

Software–Projekt 2 2014

VAK 03-BA-901.02

Projektplan

ChronoX

| | | |
|-------------------|-----------------|---------------|
| Alexander But | abut@tzi.de | 4000486 |
| Tobias Dellert | tode@tzi.de | 2936941 |
| Karsten Betjemann | mortem34@tzi.de | MAT_NR |
| (Tim Ellhoff | tellhof@tzi.de |) |

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 0 | Über uns | 4 |
| 1 | Einleitung | 5 |
| 1.1 | Projektübersicht | 5 |
| 1.1.1 | Ziele | 5 |
| 1.1.2 | Hauptarbeitsaktivitäten und -produkte | 5 |
| 1.1.3 | Haupt-Meilensteine und grober Zeitplan | 6 |
| 1.2 | Auszuliefernde Produkte | 6 |
| 1.3 | Referenzen | 7 |
| 1.4 | Definitionen und Akronyme | 7 |
| 2 | Projektorganisation | 7 |
| 2.1 | Prozessmodell | 7 |
| 2.2 | Organisationsstruktur | 8 |
| 2.3 | Organisationsgrenzen und -schnittstellen | 8 |
| 2.4 | Verantwortlichkeiten | 8 |
| 3 | Managementprozess | 8 |
| 3.1 | Managementprozess und -prioritäten | 8 |
| 3.2 | Annahmen, Abhängigkeiten und Einschränkungen | 8 |
| 3.3 | Risikomanagement | 8 |
| 3.3.1 | (Mögliche) Risiken | 9 |
| 3.4 | Projektüberwachung | 13 |
| 3.5 | Mitarbeiter | 14 |
| 4 | Technische Prozesse | 14 |
| 4.1 | Methoden, Werkzeuge und Techniken | 14 |
| 4.1.1 | Entwicklungsplattform | 14 |
| 4.1.2 | Entwicklungsmethode | 14 |
| 4.1.3 | Programmiersprache und Bibliotheken | 14 |
| 4.2 | Dokumentationsplan | 14 |
| 4.2.1 | Codingstyle | 15 |
| 4.2.2 | Kommentarsprache | 15 |
| 4.2.3 | JavaDoc | 15 |
| 4.2.4 | Begleitende Dokumentation | 15 |
| 4.3 | Unterstützende Projektfunktionen | 15 |
| 5 | Arbeitspakete, Zeitplan und Budget | 15 |
| 5.1 | Arbeitspakete | 15 |
| 5.2 | Zeitplan und Abhängigkeiten | 32 |
| 5.3 | Ressourcenanforderung | 32 |

| | | |
|----------|---|-----------|
| 6 | Sonstige Elemente | 32 |
| 6.1 | Pläne für die Konvertierung von Daten | 32 |
| 6.2 | Managementpläne für Unterauftragsnehmer | 32 |
| 6.3 | Ausbildungspläne | 33 |
| 6.4 | Raumpläne | 33 |
| 6.5 | Installationspläne | 33 |
| 6.6 | Pläne für die Übergabe des Systems | 33 |
| 6.7 | Beschaffungspläne für Hardware | 33 |

Version und Änderungsgeschichte

Die aktuelle Versionsnummer des Dokumentes sollte eindeutig und gut zu identifizieren sein, hier und optimalerweise auf dem Titelblatt.

| Version | Datum | Änderungen |
|---------|------------|---|
| 1.0 | 23.05.2014 | Erste veröffentlichte Version. |
| 1.1 | 24.05.2014 | Arbeitspakete, Zeitplan und Budget hinzugefügt. |
| 1.2 | 25.05.2014 | Risikomanagement hinzugefügt. |
| 1.3 | 25.05.2014 | Einleitung hinzugefügt. |

0 Über uns

1 Einleitung

1.1 Projektübersicht

1.1.1 Ziele

Im Rahmen des von unserer Gruppe angepeilten Gesamtzieles, nämlich des Bestehens vom Modul Softwareprojekt dieses Semesters, beabsichtigen wir im Bezug auf dieses Projekt sämtliche grundlegenden Kriterien zu erfüllen. Jene Kriterien beinhalten die fristgerechte Abgabe sämtlicher notwendiger Dokumente des Übungsbetriebes wie auch die Umsetzung aller Aufgaben des Miniprojektes SWP, womit unter anderem ein Projektplan, die konkrete Anforderungsspezifikation sowie die Architektur, der Testplan und der Blackbox Test der zu erarbeitenden Software eingeschlossen sind.

Der hauptsächliche Fokus unserer Arbeit wird sich dabei aufgrund des allgemeinen Aufwandes und der Bedeutung innerhalb des Moduls dem Miniprojekt zuwenden. Hierbei beabsichtigen wir sämtliche zuvor gelisteten Mindestanforderungen für das zu entwickelnde Stundenplan-System unter Berücksichtigung aller bekanntgegebenen technischen Randbedingungen zu erfüllen und somit ein ansprechendes System zur Planung der Stundenpläne von Lehrern wie Schülern mit etwaigen Komfortfunktionen zu entwerfen.

Die oben erwähnten Mindestanforderungen definieren das zu erarbeitende System als fähig, dem Stundenplan Lehrer, Klassen und Schüler hinzufügen zu können wobei den Lehrern eine erfassbare Wochenstundenzahl zugeteilt ist.

Weiterhin müssen sich die planbaren Aktivitäten in Pausen und Unterrichtsfächer aufteilen lassen können, sowie Startpunkte und Endpunkte besitzen und mehreren Klassen oder Lehrern zugleich zuteilbar sein.

Darüber hinaus soll die Möglichkeit bestehen für die Lehrer und Klassen jeweils spezifische Stundenpläne anzeigen zu lassen, innerhalb derer ebenfalls die Bearbeitung und Verwaltung von Aktivitäten umsetzbar sein soll.

Zuletzt soll das System zum Zweck der Problemlösung automatisch Doppelbelegungen der Lehrer oder eine im Hinblick auf die zugewiesene Wochenstundenzahl festgestellte Überbelegung anzeigen.

1.1.2 Hauptarbeitsaktivitäten und -produkte

Der Entwicklungsprozess eines Softwareprojektes lässt sich gemeinhin in mehrere Arbeitsaktivitäten oder auch Arbeitsphasen aufteilen, hierbei bringen die jeweiligen Arbeitsphasen ein spezifisches Arbeitsprodukt hervor, jene Arbeitsprodukte machen in unserem Fall den wesentlichen Bestandteil der Abgaben aus.

Im Folgenden lässt sich anhand des klassischen Musters eine Auflistung erstellen inwiefern unser Arbeitsprozess in die Arbeitsaktivitäten aufgeteilt ist und welche Arbeitsprodukte aus ihnen resultieren.

Arbeitsaktivität - Arbeitsprodukt

Projektplanung - Projektplan

Anforderungsanalyse - Angebotserstellung

Anforderungsspezifikation - Angebot

Entwurf - Architekturbeschreibung

Erstellung des Testplans - Testplan

Testvorgang - Tests und Testergebnisse

Implementierung - Programm

Dokumentation - Installationsanweisung und Skript

Auslieferung - Kunde erhält Produkt

1.1.3 Haupt-Meilensteine und grober Zeitplan

Die für dieses Projekt angesetzten Meilensteine resultieren unmittelbar aus den im Rahmen des Moduls festgelegten Abgabeterminen der einzelnen Arbeitsprodukte, abgesehen von diesen fixen Terminen, zu welchen die jeweiligen Arbeitsteile oder Gesamtarbeiten eingereicht werden müssen, werden natürlich auch vorzeitige Termine zur grundlegenden Anforderungsbesprechung oder eventueller Korrektur flexibel und auf Bedarf vor dem eigentlichen Abgabedatum eingesetzt.

