Prototyping

Praxisarbeit

vorgelegt am 17.02.2003

von

Tabea Zwiener

Ausbildungsbereich	Wirtschaft
Fachrichtung:	Wirtschaftsinformatik
Jahrgang:	2001
Halbjahr:	3. Semester
Ausbildungsleiter:	Stefan Winkler
Ausbildungsstätte:	merlin.zwo InfoDesign GmbH Karmelstr. 9 75378 Bad Liebenzell

Inhaltsverzeichnis

1. Abbildungsverzeichnis

2. Einführung

2.1. Problemdefinition

In der Vergangenheit trat bei Softwareprojekten häufig das Problem auf, dass ein Programm bei der Übergabe an den Kunden, nicht dessen Vorstellungen entsprach, obwohl die Vorgaben von den Entwicklern korrekt programmiert und umgesetzt wurden.

Das Problem lag dabei aber nicht an falschen Vorgaben, sondern meist daran dass Entwickler und Kunde verschieden Teile unterschiedlich interpretiert haben. Der Kunde und der Programmierer leben in ”verschiedenen Welten” und sprechen ”verschiedene Sprachen”. Dem Programmierer fehlt meist der nötige Abstand zu den Vorgaben, er sieht sofort die technische Realisierung und nicht die Verwendung beim Kunden. Der Kunde seinerseits hat entweder das fertige System im Kopf und schon genaue Vorstellungen über Aussehen und Funktionalität oder er hat noch gar keine Vorstellung. Ihm fällt es jedoch meist schwer seine Gedanken in Worte und somit in Vorgaben zu formulieren. Selbst ein guter Projektleiter, der sich mit dem Kunden intensiv auseinander setzt, hat oft Schwierigkeiten, diesem das genaue Bild zu entlocken.

Folgende Abbildung veranschaulicht die Problematik:

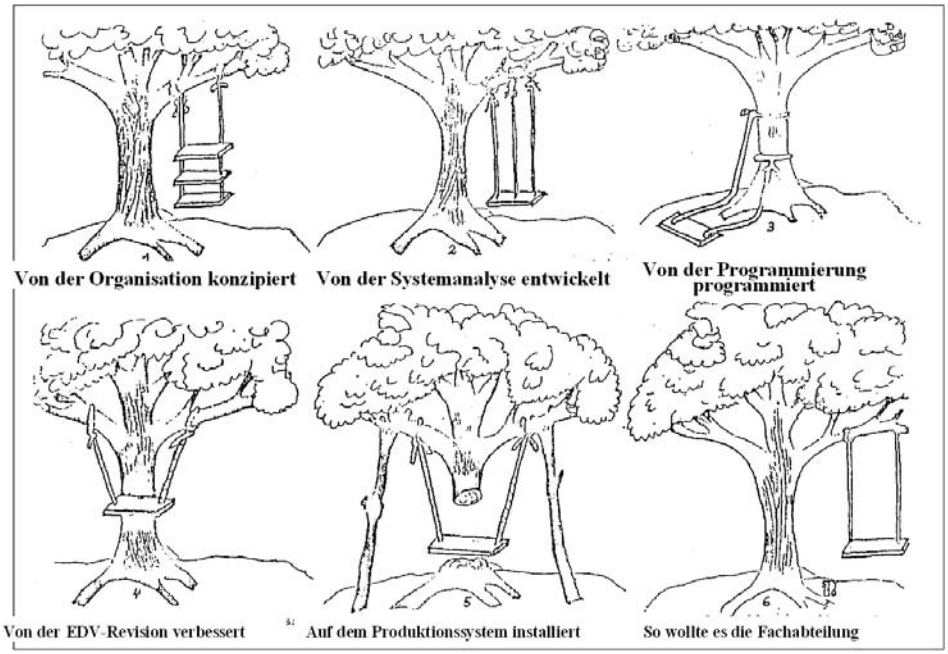


Abbildung 1: Kommunikationprobleme zwischen den Projektmitgliedern [1]

Doch wie soll man die Wünsche und Vorstellungen des Kunden erfüllen? Und woher weiß man wie der Kunde einzelne Teile interpretiert? Und welcher Vorstellungen er hat?

[1] Quelle: Prof. Dr. Schwille BA

Dazu müsste man dem Anwender während des Projekts eine prüffähige Version des Anwendungssystems vorlegen, anhand derer er Fehlinterpretationen oder Änderungswünsche feststellen und äußern kann. Hierbei sollen die Ideen der Kunden angeregt werden.

2.2. Vorgehensweise

Das oben geschilderte Problem kann mit dem Konzept des Prototyping verringert werden. Dieses Konzept wird zur Zeit an einem Projekt in meinem Ausbildungsunternehmen erprobt.

In dieser Praxisarbeit möchte ich nun zuerst das Konzept des Prototyping erläutern und anschließend die praktische Anwendung anhand des Projekts Ferdi III[1] erläutern.

[1] Erläuterungen siehe Anlage 3

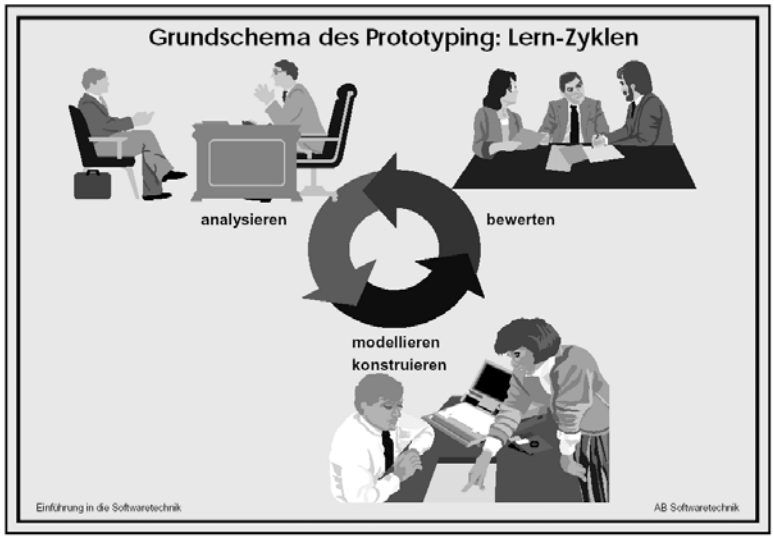
3. Der Grundgedanke des Prototyping

Der Grundgedanke des Prototyping besteht darin, während der Entwicklung eines Anwendungssystems ein Muster (Prototyp) zu erstellen. Dieser Prototyp kann unterschiedliche Zwecke, Ziele und einen unterschiedlichen Umfang haben. Ob es nur um die Funktionalität des Programms oder um den Aufbau der Masken geht, hängt von der Art des Prototyping ab, welche man verwendet.

