

Software-Projekt 2 (RE SWP 2014)

VAK 03-BA-901.02

Projektplan

Schibboleth



Patrick Hollatz	phollatz@tzi.de
Tobias Dellert	tode@tzi.de
Tim Ellhoff	tellhoff@tzi.de
Daniel Pupat	dpupat@tzi.de
Olga Miloevich	halfelv@uni-bremen.de
Tim Wiechers	tim3@tzi.de

Abgabe: 04. Mai. 2014 — Version 1.0

Inhaltsverzeichnis

1 Einleitung (Daniel Pupat)	5
1.1 Projektübersicht	5
1.1.1 Ziele	5
1.1.2 Hauptarbeitsaktivitäten und –produkte	6
1.1.3 Haupt-Milestones und grober Zeitplan	7
1.1.4 Benötigte Ressourcen	7
1.1.5 Budget	8
1.1.6 Kontaktdaten des Kunden	8
1.1.7 Mitarbeiter	9
1.2 Auszuliefernde Produkte	10
1.3 Evolution des Plans	10
1.4 Referenzen (Olga Miloevich)	10
1.4.1 Links	11
1.4.2 Weitere	12
1.5 Definitionen und Akronyme	13
2 Projektorganisation (Daniel Pupat)	14
2.1 Prozessmodell	14
2.2 Organisationsstruktur	14
2.3 Organisationsgrenzen und –schnittstellen	15
2.4 Verantwortlichkeiten	15
3 Managementprozess (Olga Miloevich, Tim Wiechers)	16
3.1 Managementprozess und –prioritäten	16
3.2 Annahmen, Abhängigkeiten und Einschränkungen	19
3.2.1 Annahmen	19
3.2.2 Abhängigkeiten	20
3.2.3 Einschränkungen	20
3.3 Risikomanagement	21
3.4 Projektüberwachung	26
3.5 Mitarbeiter	27
3.5.1 Tim Wiechers	27
3.5.2 Tim Ellhoff	27
3.5.3 Tobias Dellert	28
3.5.4 Olga Miloevich	28
3.5.5 Daniel Pupat	29
3.5.6 Patrick Hollatz	29

4 Technische Prozesse (Patrick Hollatz)	29
4.1 Methoden, Werkzeuge und Techniken	29
4.1.1 Entwicklungsplattform	29
4.1.2 Entwicklungsmethode	30
4.1.3 Programmiersprache und Bibliotheken	30
4.2 Dokumentationsplan	30
4.2.1 Codingstyle	30
4.2.2 Kommentarsprache	31
4.2.3 JavaDoc	31
4.2.4 Begleitende Dokumentation	31
4.3 Unterstützende Projektfunktionen	31
5 Arbeitspakete, Zeitplan und Budget	31
5.0.1 Annahmen	31
5.0.2 Anmerkungen	32
5.1 Arbeitspakete, Zeitplan, Abhängigkeiten u. Ressourcenanforderungen	32
5.1.1 Projektplan	34
5.1.2 Anforderungsspezifikation	41
5.1.3 Architekturbeschreibung, Schnittstellenbeschreibung, Testplan und Blackbox-Tests	51
5.1.4 Implementierung	54
5.1.5 Dokumentabgaben	56
5.1.6 Sonstiges	60
5.1.7 Meetings	64
6 Sonstige Elemente (Olga Miloevich)	65
6.1 Pläne für die Konvertierung von Daten	65
6.2 Managementpläne für Unterauftragsnehmer	65
6.3 Ausbildungspläne	65
6.4 Raumpläne	66
6.5 Installationspläne	66
6.6 Pläne für die Übergabe des Systems	66

Abbildungsverzeichnis

1 Mehrliniendiagramm	17
2 Überblick des gesamten Projekts in einem Gantt-Diagramm dargestellt	33
3 Gantt-Projektplan	34
4 Anforderungsspezifikation im Gantt-Diagramm	42

Tabellenverzeichnis

1	Hauptaktivitäten und -produkte	6
2	Mitarbeiter	9
3	Auszuliefernde Produkte	10
4	Definition und Akronyme	13

Version und Änderungsgeschichte

Version	Datum	Änderungen
0.1	23.04.2014	Ziele hinzugefügt.
0.1.1	23.04.2014	Ziele vervollständigt
0.1.2	25.04.2014	Hauptarbeitsaktivitäten und -produkte hinzugefügt.
0.1.3	25.04.2014	Meilensteine eingefügt.
0.1.4	26.04.2014	Benötigte Ressourcen -Menschen hinzugefügt.
0.1.5	26.04.2014	Ressourcen ergänzt -Hardware und -Räume.
0.1.6	26.04.2014	Budget, Kontaktdaten und Mitarbeiter in den Projektplan eingefügt.
0.2	27.04.2014	Produkte, Evolution d. Plans, Defi. und Akronyme hinzugefügt.
0.2.1	28.04.2014	Prozessmodell und Organisationsstruktur eingefügt.
0.2.2	28.04.2014	Org.grenzen und -schnitst., Verantwortlichkeiten eingefügt.
0.3	29.04.2013	Managementprozesse hinzugefügt.
0.4	30.04.2014	Sonstige Elemente und Referenzen hinzugefügt.
1.0	04.05.2014	Erste veröffentlichte Version.

Wichtiger Hinweis: Dieser Projektplan wurde zu einem großen Teil aus verschiedenen Dokumententeilen anderer Projektpläne unserer Gruppenmitglieder aus dem Wintersemester 2013/14 erstellt. Einige Teile wurden komplett übernommen, andere überarbeitet bzw. angepasst.

Diese Vereinbarung haben wir in der Kick-Off-Veranstaltung für RE SWP 2014 mit dem Veranstalter Dr. Karsten Hölscher getroffen.

An geeigneten Stellen wird darauf auch noch näher hingewiesen (siehe auch Abschnitt 1.4).

1 Einleitung (Daniel Pupat)

Dieses Dokument ist ein Projektplan für das Modul Softwareprojekt 2 im Sommersemester 2014 an der Universität Bremen. Der Projektplan entspricht der Struktur ANSI/IEEE Std. 1058.1-1987¹.

1.1 Projektübersicht

1.1.1 Ziele

Das Ziel unserer Gruppe ist es, das Softwareprojekt 2 der Universität Bremen zu bestehen. Dies setzt die Einhaltung der Fristen und Termine, eine ausreichende Fertigstellung des Projekts und die Abgabe aller in SWP2 geforderten Dokumente wie Pro-

¹<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=25325&userType=inst>

jetztplan, Anforderungsspezifikation und Angebot, Architekturbeschreibung, Schnittstellenbeschreibung, Testplan inklusive Blackbox-Tests und ein elektronisch geführtes Berichtsheft voraus. Darüber hinaus wollen wir einen GUI-Prototypen erstellen und den Akzeptanztest bestehen. Eine Quiz-App zu erstellen, steht aber im Vordergrund.

Die Quiz-App benötigt eine Webseite für die Kundin, sowie einen Zugang für mobile Geräte in Form einer App. Ziel ist es, die in der Kick-Off-Veranstaltung vorgestellten und geforderten Mindestanforderungen und eventuell weitergehende Funktionen zu implementieren.

Zu den Mindestanforderungen gehören die Erstellung und Abgabe einer Quiz-App für mobile Geräte und ein Serverprogramm mit Datenbankanbindung. Die App soll später allen Studenten der Universität Bremen zur Verfügung stehen. Zusätzlich soll die Auftraggeberin über eine Webseite neue Fragen erstellen oder alte bearbeiten sowie löschen können.

1.1.2 Hauptarbeitsaktivitäten und -produkte

In einem Softwareprojekt wird der Entwicklungsprozess einer Software in verschiedene Phasen unterteilt.

Da das Projekt aus verschiedenen Aktivitäten besteht, lassen sich diese Aktivitäten zu Arbeitsprodukten zusammenfügen. Die Folgende Tabelle bietet eine Übersicht der einzelnen Aktivitäten und den daraus resultierenden Arbeitsprodukten. Die Arbeitsprodukte werden im Laufe des Projektes nach und nach abgegeben.

Tabelle 1: Hauptaktivitäten und -produkte

Aktivität / Phase	Arbeitsprodukt
Projektplanung	Projektplan
Anforderungsanalyse, Angebotserstellung	Anforderungsspezifikation, Angebot
Entwurf (Globale Analyse, Konzeptionelles Modell, Modulblickwinkel, Ausführungsblickwinkel, Codeblickwinkel)	Architekturbeschreibung
Erstellen des Testplans, Tests	Testplan, Schnittstellentests
Implementierung	lauffähiges Programm
Dokumentation	Installationsanweisung/-Skript
Auslieferung	Kunde erhält Produkt

1.1.3 Haupt-Meilensteine und grober Zeitplan

Die Haupt-Meilensteine resultieren aus den jeweiligen Abgabeterminen der einzelnen Dokumente.

M0 - 23.04.2014 Kick-Off Veranstaltung.

M1 - 24.04.2014 Beginn des Projektes.

M2 - 04.05.2014 Abgabe initialer Projektplan.

Jedes Mitglied muss seinen Teil fertig gestellt haben. Anschließend werden alle Einzelteile zusammengeführt und von allen auf Korrektheit geprüft.

M3 - 08.05.2014 Kundengespräch.

M4 - 27.05.2014 Anforderungsspezifikation (intern).

Jedes Mitglied hat seinen Teil der Anforderungsspezifikation fertiggestellt. Anschließend werden die Teile zusammengeführt und von allen auf Korrektheit geprüft.

M5 - 01.06.2014 Abgabe der Anforderungsspezifikation, GUI-Prototyp und Angebot.
Meilenstein 4 muss bereits fertig sein. Der GUI-Prototyp muss vollständig entwickelt sein. Abgabe via MEMS.

M6 - 01.07.2014 Architektur- und Schnittstellenbeschreibung, Testplan, Tests (intern).

Jedes Mitglied muss seine Aufgaben erfüllt haben. Teile werden zusammengeführt und kontrolliert. Tests müssen implementiert sein.

M7 - 06.07.2014 Architekturbeschreibung, Testplan und Schnittstellentests fertig.

Meilenstein 6 muss bereits erreicht worden sein. Tests wurden lauffähig implementiert. Abgabe via MEMS.

M8 - 28.07-01.08.2014 Akzeptanztest.

M9 - 10.08.2014 Vollständige Abgabe der Dokumente und der Software.

Die Software muss lauffähig und vollständig implementiert sein,
Abgabe des Build-/Installationsskriptes

1.1.4 Benötigte Ressourcen

• Menschliche Ressourcen

An menschlichen Ressourcen stehen sechs Informatikstudenten der Universität Bremen zur Verfügung. Wir haben als durchschnittliche Arbeitszeit pro Woche und Person einen Aufwand von ca. 17 Stunden für das Projekt errechnet. Dieser Wert ergibt sich folgendermaßen:

Für das Modul Software Projekt 2 gibt es 9CP. 1CP entspricht 30 Semesterwochenstunden. $9 \times 30 = 270$ Stunden. Da wir 16 Wochen lang an dem Projekt arbeiten werden, ergibt sich ein aufgerundeter Wert von 17 Stunden pro Woche ($270 / 16 = 16,875$). Unsere Kontaktdaten sind dem Punkt Mitarbeiter im Abschnitt [1.1.7](#) zu entnehmen.

• Hard- und Software

Jedes unserer Mitglieder ist im Besitz oder hat Zugriff auf Computer, die folgenden Anforderungen und Verfügbarkeiten gerecht werden müssen:

- zum Anfertigen der Dokumente wird ein Textsatzprogramm benötigt (LaTeX wird bevorzugt).
- für die Entwicklung der Software müssen Java-Runtime, ein Texteditor und eine Entwicklungsumgebung mit Android-SDK installiert sein.
- Git wird zum gleichzeitigen Bearbeiten der Dokumente und zum Datenaustausch der Entwickler benötigt.

• Räume

Das Team wird sich während der gesamten Projektlaufzeit regelmäßig in der Ebene 0 im MZH an der Universität treffen. Weitere spezielle Räumlichkeiten werden nicht benötigt, da wir den Kontakt regelmäßig via Skype oder E-Mail gewährleisten.

1.1.5 Budget

Ein Budget für dieses Projekt in Form von Geld entfällt, da die Software im Rahmen des Moduls Software Projekt 2 entwickelt wird. Wenn wir über 16 Wochen (vom 24.04.2014 bis zum 10.08.2014) an dem Projekt mit 6 Studenten 17 Stunden pro Woche arbeiten, ergibt sich eine Gesamtsumme von 1632 Entwicklerstunden ($16 \times 6 \times 17 = 1632$).

Wir entnehmen einer Studie von Gulp ², dass zwei Drittel der Software-Entwickler zwischen 60 und 80 Euro fordern. Da wir alle Studenten sind und uns somit noch in der Ausbildung befinden, setzen wir den Stundenlohn für jeden Entwickler bei 40 Euro an. Somit würden sich für den Arbeitsaufwand der Entwicklerstunden Kosten von insgesamt 65.280 Euro ergeben.

1.1.6 Kontaktdaten des Kunden

Jacqueline Sprindt

*Rektorat der Universität Bremen
(vertreten durch die Pressestelle)*

²<http://www.gulp.de/presse/pressemitteilungen/markestudie-freiberufliche-software-entwickler-sind-g.html>

1.1.7 Mitarbeiter

In der Folgenden Tabelle 2 stehen die Kontaktdaten aller am Projekt Beteiligten. Ihr ist in Folge Nachname, Vorname, E-Mail und ein Foto des jeweiligen Teammitglieds zu entnehmen.

Tabelle 2: Mitarbeiter

Name	Email	Foto
Wiechers, Tim	tim3@tzi.de	
Hollatz, Patrick	phollatz@tzi.de	
Dellert, Tobias	tode@tzi.de	
Ellhoff, Tim	tellhoff@tzi.de	
Pupat, Daniel	dpupat@tzi.de	
Miloevich, Olga	halfelv@uni-bremen.de	

1.2 Auszuliefernde Produkte

Die Tabelle 3 listet alle auszuliefernden Produkte auf, die während des gesamten Projektes anfallen werden³.

