

Diplomarbeit 199
Hochschule für Technik und Architektur Bern
Abteilung Informatik



Version: 0.8
Datum: 10. 06. 2003

Autoren:
A. Hirter
O. Fankhauser
S. von Niederhäusern

Betreuer:
Dr. J. Boillat
P. Fierz

Experte:
Dr. F. Flückiger

<http://www.latex2xml.org>

Management Summary

Projektbeschreibung

Hier werden die Ausgangssituation und die Projektziele aufgelistet. Zudem werden die Projektrisiken und mögliche Lösungsvorschläge erörtert.

Vorgehensmodell

Wir wollen für unser Projekt ein agiles Vorgehensmodell verwenden. In diesem Kapitel sind die Phasen mit ihren Aktivitäten und Ergebnissen festgehalten.

Standards und Richtlinien

In diesem Teil werden die Programmierrichtlinien und die Vorgaben für die Dokumentationen geregelt.

Projektplanung

Der Projektplan umfasst mehrere Phasen. Das Projekt beginnt mit einer Situationsanalyse, anschliessend folgende mehrere iterative Entwicklungszyklen.

Projektorganisation

In unserer Projektorganisation erfolgt keine fixe Zuteilung von Aufgaben. Dafür werden in diesem Kapitel die Verantwortlichkeiten für die Koordination bestimmter Teilgebiete zugewiesen.

Konfigurationsmanagement

Agile Softwareentwicklung mit häufig ändernden Anforderungen und kurzen Iterationszyklen erfordert ein gutes Konfigurationsmanagement. Für die Quellcodeverwaltung und Versionierung setzen wir CVS ein.

Qualitätssicherung

Dieses Kapitel zeigt auf, mit welchen Mitteln die Qualität gesichert werden soll. Wir werden zu diesem Zweck im Rahmen der Projektarbeit JUnit näher betrachten.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Zweck des Dokumentes	1
1.2	Beteiligte	1
1.3	Projekt Homepage	1
2	Projektbeschreibung	2
2.1	Ausgangssituation	2
2.2	Ziele	2
2.3	Risiken	2
2.3.1	Personelle Risiken	2
2.3.2	Know-How Risiken	3
2.3.3	Technische Risiken	3
3	Vorgehensmodell	4
3.1	Agile Softwareentwicklung - Allgemein	4
3.2	Angepasstes Vorgehensmodell	4
3.2.1	Phase Projekt planen	5
3.2.2	Phase Anforderungen definieren	6
3.2.3	Phase Release entwickeln	6
3.2.4	Phasenunabhängige Aktivitäten und Ergebnisse	7
3.3	Release	8
4	Standards und Richtlinien	9
4.1	Programmierrichtlinien	9
4.1.1	Benennungen	9
4.1.2	Schreibweise	9
4.1.3	Visuelle Gestaltung	10
4.1.4	Modularisierung	11
4.1.5	Exceptionhandling	11
5	Projektplanung	12
5.1	Terminplanung	12
5.1.1	Terminplan Vorbereitung	12
5.1.2	Terminplan Situationsanalyse	12
5.1.3	Terminplan Entwicklungszyklen	13
5.1.4	Terminplan Dokumentationen	13
5.1.5	Terminplan Allgemein	14

5.2	Arbeitszeiten und Sitzungen	14
5.2.1	Teammeetings	14
5.2.2	Projektmeetings	15
5.2.3	Projektablage	15
6	Projektorganisation	16
7	Konfigurationsmanagement	17
8	Qualitätssicherung	18
A	Bewertung	19
	Literaturverzeichnis	20

Abbildungsverzeichnis

Tabellenverzeichnis

0.1	Versionen	viii
0.2	Prüfungen	viii
0.3	Freigaben	viii
1.1	Beteiligte	1
3.1	Aktivitäten im Vorgehensmodell	5
3.2	Methoden und Werkzeuge Projektplanung	5
3.3	Methoden und Werkzeuge Anforderungen	6
3.4	Methoden und Werkzeuge Releases	7
3.5	Methoden und Werkzeuge Allgemein	8
5.1	Terminplan Vorbereitung	12
5.2	Terminplan Situationsanalyse	12
5.3	Terminpläne/Entwicklungszyklen	13
5.4	Terminplan Dokumentationen	13
5.5	Terminplan Allgemein	14
5.6	Vorgesehene Projektmeetings	15
6.1	Projektkoordination	16
A.1	Bewertung	19