Beim Prototyping kann man grob zwei Ansätze unterscheiden, bei dem ersten wird vom Groben und Abstrakten zum Konkreten und Detail entwickelt.

Beim diesem Prinzip des Prototyping beginnt man mit einer Analyse des Grundproblems beim Kunden (siehe analysieren in Abbildung 2). Anschließend erstellt man eine Rohform des Programmes (siehe modellieren konstruieren in Abbildung 2), dieses wird dann bewertet (siehe Abbildung 2) und anschließend dem Kunden zur Ansicht vorgelegt, dieser testet den Prototypen und äußert eventuell Änderungsvorschläge. (siehe wieder analysieren Abbildung 2)

Mit diesen Änderungsvorschlägen geht man wieder zur Analyse, diese nimmt die neuen Vorschläge und Kritikpunkte auf und integriert Sie in das Gesamtmodell, diese gehen dann wieder zur Konstruktion, dort werden diese implementiert und so nähert sich das ganze langsam dem Endsystem an. Wenn man diese Art des Prototyping verwendet, ist es möglich



fertige Teilsysteme freizugeben und bereits zu verwenden.

Abbildung 2: Grundschema des Prototyping

Der zweite Ansatz wird allgemein als „Wasserfallmodell“ bezeichnet, dies entspricht dem Vorgehen nach dem Phasenschema[1]. Dabei gibt es die Phasen Analyse und Entwurf am Anfang des Projekts. Wenn der erste Prototyp fertig ist, wird dieser dem Kunden vorgelegt.

[1] Siehe Erläuterungen hierzu Anlage 1

Der Kunde kann seine Änderungen und Anregungen äußern. Anschließend wird ein ganz neues System entworfen. Hier ist es nicht möglich vorzeitig, das System oder Teilsysteme davon vorzeitig einzusetzen, da dass System bei jeder Analyse komplett verändert werden kann.

Dies ist aber nur eine Art das Prototyping zu unterscheiden, man kann den Prototypen auch zum Testen verschiedener Funktionalitäten oder zum Experimentieren verwenden. (Siehe hierzu die Erläuterungen in den folgenden Kapiteln)

4. Voraussetzungen

Bevor man mit dem Prototyping arbeitet, ist es wichtig das richtige Arbeitsklima für das Prototyping zu schaffen. Hierfür muss man einiges beachten und festlegen: [1]

1. Akzeptanz herstellen, sowohl beim Kunden als auch beim Management und der Projektleitung
2. Klare Abbruchkriterien festlegen
3. Zweck des Prototypen festlegen
4. Häufige Kontrolle und Feedback vereinbaren
5. Prototyp muss als solcher angesehen werden, sonst kann es dazu führen, dass dieser bereits von den Anwendern verwendet wird und das ganze Projekt zum Stillstand kommt. Deshalb sollte ein Projektleiter darauf achten, dass Prototypen nicht vorschnell für die echte Anwendung eingesetzt werden.
6. Den Benutzer darauf einstellen, dass er immer konstruktive Kritik äußern soll.
7. Immer darauf achten, dass trotz des Prototyping die Phasen (nach dem Phasenschema der Softwareentwicklung[2]) Analyse und Entwurf sorgfältig gemacht werden. Diese verschmelzen zwar in das Prototyping mit ein, man sollte aber gerade deshalb aufpassen, weil es sonst zur Programmierung auf Zuruf hinausläuft, bei der man anschließend die Änderungen nicht mehr nachvollziehen kann.
8. Prüfen, dass mit der Realisierung nicht zu früh begonnen wird. Dies hängt von der Art des gewählten Prototyping ab. Es gibt die Möglichkeit einen gleitenden Übergang zu verwenden.
9. Außerdem braucht man einen guten Softwareentwickler und ein gutes Werkzeug. Der Entwickler sollte sein Werkzeug virtuos beherrschen, damit Änderungen schnell vorgenommen werden können.
10. Der Softwareentwickler muss auf den Kunden eingehen und dessen Wünsche schnell auffassen und realisieren können.

---

[1] Punkte 1.-4. M.Schneider-Hufschmid Punkte 5. Usability Forum Punkte 6. – 7. Stahlknecht/Hasenkamp

[2] siehe hierzu Phasenschema Anlage 1

5. Arten

Es gibt verschiedene Arten von Prototypen aus denen dann verschiedene Arten des Prototyping entstehen. Das bedeutet, wenn man eine Art von Prototyp wählt, gibt es dazu passend ein Prototyping Konzept. Wie diese zusammenhängen wird im folgenden Kapitel erläutert.

5.1. Arten von Prototypen

- Wegwerfprotoypen
- Wiederverwendbare Prototypen
- Vollständige Prototypen
- Unvollständige Prototypen

5.2. Arten von Prototyping

- Rapid Prototyping
- Evolutionäres Prototyping
- Exploratives Prototyping
- Experimentelles Prototyping
- Vertikales Prototyping
- Horizontales Prototyping
- Slide Show[1]
- Wizard of Oz[2]

[1] wird in den folgenden Kapiteln erläutert

[2] wird in den folgenden Kapiteln erläutert

6. Zusammenhänge

6.1. Verwendung:

Hier werden die Prototypen hinsichtlich ihrer verschiedenen Verwendungsarten unterschieden. Dabei gibt es zwei Arten von Prototypen und das zugehörige Prototyping Konzept:

6.1.1. Wegwerfprototypen

Diese werden meist zur Sammlung von Erfahrungen erstellt und anschließend nicht wieder verwendet um das eigentliche System zu erstellen. Diese Art nennt man *rapid prototyping*. [1]
Es kann für das *rapid prototyping* verschiedene Gründe geben. Der Wegwerfprototyp dient zum Beispiel nur als Vorführmodell, welches sich der Kunde anschauen und austesten kann. Oder als Testmodell an welchem der Programmierer verschiedene Funktionen oder Darstellungen ausprobiert.
Anschließend wird der Prototyp wie der Name schon sagt weggeworfen und das eigentliche System wird völlig neu und unabhängig davon erstellt.