Tabelle 3: Auszuliefernde Produkte

Datum	Beschreibung	Anzahl
04.05.2014	Initialer Projektplan (dieses Dokument)	1
01.06.2014	Anforderungsspezifikation	1
01.06.2014	GUI-Prototyp	1
01.06.2014	Angebot	1
06.07.2014	Architekturbeschreibung	1
06.07.2014	Schnittstellenbeschreibung	1
06.07.2014	Testplan	1
10.08.2014	Vollständige Abgabe	1

1.3 Evolution des Plans

Der Projektplan wird über die gesamte Dauer der Entwicklung durch das Team aktualisiert. Die erste absehbare Aktualisierung wird nach dem Kundengespräch am 08.05.2014 durchgeführt. Weitere absehbare Aktualisierungen des Projektplans sind nach den jeweiligen Hauptabgaben vom Entwicklerteam durchzuführen.

Aufgrund der stetigen Entwicklung des Systems sind weitere Aktualisierungen abzusehen, vor allem hinsichtlich der Arbeitspakete. Alle Änderungen werden von dem jeweiligen Phasenleiter überwacht und auf Korrektheit geprüft. Als unvorhergesehene Aktualisierungen wären z.B. das Austreten eines Mitglieds aus der Gruppe zu nennen, da dies die gravierendste Umstrukturierung mit sich bringt. Die Arbeitspakete müssten in dem Fall neu auf die restlichen Teammitglieder aufgeteilt werden, was wiederum der jeweilige Phasenleiter übernehmen würde.

1.4 Referenzen (Olga Miloevich)

Im Folgenden sind alle mit diesem Dokument in Beziehung stehenden Referenzen aufgelistet. Vor allem die Referenzen auf die Gruppen sind hier hervorzuheben, da einige Inhalte, die diesem Dokument zugrunde liegen, z.T. aus vorherigen Gruppen aus SWP2 stammen bzw. weiterentwickelt wurden.

³Berichtshefte sind in dieser Tabelle bewusst nicht aufgeführt.

1.4.1 Links

- 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
- Gruppe "YNotZoidberg", Projektplan, SWP2, WS2013/14
- Gruppe "IT_R3V0LUT10N" Projektplan, SWP2, WS2013/14
- Gruppe "ZuSpaet" Projektplan, SWP2, WS2013/14
- <http://www.eclipse.org/>
Die Website von Eclipse.
- <https://developer.android.com/sdk/index.html>
Die Website von Android-SDK
- <http://maven.apache.org/>
Die Website des Mavenprojekts.
- <http://glassfish.java.net/>
Die Website von GlassFish.
- <http://junit.org/>
Die Website des Test-Frameworks JUnit
- <http://github.com/>
Die Website von GitHub.
- <http://www.ganttpoint.biz/>
Projektseite für Gantt-Diagramme.
- <http://www.latex-project.org/>
Die Website des Textsatzsystems LATEX.
- [http://www.oracle.com/technetwork/java/codeconv-138413.html/](http://www.oracle.com/technetwork/java/codeconv-138413.html)
Java Coding-Guidlines.
- <http://www.gulp.de/presse/pressemitteilungen/marktstudie-freiberuflche-software-entwickler-sind-gefragt.html>
Die Website für die Information des durchschnittlichen Lohns für Software-Entwickler.
- <https://www.dreamspark.com/>
Website von Microsoft Dreamspark zum Download von MS Project für die Gantt-Diagrammerstellung.

1.4.2 Weitere

- Vorlage dieses Dokuments - Stud.IP Software Projekt 2 1-Projektplan-Vorlage.tex
- Hinweise zu diesem Dokument - Stud.IP 1-Hinweise-Abgabe-Projektplan.pdf
- Java Referenz - <http://www.oracle.com/technetwork/java/javase/documentation/api-jsp-136079.html>

1.5 Definitionen und Akronyme

Tabelle 4: Definition und Akronyme

Begriff	Bedeutung
Andorid	Betriebssystem und Software-Plattform für mobile Geräte
Android-SDK	SDK = Software-Development-Tool
Ansi/IEEE	eine festgelegte Norm vom 'Institute of Electrical and Electronics Engineers, ANSI ist die Abkürzung für 'American National Standards Institute'
App	Programm, welches auf mobilen Endgeräten läuft
Build-/Installationsskript	Anleitung zum Installieren der Software
CP	Credit Points, 1CP entspricht 30 Semesterwochenstunden Arbeitsaufwand
Doxygen	Schnittstellendokumentationssystem
Eclipse	die von uns genutzte Entwicklungsumgebung
Git/GitHub	Online-Projekt Hoster zur Verwaltung von Dokumenten und Sourcecode
GUI	Grafische Oberfläche, Abkürzung für Graphical User Interface
Java	Java ist eine Programmiersprache
JUnit	Framework zum Testen von Java-Programmen
L ^A T _E X	Ein Textsatzprogramm
Maven	Projektkompilierungssystem
MEMS	Ein elektronisches Abgabesystem der Universität Bremen
Server	Ein dauerhaft erreichbarer Rechner, der einen Dienst bereitstellt
Skype	Online Kommunikations-Tool
Wasserfallmodell	Ein lineares Vorgehensmodell der Softwareentwicklung

2 Projektorganisation (Daniel Pupat)

2.1 Prozessmodell

Das von uns verwendete Wasserfallmodell gliedert die einzelnen Phasen sequentiell aufeinanderfolgend:

- Anforderungsspezifikation
- Architekturbeschreibung (Entwurf)
- Implementierung
- Test
- Dokumentation
- Auslieferung

In unserem Fall werden sich die Phasen Implementierung, Test und Dokumentation überschneiden, um eine effiziente Arbeitsweise zu gewährleisten. Zudem werden wir bereits vor der Implementierung Black-Box-Tests anfertigen.

2.2 Organisationsstruktur

Unsere Dateien werden im uns zur Verfügung gestellten Git-Repository gespeichert. Die Kommunikation findet über E-Mail und über Skype statt. Da eines unserer Mitglieder am Wochenende kein Internetzugang hat, muss notfalls Kontakt per Handy aufgenommen werden.

Zudem werden wir uns wöchentlich in der Ebene 0 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 Controller, der die Arbeit des Projektleiters überprüft. 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 (Abschnitt 3) zu finden.

2.3 Organisationsgrenzen und –schnittstellen

Bei dem Arbeitgeber und der übergeordneten Organisation handelt es sich um zwei verschiedene Parteien, da es sich bei dem Auftraggeber um einen echten Kunden handelt.

Auftraggeber:

*Jacqueline Sprindt
Rektorat der Universität Bremen
(vertreten durch die Pressestelle)*

Übergeordnete Organisation:

*Dr. Karsten Hölscher
Büro: ECO5 (TAB) 2.56
Telefon: +49 (421) 218 64475
Fax: +49 (421) 218 4322
E-Mail: hoelsch@uni-bremen.de*

2.4 Verantwortlichkeiten

Mitarbeiter	Rolle
Tobias Dellert	Projektleiter Phasenleiter Projektplan
Daniel Pupat	Controller Phasenleiter Anforderungsspezifikation
Olga Miloevich	Phasenleiterin Architekturbeschreibung
Tim Ellhoff	Phasenleiter Implementierung
Patrick Hollatz	Phasenleiter Testplan
Tim Wiechers	Phasenleiter Präsentation

Weitere Verantwortlichkeiten, die von jeweils einem Teammitglied während des gesamten Projektzeitraumes besetzt werden müssen, sind Qualitätsmanagement und Risikomanagement. Die Aufgaben des Qualitätsmanagers sind es, die Qualität aller Bearbeitungen der Teammitglieder sicherzustellen. Zu den Aufgaben des Risikomanagers gehören das frühzeitige Erkennen von möglichen Problemen und diese präventiv zu vermeiden oder einzudämmen, indem er z.B. Arbeitspakete umverteilt oder andere Lösungen findet. Siehe Abschnitt 3.3.

3 Managementprozess (Olga Miloevich, Tim Wiechers)

3.1 Managementprozess und -prioritäten

Folgender Abschnitt wurde vom Projektplan von der Gruppe YNotZoidberg (Vorlesung Software Projekt 2, WiSe2013/14) übernommen.

Für die Organisation unseres Projektes haben wir einen Projektleiter gewählt.

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 Prioritäteten 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 Projektleiter.

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); Projektplan von der Gruppe "YNotZoidberg" (Vorlesung Software Projekt 2 2013/14)

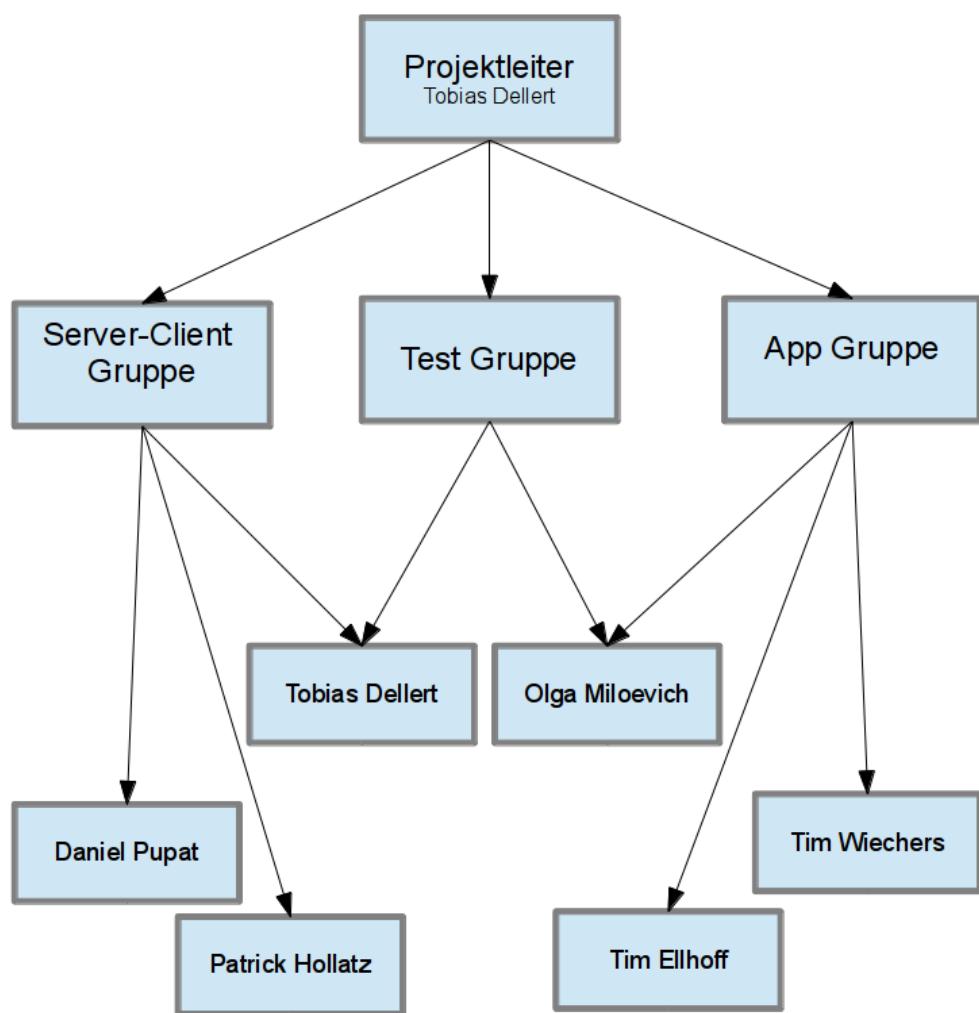


Abbildung 1: Mehrliniendiagramm

Folgender Abschnitt würde vom Projektplan von der Gruppe IT_R3V0LUT10N (Vorlesung Software Projekt 2, WiSe2013/14, übernommen.

Die folgenden Managementprozesse haben bei uns außerdem Priorität:

Qualität des Produktes:

Unser Ziel ist es, dem Produkt eine hohe Qualität zu geben. Dies ist notwendig, damit der Kunde zufrieden ist und das Produkt später evtl. verwendet wird. Die Benutzung soll einfach und benutzerfreundlich sein.

Dieses Ziel hat eine hohe Priorität, da dies notwendig ist, um den Kunden zufriedenzustellen und eine gute Note zu erreichen.

Weiterentwicklung des Produktes:

Es ist auch wichtig, das Produkt so zu entwickeln, damit dieses später bei Bedarf von Anderen weiterentwickelt werden kann. Dies erfordert eine strukturierte Implementierung und ausführliche Dokumentierung.

Dieses Ziel hat niedrige Priorität, da wir in erster Linie das Modul bestehen wollen.

Kundenzufriedenheit:

Wir wollen, dass der Kunde später zufrieden ist, was bedeutet, dass man die Mindestanforderungen erfüllt und darüber hinaus noch weitere Features einbindet, da nur so der Kunde wirklich zufrieden ist.

Dieses Ziel hat für uns mittlere Priorität, da wir in erster Linie die Mindestanforderungen schaffen wollen und nur wenn noch Zeit ist, weitere Features einbinden. Dies könnte aber noch notwendig sein, um eine gute Note zu erreichen.

Kommunikation innerhalb der Gruppe:

Ein wichtiger Faktor ist die Kommunikation innerhalb der Gruppe. Wenn man sich nicht abspricht, kann es zu Schwierigkeiten kommen, wenn z.B. ein Gruppenmitglied seinen Teil nicht rechtzeitig schafft und die anderen aber davon ausgehen.

Dieses Ziel hat bei uns eine hohe Priorität, da ohne Kommunikation das Projekt mit hoher Wahrscheinlichkeit scheitert.

Klima innerhalb der Gruppe:

Ein gutes Gruppenklima heißt, dass innerhalb der Gruppe alle gut miteinander auskommen und es keinen Streit gibt. Außerdem muss man den anderen vertrauen können, dass sie immer rechtzeitig fertig werden und bei Problemen Bescheid geben.

Dies hat ebenfalls eine hohe Priorität, da gerade das Vertrauen und die Zuverlässigkeit sehr wichtig sind, damit alles rechtzeitig fertig wird.

Gute Note:

Ziel dieser Veranstaltung ist für uns das Projekt so gut wie möglich zu bestehen. Dabei sollte jeder sein Bestes geben, damit am Ende das Maximum an Punkten für die Gruppe

erreicht wird.

Dies hat bei uns eine hohe Priorität, da wir später einen möglichst guten Abschluss haben wollen.

Kunde entscheidet sich für unser Produkt:

Da der Kunde am Ende der Veranstaltung ein Produkt aussuchen wird, welches dann in der Universität verwendet wird, wäre es möglich, dass er unser Produkt wählt.