Änderungskontrolle

Version	Datum	Ausführende Stelle	Bemerkung
0.1	09.05.2003	alle	Start
0.2	16.05.2003	ah, of	weiterarbeiten
0.3	20.05.2003	alle	Überarbeitung
0.4	23.05.2003	alle	Terminplanung
0.5	25.05.2003	svn	Vorgehensmodell
0.6	27.05.2003	alle	Korrekturen
0.7	03.06.2003	ah	Projektorganisation
0.8	10.06.2003	of, svn	Terminplan, Richtlinien

Tabelle 0.1: Versionen

Prüfungen

Version	Datum	Prüfende Stelle/n	Bemerkung
0.8	10.06.2003	of	Gesamtes Dokument geprüft

Tabelle 0.2: Prüfungen

Freigaben

Version	Datum	Freigebende Stelle/n	Bemerkung

Tabelle 0.3: Freigaben

1 Einleitung

1.1 Zweck des Dokumentes

Das Projekthandbuch enthält die organisatorischen und technischen Projektinformationen zur Diplomarbeit `latex2xml`. Es dient uns als Führungsinstrument und wird zu Beginn der Diplomarbeit soweit wie möglich fertiggestellt. Am Anfang jeder Phase ist aber die Richtigkeit dieses Dokuments zu prüfen und wo nötig anzupassen. Unter einer Phase verstehen wir zum Beispiel die Einarbeitung, die Iterationen usw.

1.2 Beteiligte

Name	KZ	Kontakte	Funktion
Stefan von Niederhäusern	svn	svn@nexplore	Student
Andreas Hirter	ah	andreas.hirter@belponline.ch	Student
Olivier Fankauser	of	ofankhauser@highspeed.ch	Student
Pierre Fierz	pf	pierre.fierz@hta-be.bfh.ch	Betreuer
Dr. Jacques Boillat	jb	jacques.boillat@hta-be.bfh.ch	Betreuer
Dr. Federico Flückiger	ff	federico.flueckiger@bluewin.ch	Experte

Tabelle 1.1: Beteiligte

1.3 Projekt Homepage

Die elektronischen Versionen aller Dokumente sind auf unserer HomePage zu finden:
www.latex2xml.org

2 Projektbeschreibung

2.1 Ausgangssituation

Das Textverarbeitungssystem \LaTeX wird von vielen Mitarbeitern der Informatik Abteilung verwendet. \LaTeX ermöglicht eine quasi perfekt gedruckte Ausgabe des Textes. Leider ist eine Publikation auf dem Web nur beschränkt möglich. Bei den vorhandenen Konvertern müssen insbesondere mathematische Formel in Gif-Dateien umgewandelt werden.

Das [W3C-Consortium](#) hat [MathML](#), ein XML-Standard für die Publikation von mathematischen Dokumenten, definiert.

2.2 Ziele

Es soll ein Werkzeug entwickelt werden, das in der Lage ist, \LaTeX -Dokumente mit Benutzer-Unterstützung in XML-Dokumente umzuwandeln. Dabei darf als XML-Spezifikation die sog. [DocBook-DTD](#) mit [MathML-Erweiterungen](#) verwendet werden.

Das Werkzeug soll als Freeware der \LaTeX -Gemeinschaft zur Verfügung gestellt werden. ¹

Die detaillierte Aufgabenstellung ist dem Pflichtenheft zu entnehmen.

2.3 Risiken

2.3.1 Personelle Risiken

- Konflikte innerhalb der Gruppe
Lösungsansatz:
Probleme offen und frühzeitig diskutieren und gemeinsam eine Lösung finden.
- Schwierigkeiten in der Zusammenarbeit Betreuer/Experte und der Gruppe
Lösungsansatz:
Regelmässige Projektmeetings werden abgehalten. Vorgesehen ist ein Intervall von zwei Wochen.