6.1.2. Wiederverwendbare Prototypen

Auch hier ist der Name sehr sprechend, der Prototyp wird nicht wie der Wegwerfprototyp (6.1.1) nach der Verwendung nicht mehr gebraucht und das eigentliche System völlig neu und unabhängig erstellt. Sondern dieser Prototyp wird wiederverwendet und weiterentwickelt. Diese Art des Prototyping wird *evolutionäres Prototyping* genannt. Der *wiederverwendbare Prototyp* (der auch *Pilotsystem* genannt wird) wird erweitert, verbessert und den Bedürfnissen des Kunden schrittweise angepasst. [2]

6.2. Zusammenhang mit dem Phasenschema

6.2.1. Exploratives Prototyping

Diese Art von Prototyping hängt eng mit dem Phasenschema[3] der Systementwicklung zusammen. Es konzentriert sich sehr auf den fachlichen Entwurf, vorwiegend wenn dieser schlecht, ungenau und unklar ist. Das bedeutet, dass vor allem funktionale Teile des Programms als Prototyp umgesetzt werden[4].

---

1 Stahlknecht/Hasenkamp Einführung WI

2 Stahlknecht/Hasenkamp Einführung WI

3 Siehe hierzu Anlage 1

4 Definition aus Stahlknecht/Hasenkamp Einführung WI

Eine genauere Definition ist: *Exploratives Prototyping* wird in der Anforderungsanalyse angewandt, meist um die Anforderungen von Benutzern und Managern an das künftige System zu klären. [1]

6.2.2. Experimentelles Prototyping

Das *experimentelle Prototyping* beschäftigt sich vorwiegend mit Alternativen der informationstechnischen Realisierung. Diese umfasst unter anderen die Strukturen der Daten und Programme, die Schnittstellen zwischen den Programmteilen oder die Benutzeroberfläche.[2] Diese dienen meist nur zu Versuchszwecke, und werden aufgrund dessen auch *Labormuster* genannt.

Beim experimentellen Prototyping stehen technische Fragen im Vordergrund. Es soll dem Entwickler helfen, die Machbarkeit eines Systems einzuschätzen und Lösungsvorschläge zu erarbeiten. [3]

6.3. Umfang des Prototypen

6.3.1. Vollständige Prototypen

Die *vollständigen Prototypen* befassen sich hauptsächlich mit den funktionalen Teilsystemen des Anwendungssystems. Diese Vorgehensweise nennt man *vertikales Prototyping*. Ein Beispiel für ein solches funktionales Teilsystem wäre während der Einführungsphase[4] die Erprobung des Anwendungssystems an einem Arbeitsplatz stellvertretend für die spätere Nutzung an mehreren gleichartigen Arbeitsplätzen.[5]

Beim *vertikalen Prototyping* werden einige Teile des zukünftigen Systems bis ins Detail simuliert. [6]

6.3.2. Unvollständige Prototypen

Hier wird wie der Name schon sagt von einem *unvollständigen Prototypen* ausgegangen. Das heißt eine einzelne Schicht aus dem Anwendungssystem wird erstellt. Die zugehörige Vorgehensweise nennt man *horizontales Prototy*ping. Eine bevorzugte und oft gewählte Schicht ist die Benutzerschnittstelle. Zum Beispiel werden in enger Zusammenarbeit von Benutzer und Systementwickler Gestaltung der Bildschirmfenster, Aufbau der Bildschirmoberflächen oder Masken für Dateneingabe und -ausgabe entwickelt. Hierbei wird die Funktionalität des Programmes in den Hintergrund gestellt. Inhalte spielen keine Rolle,

---

1 Definition aus Usability Forum

2 Definition aus Stahlknecht/Hasenkamp Einführung WI

3 Definition aus Usability Forum

4 Siehe hierzu Phasenschema Anlage 1

5 Definition aus Stahlknecht/Hasenkamp Einführung WI

6 Definition aus Usability Forum

Daten sind meist fest eingetragen oder berechnende Programmteile werden durch manuelle Berechnungen überbrückt. [1]

Beim *horizontalen Prototyping* wird ein möglichst breiter Prototyp erzeugt, der einen Überblick über das ganze System bieten soll, wobei eventuell auf Details verzichtet wird.[2]

6.4. Sonderformen

6.4.1. Slide Show

Eine *Slide Show* ist eine Menge von Bildschirmen, die dem Benutzer in einer bestimmten Reihenfolge gezeigt werden. Dadurch wird auf eine simple Art und Weise ein Ablauf in der Benutzerschnittstelle simuliert. Einfache *Slide Show* Prototypen können mit Papier und Bleistift, Grafikprogrammen oder Präsentationsprogrammen erzeugt werde. Trotz der Einfachheit dieser Techniken ermöglicht sie die Verwendung von Graphiken und groben Abläufen und kann daher sehr gut zu Kommunikationszwecken genutzt werden. [3]

6.4.2. Wizard of Oz

Bei dieser Technik simuliert ein Mensch den Computer. Einsatzmöglichkeiten ergeben sich zum Beispiel, wenn Prototyping mangels geeigneter Tools nicht möglich ist. Für Frage und Antwort Dialoge und für natürlichsprachliche Ein- und Ausgaben ist diese Technik sehr gut geeignet und vor allem sehr kostengünstig. [4]

---

1 Definition aus Stahlknecht/Hasenkamp Einführung WI

2 Definition aus Usability Forum

3 Definition aus Usability Forum

4 Definition aus Usability Forum

7. Bewertung

7.1. Pro Prototyping

- Prototypen sind Kommunikationsmittel
- Prototypen zeigen die Machbarkeit von Systemen
- Prototypen helfen Designentscheidungen zu treffen
- Prototypen sind Schulungs- und Verkaufsmittel
- Beim Prototyping kann man Kosten und Zeit sparen, wenn man auf die unter Voraussetzung aufgezählten Punkte achtet
- Der Kunde identifiziert sich besser mit den Masken und dem Programm allgemein
- Beide Parteien bekommen ein besseres Gefühl für die Masken und Programmabläufe
- Verständnisprobleme zwischen Kunde und Programmierer werden früh erkannt

