

Software–Projekt 2 2013

VAK 03-BA-901.02

Projektplan

Sylvia Kamche Tague	clar@tzi.de	2476985
Dario Treffenfeld-Mäder	dtm@tzi.de	2598686
Nils Sören Oja	nso@tzi.de	2725302
Sandor Herms	sanherms@tzi.de	2931655
Olga Miloevich	halfelv@tzi.de	2586817
Jannes Uken	ukenj@tzi.de	2787018

Inhaltsverzeichnis

1	Einleitung	4
1.1	Projektübersicht	4
1.1.1	Ziele	4
1.1.2	Hauptarbeitsaktivitäten und –produkte	4
1.1.3	Haupt–Meilensteine und grober Zeitplan	5
1.1.4	Benötigte Ressourcen	5
1.1.5	Budget	5
1.1.6	Kontaktdaten des Kunden	6
1.1.7	Mitarbeiter	8
1.2	Auszuliefernde Produkte	9
1.3	Evolution des Plans	9
1.4	Referenzen	9
1.5	Definitionen und Akronyme	10
2	Projektorganisation	11
2.1	Prozessmodell	11
2.2	Organisationsstruktur	11
2.3	Organisationsgrenzen und –schnittstellen	12
2.4	Verantwortlichkeiten	12
3	Managementprozess und -prioritäten	13
4	Risikomanagement	15
4.1	Projektüberwachung	17
4.2	Mitarbeiter	17
5	Technische Prozesse	18
5.1	Methoden, Werkzeuge und Techniken	18
5.1.1	Entwicklungsplattform	18
5.1.2	Entwicklungsmethode	18
5.1.3	Programmiersprache und Bibliotheken	18
5.2	Dokumentationsplan	18
5.2.1	Codingstyle	19
5.2.2	Kommentarsprache	19
5.2.3	JavaDoc	19
5.2.4	Begleitende Dokumentation	19
5.3	Unterstützende Projektfunktionen	19
6	Arbeitspakete, Zeitplan und Budget	19
6.1	Arbeitspakete	19
6.1.1	Projektplan	20
6.1.2	Anforderungsspezifikation	20

6.1.3	Architekturbeschreibung, Schnittstellenbeschreibung, Testplan und Blackbox-Tests	22
6.1.4	Implementierung	23
6.2	Zeitplan und Abhängigkeiten	23
6.3	Ressourcenanforderung	25
6.4	Pläne für die Konvertierung von Daten	26
6.5	Managementpläne für Unterauftragsnehmer	26
6.6	Ausbildungspläne	26
6.7	Raumpläne	26
6.8	Installationspläne	27
6.9	Pläne für die Übergabe des Systems	27

Version und Änderungsgeschichte

Version	Datum	Änderungen
1.0	20.10.2013	Endabgabe des Projektplans.

1 Einleitung

Bearbeitet von: Sylvia Kamche Tague und Dario Treffenfeld - Mäder

1.1 Projektübersicht

1.1.1 Ziele

Dieses Semester sollen wir ein Bibliothekssystem für die Oberschule Rockwinkel erstellen. Das System soll aus einer Webseite und einer Android App oder einem anderen Zugang, der für Geräte mit kleinem Display optimiert ist, bestehen. In der Mobilen App sollen Bücher vorgemerkt werden können. Die Bibliothekaren App ist ein Verwaltungstool.

1.1.2 Hauptarbeitsaktivitäten und –produkte

Aktivität	Produkt
Projektplanung, Zeitplanung	Projektplan
Prototyping, rudimentäre Fassung der GUI	Prototyp
Auswertung der Anforderungen und des Kundengesprächs, Perspektive, detaillierte Ausarbeitung	Anforderungsspezifikation
Globale Analyse, Architektursichten	Architekturbeschreibung
Abnahme- und Testendkriterien, Testfälle (u.a. White-Box-Test, Black-Box-Test)	Testplan
Erstellung der GUI, Datenbank, Serververbindung, Programmlogik	Software
Feinschliff der Software	Abgabefertige Software

1.1.3 Haupt-Meilensteine und grober Zeitplan

Produkt	Datum
Initialer Projektplan	20.10.2013
aktuelles Berichtsheft	23.10.2013
Kundengespräch	23.10.2013
Besuch der SUUB	31.10.2013
Prototyp Vorstellung	6-7.11.2013
Anforderungsspezifikation und Angebot	17.11.2013
aktuelles Berichtsheft	20.11.2013
Architekturbeschreibung, Testplan und Schnittstellentests (JUnit)	22.12.2013
aktuelles Berichtsheft	25.12.2013
Präsentation der ersten lauffähigen Version	27-31.01.2014
Akzeptanztest	10-15.02.2014
Vollständige Abgabe	23.02.2014
Abschlusspräsentation	25-26-02.2014
aktuelles Berichtsheft	26.02.2014
unterschiedene Erklärung zum Berichtsheft	10.03.2014

An verschiedenen Zeitpunkten des Projekts werden Meilensteine gesetzt. Und Zwar in dem Projektplan, Anforderungsspezifikation beim Review der Soll-Analyse, Architektur beim Review der Anforderungsspezifikation, Implementierung beim Akzeptanztest. Ein Meilenstein ist ein Zeitpunkt, zu dem ein prüfbares Ergebnis vorliegen muss. Es ist also erreicht, wenn was geleistet soll, rechtzeitig und mit guten Qualität zum Ziel führt.

1.1.4 Benötigte Ressourcen

- **Menschliche Ressourcen** Wir sind eine Gruppe von sechs Studenten und werden alle Entwickler dieses Projekt sein.
- **Hardware** Jeder besitzt ein eigenen Laptop. Außerdem hat jeder Mitarbeiter Zugang zu den Rechnern in der Ebene 0 im MZH.
- **Räume** Jedes Treffen wird in der Cafete in der Ebene 1 im MZH stattfinden. Falls für ein Treffen der Raum besetzt ist, werden wir in der Ebene 0 im Macraum arbeiten. Zudem haben wir auch die Möglichkeit eine Lerninsel im Gebäude GW2 zu nutzen. In dem Raum befindet sich ein Plasmabildschirm sowie eine Tafel.

