Hochschule für Technik Zürich

Semesterarbeit Vacation Planner

Projektplanung

Author:
Raffael Schmid

Dozent:
François Bolay

Inhaltsverzeichnis

Abbildungsverzeichnis

Einleitung

Im Rahmen einer Semesterarbeit versuche ich als Basis zu einem Ressourcenplanungs-Tool ein Ferienplaner zu erstellen. Im Folgenden werden die ausgearbeiteten Daten und Informationen im Rahmen des Projekt-Managements dargestellt.

Kapitel 1

Ausgangslage[1]

1.1 Problemstellung

In der heutigen Zeit gibt es viele Teams und Kleinunternehmen, die Ihre Ressourcenplanung noch immer auf Basis von Papier und Excel durchführen. Dies fängt bei einfachen Problemstellungen wie Ferienplanung an, und hört bei Komplexen Szenarien wie die Planung und Organisation von Ressourcen in Schichtbetrieben auf.

1.2 Vacation-Planner

Die Idee für einen Ressourcen-Planer stammt aus dem Umfeld des betreuenden Dozenten Beat Seeliger. Heruntergebrochen auf den Umfang einer Semesterarbeit von rund 120 Stunden war diese Problemstellung allerdings zu Umfangreich und man entschied sich, eine Problemstellung daraus zu erarbeiten. Daraus ergab sich die Projektidee "Ferienplaner".

1.3 Projektzielsetzung

Der Ferienplaner wird auf der Basis der Java-Plattform erstellt. Ziel der Arbeit ist es zusätzlich, eine Evaluation des auf der Sprache Scala[2] basierenden Webframeworks (Lift) durchzuführen, um danach entscheiden zu können, wie das Potential dieses Frameworks für Projekte in diesem Rahmen ist. Ansonsten werden bereits bewährte Komponenten verwendet.

[1] übernommen aus dem Projektauftrag
[2] http://www.scala-lang.org

1.4 Projektvarianten

1.4.1 Backend

Im Bereich der Softwareentwicklung auf der Java-Plattform gibt es unterschiedliche Frameworks, und immer wieder stossen neue Ideen und Ansätze hervor. Aktuell wird die Sprache Scala "gehyped". Darauf basierend existiert ein Webframework, welches viele neue Ideen und Konzepte integriert. Ein Teil der Semesterarbeit soll sich auch damit beschäftigen, ob sich dieses Framework und die Sprache Java momentan für eine weitere Betrachtung lohnen, oder ob die Zeit (noch) nicht reif dafür ist. Es werden in etwa ein viertel der zur Verfügung stehenden Zeit in die Evaluation investiert.

Scala, Lift-Framework

Der erste Milestone beschäftig damit, ob auf der Basis von Liftweb fortgefahren wird, oder ob auf ein anderes Framework ausgewichen wird.

Groovy, Grails

Groovy und Grails sind bereits bewährtere Partner, und werden verwendet, falls es fürs Scala-Framework ein "nogo" gibt.

1.4.2 Frontend

Im Bereich des Frontends gibt es zwei verschiedene Möglichkeiten, für die eine Evaluation sowie einen Variantenentscheid durchgeführt werden muss.

HTML, Javascript, CSS

Der klassische Weg mit HTML, Javascript und CSS ist für die Entwicklung von Administrations-Oberflächen weniger effizient als RIA[3] Produkte. Ausschlaggebend für einen Entscheid ist die Evaluation von Komponenten zur Darstellung von Gantt-Charts[4] respektive Scheduling-Charts.

Flex

Flex ist ein Framework von Adobe und bietet vielseitige Unterstützung bei der Erstellung von RIA Oberflächen. Dabei wird Flex-Code in Actionscript-Code umgewandelt und dann für den Flash-Player kompiliert. Die Tatsache,

[3] Rich Internet Applications
[4] Charts zur darstellung von Ressourcen, usw.

dass für die Anwendung nur ein Browser mit installiertem Flash-Plugin notwendig ist und deshalb keine Cross-Browser Probleme[5] auftreten, macht die Entwicklung von Anwendungen sehr effizient.

1.5 Projektrandbedingungen

Das Projekt "Vacation Planner" wird im Rahmen der Semesterarbeit an der Fachhochschule für Technik in Zürich durchgeführt und unterliegt deren üblichen Bedingungen.

