

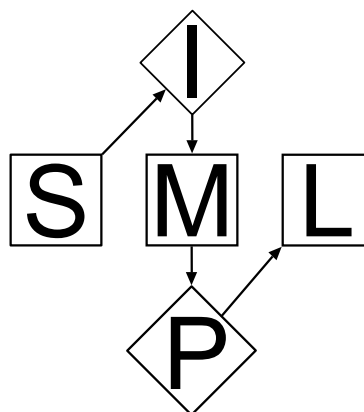
Projektplan

Version 0.1

22. Juli 2009

Verfasser:

Michael Hahn, Firas Zoabi



Dok-Status: neu

QS-Status: nicht QS-geprüft

Prüf-Status: nicht geprüft

Review-Status: kein Review durchgeführt

End-Status: -

Inhaltsverzeichnis

1	Einleitung	4
1.1	Projektübersicht	4
1.2	Auszuliefernde Produkte	5
1.3	Evolution des Plans	5
1.4	Referenzen	5
1.5	Definitionen und Akronyme	5
2	Projektorganisation	5
2.1	Prozessmodell	5
2.2	Organisationsstruktur	5
2.3	Organisationsgrenzen und -schnittstellen	6
2.4	Verantwortlichkeiten	6
3	Managementprozess	7
3.1	Managementziele und -prioritäten	7
3.2	Annahmen, Abhängigkeiten und Einschränkungen	7
3.3	Risikomanagement	7
3.4	Projektüberwachung	8
3.4.1	Berichtswesen	8
3.4.2	Qualitätssicherung und Konfigurationsmanagement	8
3.5	Mitarbeiter	12
4	Technische Prozesse	12
4.1	Methoden, Werkzeuge und Techniken	12
4.1.1	Technologien bzw. Werkzeuge für die Entwicklung	12
4.1.2	Unterstützende Werkzeuge im Projektarbeit	12
4.2	Dokumentationsplan	12
4.3	Unterstützende Projektfunktionen	12
5	Arbeitspakete, Zeitplan und Budget	12
5.1	Beschreibung der Phasen	12
5.1.1	Analyse	12
5.1.2	Einarbeitung	12
5.1.3	Spezifikation	13
5.1.4	Entwurf	13
5.1.5	Codierung	13
5.1.6	Test	13
5.1.7	Abnahmetest	13
5.1.8	Puffer	13
5.2	Arbeitspakete	13
5.3	Abhängigkeiten	15
5.4	Ressourcenanforderung	16
5.5	Zuteilung des Budgets und der Ressourcen auf Projektfunktionen und Aktivitäten	16
5.6	Zeitplan	16
6	Änderungsgeschichte	16

1 Einleitung

1.1 Projektübersicht

Das Projekt läuft von Mai 2009 bis Mai 2010. In diesem Zeitraum soll das Entwicklungsteam ein erweiterbares, generisches Rahmenwerk erstellen, welches den Zugriff auf nahezu beliebige Datenquellen ermöglichen soll. Bei den Datenquellen kann es sich beispielsweise um Sensornetze, Datenbanken und Dateisysteme handeln. Der Schwerpunkt soll klar auf wissenschaftlichen Workflows beruhen. Über das Rahmenwerk sollen beliebige Datenmanagement-Funktionen in einen BPEL-Prozess eingebunden werden können. Dafür werden bereits vorhandene Konzepte evaluiert und für unsere Zwecke erweitert oder angepasst. Dabei wird auch die Sprache BPEL für unsere Zwecke erweitert. Für eine möglichst hohe Flexibilität soll ein dynamischer Ansatz gewählt werden, so dass während der Laufzeit des Systems die Datenquellen festgelegt werden können. Weiterhin sollte auch die Möglichkeit bestehen die Datenquellen statisch anbinden zu können. Der Kunde besteht darauf, dass eine BPEL-Engine sowie ein Modellierungstool um diese gewünschten Funktionen erweitert bzw. angepasst werden. Die BPEL-Prozesse sollen mit dem entsprechenden Modellierungstool modelliert und mit der BPEL-Engine ausgeführt werden können.

Das Projekt besteht aus den folgenden Hauptmeilensteinen/Phasen: Analyse, Einarbeitung, Spezifikation, Entwurf, Implementierung und Test. Eine detailliertere Beschreibung der einzelnen Phasen findet sich in Abschnitt 5.1.

Am Ende des Projekts soll das Rahmenwerk als Eclipse Plug-In und mit allen erforderlichen Tools als Installationspaket zur Verfügung stehen. Darüber hinaus sollen alle Erweiterungen und Anpassungen, die während des Projekts an bestehenden Konzepten durchgeführt wurden, in einer einheitlichen Dokumentation zusammengefasst und ein Handbuch sowie eine Demo für die Arbeit mit dem Rahmenwerk erstellt werden (siehe Tabelle 1).

Das Zeit-Budget des Projekts liegt bei 480h pro Teammitglied bzw. 3360h für das gesamte Team, wie sich das Zeit-Budget auf die einzelnen Projektabschnitte aufteilt zeigt Tabelle ??.