¹Die vollständige Aufgabenstellung kann unter <http://www.hta-be.bfh.ch/wwwinfo/di/03/uebersetzer.shtml> eingesehen werden.

2.3.2 Know-How Risiken

- Das Know-How für die verwendeten Technologien und die Themen ist zu Beginn dieses Projektes nur teilweise oder nicht vorhanden.

Lösungsansatz:

- Genügend Zeit für das Einarbeiten in neue Themen und das Erlernen von neuen Technologien einplanen.
- Eine erste kurze Iteration einplanen, in der ein erster Prototyp erstellt werden kann. Danach muss aber genug Zeit bleiben um die Erkenntnisse aus dieser Phase in eine definitive Version umzusetzen.
- Konsequentes Controlling der Terminplanung, so dass Verzögerungen frühzeitig erkannt werden und Gegenmassnahmen eingeleitet werden können.

2.3.3 Technische Risiken

- Auftreten von technischen Problemen, welche das Lösen der Aufgabe oder einzelner Teilaufgaben mit den gewählten Ansätzen verhindern.

Lösungsansatz:

Ändern der Aufgabe bez. Teilaufgaben in Absprache mit den Betreuern und dem Experten.

3 Vorgehensmodell

3.1 Agile Softwareentwicklung - Allgemein

Bei der agilen Softwareentwicklung geht es darum, Softwareprojekte mit einer schnellen Reaktionsfähigkeit und hoher Wendigkeit zu führen, statt dogmatisch auf die Einhaltung von Prinzipien und Standards abzielen.

Agile Softwareentwicklung ist ein wichtiger Erfolgsfaktor für Projekte in einem hochdynamischen Umfeld, in denen die Anforderungen anfangs unvollständig sind, sich Rahmenbedingungen und Anforderungen stetig ändern, hoher Zeitdruck herrscht und hohe Risiken vorliegen.

Das Schlagwort agile Softwareentwicklung ist entstanden im Umfeld iterativer Entwicklungsprozesse, Extreme Programming (XP) u.ä. und umfaßt zu einem großen Teil bewährte, d.h. altbekannte Ideen und Prinzipien, die jetzt aber ins rechte Licht gerückt und geschickt kombiniert werden

Zitat aus Agile Softwareentwicklung (oose.de) [OOS]

Dabei haben folgende Punkte einen hohen Stellenwert:

- Einbeziehung des Kunden in das Projektteam,
- konsequente Orientierung an den Anforderungen, genauer an ihrem Geschäftswert für das Unternehmen des Kunden,
- kurze Releasezyklen,
- iteratives Vorgehen, das vom Projektteam selbst organisiert wird,
- integriertes Testen.

3.2 Angepasstes Vorgehensmodell

Wir wollen uns für unser Vorgehensmodell an der agilen Softwareentwicklung orientieren. Wir haben uns dabei vom Prozess ActiF von microTOOL inspirieren lassen. [MIC]

Unser Prozess soll durch folgende Eigenschaften charakterisiert sein:

- Leichtgewichtig und iterativ
- Anforderungsgetriebene Planung und Kontrolle
- Refactoring des Software-Entwurfs

- Automatisierte Unit- und Akzeptanztests
- Fortlaufende Integration

Der Prozess soll die in der Tabelle 3.1 aufgeführten Phasen und Aktivitäten beinhalten. Zusätzlich hat das Projekt mehrere Releases, die wiederum in mehreren Iterationen erstellt werden.

Projekt planen
Ziele definieren
Strategie planen
Release planen
Iterationen planen
Anforderungen definieren
Release entwickeln
Iterationen durchführen
Entwerfen (Entwurf)
Implementieren (Implementierung)
Integrieren (Integration)
Abnehmen (Test)
Release einführen

Tabelle 3.1: Aktivitäten im Vorgehensmodell

3.2.1 Phase Projekt planen

In der ersten Phase *Projekt planen* werden die Projektrahmenbedingungen definiert. Die Projektziele werden formuliert und darauf basierend wird die grobe Projektplanung vorgenommen.