1.1.5 Budget

Für dieses Projekt steht uns kein Geld zu Verfügung. Deshalb benutzen wir nur kostenlose Software und nehmen keine externe Hilfe in Anspruch. Wir können also kein Budget in Euro definieren. Was wir investieren werden sind minimal 2 Stunden pro Tag, um das Projekt zu bearbeiten.

1.1.6 Kontaktdaten des Kunden

Für dieses Projekt haben wir ein echter Kunde. Seine Daten sind folgendes:

Herr Meyhöfer

Öffentliche Schulbibliothek Rockwinkel in der Oberschule Rockwinkel

Uppe Angst 31, 28355 Bremen

Telefon 0421/361-3360

1.1.7 Mitarbeiter



Nils Sören Oja

nso@tzi.de



Sylvia Kamche Tague

clara@tzi.de



Sandor Herms

sanherms@@tzi.de



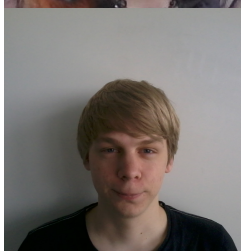
Jannes Uken

ukenj@tzi.de



Olga Miloevich

halfelv@tzi.de



Dario Treffenfeld - Mäder

dtm@tzi.de

1.2 Auszuliefernde Produkte

Das Produkt ist ein Bibliothekssystem. Dieses besteht aus einer Website und einer Mobil-App. Die App ist für Leser gedacht und die Website für Leser und Bibliothekare. Letztere können Bücher und Benutzer verwalten. Leser (Bibliotheksnutzer) können Bücher einsehen und sofern sie registrierte Nutzer sind, diese auch genau einmal bewerten. Die Bücher sind mindestens mit Titel, Autor, ISBN und Bewertung gelistet. Ein Administrator kann auf der Website Medien, Nutzer und Bibliothekare registrieren und löschen. Ein Bibliothekar kann Leser registrieren und löschen und Bücher hinzufügen und löschen. Eine für dieses Bibliothekssystem passende Serversoftware wird außerdem ausgeliefert.

1.3 Evolution des Plans

Folgender Abschnitt würde vom Projektplan der Gruppe IrgendwieCool (Vorlesung Software Projekt 2, SoSe2013, S. 9) übernommen

Der Projektplan wird hinsichtlich der Zeitplanung wöchentlich überprüft und bei starken Abweichungen vom jeweiligen Phasenleiter angepasst. Er kann auch in den Phasen Architektur, Implementierung und Test, wenn es nötig ist, erweitert werden. Bei akuten Problemen, wie z.B. Krankheitsfall oder Personalausfall, wird eine Änderung des Zeitplans und der Arbeitspakete direkt und außerplanmäßig überdacht, um die Anforderungen zu erfüllen werden reduziert. Jeder Mitarbeiter kann Vorschläge bezüglich der Zeitplanung an den Phasenleiter weiterleiten, der diese in seine Überlegungen mit einbezieht. Den Phasenleiter hat immer das letzte Wort für eine eventuell Änderung. Der Phasenleiter akzeptiert die Vorlage des Projektkoordinators oder lässt ihn den Plan nach wünschen überarbeiten. Wenn es wirklich nötig ist können wir auch die Software aktualisieren. Wir konzentrieren uns als erstes auf dass erfüllen der Mindestanforderungen und erst danach auf das Implementieren von Zusatzfunktionen.

1.4 Referenzen

Bearbeitet von: Olga Miloevich

- IT-Fachkunde, ISBN: 978-3-8085-3653-7, Kapitel 6.2 Projektmanagement
- Koschke, Rainer: Projektplan-Vorlage SWP1, SS 2013
- Koschke, Rainer: Planung (Vorlesungsskript), SS 2013
- Gruppe "irgendwiecool", Projektplan, SWP 2, WS 2012/13
- Gruppe "ICC", Projektplan, SWP1, SS2013
- Gruppe "Five and a half man", Projektplan, SWP 1, SS2013

1.5 Definitionen und Akronyme

Bearbeitet von Sylvia Kamche Togue, Olga Miloevich

- Android: Freies Betriebssystem für mobile Geräte.
- Application: ggf Programm für die Nutzung auf mobilen Geräten
- Client: bezeichnet ein Computerprogramm, das auf dem Rechner des Nutzers ausgeführt wird. Sie sind in aller Regel in eine Client-/Serverkommunikation eingebunden. (*Wikipedia, die freie Enzyklopädie, Aufruf von Client-Seite 18.10.2013 14:08*)
- Eclipse: Eine IDE, ggf hauptsächlich für Java-Programmierung benutzt.
- Gantt-Diagramm: Nach Henry L. Gantt benanntes Diagramm für Zeit- und Ressourcenmanagement. Stellt die zeitliche Abfolge von Arbeitsschritten dar.
- Git: Versionsverwaltung. Speichert die Dateien und sichert den Zugriff auf vergangene Versionen.
- Java: Objektorientierte Programmiersprache.
- JavaDoc: Dokumentationswerkzeug, das aus Quellcodekommentaren HTML-Dokumentation erstellt.
- HTML: Die Hypertext Markup Language. Eine textbasierte Auszeichnungssprache zur Strukturierung von Inhalten wie Texten, Bildern und Hyperlinks in Dokumenten. HTML-Dokumente sind die Grundlage des World Wide Web und werden von einem Webbrowser dargestellt. (*Wikipedia, die freie Enzyklopädie, Aufruf von HTML-Seite 18.10.2013 13:46*)
- IDE: Eine integrierte Entwicklungsumgebung (Abkürzung IDE, von engl. integrated development environment) ist eine Sammlung von Anwendungsprogrammen, mit denen die Aufgaben der Softwareentwicklung (SWE) möglichst ohne Medienbrüche bearbeitet werden können. (*Wikipedia, die freie Enzyklopädie, Aufruf von IDE-Seite 18.10.2013 13:49*)
- LaTeX: Softwarepaket, das die Benutzung des Textsatzsystems TeX mit Hilfe von Makros vereinfacht. (*Wikipedia, die freie Enzyklopädie, Aufruf von LaTeX-Seite 18.10.2013 13:56*)
- Makro: Ein Makro ist in der Softwareentwicklung eine unter einer bestimmten Bezeichnung (Makroname) zusammengefasste Folge von Anweisungen oder Deklarationen, um diese (anstelle der Einzelanweisungen, i. d. R. an mehreren Stellen im Programm) mit nur einem einfachen Aufruf ausführen zu können. Alle Anweisungen des Makros werden automatisch an der Programmstelle ausgeführt, an denen das Makro codiert wurde. (*Wikipedia, die freie Enzyklopädie, Aufruf von Makro-Seite 18.10.2013 13:58*)
- PDF: ein Dateiformat für Dokumente.
- Plug-in: Software, welche in andere Programme zur Erweiterung der Funktionalität

eingebunden werden kann.