1.6 Wirtschaftlichkeit

1.6.1 Theoretische Kosten

Posten	Stunden	Preis	Total
Personalaufwand	1 x 140h	125 SFr/h	17'500 SFr
Projektbetreuung	1 x 10h	200 SFr/h	2'000 SFr
evt. Lizenzkosten			2'000SFr
Total			21'500 SFr.

1.6.2 Nutzen

Ziel des End-Produktes ist es, das Produkt im Rahmen eines SaaS[6] zu vertreiben.

1.7 Konsequenzen

1.7.1 Bei Realisierung

Dieses Projekt ist mit sehr geringen Risiken verbunden. Da weder finaniziell noch personell ein grosser Aufwand geplant ist. Das grösste Risiko beinhaltet die Problematik, in der geplanten Zeit nicht über die Runden zu kommen.

1.7.2 Bei Nichtrealisierung

Nichtrealisierung ist nicht möglich. Da die Konsequenzen einer schlechten Note nicht tragbar sind;)

[5] Probleme die im Zusammenhang mit den Unterschiedlichen Browsern auftreten
[6] Software as a Service

1.7.3 Verspätete Realisierung

Verspätete Realisierung ist tendenziell auch keine Option.;)

Kapitel 2

Ubiquitous Language a.k.a Glossary

Wort	Definition
Backend	Webapplikationen bestehen grob gesagt aus Präsentationslogik und Businesslogik - zweites wird auch als Backend bezeichnet
Frontend	Webapplikationen bestehen grob gesagt aus Präsentationslogik und Businesslogik - erstes wird auch als Frontend bezeichnet

Kapitel 3

Projektorganisation

3.0.4 Aufbau

Im Organigramm wird versuch, die klassischen Rollen eines Software-Projektes auf die Semesterabeit anzuwenden.

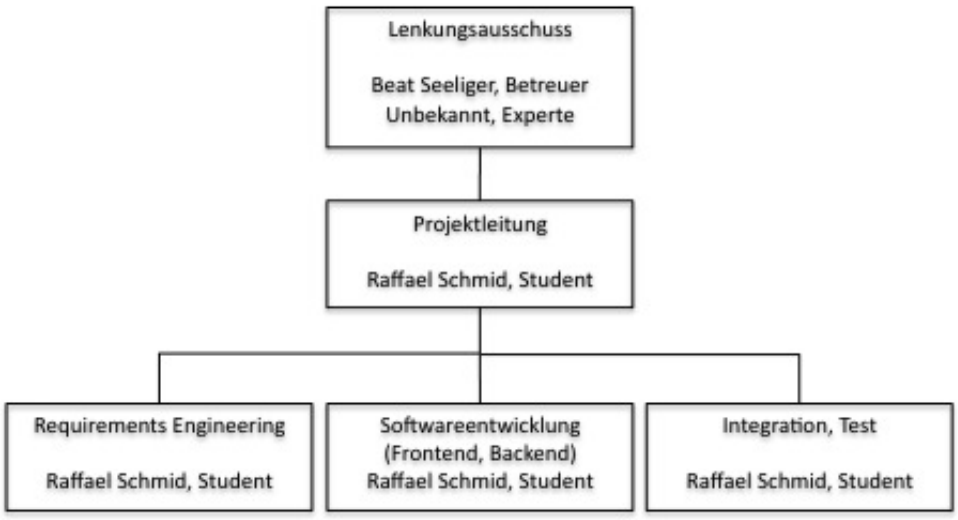

Abbildung 3.1: Organigramm

3.1 Rollen

Die Semesterarbeit ist für den Rahmen eines strukturierten, professionellen Projektmanagements zu wenig umfangreich - normalerweise entspricht der zeitliche Aufwand von 120h dem Aufwand für Projektmanagement bei einem Mini-Projekt. Die Schwierigkeit bei der Strukturierung der Organisation und Rollen liegt darin, dass der Student mehrere Rollen - ausgenommen die des Kunden - selber übernimmt.

3.1.1 Kunde, Auftraggeber

Bei der Semesterarbeit geht es in erster Linie um das Erarbeiten von Wissen auf der Basis der Erstellung eines Prototypen für eine Ferienplanung. Die Idee stammt aus der Firma des Dozenten - der Kunde, Aufraggeber ist grundsätzlich fiktiv.

3.1.2 Requirements Engineer, Entwicklung, Tester

Die Rollen des Requirements Engineers, Entwicklers, Testers werden alle vom Studenten übernommen. Die sehr subektive Perspektive ist für das Endresultat ein Risiko.