Die Kunden des Projekts sind Katharina Görlach und Peter Reimann, im folgenden ihre Kontaktdaten:

Dipl.-Inf. Katharina Görlach

Institut für Architektur von Anwendungssystemen

Universitätsstraße 38, 70569 Stuttgart (Zimmer 1.328)

Telefon: +49 (0)711 7816-333

Fax: +49 (0)711 7816-472

E-Mail: katharina.goerlach@iaas.uni-stuttgart.de

Dipl.-Inf. Peter Reimann

Institut für Parallele und Verteilte Systeme

Universitätsstraße 38, 70569 Stuttgart (Zimmer 2.467)

Telefon: +49 (0)711 7816-445

Fax: +49 (0)711 7816-424

E-Mail: Peter.Reimann@ipvs.uni-stuttgart.de

1.2 Auszuliefernde Produkte

In Tabelle 1 sind alle auszuliefernden Produkte mit den jeweiligen voraussichtlichen Lieferterminen aufgeführt.

Produkt	Liefertermin
Spezifikation	15.10.2009
Handbuch	17.04.2010
Dokumentation der Erweiterungen	17.04.2010
Testfälle	17.04.2010
Testprotokolle	17.04.2010
Installationspaket, das alle benötigten Software-Komponenten beinhaltet	17.04.2010
Eclipse Plug-In	17.04.2010
Quellcode	17.04.2010

Tabelle 1: Auszuliefernde Produkte mit Terminen

1.3 Evolution des Plans

1.4 Referenzen

1.5 Definitionen und Akronyme

2 Projektorganisation

2.1 Prozessmodell

Als Prozessmodell für das Projekt, wird ein inkrementelles Modell zum Einsatz kommen. Das heißt, dass wir nach Beendigung des Grobentwurfs mehrere Entwicklungsiterationen durchlaufen werden, welche aus Planung, Implementierung und Test des entsprechenden Softwareabschnitts bestehen. Dieses Vorgehen bietet sich für das Projekt an, da es sich hierbei um die Entwicklung einer Kernkomponente handelt, welche nach und nach durch verschiedene Module und Plug-In's erweitert werden kann. Für das Entwicklerteam, aber auch den Kunden, hat dies den Vorteil, dass am Ende jeder Iteration ein lauffähiger Prototyp zur Verfügung steht, an dem der Kunde mögliche Änderungswünsche frühzeitig äußern kann.

2.2 Organisationsstruktur

Der Projektleiter bildet die oberste Instanz des Projekts (siehe Abbildung 1). Er hat als einziger die volle Weisungsbefugnis und Kontrolle über alle Entscheidungen innerhalb des Projekts. Alle anderen Teammitglieder haben nur innerhalb ihres Aufgabengebietes Weisungsbefugnis, da sie am Besten einschätzen können, wie etwas in ihrem Aufgabengebiet umgesetzt werden sollte. Der Projektleiter kann jedoch trotzdem diese Entscheidungen stürzen und so auch demokratische Entscheidungen des Teams gegen den Verantwortlichen des Aufgabengebiets durchsetzen, sollte dies notwendig sein.

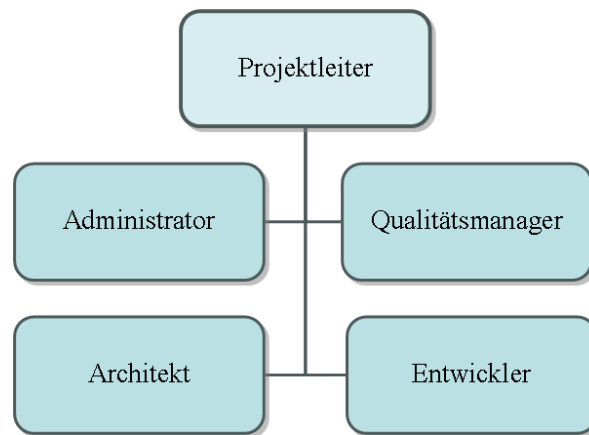


Abbildung 1: Organisationsstruktur als Organigramm

Die Kommunikation innerhalb des Projekts wird durch die Verwendung einer Mailing-Liste sowie wöchentlichen Treffen des gesamten Teams gesichert. Dadurch ist das ganze Team immer über die aktuellen Arbeiten und Fortschritte jedes Teammitglieds informiert und die gewonnenen Erkenntnisse oder getroffenen Entscheidungen sind allen bekannt und für jeden nachvollziehbar.

2.3 Organisationsgrenzen und -schnittstellen

Der Projektleiter bildet die alleinige Schnittstelle des Projektteams nach Aussen, d.h. die gesamte Kommunikation mit den Kunden und Betreuern erfolgt über den Projektleiter (siehe Abbildung 2).