Im Folgenden ergibt sich für die Hauptmeilensteine deshalb ein grober Zeitplan, der genug Flexibilität ermöglicht um auf etwaige Probleme einzugehen und gleichzeitig zur organisatorischen Steuerung des Projektverlaufs verwendet werden kann.

Abgabetermin - Dokument

18.05.2014 - Übungsblatt 1 (Zusammenfügung der Arbeitspakete vor der Abgabe)

25.05.2014 - Initialer Projektplan (Zusammenfügung der Arbeitspakete vor der Abgabe)

01.06.2014 - Übungsblatt 2 (Zusammenfügung der Arbeitspakete vor der Abgabe)

TBA - Anforderungsspezifikation (Gemeinsame Ausarbeitung und Abgabe derselben)

TBA - Architekturbeschreibung mit Schnittstellen in Java (Gemeinsame Ausarbeitung)

TBA - Testplan und JUnit-Tests (Aufteilung einzelner Tests und Fügung vor Abgabe)

Am Freitag des Blockkurs um 12 Uhr - Vollständige Abgabe (Gruppenabgabe)

1.2 Auszuliefernde Produkte

Die von uns auszuliefernden Produkte sind aufgrund der Art des Miniprojekt-Übungsbetriebes durch ebendiesen vorgegeben.

Unsere hauptsächliche Produktleistung definiert sich also über die Erstellung von einem

initialen Projektplan, der genauen Anforderungsspezifikation für die erdachte Software, der Architekturbeschreibung mit Schnittstellen in Java derselbigen, dem Testplan und der Ansammlung von über JUnit durchgeführten Tests, sowie zuletzt der vollständigen Abgabe.

1.3 Referenzen

Im Bezug auf etwaige Referenzen und unter Berücksichtigung der innerhalb des Projektbetriebes empfohlenen Literatur lässt sich für unser Projekt die folgende Referenzaufstellung anfertigen.

Ludewig, J., H. Lichter Software Engineering - Grundlagen, Menschen, Prozesse, Techniken 2. Aufl., dpunkt.verlag, Heidelberg, ISBN 978-3-89864-662-8, 2010

R. Pressman: Software Engineering - A Practitioner's Approach. 6. Auflage, McGraw-Hill, 2004.

I. Sommerville: Software Engineering. 8. Auflage, Addison-Wesley, 2006.

W. Zuser, T. Grechenig, M. Köhle: Software Engineering mit UML und dem Unified Process. 2. Auflage, Pearson Studium, 2004.

B. Brügge, A. H. Dutoit: Objektorientierte Softwaretechnik mit UML, Entwurfsmustern und Java. Pearson Studium, 2004.

H. Störrle: UML 2 für Studenten. Pearson Studium, 2005.

M. Dahm: Grundlagen der Mensch-Computer-Interaktion. Pearson Studium, 2005.

1.4 Definitionen und Akronyme

Aufgrund der starken Nähe zur Anwendung im Alltag, welche unser Projekt auszeichnet, sind auch für das Verständnis des Projektplanes wenig Fachwissen oder Alltagskenntnisse erforderlich.

2 Projektorganisation

Muss in SWP-2 ausgefüllt werden

2.1 Prozessmodell

Muss in SWP-2 ausgefüllt werden

2.2 Organisationsstruktur

Muss in SWP-2 ausgefüllt werden

Genaue Beschreibung der Rollen, Rechte und Pflichten!

z.B. auch regelmäßiges Treffen im Chat, Einrichtung einer Groupware oder eines Forums, o.ä. ...

2.3 Organisationsgrenzen und –schnittstellen

Muss in SWP-2 ausgefüllt werden

Hierher gehören auch evtl. Kontaktpersonen für Fremdbibliotheken u.ä.

2.4 Verantwortlichkeiten

Muss in SWP-2 ausgefüllt werden

3 Managementprozess

3.1 Managementprozess und –prioritäten

3.2 Annahmen, Abhängigkeiten und Einschränkungen

3.3 Risikomanagement

Wenn Ihr Euch entschieden habt, bestimmte vorbeugende Maßnahmen durchzuführen, solltet Ihr dies deutlich kennzeichnen. Hoffentlich haben diese Maßnahmen dann einen Einfluss auf Eintrittswahrscheinlichkeit oder Schadenshöhe (zum Beispiel ist die Eintrittswahrscheinlichkeit von komplettem Datenverlust durch regelmäßige Backups deutlich geringer). Daher solltet Ihr für diese Fälle dann die verringerten Werte für Eintrittswahrscheinlichkeit, Schadenshöhe und Risikopotential zusätzlich angeben.

Wie werden neue Risiken erkannt/erfasst? Wer ist für was zuständig? Wie ist der Informationsfluss? ...

Dieser Teil ist ein wichtiger Schwerpunkt des Projektplans und sollte daher ausführlich behandelt werden.

Um eine schnellere Erfassung der Risiken zu bekommen, eignen sich immer visuelle Darstellungen. Wir verwenden dafür eine Matrix/Tabelle, bei der man die Risikostufen schnell erkennen kann.

