. Sie stammen aus internen Quellen und dienen zur Bewertung des eingesetzten Marketings. Damit werden Risiken und Verbesserungsmöglichkeiten sichtbar.¹⁵

Unternehmensexterne Informationen werden in drei Blöcke unterteilt. Diese sind

- Rahmenbedingungen
- Marktteilnehmer
- Marktreaktion

Rahmenbedingungen umfassen alle Einflüsse, die durch das Unternehmen direkt gelenkt werden können. Das sind rechtliche, gesellschaftliche und wirtschaftliche Faktoren, wie zum Beispiel die Erhöhung eines Preises. Marktteilnehmer sind Konsument (Nachfrager), Konkurrenz und Absatzmittler. Diese reagieren auf Veränderungen des Marktes wie zum Beispiel eine Produktinnovation oder eine Preisveränderung. Die genannten Reaktionen werden in dem dritten Block der unternehmensexternen Informationen zusammengefasst. Beispielsweise sinkt die Konjunktur und so verändern sich das Kaufverhalten der Konsumenten und die Preispolitik der Konkurrenz, sowie der Absatzmittler.

Die daraus resultierenden Informationen werden vor der eigentlichen Marktanalyse gesammelt und anschließend zusammengefasst. Der eigentliche Prozess wird in fünf Phasen untergliedert¹⁶, welche in der folgenden Abbildung dargestellt werden.

¹⁵ Kotler, P./Schellhase, R. (2011), S. 365

¹⁶ Berekoven, L./Eckert, W./Ellenrieder, P. (2006), S. 34 ff und Scharf, A./Schubert, B./Hehn, P. (2009), S. 110

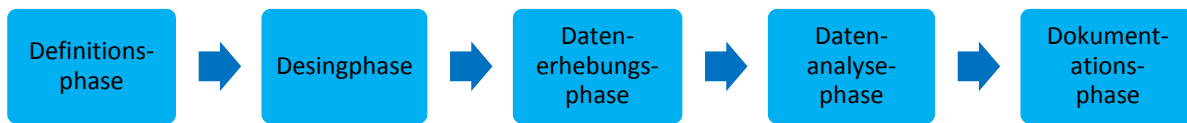


Abbildung 2: Marktforschung: Ablauf der Studie¹⁷

Die Reihenfolge dieser Phasen kann je nach definierter Aufgabenstellung geändert werden. Jedoch müssen alle detailliert geplant werden. In der ersten Phase werden das Problem sowie die daraus abgeleiteten Ziele definiert. Zum Beispiel ist das Problem eine fehlende Datenbank. So sind die daraus ableitbaren Ziele die Untersuchung verschiedener Datenbanken, um eine für das Unternehmen passende zu finden. Die zweite Phase enthält die Bestimmung des Forschungsdesign und der Rahmenbedingungen für verschiedene Themen, wie beispielsweise die Befragung der Zielpersonen, die Erhebungsform und das Erhebungsverfahren.

Die Datenerhebung ist der dritte Schritt. Dabei werden systematisch und gezielt Informationen gesammelt.¹⁸ An dieser Stelle gibt es zwei grundlegende Methoden, welche die Primär- und Sekundärforschung sind. Bei letzterem werden bereits erhobene Daten, welche zu einem anderen Zweck erhoben wurden analysiert und verwendet. Die Primärforschung erhebt neue Daten mit Hilfe verschiedener Vorgehensweisen wie beispielsweise Befragung oder Beobachtung.

Nach der Erhebung müssen die gesammelten Informationen aufbereitet und ausgewertet werden. Dies geschieht in der vierten Phase. Mit Hilfe der Daten soll das definierte Problem gelöst und die Ziele erreicht werden. Die eigentliche Analyse kann mit Hilfe unterschiedlicher Software, sowie technischen Lösungen vorgenommen werden.

Dabei werden in der Regel univariate (Betrachtung einer Variable), bivariate (Betrachtung zwei Variablen) und multivariate (Betrachtung von mindestens drei Variablen) Verfahren verwendet.¹⁹

¹⁷ Kotler, P./Schellhase, R. (2011), S. 373

¹⁸ Hammann, P./Erichson, B. (2005), S. 74

¹⁹ Scharf, A./Schubert, B./Hehn, P. (2009), S. 149

Durch die fünfte Phase wird die Marktanalyse abgeschlossen. In diesem Schritt werden alle gesammelten Informationen und Daten, sowie deren Auswertung dokumentiert und zu einem gesamten Fazit zusammengefasst. Sie dient zusätzlich zur Präsentation des eigentlichen Ergebnisses. Ein beliebtes Mittel dafür stellt der Kriterienkatalog dar, welcher in Kapitel vier dieser Arbeit auch zum Einsatz kommt.

3.2. Hadoop/HBase

Hintergrund

HBase ist dem Hochleistungs-Datenbanksystem „*Google BigTable*“ nachempfunden mit dem Ziel ein quelloffenes und ähnliches Tool bereitzustellen. Das Open Source Projekt wurde von dem Entwickler Douglass Read Cutting initiiert und im Jahre 2008 von der Apache Software Foundation übernommen.²⁰

	Apache™ Hadoop®
Hersteller	Apache Software Foundation
Letzte Version	Release 2.4.0 / 07.04.2014
Lizenz	Open Source
Plattformen	plattformunabhängig, durch Shell-Skripte nur mit Kompatibilitätsschicht (zum Beispiel Cygwin) ausführbar
Downloadlink	http://hadoop.apache.org/releases.html
Größe	15 MB

Tabelle 4: Basiseigenschaften Apache Hadoop

Funktionen

HBase ist eine partiell spaltenorientierte Datenbank. Das heißt, das System nutzt eine mehrdimensionale Struktur um eine möglichst hohe Flexibilität und Performance zu erreichen.²¹

Zur Speicherung von Daten verwendet HBase das Dateisystem Hadoop. Dies ist wiederum ein Framework, welches eine verteilte Verarbeitung von großen Datenmengen in Clustern ermöglicht. Hadoop umfasst unter anderem die wie im Folgenden aufgelisteten Module:

Hadoop Common

Dienst zur Unterstützung der weiteren Module

²⁰ Vgl. Edlich, S./u. a. (2010), S. 54 ff.

²¹ Vgl. Edlich, S./u. a. (2010), S. 53 f.

Hadoop Distributed File System (HDFS™)

Verteiltes Dateisystem, welches Zugriff auf Anwendungsdaten bietet

Hadoop YARN

Framework mit welchem Ressourcenverwaltung und Scheduling-Funktionen effizienter gestaltet werden

HadoopMapReduce

Framework zur Parallelisierung von Berechnungen bei großen Datenmengen²²

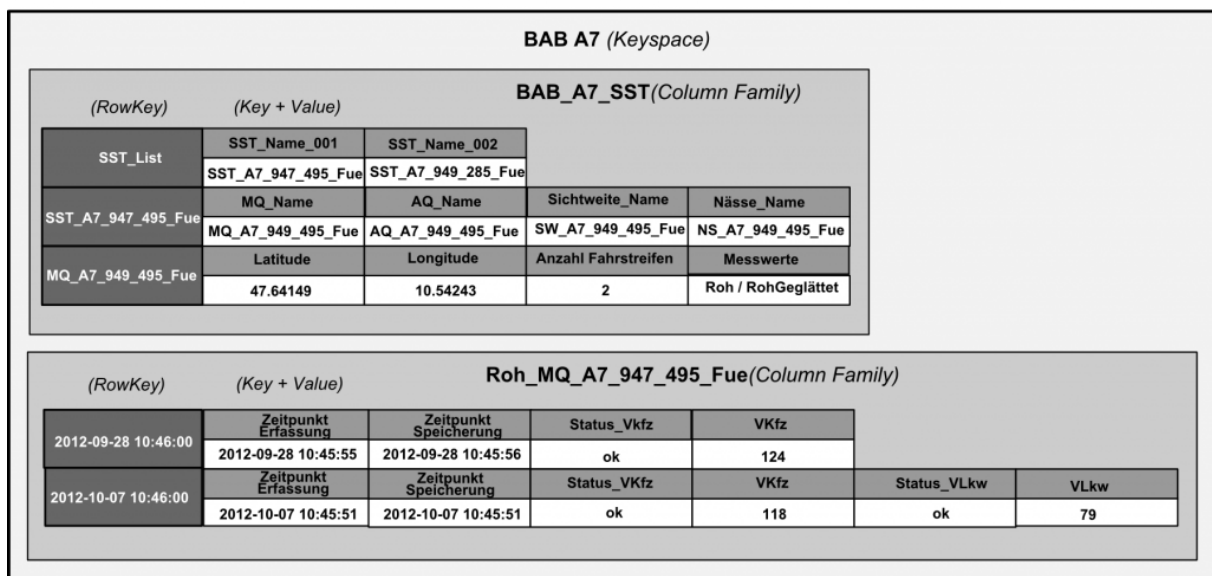


Abbildung 3: Column Family HBASE/Hadoop²³

Das in der Abbildung 3 dargestellte Beispiel zeigt eine partiell spaltenorientierte Datenbank auf. Zu erkennen ist der Zusammenhang einer Anzahl von Spalten (Keyspace), sowie die Gruppierung in Spaltenfamilien (Column-Family) und deren separate Speicherung.

