Gruppe 2: Rui Gu, Wei Zhu, Veysel Imamoglu, Dimitar Dimitrov,
Karl Oppermann, Nathalie Hrycej, Markus Schnalke, Christoph Galler

Modellgetriebene Softwareentwicklung auf Basis von TOPCASED am Beispiel eines Seminarverwaltungssystems

Development Case

Ulm, 22. Januar 2008

Betreut durch:
Prof. Dr. Klaus Baer
Hochschule Ulm
Prittwitzstrasse 10
89075 Ulm

Version dieses Dokuments

Dokument: **Development Case**
Online-Seminarbuchungssystem

Hochschule Ulm

Version	Person	Aktion	Datum	Status	Kommentar
0.1	Markus Schnalke	E	2007-11-27	O	Erste Version
0.2	Markus Schnalke	AE	2008-01-13	O	Neue Struktur des Dokuments
0.4	Markus Schnalke	AE	2008-01-16	A	Struktur überarbeitet
0.4.1	Karl Oppermann	QS	2008-01-17	A	Allgemeines Review
0.5	Markus Schnalke	AE	2008-01-18	A	Überarbeitung
0.5.1	Veysel Imamoglu	QS	2008-01-18	A	Rechtschreibkorrektur
1.0	Markus Schnalke	AB	2008-01-22	A	Finale Version für R3

Aktion: E – Erstellung; AE – Änderung; QS – Review; AB – Abnahme
Status: O – Offen; D – Diskussion; A – Akzeptiert

Inhaltsverzeichnis

1. Einleitung

1.1. Zweck des Dokuments

Dieses Dokument beschreibt den Entwicklungsprozess nach dem wir in unserem Projekt vorgehen.

1.2. Definitionen und Abkürzungen

Die verwendeten Begriffe sind im Glossary erklärt. Bei Bedarf kann dort nachgeschlagen werden.

1.3. Verweise auf andere Artefakte

Der **Software Development Plan** ist an vielen Stellen mit diesem Dokument verknüpft.

2. Entwicklungsprozess

2.1. Überblick

Wir werden unser Projekt nach dem *Rational Unified Process*‡ (kurz RUP) entwickeln.

Der RUP ist ein dynamischer und iterativer Entwicklungsprozess, der das Projekt in zwei Dimensionen (zeitlich und inhaltlich) betrachtet.

An sich ist der RUP für große Projekte, mit vielen Mannjahren, ausgelegt. Für unser kleines Projekt (90 Manntage) ist er eher weniger gut geeignet. Wir haben uns trotzdem für den RUP entschieden, da wir ihn in der Vorlesung "Softwaretechnik 1" ausführlich behandelt hatten und wir dieses Theoriewissen nun in der Praxis anwenden wollen.

Wir haben diesen mächtigen und umfangreichen Prozess für unser kleines Projekt abgespeckt und angepasst. Wie unsere Adaptation des RUP genau aussieht, das beschreibt dieses Dokument.

2.2. Der Rational Unified Process auf einen Blick

Natürlich kann man diesen umfassenden Entwicklungsprozess nicht in einem Bild komplett abbilden, jedoch zeigt die nachfolgende Grafik sehr schön, wie die Entwicklung im Bezug auf die zwei Dimensionen aussieht. Dieser Übersichtplan soll den Aufbau des Prozesses nochmal ins Gedächtnis rufen.