Kapitel 4

Stakeholderanalyse

4.1 Identifikation der Stakeholder

Folgende Tabelle zeigt eine Übersicht aller Stakeholder mit der Wichtigkeit fürs Projekt, den Interessen welche jeder Stakeholder am Projekt hat und die zu treffenden Massnahmen, damit sich der entsprechende Stakeholder engagiert.

Stakeholder	Wichtigkeit [1-10]	Interesse	Massnahmen
Student	10	sehr gross: Erwartet gutes Resultat, gute Noten	Effizientes und Effektives Arbeiten, Konzentration aufs Wesentliche
Dozent	5	gross: Erwartet Resultat	Guter Informationsfluss: Student - Dozent
Schule	2	Gibt Rahmenbedingungen vor	-
Kunde	1	nicht vorhanden	-

Kapitel 5

Risikoanalyse

5.1 Beurteilung, Massnahmen

Folgend eine Liste mit den Risiken für das Projekt mit den zu treffenden Massnahmen:

Terminliche Risiken

Risiko	Eintritts-wahrscheinlichkeik	Auswirkungsgrad	Risikoeinstufung	Massnahmen
Optimistischer Zeitplan	mittel	hoch	hoch	Projektrapportierung, Änderung der Ziele beim Design Review wenn notwendig
Fehlende, ungenügende Ressourcen	mittel	mittel	mittel	„
Terminverzug Lieferanten	klein	klein	klein	
Änderungsantrag	klein	klein	klein	-

Finanzielle Risiken

Risiko	Eintritts-wahrscheinlichkeik	Auswirkungsgrad	Risikoeinstufung	Massnahmen
Inflation	gross	klein	klein	-
Aufwandschätzung	gross	klein	klein	-

Menschliche Risiken

Risiko	Eintritts-wahrscheinlichkeik	Auswirkungsgrad	Risikoeinstufung	Massnahmen
Subjektive Perspektive des Studenten	gross	klein	mittel	Feedbacks einholen
Team-zusammensetzung	klein	gross	klein	-
Fehlendes Know How	klein	gross	klein	-
Fehlende Motivation	mittel	gross	mittel	Effizientes und Effektives Arbeiten

Technische Risiken

Risiko	Eintritts-wahrscheinlichkeik	Auswirkungsgrad	Risikoeinstufung	Massnahmen
Neue Technologie	hoch	hoch	sehr hoch	Gründliche Evaluation des Lift-Webframeworks, Kriterienkatalog aufstellen
Mangelhafte Lieferqualität	mittel	mittel	mittel	vor Auslieferung Abnahmetest, Akzeptanztest, Funktionstest (Auslieferung ist nicht bestandteil der Semesterarbeit: Ziel ist ein Prototyp)

Kapitel 6

Requirements Engineering

6.1 Einleitung

Im folgenden Abschnitt soll die Requirements-Analyse anhand von Use Cases durchgeführt werden. Requirements Engineering ist ein iterativer Prozess, als Basis für die Aufwandschätzung wird im Projektplan der erste Schritt durchgeführt.

6.2 Use Case

6.2.1 Administrator

Administrator ist die Rolle, welche Mitarbeiter erfasst und für diese die Ferienwünsche beantwortet. Normalerweise handelt es sich bei Administratoren um Team-, Abteilungs- oder Projektleiter. Sie haben bei der Einteilung von Ferien die volle Entscheidungskompetenz. Folgende Use Cases sind für sie vorgesehen:

Projekt eröffnen

Nach dem Login erfasst der Administrator (Project Manager, Teamleiter, usw.) für sein Team ein Projekt. Grundsätzlich gehört ein Projekt mehreren Administratoren, in einem 2. Projekt wird die zusätzliche Möglichkeit implementiert, anderen Benutzer Administrator-Rechte fürs eigene Projekt zu geben.