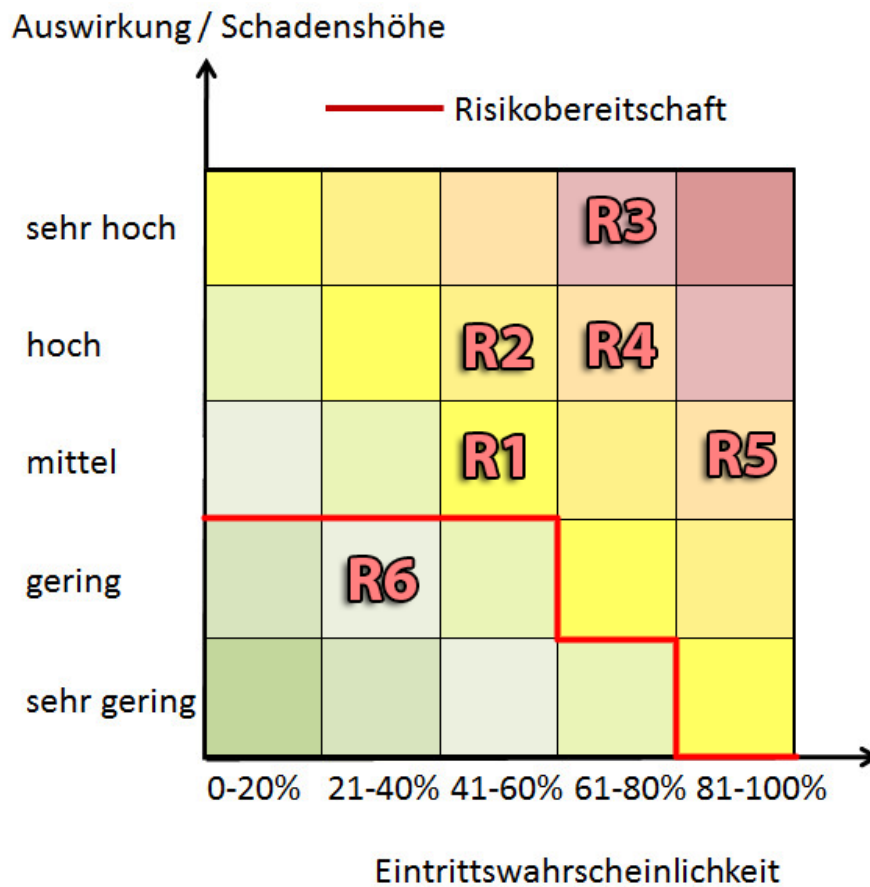


Abbildung 1: Risikomatrix vor Maßnahmenergreifung

3.3.1 (Mögliche) Risiken

- R1:** Ein Gruppenmitglied kann nicht programmieren.
- R2:** Am Tag des Blockkurses sind mindst. 2 Gruppenmitglieder nicht anwesend/verhindert.
- R3:** Konflikte bahnen sich an, die den erfolgreichen Abschluss des Projektes gefährden.
- R4:** (Interne) (Gruppen-)Kommunikationen kommen nicht (erfolgreich) zustande.
- R5:** Festgelegte Auflagen werden (oft) nicht

eingehalten.

R6: Der/Die BetreuerIn/TutorIn/AufpasserIn
fällt für längere Zeit aus.

R1: Eine Erklärung ist wohl nicht von Nöten. Wenn man sich die Ausgangssituation anschaut, kann entweder einer/eine programmieren oder nicht. Es wären ca. 50%. Der Schaden wäre mittel bis hoch, da man programmieren muss, um den Stundenplan erstellen zu können. Noch kann man nicht durch pure Gedankenkraft, materielle Gegenstände erzeugen. Sehr schlimm wird es, wenn das Risiko kurz vor der Implementierung entdeckt wird.

Das Auslösen dieses Risikos kann (gänzlich) vermieden werden, wenn man zuvor Eignungstests bzw. Programmiertests durchführt. Noch kann man der Person helfen. Sobald das Wintersemester anfängt, wird dieses Risiko zu 100% eintreten, wenn es wirklich eine Person mit diesem Defizit gibt. Jede Hilfe wäre dann vergebens.

R2: Sowas kann immer vorkommen, da plötzlich einer krank wird, die Bahn mal wieder im „Eiltempo“ die Passagiere verärgert oder ganz schlicht einer verschläft. Lustlosigkeit, da es in der vorlesungsfreien Zeit stattfindet, soll wohl auch schon mal vorgekommen sein.

Es wäre ein Desaster auf hohem Niveau, da spontan Überstunden zu praktizieren, nicht mehr einplanbar wäre. Dieses Risiko einzuschätzen, wie wahrscheinlich es eintritt, ist de facto kaum möglich. Die Indikatoren sind nämlich (fast) nicht vom Menschen beeinflussbar.

Das Miniprojekt fertig zu stellen, würde in die weite Ferne rücken.

Präventionspläne kann man nicht realisieren. Da hilft wohl nur, einen (oder mehrere) Alternativpläne vorzubereiten und wenn das Risiko auftreten sollte, wenigstens das Nötigste zu implementieren. Man würde den Schaden begrenzen.

R3: Ein oft unterschätztes Risiko. Konflikte werden bei jeder Gruppe auftreten. Das ist unvermeidbar. Glücklicherweise sind diese Konflikte nicht allzu groß und lösen sich schnell wieder auf. Das gilt jedoch nicht für alle. Die meisten ernstzunehmenden Konflikte entwickeln sich aus kleineren [Konflikten], die wie zuvor genannt, unterschätzt werden. Dies kann nicht nur die Zusammenarbeit dauerhaft schädigen, sondern auch das Projekt zum Scheitern verdammen. Unserer Einschätzung nach, gehört es zu den Gefährlichsten Risiken, die wir identifizieren konnten (siehe Abb. 1).

So gefährlich es ist, so leicht lässt es sich lösen. Man muss das Problem an der Wurzel

packen und es erst gar nicht entfalten lassen. Schon Gruppenmeetings, Gespräche mit einer neutralen Person (TutorIn, ProfessorIn) wirken Wunder, auch wenn man es nicht glauben kann.

So könnte man den Schaden bis aufs Minimum drücken und die Wahrscheinlichkeit etwas senken.

Ausnahmen bestätigen die Regeln. Manche Konflikte sind - mit der derzeitigen Konstellation - nicht lösbar. Da hilft nur eine Gruppentrennung, damit beide Seiten den Kollateralschaden möglichst gering halten können.

R4: Dieses Risiko soll 2 Fälle beinhalten.

Den ersten Fall mussten wir schon recht früh selbst erleben. 2 Gruppenmitglieder haben auf verschiedensten Kommunikationswegen einfach zu spät bis gar nicht geantwortet. Um ein Bildnis machen zu können: 1-3 Wochen dauerte es, bis auf Fragen und anderen (organisatorischen) Aufgaben, erwidert wurde. Nicht nur das es unhöflich ist - zumindest in Europa, es bringt die ganze Planung (Zeitmanagement und Taskmanagement) durcheinander. Zwar ist es üblich, dass Pläne nicht eingehalten werden, jedoch ist keine allzu hohe Varianz vorhanden. (Es müsste sich auf 5% belaufen, jedoch habe ich keine Quelle parat und somit sollte diese Aussagen nur als These angenommen werden.)

Der zweite Fall ist die Misskommunikation. Man hat sich verhöhrt oder hat es sich so eingebildet, als hätte man es so gesagt.

Das man mal etwas (akustisch) nicht mitbekommt, bedeutet noch nicht das eindeutige Aus. Es kann jedoch den kritischen Pfaden „verlängern“. D.h., Aufgaben, Prozesse, die eine hohe Abhängigkeit besitzen, müssen verschoben werden. Autarke Prozesse zu erstellen, ist nicht immer möglich bzw. Vorteilhaft.

Eine mögliche Vorbeugung: Alles Wichtige wird nochmal als expliziertes Wissen manifestiert (z.B. auf Papier, in einer E-Mail, im Forum, ...). Die Schadenshöhe würde auf dem gleichen Level bleiben. Die Eintrittswahrscheinlichkeit würde sinken, da man eventuell die Informationen immer wieder sich durchlesen könnte. Verbales wird im primären Gedächtnis gespeichert und rund 80% davon sind innerhalb des nächsten halben Tages „verschwunden“.

R5: Das sowas eintreten kann, ist hoch, da wir neben SWP auch noch andere akademische Tätigkeiten vollführen und manche noch Arbeiten gehen. Der Schaden kann in alle möglichen Stufen auftreten. Wir haben uns für den Durchschnitt entschieden und das Mittelmaß gewählt (siehe Abb. 1). Zur Vorbeugung der Schadenshöhe, haben wir einen Auflagenkatalog erstellt, der bis Juli 2014 die finale Phase erreichen wird. Damit es keine Missverständnisse geben soll, muss sich jedes Gruppenmitglied diesen Katalog durchlesen und auch zustimmen (Unterschrift). Da wir es Gruppenintern belassen wollen, wird hier nur ein kleiner Ausschnitt gezeigt:

Jeder hat ein Kontingent von **11 Punkten**

Wer diese Punktzahl erreicht bzw. überschreitet, wird aus der Gruppe **ausgeschlossen!**

Person meldet sich erst **nach 2 Tagen ohne Begründung**

– > **3,5** Punkte

Person gibt wiederholt nicht ab

– > **5,5** Punkte

Person kommt innerhalb **eines Monats** insgesamt **30min** zu spät

– > **2,0** Punkte

Zuvor zugesagte Aufgaben werden nicht erledigt (und die anderen nicht darüber informiert)

– > **1,5** Punkte

Person macht nichts/stachelt bewusst Konflikte an

– > **8,0** Punkte

Person will unbedingt die Gruppe durchfallen lassen

– > **11,0** Punkte

Es wurde wiederholt nicht nach Auflagen gehalten (Schreibtool, Dokumentsprache, falsches CVS, ...)

