

Abschlusspräsentation

—

o/ZB Webanwendung

Einführung Test-driven Development

Erstellt von

Martin Briewig, B.Sc.
Michael Leibel, B.Sc.

Betreut von

Herrn Prof. Dr. Schaefer



Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES

Inhalt

- Das o/ZB Webprojekt
- Stand vor der Übernahme
- Was ist Test-driven Development?
- Ziele der Einführung TDD
- Umsetzung TDD
- Ergebnis, Konsequenzen
- Demonstration
- Quellen, Verweise

Das o/ZB Webprojekt

- Die aktuell bestehende Webanwendung soll durch eine neue Implementierung ersetzt werden
- Im Oktober 2011 begann die Umsetzung der Webanwendung in Ruby on Rails
- Die Webanwendung soll
 - Die Geschäftsprozesse der o/ZB abdecken
 - Besser gewartet und erweitert werden können
 - Die Daten historisiert abspeichern
 - Mit neuen Technologien versorgt werden

Das o/ZB Webprojekt

Besser ohne / Zins

Meine Konten


Meine Daten

Protokolle

o/ZBlick

Mitglieder

Weiteres ▾

 Ausloggen

Meine Konten

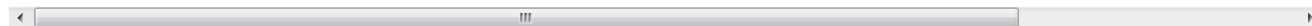
Guten Tag, Tassilo Kienle

EE-Konten

EE-Konto	Letzte Kontobewegung	Währungssaldo	Punktesaldo	Dispo-Limit	Konto-Nr.	BLZ	Kreditinstitut
50013	31.12.2011	██████	██████	██████	1011507447	12030000	DKB Bank
60013	31.03.2012	██████	██████	██████	1011507447	12030000	DKB Bank
70013	31.12.2012	██████	██████	██████	1011507447	12030000	DKB Bank

ZE-Konten

ZE-Konto	ZE-Nr.	ZE-Betrag	Rest-ZE	Punktesaldo	Letzte Kontenbewegung	Laufzeit Jahre	Modus	Tilgung	Sparen	RDU
10013	D110125	██████	0,00	0	30.04.2012	2 Monate	m	██████	0,00	0,00
20013	D100331	██████	0,00	0	30.12.2010	0 Monate	m	██████	██████	0,00
30013	D100430	██████	0,00	0	02.11.2011	0 Monate	m	██████	██████	0,00



Dashboard der neuen Webanwendung

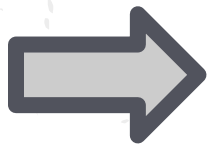
Stand vor der Übernahme

- Weit fortgeschrittene Ruby on Rails Webanwendung
- Unstetiges Datenmodell, inkonsistente Umsetzung
- Erste manuell durchgeführte Tests
- Grobe Fehler in den Kern-geschäftsprozessen, wie z.B. Darlehensverlauf, Punkteberechnung, Webimport, Benutzerverwaltung ...

Stand vor der Übernahme

- **Fazit**

- Keine genaue Aussage über den Zustand der Webanwendung möglich
- Von Hand durchgeführte Tests sind mühsam und sehr aufwendig
- Es existiert kein definierter Entwicklungsprozess oder Prozess für die Softwareverteilung auf den Server