Dieses Ziel hat bei uns eine niedrige Priorität, da wir in erster Linie gut abschneiden wollen, aber nicht darauf hinarbeiten, unbedingt das beste Produkt der Veranstaltung zu entwickeln, da dies zu zeitaufwendig wäre.

3.2 Annahmen, Abhängigkeiten und Einschränkungen

Folgender Abschnitt wurde vom Projektplan von der Gruppe IT-R3V0LUT10N (Vorlesung Software Projekt 2, WiSe2013/14, übernommen.

3.2.1 Annahmen

Mindestanforderungen werden nicht verändert:

Die erste Annahme ist, dass der Kunde die Mindestanforderungen nicht verändert. Dies bedeutet, dass es keine Möglichkeit gibt, andere Mindestanforderungen auszuhandeln und der Kunde auch keine neuen stellt.

Deadline wird nicht verschoben:

Noch eine Annahme ist, dass sich die Deadlines der verschiedenen Abgaben unter normalen Umständen nicht verändern. Dies bedeutet, dass der Kunde diese nicht vorverlegt und wir diese nicht nach hinten verlegen können.

Erfolgreiche Teilnahme:

Eine weitere Annahme ist, dass alle Gruppenmitglieder die Veranstaltung erfolgreich bestehen wollen. Da sich alle für dieses Modul eingetragen haben, kann man davon ausgehen, dass alle ihr Bestes geben, um diese Veranstaltung zu bestehen.

Grundkenntnisse in Java:

Man kann auch annehmen, dass alle Mitglieder Grundkenntnisse in Java haben, da alle Gruppenmitglieder bereits die Veranstaltungen Praktische Informatik 1 und 2 besucht haben.

3.2.2 Abhängigkeiten

Laptop:

Da jeder von uns ein Laptop besitzt, werden wir diesen hauptsächlich verwenden, da so jeder mobil ist und überall weiterarbeiten kann.

GitHub:

Zum Teilen der Dokumente verwenden wir GitHub. So kann jede Person einen Teil bearbeiten und die Dokumente können dann zusammengeführt werden.

Mitglieder:

Da dies eine Gruppenarbeit ist, muss jedes Gruppenmitglied seinen Teil leisten, da die Arbeit auf sechs Leute ausgelegt ist.

Von den eben genannten Punkten ist das Projekt abhängig, da bei einem Ausfall der Punkte Schwierigkeiten auftreten können.

3.2.3 Einschränkungen

Weitere belegte Module:

Jeder von uns belegt noch weitere Module und hat deswegen nur eine gewisse Zeit für SWP2. Hinzu kommt noch, dass wir Mitglieder haben, die in unterschiedlichen Semestern sind, wodurch es schwierig ist, einen gemeinsamen Termin zu finden.

Deadlines:

Weiterer zeitliche Einschränkung ist bedingt durch die festgelegten Zeitpunkte der einzelnen Abgaben.

3.3 Risikomanagement

Folgender Abschnitt würde vom Projektplan von der Gruppe *IT_R3V0LUT10N* (Vorlesung Software Projekt 2, WiSe2013/14, übernommen.

Risiko	EW (1-10)	SH (1-10)	RH
Krankheitsbedingter Ausfall eines Gruppenmitglieds	5	4	20
Krankheitsbedingter Ausfall mehrerer Gruppenmitglieder	2	7	14
Austritt eines Gruppenmitglieds	4	5	20
Austritt mehrerer Gruppenmitglieder	1	8	8
Inkompetenz eines Gruppenmitglieds	2	7	14
Mangelhafte Kommunikation innerhalb der Gruppe	4	6	24
Auflösung/Teilung der Gruppe	2	10	20
Unstimmigkeiten in der Gruppe	2	5	10
Mangelnde Motivation in der Gruppe	6	5	30
Zeitmangel	6	6	36
Probleme mit neuen Technologien	5	3	15
Ausfall von GitHub	1	8	8
Ausfall des Glassfish-servers	1	8	8
Mindestanforderung des Kunden falsch interpretiert	3	9	27
Fehler in Architekturbeschreibung	3	7	21
Ausfall des Rechners eines Gruppenmitglieds	2	7	14
Arbeitsfortschritt geht verloren	2	5	10

EW = Eintrittswahrscheinlichkeit (Skala 1:gering - 10:hoch)

SH = Schadenshöhe (Skala 1:gering - 10:hoch)

RH = Risikohöhe (EW * SH)

Krankheitsbedingter Ausfall eines/mehrerer Gruppenmitglieds/er:

Aufgrund von Krankheiten fallen eine oder mehrere Personen aus und können nicht mehr richtig oder für eine gewisse Zeit überhaupt nicht mehr mitarbeiten. Dadurch kommt auf die restliche Gruppe mehr Arbeit zu.

Maßnahmen zur Schadensbegrenzung:

1. Gruppenmitglied benachrichtigt die anderen Mitglieder so früh wie möglich, damit diese sich darauf einstellen können.
2. Die Gruppe sucht Gespräch mit dem Tutor, wenn mehrere Personen ausfallen.

Austritt eines/mehrerer Gruppenmitglieds/er:

Aufgrund von Zeitmangel, Studienabbruch und anderen Gründen kann es jederzeit

passieren, dass Gruppenmitglieder aus der Gruppe austreten. Dadurch müssen die anderen Personen dann entsprechend mehr arbeiten, was zu Problemen führen kann.

Maßnahmen zur Vorbeugung:

1. Im Zeitplan vor den Deadlines immer ein wenig Zeit überlassen, um durch einen plötzlichen Austritt die Abgabe noch rechtzeitig zu schaffen.

Maßnahmen zur Schadensbegrenzung:

2. Bei einem Austritt aus der Gruppe gibt das Mitglied den anderen sofort Bescheid, damit diese sich rechtzeitig auf die Mehrarbeit einstellen können.
3. Sollten mehrere Mitglieder austreten, Gespräch mit dem Tutor suchen um gegebenenfalls die Anforderungen zu senken.

Inkompetenz eines Gruppenmitglieds:

Es kann passieren, dass ein Gruppenmitglied inkompotent ist und somit nicht in der Lage ist, die ihm zugewiesenen Aufgaben zu lösen. Das kann daran liegen, dass dieses Mitglied im bisherigen Studienverlauf immer durch andere Gruppenmitglieder die Module bestanden hat. Dabei kann es passieren, dass er unentdeckt bleibt und so erst spät festgestellt wird, dass ein Teil nicht funktioniert.

Maßnahmen zur Schadensbegrenzung:

1. Durch wöchentliche Treffen wird der Fortschritt besprochen und wenn jemand dies nicht schafft, wird dies frühzeitig erkannt und der Teil kann unter den anderen Mitgliedern aufgeteilt werden.
2. Jedes Gruppenmitglied sollte sich bei Problemen frühzeitig an die Gruppe wenden, damit die anderen diesem helfen kann.
3. Wenn dieses Gruppenmitglied zu einer Belastung für die Gruppe wird, kann durch eine Gruppenentscheidung dieses Mitglied aus der Gruppe ausgeschlossen werden.

Mangelhafte Kommunikation innerhalb der Gruppe:

Da wir in einer relativ großen Gruppe arbeiten, wird die Arbeit aufgeteilt. Dabei sind viele Teilaufgaben abhängig von anderen, was dazu führen kann, dass ohne Kommunikation die Teilaufgaben oder Implementierungen nicht zusammenpassen. Dadurch kann es im späteren Verlauf zu großen Problemen kommen, da so die Software evtl. nicht läuft.

Maßnahmen zur Vorbeugung:

1. Die Mitglieder, die stark voneinander abhängige Aufgaben haben, sollten sich vorher genau absprechen und auch die ganze Zeit in Kontakt bleiben, um Missverständnisse zu vermeiden.
2. In den wöchentlichen Treffen jede Aufgabe ansprechen, damit jeder weiß, was die anderen Mitglieder wie machen.

Maßnahmen zur Schadensbegrenzung:

3. Jeder sollte sich die Aufgaben der anderen Mitglieder immer durchlesen und gerade bei abhängigen Aufgabenteilen genau darauf achten, dass diese zusammenpassen.

Teilung der Gruppe:

Wenn es Probleme oder Unstimmigkeiten innerhalb der Gruppe gibt und sich zwei Lager bilden, kann es dazu führen, dass sich die Gruppe trennen muss. Dies kann auch passieren, wenn sich herausstellen sollte, dass die Mitglieder starke unterschiedliche Fähigkeiten besitzen und so die besseren Personen die Hauptarbeit verrichten müssen und diese damit nicht einverstanden sind.

Maßnahmen zur Vorbeugung:

1. Bei unterschiedlichen Fähigkeiten früh festlegen, dass diese später bessere Noten bekommen, damit diese auch zufrieden sind.

Maßnahmen zur Schadensbegrenzung:

2. Bei Problemen und Unstimmigkeiten das Gespräch suchen und diese ansprechen und gegebenenfalls auch den Tutor hinzuziehen, um die Probleme zu lösen.

Auflösung der Gruppe:

Es kann aufgrund von schlechten Abgaben, Zeitmangel, Unstimmigkeiten oder Gruppenaustritten eine Gruppenauflösung geben, was zu einem Nichtbestehen des Moduls führen würde, da man alleine dieses wahrscheinlich nicht schaffen würde.

Maßnahmen zur Vorbeugung: 1. Bei Problemen frühzeitig das Gespräch suchen, um zu verhindern, dass sich die Gruppe auflöst.

Unzuverlässigkeit eines Gruppenmitglieds:

Wenn ein Gruppenmitglied seine Aufgaben nicht zu dem geplanten Zeitpunkt fertig bekommt, da er keine Zeit hatte oder andere Prioritäten gesetzt hat, kann es dazu führen, dass die Abgabe nicht vollständig ist und es eine 5.0 für die Abgabe gibt.

Maßnahmen zur Vorbeugung:

1. Bei der Planung der Zeit immer ein wenig Luft lassen, damit man noch reagieren kann, falls ein Mitglied seine Aufgaben nicht gemacht hat.

Maßnahmen zur Schadensbegrenzung:

2. Im wöchentlichen Treffen den Fortschritt jedes Mitglieds begutachten und evtl. darauf reagieren, sollte ein Mitglied nicht im Zeitplan sein.
3. Das Mitglied, wenn es die Aufgaben nicht macht, beim ersten Mal ermahnen und beim wiederholten Male aus der Gruppe ausschließen.

Unstimmigkeiten in der Gruppe:

Da sich in der Gruppe Leute befinden, die sich vorher nicht kannten, kann es passieren, dass sich Mitglieder mit anderen Mitgliedern nicht verstehen und so das Gruppenklima stören.

Maßnahmen zur Schadensbegrenzung:

1. Bei einem Problem müssen die anderen Gruppenmitglieder schlichten und z.B. die Probleme in einem Gruppentreffen ansprechen und lösen.

Mangelnde Motivation in der Gruppe:

Da SWP 2 sehr zeitaufwendig ist und sich Abgaben über mehrere Wochen erstrecken, kann es gerade am Anfang einer neuen Abgabe zu mangelnder Motivation kommen, da man denkt, dass man noch genug Zeit hat.

Maßnahmen zur Vorbeugung:

1. Treffen mit der Gruppe planen, ohne am Projekt zu arbeiten, um die Moral der Gruppe zu stärken und sich gegenseitig zu motivieren.

Zeitmangel:

Da es feste Deadlines für die einzelnen Abgaben gibt, kann es zu Zeitproblemen kommen. Auch durch Ausfälle von Mitgliedern oder schlechte Zeiteinplanung kann es zu zeitlichem Mangel kommen.

Maßnahmen zur Vorbeugung:

1. Bei der Planung immer einen Zeitpuffer lassen, damit man noch vor den Abgaben Luft hat, falls es irgendwelche Probleme während der Bearbeitungsphase gibt.
2. Bei den wöchentlichen Treffen immer überprüfen, ob jeder im Zeitplan ist, um notfalls noch früh genug auf Zeitprobleme zu reagieren.

Maßnahmen zur Schadensbegrenzung:

3. Wenn ein Aufgabenteil in Zeitverzug kommt, diesen auf mehreren Mitgliedern aufteilen, damit dieser noch rechtzeitig fertig wird.

Probleme mit neuen Technologien:

Da wir mit Technologien arbeiten müssen, mit denen wir noch keine oder nur wenig Erfahrung haben wie z.B. Android, kann dies zu unerwarteten Problemen führen. Dadurch kann sich der zeitliche Aufwand stark erhöhen.

Maßnahmen zur Vorbeugung:

1. Im Zeitplan bereits Zeit zur Einarbeitung neuer Programme einplanen, damit es noch genug Zeit zur Bearbeitung gibt.
2. Vorher die Mitglieder so aufteilen, dass die Personen, die bereits Erfahrung mit den jeweiligen Technologien haben, diese Aufgabenteile übernehmen.

Maßnahmen zur Schadensbegrenzung:

3. Falls ein Gruppenmitglied mit einer Technologie überhaupt nicht zurecht kommt, können die Aufgaben mit einem anderen Mitglied getauscht werden.

Ausfall von GitHub

Wir benutzen GitHub als Repository. Es kann passieren, dass es Probleme mit dem Server gibt, was bei uns zu organisatorischen Problemen führen kann. **Maßnahmen zur Vorbeugung:**

1. Jeder sollte regelmäßig Backups durchführen, damit man sichergehen kann, dass keine Daten bei einem Serverabsturz verloren gehen.

Maßnahmen zur Schadensbegrenzung:

2. Bei einem Serverausfall werden dem Phasenleiter alle Dokumente per E-Mail geschickt, damit dieser diese zusammenfügen und abschicken kann.

Mindestanforderung des Kunden falsch interpretiert:

Da es viele Forderungen des Kunden gibt, kann es sein, dass eine falsch verstanden oder vergessen wird, was zu einem Nichtbestehen führen würde.

Maßnahmen zur Vorbeugung:

1. Jedes Mitglied sollte vor der Abgabe für sich alleine noch einmal alle Anforderungen überprüfen und beim Zweifeln die Gruppe informieren.
2. Bei unklaren Forderungen im Kundengespräch bereits konkrete Fragen dazu stellen.