– > **jeweils 1,0** Punkte

Interne Deadlines wurden nicht eingehalten - und nicht rechtzeitig informiert

– > **0,5 - 2,5 (je nach Häufigkeit)** Punkte

Person erledigt mehr Aufgaben als besprochen wurde - bei einer konstant hohen Qualität

– > **-3,5** Punkte

Person investiert mehr Zeit (freiwillig) z.B. Überstunden (mit einer Obergrenze)

– > **-1,0 - -5,0** Punkte

Alle 2 Wochen fällt der Kontostand um 7 Punkte!

Ein Programm übernimmt diese Aufgaben.

R6: Bei diesem Risiko sind uns die Hände gebunden. Zwar ist der Schaden und der Eintritt gering, jedoch können wir es nicht weiter kompensieren.

Es ist aber auch das Risiko, bei dem wir die Schadenshöhe von Beginn an akzeptieren.

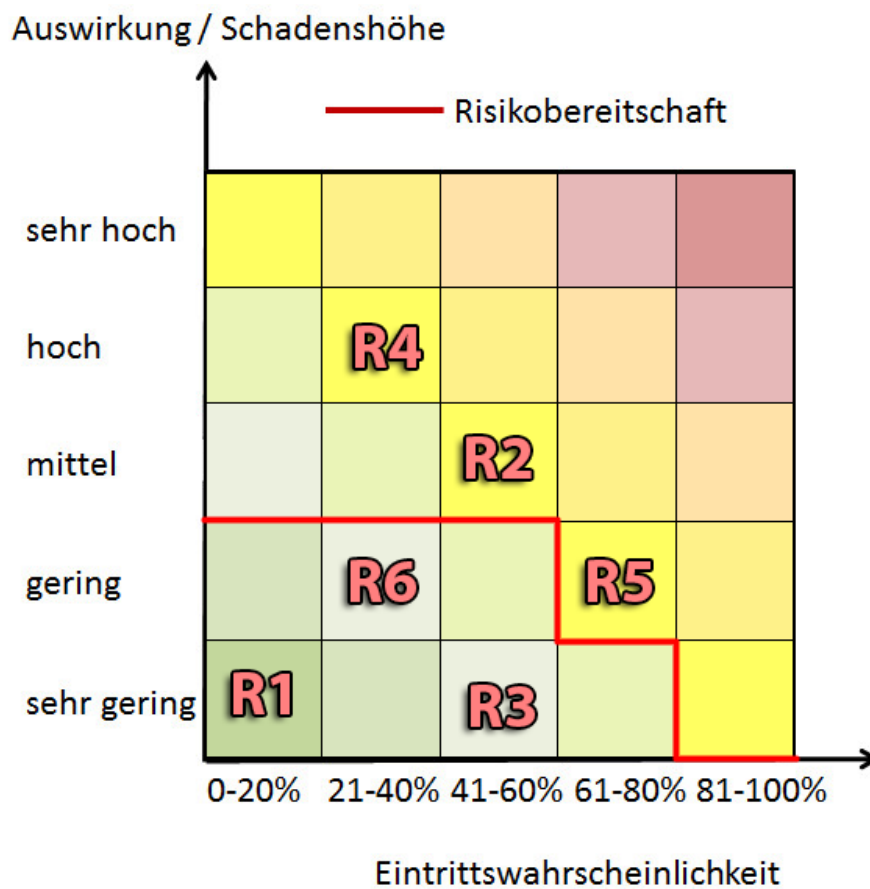


Abbildung 2: Risikomatrix nach Maßnahmenenergreifung

3.4 Projektüberwachung

Wie wird der Projektstatus verfolgt? Wie stellt Ihr sicher, dass der Phasenleiter jederzeit über den Stand der Entwicklung informiert ist? Wie werden Probleme bzw. Verzögerungen frühzeitig erkannt und angegangen?

Wir treffen uns mindestens einmal die Woche und geben den Status quo wider. Bei Anomalien wird sofort benachrichtigt, gewarnt und gehandelt, damit man eventuell Krisensitzungen abhalten kann. Notfalls wird eine Skype-Videokonferenz abgehalten, wenn es nicht anders geht.

Im Blockkurs werden wir uns an die Meilensteine (grob) richten. Alle 2h wird der derzeitige Stand (kurz) geschildert. Ebenso Vorschläge als auch Probleme bzw. Prognosen zu zukünftigen Problemen, da 12 Augen mehr erblicken können, als nur ein Einzelner.

3.5 Mitarbeiter

Kompetenzen der und Anforderungen an die Mitarbeiter.

Die Anzahl der Anforderungen ist - in diesem Fall - recht überschaubar, jedoch im Detail komplex.

Ein Minimum an Java-Kenntnissen reicht nur in der Theorie aus. Objektorientierung, Fehleranalyse und Kreation von effizienten Algorithmen, sollte man auch vorweisen können, da kein Kunde gerne 10min wartet, bis eine große Datenbank geladen hat und die Werte erst dann auf dem Bildschirm darstellt.

Natürlich ist es nicht der Standard. Es wird (verständlicherweise) immer Gruppenmitglieder geben, die eine Sache nicht vorweisen können.

Um dieser Lage Herr zu werden, ist eine optimale Aufgabenaufteilung erforderlich. Dazu später mehr [5]

Beobachtung und Analyse wären auch sehr empfehlenswert, wenn man später die Anforderungsspezifikation erstellen muss und in der Ist-/Soll-Phase, Personas-Phase, ... sich befindet. Da es keine richtigen Anforderungen sind und langsam am Thema vorbei geschrieben wird, wird dieser Aspekt offen in dem Raum gestellt.

4 Technische Prozesse

Muss in SWP-2 ausgefüllt werden

4.1 Methoden, Werkzeuge und Techniken

Muss in SWP-2 ausgefüllt werden

4.1.1 Entwicklungsplattform

4.1.2 Entwicklungsmethode

Ist der Einsatz spezieller Methoden vorgesehen?

4.1.3 Programmiersprache und Bibliotheken

4.2 Dokumentationsplan

Muss in SWP-2 ausgefüllt werden

4.2.1 Codingstyle

4.2.2 Kommentarsprache

4.2.3 JavaDoc

4.2.4 Begleitende Dokumentation

4.3 Unterstützende Projektfunktionen

Muss in SWP-2 ausgefüllt werden Wie wird Euer Konfigurationsmanagement funktionieren? Wer ist verantwortlich? Benötigt Ihr dazu Ressourcen oder Zeit? Plant Ihr Datensicherung?