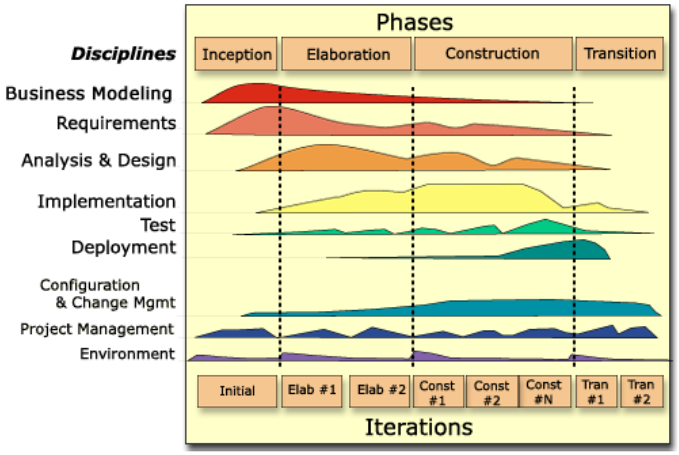

Abbildung 2.1.: Übersicht über einen Zyklus im RUP‡

3. Zeitliche Dimension

3.1. Anpassungen

Wir führen in unserem Projekt drei Zyklen durch. Jeder der drei Zyklen wird circa fünf Wochen (30 Manntage) umfassen. An deren Ende wird jeweils ein Release stehen. (siehe *Software Development Plan*)

Die einzelnen Phasen in den Zyklen versuchen wir, so gut es geht, zu berücksichtigen. Es muss bedacht werden, dass bei uns pro Phase ganz grob nur etwa 4 Manntage (d.h. circa 4 Stunden pro Person) zur Verfügung stehen.

Iterationen innerhalb der Zyklen werden wir, auf Grund der kurzen Zyklen, komplett außen vor lassen.

Ein Beispiel um ein Gefühl für die Größenverhältnisse zu bekommen: Unsere 90 Manntage, entsprechen realistischerweise eher einer einzelnen Iteration, als den drei Zyklen, die wir für uns geplant haben.

3.2. Konkrete Projektplanung

Die konkrete Planung der einzelnen Zyklen und ihrer Meilensteine finden sich im *Software Development Plan*.

4. Inhaltliche Dimension

In der zweiten Dimension wird festgelegt, *wer* (Rolle), *wie* (Aktivität), *was* (Artefakt), *wann* (Zeitpunkt) macht. Diese Elemente werden in Workflows vereint.

Hier beschreiben wir, wie wir die vorgegebenen Workflows des RUP anpassen.

4.1. Business Modeling

Zweck

Gemeinsames Verständnis zwischen Entwicklern und Anwendern schaffen.

Anpassungen

Wir erstellen keinen Business Use Case, weil das Seminarsystem ein gestelltes, abgeschlossenes Szenario ist und nicht in bestehende Geschäftsabläufe eingebunden werden muss.

Artefakte

Artefakt	**Glossary**
Rolle	Fachliches Team, Kunde
Aktivität	erstellen gemeinsam
Zeitpunkt	Inception, Ergänzungen jeder Zeit
Review	Fachliches Team und Kunde durch gemeinsames Erstellen

4.2. Requirements

Zweck

Ermitteln, was das System leisten soll. Die funktionalen Anforderungen erfassen.

Anpassungen

Keine besonderen.

Artefakte

Artefakt	**Vision**
Rolle	Fachliches Team, Kunde
Aktivität	erarbeiten im Gespräch
Zeitpunkt	Inception
Review	Fachliches Team und Kunde durch gemeinsames Erarbeiten

Artefakt	**Use Cases**
Rolle	Fachliches Team, Kunde
Aktivität	erarbeiten im Gespräch
Zeitpunkt	Inception und Elaboration
Review	Fachliches Team und Kunde durch gemeinsames Erarbeiten

4.3. Analysis & Design

Zweck

Aufbau und Technologie des Systems festlegen. Festlegung, wie das System realisiert wird.

Anpassungen

Die Technologie und die Rahmenbedingungen der Umsetzung sind durch das Projekt vorgegeben und somit fix.

Zum jetzigen Zeitpunkt arbeiten wir uns vor allem in die neue Technologie ein. Unsere Softwarearchitektur ist bisher vor allem eine Vorarbeit für später.

Artefakte

Artefakt	**Software Architecture Document**
Rolle	Team "Technische Architektur"
Aktivität	erstellt
Zeitpunkt	Elaboration
Review	Entwickler nach Änderungen

4.4. Implementation

Zweck

Systemteile entwickeln und zusammenfügen. Komponententests.

Anpassungen

Momentan besteht dieser Workflow in erster Line aus der Entwicklung von Prototypen jeder Art (Modelle, Templates, etc) um die Technologie zu erforschen.

Konkrete Artefakte werden bisher nicht erstellt, weil es zum jetzigen Stand nicht sinnvoll ist nach festen Plänen vorzugehen. Unser Kenntnisstand ändert sich sehr schnell und wir wollen flexibel reagieren können.