Fehler in Architekturbeschreibung:

Es kann passieren, dass man in der Architekturbeschreibung bereits Fehler eingebaut hat, die dann später bei der Implementierung entdeckt werden. Dadurch können große Probleme auftreten, gerade wenn man versucht, zwei Codestücke zusammenzufügen.

Maßnahmen zur Vorbeugung:

1. Bei der Architekturbeschreibung konzentriert arbeiten und sich Zeit lassen für Klassendiagramme etc., damit dort möglichst keine Fehler auftreten.

Maßnahmen zur Schadensbegrenzung:

2. Sollte jemandem ein Fehler auffallen, der mehrere Komponenten betrifft, sollte dieser möglichst schnell mit der ganzen Gruppe gelöst werden.

Ausfall des Rechners eines Gruppenmitglieds:

Ohne einen Rechner kann nicht effektiv am Projekt mitgearbeitet werden. Ein Ausfall ist demnach ein Risiko, bei dem dringend ein Ersatz gefunden werden muss.

Maßnahmen zur Vorbeugung:

1. Verantwortungsbewusster Umgang und Vorsicht im Zeitraum des Projekts.

Maßnahmen zur Schadensbegrenzung:

2. Als Alternative kann man auf die Rechner der Uni Bremen ausweichen und büßt so nur Mobilität ein.

Arbeitsfortschritt geht verloren:

Ein stundenlanges Arbeiten kann durch unvorhersehbare, z.B. technische Probleme verloren gehen.

Maßnahmen zur Vorbeugung:

1. Häufiges Abspeichern.
2. Häufiges Committern der Änderungen ins GitHub-Repository.

Die Maßnahmen zur Vorbeugung werden über die gesamte Zeit des Projektes angewendet. Bei den Maßnahmen zur Schadensbegrenzung nach Eintritt entscheidet der Phasenleiter, welche Maßnahmen getroffen werden.

Aufgrund der getroffenen Maßnahmen verändern sich die Werte wie folgt:

Risiko	NEW (1-10)	NSH (1-10)	NRH
Krankheitsbedingter Ausfall eines Gruppenmitglieds	5	3	15
Krankheitsbedingter Ausfall mehrerer Gruppenmitglieder	2	5	10
Austritt eines Gruppenmitglieds	4	3	12
Austritt mehrerer Gruppenmitglieder	1	6	6
Inkompetenz eines Gruppenmitglieds	1	4	4
Mangelhafte Kommunikation innerhalb der Gruppe	3	5	15
Teilung der Gruppe	1	8	8
Auflösung der Gruppe	1	10	10
Unzuverlässigkeit eines Gruppenmitglieds	2	7	14
Unstimmigkeiten in der Gruppe	1	5	5
Mangelnde Motivation in der Gruppe	5	5	25
Zeitmangel	4	5	20
Probleme mit neuen Technologien	4	2	8
Ausfall von GitHub	1	6	6
Ausfall des Glassfish-servers	1	8	8
Mindestanforderung des Kunden falsch interpretiert	2	9	18
Fehler in Architekturbeschreibung	3	5	15
Ausfall des Rechners eines Gruppenmitglieds	2	2	4
Arbeitsfortschritt geht verloren	2	3	6

NEW = Neue Eintrittswahrscheinlichkeit(Skala 1:gering - 10:hoch)

NSH = Neue Schadenshöhe (Skala 1:gering - 10:hoch)

NRH = Neue Risikohöhe (NEW * NSH)

3.4 Projektüberwachung

Um das Projekt zu überwachen, wird wöchentlich ein Treffen stattfinden, wo überprüft wird, wie weit jeder ist. Später, wenn die Aufgaben stark voneinander abhängig sind, sollte es ein weiteres Treffen geben, um sich untereinander besser abzusprechen. Dies wird gerade ab der Architekturplanung wichtig. Außerdem wird es einen permanenten Austausch über Skype geben, indem über Probleme und Anliegen diskutiert wird. Dieser läuft in erster Linie über einen Gruppenchat ab, sollte es aber einmal etwas konkreter

werden, wird untereinander telefoniert.

Es wird auch für jede Phase einen Phasenleiter geben, der dafür zuständig ist, den Zeitplan im Auge zu behalten. Diesem Phasenleiter muss dann jedes Gruppenmitglied regelmäßig Bescheid geben, wie weit die Teilaufgabe bereits bearbeitet ist. Sollte es Probleme geben, ist die Aufgabe des Phasenleiters, das Risiko einzuschätzen und Maßnahmen zu treffen, welche im Punkt Risikomanagement erläutert sind.

Bei größeren Problemen kann ein Treffen spontan einberufen werden oder es wird direkt bei Skype angesprochen und da versucht, dies zu lösen.

3.5 Mitarbeiter

Die Mitarbeiter sollten ausreichend Programmierkenntnisse in Java besitzen, damit diese später auch in der Lage sind, die Quizsoftware zu entwerfen. Auch sollten diese zumindest an SWP1 teilgenommen haben, um die Grundkenntnisse zu haben, die Aufgaben zu bearbeiten.

Die Mitarbeiter sollten auch in der Lage sein, im Team zu arbeiten und sich eigenständig in Themengebiete einzuarbeiten. Bei Problemen sollten die Mitarbeiter sich direkt an die Gruppe wenden, damit diese frühzeitig gelöst werden. Außerdem sollten sie genügend Kenntnisse in Deutsch und Englisch haben, um sich in den Sprachen vernünftig ausdrücken zu können. Die Mitarbeiter sollten zudem noch genügend Kenntnisse in Latex haben, sowie SQL Statements erstellen können.

3.5.1 Tim Wiechers

Hardskills

- Gute Kenntnisse in Java.
- Sehr gute Deutsch-/ und Englischkenntnisse
- SQL-Statementerstellung
- solide L^AT_EX-Kenntnisse

Softskills

- Selbstorganisation
- Teamfähigkeit
- Hilfsbereitschaft

3.5.2 Tim Ellhoff

Hardskills

- Gute Kenntnisse in Java.
- Sehr gute Deutsch-/ und Englischkenntnisse

- SQL-Statementerstellung
- solide L^AT_EX-Kenntnisse
- Gute Kenntnisse im Grafikdesign

Softskills

- Selbstorganisation
- Teamfähigkeit
- Hilfsbereitschaft

3.5.3 Tobias Dellert

Hardskills

- Gute Kenntnisse in Java.
- Sehr gute Deutsch- und Englischkenntnisse
- SQL-Statementerstellung
- solide L^AT_EX-Kenntnisse
- Gute Kenntnisse im Grafikdesign

Softskills

- Selbstorganisation
- Teamfähigkeit
- Hilfsbereitschaft

3.5.4 Olga Miloevich

Hardskills

- Gute Kenntnisse in Java.
- Sehr gute Deutsch- und Englischkenntnisse
- SQL-Statementerstellung
- solide L^AT_EX-Kenntnisse
- Gute Kenntnisse im Grafikdesign

Softskills

- Selbstorganisation
- Teamfähigkeit
- Hilfsbereitschaft

3.5.5 Daniel Pupat

Hardskills

- Gute Kenntnisse in Java.
- Sehr gute Deutsch- / und Englischkenntnisse
- SQL-Statementerstellung
- solide L^AT_EX-Kenntnisse
- Gute Kenntnisse im Grafikdesign

Softskills

- Selbstorganisation
- Teamfähigkeit
- Hilfsbereitschaft

3.5.6 Patrick Hollatz

Hardskills

- Gute Kenntnisse in Java.
- Sehr gute Deutsch- / und Englischkenntnisse
- SQL-Statementerstellung
- solide L^AT_EX-Kenntnisse
- Gute Kenntnisse im Grafikdesign

Softskills

- Selbstorganisation
- Teamfähigkeit
- Hilfsbereitschaft

4 Technische Prozesse (Patrick Hollatz)

4.1 Methoden, Werkzeuge und Techniken

4.1.1 Entwicklungsplattform

Zur Erstellung und Verwaltung der einzelnen Dateien nutzen wir die IDE „Eclipse Kepler“ ⁴.

⁴<http://www.eclipse.org/>

UML-Diagramme erstellen wir mit „Dia“, Version 0.97.2⁵.

Die Dokumenten-Erstellung erfolgt mithilfe von Latex⁶.

Um von überall auf unsere Dateien zugreifen zu können, nutzen wir ein Git-Repository bei „GitHub“⁷.

Für die komplexe Zeitplanung erstellen wir mithilfe von „Microsoft Project“ ein Gantt-Diagramm⁸.

4.1.2 Entwicklungsmethode

Unsere Entwicklungsmethode ist das Wasserfallmodell, wie in [2.1](#) vorgestellt. Dabei wird nicht alles sequentiell abgearbeitet, sondern die Phasen Implementierung, Test und Dokumentation werden sich überschneiden.

4.1.3 Programmiersprache und Bibliotheken

Die Software wird mit Java auf Basis des „Java SE Development Kit (JDK) 7“ entwickelt⁹.

4.2 Dokumentationsplan

Wir werden als Ergebnis verschiedene Dokumentationen vorweisen können. Diese sind:

- Nutzerhandbuch
- Installationsanleitung
- Dokumentation des Quellcodes

4.2.1 Codingstyle

Unsere Implementierungen werden sich an die *Code Conventions for the Java Programming Language*¹⁰ halten.

Unsere .tex Dateien gehen aus der Vorlage für das SWP2-Projekt hervor. Es ist dafür zu sorgen, dass es jederzeit kompilierbar ist.

⁵<https://wiki.gnome.org/Apps/Dia>

⁶<http://www.latex-project.org/>

⁷<http://github.com/>

⁸<http://office.microsoft.com/de-de/projekt-und-portfolioverwaltung-microsoft-project-FX103472268.aspx>

⁹<http://www.java.com/>

¹⁰<http://www.oracle.com/technetwork/java/codeconv-138413.html>

4.2.2 Kommentarsprache

Die Kommentare werden in Englisch geschrieben, um Probleme mit Sonderzeichen zu vermeiden und das Verständnis des Source Codes, bei der Wartung oder Weiterentwicklung, auch nicht-deutschsprachigen zu erleichtern bzw. zu ermöglichen.

4.2.3 JavaDoc

JavaDoc-Kommentare werden für alle öffentlichen Klassen, Methoden und Variablen auf Englisch verfasst und dienen der Übersicht und schnellerem Verstehen der Zusammenhänge des Projekts. Private Klassen, Methoden und Variablen werden nur mit JavaDoc Kommentaren versehen, wenn diese komplex oder für das Verständnis wesentlich sind.

4.2.4 Begleitende Dokumentation

Begleitend werden eine Installationsanleitung für die Anwendung und den Server, ein Benutzerhandbuch zur Bedienung und eine Testprotokoll zum schnellen Überblick über durchgeführte Test und deren Ergebnis geschrieben.

4.3 Unterstützende Projektfunktionen

Alle erstellten Java-, SQL- und Latex-Dateien werden im Git Repository gespeichert. Dadurch haben wir nicht nur eine Datensicherung, sondern auch eine Versionskontrolle. Das Repository wird beim ersten Gruppentreffen erstellt und anschließend wird der Pfad zum Repository an dieser Stelle hinzugefügt. Das Repository liegt unter <https://github.com/timeathotmail/ReSWP.git>.

Zur Qualitätssicherung wird jedes Arbeitspaket nach Fertigstellung auf Erfüllung der Anforderungen überprüft, ob es unseren internen Richtlinien entspricht und wird gegebenenfalls noch einmal vom Qualitätsmanager für eine Überarbeitung empfohlen. Dafür sind nur wenige Arbeitsstunden vorgesehen, wenn der jeweils zuständige nicht selbst an den Arbeitsprodukten arbeitet.

5 Arbeitspakete, Zeitplan und Budget

bearbeitet von Tim Ellhoff, Tobias Dellert, Olga Miloevich

5.0.1 Annahmen

Wir gehen von folgenden Annahmen aus:

- jedes Mitglied der Gruppe hat rund 14,5 Stunden Zeit in der Woche fur das Projekt, davon ausgehend, dass die Veranstaltung SWP2 9CP bringt, was 270 Stunden an Zeit entspricht (1CP = 30 Stunden). Diesen Wert aufgesplittet in 19 Wochen ergibt rechnerisch 14,21 Stunden, also aufgerundet 14,5 Stunden pro Mitglied.
- eine Arbeitswoche hat 5 Tage, wobei jeder Arbeitstag 8 Arbeitsstunden hat.
- des Weiteren ist von folgenden Feiertagen und Urlaubszeiten auszugehen:
 - Tag der Arbeit: 01.05.2014
 - Himmelfahrt: 29.05.2014
 - Pfingstmontag: 09.06.2014

5.0.2 Anmerkungen

Hinweis: Wir haben die Punkte 'Arbeitspakete', 'Zeitplan und Abhangigkeiten' sowie 'Ressourcenanforderung' zu einem Unterpunkt zusammengefasst.

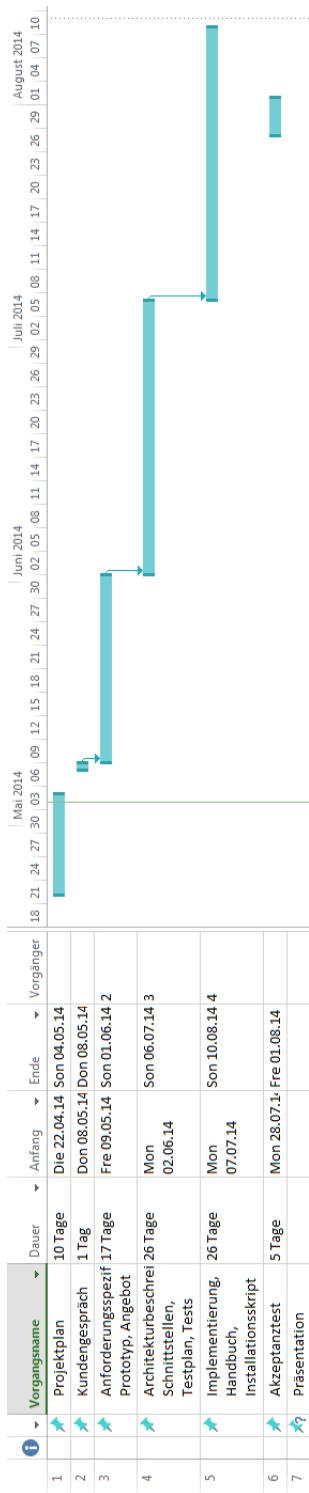
Wir haben bisher nur die Phasen 'Projektplan' und 'Anforderungsspezifikation' vollstandig in den Arbeitspaketen und Zuteilungen behandelt. Die ubrigen Phasen des Projekts - Entwurf, Implementierung und Test - konnen zu diesem Zeitpunkt noch nicht detailliert beschrieben werden, sondern erfolgen stattdessen in grobem Format.