²²Vgl. The Apache Software Foundation (2014)

²³Vgl. Hahn Projects (o. J.)

Vor-/Nachteile

Ein Vorteil von HBase ist die mögliche Nutzung des Systems auf Standardhardware. Dies ist durch die Skalierung und dem Hinzufügen eines Region Servers realisierbar. Eine weitere vorteilhafte Eigenschaft ist die Parallelisierung von Abfragen durch das Modul MapReduce, durch welches eine effiziente und effektive Verarbeitung stattfindet. Des Weiteren gewähren die integrierten Schnittstellen (zum Beispiel Java-API) eine Kompatibilität gegenüber weiteren Anwendungskontexten. Da hingegen ist die Bearbeitung von Clustern, sowie das Erstellen komplexer Abfragen sehr zeitaufwendig und umfangreich.²⁴

3.3. Cassandra

Hintergrund

Ursprünglich wurde Cassandra zur Lösung des Inbox Search Problems von Facebook entwickelt und im Jahre 2010 als Open-Source System von Apache übernommen. Seit dem wurde es als Top-Level-Projekt eingestuft und demnach kontinuierlich weiterentwickelt und optimiert. Auch Cassandra ist an „Googles BigTable“ angelehnt und hat zum Ziel eine hohe Skalierbarkeit und Flexibilität zu ermöglichen.²⁵ Twitter, Reddit oder auch die NASA sind nur einige von vielen Nutzern des Datenbanksystems.²⁶

	Cassandra
Hersteller	Apache Software Foundation
Letzte Version	Apache Cassandra 2.0.8 (vom 29.05.2014)
Lizenz	Open-Source(Apache Lizenz)
Plattformen	plattformunabhängig
Downloadlink	http://cassandra.apache.org/download/
Größe	17 MB

Tabelle 5: Basiseigenschaften Cassandra

Funktionen

Das von Cassandra verwendete Datenmodell ist eine Kombination aus einem Key-Value-Store und Wide-Column-Store. Die Organisation der Daten erfolgt hierbei über Column-Families, welche aus Row Keys und die dazugehörigen Key-Value-Pairs bestehen. Die fol-

²⁴Vgl. Edlich, S./u. a. (2010), S. 68 f.

²⁵Vgl. Schnelle J. (2011)

²⁶Vgl. Planet Cassandra (o. J.)

gende Abbildung # zeigt ein Beispiel einer Column-Family mit den zwei Row Keys „User_1“ und „User_2“.

```

Row Key: "User_1"
Column "Name"      - Wert: "Otto"
Column "Alter"     - Wert: "40"

Row Key: "User_2"
Column "Name"      - Wert: "Susi"
Column "Alter"     - Wert: "25"
Column "Nickname"  - Wert: "TurboS"

```

Abbildung 4: Beispiel Column Family Cassandra²⁷

Eine Erweiterung bezüglich der Column-Families bilden die Super Columns. Hierbei kann eine zusätzliche Ebene hinterlegt werden, wodurch die Super Column wiederum zu einer Column-Family wird. Die Abbildung 5 zeigt ein Beispiel solch einer Super Column.

```

Row Key: "User_1":
Super Column "Name":
Column "Vorname"   - Wert: "Otto"
Column "Nachname" - Wert: "Normal"
Super Column "Kontaktdaten":
Column "E-Mail"    - Wert: "otto@example.de"
Column "Jabber"    - Wert: "on@superjabber.de"

Row Key: "User_2":
Super Column "Vorname":
Column "Vorname"   - Wert: "Susi"
Column "Nachname" - Wert: "Sorglos"
Column "Maedchenname" - Wert: "Sonnenschein"
Super Column "Kontaktdaten":
Column "E-Mail"    - Wert: "susi@sorglos.de"

```

Abbildung 5: Beispiel Super Column Family Cassandra²⁸

Das Einpflegen und Abfragen von Daten erfolgt innerhalb Cassandra über die Abfragesprache CQL (Cassandra Query Language). Die Syntax ist an SQL angelehnt und beinhaltet unter anderem die folgenden Operatoren:

SET – Hinzufügen von Daten

GET – Abfragen von Daten und

DEL – Löschen von Daten.

²⁷Vgl. Schnelle J. (2011)

²⁸Vgl. Schnelle J. (2011)

Zudem wird die Map/Reduce Methode eingesetzt, was eine effektive Parallelisierung der Berechnungen ermöglicht.²⁹

Vor-/Nachteile

Cassandra bietet hohe Schreib- und Leseraten, schließlich wurde das System zu diesem Zweck entwickelt. Außerdem kann eine sehr hohe Anzahl von Datenmengen verarbeitet werden (zweistelliger Terabyte-Bereich). Ein weiterer Vorteil liegt in der Strukturierung über Column-Families oder Super Columns und deren freie und individuellen Handhabung.

Im Gegensatz zu anderen Wide-Column-Datenbanken, wie zum Beispiel HBase/Hadoop, gestaltet sich die Installation von Cassandra einfacher. Ein Nachteil ist jedoch die beschränkte Auswahl an Abfragemöglichkeiten.³⁰

3.4. Hypertable

Hintergrund

Hypertable ist wie HBase/Hadoop der Datenbank „GoogleBigTable“ nachempfunden und nutzt ebenfalls die Möglichkeit Daten durch und in das Apache Hadoop Distributed File System (HDFS) Dateisystem zu speichern. Das Open-Source Projekt wurde von Doug Judd gegründet und im Jahre 2009 veröffentlicht. Neben dem kostenlosen Systemdownload bietet Hypertable zusätzlich einen kostenpflichtigen Support an. Die bekanntesten Nutzer von dem System sind die chinesische Suchmaschine baidu.com, sowie das indische Webportal rediff.com.³¹

²⁹Vgl. Jansen R. (o. J.)

³⁰Vgl. Edlich, S./u. a. (2010), S. 82 f.

³¹Vgl. Edlich, S./u. a. (2010), S. 256 f.

Hypertable	
Hersteller	Hypertable Inc.
Letzte Version	Hypertable Binary Version 0.9.8.0
Lizenz	Open-Source(Apache Lizenz)
Plattformen	plattformunabhängig, durch Shell-Skripte nur mit Kompatibilitätsschicht (zum Beispiel Cygwin) ausführbar (siehe HBase/Hadoop)
Downloadlink	http://hypertable.org/download
Größe	99 MB

Tabelle 6: Basiseigenschaften Hypertable

Funktionen

Hypertable basiert ebenfalls auf dem Dateisystem Hadoop und konzentriert sich auf die Verarbeitung von sehr großen Datenmengen. Ein Unterschied liegt in der Umsetzungsweise Entwicklung. Hypertable verwendet an dieser Stelle C++, was zu einer besseren Performance führen soll.³²

Verarbeitete Daten können zudem über Hypertable für Online-Anwendungen zur Verfügung gestellt werden. Hierzu werden Datensätze tabellarisch aufbereitet und mit einem Primärschlüssel versehen. Die Speicherung über die Primärschlüsselabfolge führt des Weiteren dazu, dass speziell für eine Analyseanwendung (zum Beispiel im Bereich Business Intelligence) eine besonders effiziente Vorgehensweise möglich ist. Ein weiterer zu erwähnender Aspekt von Hypertable ist die Abfrage von Daten. Dies geschieht über Hypertable Query Language (HQL) einer durch Erweiterungen definierten SQL-Teilmenge, durch welche die Auswahl von Reihen, Bereichen und Zeit besonders vorteilhaft ist.