7.2. Kontra Prototyping

- Prototyping verursacht Mehrkosten, wenn man die unter Voraussetzung aufgeführten Punkte nicht beachtet
- Prototyping braucht mehr Zeit, wenn man zu viele Änderungen zulässt
- Oft werden die Phasen Analyse und Entwurf zu Gunsten des Prototyps stark gekürzt oder ganz weggelassen. Das funktioniert aber nicht, das Prototyping ersetzt diese Phasen auf keinen Fall
- Es fällt schwer den Kunden konstruktive Kritik zu entlocken, die den Prototypen wirklich verbessert
- Prototyping braucht viel Kommunikation, Qualifikation und Einsatzbereitschaft beider Parteien
- Wenn man keinen fixen Endzeitpunkt festsetzt, dann kann das Projekt zu einem „Selbstläufer“ werden und die Anwender wollen immer wieder Änderungen und haben ständig neue Ideen
- Sinnvolle Vorschläge aus den vielen Vorschlägen heraus filtern
- Der Kunde gibt keine Kritik, am Ende gefällt ihm aber die Anwendung nicht

7.3. Anwendung

Das Prototyping ist sowohl getrennt in den einzelnen Phasen der Systementwicklung (Phasenschema[1]) als auch phasenübergreifend anwendbar. Bei richtiger Handhabung läßt sich durch Prototyping eine Verkürzung der Entwicklungszeit erreichen.

Auf keinen Fall ersetzt das Prototyping die normale Entwicklung mit dem Phasenschema der Systementwicklung. Dies ist ein in der Systementwicklung häufig begangener Fehler. Das

[1] Das Phasenschema wird im Anhang kurz erläutert. Siehe hierzu Anlage 1

funktioniert aber nicht, das Prototyping ergänzt und unterstützt die Vorgehensweise mit dem Phasenschema nur, es ersetzt sie nicht. Ansonsten wäre keine Kontrolle, Überwachung und Steuerung des Projektes von Seiten des Managements mehr möglich.
Wie beim Grundgedanke schon erläutert, gehen Phasenschema und Prototyping ineinander über, das Prototyping erfordert mehrere Analysen und Entwürfe.
Der Einsatz von Werkzeugen, die das Prototyping unterstützen, kann sehr zweckmäßig sein.

7.4. Nutzen

Der Nutzen wurde in der Vergangenheit oft überbewertet, außerdem läßt sich das Prototyping nicht bei allen Entwicklungen anwenden. Man muss im Voraus abwägen, ob in dem betreffenden Projekt der einzubringende Einsatz den gewünschten Nutzen stiftet.
Ein Fall in dem Prototyping einen Nutzen bringt, ist wenn der Kunde nur eine geringe Vorstellung von dem zu erstellenden Programms hat.
Im praktischen Einsatz muss man unbedingt die unter Voraussetzung aufgeführten Punkte beachten.

8. Zusammenfassung der Ergebnisse

Ein Prototyp ist ein ablauffähiges Modell des Anwendungssystems. Er realisiert ausgewählte Aspekte des zukünftigen Systems.[1]

1. Prototypen sind eine Diskussions- und Entscheidungsbasis für Entwickler und Anwender
2. Prototypen dienen den Beteiligten zum Experimentieren und zum Sammeln von Erfahrungen
3. Prototypen helfen, relevante Spezifikations- oder Entwicklungsprobleme zu klären. Sie werden durch schriftliche Spezifikationen ergänzt.

Es verschiedene Arten von Prototyping, die je nach Projekt mehr oder weniger sinnvoll sind. Wichtig ist, dass zuvor eine gründliche Analyse gemacht wird, ob die Anwendung des Prototyping sinnvoll ist oder nicht.
Darüber hinaus ist zu beachten, dass das Prototyping unter keinen Umständen die Phasen der Systementwicklung ersetzt, sondern diese nur ergänzt. Bei richtiger Verwendung des Prototyping kann die Entwicklungszeit verkürzt werden.

[1] Entnommen aus FU Berlin

9. Vorkommen im Unternehmen

9.1. Vorkommen

In meinem Ausbildungsunternehmen sind wir noch in der Erprobungsphase, das heißt, wir haben Prototyping noch nicht so häufig in Projekten verwendet.

Wir verwenden das Prototyping weil:

- der Kunde keine optische Vorstellung der künftigen Software hatte
- keine abstraktes Design des Anwendungssystems möglich war
- ein neuer Ansatz ausgetestet werden sollte, da in der Vergangenheit solche Vorgaben mit dem Prinzip der Slide Show erstellt wurden. Diese führte jedoch zu zahlreichen Interpretationsproblemen. Da jedoch keine Interaktion möglich war, konnte sich der Kunde den Programmablauf nur schwer vorstellen.

Aus den oben genannten Gründen testen wir nun den Ansatz des Prototyping.

Da der Kunde bei dem Projekt Ferdi III[1] nur eine sehr geringe Vorstellung des Endprodukts hatte, schien es uns hier sinnvoll den Ansatz des Prototyping dort auszutesten.

9.2. Realisierung

Ferdi III wird in Delphi[2] realisiert. Für diese Realisierung ist hauptsächlich einer meiner Kollegen verantwortlich, er baut Änderungswünsche ein und führt das Programm in regelmäßigen Abständen dem Kunden vor. Vor allem nach Änderungen und Anpassungen ist eine solche Vorführung notwendig.

[1] Wird in Anlage 3 beschrieben

[2] Wird in Anlage 4 beschrieben

10. Art des verwendeten Prototyping

Nun werde ich die oben erläuterten Prototypingtypen und -konzepte analysieren und erläutern, welche wir bei Ferdi III verwendet haben. Ich werde diese in der gleichen Reihenfolge und Kategorien wie oben erläutern

10.1. Grundgedanke

Wir gehen nicht nach dem Wasserfallmodell vor, sondern wir gehen vom Groben zum Feinen, wie in der unten stehenden Graphik erläutert wird.