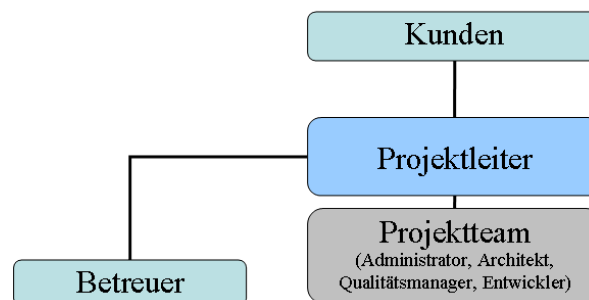


Abbildung 2: Schnittstellen innerhalb der Organisation

2.4 Verantwortlichkeiten

Projektleiter: Wolfgang Hüttig (Stellvertreter: Michael Hahn)

Der Projektleiter ist der zentrale Ansprechpartner für den Kunden. Er koordiniert sämtlichen Schriftverkehr und die Kommunikation zwischen den Kunden, den Betreuern und dem Team. Er ist auch an erster Stelle für das Projekt verantwortlich.

Seine zentralen Aufgaben liegen in der Verwaltung des Projekts, d.h. er plant den Ablauf des Projekts und der einzelnen Phasen, er bewertet und kontrolliert abschließend als letzte Instanz alle erstellten Dokumente vor ihrer Freigabe und kümmert sich um die Verteilung der einzelnen Arbeitspakete während des Projekts.

Architekt: Michael Schneidt (Stellvertreter: Daniel Brüderle)

Der Architekt ist zuständig für die Architektur des zu entwickelnden Softwaresystems, die Beschreibung der grundlegenden Komponenten und deren Zusammenspiel innerhalb des Softwaresystems. Er trifft die grundlegenden Entscheidungen über die Art und den Aufbau der Software-Architektur und überwacht deren Einhaltung während der Projektlaufzeit. Die Architektur soll die Einhaltung der unter Abschnitt ?? genannten Softwarequalitäten sicherstellen.

Administrator: René Rehn (Stellvertreter: Michael Schneidt)

Der Administrator ist in der Projektarbeit zuständig für die gesamte Infrastruktur auf Software- sowie auch auf Hardware-Ebene. Er ist für den Betrieb und die Wartung der Server und den Betrieb der darauf ausgeführten Software verantwortlich. Weiterhin sorgt er für die Bereitstellung der nötigen Software-Infrastruktur, d.h. er sorgt dafür, dass die zur Verfügung stehende Software, wie z.B. Werkzeuge für Entwicklung, Test und Entwurf, die zur Realisierung des Projekts benötigt wird, für das Projekt eingerichtet und betriebsbereit ist..

Qualitätsmanagement: Michael Hahn, Firas Zoabi

Der Qualitätsmanager sorgt dafür, dass alle Dokumente und auch Software-Komponenten die bestmögliche Qualität besitzen. Diese Qualität soll durch die Definition und Einhaltung von Richtlinien, durch die ständige Durchführung von Qualitätskontrollen und durch die Erstellung von einheitlichen Dokumentvorlagen erreicht werden. So soll während des gesamten Projektverlaufs die Qualität der Dokumente und Software-Einheiten maximiert werden.

Entwickler: Michael Schneidt, René Rehn, Daniel Brüderle, Michael Hahn, Firas Zoabi, Xi Tu

Als Entwickler arbeiten alle Teammitglieder außer dem Projektleiter. Ein Entwickler arbeitet dabei im Rahmen der vom Projektleiter erteilten Aufgaben an Dokumenten und Software-Komponenten.

3 Managementprozess

3.1 Managementziele und -prioritäten

3.2 Annahmen, Abhängigkeiten und Einschränkungen

3.3 Risikomanagement

Aufgrund der geforderten sehr breiten Verwendungsmöglichkeiten im Rahmen der wissenschaftlichen Workflows besteht ein hohes Unsicherheitspotential im Bezug auf die Implementierungsdauer und auf sich ändernde Anforderungen.

Tabelle 2: Übersicht der Risiken und geplanten Gegenmaßnahmen

Risiko	%	Kosten	Gegenmaßnahmen
Mitarbeiter fällt zeitweise aus	15	2 MM	Wöchentliche Treffen (Informationsaustausch), Ernennung von Stellvertretern und Erstellung von internen Dokumentationen
Mitarbeiter fällt dauerhaft aus	10	4 MM	Rücksprache mit den Kunden
Änderung der Anforderungen	20	1-5 MM	genaue Anforderungserhebung und regelmäßige Rücksprache mit den Kunden
Anforderungen werden nicht oder nur teilweise erfüllt	30	3 MM	ausgiebige Tests, regelmäßige Rücksprache mit den Kunden und Prototyping
Ausgewähltes Werkzeug stellt sich als unbrauchbar heraus	20	0,5 MM	genaue Evaluation der Werkzeuge vor ihrem Einsatz
Nicht realisierbare Anforderungen	10	4 MM	frühzeitige Evaluierung der einzusetzenden Technologien und Rücksprache mit den Kunden
Chaotische Änderungen der Dokumente	10	2 MM	ausgiebige Dokumentprüfung durch Qualitätssicherung und Projektleiter Konfigurationsverwaltung mit Subversion