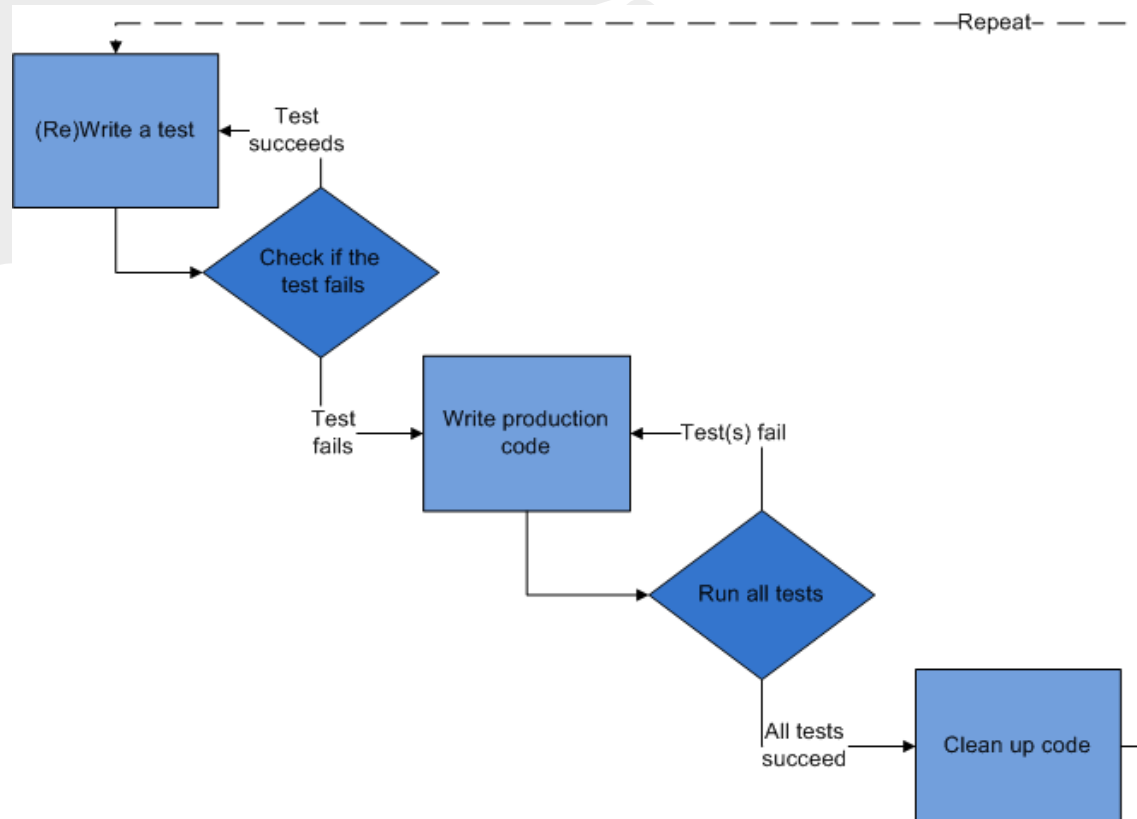


Eine Lösung muss her, die sowohl eine fachlich als auch technisch korrekte Umsetzung unterstützt.

Was ist Test-driven Development?

Hauptmerkmal

Software-Tests werden konsequent **vor** den zu testenden Komponenten erstellt



Einführung TDD


Ziele

Das Hauptmerkmal kann in diesem Projekt erst wieder bei neuen Funktionalitäten angewandt werden

- Mit Hilfe der Tests soll der bisherige Zustand, der historisch gewachsenen Webanwendung, aufgenommen werden
 - Konsistenz: Ruby Anwendung <-> Datenbank
 - Fachlich und technische Validierung der einzelnen Modelle (Attribute, Beziehungen und Funktionen) und Controller (Logik, Controller übergreifende Funktionalitäten)

Umsetzung TDD

Diese Webanwendung besteht aus 3 Teilen, angelehnt an das MVC (Model, View, Controller) Konzept. Jeder Teil wird gesondert und in der folgenden Reihenfolge getestet.

- 
- **Model**
 - Person, OZBPerson, OZBKonto, Buchung, ...
 - Validierung, Datenbank, Historisierung
 - **Controller**
 - Schnittstelle zwischen Benutzeroberfläche und Model
 - Businesslogik
 - **View (Benutzeroberfläche)**
 - Webseite, Buttons, Texteingabe, ...

Umsetzung TDD

Model

- Unit Test
- Jedes Model wird isoliert getestet
- Es wird getestet (isoliert)
 - korrekte Implementierung der Attribute inkl. Validierung
 - alle Funktionen, auch private
- Dabei werden sowohl die gültigen als auch ungültigen Fälle berücksichtigt (z.B. Verletzung des Wertebereiches oder nicht vorhande Werte/Beziehungen)

Umsetzung TDD

Controller

- Feature Test
- Jeder Controller wird isoliert getestet
- Es wird getestet (isoliert)
 - korrekte Implementierung der aller Funktionen
- Mögliche Benutzereingaben (=Testszenarien) werden dem Controller vorgegaukelt
- Diese Benutzereingaben werden nicht durch die vorhandene Weboberfläche generiert

Umsetzung TDD

Views

- Integration Test
- Es wird das Zusammenspiel aller Controller in mehreren Testszenarien getestet
- Diese Benutzereingaben werden in diesem Fall durch die vorhandene Weboberfläche generiert (durch simulierte Benutzereingaben auf der Weboberfläche)
- Wurde im Rahmen dieser Projektarbeit nicht durchgeführt, aber in der Dokumentation berücksichtigt