Vor-/Nachteile

Wie bereits erwähnt, wird durch die Umsetzung in C++ eine hohe Performanceleistung gewährleistet. Für Offline-Anwendungen kann hierbei ein höherer Datendurchlauf erzielt werden, dies heißt, dass innerhalb eines bestimmten Zeitraumes eine höhere Verarbeitung stattfindet. Des Weiteren zeichnet sich Hypertable durch die Skalierbarkeit und die Konsistenz aus.

³²Vgl. Bogaard, M. (2011)

Die Sortierung über den Primärschlüssel lässt eine große Anzahl von Anwendungen zu. Ein Nachteil von Hypertable liegt im Bereich HQL, welche für komplexe Abfragen nicht geeignet ist.³³

3.5. Appach Accumulo

Hintergrund

Appach Accumulo wurde vor sechs Jahren (2008) von der National Security Agency programmiert und drei Jahre später als Open Source unter der Apache Lizenz freigegeben. Es basiert auf der „BigTable“-Technologie von Google, welche im Jahre 2006 veröffentlicht wurde und auch in verschiedenen anderen Datenbanken wie Cassandra oder Hadoop verwendet wird.³⁴ Es wurde die Programmiersprache Java verwendet.

Accumulo kann auf Hadoop, Zookeeper und Trift aufgesetzt werden und bietet so noch zusätzliche Funktionen neben dem eigentlichen Tool. Diese werden in dem folgenden Kapitel detailliert beschrieben.

	Appach Accumulo
Hersteller	Apach Software Foundation
Letzte Version	1.6
Lizenz	Apache Lizenz 2.0
Plattform	Cross Platform
(Download)Link	https://accumulo.apache.org/
Größe	12,5 MB

Tabelle 7: Basiseigenschaften Appache Accumulo

Funktionen

Das Design von Accumulo beruht auf einer spaltenorientierten, schemalosen Key-value Datenbank. Die dabei verwendete Abfragesprachen sind unter anderem Java Application Programming Interface (API), Thrift, API Aufrufe oder Representational State Transfer (REST).

³³Vgl. Hypertable Inc. (27.06.2014)

³⁴ Vgl. o.V. (2014a)

Accumulo ist Unicode sowie map and reduce fähig. Es sichert die Dauerhaftigkeit der Daten, also die Datenspeicherung gewährleistet. Transaktionen sind nicht möglich.³⁵

In der neuen Version 1.6 können zusätzlich Dateien auf verschiedenen Masterknoten bzw. Namensknoten gespeichert werden.³⁶ Durch dieses Feature wird ein Engpass der grundlegenden Technologie umgangen. Zudem können neben anderen Features nun auch die Tabellen-Namespaces gemeinsam bearbeiten und die Datenübertragung per Secure Sockets Layer (SSL) verschlüsselt werden.

Vor- und Nachteile

Die Vorteile von Apache Accumulo liegen primär in der großen Verarbeitung sowie Speicherung der Daten. Dabei bietet es viele neue Features für den Anwender, was deutlich macht, dass seit der Entstehung bis heute an der Weiterentwicklung dieser Datenbank gearbeitet wurde.

Die Nachteile zeigen sich in der fehlenden referentiellen Integrität, sowie der fehlenden Volltextsuche innerhalb des Systems. Zudem muss vor der Installation von Accumulo ein anderes System, wie beispielsweise Hadoop installiert sein. Ohne dieses andere System ist eine Verwendung nicht möglich.

3.6 Amazon SimpleDB

Hintergrund

Auch Amazon SimpleDB ist eine partiell spaltenorientierte Datenbank (siehe Unterkapitel 3.2 HBase/Hadoop). Ein Unterschied zu den zuvor beschriebenen Datenbanken liegt innerhalb des Lizenzmodells. Die Amazon SimpleDB ist ein Web-Service des Anbieters *Amazon* und beruht auf dem Software-as-a-Service Modell. Dies heißt Datensätze werden hierbei über den Webbrowser erstellt, gespeichert und aufgerufen.³⁷

³⁵ Vgl. o.V. (2014b)

³⁶ Vgl.o.V. (2014c)

³⁷ Vgl. Edlich, S./u. a. (2010), S. 83 f.

Amazon SimpleDB ^(Beta)	
Hersteller	Amazon.com
Letzte Version	API Version 2009-04-15
Lizenz	Software-as-a-Service
Plattformen	Plattformunabhängig, Aufruf über Webbrowser
(Download)link	http://aws.amazon.com/de/simpledb/
Größe	-

Tabelle 8: Basiseigenschaften Amazon SimpleDB

Funktionen

Innerhalb der *Amazon SimpleDB* werden Datensätze in unterschiedlichen und von dem User angelegten Domänen organisiert. Dies heißt Daten werden in einer Gruppe von Elementen (Domäne) durch Attributwerte beschrieben. Die Abfragen erfolgen schließlich über diese erstellten Domains. Unterschiedliche Optionen, wie zum Beispiel *CreateDomain*, *DeleteDomain* oder *ListDomain* ermöglichen das Erstellen und Bearbeiten solch einer Elementensammlung.³⁸

Domäne „Kunden“

Kunden ID	Name	PLZ	Telefon
123	Müller	70159	0711/23568
456	Meier	20358	0526/568945

← **Attribute**

← **Elemente/Zeilen**

Abbildung 6: Beispiel Datenmodell Amazon SimpleDB

Die Abbildung sechs zeigt beispielhaft ein Amazon SimpleDB Datenmodell auf. Hierbei wird die Gliederung in die zuvor beschriebenen Bereiche Domäne, Attribute und Elemente ersichtlich. Durch einen **PUT-Befehl** können schließlich Daten in diese Domäne eingefügt werden; zum Beispiel *PUT (Element, 123), (Name, Müller), (PLZ, 70159), (Telefon, 0711/23568)*.

³⁸ Vgl. Edlich, S./u. a. (2010), S. 85

Weitere mögliche API-Befehle innerhalb der Amazon Simple DB sind wie folgt definiert:

BatchPutAttributes

Massenerfassung von Daten (bis zu 25 PutAttribute)

DeleteAttributes

Löschung von Elementen, Attributen oder Attributwerten

BatchDeleteAttributes

Massenlöschung (bis zu 25 DeleteAttribute)

GetAttributes

Abrufen von Elementen oder Attributen

Select

Abfragen von Datensätzen durch SQL-Syntax³⁹

Vor-/Nachteile

Einer der wesentlichsten Vorteile der Amazon SimpleDB ist die Flexibilität, welche einem Web-Service zugeschrieben werden kann. Der Benutzer besitzt die Möglichkeit jederzeit und –orts auf die gespeicherten Daten zuzugreifen. Auch die individuelle Gestaltung der Datenbank und die einfache Handhabung sind im Gegensatz zu anderen Systemen vorteilhaft und somit positiv hervorzuheben. Ein klarer Nachteil liegt bei den Kosten des Web-Services. Zwar werden die Gebühren nur für tatsächlich genutzten Ressourcen berechnet, im Vergleich zu Open-Source Produkten ist die Amazon SimpleDB dennoch kostenintensiver. Der entfallende Aufwand für die Bereitstellung der Hardware- und Softwareverwaltung, Replikation, Infrastruktur und Leistungsoptimierungen durch den Hersteller Amazon, stehen den anfallenden Kosten allerdings gegenüber.⁴⁰

³⁹ Amazon Web Services (2014a)

⁴⁰ Amazon Web Services (2014b)

4. Kriterienkatalog

In einem Kriterienkatalog werden alle Kriterien, welche eine Relevanz zur Beurteilung der Anwendungen besitzen, zusammengefasst. Dieser Katalog ist ein Anwendungsverfahren, welches für das Unternehmen keinen großen Aufwand bedeutet. In der Regel werden nicht alle Funktionen und Daten der verschiedenen Anwendungen betrachtet, sondern nur jene, die für die Auswertung wichtig sind. Aus diesem Grund ist die Gewichtung und nicht die Vollständigkeit ein Zeichen für die Qualität.