3.4 Projektüberwachung

3.4.1 Berichtswesen

3.4.2 Qualitätssicherung und Konfigurationsmanagement

Um die Qualität der erstellten Dokumente gewährleisten zu können wird jedes Dokument bzw. jede Software-Einheit von zwei unabhängigen Instanzen geprüft. Zuerst wird die Software-Einheit durch die Qualitätsmanager auf formale Aspekte wie Style-Guide Konformität oder die Einhaltung des vorgeschriebenen Layouts hin untersucht. Anschließend wird die SE (Software-Einheit) falls sie Fehler enthält (positive QS-Prüfung) mit entsprechenden Kommentaren zurück an die Verfasser geschickt. Eine formal korrekte (negativ QS-geprüfte) SE wird an den Projektleiter weitergeleitet, der abschließend den Inhalt und die Korrektheit der gesamten SE prüft. Der Projektleiter leitet positiv geprüfte SE mit entsprechenden Kommentaren ebenfalls zurück an die Verfasser, der dann die entsprechenden Korrekturen ausführt. Negativ geprüfte SE werden vom Projektleiter freigegeben und erhalten so Auslieferungsstatus. Wird eine bereits freigegebene SE verändert oder an neue Bedürfnisse angepasst muss sie noch einmal den gesamten QS-Prozess durchlaufen. Sollte der Projektleiter eine SE prüfen, die seiner Meinung nach so viele Fehler oder Mängel enthält das eine Korrektur sich nicht lohnt, so kann er die SE verwerfen, d.h. die Verfasser erhalten eine Nachricht und müssen die SE neu verfassen. Eine detaillierte Beschreibung des gesamten Ablaufs des QS-Prozesses zeigt Abbildung 3.