Gibt es Maßnahmen zur Qualitätssicherung? Wer ist zuständig? Wieviel Zeit ist dafür vorgesehen?

5 Arbeitspakete, Zeitplan und Budget

Dieser Teil ist ein zweiter Schwerpunkt des Projektplans. Hier sollt Ihr die nächste Phase detailliert planen (siehe Arbeitspakete). Die weiteren Phasen sollen ebenfalls wenigstens grob geplant werden. Ein Gantt-Diagramm ist zwingend!

Ihr sollt den Plan in der kommenden Phase auch tatsächlich benutzen – und so Erfahrungen sammeln, was evtl. bei der Planung unberücksichtigt blieb. Bei der nächsten Zeitplanung (für die nächste Phase) bekommt Ihr dann evtl. eine noch bessere Planung hin.

5.1 Arbeitspakete

Besonderen Wert legen wir auf die Granularität der APs. Diese sollten von 1-2 Personen in max. einer Woche Zeitdauer (kalendarisch, nicht Aufwand) bearbeitbar sein. Die Beschreibungen sollten so genau sein, dass der Bearbeiter damit genau weiß, was zu tun ist.

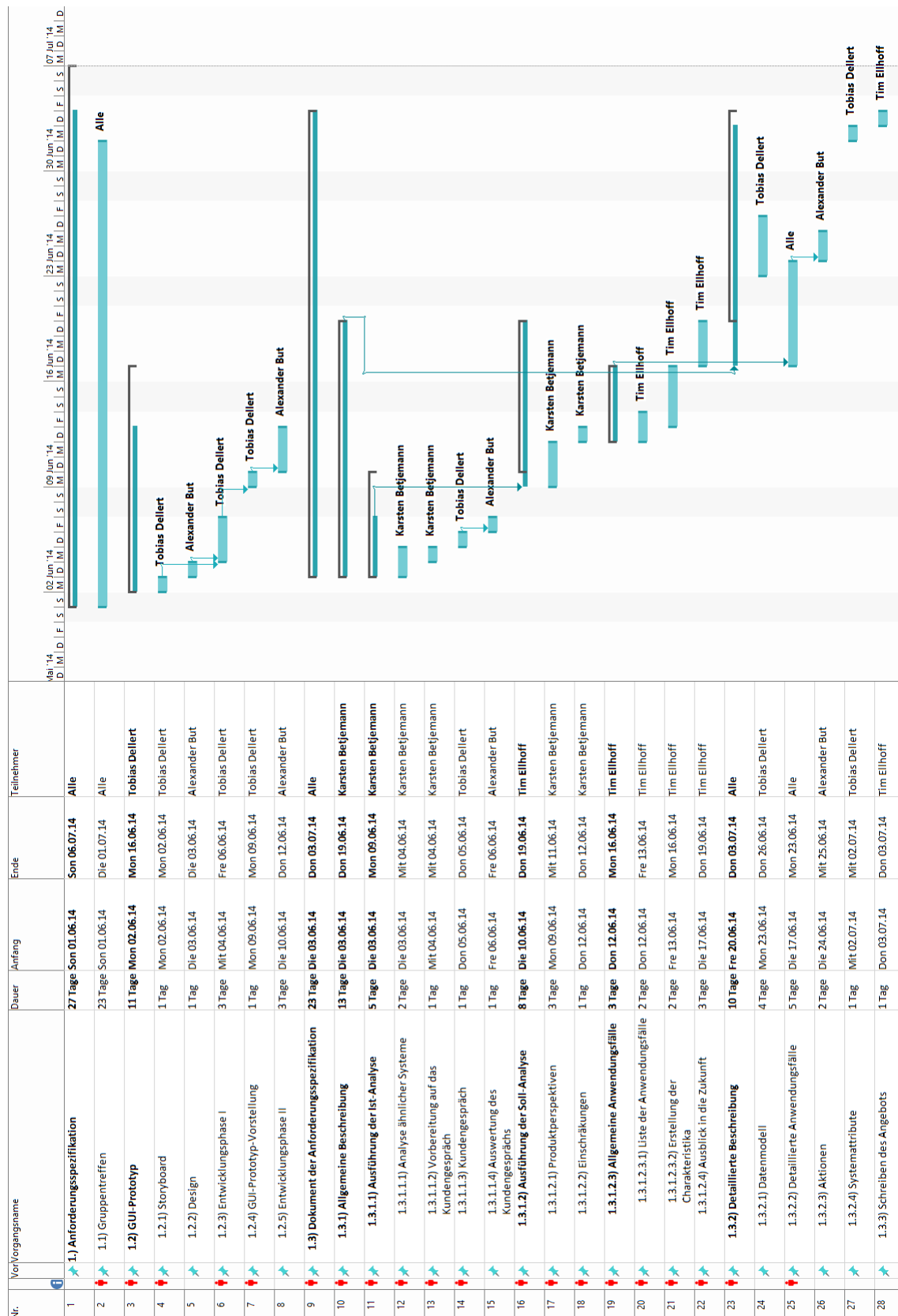


Abbildung 3: Anforderungsspezifikation im Gantt-Diagramm

| | |
|------------------------------|--|
| Arbeitspaket Nr.: 1.0 | Bezeichnung: Anforderungsspezifikation |
| Beginn | 01.06.14 |
| Ende | 06.07.14 |
| Zuständig | Alle |
| Ressourcen | <ul style="list-style-type: none"> • Alle |
| Abhängigkeit | |
| Aufwand | |
| Dauer | |
| Beschreibung: | |
| Mindestkriterien: | |

| | |
|--|---|
| Arbeitspaket Nr.: 1.1 | Bezeichnung: Gruppentreffen |
| Beginn | 01.06.14 |
| Ende | 01.07.14 |
| Zuständig Alle | |
| Ressourcen | <ul style="list-style-type: none"> • • blaa |
| Abhängigkeit | |
| Aufwand | 8h |
| Dauer | ? |
| Beschreibung: Um die Qualität der einzelnen Bearbeitungen sicherzustellen und eine Möglichkeit für Fragen und Besprechungen zu bieten, werden wir uns als Gruppe mindestens einmal in der Woche gemeinsam treffen. | |
| Mindestkriterien: Jeder Arbeitsvorgang kann bei Bedarf ausreichend für die weitere Bearbeitung besprochen werden. | |

| | |
|--|---|
| Arbeitspaket Nr.: 1.2 | Bezeichnung: GUI-Prototyp |
| Beginn | 02.06.14 |
| Ende | 16.06.14 |
| Zuständig | Tobias Dellert |
| Ressourcen | <ul style="list-style-type: none"> • Alexander But |
| Abhängigkeit | |
| Aufwand | 16h |
| Dauer | |
| Beschreibung: Der hier erstellte Prototyp soll die Beschaffenheit der späteren Benutzeroberfläche darstellen. | |
| Mindestkriterien: Der Kunde ist mit dem Prototypen einverstanden. | |

| | |
|---|--|
| Arbeitspaket Nr.: 1.2.1 | Bezeichnung: Story-Board |
| Beginn | 02.06.14 |
| Ende | 02.06.14 |
| Zuständig | Tobias Dellert |
| Ressourcen | <ul style="list-style-type: none"> • Tobias Dellert |
| Abhängigkeit | - |
| Aufwand | 1h |
| Dauer | 1 Tag |
| Beschreibung: Das Story-Board beschreibt die erforderlichen Darstellungen, um die möglichen Vorgänge abzudecken | |
| Mindestkriterien: Alle mögliche Vorgänge sind abgedeckt. | |