- Server: Rechner, der Anfragen von Client-Rechnern entgegen nimmt und verarbeitet.
- SQL: Structured Query Language, Datenbanksprache für die Bearbeitung und Abrufen von Daten innerhalb einer Datenbank

2 Projektorganisation

Bearbeitet von: Sandor Herms

2.1 Prozessmodell

Folgender Abschnitt würde vom Projektplan der Gruppe IrgendwieCool (Vorlesung Software Projekt 2, SoSe2013, S. 11) übernommen Wir arbeiten nach dem Wasserfallmodell mit den Phasen Projektplan, Anforderungen, (Architektur-)Entwurf, Schreiben erster Tests und Festlegung der Schnittstellen, Implementierung und Dokumentation sowie abschließendem Testen. Die Punkte werden strukturiert nacheinander abgearbeitet, wobei das Testen auch während der Implementierung noch eine Rolle spielt. Wenn es während einer Phase Probleme gibt, haben wir jederzeit die Möglichkeit, zum vorherigen Punkt zurückzuspringen.

Es muss ein Plan des weiteren Vorgehens erstellt werden, in welchem unter anderem Aufgaben und Zeiteinteilungen vorzunehmen sind.

Darauf folgt die Analyse des Projekts. Dabei ist es wichtig zu überprüfen, was mit den uns zur Verfügung stehenden Ressourcen umzusetzen ist.

Vor der Implementierung des Codes schreiben wir bereits sämtliche Black Box-Tests. So können wir beim Schreiben des Codes jederzeit überprüfen, ob er valide ist.

Wir müssen stets auf die begrenzten Ressourcen der mobilen Geräte achten. Neben weniger Speicher ist die Rechenleistung auch um einiges geringer, als bei einem heutigen handelsüblichen PC. Des weiteren muss auf die Stabilität der Software geachtet werden. Nach der Implementierung des Codes wird die Software vor der Endabgabe nochmals ausführlich getestet, u.a. durch White Box-Tests. Nur so kann gewährleistet werden, dass ein funktionierendes und hochwertiges Produkt erstellt wurde. Für das Handbuch und allgemein alle Dokumente gelten ähnliche Maßstäbe; die Grammatik muss überprüft und eventuelle Rechtschreibfehler beseitigt werden.

2.2 Organisationsstruktur

Unsere Dateien werden im uns zur Verfügung gestellten Git-Repository gespeichert.

Die Kommunikation findet über E-Mail statt. Da eines unserer Mitglieder am Wochenende kein Internetzugang hat, muss notfalls Kontakt per Handy aufgenommen werden.

Zudem werden wir uns wöchentlich in der Cafete im MZH treffen. Der genaue Tag dafür steht noch nicht fest.

Für das Projekt haben wir einen Projektleiter, welcher für die allgemeine Leitung und Organisation des Projekts zuständig ist und einen Kontroller, welcher die Arbeit des Projektleiters überprüft bestimmt. Für die einzelnen Arbeitspakete haben wir jeweils einen Phasenleiter bestimmt. Dieser ist für die vollständige und rechtzeitige Bearbeitung sowie für die Qualitätssicherung der Abgaben zuständig. Näheres dazu ist im Abschnitt Managementprozess zu finden.

2.3 Organisationsgrenzen und –schnittstellen

Unsere Organisationsgrenze wird zu unserer Auftragsgeberin gezogen, die unsere übergeordnete Organisation darstellt.

Über folgende Daten kann mit ihr Kontakt aufgenommen werden.:

Hui Shi

Kontakt Technologiezentrum Informatik, Universität Bremen

Briefe Postfach 330 440, D-28344 Bremen, Germany

Büro Cartesium 1.053

Telefon +49 (421) 218-64260

Telefax +49 (421) 218-9864260

Email shi@tzi.de

2.4 Verantwortlichkeiten

Mitarbeiter	Rolle
Sandor Herms	Projektleiter Phasenleiter Projektplan
Sylvia Kamche Tague	Controller Phasenleiterin Architekturplan
Olga Miloevich	Phasenleiterin Anforderungsspezifikation
Dario Treffenfeld-Mäder	Phasenleiter Implementierung
Jannes Uken	Phasenleiter Testplan
Nils Sören Oja	Phasenleiter Präsentation

3 Managementprozess und -prioritäten

Bearbeitet von: Olga Miloevich, Sandor Herms

Folgender Abschnitt würde von Projektplan von der Gruppe Five and a half men (Vorlesung Software Projekt 1, SoSe2013, S. 7 übernommen

Für die Organisation unseres Projektes haben wir einen Projektleiter gewählt. Außerdem, haben wir auch die Phasenleiter ausgesucht. Diese sind für die rechtzeitigen und vollständigen Abgaben der Phasen verantwortlich. Die Kernidee ist, daß damit die gesamte Aufgabe von dem Projektleiter erleichtert wird.

Wir haben zudem die Gruppe in einzelne Arbeitsgruppen eingeteilt. Dafür haben wir nach einem schnellen dynamischen System aus der Wirtschaft gesucht und haben das sogenannte Mehrliniensystem ausgewählt. Dieses Mehrliniensystem ist im Gegensatz zur Matrixorganisation oder dem Einzelliniensystem am besten für unsere Umstände und Zwecke geeignet. Das System passt am besten für kleine Gruppen, wie unsere. Es lassen sich mehrere Personen zu einer Gruppe zuordnen.

Der Projektmanager überwacht den Ablauf des gesamten Projektes und kontrolliert die Arbeitsgruppen. Die Arbeitsgruppen kommunizieren mit den jeweiligen Mitgliedern ihrer Arbeitsgruppe sowie mit allen Gruppenleitern. Bei Bedarf kommunizieren die Arbeiter einer Arbeitsgruppe auch mit Arbeitern anderer Arbeitsgruppen.

