



Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES

TEST-DRIVEN DEVELOPMENT MIT RUBY ON RAILS

AM BEISPIEL DER O/ZB WEBAPPLIKATION

PROJEKTARBEIT

vorgelegt von

Briewig, Martin, B.Sc. (Matr.-Nr.: 43509)

Leibel, Michael, B.Sc. (Matr.-Nr.: 43674)

Betreuer: Prof. Dr. Frank Schaefer

Fachbereich: Informatik (Master)

Inhaltsverzeichnis

1	Einleitung	1
1.1	Aufgabenbeschreibung	1
2	Hintergrundwissen	3
2.1	Das o/ZB Projekt	3
2.2	Das Testsystem	3
2.3	Test-Driven Development	3
2.4	o/ZB Darlehensvertrag	3
3	Allgemeine Dokumentationen	4
3.1	Das korrigierte ER-Diagramm	4
3.2	Das korrigierte Relationenmodell	4
3.3	Deployment und Versionskontrolle	4
3.4	Datenbank Migration	5
4	Korrekturen, Bugfixes	6
4.1	WebImport	6
4.2	Punkteberechnung	6
5	Testdriven Development	7
5.1	Analyse	7
5.2	Implementierung	7
6	Neue Features	8
6.1	Dokumenten-Export	8
7	Ergebnis und Ausblick	9
7.1	Ergebnis	9
7.2	Ausblick	9
8	Anhang	10

Abbildungsverzeichnis	11
Tabellenverzeichnis	12
Listings	13

Abkürzungsverzeichnis

AP Arbeitspaket

o/ZB Ohne Zins Bewegung

RoR Ruby on Rails

1 Einleitung

Die ohne Zins Bewegung Stuttgart arbeitet nun seit einigen Jahren mit der Hochschule Karlsruhe zusammen an einer Webanwendung. Diese Webanwendung soll die Geschäftsprozesse der Bewegung unterstützen und verwalten. Im Laufe der letzten Jahre ist aus der ursprünglichen Idee eine komplexe Webanwendung entstanden, die nun gründlich auf Fehler überprüft und gegebenenfalls ausgebessert werden soll. Diese Aufgabe wird nun uns, Herr Briewig und Herr Leibel, anvertraut. Im Folgenden wird eine detaillierte Aufgabenstellung formuliert, an dieser sich diese Projektarbeit messen lassen wird.

1.1 Aufgabenbeschreibung

Die Aufgaben dieser Projektarbeit sind recht vielfältig. Sie lassen sich in 3 Phasen einteilen. Darüber hinaus gibt es für jede Phase ein Zeitlimit von etwa 3 Wochen.

1.1.1 Phase 1: Korrekturen, Dokumentieren

(Duedate: 02.05.2013)

Die erste Phase wird von den Korrekturen am ER-Diagramm und dem dazugehörigen Relationsmodell dominiert. Das ER-Diagramm weist noch ein paar Unstimmigkeiten auf, die im Laufe der Zeit entstanden sind. Diese gilt es zu bereinigen und die vorgenommenen Korrekturen in der Implementierung umzusetzen. Ziel ist ein ER-Diagramm, welches den nicht-historisierten Zustand darstellt. Dieses ER-Diagramm soll 1:1 in der Implementierung vorzufinden zu sein. Es soll vereinfacht dargestellt werden, lediglich die Primär- und Fremdschlüssel sollen den Entitäten als Zusatzinformationen dienen.

Neben den Korrekturen am ER-Diagramm sind Dokumentationen der Vorgehens- und Funktionsweise des Datenbank Migrationstools notwendig. Hierfür sind zwei Batch-Skripte angefertigt worden, deren Nutzungs- und Funktionsweisen zu dokumentieren sind.

Im Laufe der Zeit sind die verschiedensten Techniken zur Bereitstellung und Versionierung der ozb Webanwendung verwendet worden. Es gilt nun, einen einheitlichen Vorgang zu definieren der für die nachfolgenden Projektgruppen verwendet werden soll. Basierend auf der anzufertigten Dokumentation, kann die Bereitstellung und Versionierung stetig verbessert werden.

Zusätzlich soll in dieser Phase ein kritischer Fehler bereinigt werden. Der WebImport, welcher für den Import der Kontobewegungen verantwortlich ist, arbeitet nicht richtig. Zur Zeit ist der Datei-Upload auf dem Testsystem nicht möglich. Darüber hinaus kommt es bei einer lokalen Testumgebung zu Laufzeitfehlern. Auch die im Webfrontend angezeigte Anzahl der importierten Datensätze ist nicht korrekt.

1.1.2 Phase 2: Korrektur der Punkteberechnung, Word Export

(Duedate: 23.05.2013)

1.1.3 Phase 3: Einrichtung Testdriven Development

(Duedate: 13.06.2013)

2 Hintergrundwissen

2.1 Das o/ZB Projekt

2.2 Das Testsystem

2.3 Test-Driven Development

2.3.1 Definition

2.3.2 Unit-Tests

2.3.3 Functional-Tests

2.3.4 Integration-Tests

2.4 o/ZB Darlehensvertrag

3 Allgemeine Dokumentationen

3.1 Das korrigierte ER-Diagramm

3.2 Das korrigierte Relationenmodell

3.3 Deployment und Versionskontrolle

In diesem Abschnitt werden die Begriffe *Deployment* und *Versionskontrolle* erläutert. In jedem Teilabschnitt wird die aktuelle Umsetzung beschrieben. Der letzte Teil dieses Abschnitts beschäftigt sich mit dem *Commit & Deployment Workflow*.

3.3.1 Deployment

In der Softwareentwicklung wird unter dem Begriff *Deployment* der Prozess zur *Bereitstellung* und *Verteilung* einer Software verstanden. Der Prozess ist von Software zu Software unterschiedlich, da z.B. die Konfiguration einer Software immer an die Umgebung angepasst werden muss, in der diese zum Einsatz kommt.

Im Falle der o/zb muss der Deployment Prozess die Bereitstellung der o/zb Webanwendung auf einem Server bewerkstelligen. Dies wirft die folgenden Fragen auf: Welcher Webserver wird eingesetzt? Wie gelangt die RoR Webanwendung auf den Server? Und wie arbeiten Anwendungsserver und Webserver auf dem Server zusammen? Diese Fragen werden die nun folgenden Teilabschnitte klären.

Webserver - Apache

Application Server - Passenger

Capistrano

Das Deploymentskript

3.3.2 Versionskontrolle

Git

3.3.3 Workflow, Umsetzung

3.4 Datenbank Migration

4 Korrekturen, Bugfixes

4.1 WebImport

4.2 Punkteberechnung

5 Testdriven Development

5.1 Analyse

5.2 Implementierung

5.2.1 RSpec

6 Neue Features

6.1 Dokumenten-Export

6.1.1 Analyse

6.1.2 Implementierung

7 Ergebnis und Ausblick

7.1 Ergebnis

7.2 Ausblick

8 Anhang

Abbildungsverzeichnis

Tabellenverzeichnis

Listings