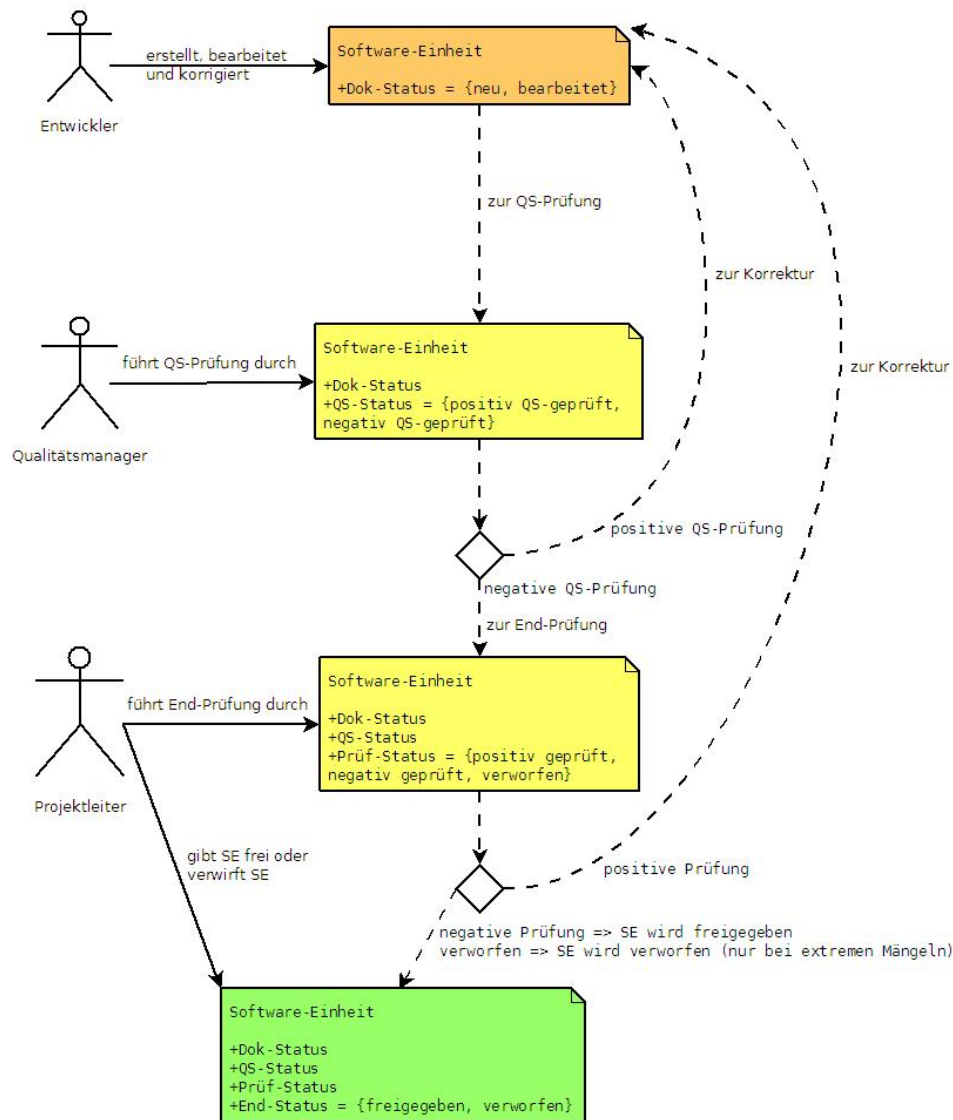


Abbildung 3: Prüfungsablauf für die Qualitätssicherung

Die Konfigurationsverwaltung ist ein wichtiger Bestandteil der Infrastruktur eines Projekts. In diesem Projekt wird zur Realisierung das Konfigurationsverwaltungssystem *Subversion* genutzt. Abbildung 4 zeigt das grundlegende Modell der Konfigurationsverwaltung und die Verwendung von Subversion (SVN). Hierzu werden verschiedene Umgebungen eingerichtet, die durch gewisse Rahmenbedingungen für bestimmte Tätigkeiten ausgelegt sind und so z.B. speziell für den Test oder die Implementierung eingerichtet sind. Die Referenz-Umgebung bildet die Zentrale der Konfigurationsverwaltung und wird durch Subversion realisiert. In Abbildung 5 wird der Entstehungsprozess von SE aufgezeigt und welche Umgebungen dabei durchlaufen werden.