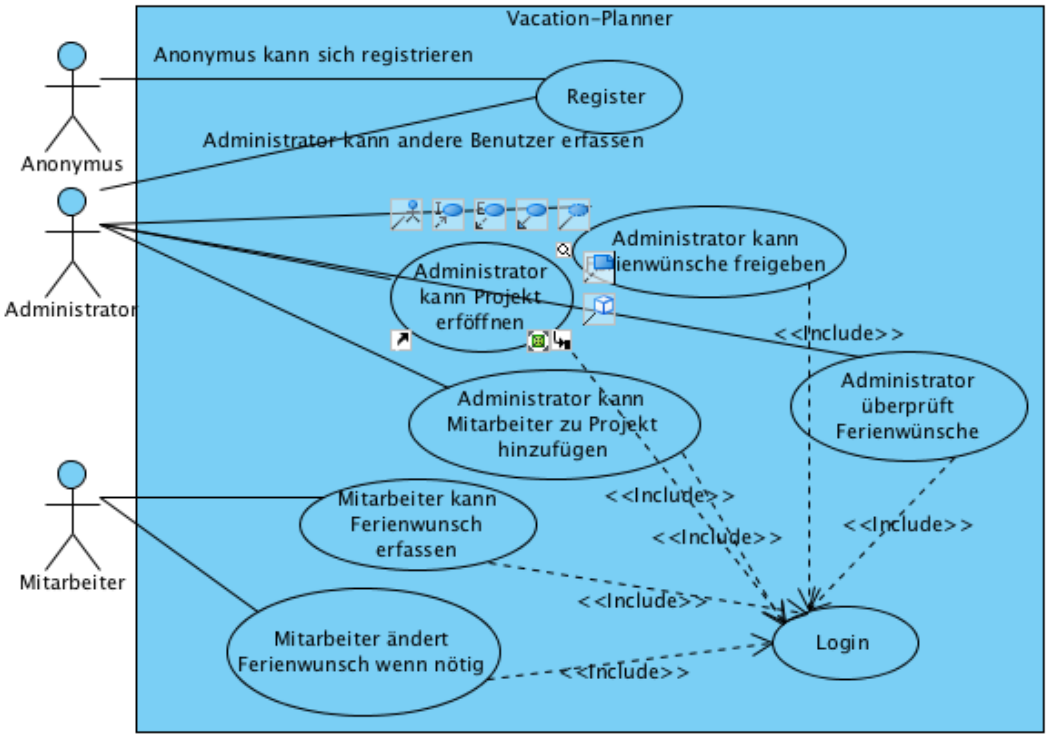

Abbildung 6.1: Use Case Diagramm

Mitarbeiter in Projekt erfassen

Projekte alleine sind leere "Behälter" für Mitarbeiter. Um Mitarbeiter ins eigene Projekt zu nehmen, sucht der Administrator nach einem bestehenden User. Sofern es den gesuchten Benutzer im System noch nicht gibt, wird er durch den Administrator neu erfasst. Der neu erfasste Benutzer wird mittels Mail auf diese Aktion aufmerksam gemacht.

Ferienwünsche bearbeiten

Die vom Mitarbeiter erfassten Ferien können durch den Administrator bezüglich Status und Termin verändert werden. Die Bewilligung von Ferien wird mittels des Status confirmed / bestätigt erteilt. Ansonsten gibt es folgende Möglichkeiten:

- erwünscht - requested

- abgelehnt - rejected
- bestätigt - confirmed

6.2.2 Mitarbeiter

Ferienwunsch erfassen

Der Mitarbeiter kann seine Ferienwünsche pro Projekt erfassen. Diese befinden sich zu Beginn im Status "requested" und können vom Administrator in die Status "rejected" und "confirmed" geändert werden.

Ferienwunsch bearbeiten

Sollte ein Ferienwunsch abgelehnt werden, kann er erneut bearbeitet werden, was eine erneute Anfrage beim Administrator zur Folge hat.

6.2.3 Mail Versand

Verschiedene Aktionen und Status-Änderungen haben einen Mail-Versand zur Folge:

Mitarbeiter erfassen

Wird ein Mitarbeiter erfasst - egal ob selbständig oder durch einen Administrator - wird ein Mail versendet, in welchem die Erstellung bestätigt werden muss. Erst dadurch gelangt der User / Mitarbeiter vom Status "inaktiv" in den Status "aktiv".

Mitarbeiter hinzufügen

Wird ein Mitarbeiter einem Projekt hinzugefügt, wird eine Notifikation an den Mitarbeiter versendet.

Ferien erfassen

Werden durch den Mitarbeiter Ferien erfasst, erhält der Administrator ein Mail, bei welchem er auf diese Aktion aufmerksam gemacht wird. Er hat somit die Möglichkeit, die Wünsche schnellstmöglich zu bestätigen.