Der Projektrahmen soll in einem Projekthandbuch festgelegt werden.

Aktivitäten

- Ziele definieren
- Strategie planen
- Release planen

Aktivität	Methode	Werkzeug
Dokumentation		L ^A T _E X
Projektplanung		MS-Project

Tabelle 3.2: Methoden und Werkzeuge Projektplanung

Ergebnisse

- Projekthandbuch
- Projektplan
- Releaseplanung

3.2.2 Phase Anforderungen definieren

Am Anfang werden die Anforderungen des Projektes definiert. Dabei wird die gewünschte Funktionalität niedergeschrieben. Die Anforderungen werden soweit möglich bei Projektstart definiert. Der Prozess erlaubt es aber jederzeit, neue Anforderungen zu formulieren, Anforderungen zu streichen oder anzupassen.

Basierend auf den Anforderungen kann dann die Release- und Iterationsplanung gemacht werden.

Aktivitäten

- Anforderungen aufnehmen
- Projektumfang abgrenzen

Aktivität	Methode	Werkzeug
Anforderungsanalyse	User-Stories / UseCases	Rational XDE
Dokumentation		L ^A T _E X

Tabelle 3.3: Methoden und Werkzeuge Anforderungen

Ergebnisse

- Pflichtenheft
- L^AT_EX- und DocBook/MathML-Befehlskatalog

3.2.3 Phase Release entwickeln

Diese Phase wird für jedes Release, das während dem Projekt erstellt wird, einmal durchgeführt. Dabei besteht jedes Release aus mehreren Iterationen.

In jeder Iteration wird nur ein Teil der gesamten Anforderungen bearbeitet. Dies bedeutet natürlich auch, dass der Entwurf in jeder Iteration erweitert und angepasst wird. In jeder Iteration wird ein Refactoring gemacht, so dass der Entwurf immer möglichst schlank und leicht verständlich bleibt.

Durch die verschiedenen Iterationen entstehen natürlich auch viele Änderungen an den Dokumenten. Dem soll mit einer Versionierung, unterstützt durch CVS [CVS] begegnet werden.

Der Entwurf soll mit objektorientierten Methoden erfolgen und sich wo immer möglich auf bewährte Patterns stützen.

Wenn man viele Iterationen plant, ist die Integration ein wichtiger Punkt. Bei jeder Integration soll das System automatisch mittels JUnit [JUN] getestet werden. Deshalb werden vor der Implementation die TestCases implementiert.

Aktivitäten

- Entwerfen
- Implementieren
- Integrieren
- Abnehmen

Aktivität	Methode	Werkzeug
Entwurf	UML-Diagramme, Refactoring	Rational XDE
Implementation	Java (OO)	Eclipse
Konfigurationsmanagement	Versionskontrolle	CVS
Qualitätssicherung	Unit-Testing	JUnit
Dokumentation		L ^A T _E X

Tabelle 3.4: Methoden und Werkzeuge Releases

Ergebnisse

- Entwurfsdokument
- Systemdokumentation
- Benutzerdokumentation

3.2.4 Phasenunabhängige Aktivitäten und Ergebnisse

In diesem Abschnitt werden Aktivitäten und Ereignisse aufgelistet, die nicht eindeutig einer Phase zugeordnet werden können, oder mehrere Phasen übergreifen.

Aktivitäten

- Dokumentieren
- Protokollieren

Aktivität	Methode	Werkzeug
Journal	Projekthomepage	http://www.latex2xml.org
Dokumentation		L ^A T _E X

Tabelle 3.5: Methoden und Werkzeuge Allgemein

Ergebnisse

- Journal
- Projektbericht

3.3 Release

Das Projekt wird in zwei Releases aufgeteilt. Das erste Release ist bewusst sehr kurz geplant. Das Ziel dieses Releases besteht darin einen besseren Einblick in das gesamte Themengebiet dieser Diplomarbeit zu erhalten. Dabei soll eine erste Implementation mit eingeschränktem Funktionsumfang erstellt werden. Vor dem zweiten Release werden die Anforderungen im Pflichtenheft anhand der Erkenntnisse aus dem ersten Release überarbeitet.