| | |
|--|---|
| Arbeitspaket Nr.: 1.2.2 | Bezeichnung: Design |
| Beginn | 03.02.14 |
| Ende | 03.02.14 |
| Zuständig | Alexander But |
| Ressourcen | <ul style="list-style-type: none">• Alexander But |
| Abhängigkeit | - |
| Aufwand | 4h |
| Dauer | 1 Tag |
| Beschreibung: Hier werden die einzelnen Bestandteile des Prototypen gestaltet. | |
| Mindestkriterien: Das Design soll den Wünschen des Kunden bestmöglich entsprechen | |

| | |
|--|--|
| Arbeitspaket Nr.: 1.2.3 | Bezeichnung: Entwicklungsphase I |
| Beginn | 04.06.14 |
| Ende | 06.06.14 |
| Zuständig | Tobias Dellert |
| Ressourcen | <ul style="list-style-type: none">• Tobias Dellert• Alexander But |
| Abhängigkeit | Story-Board, Design |
| Aufwand | 16h |
| Dauer | 3 Tage |
| Beschreibung: Das Konzept von Story-Board und Design wird umgesetzt. Die Ergebnisse der ersten Entwicklungsphase werden dem Kunden vorgestellt. | |
| Mindestkriterien: Die Entwicklungsphase ist ein Ergebnis aus dem Story-Board und dem Design, welche auf dem Verständnis der Wünsche des Kunden basieren. | |

| | |
|---|---|
| Arbeitspaket Nr.: 1.2.4 | Bezeichnung: GUI-Prototyp-Vorstellung |
| Beginn | 11.06.14 |
| Ende | 11.06.14 |
| Zuständig | Tobias Dellert |
| Ressourcen | <ul style="list-style-type: none"> • Tobias Dellert • Alexander But |
| Abhängigkeit | Entwicklungsphase I |
| Aufwand | 6h |
| Dauer | 1 Tag |
| Beschreibung: Das Ergebnis der ersten Entwicklungsphase wird dem Kunden vorgestellt. Es wird bestmöglich versucht die Meinung des Kunden dazu und eventuelle Verbesserungswünsche zu verstehen. | |
| Mindestkriterien: Die Meinung des Kunden wurde erfasst und das weitere Vorgehen beim GUI-Prototypen ist klar. | |

| | |
|---|---|
| Arbeitspaket Nr.: 1.2.5 | Bezeichnung: Entwicklungsphase II |
| Beginn | 12.06.14 |
| Ende | 16.06.14 |
| Zuständig | Alexander But |
| Ressourcen | <ul style="list-style-type: none"> • Alexander But • Tobias Dellert |
| Abhängigkeit | GUI-Prototyp-Vorstellung |
| Aufwand | 8h |
| Dauer | 5 tage |
| Beschreibung: Der Prototyp wird gegebenenfalls angepasst. | |
| Mindestkriterien: Die Erkenntnisse aus der Prototyp-Vorstellung werden umgesetzt. | |

| | |
|---|--|
| Arbeitspaket Nr.: 1.3 | Bezeichnung: Dokument der Anforderungsspezifikation |
| Beginn | 03.06.14 |
| Ende | 04.07.14 |
| Zuständig | Alle |
| Ressourcen | <ul style="list-style-type: none"> • Alle |
| Abhängigkeit | |
| Aufwand | 79h |
| Dauer | 30 tage |
| Beschreibung: Im Dokument werden die Anforderungen an das System dargestellt, was grundlegend für die weitere Entwicklung in unserem Projekt ist. Es wird eine Gespräch bezüglich der Wünsche abgehalten, ähnliche Systeme werden untersucht und verglichen, und die gewonnenen Erkenntnisse werden anwendungsbezogen dargestellt. | |
| Mindestkriterien: Das Dokument stellt die Wünsche des Kunden ausreichend dar, um aufgrund dessen eine gelungene Architektur zu entwerfen. | |

| | |
|--|---|
| Arbeitspaket Nr.: 1.3.1 | Bezeichnung: Allgemeine Beschreibung |
| Beginn | 03.06.14 |
| Ende | 19.06.14 |
| Zuständig | Karsten Betjemann |
| Ressourcen | <ul style="list-style-type: none"> • Tim Ellhoff • Tobias Dellert • Karsten Betjemann • Alexander But |
| Abhängigkeit | |
| Aufwand | 23h |
| Dauer | 6 Tage |
| Beschreibung: Um den bisherigen Zustand bezüglich bestehender Anforderungen oder Wünsche des Kunden zu erfassen, wird sowohl ein Gespräch geführt, als auch zuvor ein Vergleich mit ähnlichen Systemen durchgeführt. | |
| Mindestkriterien: Die Anforderungen und Wünsche des Kunden werden erfasst. | |

| | |
|--|---|
| Arbeitspaket Nr.: 1.3.1.1 | Bezeichnung: Ausführung der Ist-Analyse |
| Beginn | 03.06.14 |
| Ende | 09.06.14 |
| Zuständig | Karsten Betjemann |
| Ressourcen | <ul style="list-style-type: none"> • Tim Ellhoff • Tobias Dellert • Karsten Betjemann • Alexander But |
| Abhängigkeit | |
| Aufwand | 13h |
| Dauer | 6 Tage |
| Beschreibung: Um den bisherigen Zustand bezüglich bestehender Anforderungen oder Wünsche des Kunden zu erfassen, wird sowohl ein Gespräch geführt, als auch zuvor ein Vergleich mit ähnlichen Systemen durchgeführt. | |
| Mindestkriterien: Die Anforderungen und Wünsche des Kunden werden erfasst. | |

| | |
|--|---|
| Arbeitspaket Nr.: 1.3.1.1.1 | Bezeichnung: Analyse ähnlicher Systeme |
| Beginn | 03.06.14 |
| Ende | 04.06.14 |
| Zuständig | Karsten Betjemann |
| Ressourcen | <ul style="list-style-type: none"> • Karsten Betjemann |
| Abhängigkeit | |
| Aufwand | 3h |
| Dauer | 1 Tag |
| Beschreibung: Es werden die Funktionen und daraus folgenden Vor- und Nachteile abgewogen, um später diese mit den Anforderungen des Kunden zu vergleichen. | |
| Mindestkriterien: Es wird mindestens ein ähnliches System analysiert | |

| | |
|---|--|
| Arbeitspaket Nr.: 1.3.1.1.2 | Bezeichnung: Vorbereitung auf das Kundengespräch |
| Beginn | 04.06.14 |
| Ende | 04.06.14 |
| Zuständig | Karsten Betjemann |
| Ressourcen | <ul style="list-style-type: none">• Tobias Dellert• Alexander But• Karsten Betjemann |
| Abhängigkeit | |
| Aufwand | 3h |
| Dauer | 1 Tag |
| Beschreibung: Es werden eventuell offene Fragen zur den Anforderungen und dessen Durchführungen notiert. Unter anderem auch basierend auf den Erkenntnissen der vorherigen Analyse ähnlicher Systeme. | |
| Mindestkriterien: Der erhaltene Fragenkatalog enthält die wichtigsten zu besprechende Punkt. | |