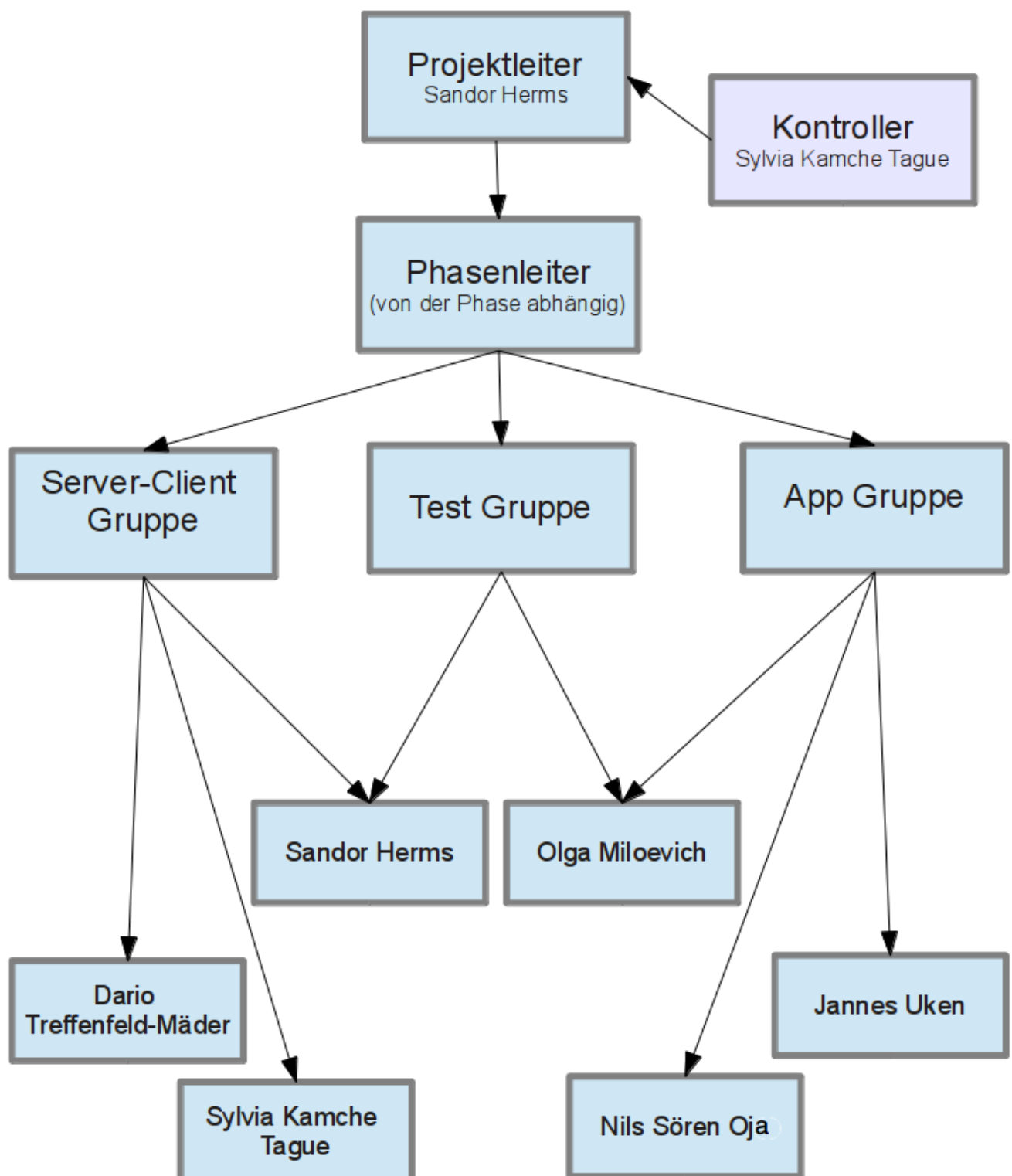
Die Kommunikationen innerhalb der gesamten Projektgruppen wird durch regelmäßige Meetings gewährleistet. Bei Konflikten oder Unstimmigkeiten ist der Projektleiter ein Ansprechpartner und auch Entscheidungsträger.

Unsere Prioritäten liegen bei diesem Projekt hauptsächlich bei der Erfüllung der Mindestanforderungen in dem gegebenen Zeitraum. Dabei haben wir auch einen Puffer eingeplant. Werden die Mindestanforderungen frühzeitig erfüllt, dann wird dieser Puffer dazu benutzt, einige Zusatzfunktionen in unsere Software zu integrieren.

Unser Budget besteht aus Personen und Personenstunden. Die Planung dieses Budgets übernimmt der Phasenleiter.

Es folgt unser Mehrliniendiagramm.

Quellen: Projektplan von der Gruppe "Five and a half men" (Vorlesung Software Projekt 1 2013); Projektplan von der Gruppe "irgendwiecool" (Vorlesung Software Projekt 2 2013); Projektplan von der Gruppe "ICC" (Vorlesung Software Projekt 1 2013)



4 Risikomanagement

Bearbeitet von: Olga Miloevich

Folgender Abschnitt würde aus dem Projektplan von der Gruppe Five and a half men (Vorlesung Software Projekt 1, Sose 2013, S. 12-14) übernommen

Um mögliche Risiken zu identifizieren und zu beschreiben, haben wir hier eine sogenannte FMEA verwendet. Dies ist als “Fehlermöglichkeits- und Einflußanalyse” bekannt und wird in ähnlicher Form auch bei Siemens oder Toyota durchgeführt. Hier geht es um die konkreten Risiken, welche die Entwicklergruppe zu berücksichtigen hat. Die FMEA-Tabelle lässt die eintretende Risiken in Zahlen analysieren. Sie verwendet dazu die Eintrittswahrscheinlichkeit, die Schwere des Fehlers, so wie die Wahrscheinlichkeit, den Fehler frühzeitig zu entdecken.