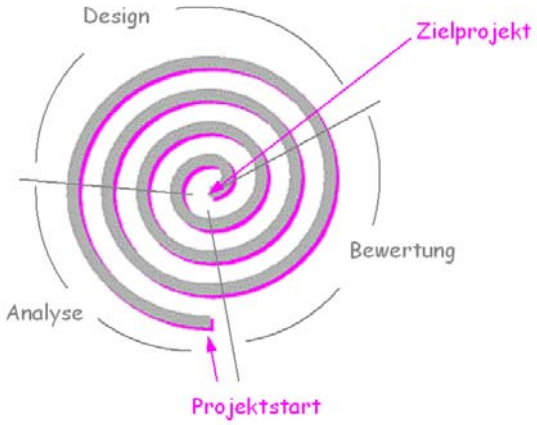


Abbildung 3: Spiraldarstellung des Prototyping

Am Anfang steht der Projektstart. Das Projekt beginnt mit einer Analysephase, anschließend wird das System modelliert. In der Abbildung befinden wir uns beim Design. Ist das Design abgeschlossen, wird dieser erste Prototyp bewertet und vom Kunden begutachtet. Dieser reicht Änderungswünsche und Anregungen ein. Wir gehen erneut in die Analysephase, jedoch machen wir keine neue Analyse sondern erweitern die vorherige. Anschließend wenden wir uns wieder dem Design zu und so weiter. Mit jedem Schritt nähern wir uns dem Zielprojekt. So kommen wir von einer groben Programmdarstellung zu einem ausgeprägten Anwendungsprogramm.

Bei der von uns verwendet Art des Prototyping, geht der Prototyp fließend in das Anwendungsprogramm über. Bei uns ist es nicht so, dass zu einem Zeitpunkt das komplette System an den Kunden übergeben wird, sondern es werden immer Teilbereiche an den Kunden übergeben. Diese Teilsysteme sind funktional immer vollständig. Das heißt diese Anwendung wird weiterentwickelt auch während des Einsatzes, werden neue selbstständige Teile implementiert.

10.2. Verwendung

Bei Ferdi III verwenden wir wie in 6.1.2 geschildert einen *wiederverwendbaren Prototyp*. Wir haben diesen Ansatz gewählt, da Ferdi III ein sehr umfangreiches System ist und häufig vom Kunden geprüft und vom Entwickler weiterentwickelt wird.

Wir verwenden folglich *evolutionäres Prototyping.* Der Kunde kann seine Wünsche und Änderungen anbringen und das System wird danach angepasst und verbessert.

In regelmäßigen Abständen wird das System dem Kunden vorgestellt und von diesem getestet und beurteilt.

10.3. Zusammenhang mit dem Phasenschema

Ferdi III wird mit dem Phasenschema[1] der Systementwicklung entwickelt. Es gibt einen fachlichen Entwurf und Design, nur ist es hier so, dass diese Schichten immer wieder vorkommen. Sie werden wie in Abbildung 3 dargestellt häufig wiederholt.

Es wird folglich *Explorative*s *Prototyping* angewandt.

10.4. Umfang des Prototypen

Ferdi III ist ein sehr umfangreiches Anwendungssystem und daher ist es wichtig, dass die Maske und Benutzerführung übersichtlich bleibt. Die Funktionalität, die dahinter liegt, steht zur Zeit noch im Hintergrund. Die Daten, die später auf der Maske zu sehen sein sollen, sind ebenfalls noch nicht relevant, diese sind zur Zeit fest eingetragen.

Es handelt sich ausschließlich um die Maske und den Maskenfluß, nicht um die Funktionalität oder die Daten, daraus folgt, dass wir einen *unvollständiger Prototyp* verwenden. Das heißt eine einzelne Schicht aus dem Anwendungssystem wird erstellt. Folglich verwenden wir bei Ferdi III *horizontales Prototy*ping. in unserem Fall die Benutzerschnittstelle. Bei Ferdi III wird wie im Beispiel oben in 7.3.2 erläutert wird in enger Zusammenarbeit von Benutzer und Systementwickler die Gestaltung der Bildschirmmasken entwickelt.

1 Siehe hierzu Anlage 1

11. Analyse

In diesem Teil der Arbeit möchte ich nun das Vorgehen, die Probleme und Vorteile, die in meinem Ausbildungsunternehmen aufgekommen sind, schildern.

11.1. Voraussetzungen

Natürlich mussten wir darauf achten das alle Voraussetzungen (wie in Kapitel 4 erläutert) erfüllt werden. Aber einige dieser Voraussetzungen sind für uns besonders wichtig, andere wiederum nicht. Deshalb möchte ich die Voraussetzungen an dieser Stelle speziell für uns bewerten.

1. Häufige Kontrolle und Feedback
   Das ist bei uns ebenfalls sehr wichtig,
2. Prototyp muss als solcher angesehen werden, sonst kann es dazu führen, dass dieser bereits von den Anwendern verwendet wird und das ganze Projekt einschläft, dann muss das Projekt wieder geweckt werden. Deshalb sollte ein Projektleiter darauf achten, dass Prototypen nicht vorschnell für die echte Anwendung eingesetzt werden.
   Bei uns wird der Prototyp zum Programm, deshalb wird der Prototyp nicht
3. Den Benutzer darauf vorbereiten, dass er immer konstruktive Kritik äußern soll.
4. Immer darauf achten, dass trotz des Prototyping die Phasen (nach dem Phasenschema der Softwareentwicklung) Analyse und Entwurf sorgfältig gemacht werden. Diese fließen in das Prototyping ein, man sollte aber gerade deshalb aufpassen, weil es sonst zu Programmierung auf Zuruf hinausläuft.
   Das ist ein sehr wichtiger und zentraler Punkt, vor allem muss man auf die Hinweise von den Anwendern achten, denn diese können den zeitlichen und finanziellen Rahmen des Projektes sprengen.
5. Prüfen, dass mit der Realisierung nicht zu früh begonnen wird. Dies hängt von der Art des gewählten Prototyping ab. Es gibt die Möglichkeit einen gleitenden Übergang zu verwenden.
   Wir verwenden den Ansatz, in dem der Prototyp langsam in die Anwendung übergeht.
6. Außerdem braucht man einen guten Softwareentwickler und ein gutes Werkzeug. Der Entwickler sollte sein Werkzeug virtuos beherrschen, damit Änderungen schnell vorgenommen werden können. Gegebenenfalls sogar während der Projekt Besprechnung.
   Dieser Punkt ist bei uns ebenfalls sehr wichtig und auch erfüllt.
7. Der Softwareentwickler muss auf den Kunden eingehen und dessen Wünsche schnell auffassen und realisieren können.
   Das ist bei allen Entwicklungen wichtig, deshalb auch für uns. Da wir den Kunden und den jeweiligen Projektleiter bereits aus anderen Projekten kennen, ist die Kommunikation

zwischen den Parteien bereits gut. Aufgrund dessen ist dieser Punkt bei uns kein Problem.