Artefakte

Keine definiert.

4.5. Testing

Zweck

Test des Zusammenspiels der Komponenten. Funktionsweise des Systems gegen die Anforderungen prüfen.

Anpassungen

Wir erkunden eine neue Technologie. Unser Ziel ist es, in kurzer Zeit möglichst viele Bereiche und Möglichkeiten zu erkunden. Tests bremsen die Entwicklungsgeschwindigkeit zugunsten von Qualität. Unser Hauptaugenmerk ist es voran zu kommen, nicht komplett fehlerfreie Ergebnisse zu liefern. Deshalb verzichten wir komplett auf Software-Tests. So können wir die dadurch verfügbaren Ressourcen an anderer Stelle effektiv nutzen. Sobald wir nicht mehr nur Prototypen erzeugen, werden wir natürlich Software-Tests einführen.

Alle Dokumente müssen von mindestens einer weiteren Person begutachtet werden. Die genauen Vorgaben stehen bei den Artefaktbeschreibungen in diesem Kapitel.

Artefakte

Keine definiert.

4.6. Deployment

Zweck

Auslieferung des Systems an den Kunden und Inbetriebnahme.

Anpassungen

Solange wir keine lauffähigen Ergebnisse haben, vernachlässigen wir diesen Workflow. Wenn dies nicht mehr der Fall sein wird, muss eine Anleitung zur Inbetriebnahme des Programms erstellt werden. Ebenso muss dann der genaue Funktionsumfang des Systems beschrieben sein.

Artefakte

Artefakt	**Release Notes**
Rolle	Projektleiter, Entwickler
Aktivität	sollen erstellen
Zeitpunkt	Transition
Review	anderer Entwickler vor der Auslieferung

4.7. Configuration & Changemanagement

Zweck

Verwaltung der zum Projekt gehörenden Daten. Versionierung der Daten. Change-Request-Management.

Anpassungen

Alle Daten müssen im zentralen Project Repository abgelegt werden.

Jeder Mitarbeiter darf an jeder Stelle des Projekts Änderungen durchführen.

Change-Requests werden nach dem vertraglich festgelegten Verfahren bearbeitet.

Artefakte

Keine definiert.

4.8. Projectmanagement

Zweck

Planung des Projekts. Zwischen konkurrierenden Zielen vermitteln. Auf Risiken reagieren.

Anpassungen

Keine besonderen.

Artefakte

Artefakt	**Software Development Plan**
Rolle	Projektleiter
Aktivität	erstellt
Zeitpunkt	Inception
Review	Entwickler und Risikomanager nach Änderungen

Artefakt	**Risk Management Plan**
Rolle	Riskmanager
Aktivität	erstellt
Zeitpunkt	Alle Phasen
Review	Komplettes Team in Inception Phase

4.9. Environment

Zweck

Rahmenbedingungen schaffen. Bereitstellung von Hardware, Software und Know-How.

Anpassungen

Die Hochschule Ulm stellt uns ein Project Repository zur Verfügung.

Die offiziellen Kommunikationwege im Team sind das wöchentliche Teammeeting und die Projekt-Mailingliste.

Artefakte

Artefakt	**Development Case**
Rolle	Projektleiter
Aktivität	erstellt
Zeitpunkt	Inception
Review	Komplettes Team in Inception Phase

Artefakt	**Tutorials**
Rolle	Toolspezialist
Aktivität	kann erstellen
Zeitpunkt	Alle Phasen
Review	eine Person für die das Tutorial geschrieben wurde

A. Quellen

- Dokumentation zum *Rational Unified Process* (`http://www-306.ibm.com/software/awdtools/rup/`)
- Skript von Herrn Prof. Dr. Baer zur Vorlesung *Softwaretechnik 1* an der Hochschule Ulm
- http://wikipedia.org
- *Rational Unified Process - Best Practices for Software Development Teams*

‡ The image 2.1 is from the Rational Unified Process (software product) version 2003.06.12.01. This image and the names "Rational Unified Process" and "RUP" are copyright by Rational Software Corporation, now a division of IBM.