Die Gewichtung sagt, wie schwer das Risiko ist und die Entdeckung wie offensichtlich das Risiko zu entdecken ist. Die Tabelle zeigt außerdem welche Person für welchen Bereich die Verantwortung übernommen hat. Zuletzt sind auch mögliche Maßnahmen zur Risikomilderung genannt.

Folgende Informationen sind zu beachten:

- Die Zahlen in den Kategorie-Spalten “Auftreten”, “Bedeutung” und “Entdeckung” repräsentieren eine Eingliederung des Risikos entsprechend der genannten Kategorie auf einer Skala von 1 bis 10, wobei eine niedrigere Zahl positiv, eine höhere Zahl negativ zu verstehen ist.
- Die RPZ (die *Risiko-Prioritätszahl*) ist das Produkt aus den drei vorhergehenden Kategorien Auftreten, Bedeutung und Entdeckung.
Eine RPZ unter 100 ist als unkritisch einzustufen, eine RPZ über 700 als sehr kritisch. Hier ist es sehr wahrscheinlich, dass bei Eintritt dieses Fehlers oder Risikos das Projekt scheitern wird.

Systeme, Merkmale	potentielle Fehler	potentielle Folgen des Fehlers	potentielle Fehlerursache	Derzeitiger Zustand				Empfohlene Abstellmaßnahmen	Verantwortlichkeit
				Auftreten	Bedeutung	Entdeckung	RPZ		
Rechner-Arbeitsplätze und Laptops	ungeschützt	nicht authentifizierter Zugriff auf sensible Daten	versäumt den Bildschirm zu sperren	2	5	5	50	NFC am Körper	Jeder User selbst
Rechner-Arbeitsplätze und Laptops	Diebstahl	Entwendung von Daten	Laptop nicht physisch fest verbunden mit z.B. Tisch	2	5	10	100	Kensington-Schloß	der User selbst
Lasten- und Pflichtenheft	Verlust	keine Chance auf Fertigstellung des Auftrages	Datenverlust	2	6	10	200	Sicherungskopien	Projektleiter
Kundengespräch	findet nicht statt	Keine genauen Informationen zur Vollaendung des Projektes bzw. keine Hintergrundinformationen zu gegebenen Sachverhalten	Termin versäumt	1	10	10	100	Terminplan	Projektleiter
Lasten- und Pflichtenheft	Wünsche des Kunden nicht verstanden	unzufriedener Kunde	Anforderungen falsch verstanden	5	9	8	360	regelmäßige Kommunikation mit dem Kunden	Projekt- und/oder Teamleiter
IST-Analyse	Fehler beim Analysieren des aktuellen Vorgehens des Kunden		inkompetenter Mitarbeiter fehlerhafte bzw. ungenaue Beschreibung des Kunden	5	4	6	120	Vier-Augen Prinzip	Projektleiter
Prototyp	nicht akzeptiert		nicht gemäß den Wünschen des Kunden	7	3	3	63	Prototyp möglichst nahe am gewünschten Produkt entwickeln	Projektleiter
SOLL-Analyse	Problemstellungen falsch erfasst und/oder interpretiert	falsche Modelle für weiteres Vorgehen	fehlerhafte Problembeschreibung fehlerhafte Dokumentation	4	8	8	256	Vier-Augen Prinzip	Projektleiter
Lastenheft	nicht alle Kundenwünsche berücksichtigt	kein Vertrag verzögerter Vertrag	fehlerhafte Analysen	3	9	6	162	regelmäßige Kommunikation mit dem Kunden	Projektleiter
Lastenheft	ist inkonsistent	falsche Modellbildung	fehlerhafte Analysen	5	7	9	315	Prüfung durch unabhängigen Fachmann	Projektleiter
Pflichtenheft	unvollständig	fehlerhafter Prototyp	fehlerhafte Anforderungsanalyse	8	4	6	192	detaillierte Kommunikation mit dem Kunden	Projektleiter und Entwickler
Pflichtenheft	inkonsistent	fehlerhafter Prototyp		4	4	7	112	Prüfung durch unabhängigen Fachmann	Projektleiter und Entwickler
UML-Diagramm	inkonsistent	falsche Implementierung	fehlerhafte SOLL-Analyse	4	8	9	288	Vier-Augen Prinzip	Entwickler

Systeme, Merkmale	potentielle Fehler	potentielle Fol- gen des Fehlers	potentielle Fehlerursache	Derzeitiger Zustand				Empfohlene Abstellmaß- nahmen	Verantwort- lichkeit
				Auftreten	Bedeutung	Entdeckung	RPZ		
Implementier- ung	nicht gemäß der Anforder- ungen des Lastenheftes	Kunde lehnt das Produkt ab	Fehler im UML Dia- gramm	4	9	7	252	Fähigkeiten der einzelnen Mitarbeiter kennen	Entwickler
Implementier- ung	Implementier- ung nicht geforderter bzw. nicht gewünschter Inhalte		ungenau- es Abhandeln des Lasten- und Pflich- tenheftes	2	6	6	72	erneute Absprache mit Kun- den anhand Prototyp	Entwickler
Testfälle	falsche Test- modelle	Testen des Produktes wird verzögert	Fehlerhaftes Modell inkompetente Mitarbeiter	3	4	9	108	sich über die Fähigkeiten der einzelnen Mitarbeiter bewusst sein	Entwickler
Abnahme	nicht akzep- tiert	Kunde lehnt weitere Zu- sammenarbeit ab	dem Kunden gefällt das Ergebnis nicht	2	10	10	200		Projektleiter
Zeitmanage- ment	stimmt nicht	ein oder meh- rere Arbeits- pakete zögern sich in ihrer Fertigstellung hinaus	falsch ein- geschätzt	6	9	7	378	höhere Puf- ferzeiten einbauen und par- allelisierte APs	Projektleiter
Krankheits- fälle	Mitarbeiter erkranken	zu erledigen- de Arbeit kann nicht fertiggestellt werden	Grippe o.ä.	5	2	1	10	vorausschau- ende Arbeits- verteilung bzw. Umver- teilung	Projekt- und/oder Teamleiter