11.2. Probleme und Risiken

- Da viele Personen zum Kreis der Testenden gehören, gibt es sehr viele Vorschläge. Deshalb besteht bei uns das Problem, dass man aus vielen verschiedenen Vorschlägen die Guten heraus suchen muss.
- Es besteht das Risiko der „Hey-Joe-Programmierung“[1] (Programmierung auf Zuruf), da wir den Kunden bereits gut kennen ruft dieser bei neuen Ideen im Unternehmen an und schildert diese. Jder Kunde hofft, dass diese Änderungen sofort implementiert werden. Das ist nicht möglich, da man dann an der Analyse und dem Entwurf vorbei programmieren würde.
- Ein weiteres Problem das bei uns auftrat war, dass „Auswüchse“ oder Sonderwünsche, die nicht direkt mit dem zu erstellenden Programm in Verbindung stehen aufkamen. Das mussten wir unterbinden, denn so etwas kann einen Projektrahmen sprengen.
- Prototyping erfordert eine hohe Disziplin seitens des Projektleiters, da dieser darauf achten muss, dass die oben genannten Auswüchse nicht oder nur wenige davon vorkommen.

11.3. Vorteile

- Ein großer Vorteil der bei uns aufgetreten ist, war dass der Kunde und der Programmierer ein gutes Gefühl für die Masken und die Maskenabläufe bekommen. Und aufgrund dessen konnte der Kunde gute Vorschläge und Kritik äußern.
- Außerdem fand bei uns eine bessere und frühere Identifikation mit den Masken und dem Programm im allgemeinen statt.
- Bei uns konnten aufgrund des Prototyping Kosten gesenkt und die Projektzeit minimiert werden.

[1] Dieser Begriff wurde von unserem Kunden häufig verwendet und er trifft genau das was ich hier sagen möchte, damit ist Programmierung auf Zuruf gemeint

12. Schlußfolgerung

Wir werden Prototyping wieder anwenden, wenn wir auf ähnlich optimale Voraussetzungen treffen. Ein solcher Fall wäre für uns, wenn der Kunden nur wenig oder gar keine Vorstellung von dem Endprogramms hat.

Für uns hat sich das Prototyping gelohnt, da wir den Kunden gut kennen und die Kommunikation zwischen den Parteien gut war. Deshalb konnten wir das Projekt schneller und kostengünstiger abschließen, als mit unserem „normalen“ Projektablauf.

Außerdem war der Kunde ebenfalls zufrieden, und aufgrund dessen steht einem weiteren Einsatz des Prototyping nichts im Weg.

Anlagenverzeichnis

Anlagen

Anlage 1

Phasenschema[1]:

Das Phasenschema der Systementwicklung ist folgendermaßen aufgebaut:

Phasenbezeichnung	Phaseninhalt
Vorphase	Projektbegründung
Analyse	Istanalyse - Erhebung des Istzustandes - Bewertung des Istzustandes Sollkonzept - Fachentwurf - Grobentwurf - Wirtschaftlichkeitsanalyse
Entwurf	Systementwurf Programmspezifikation Programmentwurf
Realisierung	Programmierung Test
Einführung	Systemfreigabe Systemeinführung

Abbildung 4: Phasenschema der Systementwicklung

Das Phasenschema zeigt die Reihenfolge in welcher die Aktivitäten eines Projektes durchzuführen sind. Der Gesamtprozess der Systemanalyse besteht aus mehreren aufeinander folgenden Stufen mit den vier Grundphasen Systemanalyse, Systementwicklung, Systemeinführung und Systempflege. Von dieser Grundeinteilung ausgehend, sind für die Entwicklungsphasen von Anwendungssystemen unter der Bezeichnung Phasenkonzept verschiedene Vorschläge entwickelt worden. Hier ist ein mögliches Beispiel dargestellt.

[1] Definition entnommen aus Stahlknecht/Hasenkamp Einführung in die WI Seite 213