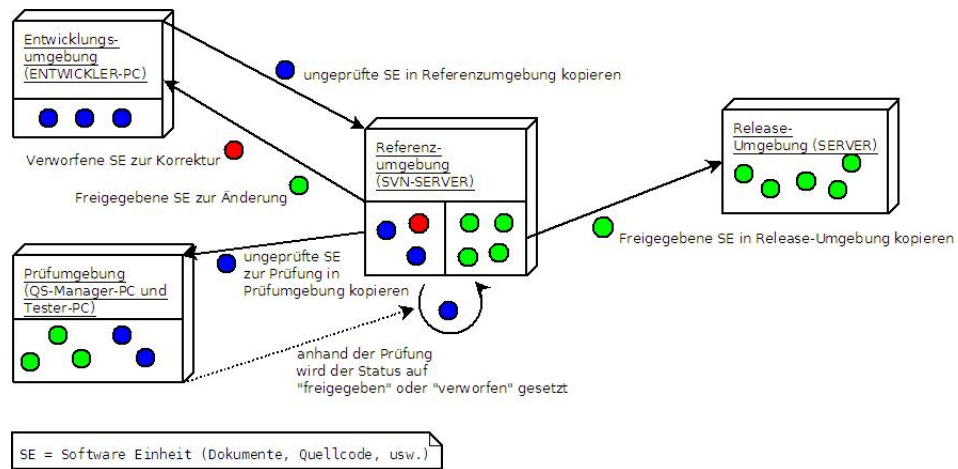


Abbildung 4: Das grundlegende Modell der Konfigurationsverwaltung

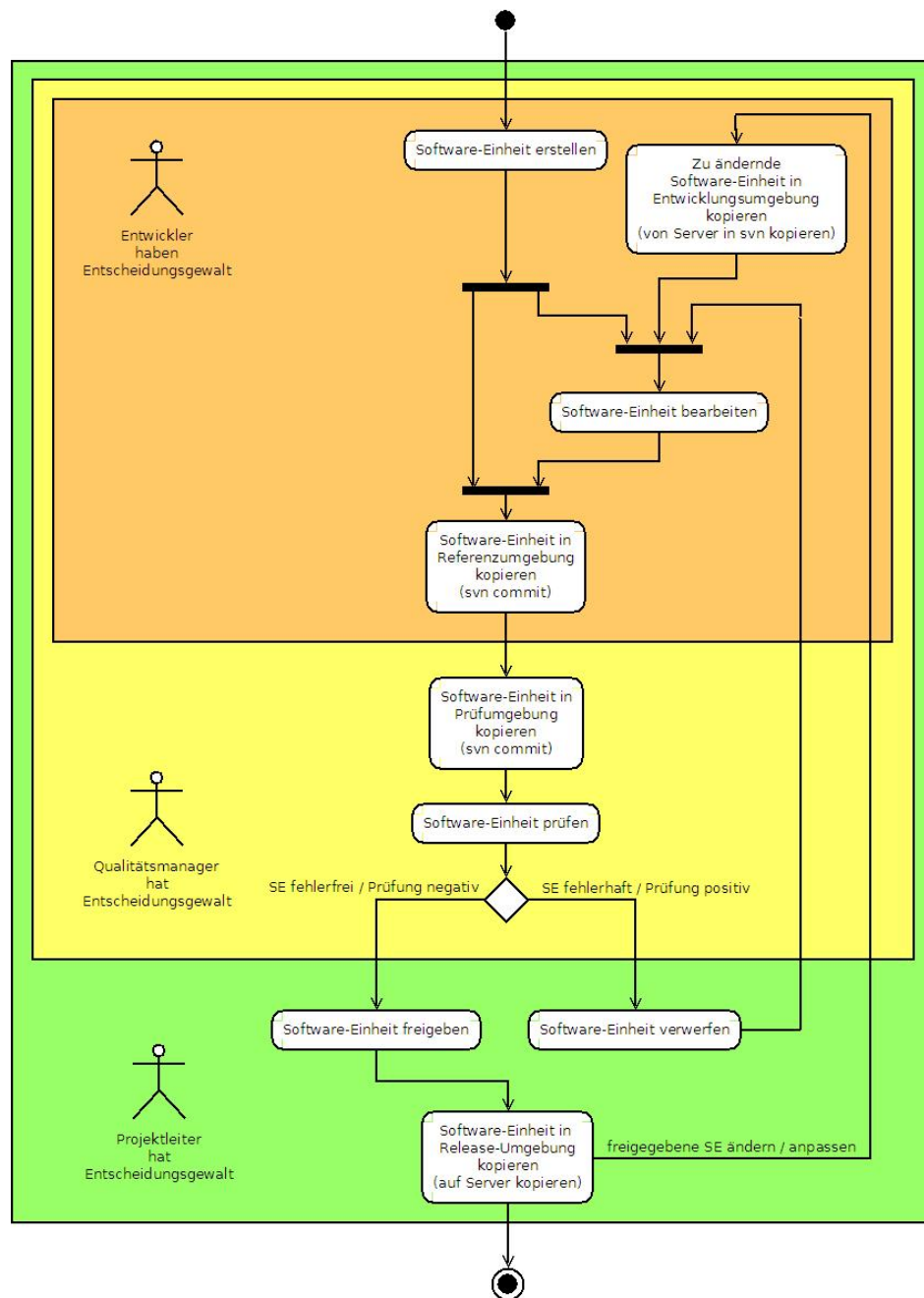


Abbildung 5: Entstehungsprozess für Software-Einheiten

3.5 Mitarbeiter

4 Technische Prozesse

4.1 Methoden, Werkzeuge und Techniken

4.1.1 Technologien bzw. Werkzeuge für die Entwicklung

Es werden die Verschiedene Technologien in Entwicklung eingesetzt:

Hudson <https://hudson.dev.java.net/>

ApacheODE <http://ode.apache.org>

Eclipse-Bpel-Designer <http://www.eclipse.org/bpel>

IBM-DB2 <http://pub...w/v9r5/index.jsp>

4.1.2 Unterstützende Werkzeuge im Projektarbeit

Subversion <http://svnbook.red-bean.com>

Maven <http://svnbook.red-bean.com>

Lyx xxxxx

L^AT_EX xxx

Eclipse-Subversion xxxxx

TortoiseSVN xxx

4.2 Dokumentationsplan

4.3 Unterstützende Projektfunktionen

5 Arbeitspakete, Zeitplan und Budget

5.1 Beschreibung der Phasen

In diesem Abschnitt wird jede der definierten Phasen und die darin enthaltenen Aufgaben kurz näher erläutert.

5.1.1 Analyse

Das Ziel der Analysephase ist es durch Kundengespräche die funktionalen und nichtfunktionalen Anforderungen zu erheben, die der Kunde an das zu entwickelnde Softwaresystem stellt. Das Ziel ist es einen Einblick in den momentanen Zustand des Arbeitsbereichs für den die Software entwickelt werden soll zu bekommen, um bereits realisierte Anforderungen aufzunehmen.

Am Ende der Analysephase wird aus den gewonnenen Erkenntnissen ein Anforderungskatalog erstellt, der als Referenz für die spätere Spezifikation der Software verwendet wird.

5.1.2 Einarbeitung

Da wenige bis gar keine Ansätze für Teilbereiche des Projekts wie z.B. die Verwendung von BPEL in Scientific Workflows existieren, müssen alle vorhandenen Ansätze evaluiert oder z.T. auch neue Ansätze geschaffen werden. Die Einarbeitungsphase dient dazu sich in die verschiedenen Themenbereiche einzuarbeiten und festzustellen in welchem Umfang Erweiterungsmaßnahmen notwendig sind.

5.1.3 Spezifikation

In dieser Phase wird die zu entwickelnde Software genau spezifiziert, d.h. in einem Dokument, der Spezifikation, werden alle funktionalen und nichtfunktionalen Anforderungen an die Software und ihre Schnittstellen präzise, vollständig und überprüfbar definiert. Ebenso werden spätere Abläufe im Umgang mit der Software und ihren Funktionen beschrieben. Die Spezifikation dient darüber hinaus als Referenz für viele nachfolgende Dokumente, wie z.B. das Handbuch und den Entwurf und sie ist unerlässlich für die Testphase und die spätere Wartung oder für spätere Erweiterungen der Software. Anhand der Spezifikation kann sich auch der Kunde davon überzeugen, dass wirklich seine Anforderungen umgesetzt werden.