4.1 Projektüberwachung

Bearbeitet von: Sylvia Kamche Tague

Für die Fortschrittsüberwachung des Projektes werden wir uns mindestens einmal pro Woche nach Absprache treffen. Der Phasenleiter wird das Treffen moderiert. Er prüft auch, ob der Zeitplan angehalten ist oder nicht. Jeder wird sagen, was er vom letzten Treffen bis diesem gemacht hat, und was er für Probleme getroffen hat. Also können die anderen auch eine Lösung dafür geben, falls sie eine Ahnung haben. Dann soll man auch mitteilen, was er als weiteres machen wird. Es wird jedes Treffen ein Protokollant gewählt. Der schreibt alles was noch zu machen ist und schickt später eine Mail an alle. Wenn eine Thematik nicht abschließend behandelt werden kann, wird diese auf einen späteren Zeitpunkt vertagt (nächstes Meeting oder zu einem anderen geeigneten Zeitpunkt). Der Phasenleiter ist durch diese Meetings besser dazu in der Lage, die Aufgaben zu delegieren und mögliche Behinderungen zu beseitigen. Sollten während der Arbeit Probleme auftreten, werden diese direkt dem Phasenleiter gemeldet, sodass er in der Lage ist frühzeitig entsprechende Maßnahmen zu ergreifen.

4.2 Mitarbeiter

Bearbeitet von: Sylvia Kamche Tague

Die technische Kompetenzen von Projektmitgliedern sind folgende:

Name	Kompetenzen
Jannes Uken	Java, Latex, SQL, HTML, C, Android, PHP
Sylvia Kamche Tague	Latex, Java, SQL, Android, C++
Dario Treffenfeld-Mäder	Latex, Java, MySQL, Webprogrammierung, Android, C
Sandor Herms	Java, Latex, SQL, HTML, C
Olga Miloevich	Java, Latex, SQL, Android, C++
Nils Sören Oja	Java, Latex, Haskell, HTML, PHP, C

5 Technische Prozesse

Bearbeitet von: Sylvia Kamche Tague

5.1 Methoden, Werkzeuge und Techniken

5.1.1 Entwicklungsplattform

Unser Produkt wird mit Hilfe der IDE Eclipse (letzte oder vorletzte Version) entwickelt. In Eclipse wird auch Maven integriert. Wir werden für dieses Projekt GlassFish 3.1 als Application-Server benutzen.

5.1.2 Entwicklungsmethode

Zuerst entwickeln wir die Schnittstellen und implementieren die dazugehörigen Blackboxtests, erst danach realisieren wir die eigentliche Funktionalität. Anschließend testen wir den fertigen Code mit Whiteboxtests.

5.1.3 Programmiersprache und Bibliotheken

Die genutzten Programmiersprachen werden Java und Javascript sein, zudem benutzen wir HTML, CSS und XML und sowie SQL-Datenbank.

5.2 Dokumentationsplan

Klassen, Methoden und Variablen werden in der Phase der Implementierung dokumentiert.

5.2.1 Codingstyle

Wir verwenden die Formatierungsfunktion von Eclipse um einheitlich strukturierten Code zu schaffen. Klassen-, Methoden- und Variablen sollten einen aussagekräftigen, englischen Namen erhalten, wobei gegebenenfalls ein Unterstrich als Trennzeichen genommen wird.

5.2.2 Kommentarsprache

Die Kommentare werden in Englisch geschrieben, um Probleme mit Sonderzeichen zu vermeiden und das Verständnis des Sourcecodes bei der Wartung auch Nicht-Deutschsprachigen zu erleichtern bzw. zu ermöglichen.

5.2.3 JavaDoc

Alle JavaDoc Kommentaren werden auf Englisch verfasst, und dienen der Übersicht und dem schnelleren Erfassen der Zusammenhänge im Projekt.

5.2.4 Begleitende Dokumentation

Begleitend wird ein Benutzerhandbuch angefertigt, das Installation und Bedienung aller Komponenten des Produkts erklärt.

5.3 Unterstützende Projektfunktionen

Die Dateien werden im Git-Repository gespeichert. Für die Qualitätssicherung ist der jeweilige Phasenleiter zuständig. Dieser wird jede Abgabe auf Vollständigkeit und Inhalt prüfen und gegebenenfalls überarbeiten.

6 Arbeitspakete, Zeitplan und Budget

Bearbeitet von: Dario Treffenfeld - Mäder, Jannes Uken und Nils Sören Oja

6.1 Arbeitspakete

Hier listen wir die Arbeitspakete auf. Wir planen die Anforderungsspezifikation detailliert und die anderen Phasen nur grob.

6.1.1 Projektplan

...

6.1.2 Anforderungsspezifikation

AP Titel: 1. Einführung	AP Nummer: 1
Dauer: 21.10.13 - 22.10.13	Aufwand: 8 Std.
AP-Teilnehmer: Sandor Herms, Olga Miloevich	
Beschreibung: Bearbeiten des Abschnittes Einleitung	
Voraussetzung: keine	
Ziele: Die Punkte Zweck, Rahmen, Definitionen, Akronyme und Abkürzungen sowie Referenzen und die Übersicht über das Dokument sind erstellt.	

AP Titel: Kundengespräch	AP Nummer: 2
Dauer: 23.10.13	Aufwand: 4 Std.
AP-Teilnehmer: Sandor Herms, Olga Miloevich	
Beschreibung: Gespräch mit dem Kunden um dessen Wünsche und Erwartungen an das Produkt festzustellen.	
Voraussetzung: keine	
Ziele: Klarheit über die Anforderungen herstellen.	

AP Titel: 2.1 Ist-Analyse und 2.2 Produktperspektive	AP Nummer: 3
Dauer: 23.10.13 - 27.10.13	Aufwand: 20 Std.
AP-Teilnehmer: Sandor Herms, Olga Miloevich	
Beschreibung: Analyse des momentanen Zustands betreffend der Ressourcen und Kompetenzen der Entwickler und des Kunden. Beschreibung der Einbettung des Produkts in das Gesamtsystem mit den dazu nötigen Schnittstellen.	
Voraussetzung: stattgefundenes Kundengespräch	
Ziele: Die Punkte Ist-Analyse und Produktperspektive der allgemeinen Beschreibung abarbeiten.	