| | |
|---|--|
| Arbeitspaket Nr.: 1.3.1.1.3 | Bezeichnung: Kundengespräch |
| Beginn | 05.06.14 |
| Ende | 05.06.14 |
| Zuständig | Tobias Dellert |
| Ressourcen | <ul style="list-style-type: none">• Tobias Dellert• Alexander But |
| Abhängigkeit | |
| Aufwand | 6h |
| Dauer | 1 Tag |
| Beschreibung: Anhand des Fragenkataloges werden diese und vermutlich noch aufkommende Fragen dem Kunden gestellt. | |
| Mindestkriterien: Am Ende sollen alle Fragen geklärt worden sein. | |

| | |
|---|---|
| Arbeitspaket Nr.: 1.3.1.1.4 | Bezeichnung: Auswertung des Kundengesprächs |
| Beginn | 05.06.14 |
| Ende | 05.06.14 |
| Zuständig | Alexander But |
| Ressourcen | <ul style="list-style-type: none"> • Alexander But • Tobias Dellert |
| Abhängigkeit | Kundengespräch |
| Aufwand | 1h |
| Dauer | 1 Tag |
| Beschreibung: Die Beantwortungen der Fragen werden Zusammengesetzt und mit den bestehenden Mindestanforderungen in Verbindung gebracht. | |
| Mindestkriterien: Es soll ein umfassendes Bild der Anforderungen erlangt worden sein. | |

| | |
|---|--|
| Arbeitspaket Nr.: 1.3.1.2 | Bezeichnung: Ausführung der Soll-Analyse |
| Beginn | 09.06.14 |
| Ende | 16.06.14 |
| Zuständig | Tim Ellhoff |
| Ressourcen | <ul style="list-style-type: none"> • Tim Ellhoff • Karsten Betjemann |
| Abhängigkeit | Ausführung der Ist-Analyse |
| Aufwand | 10h |
| Dauer | 8 Tage |
| Beschreibung: Die Erkenntnisse aus dem Kundengespräch ermöglichen eine gute und zutreffende Interpretation der gestellten Anforderungen. Diese werden hier in Form von einfachen Anwendungsfällen beschrieben werden. Zudem werden die Rahmenbedingungen zur Ausführung festgelegt. | |
| Mindestkriterien: Die Wünsche des Kunden sind in dem beschriebenen Soll-Zustand zutreffend dargestellt. | |

| | |
|---|---|
| Arbeitspaket Nr.: 1.3.1.2.1 | Bezeichnung: Produktperspektiven |
| Beginn | 09.06.14 |
| Ende | 11.06.14 |
| Zuständig | Karsten Betjemann |
| Ressourcen | <ul style="list-style-type: none">• Karsten Betjemann• Tim Ellhoff |
| Abhängigkeit | Kundengespräch |
| Aufwand | 3h |
| Dauer | 2 Tage |
| Beschreibung: Hier werden die grundlegenden Rahmenbedingungen zusammengestellt um für den weiteren Projektverlauf realistische Planungen durchführen zu können. | |
| Mindestkriterien: Alle wichtigen und systembetreffende Bereiche wurden beachtet. | |

| | |
|---|---|
| Arbeitspaket Nr.: 1.3.1.2.2 | Bezeichnung: Einschränkungen |
| Beginn | 12.06.14 |
| Ende | 12.06.14 |
| Zuständig | Karsten Betjemann |
| Ressourcen | <ul style="list-style-type: none">• Karsten Betjemann |
| Abhängigkeit | |
| Aufwand | 1h |
| Dauer | 1 Tag |
| Beschreibung: Die technischen und rechtlichen Einschränkungen während der Durchführung des Projektes und Ausführung der fertigen Software werden erfasst. | |
| Mindestkriterien: Die Einschränkungen wurden soweit erfasst, dass das Projekt technisch und rechtlich durchführbar ist. | |

| | |
|---|--|
| Arbeitspaket Nr.: 1.3.1.2.3 | Bezeichnung: Allgemeine Anwendungsfälle |
| Beginn | 12.06.14 |
| Ende | 16.06.14 |
| Zuständig | Tim Ellhoff |
| Ressourcen | <ul style="list-style-type: none"> • Tim Ellhoff • Karsten Betjemann |
| Abhängigkeit | |
| Aufwand | 4h |
| Dauer | 5 Tage |
| Beschreibung: siehe Unterpunkte | |
| Mindestkriterien: Die Anwendungsfälle decken die möglichen Vorgänge im gewünschten System ab. | |

| | |
|--|---|
| Arbeitspaket Nr.: 1.3.1.2.3.1 | Bezeichnung: Liste der Anwendungsfälle |
| Beginn | 12.06.14 |
| Ende | 13.06.14 |
| Zuständig | Tim Ellhoff |
| Ressourcen | <ul style="list-style-type: none"> • Tim Ellhoff |
| Abhängigkeit | |
| Aufwand | 2h |
| Dauer | 2 Tage |
| Beschreibung: Die allgemein gehaltenen Anwendungsfälle beschreiben den Vorgang in der späteren Software. | |
| Mindestkriterien: siehe Oberpunkt. | |

| | |
|---|---|
| Arbeitspaket Nr.: 1.3.1.2.3.2 | Bezeichnung: Erstellung der Charakteristika |
| Beginn | 13.06.14 |
| Ende | 16.06.14 |
| Zuständig | Tim Ellhoff |
| Ressourcen | <ul style="list-style-type: none"> • Tim Ellhoff |
| Abhängigkeit | |
| Aufwand | 2h |
| Dauer | 2 Tage |
| Beschreibung: Es werden typisch denkbare Personen in der Rolle als Nutzer der Software erstellt. Die Personen sollen helfen sich in die Lage zukünftiger Nutzer hineinzusetzen, da jede dieser Personen unterschiedliche Prioritäten besitzt, die mit der Software aber verfolgbar sein sollen. | |
| Mindestkriterien: Die Software wird typischen Nutzern gerecht. | |

| | |
|---|---|
| Arbeitspaket Nr.: 1.3.1.2.4 | Bezeichnung: Ausblick in die Zukunft |
| Beginn | 17.06.14 |
| Ende | 19.06.14 |
| Zuständig | |
| Ressourcen | <ul style="list-style-type: none"> • Tim Ellhoff |
| Abhängigkeit | |
| Aufwand | 2h |
| Dauer | 3 Tage |
| Beschreibung: Hier werden zu erwartende Änderungen softwarebeeinflussender Faktoren wie Datenschutzrecht und technische Weiterentwicklungen und deren Einfluss auf unser Projekt und System betrachtet. | |
| Mindestkriterien: Der Ausblick war umfassen genug und gab Beiträge zu eventueller Anpassung der Durchführung unseres Projekts in Sachen Wartbar- und Erweiterbarkeit. | |

| | |
|---|--|
| Arbeitspaket Nr.: 1.3.2 | Bezeichnung: Detaillierte Beschreibung |
| Beginn | 23.06.14 |
| Ende | 03.07.14 |
| Zuständig | Alle |
| Ressourcen | <ul style="list-style-type: none"> • Alle |
| Abhängigkeit | Allgemeine Beschreibung |
| Aufwand | 56h |
| Dauer | 10 Tage |
| Beschreibung: Hier werden unter Anderem die Anwendungsfälle spezifiziert und mit den entsprechenden GUI-Oberflächen versehen. Detaillierungen unterstützen den späteren Übergang der Konzepte der Anforderungsspezifikation in die technische Sicht der Architektur. | |
| Mindestkriterien: | |