Hinweis zum Tabellenlayout: Da wir in diesem Projektplan Inhalte aus anderen Projektplanen vorheriger SWP-Veranstaltungen z.T. ubernehmen (wie mit dem Veranstalter abgesprochen), unterscheidet sich die auere Form der Arbeitspaket-Tabellen manchmal.

5.1 Arbeitspakete, Zeitplan, Abhangigkeiten u. Ressourcenanforderungen

Im Folgenden liefert eine grafische Uberblick in Form eines Gantt-Diagramms die grundlegenden Arbeitspakete (Abbildung 2).

Abbildung 2: Überblick des gesamten Projekts in einem Gantt-Diagramm dargestellt



Hinweise zu den Begriffen in den Tabellen: Der Punkt 'Gesamtdauer' der jeweiligen Arbeitspakete ist der Zeitbereich zwischen dem Beginn und dem Ende einer bestimmten

Aktivität. Der 'Aufwand' ist die tatsächlich aufgewendete Zeit, in der auch gearbeitet wurde. Somit kann die 'Gesamtdauer' häufig höher ausfallen, wenn sich bestimmte Aktivitäten z.B. über Feiertage hinziehen. Mit der 'Abhängigkeit' ist gemeint, ob das Arbeitspaket von anderen abhängig ist, also im Prinzip einen Vorgänger hat. Mit den 'Ressourcen' sind stets Akteure unserer Gruppe gemeint. Der Punkt 'Mindestanforderungen' beschreibt gewissermaßen das Minimalziel des Arbeitspaketes (Minimalbedingung), die mindestens erfüllt werden muss, damit es fertig ist.

5.1.1 Projektplan

In den folgenden Tabellen sind die Arbeitspakete des Abschnitts 'Projektplan' dargestellt. Grafisch sind diese in einem weiteren Gantt-Diagramm realisiert (Abbildung 3).

Abbildung 3: Gantt-Projektplan

Folgender Abschnitt würde vom Projektplan von der Gruppe IT_R3V0LUT10N (Vorlesung Software Projekt 2, WiSe2013/14), teils übernommen und überarbeitet.

A R B E I T S P A K E T 1	
Bezeichnung	Gesamter Projektplan
Beschreibung:	Anfertigung und Komplettierung des initialen Projektplans. Details erfolgen in den jeweiligen Unterpunkten.
Hauptverantwortlicher	Tim Ellhoff
Abhängigkeit	-
Ressourcen	-
Aufwand, Gesamtdauer	
Beginn	22.04.2014
Ende	04.05.2014
Mindestanforderungen:	
Die Fertigstellung des initialen Projektplans ist erfolgt und wird durch das interne Gruppen-Review von allen Gruppenmitgliedern überprüft und abgesegnet (Qualitätssicherung).	

A R B E I T S P A K E T 1.1

Bezeichnung	Projektplanerstellung
Beschreibung: siehe Unterpunkte	
Hauptverantwortlicher	Tobias Dellert und Tim Ellhoff
Abhangigkeit	-
Ressourcen	-
Aufwand, Gesamtdauer	
Beginn	27.04.2014
Ende	03.05.2014

Mindestanforderungen:

Die Unterpunkte des Projektplans wurden abgearbeitet und der Projektplan kann im Gruppenreview uberpruft werden.

A R B E I T S P A K E T 1.1.1

Bezeichnung	Einleitung
Beschreibung:	
Erlledigung des Projektplanteils:	Die Projektubersicht liefert die Ziele, Hauptarbeitsaktivitaten und -produkte, die Hauptmeilensteine und einen groben Zeitplan, die Erfassung der Ressourcen, die zur Verfugung stehenden Budgets sowie die Kontakt- daten des Kunden und Informationen uber die Mitarbeiter am Projekt. Des Weiteren beinhaltet die Einleitung eine Uberblick der auszuliefernden Produkte mit Terminangaben , sowie Informationen uber die Evolution des Plans und die Festlegung von Definitionen bzw. Akronymen.
Hauptverantwortlicher	Daniel Pupat
Abhangigkeit	-
Ressourcen	<ul style="list-style-type: none">• Daniel Pupat
Aufwand, Gesamtdauer	
Beginn	25.04.2014
Ende	26.04.2014

Mindestanforderungen:

Sobald der Projektplanteil fertig ist, kann er zum Gruppenreview weitergegeben werden.

A R B E I T S P A K E T 1.1.2

Bezeichnung	Projektorganisation
Beschreibung:	<i>Erledigung des Projektplanteils:</i> In diesem Abschnitt wird das verwendete Prozessmodell beschrieben, die Organisationsstruktur festgelegt (insbes. Pflichten der Gruppenmitglieder, Kommunikationswege, wochentliche Treffen usw.), Informationen zu den Organisationsschnittstellen wie Auftraggeber und Arbeitgeber gegeben und letztlich die Verantwortlichkeiten im Projekt festgelegt und unter den einzelnen Gruppenmitgliedern aufgeteilt.
Hauptverantwortlicher	Olga Miloevich
Abhangigkeit	-
Ressourcen	<ul style="list-style-type: none"> • Olga Miloevich
Aufwand, Gesamtdauer	
Beginn	28.04.2014
Ende	30.04.2014

Mindestanforderungen:

Sobald der Projektplanteil fertig ist, kann er zum Gruppenreview weitergegeben werden.

A R B E I T S P A K E T 1.1.3

Bezeichnung	Managementprozess
Beschreibung:	
<i>Erledigung des Projektanteils:</i> Im Managementprozess erfolgt die Festlegung der Managementprozesse (insbes. Ziele des Projekts), sowie die Einhaltung von gewissen Regeln im Punkt 'Annahmen', die eine Vereinbarung der Abhangigkeiten, an denen das Projekt unmittelbar geknüpft ist und die Erfassung von Einschrankungen, die berucksichtigt werden mussen, mit einschliet. Es folgen Angaben zum Risikomanagement (insbes. Ausfall bzw. Austritt von Gruppenmitgliedern, allgemeine Probleme etc.) , Manahmen zur Projektuberwachung sowie die Fixierung der Kompetenzen der Mitarbeiter im Projekt.	
Hauptverantwortlicher	Olga Miloevich
Abhangigkeit	-
Ressourcen	<ul style="list-style-type: none">• Tim Ellhoff
Aufwand, Gesamtdauer	
Beginn	27.04.2014
Ende	29.04.2014
Mindestanforderungen: Sobald der Projektanteil fertig ist, kann er zum Gruppenreview weitergegeben werden.	

A R B E I T S P A K E T 1.1.4

Bezeichnung	Technische Prozesse
Beschreibung:	
<i>Erledigung des Projektplanteils:</i> In diesem Abschnitt erfolgt die Festlegung der Entwicklungsplattformen, -methoden und der verwendeten Programmiersprache, sowie eine Vereinbarung uber die Eckpfeiler der im Laufe des Projekts abzugebenden Dokumentationen. Unterstutzende Projektfunktionen wie die Ernennung von Phasenleitern und die Verwendung eines Versionskontrollsystems nebst Datensicherungen wird schlielich vereinbart.	
Hauptverantwortlicher	Patrick Hollatz
Abhangigkeit	-
Ressourcen	<ul style="list-style-type: none"> • Patrick Hollatz
Aufwand, Gesamtdauer	
Beginn	29.04.2014
Ende	29.04.2014

Mindestanforderungen:

Sobald der Projektplanteil fertig ist, kann er zum Gruppenreview weitergegeben werden.

A R B E I T S P A K E T 1.1.5

Bezeichnung	Arbeitspakete, Zeitplan u. Budget
Beschreibung:	
<i>Erledigung des Projektplanteils:</i> In diesem Punkt werden die Arbeitspakete aller Projektphasen erarbeitet. Zunachst folgen gewisse Annahmen und Anmerkungen sowie Begriffserklarungen fur die Arbeitspakete, bevor dann kleinschrittig die Arbeitspakete in tabellarischer und grafischer Form erfasst werden. Es folgen weitere Arbeitspakete fur die Dokumentabgaben, Informationen und Vereinbarungen uber Meetings innerhalb der Gruppe und der Festlegung des kritischen Pfads fr das Projekt.	
Hauptverantwortlicher	Tobias Dellert, Tim Ellhoff
Abhangigkeit	-
Ressourcen	<ul style="list-style-type: none">• Tobias Dellert• Tim Ellhoff• Olga Miloevich
Aufwand, Gesamtdauer	
Beginn	23.04.2014
Ende	28.04.2014
Mindestanforderungen: Sobald der Projektplanteil fertig ist, kann er zum Gruppenreview weitergegeben werden.	

A R B E I T S P A K E T 1.1.6

Bezeichnung	Sonstige Elemente
Beschreibung:	
<i>Erledigung des Projektplanteils:</i> Der Abschnitt 'Sonstiges' liefert Informationen uber die Plane fur die Konvertierung von Daten, Management-, Ausbildungs- und Raum- und Installationsplane sowie den Plan uber die letztendliche Abgabe bzw. Ubergabe des Systems.	
Hauptverantwortlicher	Olga Miloevich
Abhangigkeit	-
Ressourcen	<ul style="list-style-type: none"> • Tim Ellhoff
Aufwand, Gesamtdauer	
Beginn	02.05.2014
Ende	03.05.2014
Mindestanforderungen:	
Sobald der Projektplanteil fertig ist, kann er zum Gruppenreview weitergegeben werden.	

A R B E I T S P A K E T 1.2

Bezeichnung	Gruppenreview des Projektplans
Beschreibung: Im internen Gruppenreview wird der fertige Projektplan diskutiert, ggf. verbessert bzw. noch geandert.	
Hauptverantwortlicher	Tobias Dellert, Tim Ellhoff
Abhangigkeit	1 Projektplan, 1.1 Projektplanerstellung
Ressourcen	<ul style="list-style-type: none">• Tobias Dellert• Tim Ellhoff• Tim Wiechers• Patrick Hollatz• Daniel Pupat• Olga Miloevich
Aufwand, Gesamtdauer	
Beginn	03.05.2014
Ende	04.05.2014
Mindestanforderungen: Der Projektplan ist nun von allen Gruppenmitgliedern in einem Review noch einmal auf den Prufstand gestellt worden und ist nun fertig.	

5.1.2 Anforderungsspezifikation

In den folgenden Tabellen sind die Arbeitspakete des Abschnitts 'Anforderungsspezifikation' dargestellt. Grafisch sind diese in einem weiteren Gantt-Diagramm realisiert (Abbildung 4).

Folgender Abschnitt wurde vom Projektplan von der Gruppe YNotZoidberg (Vorlesung Software Projekt 2, WiSe2013/14, teils ubernommen und uberarbeitet.

5 ARBEITSPAKETE, ZEITPLAN UND BUDGET

5.1 Arbeitspakete, Zeitplan, Abhängigkeiten u. Ressourcenanforderungen Projektplan

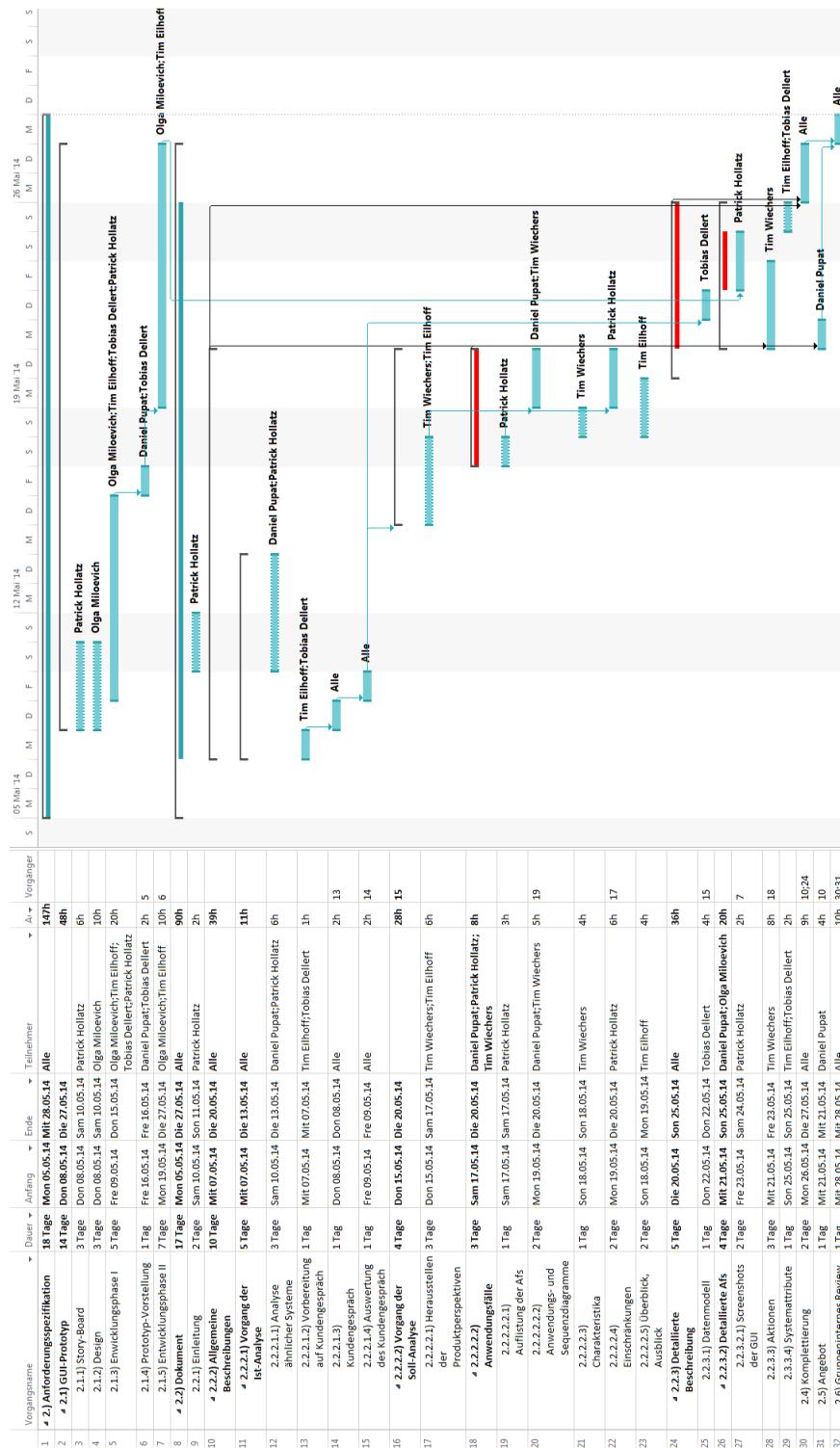


Abbildung 4: Anforderungsspezifikation im Gantt-Diagramm

Bearbeitet von: Olga Miloevich, Tobias Dellert, Tim Ellhoff, Tim Wiechers

AP Titel: 2. Anforderungsspezifikation	AP Nummer: 1
Dauer: 05.05.14- 29.05.14	Aufwand: 147 Std.
AP-Teilnehmer: alle Mitglieder	
Beschreibung: Anfertigung und Komplettierung der Anforderungsspezifikation. Details erfolgen in den jeweiligen Unterpunkten	
Voraussetzung: keine	
Ziele: Die Fertigstellung der Anforderungsspezifikation ist erfolgt.	

