

LOS

Lecture **O**perator **S**ystem

Mathias Hölzl, Felix Putz, Florian Tischler

Agenda

- App
- Architektur
- Build- und Testautomation
- Demo

App

- Problem
 - niedrige Anwesenstraten bei Vorlesungen
 - wenig Feedback über Qualität der Vermittlung des Lehrstoffs
 - Feedback erst bei der Klausur
- Lösung
 - Online Plattform für interaktive Vorlesungen

App

- Online
 - im Intranet der Universität
- Plattform
 - Webseite
 - auf allen browser-fähigen Endgeräten verfügbar
- Interaktive Vorlesung
 - Dozenten können Fragen an die Anwesenden richten

App

- Funktionen
 - Anwesenheitsprüfung
 - Quiz während der Vorlesung
 - Abgeben von Feedback
 - Statistiken erstellen
- Möglichkeiten
 - Frühzeitige Erkennung schwer verständlicher Themen
 - Anrechnen der Quizresultate bei der Klausur
 - Steigerung der Anwesenheitsrate

Architektur

MVC – Webapplikation

- View
 - Bootstrap, Lumx
- Controller
 - Client - Angular JS
 - Server - Spring Framework (Java)
- Model
 - Java
 - MongoDB

Build- und Testautomation

- Buildautomation
 - Auf dediziertem Buildserver
 - kompilieren
 - Tests ausführen (Testautomation)
 - Deployment
 - 3 Arten
 - On-Demand
 - Scheduled (CI)
 - Triggered (CI)

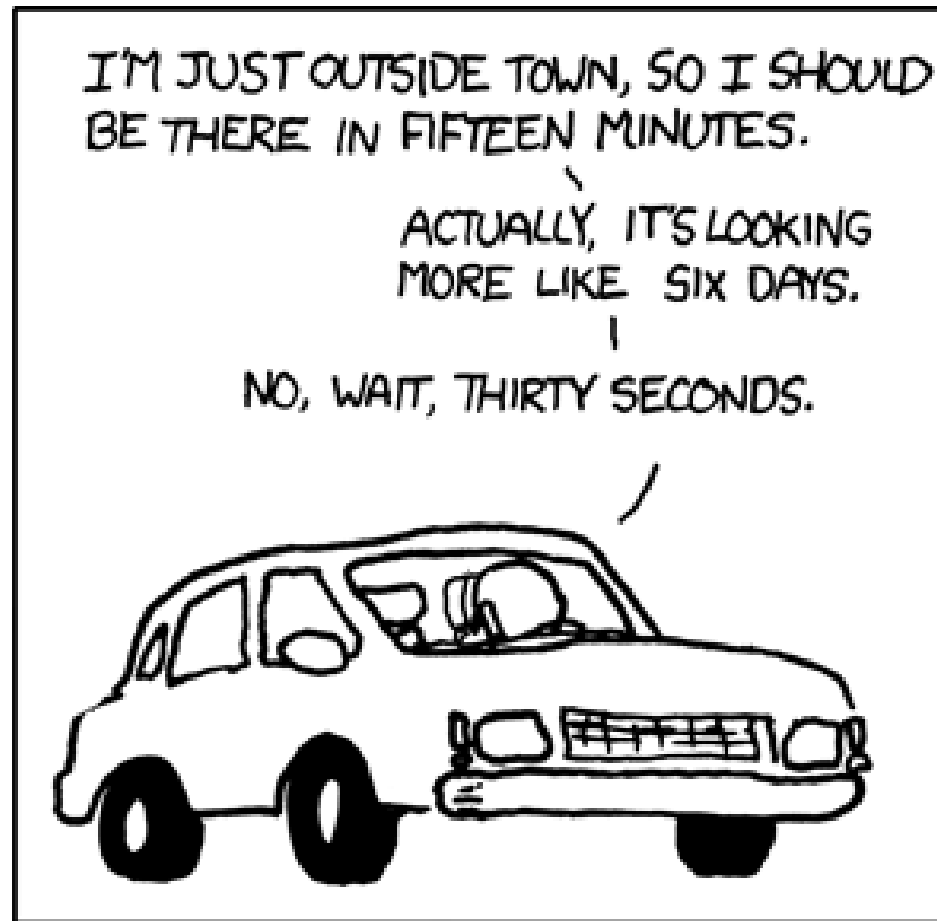
Build- und Testautomation

- Testautomation
 - automatisch nach jedem erfolgreichen Build
 - Regressionstests
- 3 Arten
 - Code-driven testing
 - GUI testing
 - API driven testing

Build- und Testautomation

- Umsetzung
 - Continuous Integration über TravisCI
 - frei konfigurierbar (Ubuntu)
 - Github-Integration
- Ablauf
 1. neuer Commit auf das Repository gepusht
 2. TravisCI checkt diesen aus
 3. Start des Builds und der Testausführung (Maven)
 4. Emailbenachrichtung bei Fehler

Demo - Just a Second....



THE AUTHOR OF THE WINDOWS FILE
COPY DIALOG VISITS SOME FRIENDS.

Vielen Dank für Ihre Aufmerksamkeit!

- Wir stehen nun gerne für Fragen zur Verfügung