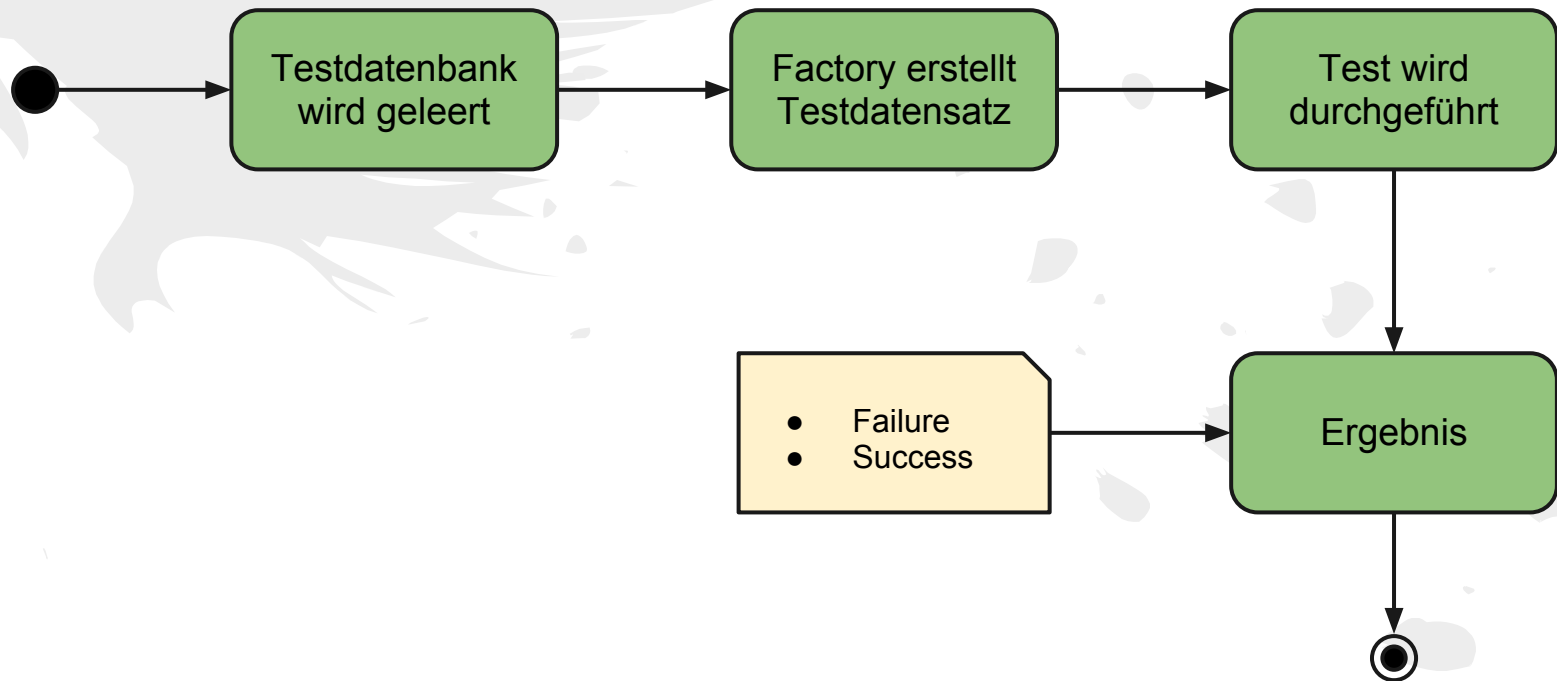
Umsetzung TDD

Tools

- Sublime Text 2 inkl. RubyTest (Plugin)
- Cygwin (Unix Kommandozeile für Windows)
- RSpec
 - Ruby Testing-Framwork
- FactoryGirl
 - Erzeugt Testdatensätze
- Faker
 - Erzeugt zufällige Namen, Straßen, Telefonnummern, ...

Umsetzung TDD

Ablauf



Ergebnis

Zahlen, Daten, Fakten

393 examples,

0 failures,

64 pending

[16.09.2013]

12 Modelle getestet

1 Modul getestet (HistoricRecord)

1 Feature Test (**5** BenutzerSzenarien, inkl. Controller Funktionen)

Ergebnis

Probleme

- Erhebliche inkonsistenzen in der Model-Umsetzung
 - Datenbank <-> Model stimmen nicht überein
 - Validierungen fehlen teilweise komplett oder sind fehlerhaft
 - Jede Menge Redundanzen (z.B. Umsetzung Historisierung) -> kaum gut wartbarer Code
- Datenbankmodel unstetig
 - Historisch gewachsene Veränderungen
 - Das Datenmodell bildet die Basis für die gesamte Anwendung und sollte daher stabil bleiben