AP Titel: 2.3 Anwendungsfälle (kurz) bis 2.7 Ausblick	AP Nummer: 4
Dauer: 21.10.13 - 25.10.13	Aufwand: 20 Std.
AP-Teilnehmer: Sylvia Kamche Tague, Nils Sören Oja	
Beschreibung: Grobe Liste der Anwendungsfälle aus den Mindestanforderungen erstellen.	
Voraussetzung: keine	
Ziele: Übersicht schaffen, Liste ist auch für den Prototypen wichtig	

AP Titel: 3. Datenmodell	AP Nummer: 5
Dauer: 25.10.13 - 30.10.13	Aufwand: 20 Std.
AP-Teilnehmer: Sandor Herms, Olga Miloevich	
Beschreibung: Das Datenmodell beschreibt die in unserem Bibliotheksprojekt verwendeten Daten	
Voraussetzung: 2.1 Ist-Analyse und 2.2 Produktperspektive	
Ziele: Fertiges Datenmodell, das die Struktur des Systems veranschaulicht	

AP Titel: 3.2 Anwendungsfälle	AP Nummer: 6
Dauer: 28.10.13 - 01.11.13	Aufwand: 20 Std.
AP-Teilnehmer: Sylvia Kamche Tague, Nils Sören Oja	
Beschreibung: Aufzählung aller möglichen Szenarien die vorkommen können.	
Voraussetzung: 2.3 Anwendungsfälle (kurz) bis 2.7 Ausblick	
Ziele: Detaillierte Liste von Anwendungsfällen	

AP Titel: 3.3 Aktionen	AP Nummer: 7
Dauer: 4.11.13 - 06.11.13	Aufwand: 12 Std.
AP-Teilnehmer: Sylvia Kamche Tague, Nils Sören Oja	
Beschreibung: Aufzählung aller möglichen durchführbaren Handlungen die beim durchführen eines Anwendungsfall auftreten.	
Voraussetzung: 3.2 Anwendungsfälle	
Ziele: Detaillierte Liste aller Aktionen, die in den Anwendungsfällen vorkommen	

AP Titel: 3.4 Entwurfseinschränkungen und 3.5 Softwaresystemattribute	AP Nummer: 8
Dauer: 6.11.13 - 12.11.13	Aufwand: 20 Std.
AP-Teilnehmer: Sandor Herms, Olga Miloevich	
Beschreibung: Merkmale des Entwurfes und des Systems, die uns Einschränken.	
Voraussetzung: Datenmodell	
Ziele: Liste von Entwurfseinschränkungen und Softwaresystemattribute.	

AP Titel: GUI-Prototyp: Vorbereitung	AP Nummer: 9
Dauer: 21.10.13 - 23.10.13	Aufwand: 12 Std.
AP-Teilnehmer: Dario Treffenfeld-Mäder, Jannes Uken	
Beschreibung: Erarbeitung eines Konzepts für die GUI unter Gesichtspunkten der Bedienbarkeit, Optik sowie Umsetzbarkeit und Effizienz.	
Voraussetzung: keine	
Ziele: Ein Ziel für die Implementierung definieren.	

AP Titel: GUI-Prototyp: Implementierung	AP Nummer: 10
Dauer: 24.10.13 - 30.10.13	Aufwand: 28 Std.
AP-Teilnehmer: Dario Treffenfeld-Mäder, Jannes Uken	
Beschreibung: Umsetzung des zuvor bestimmten Konzepts.	
Voraussetzung: Abschluss von Arbeitspaket 9	
Ziele: Etwas zwecks Visualisierung präsentieren können.	

AP Titel: GUI-Prototyp: Präsentation	AP Nummer: 11
Dauer: 06.11.13 - 07.11.13	Aufwand: 3 Std.
AP-Teilnehmer: alle	
Beschreibung: Vorstellung des Prototyps.	
Voraussetzung: GUI-Prototyp: Implementierung	
Ziele: Rückmeldung zur Qualität und Vollständigkeit des Prototyps.	

AP Titel: Angebot	AP Nummer: 12
Dauer: 6.11.13	Aufwand: 4 Std.
AP-Teilnehmer: Dario Treffenfeld-Mäder, Jannes Uken	
Beschreibung: Aufzeigen aller zu gewährleistenden und nicht zu gewährleistenden Merkmale und Szenarien des Projektes sowie eine Kosten Einschätzung.	
Voraussetzung: keine	
Ziele: Fertiges Angebot in Briefform.	

6.1.3 Architekturbeschreibung, Schnittstellenbeschreibung, Testplan und Blackbox-Tests

AP Titel: Architekturbeschreibung, Schnittstellenbeschreibung, Testplan und Blackbox-Tests	AP Nummer: 13
Dauer: 18.11.13 - 22.12.13	Aufwand: 150 Std.
AP-Teilnehmer: alle	
Beschreibung: Erstellung von Architekturbeschreibung, Schnittstellenbeschreibung, Testplan und Blackbox-Tests	
Voraussetzung: Anforderungsspezifikation	
Ziele: Fertige Architekturbeschreibung, Schnittstellenbeschreibung, Testplan und Blackbox-Tests.	

...

6.1.4 Implementierung

AP Titel: Implementierung	AP Nummer: 14
Dauer: 22.12.13 - 23.02.14	Aufwand: 380 Std.
AP-Teilnehmer: alle	
Beschreibung: Implementieren des Projekts	
Voraussetzung: Architekturbeschreibung, Schnittstellenbeschreibung, Testplan und Blackbox-Tests	
Ziele: Fertig Implementiert	

...