4 Standards und Richtlinien

[svn] Dokumentationsrichtlinien hinzufügen

4.1 Programmierrichtlinien

Durch das Einhalten der folgenden Programmierrichtlinien erreichen wir einen einheitlichen, gut lesbaren und -erweiterbaren Code.

4.1.1 Benennungen

Programmtexte

Werden in englischer Sprache geschrieben

Kommentare

Werden in deutscher Sprache verfasst

4.1.2 Schreibweise

Wortanfang

- Klassen fangen mit Großbuchstaben an
- Funktionen fangen mit Kleinbuchstaben an
- Variablen fangen mit Kleinbuchstaben an
- Konstantennamen werden in Grossbuchstaben geschrieben

Teilworte

Die Trennung zwischen Teilworten geschieht durch Großbuchstaben
Beispiel:

Listing 4.1: Codebeispiel Teilworte

```
1 setDefaultWindowColor
```

Funktionsnamen

Für Funktionen ohne Rückgabewert (void-Funktionen) wird als Anfang ein Verb im Imperativ verwendet.

Listing 4.2: Funktion ohne Rückgabewert

```
1 public void deleteListItems()
```

Für Funktionen mit booleschem Rückgabewert beginnen mit is.

Listing 4.3: Funktion ohne Rückgabewert

```
1 public boolean isEmpty()
```

Für alle anderen Funktionen sollte ein Name verwendet werden, der im Zusammenhang mit der Aufgabe der Funktion steht.

4.1.3 Visuelle Gestaltung

Listing 4.4: Codebeispiel Visuelle Gestaltung

```
1  if (size < currentSize)
2  {
3      try
4      {
5          // Kommentare oberhalb der zu kommentierenden Zeile
6          size = (long) inStream.available();
7      }
8      catch (IOException e)
9      {
10     }
11 }
12 else if (size == currentSize)
13 {
14     ++size;
15 }
16 else
17 {
18     --size;
19 }
20 }
21 }
```

Zeilenkommentare

Es werden grundsätzlich nur Zeilenkommentare verwendet.
Diese stehen grundsätzlich über der Codezeile auf die sie sich beziehen.

Kommentare zu Funktionen

Diese befinden sich oberhalb der Funktionsdefinition und berücksichtigen die Standards zur Generierung der Java-Doc Dokumente.

Einrücken

Jeder Unterblock wird genau um einen Tabulator eingerückt.

Tabulator

Der Tabulator ist auf 4 Zeichen einzustellen.

Klammern und Blöcke

Die öffnende Klammer steht eine Zeile tiefer als das Schlüsselwort in der gleichen Spalte. Die schliessende Klammer steht auf einer neuen Zeile und ebenfalls in der gleichen Spalte wie das Schlüsselwort.

4.1.4 Modularisierung

Membervariablen

Membervariablen einer Klasse sind niemals public. Der Zugriff auf diese Variablen erfolgt immer über Getter- und Setter-Methoden.

Wahl der Modifier

Die Wahl der Modifier erfolgt möglichst restriktiv.

4.1.5 Exceptionhandling

Alle aufgetretenen Exceptions, welche den korrekten Programmlauf beeinträchtigen, werden einer ApplicationException übergeben.

Es werden nur ApplicationException an den jeweiligen Aufrufer weitergegeben.

5 Projektplanung

5.1 Terminplanung

5.1.1 Terminplan Vorbereitung

Task NR.	Vorgangname	Dauer	Anfangstermin	Endtermin	Ref.
1	Findungsphase	1	25.04.03	25.04.03	-
2	Projekt-Kickoff	1	25.04.03	25.04.03	-
3	Projekthandbuch	19	01.05.03	14.11.03	-