Der im Rahmen dieses Projektes erstellte Kriterienkatalog ist in zwei Teile gegliedert. Zu Beginn werden Kriterien im Bereich der *Datenbank-Eigenschaften* aufgezeigt. Im Anschluss werden die potenziellen *Zugriffsmethoden* genannt. Die Ergebnisse beziehungsweise Beschaffenheiten der untersuchten Tools werden innerhalb der darauffolgenden und je nach Datenbank unterteilten Spalten benannt. Dieses Kapitel beschäftigt sich zunächst mit der Definition der einzelnen Kriterien.

4.1 Definition und Merkmale

Je nach Anwendung oder Einsatzgebiet kann zwischen unterschiedlichen ***Datenbank Typen*** entschieden werden. Neben „Wide-Column Datenbanken“ gibt es beispielsweise „objektorientierte – oder relationale Datenbanken“. Unter dem Begriff ***Implementierungssprache*** wird die bei der Umsetzung verwendete Syntax, wie zum Beispiel JAVA oder C++ verstanden. Die Kategorie ***Datenschema*** gibt die Struktur von Daten an, das heißt Vorgaben bezüglich Datentypen und der Anzahl von Attributen. Eine schemafrei Datenbank (meist spalten- oder dokumentenorientierte Datenbanken⁴¹) unterstützt, beziehungsweise speichert, hierbei nicht nur Datensätze die der vorgegebenen Struktur entsprechen, sondern auch solche ohne zuvor definierten Typen oder Bezeichnungen. Auch die Anzahl der Datensätzen oder Spalten kann an dieser Stelle variieren.

Unter dem Kriterium ***Typisierung*** wird das Vorhandensein von vordefinierten Datentypen, wie zum Beispiel Integer, Char oder Boolean verstanden. Um das Abrufen von Nicht-Schlüsselattributen zu ermöglichen, kann ein sogenannter ***Sekundärindex*** definiert werden. Dies soll eine effektivere und effizientere Abfrage von Daten erlauben.

⁴¹ Siehe Unterkapitel 2.1.1

Ein weiteres und wichtiges Kriterium ist die **Partitionierung**. Hierrunter wird die Aufteilung und somit getrennte Speicherung (zum Beispiel auf verschiedenen Rechnern) verstanden. Unterschieden wird zum einen zwischen der „*vertikalen Partitionierung*“, bei welcher Datensätze gegliedert und anschließend gespeichert werden. Eine weitere Methode ist die „*horizontale Partitionierung*“, welche eine Gruppierung der Datensätze in logischer Reihenfolge und das verteilte Speichern vorsieht. „*Sharing*“ hat das gleiche Vorgehen wie die horizontale Partitionierung, allerdings mit dem Unterschied dass an dieser Stelle die aufgeteilten Datensätze mit Hilfe einer eigenen Serverinstanz verwaltet werden kann. Zur Gewährleistung der Ausfallsicherung eines Datenbanksystems, werden durch eine **Replikation** Daten redundant auf unterschiedlichen Knoten innerhalb eines Netzwerkes abgespeichert. **MapReduce** beschreibt hingegen die parallele Verarbeitung von Daten durch einen speziellen Algorithmus. Diese Methode wird insbesondere bei NoSQL Datenbanken angewendet (siehe hierzu Unterkapitel 2.1.2).

Die Zeile **Konsistenz** beschreibt die Methoden welche genutzt werden um die Korrektheit, beziehungsweise Gleichheit der durch die Replikation erstellten Replikas sicherzustellen. Unterschieden wird zwischen „Immediate Consistency“ und „Eventual Consistency“. Wobei ersteres eine Speicherung nach jeder Änderung vorsieht und die zweite Methode durch die Anwendung von Algorithmen erst nach Beendigung der Schreiboperation Konsistenz sicherstellen kann. Die **Zugriffskontrolle** gibt die Berechtigungen und Rechte für Benutzer oder Objekte und die verwendeten Methoden an. Unter dem Kriterium **Verwendbare Programmiersprache**, werden alle unterstützte und zur Anwendung geeigneten Programmiersprachen aufgelistet. Des Weiteren wird der Begriff **Stored Procedures** aufgeführt. Hierunter wird, wie die deutsche Übersetzung „gespeicherte Prozedur“ erahnen lässt, das Auslagern von vielen einzelner Befehlen innerhalb des Datenbanksystems verstanden. Das heißt, dass oft verwendete Anwendungen und deren Logik auf einem Server abrufbar sind und verwaltet werden können.

Das im Kriterienkatalog ausgewiesenen Merkmal „**parallele Datenmanipulation**“ zeigt auf, ob das Datenbanksystem eine Nebenläufigkeit beziehungsweise das gleichzeitige Ausführen von Anweisungen, Befehlen oder Berechnungen erlaubt. Die **Beständigkeit/dauerhafte Speicherung** stellt sichert das auch, zum Beispiel bei einem Absturz des Systems, die zuvor gespeicherten Daten oder Änderungen der Datenbank abrufbar bleiben. Dies geschieht durch das Schreiben der Daten auf einen nicht-flüchtigen Speicher.

Das letzte Kriterium im Bereich Datenbank-Eigenschaften gibt an, ob für das Datenbanksystem die Programmiersprache **SQL** verwendet werden kann. Unter der Kategorie Zugriffsmöglichkeiten werden schließlich die Methoden oder Schnittstellen aufgeführt, welche von dem System unterstützt werden. Dies sind beispielsweise die APIs Java, C++ oder Apache Thrift.⁴² (siehe Unterkapitel 2.1.2)

4. 2 Kriterien

Dieses Kapitel zeigt den im Rahmen dieses Projektes erstellten Kriterienkatalog auf. Aus technischen Gründen und zur besseren Übersicht, wurde dieser in drei Teile mit je zwei Tools aufgegliedert. Der vollständige und ganzheitliche Katalog wird im Anhang 1 abgebildet.

		HBase/Hadoop	Cassandra
DB-Eigenschaften	Datenbank Typ	Wide Column Store	Wide Column Store
	Implementierungssprache	Java	Java
	Datenschema	schemafrei	Schemafrei
	verwendbare Programmiersprachen	C, C#, C++, Java, PHP, Python, Scala, Groovy	C#, C++, Java, JavaScript, PHP, Python, Scale, u.a.
Bewertungskriterien	Typisierung	nein	Ja
	Sekundärindizes	nein	nur Gleichheitsabfragen
	Partitionierung	Sharding	Sharding
	Replikation	frei wählbar	frei wählbar
	MapReduce	ja	Ja
	Konsistenz	Immediate Consistency	Eventual Consistency Immediate Consistency

⁴²Vgl. solid IT (o. J.)

		HBase/Hadoop	Cassandra
			individuelleinstellbar
	Zugriffskontrolle	durch Access Control Lists (ACL) (Hadoop/ZooKeeper)	für einzelne Objekte
	StoredProcedures	ja	Nein
	parallele Datenmanipulation	ja	Ja
	Beständigkeit/dauerhafte Speicherung	ja	Ja
	SQL	nein	Nein
Bewertung DB-Eigenschaften	\sum erfüllte Kriterien	8	9
Zugriffsmöglichkeiten	JAVA API	x	
	RESTful HTTP API	x	
	C++ API		
	Thrift	x	X
	Sonstige		CQL
Bewertung Zugriffsmöglichkeiten	\sum erfüllte Kriterien	3	2

Tabelle 9: Kriterienkatalog HBase/Hadoop / Cassandra

Der erste Auszug des Kriterienkatalogs zeigt die Datenbanksysteme HBase/Hadoop und Cassandra auf. Zu erkennen sind die Unterschiede in den Bereichen Typisierung und Sekundärindizes, welche Cassandra im Gegensatz zu Hadoop abdeckt. Auch in dem Gebiet Konsistenz liegt durch die zusätzliche Methode Eventual Consistency ein Vorteil bei Cassandra. StoredProcedures wird hingegen nur von Hadoop erfüllt.