| | |
|--|--|
| Arbeitspaket Nr.: 1.3.2.1 | Bezeichnung: Datenmodell |
| Beginn | 23.06.14 |
| Ende | 26.06.14 |
| Zuständig | Tobias Dellert |
| Ressourcen | <ul style="list-style-type: none"> • Tobias Dellert |
| Abhängigkeit | |
| Aufwand | 6h |
| Dauer | 3 Tage |
| Beschreibung: Das Datenmodell beschreibt einen Ausschnitt der realen Welt in Form vom Austausch von Informationen verschiedener Klassen. Dieses Modell wird als UML-Klassendiagramm erstellt und bezieht sich auf ein fachlogisches Konzept, ohne Implementierungsdetails. | |
| Mindestkriterien: Das Datenmodell beschreibt den Informationsfluss treffend und einfach. | |

| | |
|--|--|
| Arbeitspaket Nr.: 1.3.2.2 | Bezeichnung: Detaillierte Anwendungsfälle |
| Beginn | 24.06.14 |
| Ende | 30.06.14 |
| Zuständig | Alle |
| Ressourcen | <ul style="list-style-type: none">• Alle |
| Abhängigkeit | Allgemeine Anwendungsfälle |
| Aufwand | 40h |
| Dauer | 7 Tage |
| Beschreibung: Hier werden die Anwendungsfälle detailliert und mit Screenshots der GUI-Oberfläche versehen. | |
| Mindestkriterien: Die Anwendungsfälle stellen alle möglichen Vorgänge des späteren Systems dar. | |

| | |
|---|---|
| Arbeitspaket Nr.: 1.3.2.3 | Bezeichnung: Aktionen |
| Beginn | 01.07.14 |
| Ende | 02.07.14 |
| Zuständig | Alexander But |
| Ressourcen | <ul style="list-style-type: none">• Alexander But |
| Abhängigkeit | Detaillierte Anwendungsfälle |
| Aufwand | 6h |
| Dauer | 2 Tage |
| Beschreibung: Hier werden die detaillierteren Anwendungsfälle implementierungsspezifisch dargestellt. Ihnen werden also zukünftige Methodennamen und Parameter zugeteilt, welche die entsprechende Methode für die Ausführung benötigen wird. | |
| Mindestkriterien: Allen Anwendungsfällen wurden Methoden und entsprechende Parameter zugeordnet. | |

| | |
|---|--|
| Arbeitspaket Nr.: 1.3.2.4 | Bezeichnung: Systemattribute |
| Beginn | 02.07.14 |
| Ende | 02.07.14 |
| Zuständig | Tobias Dellert |
| Ressourcen | <ul style="list-style-type: none"> • Tobias Dellert |
| Abhängigkeit | |
| Aufwand | 2h |
| Dauer | 1 Tag |
| Beschreibung: Betrachtung der für die Implementierung benötigten Attribute. | |
| Mindestkriterien: Die Attribute decken die Anforderungen ab. | |

| | |
|--|---|
| Arbeitspaket Nr.: 1.3.3 | Bezeichnung: Schreiben des Angebotes |
| Beginn | 03.07.14 |
| Ende | 03.07.14 |
| Zuständig | Tim Ellhoff |
| Ressourcen | <ul style="list-style-type: none"> • Tim Ellhoff |
| Abhängigkeit | Fertigstellung des Dokuments |
| Aufwand | 2h |
| Dauer | 1 Tag |
| Beschreibung: Hier wird dem Kunden schriftlich ein Angebot unterbreitet, welche Kosten der Erwerb der gewünschten Software zuzuordnen ist. | |
| Mindestkriterien: Die Kostenberechnung wird im Dokument kurz erläutert und Formalitäten wurden beachtet. | |

Architekturbeschreibung und Implementierung:

Die anschließenden Pakete der Architektur und Implementierung sind hier lediglich grob unterteilt.

| | |
|---|--|
| Arbeitspaket Nr.: 2.0 | Bezeichnung: Erstellung der Architektur |
| Beginn | 07.07.14 |
| Ende | 25.07.14 |
| Zuständig | Alle |
| Ressourcen | <ul style="list-style-type: none">• Alle |
| Abhängigkeit | Anforderungsspez. |
| Aufwand | 120h |
| Dauer | 19 Tage |
| Beschreibung: Die Konzepte der Anforderungsspez. werden in verschiedene technisch orientiertere Sichten unterteilt, die als Programmieranleitung dienen und insgesamt eine komplette Systemstruktur bildet. | |
| Mindestkriterien: Die Architektur ist wegweisend für die Implementierung. | |

| | |
|---|--|
| Arbeitspaket Nr.: 2.1 | Bezeichnung: Blackbox-Tests |
| Beginn | 23.07.14 |
| Ende | 24.07.14 |
| Zuständig | Alle |
| Ressourcen | <ul style="list-style-type: none">• Alle |
| Abhängigkeit | Fertigstellung der Architektur |
| Aufwand | 20h |
| Dauer | |
| Beschreibung: Dieser Test achtet lediglich auf die Korrektheit der Ergebnisse der Kommunikation zwischen den einzelnen Modulen. | |
| Mindestkriterien: Alle Schnittstellen wurden überprüft. | |

| | |
|---|--|
| Arbeitspaket Nr.: 3.0 | Bezeichnung: Implementierung |
| Beginn | 04.08.14 |
| Ende | 08.08.14 |
| Zuständig | Alle |
| Ressourcen | <ul style="list-style-type: none"> • Alle |
| Abhängigkeit | Architekturbeschreibung |
| Aufwand | 100h |
| Dauer | 5 Tage |
| Beschreibung: Implementierung der erstellten Programmstrukturen. | |
| Mindestkriterien: Die Mindestanforderungen sind im System realisiert. | |

5.2 Zeitplan und Abhängigkeiten

Die Abhängigkeiten zwischen Arbeitspaketen oder Meilensteinen müssen genannt werden, so wie im Gantt-Diagramm eingezeichnet werden. Der kritische Pfad soll angegeben und/oder eingezeichnet werden!

5.3 Ressourcenanforderung

Jedem Arbeitspaket muss mind. ein Bearbeiter zugeordnet werden. Die Zuordnung der ganzen Gruppe sollte nur in Ausnahmefällen erfolgen – und dann vermutlich begründet werden!

6 Sonstige Elemente

Muss in SWP-2 ausgefüllt werden

6.1 Pläne für die Konvertierung von Daten

Muss in SWP-2 ausgefüllt werden

6.2 Managementpläne für Unterauftragsnehmer

Muss in SWP-2 ausgefüllt werden Wenn Fremdbibliotheken benutzt werden...

6.3 Ausbildungspläne

Muss in SWP-2 ausgefüllt werden *Hierunter fallen z.B. auch interne Schulungen, die Ihr durchführen wollt.*

6.4 Raumpläne

Muss in SWP-2 ausgefüllt werden...

6.5 Installationspläne

Muss in SWP-2 ausgefüllt werden...

6.6 Pläne für die Übergabe des Systems

Muss in SWP-2 ausgefüllt werden...

6.7 Beschaffungspläne für Hardware

Muss in SWP-2 ausgefüllt werden...