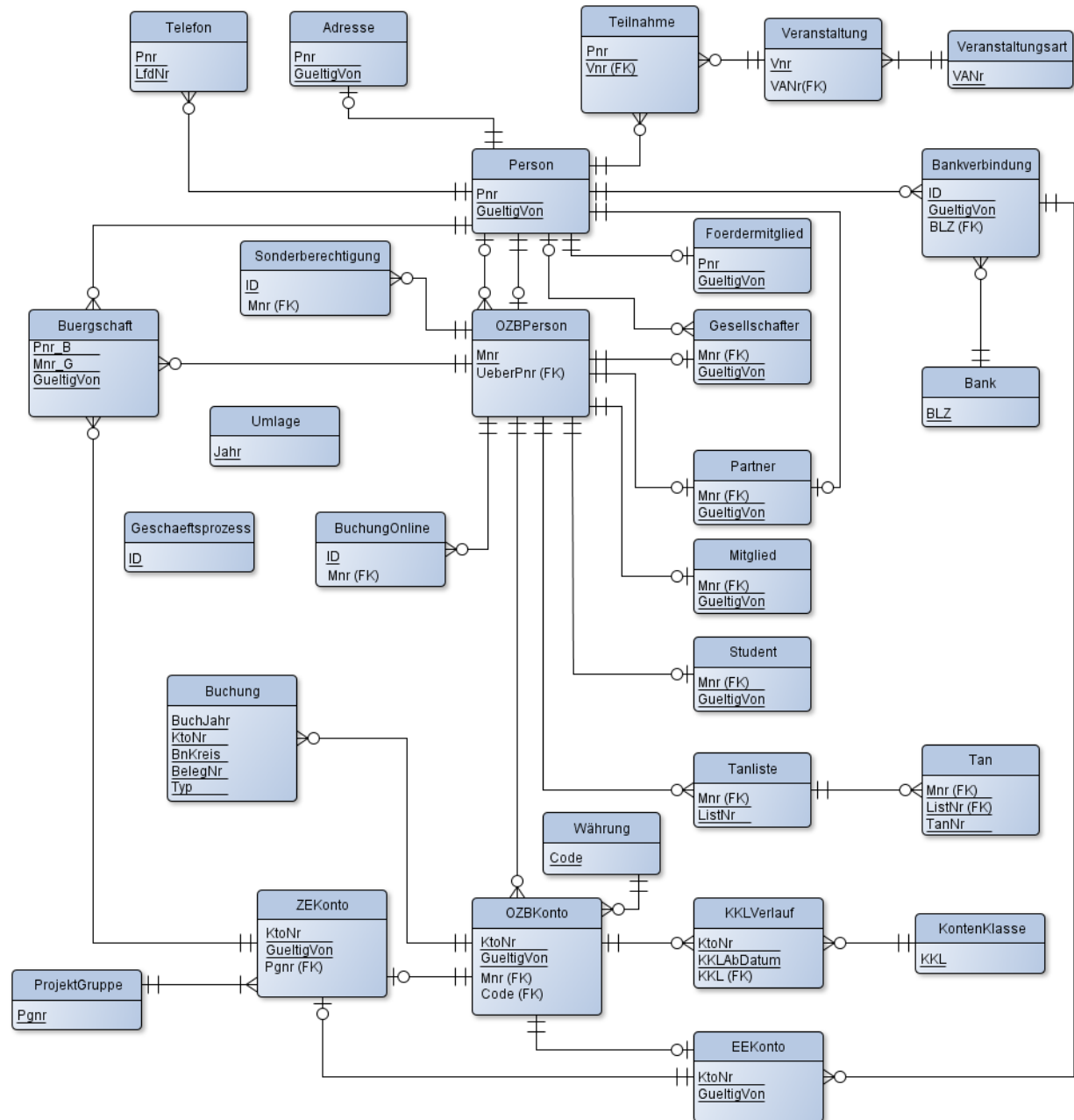
Ergebnis

Konsequenzen

- Datenmodell wurde festgelegt
- Migrationstool wurde angepasst
- Refactoring aller Modelle
- Weitere Unit/Feature Tests
- Historisierung wurde neu implementiert und getestet
- Entwicklung eines Software-Verteilungsprozess mit Hilfe von Git, Capistrano und Passenger (ist in der Doku enthalten)

Ergebnis

- Darstellung der Relationen im nicht historisierten Fall
- Eindeutige Kennzeichnung von Primary und Foreign Keys
- Neue Tabellen und Assoziationen hinzugefügt
- Dient zur fachlichen Übersicht der Datenbank



Demonstration



Quellen, Verweise

GitHub Repository der Webanwendung

<http://github.com/Avenel/FirstApp>

everyday rails - Testing with RSpec (Aaron Summer, 2013)

Link zur Webanwendung (Development)

<http://ozbapp.moood.com/>