		Hypertable	Accumulo
DB-Eigenschaften	Datenbank Typ	Wide Column Store	Wide Column Store, Key-Value Store, Graph DBMS, Document Store
	Implementierungssprache	C++	JAVA
	Datenschema	Schemafrei	Schemafrei
	verwendbare Programmiersprachen	C++, Java, Perl, PHP, Python, Ruby	C, C#, C++, Java, JavaScript, Perl, PHP, Python, Ruby, u.a.
Bewertungskriterien	Typisierung	Nein	Ja
	Sekundärindizes	nur Gleichheits- und Präfixabfragen	Ja
	Partitionierung	Sharding	Sharding
	Replikation	wählbarer Faktor (Filesystemebene)	Ja
	MapReduce	Ja	ja
	Konsistenz	Immediate Consistency	Immediate Consistency
	Zugriffskontrolle	nicht vorhanden	auf Zellebene
	StoredProcedures	Nein	Nein
	parallele Datenmanipulation	Ja	Ja
	Beständigkeit/dauerhafte Speicherung	Ja	Ja
	SQL	Nein	Nein
Bewertung DB-Eigenschaften	\sum erfüllte Kriterien	7	9

Zugriffsmöglichkeiten	JAVA API		X
	RESTful HTTP API		X
	C++ API	X	
	Thrift	X	X
	Sonstige		Accumulo Shell, JDBC, ODBC
Bewertung Zugriffsmöglichkeiten	∑ erfüllte Kriterien	2	6

Tabelle 10: Kriterienkatalog Hypertable / Accumulo

Der nächste Abschnitt in Tabelle 11 bildet die Tools Hypertable und Accumulo ab. Auch hier besteht ein Unterschied im Bereich der Typisierung. Des Weiteren wird die Zugriffskontrolle von Accumulo auf Zellebene umgesetzt, wohingegen Hypertable keine Methode beinhaltet.

		Amazon SimpleDB
DB-Eigenschaften	Datenbank Typ	Key-Value Store
	Implementierungssprache	-
	Datenschema	Schemafrei
	verwendbare Programmiersprachen	C, C++, Java, PHP, Ruby, Scala, .Net, Erlang, Python

Bewertungskriterien	Typisierung	Nein
	Sekundärindizes	automatische Indizierung aller Spalten
	Partitionierung	Nein
	Replikation	Ja
	MapReduce	Nein
	Konsistenz	Eventual Consistency Immediate Consistency
	Zugriffskontrolle	durch AWS Identity and Access Management (IAM)
	StoredProcedures	Nein
	parallele Datenmanipulation	Ja
	Beständigkeit/dauerhafte Speicherung	Ja
	SQL	Nein
Bewertung DB-Eigenschaften	\sum erfüllte Kriterien	6
Zugriffsmöglichkeiten	JAVA API	
	RESTful HTTP API	X
	C++ API	
	Thrift	
	Sonstige	
Bewertung Zugriffsmöglichkeiten	\sum erfüllte Kriterien	1

Tabelle 11: Kriterienkatalog AmazonSimpleDB

Der letzte Abschnitt zeigt das Tool AmazonSimple DB, welches im Gegensatz zu den zuvor beschriebenen Systemen eine Software-as-a-Service Lizenz beinhaltet. Hierbei werden weder das Kriterium Typisierung, noch MapReduce abgedeckt. Auch die Funktion Stored Procedures ist kein Bestandteil von AmazonSimpleDB.

4.3 Bewertung

Datenbanken finden heutzutage eine große Bandbreite an Einsatzgebieten. Je nach Szenario muss gründlich abgewogen werden, welche Kriterien beziehungsweise Eigenschaften zwingend vorhanden sein müssen und ob bestimmte Methoden favorisiert werden sollten. Aus diesem Grund kann nicht verallgemeinert werden welches der beschriebenen Tools prinzipiell am geeignetsten ist. Auch eine Gewichtung der einzelnen Kriterien ist an dieser Stelle, durch die unterschiedlichen Ausgangssituationen nicht sinnvoll. Dennoch wurde anhand der Anzahl der erfüllten Kriterien eine Beurteilung erstellt, welche in der letzten Zeile jeder Kategorie des Kriterienkatalog ersichtlich ist. Die Gesamtbewertung beinhaltet die anteilige Summen beider Bewertungen, das heißt die Datenbankeigenschaften wurden mit 11 Kriterien gewichtet, die Zugriffsmöglichkeiten mit einem Kriterium.

		HBase/Hadoop	Cassandra	Hypertable	Accumulo	Amazon SimpleDB
Bewertung DB-Eigenschaften	Σ erfüllte Kriterien (Anzahl: 11)	8	9	7	9	6
Bewertung Zugriffsmöglichkeiten	Σ erfüllte Kriterien (Anzahl: 1)	3	2	2	6	1
Bewertung Gesamt	Σ erfüllte Kriterien	7,58	8,42	6,58	8,75	5,58
Ranking		3	2	4	1	5

Tabelle 12: Bewertung Tools

Ersichtlich ist das Accumulo mit 8,75 die beste Bewertung erhält, gefolgt von Cassandra mit 8,42. HBase/Hadoop befindet sich im Ranking auf Platz 3 und Hypertable auf dem Vierten. Der Web-Service AmazonSimple DB belegt mit 5,58 hingegen den letzten Platz.

Zu erwähnen ist zudem das Kriterien, wie zum Beispiel die Benutzerfreundlichkeit oder Verbreitung nicht berücksichtigt wurden, aber dennoch ein wichtiger Aspekt sind. Aus diesem Grund wird an dieser Stelle zusätzlich auf das „DB-Engines Ranking von Wide Column Stores“ des IT-Beratungsunternehmens *solidITconsulting&softwaredevelopment gmbh* eingegangen. Hierbei wurde eine Bewertung einer Anzahl von spaltenorientierten Datenbanken anhand der folgenden Kriterien erstellt:

- Nennung der Datenbank auf Websites (über Google und Bing)
- Allgemeines Interesse an dem System (über Google Trends)
- Technische Diskussionen über das System (über Stack Overflow und DBA Stack Exchange)
- Anzahl der Job-Angebote (über Indeed und Simple Hired)
- Anzahl der Profile in Netzwerken (über LinkedIn)
- Relevanz in sozialen Netzwerken (über Twitter)⁴³

Unter Betrachtung der genannten Aspekte wurde das wie in der folgenden Abbildung dargestellte Ranking erstellt. Die Tools Amazon Simple DB wird bei dieser Erhebung nicht berücksichtigt.

5 Systeme im Ranking, Juli 2014


Rang	Vormonat	DBMS	Datenbankmodell	Punkte	Änderung
1.	1.	Cassandra	Wide Column Store	81,58	-0,26
2.	2.	HBase	Wide Column Store	41,75	+0,95
3.	3.	Accumulo	Wide Column Store	2,56	+0,13
4.	4.	Hypertable	Wide Column Store	0,61	+0,02
5.	5.	Sqrrl	Multi-Model 	0,05	+0,01

Abbildung 7: Ranking Wide Column Stores solidIT⁴⁴

Zu erkennen ist, dass das Datenbanksystem Cassandra mit einem großen Unterschied (39,83 Punkte Differenz zu HBase) an erster Stelle steht. Gefolgt mit 41,75 Punkten von HBase. Die Systeme Accumulo, Hypertable und Sqrrl bilden mit einer Gesamtpunktezahl von 3,22 die am wenigsten verwendeten, beziehungsweise genannten Systemen.

⁴³ Vgl. solid IT (2014)

⁴⁴ Vgl. Ebenda

Auch im Rahmen dieses Projektes wurde der NoSQL Prototyp durch das System Cassandra erstellt. Bei der Entscheidung bezüglich der Auswahl des Tools, wurden nicht nur die Eignung anhand der Funktionen berücksichtigt, sondern auch die unkomplizierte Installation und der Umfang der Dokumentation. Im Gegensatz zu den weiteren untersuchten Systemen liegen die Vorteile bezüglich der genannten Aspekte sowie in Sachen Benutzerfreundlichkeit und Komplexität eindeutig bei Cassandra.

5. Prototyp

Ein Prototyp dient zum praktischen Testen mit verschiedenen Testdaten. Dabei wurde eine MySQL sowie Wide-Column-Datenbank erstellt und getestet. Im ersten Schritt werden die Testdaten und Abfragen formuliert, im zweiten eine kurze Beschreibung der Benutzeroberfläche und zum Schluss die eigentliche Datenbank.

5.2 Testdaten

Der Test soll einen Vergleich zwischen einer MySQL und einer spaltenorientierten Wide-Column-Datenbank umsetzen. Dazu werden der Aufbau, sowie die Möglichkeiten dieser beiden Lösungen gegenüber gestellt und verglichen. Dazu dient das Beispiel eines Versicherungsunternehmens.