Statusänderung Ferien

Wird eine Statusänderung durch den Administrator gemacht, wird ebenfalls ein Mail versendet.

Kapitel 7

Aufwandschätzung

7.1 Einleitung

Semesterarbeiten haben in Sachen Projektmanagement nicht die selben Probleme, wie sie bei anderen Projekten mit vielen unterschiedlichen Resourcen, Rollen und Personen natürlich entstehen. Zur schätzung wird aus folgenden Gründen die "Informelle Expertenschätzung"[1, S. 60] (bottom-up) verwendet:

- keine formale Vorgabe
- Schätzungen werden durch die am besten geeignete Person (dem Studenten und evt. Dozenten in der Rolle als Experten) durchgeführt
- kleines Projekt mit hohem Zeitdruck
- hohe Akzeptanz

7.2 Schätzungen

	Requirement	**Zeitaufwand in h**
Diverses	Project Management	5h
	Dokumentation	25h
	Evaluation, Einarbeitung Lift Framework	16h
	Project Setup	4h
	Testing, Integration	5h
Total		**65h**
Frontend	**Start Screen**	
	Login-Maske	2h
	Registration-Maske	3h
	Mitarbeiter Ansicht	
	Evaluation Scheduler (Gantt Chart) für Ferienübersicht	10h
	Ferien beantragen, löschen, editieren	10h
	Administrator Ansicht	
	Mitarbeiter hinzufügen, löschen	10h
	Team hinzufügen, löschen	10h
	Diverses	5h
Total		**50h**
Backend	Webservice: CRUD[1] Mitarbeiter	10h
	Webservice: CRUD Team	10h
	Webservice: CRUD Vacation	10h
	Unit-, Func.-, Integr.-Testing	10h
	Diverses	15h
Total		**55h**
Gesamt		**170 h**

Bei einer Gesamtzeit von erwarteten 170 Stunden übersteigt der Aufwand die Vorgabe bei weitem. Das war für eine Semesterarbeit allerdings durchaus zu erwarten.

Kapitel 8

Projektplanung

8.1 Phasen

Phase	von	bis	Milestone-Dokumente	Output
Vorbereitung	25.02.2010	10.3.2010	Aufgabenstellung, Kickoff-Protokoll	
Evaluation Framework (Scala, Lift)	11.03.2010	07.04.2010	Framework Analyse Dokumentation	
Project Setup	08.04.2010	09.04.2010	-	Development Environment under Source Control
Frontend Design, Review	14.04.2010	15.06.2010	Review Document	Source Code
Backend Design, Test	16.06.2010	17.08.2010	Review Document	Source Code, Tests, Test Results
Integration, Test	18.08.2010	19.08.2010	-	Deployment
Dokumentation, Abschlussarbeiten, Präsentation, Bewertung	20.08.2010	03.09.2010	Dokumentation, Präsentation	Source Code, Applikation

8.2 Termine

Datum	Termin	Deliverables
25.02.2010	Projektbeginn	Projekteingabe
10.03.2010	Kickoff Meeting	Protokoll
24.06.2010	Design Review	Protokoll
18.08.2010	Abgabetermin	Software, Dokumentation Design-Entscheide
03.09.2010	Präsentation	Präsentation, Protokoll

Der folgende Abschnitt zeigt ein ein Ausschnitt aus der "MS Project" Datei, welche für die Rapportierung und Kontrolle parallel bearbeited wird. Das Projekt wird in 5 verschiedene Phasen unterteilt:

- Konzeptphase
- Definitionsphase
- Entwurfs- und Entwicklungsphase Frontend
- Entwurfs- und Entwicklungsphase Backend
- Fertigstellung, Dokumentation, Präsentation