Tabelle 5.1: Terminplan Vorbereitung

5.1.2 Terminplan Situationsanalyse

Task NR.	Vorgangname	Dauer	Anfangstermin	Endtermin	Ref.
4	Einlesen LaTeX	14	28.04.03	11.05.03	-
5	Einlesen XML, Doc-Book, MathML	14	05.05.03	18.05.03	-
6	Einlesen Compilerbau	7	19.05.03	25.05.03	-
7	Mapping-Katalog (LaTeX nach DocBook)	7	26.05.03	01.06.03	6
8	Pflichtenheft		27.05.03	17.06.03	-
9	Pflichtenheft erstellen	14	27.05.03	09.06.03	-
10	Review Pflichtenheft	1	10.06.03	10.06.03	9
11	Korrekturen	6	11.06.03	16.06.03	10
12	Abnahme Pflichtenheft	1	17.06.03	17.06.03	11

Tabelle 5.2: Terminplan Situationsanalyse

5.1.3 Terminplan Entwicklungszyklen

Task NR.	Vorgangname	Dauer	Anfangstermin	Endtermin	Ref.
13	1. Entwicklungszyklus (Prototyp)		18.06.03	05.08.03	8
14	Analyse & Design	7	18.06.03	24.06.03	-
15	Implementation	35	25.06.03	29.07.03	14
16	Test & Korrekturen	6	30.07.03	04.08.03	15
17	Review	1	05.08.03	05.08.03	16
18	Milestone Release 1	0	05.08.03	05.08.03	17
19	2. Entwicklungszyklus		06.08.03	12.11.03	13
20	Analyse & Design	21	06.08.03	26.08.03	-
21	Implmentation	56	27.08.03	21.10.03	20
22	Test & Korrekturen	21	22.10.03	11.11.03	21
23	Review	1	12.11.03	12.11.03	22
24	Milestone Release 2	0	12.11.03	12.11.03	23

Tabelle 5.3: Terminpläne/Entwicklungszyklen

5.1.4 Terminplan Dokumentationen

Task NR.	Vorgangname	Dauer	Anfangstermin	Endtermin	Ref.
27	Projektdokumentation	23	25.04.03	14.12.03	-
28	Benutzerdokumentation	32	13.11.03	14.12.03	-
29	Journal	25	25.04.03	04.01.04	-
30	Abgabe	1	06.01.04	06.01.04	-
31	D Ausstellung	2	23.01.04	24.01.04	-
32	DA Review	2	26.01.04	27.01.04	-

Tabelle 5.4: Terminplan Dokumentationen

5.1.5 Terminplan Allgemein

Task NR.	Vorgangsname	Dauer	Anfangstermin	Endtermin	Ref.
33	Ferien	22	05.07.03	26.07.03	-
34	S. von Niederhäusern	16	05.07.03	20.07.03	-
35	O. Fankhauser	16	05.07.03	20.07.03	-
36	A. Hirter	15	12.07.03	26.07.03	-
37	Diplomprüfungen	14	15.09.03	28.09.03	-
38	S. von Niederhäusern	14	15.09.03	28.09.03	-
39	O. Fankhauser	14	15.09.03	28.09.03	-
40	A. Hirter	14	15.09.03	28.09.03	-

Tabelle 5.5: Terminplan Allgemein

5.2 Arbeitszeiten und Sitzungen

5.2.1 Teammeetings

Die Teammeetings sind fest für die gesamte Dauer der Diplomarbeit geplant. Sofern der jeweilige Tag kein Feiertag ist, treffen sich die Teammitglieder jeden Dienstag von 18.00 Uhr bis 21.30 Uhr und jeden Freitag von 08.30 Uhr bis 17.30 Uhr. Die Themen der Sitzungen werden jeweils an dem vorausgehenden Meeting festgelegt und im Journal auf der Projektablage festgehalten.

5.2.2 Projektmeetings

Projektmeetings sollen ca. alle zwei Wochen stattfinden. Die Meetings werden laufend vereinbart. Sind Dokumente Bestandteil der Projektmeetings, müssen diese vorher auf der Homepage veröffentlicht sein.

Von jedem Projektmeeting ist ein Protokoll zu führen. Die Protokolle werden nicht jedem Teilnehmer einzeln zugesandt, sondern bis Ende der entsprechenden Woche auf der Projekttafel unter der Rubrik Protokolle veröffentlicht. Zu jedem Meilenstein muss ein Projektmeeting mit allen Projektbeteiligten einberufen werden.