Das Unternehmen besitzt verschiedene Verträge und einen Kundenstamm von 10 Versicherten. Hierzu werden die Stammdaten von Kunden, die Betreuer und die Verträge in verschiedenen Tabellen gespeichert (siehe Abbildung 8). Es werden nun Abfragen generiert. Dabei werden die folgenden Anweisungen beachtet:

- SELECT
- INSERT
- DELETE

SELECT-Anweisungen dienen zur Ausgabe von Daten welche aus unterschiedlichen Tabellen stammen können. Auf Grund der vielen verschiedenen Möglichkeiten, welche die Abfrage bietet, werde zehn Anweisungen durchgeführt und im folgenden beschrieben. Dabei erhöht sich die Komplexität nach und nach.

1) Suche alle Datensätze, die sich in der Tabelle „Kunde“ befindet.

Select *

FROM Kunde;

2) Suche alle Versichertendaten, von jenen Versicherten, die eine Lebensversicherung abgeschlossen haben.

```
Select * FROM Verträge  
Where Vertrag_Art = 2;
```

3) Suche alle Versichertennamen, die mit dem Buchstaben T beginnen.

```
SELECT Versicherter_nachname, Versicherter_vorname FROM Versicherter  
WHERE Versicherter_nachname < 'T';
```

4) Suche alle Versichertendaten von Kunden, deren Handynummer mit 0178 beginnen.

```
Select * From Versicherter  
Where Versicherter_Telefon_Handy like "0178%";
```

5) Suche alle Namen und Telefonnummern der Versicherten. Sortiere diese Daten nach den Nachnamen der Versicherten.

```
Select Versicherter_Nachname, Versicherter_Telefon_Privat From Versicherter  
Order By Versicherter_Nachname;
```

6) Suche Nachname, Vorname und Geburtsdatum aller Versicherten, die zwischen 1970 und 2014 geboren sind.

```
Select Versicherter_Nachname, Versicherter_Vorname, Versicherter_Geburtsdatum  
AS GebFrom Versicherter  
Where Versicherter_Geburtsdatum Between '01.01.1970' AND '01.01.2014';
```

7) Füge in die Tabelle Betreuer einen neuen Mitarbeiter ein.

```
INSERT intobetreuer (Betreuer_ID, Betreuer_name, Betreuer_nachname,  
betreuer_Alter) VALUES (6 , "Timm", "Timmson", 29);
```

8) Löschen Sie den Betreuer Timmson.

DELETE -- Betreuer_ID, Betreuer_name, Betreuer_nachname, betreuer_Alter

From betreuer

Diese Abfragen werden mit Hilfe der MySQL-Workbench getestet. Hierfür werden die Zeiten dokumentiert. Die kleine Datenmenge beträgt 10 Datensätze, wobei die große 3000 umfasst. Das Ergebnis ist in der folgenden Tabelle dargestellt.

5.2. Benutzeroberfläche

Dieses Kapitel beschreibt kurz die allgemeine Benutzeroberfläche der MySQL-Workbench.

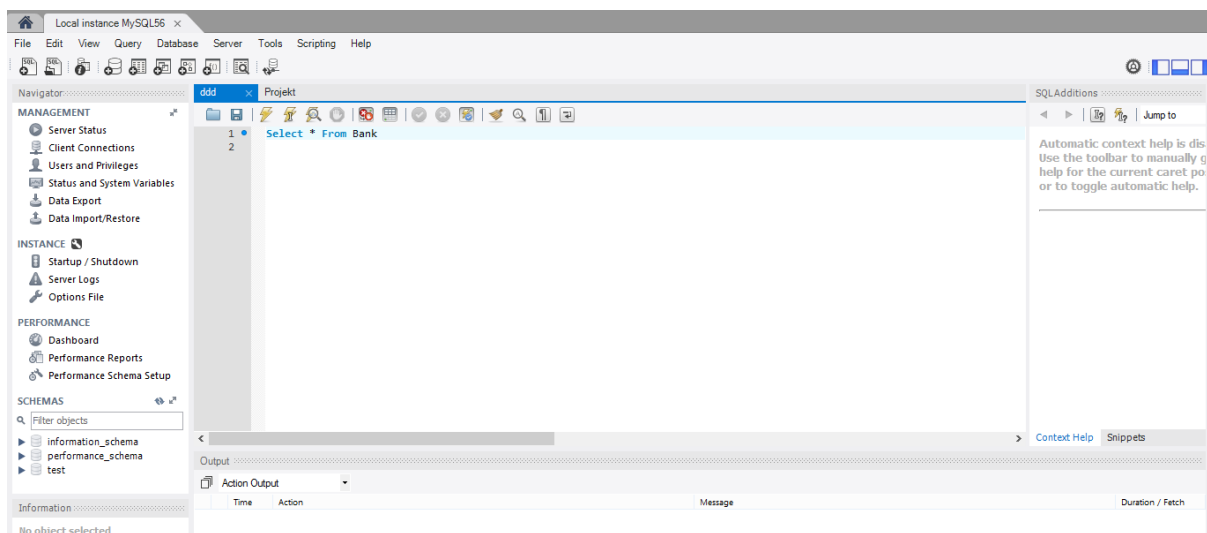


Abbildung 8: MySQL Workbench

Die Abbildung zeigt den Aufbau der Datenbank. Zu sehen ist der Bereich, in welchen die verschiedenen SQL-Anweisungen eingefügt werden können. Durch das Klicken auf den Ausführen-Button, werden die verschiedenen Anweisungen ausgeführt. Die Anwendung lässt sich einfach bedienen. Ein Nachteil zeigt sich hierbei in der Integration der Daten. Sie müssen über Anweisungen importiert werden, da sie sonst nicht gespeichert werden können. Die Oberfläche ist minimalistisch gehalten und die wichtigsten Funktionen können über einen Klick ausgeführt werden. Diese sind zum Beispiel das Anlegen einer neuen Tabelle, eines neuen Skripts oder einer neuen Datenbank.

Cassandra besitzt, wie schon beschrieben, keine separate Benutzeroberfläche. Aus diesem Grund wurde auf eine Abbildung sowie Beschreibung an dieser Stelle verzichtet.

5.1. Aufbau der Datenbank

In diesem Kapitel wird lediglich beschrieben wie die Datenbank auf Cassandra aufgesetzt wurde. Die MySQL Datenbank dient dabei zum Vergleich der unterschiedlichen Systeme sowie zur Veranschaulichung, da Cassandra keine grafische Oberfläche bietet.

Als Prototyp wurde Cassandra auf einem Windows 8 System installiert. Um diesen mit einer relationalen Datenbank vergleichen zu können, wurde auf demselben Gerät eine MySQL Datenbank aufgesetzt. Die Datenbanken wurden von den Strukturen sowie den Inhalten identisch aufgebaut. D.h. Es wurden Tabellen mit gleichen Namen und Attributen angelegt sowie die Datensätze eingefügt. So sind die beiden Anwendungen miteinander vergleichbar. Die folgende Abbildung zeigt den Aufbau.