5.1.4 Entwurf

In der Entwurfsphase wird die Architektur der Software festgelegt und in verschiedenen Granularitäten in den Dokumenten Grob-Entwurf (Architektur des Gesamtsystems) und Fein-Entwurf (Architektur der einzelnen Module) festgehalten.

5.1.5 Codierung

In der Codierungsphase werden die Anforderungen aus der Spezifikation mit der Architektur des Entwurfs umgesetzt. Am Ende dieser Phase existiert eine ausführbare Software.

5.1.6 Test

Die Testphase dient dazu die Software gegen die Spezifikation zu prüfen, d.h. alle Anforderungen aus der Spezifikation werden mit geeigneten Testfällen überprüft und die Software entsprechend korrigiert, falls sie nicht die gewünschten Ergebnisse liefert. Dazu wird eine Reihe von Testfällen definiert und deren Ausführung in einem Testprotokoll aufgezeichnet. Diese zwei Dokumente sind auch für den Kunden wichtig, da sie als Referenz für seine internen Tests während der Abnahme genutzt werden können.

5.1.7 Abnahmetest

Am Ende der Testphase erfolgt die Abnahme des Projekts und aller genannten Dokumente (siehe Tabelle 1). Dem Kunden wird damit Zeit zur Verfügung gestellt sicherzustellen, dass das von uns gelieferte Produkt seinen Anforderungen entspricht. Ist das der Fall, dann gilt das Projekt als abgeschlossen, andernfalls werden die nötigen Nachbesserungen oder Änderungen durchgeführt.

5.1.8 Puffer

Ein extra Zeitfenster vor dem endgültigen Abgabetermin Ende Mai 2010. In dieser Phase können noch kleinere Änderungswünsche, die der Kunde nach dem Abnahmetest eventuell hat, umgesetzt werden.

5.2 Arbeitspakete

Die im vorigen Abschnitt definierten Phasen werden nun in Arbeitspakete gegliedert und aufgeteilt. Die nachfolgende Tabelle enthält alle Arbeitspakete und eine kurze Beschreibung der darin enthaltenen Aufgaben.

Id	Bezeichnung	Aufgaben	Dokumente
<i>anly#proplan</i>	Projektplan	Erstellung des Projektplans	Projektplan
<i>anly#angeb</i>	Angebot	Erstellung des Angebots für die Kunden	Angebot

Id	Bezeichnung	Aufgaben	Dokumente
<i>einb#tools</i> <ul style="list-style-type: none"> • #1 • #2 • #3 • #4 • #5 • #6 	Einarbeitung in Tools	Einarbeitung in: <ul style="list-style-type: none"> • Apache ODE (einb#tools#1) • Eclipse BPEL Designer • Maven • Hudson • SVN • IBM DB2 (einb#tools#6) 	Foliensammlung als Nachschlagewerk
<i>einb#konzept</i> <ul style="list-style-type: none"> • #1 • #2 • #3 • #4 	Einarbeitung in Konzepte	Einarbeitung in: <ul style="list-style-type: none"> • BPEL (einb#konzept#1) • Apache ODE • Eclipse BPEL Designer • BPEL/SQL (einb#konzept#4) 	Berichte als Grundlage für Spezifikation und Implementierung bzw. Erweiterung der Technologien.
<i>spez#erstell</i>	Spezifikation erstellen	Erstellung der Spezifikation	Spezifikationsdokument
<i>spez#rev1</i>	1.Review der Spezifikation	Interner Review der Spezifikation	Reviewprotokoll
<i>spez#kor1</i>	1.Korrektur der Spezifikation	Korrektur der Spezifikation nach dem 1.Review	Korrigiertes Spezifikationsdokument
<i>spez#rev2</i>	2.Review der Spezifikation	Weiterer Review der Spezifikation diesmal mit den Kunden	Reviewprotokoll
<i>spez#kor2</i>	2.Korrektur der Spezifikation	Korrektur der Spezifikation nach dem 2.Review	Kundenfassung der Spezifikation
<i>grEntw#erstell</i>	Grobentwurf erstellen	Erstellung des Grobentwurfs (zusammenhang zwischen Komponenten)	Dokument des Grobentwurfs
<i>grEntw#rev</i>	Review des Grobentwurfs	Interner Review des Grobentwurfs	ReviewProtokoll
<i>entw#kor</i>	Korrektur des Entwurfs	Korrektur des Entwurfs nach dem Review	Grobentwurf
<i>feinEntw#erstell</i>	Feinentwurf erstellen	Erstellen des Feinentwurfes	Dokument des Feinentwurf
<i>code#</i>			
<i>code#</i>			
<i>code#</i>			
<i>code#</i>			

Tabelle 4: Phasen des Projekts

Phase	Von / Bis	Aufwand
Analyse	13.05.2009-17.07.2009	306 h
Einarbeitung	17.07.2009-17.09.2009	208 h
Spezifikation	07.08.2009-15.10.2009	544 h
Entwurf	15.10.2009-15.12.2009	628 h
Implementierung	13.11.2009-25.02.2010	880 h
Test	15.12.2010-17.04.2010	544 h
Abnahmetest	17.04.2010-27.04.2010	–
Puffer	17.04.2010-13.05.2010	250 h