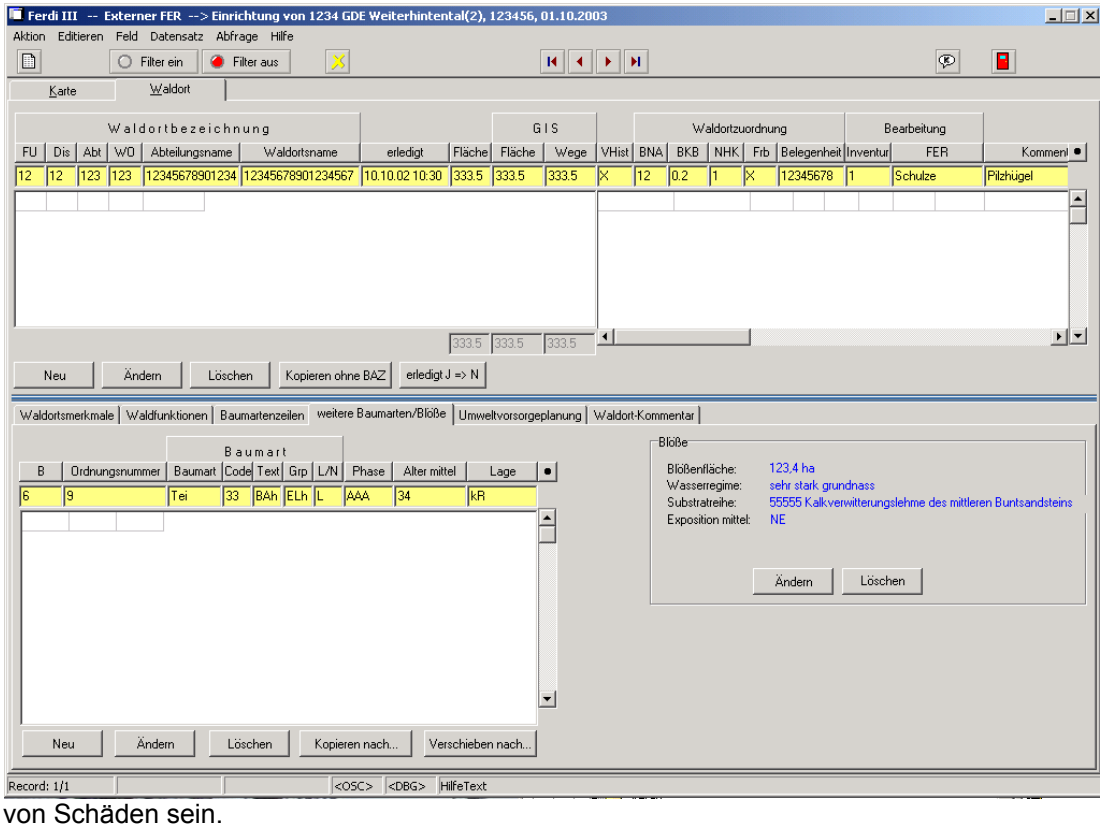
Anlage 2

Ferdi III:

Ferdi III ist eine Anwendung, die wir für das Ministerium für Umwelt und Forsten in Rheinland-Pfalz erstellen. Bei Ferdi III handelt es sich um ein zentrales Inventur- und Planungssystem für Wald. Jeder Waldbesitzer muss eine Inventur und eine Planung für seinen Wald vornehmen, dies ist gesetzlich verankert. Der zentrale Begriff ist **Nachhaltigkeit**, das bedeutet, dass jeder Waldbesitzer seinen Wald so planen und anlegen muss, dass auf Dauer der Waldbestand erhalten bleibt. Jeder darf folglich nur soviel Holz aus dem Wald entnehmen, wie wieder nachwächst.

Der Planungs- und Inventurzeitraum beträgt bei Wald 10 Jahre. Eine Inventur und Planung wird immer von einem Forsteinrichter vorgenommen.

Der Forsteinrichter erhält eine Karte (Waldbereich) und einen zuständigen Revierbeamten, zusammen gehen diese beiden durch den Wald und nehmen alles auf, was an Inventur aufzunehmen ist. Dieses wird dann in das Ferdi III eingegeben. Das kann zum Beispiel Anzahl und Art der Bäume, Größe der Blößen (Freie Flächen) oder das Ausmaß und die Art



von Schäden sein.

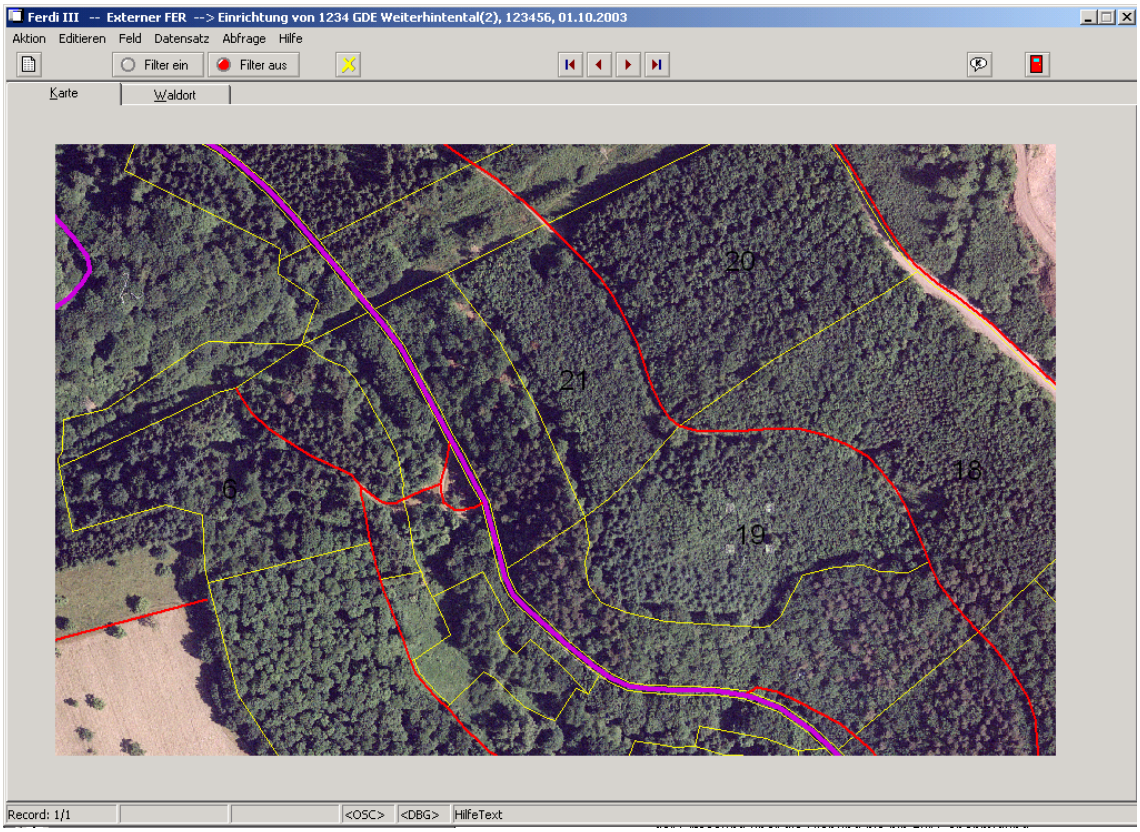
Abbildung 5: Ferdi III Maske Sachdaten [1]

[1] Screenshot aus Ferdi III

Die Erfassung der Daten werden in einer Maske (siehe Abbildung Ferdi III Maske Sachdaten) eingegeben.

Nachdem die Inventur vorgenommen wurde, das heißt alle Walddaten wurden im Ferdi III erfasst, beginnt der zweite Teil, die Planung. Nun darf der Waldbesitzer das Ziel festlegen, hierbei ist er an gesetzliche Richtlinien gebunden. Diese Ziele werden ebenfalls in Ferdi III eingegeben.