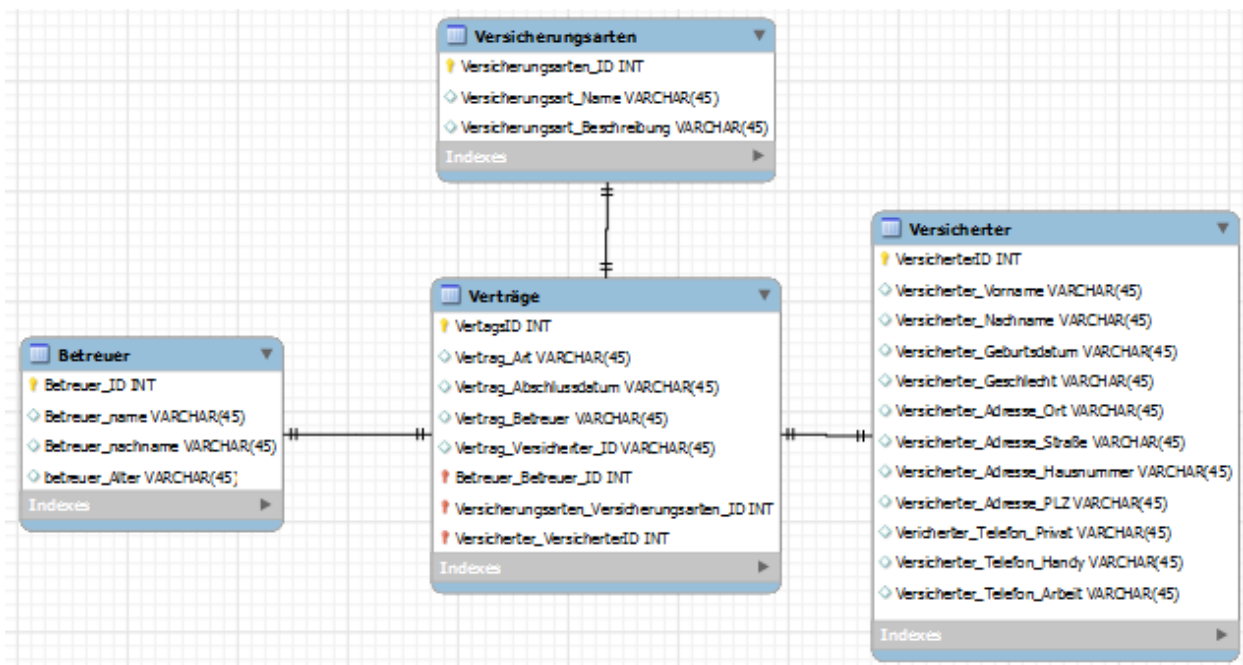


Abbildung 9: Aufbau der MySQL-DB

Da lediglich MySQL eine grafische Oberfläche bietet, stammt die schematische Darstellung von eben dieser. Wie zu sehen ist, gibt es die Tabellen **Betreuer**, **Versicherungen**, **Versicherter** sowie **Verträge**. Um sie Tabellen anlegen zu können, muss zunächst eine Datenbank aufgesetzt werden.

Über den Befehl „DESC KEYSPACE“ ist es möglich, die bisher vorhandenen Datenbanken anzuzeigen. Angelegt werden Datenbanken mit dem Befehl „CREATE KEYSPACE“. Dem folgt dann der Name sowie die Eingabe einige Parameter. Daraufhin legt der Anwender die

dazugehörigen Tabellen mit den Attributen an. Dazu wird zunächst mit „USE project;“ in den Keyspace gewechselt. Mit Hilfe des Kommandos „CREATE TABLE“ werden die einzelnen Tabellen angelegt. Darunter folgen die einzelnen Attribute und deren Datentypen. Es muss immer ein PRIMARY KEY angegeben werden, sonst funktioniert der Befehl nicht. Der erste Befehl für die Tabelle Betreuer lautet demnach:

```
Cql:project> CREATE TABLE betreuer (betreueridint, betreuer_altervarchar, betreuer_nachnamevarchar, betreuer_namevarchar, PRIMARY KEY (betreuerid));
```

Wie bei der Tabelle „Betreuer“ wurde auch bei „Versicherungsart“, „Versicherter“ und „Verträge“ vorgegangen. Mit dem Befehl „DESC SCHEMA“ kann dies überprüft werden. Wenn mehrere Tabellen angelegt sind, werden alle angezeigt.

Dem folgt das Einfügen der Datensätze. Dies kann einfach mit einem „INSERT INTO“ Befehl erfolgen. Unter MySQL ist es etwas einfacher, da die Daten aus der zuvor angelegten Excel Tabelle importiert werden können. In jeder Tabelle wurden ca. 10 Datensätze angelegt.

Unter Cassandra können die Tabelle, wie in Abbildung 9 zu sehen, ausgegeben werden. In beiden Datenbanksystemen sind die Tabellen damit eindeutig als solche darstellbar. Dies erfolgt über den SQL Befehl „SELECT * FROM Betreuer“.

```
cqlsh:project> SELECT * FROM betreuer;
```

betreuerid	betreuer_alter	betreuer_nachname	betreuer_name
5	34	Gregson	Greg
1	28	Svenson	Sven
2	22	Bobson	Bob
4	29	Paulson	Paul
3	45	Franzson	Franz

```
(5 rows)  
cqlsh:project>
```

Abbildung 10: Benutzeroberfläche Cassandra

Die Abfragen wurden auf beiden Systemen wie in Unterkapitel 5.2 dargestellt durchgeführt. Auf MySQL funktionierten die Abfragen ohne Probleme und haben eine Laufzeit von unter einer Sekunde. Wie bereits in Kapitel 3.3 beschrieben, ist dies unter Cassandra aufgrund eingeschränkter Abfragemöglichkeiten nicht so einfach. Die Einschränkungen können zwar teilweise umgangen werden, es ist jedoch wesentlich aufwändiger. In Cassandra ist es ledig-

lich möglich nach PRIMARY KEY zu sortieren, daher sollte schon im Vorfeld überlegt werden, welches Attribut dafür sinnvoll ist.

Bei der Abfragedauer gibt es jedoch bei so geringen Datenmengen keine erkennbaren Unterschiede.

6. Fazit

In dieser Arbeit wurden verschiedene Wide-Column-Datenbanken untersucht. Für die Vergleichbarkeit wurde ein Kriterienkatalog erstellt und die ausgewählten Systeme in diesen eingefügt. Anschließend wurde die Anwendung mit der besten Bewertung ausgewählt. In diesem Fall entschieden sich die und ein Prototyp erstellt. Es wurde eine vergleichbare MySQL Datenbank entworfen und dem Prototypen gegenüber gestellt.

Wide-Column-Datenbanken bieten eine gute Alternative zu relationalen Datenbanken. Diese müssten jedoch noch ausreichend mit einer größeren Anzahl von Datensätzen getestet werden, um eindeutige Unterschiede feststellen zu können. Es wurde zunächst versucht mit HBase eine Datenbank aufzusetzen, jedoch war dies sehr schwierig, da keine gute Anleitung dafür zu finden war. Dafür mussten zudem einige Programme zusätzlich installiert werden. Cassandra als virtuelle Maschine war dies sehr viel einfacher und bedurfte wenig Aufwand. Daher sollte bei der Auswahl einer Wide-Column Datenbank vor allem auf die Dokumentationen dazu geachtet werden. Denn dies ist für die Verwendung essentiell. Zudem bietet Cassandra auf der Webseite planetcassandra.org eine Vielzahl an Video-Tutorials, mit denen sich der erste Kontakt mit Cassandra relativ einfach gestaltet. Zudem gibt es sehr viel Hilfestellung in Foren.

Die Theorieteile stellten keine größere Herausforderung dar. Zu dem Thema Wide-Column-Datenbank sowie NoSQL und MySQL konnten viele geeignete Quellen identifiziert und verwendet werden. Die theoretischen Beschreibungen der Datenbanken gestalteten sich wiederum komplizierter. Beispielsweise waren für Apache Accumulo nur wenige Literatur zu finden. Diese unterschieden sich zudem noch.

Der Prototyp stellte die Autorinnen zusätzlich vor eine große Herausforderung. Die Installation sowie die Arbeit mit Cassandra erwiesen sich durch die nicht Vorhandene Benutzeroberfläche und spezielle Kommandos als problematisch. Nach einiger Einarbeitungszeit konnten diese aber erfolgreich bewältigt werden. Die MySQL Datenbank konnte einfach installiert werden. Die Datenimplementierung stellte wiederum eine Herausforderung dar, da diese nur

über Anweisungen funktioniert und kleinere Schreibfehler sofort zu Funktionsfehlern führten. Aber auch dieses wurde im Laufe des Projekts erfolgreich bewältigt.

Das Ergebnis zeigt, dass sich die Unterschiede vor allem in der Benutzeroberfläche sowie der Anwendbarkeit und der Zeit befinden. Das geeignetste Tool stellt Cassandra dar, da dieses die beste Bewertung in dem Kriterienkatalog erhalten hat. Es hat vergleichsweise kurze Bearbeitungszeiten, ist jedoch nicht so einfach zu handhaben. Aus diesem Grund eignet sich das Tool speziell für geübte Benutzer. Schlussendlich ist die Anwendung von einer Wide-Column-Datenbank bei einer Anzahl großer Daten für ein Unternehmen zu empfehlen. Die Alternative einer relationalen Datenbank ist jedoch ebenfalls möglich.