Id	Bezeichnung	Aufgaben	Dokumente
<i>handb#erstell</i>	Handbuch erstellen	Erstellung eines Handbuchs	Handbuch
<i>handb#rev</i>	Handbuch review	Interner review des Handbuchs	Review Protokoll
<i>handb#kor</i>	Korrigieren des Handbuchs	Handbuch Korrektur	Kundenfassung Handbuch
<i>test#mod</i>	Modultest	Planung und Durchführung des Modultests mit anschließender Korrektur der entdeckten Fehler	Testprotokoll
<i>test#int</i>	Integrationstest	Planung und Durchführung des Integrationstests mit anschließender Korrektur der entdeckten Fehler	Testprotokoll
<i>test#sys</i>	Systemtest	Planung und Durchführung des Systemtests mit anschließender Korrektur der entdeckten Fehler	Testpotokoll
<i>puff#abn</i>	Abnahme	Abnahme durch die Kunden	
<i>puff#erw</i>	Erweiterung/Anpassung	Durchführung möglicher Erweiterungen und/oder Anpassungen	

5.3 Abhängigkeiten

a

5.4 Ressourcenanforderung

5.5 Zuteilung des Budgets und der Ressourcen auf Projektfunktionen und Aktivitäten

5.6 Zeitplan

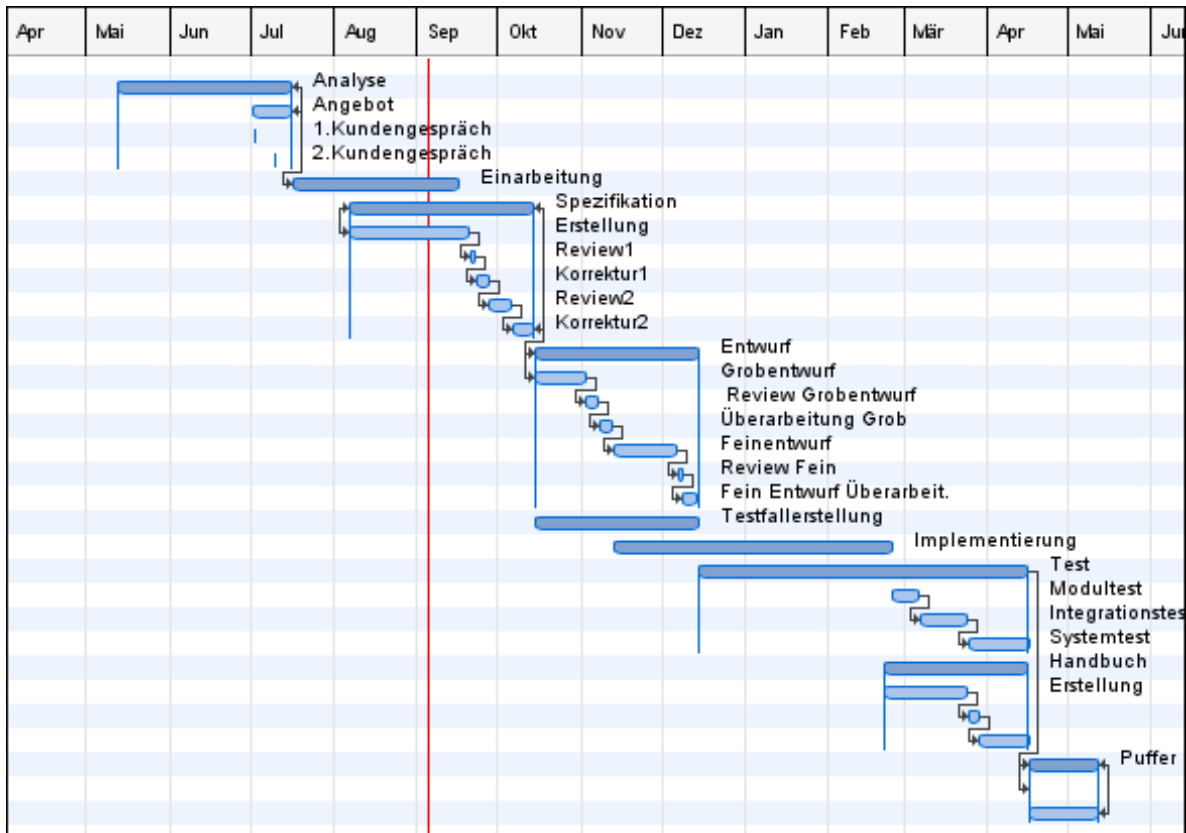


Abbildung 6: Gantt-Diagramm des Projektverlaufs

6 Änderungsgeschichte

- **Version 0.1**, 22. Juli 2009: Erstellung des Dokuments.
- **Version 0.2**, 02. August 2009: Überarbeitung des Dokuments.