Ferdi III besteht aber nicht nur aus dem oben geschilderten Sachdatenteil, sondern auch aus einem Geodatenteil (geologische Daten). Ein Wald liegt in verschiedenen Gebieten. Es gibt politische (Gemeinden, Länder, Städte,...) und natürliche Grenzen (Bodenbeschaffenheit, Naturschutzgebiete, ...) diese Gebiete werden in verschiedenen Karten dargestellt. Diese Karten werden übereinander gelegt, diese Vorgehensweise nennt man Layering. Durch dieses Übereinander legen der Karten erhält man alle geologischen Daten über den jeweiligen Wald.



Eine solche Karte wird in Abbildung 6 dargestellt.

Abbildung 6: Ferdi III Maske Geodaten[1]

Die Sach- und Geodaten werden unabhängig voneinander erfasst. Wenn beide Teile vollständig erstellt wurden, dann werden sie vereint und ausgegeben.

[1] Screenshot aus Ferdi III

Diese Ausdrucke werden den jeweiligen Stellen, zum Beispiel dem Waldbesitzer, dem zugehörigen Forstamt oder der Oberfinanzdirektion vorgelegt. Wenn alles in Ordnung ist werden die Inventur und die Planung freigegeben. Diese Planung gilt dann wieder für die nächsten 10 Jahre.

Ferdi III unterstützt folglich die Inventur und Planung des Waldes in Rheinland-Pfalz von der Erfassung über die Planung bis hin zur Genehmigung.

Anlage 3

Delphi:[1]

Bei Delphi handelt es sich um eine visuelle Entwicklungsumgebung (IDE) der US-amerikanischen Firma Borland. Die Programmiersprache, mit der unter Delphi programmiert wird, heißt Object Pascal, eine vollständig objektorientierte Weiterentwicklung von Borland Pascal. Nun wird diese Sprache "**Delphi Language**" genannt.

Die Delphi- Entwicklungsumgebung (siehe Abbildung 7) läuft unter Windows und erzeugt seit Version 2 32-Bit-Windows-Anwendungen. Seit 2001 gibt es mit Kylix auch eine IDE für Linux, die mit der Delphi-Sprache arbeitet. Delphi ist bekannt für seine **einfache Bedienbarkeit**. Dabei haben die Borland-Entwickler Albert Einsteins Worte berücksichtigt: "Alles sollte so einfach wie möglich gemacht werden, aber nicht einfacher." Es ist somit sehr einfach, mit ein paar Mausklicks eine Anwendung zusammenzustellen und laufen zu lassen. Anfänger haben recht schnell Erfolgserlebnisse. Es ist mit Delphi jedoch genauso möglich, komplexe Anwendungen zu programmieren.

Außerdem können mit Delphi alle Möglichkeiten der Objektorientierung eingesetzt werden. Es besteht jedoch kein Zwang dazu (im Gegensatz zu Java).

Klassen können visuell gekapselt und als sog. **Komponenten** weitergegeben werden. So gibt es im Internet einige Seiten, auf denen sich Delphi-Programmierer kostenlos mit solchen Komponenten versorgen können, die sie dann in eigenen Programmen einsetzen können. Delphi selbst ist im Lieferzustand bereits mit ungefähr 100 Komponenten ausgestattet.

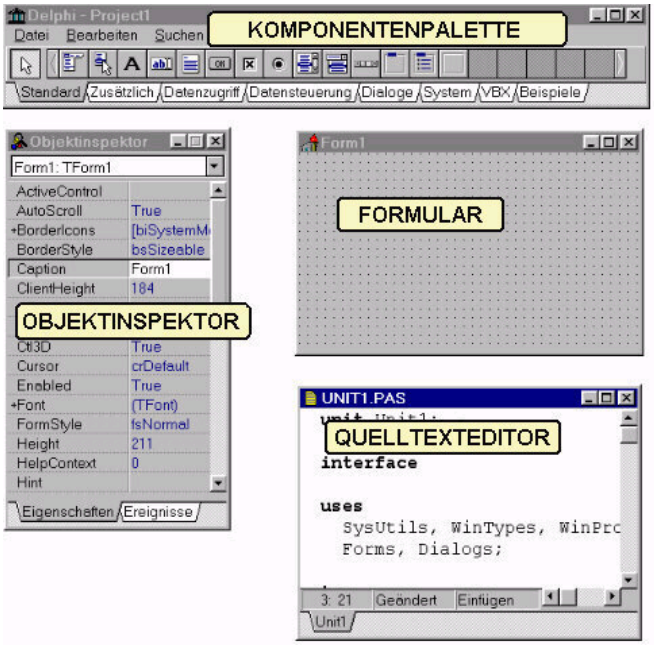


Abbildung 7: Delphi Entwicklungsumgebung

[1] Text: Delphi Grundlagen Grafik: Plauener

13. Literaturverzeichnis

Delphi Grundlagen

entnommen aus:

www.grundlagen.delphi-source.de/delphi/delphi.shtml

Autor: nicht angegeben

FU Berlin

entnommen aus:

www.inf.fu-berlin.de/lehre/WS01/SWT/Prototyping.pdf

Autoren: Christiane Floyd, Guido Gryczan, Julian Mack

M.Schneider-Hufschmid

entnommen aus:

www.mmk.ei.tum.de/~alt/ebof/ulss01/ebof_SS01_ul09.pdf

Autor: M.Schneider-Hufschmid

Plauener

entnommen aus:

http://www.plauener.de/lessing/delphi/delphi02.htm#2.1.

Prof. Dr. Schwille BA

entnommen aus:

Skript: Projektmanagement

Autor: Prof. Dr. Jürgen Schwille Berufsakademie Stuttgart

Stahlknecht/Hasenkamp Einführung in die WI

entnommen aus:

Stahlknecht und Hasenkamp Einführung in die Wirtschaftsinformatik 10.Auflage

Autoren: Peter Stahlknecht, Ulrich Hasenkamp

Usability-Forum

entnommen aus:

www.usability-forum.com/bereiche/Prototyping.pdf

Autor: nicht angegeben