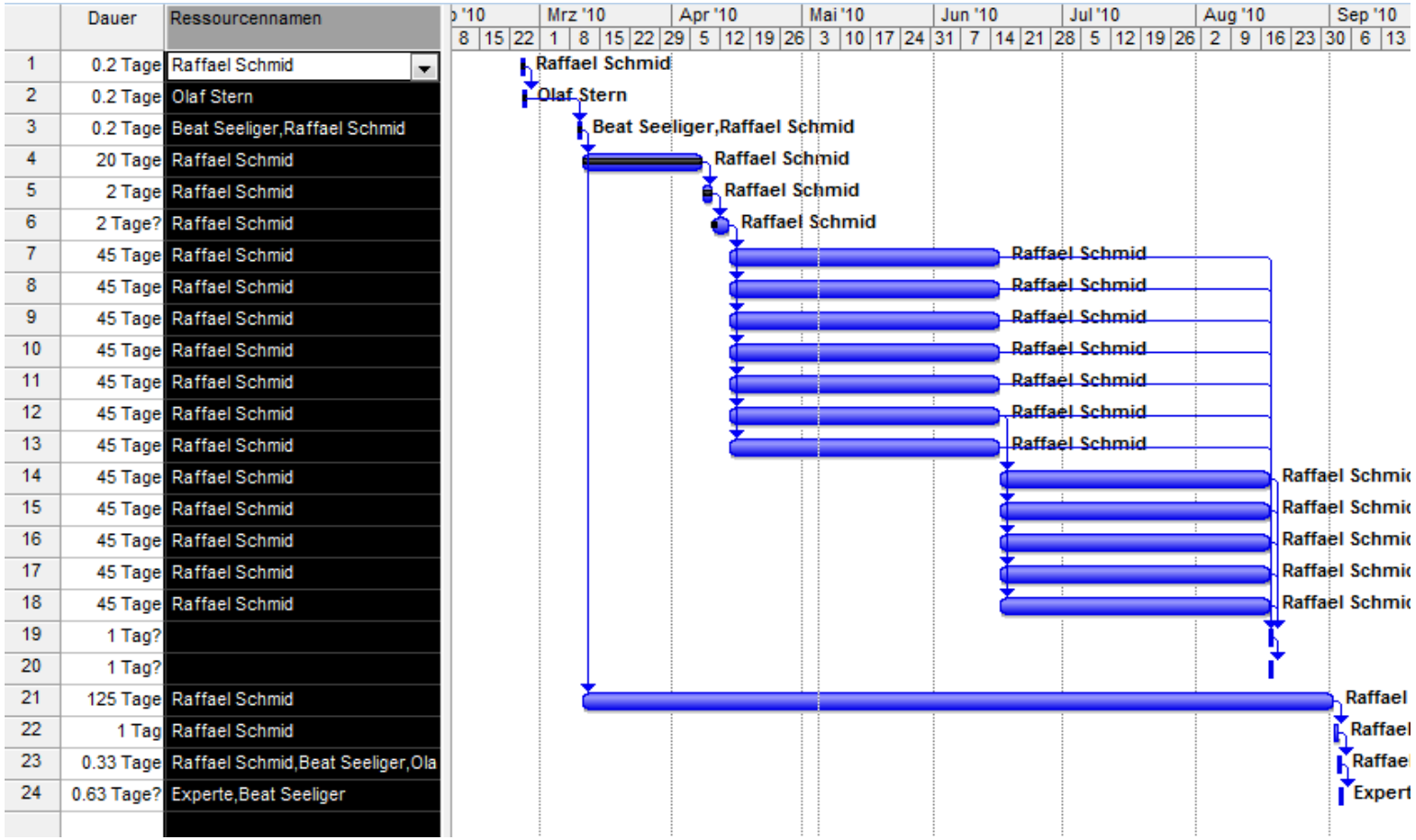

Abbildung 8.1: Projektplanung MS Project 2007 (stand: 5. Mai 2010)

Am ende Jeder Phase gibt es einen Meilenstein (Status-Meeting mit Dozenten, Protokoll, Journal der Tätigkeiten, Problemen und anstehenden Arbeiten.

Kapitel 9

Projektkosten

Projektkosten (Budget, Kalkulationsart) werden für die Semesterarbeit nicht erarbeitet. Diese Punkte sind aus Sicht der Projektleitung nicht angebracht [1].

[1] Aufwand / Ertrag sind aus finanzieller Sicht sowieso in einem schlechten Verhältnis;)

Kapitel 10

Projektrapportierung, Kontrolle

Bis zum Kickoff Meeting wurden verschiedene Ziele der Semesterarbeit definiert. Die Bewertung der Semesterarbeit basiert allerdings auf den Zielen, die beim Design-Review verabschiedet werden. Spätestens bis zu diesem Zeitpunkt muss eine Erfassung des Aufwandes, des Fertigstellungsgrads, des Restaufwands und eine Aussage darüber gemacht werden, ober der fixe Endtermin eingehalten werden kann. Ist dies nicht der Fall, müssen bestimmte Ziele aus dem Katalog gestrichen werden.
Für die Rapportierung des Fertigstellungsgrades wurde ein Microsoft Project 2007 Projekt aufgesetzt (siehe Projektplanung). Darin folgende Arten von Tätigkeiten rapportiert:

- Erledigte Tätigkeiten
- Aktuelle Tätigkeiten, Problemstellungen (als Basis für eine "wissenschaftliche" Dokumentation)
- Anstehende Tätigkeiten

Kapitel 11

Projektabschluss

Das Resultat respektive der Output der Semesterarbeit ist ein Prototyp. Es existieren Ideen für die weitere Verwendung im Bereich Software as a Service. Die dafür zu erreichende Produktreife ist nicht Bestandteil der Semesterarbeit. Die Software muss zuvor folgende Kriterien erfüllen:

11.1 Performancetest

Performancetests können mit der Unterstützung von Tools durchgeführt werden. Für das Testen des Vacation-Planners wird Grinder[1] verwendet. Grinder ist ein Load-Testing Framework und beinhaltet Support für HTTP[2] basierte Services.

11.2 Abnahmetest

Die Idee zum Ressourcenplaner, Ferienplaner stammt von einer Person aus dem Umfeld des betreuenden Dozenten. Die erwähnte Person wird den Abnahmetest durchführen.

11.3 Akzeptanztest

Unter kundenspezifischen Bedingungen wird getestet, wobei das fertige System durch Test-Benutzer getestet wird. Die Vorbereitung zu den Tests wird durch eine im Projekt involvierte Person durchgeführt.

[1] http://grinder.sourceforge.net
[2] http://de.wikipedia.org/wiki/Hypertext_Transfer_Protocol

Kapitel 12

Qualitätssicherung

Im folgenden werden die Qualitätssichernden Massnahmen in konstruktive, analytische und organisatorische Massnahmen eingeteilt. Die grösse des Projektes lässt es nicht zu, allzu viel im Bereich Qualitätssicherung zu tun.

12.1 Konstruktive Massnahmen

12.1.1 UML

Die Modellierung wird mit der Unified Modeling Language durchgeführt.

12.2 Analytische Massnahmen

12.2.1 Continuous Integration, Testing

Während der Entwicklungszeit wird als Continuous Integration Server Hudson[1] eingesetzt. Hudson checkt täglich den Code aus dem Git[2]-Repository aus und führt einen Build der Applikation durch, um anschliessend die Unit-Tests, Functional-Tests und Integration-Tests durchlaufen zu lassen. Zur statischen Sourcecode Analyse stellt Hudson eine Reihe von Plugins für bereits bestehende Tools zur Verfügung:

- FINDBUGS: FindBugs ist ein Open Source Programm ursprünglich von Bill Pugh und David Hovemeyer entwickelt, welches in Java-Code nach Fehlermustern sucht. Solche Fehlermuster deuten meist auf tatsch-liche Fehler hin. Das Programm wurde von der University of Maryland

[1] http://hudson-ci.org
[2] Git ist eine Software zur Source-Code Kontrolle: http://git-scm.com

aus initiiert, mittlerweile umfasst das Entwicklerteam fast ein Dutzend Personen

- PMD: Die Fehler, die PMD findet, sind typischerweise keine echten Fehler, sondern eher ineffizienter Code, d. h. die Software wird in der Regel trotzdem korrekt ausgeführt, wenn die Fehler nicht korrigiert werden. PMD findet auf Basis von statischen Regeln potentielle Probleme wie beispielsweise mögliche Bugs, toter Code, Überkomplizierte Ausdrücke, Suboptimaler Code, Klassen mit hoher zyklomatischer Komplexität

- CHECKSTYLE: Das Plugin Checkstyle ermöglicht eine automatische Überprüfung der Einhaltung von Coding Conventions bei der Erstellung von Code.

Die Reports der verschiedenen Plugins können auf einer Webseite angezeigt werden. Sofern die Tests (Unit-, Functional-, Integration-Tests) nicht erfolgreich durchlaufen werden, wird ein Mail an die Entwickler-Crew (in diesem Fall an den Studenten) versendet.

12.3 Organisatorische Massnahmen

Organisatorische Massnahmen wie Audits, Programmier- und Dokumentationsrichtlinien, Konfigurationsmanagement werden keine definiert. Das Glossar dient zum Verständnis der Dokumentation.

Literaturverzeichnis

[1] Hindel, Hörmann, Müller, Schmied, *Basiswissen Software-Projektmanagement*