6.2 Zeitplan und Abhängigkeiten

Abbildung 2: Gantt-Diagramm

6.3 Ressourcenanforderung

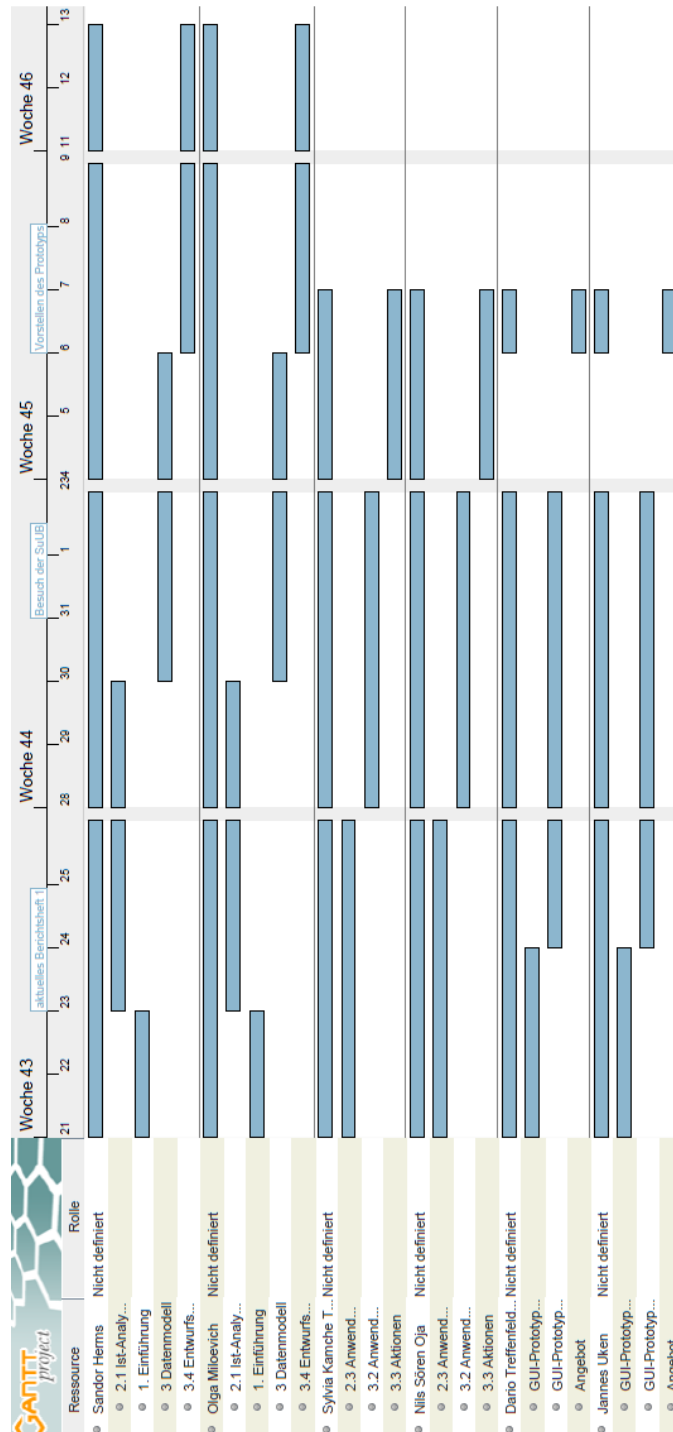


Abbildung 3: Ressourcendiagramm

Die benötigten Ressourcen für das Projekt sind die Mitarbeiter. Die Arbeitspakete werden unter Berücksichtigung der Kompetenzen der Mitarbeiter verteilt.

Der Aufwand eines Arbeitspaketes bestimmt gleichzeitig den Bearbeitungszeitraum sowie die Anzahl der beteiligten Mitarbeiter. So können Arbeitspakete mit überschaubarem Inhalt und Aufwand von einer Person bearbeitet werden.

Die Mitarbeiter müssen die ihnen zugeteilten Arbeitspakete innerhalb des festgelegten Zeitraums abschließen. Der genaue Zeitpunkt der Bearbeitung innerhalb dieses Zeitraums ist den Mitarbeitern überlassen.

Die regelmäßigen Besprechungen werden, so fern möglich, nur mit Anwesenheit aller Mitarbeiter abgehalten. Weiterhin sind alle Mitarbeiter an den Punkten 1.4 Referenzen sowie 1.5 Definitionen und Akronyme beteiligt. Referenziert ein Mitarbeiter etwas oder führt er neue Fachbegriffe ein, so aktualisiert er diese Punkte mit einer sinnvollen Erläuterung.

Bearbeitet von: Olga Miloevich

6.4 Pläne für die Konvertierung von Daten

Es wird geplant, keine Daten zu benutzen, die konvertiert werden müssen.

6.5 Managementpläne für Unterauftragsnehmer

Es gibt keine Unterauftragsunternehmer, alle Arbeiten werden von der Gruppe selbst ausgeführt, insofern ist es auch unnötig, eine dahingehende Schnittstelle zu definieren.

6.6 Ausbildungspläne

In 3.5 ist bereits darauf hingewiesen, was für Fähigkeiten unsere Mitarbeiter haben. Wir gehen davon aus, dass folgende Fähigkeiten noch verbessert werden müssen:

Name	Zuverbessernde Kompetenzen
Olga Miloevich	SQL, Android
...	...

6.7 Raumpläne

Wir haben geplant, uns regelmäßig ein Mal pro Woche zu einem Gruppenmeeting zu treffen. Als Standardtreffpunkt wird uns die Stuga-Cafete in MZH E1 dienen. Treffzeiten werden erst in nächste Woche festgelegt, wenn alle Tutoriumtermine bekannt gegeben werden. Auf diesen Meetings werden Treffen zur inhaltlichen Arbeiten festgelegt.

6.8 Installationspläne

Jeder Gruppenmitglied muß für erfolgreiches Arbeiten folgende Software installiert haben:

1. Java 7 (Programmiersprache)
2. Eclipse (IDE)
3. Android SDK (SDK zur Android Entwicklung)
4. Maven (Projekt-build-tool)
5. Glassfish
6. MS Visio (Software zur Erstellung der UML-Diagramme)
7. Gantt (Projektmanagementsoftware)

Außerdem, muß jedes Gruppenmitglied die Software zur Versionenverwaltung (SVN-Client) und Bearbeitung der Dokumentation (LaTeX-Editor) installiert haben. Die Client-Versionenwahl und Latex-Editor stehen für jedes Gruppenmitglied frei und hängen von seinem Betriebssystem ab.

6.9 Pläne für die Übergabe des Systems

Das Endprodukt wird im Form einer finalen Präsentation dem Kunden und dem Tutor vorgestellt. Die Software wird in entsprechenden Dateien (*.jar/ *.apk) an den Kunden übergeben. Die Software wird auf CD-Disk geschrieben. Außerdem, übergeben wir zu der Software gehörende Dokumentationen, wie Projektplan, Anforderungsspezifikation, GUI-Prototyp, Test-, Architektur- und Implementierungsplan, Lasten und Pflichtenhefte und User Manual.