AP Titel: 2.0.1 Gruppentreffen	AP Nummer: 2
Dauer: 09.05.14-23.05.14	Aufwand: 9 Std.
AP-Teilnehmer: Alle Mitglieder	
Beschreibung: Dient als qualitätssichernde Maßnahme, indem Fortschritte und Unklarheiten in der Gruppe besprochen werden. Ein bis zweimal wöchentlich werden wir uns in der Universität treffen.	
Voraussetzung: keine	
Ziele: Jedes Gruppenmitglied ist auf den aktuellen Stand was Änderungen bzgl. des eigenen Arbeitsfeldes angeht.	

AP Titel: 2.1 GUI-Prototyp	AP Nummer: 3
Dauer: 08.05.14 - 25.05.14	Aufwand: 48 Std.
AP-Teilnehmer: Tobias Dellert, Olga Miloevich, Patrick Hollatz, Tim Ellhoff, Daniel Pupat	
Beschreibung: Dem Kundengespräch folgend, wird dem Kunden erstmal eine Benutzeroberfläche vorgestellt.	
Voraussetzung: stattgefundenes Kundengespräch	
Ziele: Die Grundfunktionen müssen anhand des Prototyps präsentierbar sein.	

AP Titel: 2.1.1 Story-Board	AP Nummer: 4
Dauer: 08.05.14 - 10.05.14	Aufwand: 6 Std.
AP-Teilnehmer: Patrick Hollatz	
Beschreibung:	
Voraussetzung: Kundengespräch	
Ziele: Mit dem Prototyp lassen sich die Leistungsfaktoren präsentieren.	

5 ARBEITSPAKETE, ZEITPLAN UND BUDGET

5.1 Arbeitspakete, Zeitplan, Abhangigkeiten u. Ressourcenanforderungen Projektplan

AP Titel: 2.1.2 Design	AP Nummer: 5
Dauer: 08.05.14- 10.05.15	Aufwand: 10 Std.
AP-Teilnehmer: Olga Miloevich	
Beschreibung:	
Voraussetzung: keine	
Ziele: Einzusetzendes Design.	

AP Titel: 2.1.3 Entwicklungsphase I	AP Nummer: 6
Dauer: 09.05.14 - 15.05.14	Aufwand: 20 Std.
AP-Teilnehmer: Olga Miloevich, Tim Ellhoff, Tobias Dellert	
Beschreibung: Entwicklung des GUI-Prototyps.	
Voraussetzung: Kundengesprach erfolgt.	
Ziele: Mit dem Prototyp lassen sich die Leistungsfaktoren präsentieren.	

AP Titel: 2.1.4 Prototyp-Vorstellung	AP Nummer: 7
Dauer: xx.05.14	Aufwand: 2 Std.
AP-Teilnehmer: Daniel Pupat, Tobias Dellert	
Beschreibung: Vorstellung des GUI-Prototyps fur den Kunden.	
Voraussetzung: Alle anderen Arbeitspakete zum GUI-Abschnitt.	
Ziele: Vorstellung erfolgt.	

AP Titel: 2.1.5 Entwicklungsphase II	AP Nummer: 8
Dauer: 18.05.14-24.05.14	Aufwand: 10 Std.
AP-Teilnehmer: Olga Miloevich, Tim Ellhoff	
Beschreibung:	
Voraussetzung:	
Ziele:	

AP Titel: 2.2 Anforderungsspezifikation Dokument	AP Nummer: 9
Dauer: 05.05.14 - 27.05.14	Aufwand: 90 Std.
AP-Teilnehmer: alle Mitglieder	
Beschreibung: Herstellung des Dokuments Anforderungsspezifikation.	
Voraussetzung: keine.	
Ziele: Der Ist-Zustand ist analysiert und dokumentiert.	

AP Titel: 2.2.1 Einleitung	AP Nummer: 10
Dauer: 10.05.14-11.05.14	Aufwand: 2 Std.
AP-Teilnehmer: Patrick Hollatz	
Beschreibung: Erledigung des Projektplanteils. Die Projektubersicht liefert die Ziele, Hauptarbeitsaktivitaten und -produkte, die Hauptmeilesteine und einen groben Zeitplan, die Erfassung der Ressourcen, die zur Verfugung stehenden Budgets sowie die Kontaktdaten des Kunden und Informationen uber die Mitarbeiter am Projekt. Des Weiteren beinhaltet die Einleitung eine ubersicht der auszuliefernden Produkte mit Terminangaben, sowie Informationen uber die Evaluation des Plans und die Festlegung von Definitionen bzw. Akronymen.	
Voraussetzung: keine.	
Ziele: Projektplanteil wird zum Gruppenreview weitergegeben.	

AP Titel: 2.2.2 Allgemeine Beschreibung	AP Nummer: 11
Dauer: 08.05.14-13.05.14	Aufwand: 39 Std.
AP-Teilnehmer: Tim Ellhoff, Tobias Dellert, Daniel Pupat, Patrick Hollatz	
Beschreibung: siehe Unterpunkte.	
Voraussetzung:	
Ziele:	

AP Titel: 2.2.2.1 Vorgang der Ist-Analyse	AP Nummer: 12
Dauer: 08.05.14-13.05.14	Aufwand: 11 Std.
AP-Teilnehmer: Tobias Dellert, Tim Ellhoff	
Beschreibung: siehe Unterpunkte.	
Voraussetzung: Kundengesprach erfolgt.	
Ziele: Die Unterpunkte des Kundengesprachs wurden abgearbeitet, somit ist der Ist-Zustand analysiert und dokumentiert.	

AP Titel: 2.2.2.1.1 Analyse ahnlicher Systeme	AP Nummer: 13
Dauer: 10.05.14-13.05.14	Aufwand: 6 Std.
AP-Teilnehmer: Patrick Hollatz, Daniel Pupat	
Beschreibung: Das ermittelte Gesamtbild wird mit ahnlichen bewahrten Software-systemen verglichen. Deren Eigenheiten und daraus resultierende Vor- und Nachteile werden erschlossen und die Systeme als Ganzes bewertet.	
Voraussetzung:	
Ziele: Analyse ahnlicher Systeme.	

AP Titel: 2.2.2.1.2 Vorbereitung auf Kundengesprach	AP Nummer: 14
Dauer: 07.05.14	Aufwand: 1 Std.

AP-Teilnehmer: Tim Ellhoff, Tobias Dellert	
Beschreibung: Um ein effektives, aufschlussreiches Interview zu gewährleisten, ist eine konkrete Vorstellung der zu stellenden Fragen von Nöten. Ziel ist es, nach Möglichkeit alle Anforderungen und Wünsche des Kunden, sowohl die bewussten, als auch die unbewussten, herauszustellen. Dafür wird hier ein Fragenkatalog erstellt, der auf Kunden mit beliebigem Wissenstand über technische Details anwendbar ist.	
Voraussetzung: keine.	
Ziele: Der entstehende Fragenkatalog muss weitestgehend alle Fragen enthalten, die für genaues Verständnis der Wünsche und Anforderungen des Kunden nötig sind.	
AP Titel: Kundengespräch	AP Nummer: 15
Dauer: 08.05.14	Aufwand: 2 Std.
AP-Teilnehmer: Tim Ellhoff, Tobias Dellert	
Beschreibung: Gespräch mit dem Kunden um dessen Wünsche und Erwartungen an das Produkt festzustellen.	
Voraussetzung: keine	
Ziele: Klarheit über die Anforderungen des Kunden verschaffen.	
AP Titel: 2.2.2.1.4 Auswertung des Kundengesprächs	AP Nummer: 16
Dauer: 08.05.14	Aufwand: 2 Std.
AP-Teilnehmer: alle Mitglieder	
Beschreibung: Die erhaltenen Auskünfte werden zusammengebracht und zu einem großen Bild aus Anforderungen zusammengefügt.	
Voraussetzung: Kundengespräch erfolgt.	
Ziele: Das erhaltene Gesamtbild enthält alle gegebenen Anforderungen, die nach dem Kundengespräch vom Kunden als Leistungsfaktoren angesehen werden.	
AP Titel: 2.2.2.2 Vorgang der Soll-Analyse	AP Nummer: 17
Dauer: 15.05.14 - 19.05.14	Aufwand: 28 Std.
AP-Teilnehmer: Daniel Pupat, Tim Ellhoff, Patrick Hollatz, Tim Wiechers	
Beschreibung:	
Voraussetzung: Kundengespräch erfolgt.	
Ziele:	
AP Titel: 2.2.2.2.1 Herausstellen der Produktperspektiven	AP Nummer: 18
Dauer: 15.05.14 - 17.05.14	Aufwand: 6 Std.

AP-Teilnehmer: Tim Ellhoff, Tim Wiechers

Beschreibung: Hier werden die gegebenen Rahmenbedienungen und Moglichkeiten analysiert, um die realistische Durchführung und dessen Aufwand zu erfassen. Beachtet und analysiert werden folgende Bereiche:

1. Systemschnittstellen
2. Benutzerschnittstellen
3. Hardwareschnittstellen
4. Softwareschnittstellen
5. Kommunikationsschnittstellen
6. Speicherbeschrankungen
7. Betriebsmodi
8. lokale Anpassungen

Voraussetzung: Siehe Oberpunkt.

Ziele: Die in der Beschreibung aufgezahlten Punkte konnen in der Anforderungsspezifikation eindeutig und exakt beschrieben werden.

AP Titel: 2.2.2.2 Anwendungsfalle **AP Nummer:** 19

Dauer: 17.05.14 - 19.05.14 **Aufwand:** 8 Std.

AP-Teilnehmer: Daniel Pupat, Patrick Hollatz, Tim Wiechers

Beschreibung: Siehe Unterpunkte.

Voraussetzung: Siehe Oberpunkt.

Ziele: Anwendungsfalle aller Grundfunktionen sind erfasst und kurz beschrieben.

AP Titel: 2.2.2.2.1 Auflistung der Anwendungsfalle **AP Nummer:** 20

Dauer: 17.05.14 **Aufwand:** 3 Std.

AP-Teilnehmer: Patrick Hollatz

Beschreibung: Die einzelnen Anwendungsfalle werden hier aufgelistet und kurz beschrieben.

Voraussetzung: Kundengesprach ausgewertet.

Ziele: Anwendungsfalle aller Grundfunktionen sind erfasst und kurz beschrieben.

AP Titel: 2.2.2.2 Anwendungs- und Sequenzdiagramme **AP Nummer:** 21

Dauer: 18.05.14 - 19.05.14 **Aufwand:** 5 Std.

AP-Teilnehmer: Tim Wiechers, Daniel Pupat
Beschreibung: Die Anwendungsfalle werden passenderweise in Anwendungsdia-gramme bertragen, wobei bei komplexeren Fallen zum Wohle der bersicht auch Sequenzdiagramme hinzugezogen werden knnen.
Voraussetzung: Anwendungsfalle ausgewertet.
Ziele: Die Anwendungsfalle der Grundfunktionen sind in einem Anwendungsfalldia-gramm dargestellt.

AP Titel: 2.2.2.2.3 Charakteristika	AP Nummer: 22
Dauer: 18.05.14	Aufwand: 4 Std.

AP-Teilnehmer: Tim Wiechers
Beschreibung: Sinn dieses Arbeitspakets ist, einen vollstndigen Einblick des fr uns relevanten Ausschnitts der Realitt zu erhalten, um damit in sich geschlossene Strukturen von den einzelnen Akteuren untereinander zu schaffen und damit einen verbesserten berblick der mglichen Anwendungsfalle zu bekommen. Hierzu werden Benutzerklassen erstellt, die so gut wie mglich die unterschiedlichsten Charaktere, Umstnde und Motive abdecken.
Voraussetzung:
Ziele: Die entstandenen Persona sind umfassend und untereinander verschieden ge-nug, um alle mglichen Anwendungsfalle in Bezug auf die erforderlichen Grundfunktio-nen der Software zu erhalten.

AP Titel: 2.2.2.2.4 Einschrnkungen	AP Nummer: 23
Dauer: 17.05.14 - 19.05.14	Aufwand: 6 Std.

AP-Teilnehmer: Patrick Hollatz
Beschreibung: Es werden sowohl technische, als auch gesetzliche Rahmenbedingun-gen festgelegt. Zustzlich sollen sicherheitskritische Aspekte durch genauere Spezifika-tionen der einzelnen Schnittstellen im vorigen Arbeitspaket 2.2.1 begutachtet werden.
Voraussetzung: Herausstellen der Produktperspektiven.
Ziele: Die technischen und gesetzlichen Rahmenbedingungen sind eindeutig spezifi-ziert. Alle sicherheitskritischen Aspekte wurden erfasst.

AP Titel: 2.2.2.2.5 Uberblick, sowie Ausblick in die Zukunft	AP Nummer: 24
Dauer: 18.05.14 - 19.05.14	Aufwand: 4 Std.