Vorgesehen sind die folgenden Termine:

Datum	Bemerkung
24.06.2003	Abnahme Pflichtenheft
08.07.2003	Ferien
22.07.2003	Ferien
05.08.2003	Ende 1.Entwicklungszyklus
19.08.2003	
02.09.2003	
16.09.2003	
30.09.2003	
14.10.2003	
28.10.2003	
11.11.2003	Ende 2.Entwicklungszyklus
25.11.2003	
09.12.2003	Stand Benutzer- / Projektdokumentation
23.12.2003	
06.01.2004	Abgabe

Tabelle 5.6: Vorgesehene Projektmeetings

5.2.3 Projektablaage

Die Projektablaage befindet sich auf dem Internet unter der folgenden Adresse.

<http://www.latex2xml.org> Unter dieser Adresse sind nur die Dokumente veröffentlicht, welchen als Version oder als Final vorliegen. Die Draft-Dokumente sind nur für die Diplomanden zugänglich und werden mittels CVS verwaltet.

6 Projektorganisation

Die einzelnen Teilgebiete dieser Diplomarbeiten sollen nicht fix verteilt werden. Alle Teammitglieder sollen an der ganzen Diplomarbeit beteiligt sein. Dennoch sollen die Kräfte des Team effizient eingesetzt werden. Innerhalb der einzelnen Themengebiete können die Arbeiten aufgeteilt werden.

Um die Koordination der Teilaufgaben sicherstellen zu können, haben wir folgende Organisation gewählt.

Themengebiet	Koordinator	Bemerkung
Kommunikation	A.Hirter	Team / Experte / Betreuer
Anforderungen / Release	S.von Niederhäusern	
Dokumentation	O.Fankhauser	inkl. Webauftritt
Termin Controlling	A.Hirter	
Qualitäts Sicherung	O.Fankhauser	

Tabelle 6.1: Projektkoordination

7 Konfigurationsmanagement

8 Qualitätssicherung

Im Rahmen der Seminararbeit haben wir das Thema JUnit gewählt, mit welchem wir auch die Qualität dieser Diplomarbeit sichern wollen. Nach Fertigstellung dieser Arbeit wird das Kapitel noch nachgereicht.

A Bewertung

In der folgenden Tabelle sind die Beurteilungskriterien und deren Gewichtung dargestellt.

Diese Gewichtung kann in Absprache mit dem Experten und den Betreuern vor der Unterzeichnung des Pflichtenheftes angepasst werden.

Arbeitsschritt	Gewichtung	Relativ
Vorbereitungsphase		
Aufbau und Vollständigkeit des Pflichtenheftes	3	11%
Durchführung		
Arbeits- und Zeiteinteilung	3	11%
Kreativität, Initiative und Selbständigkeit	1	4%
Wahl und Anwendung der (Entwurfs-)Methodik	4	15%
Implementation, Robustheit und Programmierziel	2	7%
Systemtest (Verfahren, Durchführung, Bericht)	2	7%
Kommunikation mit dem Experten und Betreuer	1	4%
Ergebnis		
Uebereinstimmung Produkt/Pflichtenheft	5	18%
Allgemeiner Eindruck aus der Besichtigung	1	4%
Projektbericht		
Inhalt korrekt, vollständig und verständlich	3	11%
Sprache, Stil und Übersichtlichkeit	1	4%
Klare aussagekräftige Zusammenfassung	1	4%

Tabelle A.1: Bewertung

Literaturverzeichnis

[CVS] Concurrent Versions System. <http://www.cvshome.org>.

[JUN] JUnit - Testing Resources for Extreme Programming. <http://www.junit.org>.

[MIC] MICROTOOL. actif - Ein automatisierter Prozess für die agile Softwareentwicklung.
http://download.microtool.de/mT/pdf/in-Step/49/is_actif_extreme.pdf.

[OOS] OOSE.DE. Agile Softwareentwicklung. <http://www.oose.de/agilitaet/index.htm>.