Anhang 1

DB-Eigenschaften	Bewertungskriterien				
	HBase/Hadoop	Cassandra	Hypertable	Accumulo	Amazon SimpleDB
Datenbank Typ	Wide Column Store	Wide Column Store	Wide Column Store	Wide Column Store, Key-Value Store, Graph DBMS, Document Store	Key-Value Store
Implementierungssprache	Java	Java	C++	JAVA	-
Datenschema	schemafrei	schemafrei	schemafrei	schemafrei	schemafrei
verwendbare Programmiersprachen	C, C#, Java, PHP, Python, Scala, Groovy	C#, C++, Java, JavaScript, PHP, Python, Scala, u.a.	C++, Java, Perl, PHP, Python, Ruby	C, C#, Java, JavaScript, Perl, PHP, Python, Ruby, u.a.	C, C++, Java, PHP, Ruby, Scala, .Net, Erlang, Python
Typisierung	nein	ja	nein	ja	nein
Sekundärindizes	nein	nein	nur Gleichheits- und Präfixabfragen	ja	automatische Indizierung aller Spalten
Partitionierung	Sharding	Sharding	Sharding	Sharding	nein
Replikation	frei wählbar	frei wählbar	wählbarer Faktor (Filesystemebene)	ja	ja
MapReduce	ja	ja	ja	ja	nein
Konsistenz	Immediate Consistency	Immediate Consistency individuell einstellbar	Immediate Consistency	Immediate Consistency	Eventual Consistency Immediate Consistency
Zugriffskontrolle	durch Access Control Lists (ACL) (Hadoop/ZooKeeper)	für einzelne Objekte	nicht vorhanden	auf Zellebene	durch AWS Identity and Access Management (IAM)
Stored Procedures	ja	nein	nein	nein	nein
parallele Datenmanipulation	ja	ja	ja	ja	ja
Beständigkeit/dauerhafte Speicherung	ja	ja	ja	ja	ja
SQL	nein	nein	nein	nein	nein
Σ erfüllte Kriterien	8	9	7	9	6
Bewertung DB-Eigenschaften					
Zugriffsmöglichkeiten	JAVA API	x	x	x	x
	RESTful HTTP API	x	x	x	x
	C++ API	x	x	x	x
	Thrift	x	x	x	x
	Sonstige	COQL		Accumulo Shell, JDBC, ODBC	
Σ erfüllte Kriterien	3	2	2	6	1
Bewertung Gesamtwertung	7,58	8,42	6,58	8,75	5,58

Literaturverzeichnis

- Berekoven, L./Eckert, W./Ellenrieder, P. (2006):** Marktforschung, Methodische Grundlagen und praktische Anwendung, in: Gabler Lehrbuch, 11., überarb. Aufl, Wiesbaden: Gabler
- Edlich, S./u. a. (2010):** NoSQL, Einstieg in die Welt nichtrelationaler Web-2.0-Datenbanken, München: Hanser
- Fankhauser, K./Wälty, H. F. (2011):** Marktforschung, Grundlagen mit zahlreichen Beispielen, Repetitionsfragen mit Antworten und Glossar, in: Betriebswirtschaftslehre, 3., überarb. Aufl, Zürich: Compendio Bildungsmedien
- Fowler, M./Sadalage, P. (2012):** NoSQL distilled. Boston, Mass, London: Addison-Wesley.
- Hammann, P./Erichson, B. (2005):** Marktforschung, 71 Tabellen, in: UTB für Wissenschaft, 805. Jg., 5., überarb. und erw. Aufl, Stuttgart: Lucius & Lucius
- Kotler, P./Schellhase, R. (2011):** Grundlagen des Marketing, in: Grundlagen des Marketing
- Scharf, A./Schubert, B./Hehn, P. (2009):** Marketing, Einführung in Theorie und Praxis, 4., überarb. und erw. Auflage, Stuttgart: Schäffer-Poeschel
- Wöhe, G./Döring, U. (2013):** Einführung in die allgemeine Betriebswirtschaftslehre, in: Vahlens Handbücher der Wirtschafts- und Sozialwissenschaften, 25., überarb. und aktualisierte Aufl, München: Vahlen

Verzeichnis der Internetquellen

- Amazon Web Services (2014a):** Amazon SimpleDB (Beta),
<http://aws.amazon.com/de/simplifiedb/>,
Abruf: 10.06.2014
- Amazon Web Services (2014b):** Amazon SimpleDB -Produktdetails,
<http://aws.amazon.com/de/simplifiedb/details/>,
Abruf: 10.06.2014
- Bogaard, M. (2011):** NoSQL, Big Data | HBase, Hypertable, Cassandra und Co.,
<http://www.internet-programmierung.net/datenbanken/nosql-big-data-hbase-hypertable-cassandra/>,
Abruf: 27.06.2014
- Bösswetter D. (2010):** Spaltenorientierte Datenbanken,
<http://www.gi.de/service/informatiklexikon/detailansicht/article/spaltenorientierte-datenbanken.html>,
Abruf: 27.05.2014
- DJ Walker-Morgan (2014):** NoSQL im Überblick,
<http://www.heise.de/open/artikel/NoSQL-im-Ueberblick-1012483.html/>,
Abruf: 30.06.14
- Edlich, S./Friedland, A./al., Jens Hampe et (2011):** NoSQL, Einstieg in die Welt nichtrelationaler Web 2.0 Datenbanken, 1. Aufl, s.l: Carl Hanser Fachbuchverlag
- Apache Software Foundation (2014):** <http://hadoop.apache.org/>, What is Hadoop?
Abruf: 06.06.2014
- Fromm R./Metzger R. (2014):** Big Data Analytics mit Stratosphere am Beispiel von Risk Management bei der Deutschen Telekom,
<http://www.bitkom-bigdata.de/programm/2014-03-26/big-data-analytics-mit-stratosphere-am-beispiel-von-risk-management-bei-der->,
Abruf: 12.06.2014
- Hahn Projects (o. J.):** Column Family,
<http://hahnpro.com/wp-content/uploads/2012/11/ColumnFamily23.png>,
Abruf: 06.06.2014
- Hypertable Inc. (o. J.):** Why Choose Hypertable?,
http://hypertable.com/why_hypertable/,
Abruf: 27.06.2014
- Jansen R. (o. J.):** Cassandra im Einsatz,
<http://www.heise.de/developer/artikel/Einsatz-der-Cassandra->

- Datenbank-1349878.html?artikelseite=2,
 Abruf: 28.06.2014
- o.V. (2009):** Begriffserklärung Marktforschung/ Marketingforschung,
<http://www.teialehrbuch.de/Kostenlose-Kurse/Marketing/15162-Begriffserklaerung-Marktforschung-Marketingforschung.html>,
 Abruf: 22.03.2014
- o.V. (2014a):** Accumulo,
<https://accumulo.apache.org/>,
 Abruf: 22.06.2014
- o.V. (2014b):** Accumulo vs Cassandra,
<http://vschart.de/vergleich/apache-accumulo/vs/apache-cassandra>,
 Abruf: 20.06.2014
- o.V. (2014c):** Release Notes 1.6.0,
https://accumulo.apache.org/release_notes/1.6.0.html,
 Abruf: 20.09.2014
- Planet Cassandra (o. J.):** Companies Using NoSQL Apache Cassandra,
<http://planetcassandra.org/companies/>,
 Abruf: 28.06.2014
- Schnelle J. (2011):** Cassandra – Die Datenbank hinter Facebook,
<http://www.pro-linux.de/artikel/2/1531/cassandra-die-datenbank-hinter-facebook.html>,
 Abruf: 28.06.2014
- solid IT (2014):** DB-Engines Ranking von Wide Column Stores,
<http://db-engines.com/de/ranking/wide+column+store>
- solid IT (o. J.):** Enzyklopädie Index,
<http://db-engines.com/de/articles>,
 Abruf: 05.07.2014
- Wolfgang Klimt (26.06.2013):** Was Sie über NoSQL wissen sollten,
<http://www.computerwoche.de/a/was-sie-ueber-nosql-wissen-sollten,2528753>,
 Abruf: 28.06.2014
- Wübbenhorst, K. (03.2014):** Marktforschung,
<http://wirtschaftslexikon.gabler.de/Definition/marktforschung.html>,
 Abruf: 25.03.2014