AP-Teilnehmer: Tim Ellhoff

Beschreibung: Uberblick der Abhangigkeiten von projekteigenen Faktoren untereinander, sowie ein Blick in die nahe Zukunft, was Veranderungen und Erweiterungen sowohl im technischen und rechtlichen Bereich, sowie im Anwendungsumfeld der zu entwickelnden Software angeht. Die Anforderungen sollen entsprechend verander- und anpassbar sein.

Voraussetzung:

Ziele: Die Abhangigkeit des Entwicklungs- und Erweiterungsprozess von einzelnen projekteigenen Faktoren wurden umfassend herausgestellt. Zu erwartende Veranderungen in der nahen Zukunft (sofern vorhanden) wurden dokumentiert.

AP Titel: 2.2.3 Detallierte Beschreibungen	AP Nummer: 25
---	----------------------

Dauer: 20.05.14 -25.05.14	Aufwand: 36 Std.
----------------------------------	-------------------------

AP-Teilnehmer: Alle Mitglieder

Beschreibung:

Voraussetzung: Bedingt. Siehe Unterpunkte.

Ziele: Hier findet der Ubergang von allgemeinen Anforderungen zu detaillierteren Anwendungen und Konzepten, die fur die Architekturbeschreibung wegweisend sein werden.

AP Titel: 2.2.3.1 Datenmodell	AP Nummer: 26
--------------------------------------	----------------------

Dauer: 22.05.14	Aufwand: 4 Std.
------------------------	------------------------

AP-Teilnehmer: Tobias Dellert

Beschreibung: Das zu entwickelnde Datenmodell soll eine konzeptionelle Darstellung von dem Weltausschnitt der Quiz-App, als UML-Diagramm, werden.

Voraussetzung: Kundengesprach, Gesprachs auswertung.

Ziele: Die Informationen und deren Beziehungen sind dem Weltausschnitt entsprechend in das Datenmodell gesetzt und es lsst sich somit als Konzept fur sptere Spezifizierungen benutzen.

AP Titel: 2.2.3.2 Detallierte Anwendungsfalle	AP Nummer: 27
---	----------------------

Dauer: 21.05.14 - 25.05.14	Aufwand: 20 Std.
-----------------------------------	-------------------------

AP-Teilnehmer: Daniel Pupat, Olga Miloevich

Beschreibung: Die bereits aufgelisteten Anwendungsfalle werden mit den Personas und genauerer Beschreibungen erganzt. Das Hineinversetzen in die unterschiedlichen Personas hilft die bisherige Funktionalitat aus der Sicht der Benutzer zu sehen.

Voraussetzung:

Ziele: Die Anwendungsfalle mussen insgesamt die ganze gewunschte Funktionalitat der Software wiedergeben.

AP Titel: 2.2.3.2.1 Screenshots der Benutzeroberfache	AP Nummer: 28
---	----------------------

Dauer: 23.05.14-24.05.14	Aufwand: 2 Std.
---------------------------------	------------------------

AP-Teilnehmer: Patrick Hollatz	
Beschreibung: Die Screenshots der Benutzeroberfache werden benutzt, um die spezifischen Anwendungsfalle zu illustrieren. Dazu werden die Screenshots an den passenden Stellen beschriftet.	
Voraussetzung: GUI-Prototyp ist fertig.	
Ziele:	
AP Titel: 2.2.3.3 Aktionen	AP Nummer: 29
Dauer: 20.05.14 - 23.05.14	Aufwand: 8 Std.
AP-Teilnehmer: Tim Wiechers	
Beschreibung: Die bisher beschriebenen Anwendungsfalle werden noch genauer ausgefrt, sodass jede aufgelistete Aktion eine einzelne Nutzung einer beliebigen Funktion(Knopfdruck, Klick ins Textfeld) der Software beschreibt	
Voraussetzung: Anwendungsfalle	
Ziele: Alle Anwendungsfalle sind in die kleinstmoglichen Aktionen aufgeteilt und geordnet.	
AP Titel: 2.2.3.4 Systemattribute	AP Nummer: 30
Dauer: 25.05.14	Aufwand: 2 Std.
AP-Teilnehmer: Tobias Dellert, Tim Ellhoff	
Beschreibung:	
Voraussetzung:	
Ziele: Die Informationen und deren Beziehungen sind dem Weltausschnitt entsprechend in das Datenmodell gesetzt und es lsst sich somit als Konzept fr sptere Spezifizierungen benutzen.	
AP Titel: 2.4 Anforderungsspezifikation komplettieren	AP Nummer: 31
Dauer: 25.05.14-26.05.14	Aufwand: 9 Std.
AP-Teilnehmer: Alle Mitglieder	
Beschreibung: Manahme, um Ergebnisse zusammenzufassen und zu korrigieren. Zusätzlich wird noch besonders Wert auf langfristige Schwachstellen des momentanen Ist-Zustand gelegt.	
Voraussetzung: Allgemeine Beschreibungen, Detaillierte Beschreibungen.	
Ziele: Sobald der Projektanteil fertig ist, kann er zum Gruppenreview weitergegeben werden.	
AP Titel: 2.5 Angebot fr den Kunden	AP Nummer: 32
Dauer: 27.05.14	Aufwand: 4 Std.
AP-Teilnehmer: Daniel Pupat	
Beschreibung: Angebot fr den Kunden inkl. Kostenaufwand wird erstellt.	
Voraussetzung: Allgemeine Beschreibungen.	
Ziele: Das Angebot fr den Kunden ist hinreichend erstellt und kann nun dem Gruppenreview bergeben werden.	

AP Titel: 2.6 Gruppeninternes Review u. Finalisierung Anforderungsspezifikation	AP Nummer: 33
Dauer: 28.05.14	Aufwand: 10 Std.
AP-Teilnehmer: alle Mitglieder	
Beschreibung: Das Dokument wird in der Gruppe auf den Prifstand gestellt und ggf. noch verandert bzw. verbessert.	
Voraussetzung: 2.5 Anforderungsspezifikation komplettieren, 2.6 Angebot.	
Ziele: Nach der Diskussion in der Gruppe wurde das Dokument optimiert und fertiggestellt.	

5.1.3 Architekturbeschreibung, Schnittstellenbeschreibung, Testplan und Blackbox-Tests

Folgender Abschnitt wurde vom Projektplan von der Gruppe YNotZoidberg (Vorlesung Software Projekt 2, WiSe2013/14), teils ubernommen und uberarbeitet.

AP Titel: 3. Architekturbeschreibung, Schnittstellenbeschreibung, Testplan und Blackbox-Tests	AP Nummer: 1
Dauer: 02.06.14 - 30.06.14	Aufwand: 150 Std.
AP-Teilnehmer: alle	
Beschreibung: Hier werden zur besseren Implementierbarkeit die zusammengestellten Konzepte der Anforderungsspezifikation in technischer Form umgestaltet und denkbare Probleme und Schwierigkeiten technischer und personeller Art entsprechende Losungsstrategien zugeordnet.	
Voraussetzung: Anforderungsspezifikation	
Ziele: Es soll insgesamt eine solide und anwendbare Architektur, was Gruppenmanagement und Implementierungsmodelle angeht, fur unser Projekt entstehen.	

AP Titel: 3.1 Globale Analyse	AP Nummer: 2
Dauer: 04.06.14 - 08.06.14	Aufwand: 24 Std.
AP-Teilnehmer: Olga Miloevich, Daniel Pupat, Tobias Dellert	
Beschreibung: siehe Unterpunkte	
Voraussetzung: Systemattribute der AS	
Ziele: Eine umfassende Analyse denkbarer Einflussfaktoren und ein ausreichender Vorrat an entsprechenden Losungsstrategien	

AP Titel: 3.1.1 Einflussfaktoren	AP Nummer: 3
Dauer: 04.06.14 - 05.06.14	Aufwand: 6 Std.
AP-Teilnehmer: Olga Miloevich, Tobias Dellert, Daniel Pupat	
Beschreibung: Hier werden die gegebenen Umstände in technischer und personeller Hinsicht auf Abhängigkeiten untersucht, um so mögliche Einflussfaktoren herauszuarbeiten.	
Voraussetzung: Siehe Oberpaket	
Ziele: Alle Einflussfaktoren, die das Bearbeiten eines Arbeitspaketes stark einschränkt oder komplett behindern, sind herausgearbeitet.	

AP Titel: 3.1.2 Problem- und Strategiekarten	AP Nummer: 4
Dauer: 05.06.14 - 08.06.14	Aufwand: 18 Std.
AP-Teilnehmer: Olga Miloevich, Tobias Dellert, Daniel Pupat	
Beschreibung: Zu den Einflussfaktoren werden Lösungsstrategien entwickelt.	
Voraussetzung: Siehe Oberpaket	
Ziele: Zu jedem Einflussfaktor sollen mindestens zwei Lösungsstrategien bereitgestellt werden	

AP Titel: 3.2 Konzeptionelle Sicht	AP Nummer: 5
Dauer: 04.06.14 - 04.06.14	Aufwand: 15 Std.
AP-Teilnehmer: alle	
Beschreibung: Ein UML-Diagramm, welches auf hoher Abstraktionsebene die Struktur unseres zu implementierendes System darstellt.	
Voraussetzung: Anforderungsspezifikation	
Ziele: Das Diagramm ist in nicht mehr zusammenfassbare Module bezüglich ihrer Aufgaben aufgeteilt	

AP Titel: 3.3 Modulsicht	AP Nummer: 6
Dauer: 08.06.14 - 20.06.14	Aufwand: 60 Std.
AP-Teilnehmer: Tim Wiechers, Patrick Hollatz, Tim Ellhoff	
Beschreibung: Eine detailliertere statische Form der konzeptionellen Sicht, indem dessen grobe Module in Teilmodule unterteilt werden, um ein Arbeitspaket im Ausmaß von höchstens einer Arbeitswoche eines Gruppenmitglieds darzustellen. Zudem werden Schnittstellen zwischen den Teilmodulen erstellt.	
Voraussetzung: Konzeptionelle Sicht, (Globale Analyse)	
Ziele: Die Module sind detailliert genug, um als Anleitung für die Programmierung zu dienen.	

AP Titel: 3.4 Datensicht	AP Nummer: 7
Dauer: 20.06.14 - 21.06.14	Aufwand: 12 Std.
AP-Teilnehmer: Tobias Dellert	
Beschreibung: Das anwendungs- und vorgangsaufzeigende Datenmodell wird bezüglich implementierungsspezifischer Details erweitert.	
Voraussetzung: Datenmodell	
Ziele: Die Datensicht stellt auf praktische Weise den Übergang vom Anwendungskonzept zur tatsächlichen Implementierungsstruktur dar.	

AP Titel: 3.5 Ausführungssicht	AP Nummer: 8
Dauer: 20.06.14 - 22.06.14	Aufwand: 12 Std.
AP-Teilnehmer: Daniel Pupat	
Beschreibung: Stellt den zeitlichen Verlauf unseres Systems während der Anwendung dar.	
Voraussetzung: Modulsicht	
Ziele: Die Ausführungssicht spezifiziert den Implementierungsvorgang, indem es deutlich einzuhaltende Reihenfolgen miteinander kommunizierender Module aufzeigt.	

AP Titel: 3.6 Zusammenhänge zw. Anwendungsfälle und Architektur	AP Nummer: 9
Dauer: 22.06.14 - 24.06.14	Aufwand: 30 Std.
AP-Teilnehmer: Tobias Dellert, Olga Miloevich, Tim Wiechers	
Beschreibung: Die Anwendungsfälle werden im bisherigen System in Form von Sequenzdiagrammen dargestellt, um die Anwendbarkeit unseres Systems an unserem Projekt zusätzlich zu sichern.	
Voraussetzung: Detaillierte Anwendungsfälle, Modulsicht, (Ausführungssicht)	
Ziele: Das Funktionieren des Zusammenspiels aller Module in mindestens einem Anwendungsfall muss aufgezeigt sein. Es gibt keine Fehler in der Modulstruktur.	

AP Titel: 3.7 Evolution	AP Nummer: 10
Dauer: 26.06.14 - 27.06.14	Aufwand: 16 Std.
AP-Teilnehmer: Patrick Hollatz, Tim Ellhoff	
Beschreibung: Hier werden Änderungen in unserem System, bei sich ändernden Rahmenbedingungen und Anforderungen aufgezeigt.	
Voraussetzung: Modulsicht	
Ziele: Die genannten Punkte aus dem Abschnitt Ausblick der Anforderungsspezifikation werden mit unserem bisherigen System konfrontiert. Am Ende soll sich das System als sehr gut Wart- und Änderbar herausstellen.	

AP Titel: 3.8 Einführung: Zweck/Status/Definitionen/Referenzen/Übersicht	AP Nummer: 11
Dauer: 28.06.14 - 30.06.14	Aufwand: 6 Std.
AP-Teilnehmer: Tobias Dellert, Tim Ellhoff	
Beschreibung: Eine Zusammenfassung des Zwecks der Architekturbeschreibung, den Status unseres Projekts und kleinerer Unterstützungen beim Zurechtfinden in diesem Dokument.	
Voraussetzung: Alle dem Architekturbeschreibungsdokument zugehorigen Pakete	
Ziele: Fur eine nicht in das Projekt eingeweihte Person soll es moglich sein, unsere Vorhaben mit den einzelnen Arbeitsschritten nachzuvollziehen.	

AP Titel: Testplan	AP Nummer: 12
Dauer: 24.06.14 - 28.06.14	Aufwand: 24 Std.
AP-Teilnehmer: Olga Miloevich, Tobias Dellert, Patrick Hollatz	
Beschreibung: Erstellung eines Testplans zur besseren Verifizierung der Funktionalitaten unseres Systems.	
Voraussetzung: Modulsicht, (Ausführungssicht)	
Ziele: Die wichtigsten Funktionen des Systems und typische Fehlerquellen sind im Testplan enthalten.	

AP Titel: Blackbox-Tests	AP Nummer: 13
Dauer: 27.06.14 - 02.07.14	Aufwand: 30 Std.
AP-Teilnehmer: Olga Miloevich, Daniel Pupat, Tobias Dellert	
Beschreibung: Die Erfullung der spezifizierten Anforderungen unseres Systems, ohne Beachtung der Implementierungsspezifischen Details einzelner Module.	
Voraussetzung: (Testplan), Modulsicht	
Ziele: Bei allen fur einen Anwendungsvorgang verwendeten Schnittstellen stehen BlackBox-Tests bereit.	

5.1.4 Implementierung

Folgender Abschnitt wurde vom Projektplan von der Gruppe YNotZoidberg (Vorlesung Software Projekt 2, WiSe2013/14), teils ubernommen und uberarbeitet.

Hinweis: Der folgende Abschnitt sowie die darin enthaltenen Arbeitspakete sind zu diesem Zeitpunkt des Projekts noch nicht vollstandig. Es handelt sich vielmehr um eine grobe Planung.

Die wahrend der Implementierungsphase zu bewaltigenden Arbeitspakete sind zu diesem Zeitpunkt noch nicht ersichtlich, bzw. dessen Zuordnungen von Bearbeitern, weswegen hier auf Paket- und Arbeitseinteilung weitestgehend verzichtet wird. Was jedoch bereits feststehen soll, ist das regelmige Treffen als Gruppe.

AP Titel: 4. Implementierungsvorgang	AP Nummer: 1
Dauer: 09.07.14 - 08.08.14	Aufwand: 516 Std.
AP-Teilnehmer: alle	
Beschreibung: Implementierung unserer Architektur	
Voraussetzung: Architekturbeschreibung, Schnittstellenbeschreibung, Testplan und Blackbox-Tests	
Ziele: Fertig Implementiert	
AP Titel: 4.1 Gruppentreffen	AP Nummer: 2
Dauer: 09.07.14 - 08.08.14	Aufwand: 96 Std.
AP-Teilnehmer: alle	
Beschreibung: Qualitatssichernde Manahme zur Gruppenkommunikation	
Voraussetzung:	
Ziele: Jedes Gruppenmitglied hat die Gelegenheit Fragen und Unklarheiten umfassend in der Gruppe zu klaren.	
AP Titel: 4.2 Implementierung	AP Nummer: 3
Dauer: 09.07.14 - 09.08.14	Aufwand: 420 Std.
AP-Teilnehmer: alle	
Beschreibung: Implementieren des Projekts	
Voraussetzung: Architekturbeschreibung, Schnittstellenbeschreibung, Testplan und Blackbox-Tests	
Ziele: Die Anforderungen des Kunden wurden erfüllt.	
AP Titel: 4.3 Whitboxtests	AP Nummer: 4
Dauer: 29.07.14 - 08.08.14	Aufwand: 120 Std.
AP-Teilnehmer: Alle	
Beschreibung: Hier wird nicht nur das Ergebnis einer Anfrage auf Ubereinstimmung der spezifizierten Anforderungen betrachtet, sondern auch die Vorgange in den einzelnen Modulen.	
Voraussetzung: Architekturbeschreibung, Schnittstellenbeschreibung, Testplan und Blackbox-Tests	
Ziele: Die Funktionalitat der Modulstruktur ist auch bei wiederholten Gebrauch gegeben.	

AP Titel: 4.4 Installationshandbuch anlegen	AP Nummer: 5
Dauer: 07.08.14 - 08.08.14	Aufwand: 20 Std.
AP-Teilnehmer: Tim Ellhoff, Daniel Pupat	
Beschreibung: Erstellung eines umfassenden Handbuchs zur Installation unseres Systems.	
Voraussetzung: Architekturbeschreibung, Schnittstellenbeschreibung, Testplan und Blackbox-Tests	
Ziele: Auch nicht-technisch versierte Leute konnen den Anweisungen folgen und so das System erfolgreich installieren.	

5.1.5 Dokumentabgaben

Folgender Abschnitt wurde vom Projektplan von der Gruppe IT-R3V0LUT10N (Vorlesung Software Projekt 2, WiSe2013/14, teils ubernommen und uberarbeitet).

In den folgenden Tabellen sind die Arbeitspakete des Abschnitts 'Dokumentabgaben' dargestellt. Grafisch sind diese in einem weiteren Gantt-Diagramm realisiert.

A R B E I T S P A K E T 5

Bezeichnung	Dokumentabgaben
Beschreibung:	
Die Abgaben beinhalten die regularen Abgaben in SWP2 genauso wie die jeweiligen Meilensteine	
Hauptverantwortlicher	Phasenleiter
Abhangigkeit	-
Ressourcen	-
Aufwand, Gesamtdauer	
Beginn	22.04.2014
Ende	10.08.2014

Mindestanforderungen:

siehe Unterpunkte (vgl. auch Meilensteine in Abschnitt 1.1.3)

A R B E I T S P A K E T 5.1

Bezeichnung	Dokument in Gruppe finalisiert (Gesamter Projektplan)
Beschreibung: siehe Abschnitt 1.1.3	
Hauptverantwortlicher	Tobias Dellert
Abhangigkeit	1.2 Gruppenreview des Projektplans
Ressourcen	-
Aufwand, Gesamtdauer	
Beginn	03.05.2014
Ende	03.05.2014

Mindestanforderungen:

Projektplan fur Abgabe bereit.

A R B E I T S P A K E T 5.2

Bezeichnung	Gesamter Projektplan
Beschreibung: siehe Abschnitt 1.1.3	
Hauptverantwortlicher	Tobias Dellert
Abhangigkeit	Dokument in Gruppe finalisiert (Gesamter Projektplan)
Ressourcen	-
Aufwand, Gesamtdauer	
Beginn	04.05.2014
Ende	04.05.2014

Mindestanforderungen:

Projektplan vor der Deadline in MEMS hochgeladen.

A R B E I T S P A K E T 5.3

Bezeichnung	Dokument in Gruppe finalisiert (Gesamte Anforderungsspezifikation)
Beschreibung: siehe Abschnitt 1.1.3	
Hauptverantwortlicher	Daniel Pupat
Abhangigkeit	Gruppeninternes Review u. Finalisierung Anforderungsspezifikation Gruppeninternes Review u. Finalisierung Angebot
Ressourcen	-
Aufwand, Gesamtdauer	
Beginn	30.05.2014
Ende	30.05.2014
Mindestanforderungen: siehe Abschnitt 1.1.3	

A R B E I T S P A K E T 5.4

Bezeichnung	Gesamte Anforderungsspezifikation
Beschreibung: siehe Abschnitt 1.1.3	
Hauptverantwortlicher	Daniel Pupat
Abhangigkeit	Dokument in Gruppe finalisiert (Gesamte Anforderungsspezifikation)
Ressourcen	-
Aufwand, Gesamtdauer	
Beginn	01.06.2014
Ende	01.06.2014
Mindestanforderungen: siehe Abschnitt 1.1.3	

A R B E I T S P A K E T 5.5

Bezeichnung	Dokument in Gruppe finalisiert (Gesamte Architekturbeschreibung, Testplan und Schnittstellentests (JUnit))
--------------------	--

Beschreibung:

siehe Abschnitt 1.1.3

Hauptverantwortlicher	Olga Miloevich
Abhangigkeit	Gesamte Anforderungsspezifikation
Ressourcen	-
Aufwand, Gesamtdauer	
Beginn	04.07.2014
Ende	04.07.2014

Mindestanforderungen:

siehe Abschnitt 1.1.3

A R B E I T S P A K E T 5.6

Bezeichnung	Gesamte Architekturbeschreibung, Testplan und Schnittstellentests (JUnit)
--------------------	---

Beschreibung:

siehe Abschnitt 1.1.3

Hauptverantwortlicher	Olga Miloevich
Abhangigkeit	Dokument in Gruppe finalisiert (Gesamte Architekturbeschreibung, Testplan und Schnittstellentests (JUnit))
Ressourcen	-
Aufwand, Gesamtdauer	
Beginn	06.07.2014
Ende	06.07.2014

Mindestanforderungen:

siehe Abschnitt 1.1.3

A R B E I T S P A K E T 5.7

Bezeichnung	Dokument in Gruppe finalisiert (Vollstandige Abgabe)
Beschreibung: siehe Abschnitt 1.1.3	
Hauptverantwortlicher	Tobias Dellert
Abhangigkeit	Gesamte Architekturbeschreibung Implementierung Test Installationspaket Benutzerdokumentation Installationshandbuch
Ressourcen	-
Aufwand, Gesamtdauer	
Beginn	09.08.2014
Ende	09.08.2014
Mindestanforderungen: siehe Abschnitt 1.1.3	

A R B E I T S P A K E T 5.8

Bezeichnung	Vollstandige Abgabe
Beschreibung: siehe Abschnitt 1.1.3	
Hauptverantwortlicher	Tobias Dellert
Abhangigkeit	Dokument in Gruppe finalisiert (Vollstandige Abgabe)
Ressourcen	-
Aufwand, Gesamtdauer	
Beginn	10.08.2014
Ende	10.08.2014
Mindestanforderungen: siehe Abschnitt 1.1.3	

5.1.6 Sonstiges

Folgender Abschnitt wurde vom Projektplan von der Gruppe IT_R3VOLUT10N (Vorlesung Software Projekt 2, WiSe2013/14), teils ubernommen und uberarbeitet.

In den folgenden Tabellen sind die Arbeitspakete des Abschnitts 'Sonstiges' dargestellt.

Grafisch sind diese in einem weiteren Gantt-Diagramm realisiert.

A R B E I T S P A K E T 6.1

Bezeichnung	Projektbeginn
Beschreibung:	
Hauptverantwortlicher	-
Abhangigkeit	-
Ressourcen	-
Aufwand, Gesamtdauer	
Beginn	22.04.2014
Ende	22.04.2014

Mindestanforderungen: -

A R B E I T S P A K E T 6.2

Bezeichnung	Angebotsannahme
Beschreibung:	
Angebotsannahme durch den Kunden	
Hauptverantwortlicher	-
Abhangigkeit	-
Ressourcen	-
Aufwand, Gesamtdauer	
Beginn	01.06.2014
Ende	01.06.2014

Mindestanforderungen: -

A R B E I T S P A K E T 6.3

Bezeichnung	Mindestanforderungen SWP2
Beschreibung:	
-	
Hauptverantwortlicher	-
Abhangigkeit	-
Ressourcen	-
Aufwand, Gesamtdauer	
Beginn	22.04.2014
Ende	22.04.2014

Mindestanforderungen: Die Mindestanforderungen fur SWP2 werden auf StudIP veröffentlicht.

A R B E I T S P A K E T 6.4

Bezeichnung	Installationspaket
Beschreibung:	
-	
Hauptverantwortlicher	Tobias Dellert
Abhangigkeit	4 Implementierung
Ressourcen	<ul style="list-style-type: none"> • Tobias Dellert • Daniel Pupat • Tim Ellhoff
Aufwand, Gesamtdauer	
Beginn	02.08.2014
Ende	06.08.2014

Mindestanforderungen: -

A R B E I T S P A K E T 6.5

Bezeichnung	Benutzerdokumentation
Beschreibung: -	
Hauptverantwortlicher	Patrick Hollatz
Abhangigkeit	3 Gesamte Architekturbeschreibung
Ressourcen	<ul style="list-style-type: none">• Olga Miloevich• Tim Wiechers• Patrick Hollatz
Aufwand, Gesamtdauer	
Beginn	20.07.2014
Ende	25.07.2014
Mindestanforderungen: -	

A R B E I T S P A K E T 6.6

Bezeichnung	Installationshandbuch
Beschreibung: -	
Hauptverantwortlicher	Tim Ellhoff
Abhangigkeit	Installationspaket
Ressourcen	<ul style="list-style-type: none">• Tobias Dellert
Aufwand, Gesamtdauer	
Beginn	07.08.2014
Ende	09.08.2014
Mindestanforderungen: -	

A R B E I T S P A K E T 6.7

Bezeichnung	Vorbereitung Abschlusspräsentation
Beschreibung:	
Hauptverantwortlicher	-
Abhängigkeit	-
Ressourcen	-
Aufwand, Gesamtdauer	
Beginn	08.08.2014
Ende	08.08.2014
Mindestanforderungen: -	

A R B E I T S P A K E T 6.8

Bezeichnung	Abschlusspräsentation
Beschreibung:	
Hauptverantwortlicher	-
Abhängigkeit	-
Ressourcen	-
Aufwand, Gesamtdauer	
Beginn	nach Vereinbarung
Ende	nach Vereinbarung
Mindestanforderungen: -	

5.1.7 Meetings

Folgender Abschnitt würde vom Projektplan von der Gruppe IT-R3V0LUT10N (Vorlesung Software Projekt 2, WiSe2013/14), teils übernommen und überarbeitet.

Im Folgenden sind die Aktivitäten des Abschnitts 'Meetings' als ein Arbeitspaket dargestellt. Auf ein Gantt-Diagramm wird an dieser Stelle verzichtet.

Exemplarisch haben stellen wir ein verallgemeinertes Arbeitspaket für unsere Meetings dar:

A R B E I T S P A K E T 7

Bezeichnung	Meetings
Beschreibung: siehe Abschnitt ??	
Hauptverantwortlicher	alle Gruppenmitglieder
Abhängigkeit	-
Ressourcen	alle Gruppenmitglieder
Aufwand, Gesamtdauer	variabel
Beginn	-
Ende	-

Mindestanforderungen: Probleme müssen gelöst, Aufgaben verteilt und Unklarheiten beseitigt worden sein.

6 Sonstige Elemente (Olga Miloevich)

Folgender Abschnitt wurde vom Projektplan von der Gruppe YNotZoidberg (Vorlesung Software Projekt 2, WiSe2013/14), übernommen.

6.1 Pläne für die Konvertierung von Daten

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

6.2 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.

6.3 Ausbildungspläne

Wir behaupten, unsere Kenntnisse reichen aus, um ein laufendes Projekt realisieren zu können. Es wird auf jeden Fall im Laufe des Semesters Kenntnisse durch Lernprozesse verbessert. Falls es nötig wird, setzen sich die Gruppenmitglieder intensiv mit Literatur und Ausbildungsvideokursen auseinander.

6.4 Raumpläne

Wir haben geplant, uns regelmäßig, ein Mal pro Woche, zu einem Gruppenmeeting zu treffen. Als Standardtreffpunkt wird uns der Mac-Raum im MZH E0 dienen. Treffzeiten werden erst in der nächsten Woche festgelegt, wenn alle Tutoriumstermine bekannt gegeben werden. Auf diesen Meetings werden Treffen zu inhaltlichen Arbeiten festgelegt.

6.5 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. MS Project (zur Erstellung von Gantt-Diagrammen)

Außerdem muss jedes Gruppenmitglied die Software zur Versionenverwaltung (Git-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.

6.6 Pläne für die Übergabe des Systems

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