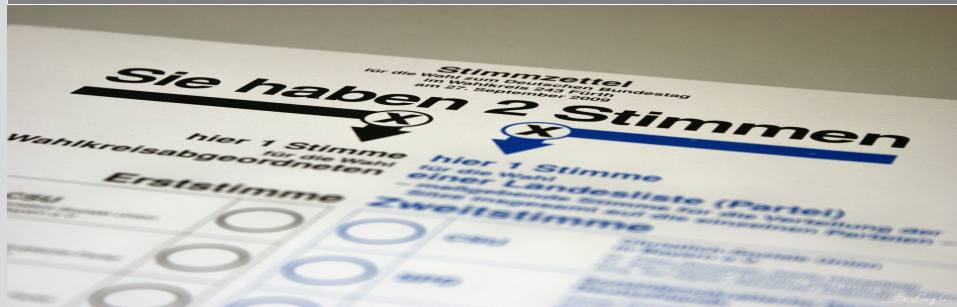


# OpenBundestagswahl

Qualitätssicherungsphase

Simon Schürg | 10. März 2014

PRAXIS DER SOFTWAREENTWICKLUNG, WS 2013/14



- 1 Modultests
  - Mandatsrechner
  - Model
  - Wahlgenerator
  - Wahlvergleich
  - Import/Export und Config
  - Testabdeckung
- 2 Integrations- und Systemtests
  - GUI
  - Plattformunabhängigkeit
- 3 Veränderungen und Sonstiges
- 4 Fazit
- 5 Ausblick

- Regressionstests mittels **JUnit**
  - Zum automatischen und wiederholbaren Testen
  - Nur für Module der Anwendungslogik
- Codeanalyse mit **FindBugs**
  - Untersuchung von Code auf bekannte Fehlermuster
- Organisation mit **Redmine** Ticketsystem

- Kern-Feature der Anwendung
- Korrekte Berechnung gemäß Wahlgesetz sehr wichtig
- Viele Spezialfälle und Zufall (Auslosen) macht Testen aufwändig
- Probleme mit Endlosschleifen

- Komplexe Objektstruktur des Models macht Testen aufwändig
  - Viele Ausnahmen müssen überprüft werden
  - Konsistenz der Datenobjekte muss gewährleistet sein (Anforderung aus dem Pflichtenheft)

- Generierung nur mit gültigen Stimmanteilen
- Erzeugen einer Wahl muss mit gültigen Parametern immer möglich sein
- Probleme mit Wahlkreisen ohne Stimmen









- Berechnung der Differenzen muss stimmen

- Prüfen von CSV-Dateien auf Gültigkeit
- Alle relevanten Informationen korrekt einlesen und wieder schreiben
- Probleme mit unterschiedlichen Zeichensätzen











- EcEmma zur automatischen Überprüfung der Testabdeckung
- Methoden gefunden die nicht benötigt wurden
- Methoden gefunden deren Funktionalität doppelt vorhanden war

## Anweisungsüberdeckung

Element		Coverage	Covered Instructions	Missed Instructions	Total Instructions
▷ main.java.chronik		80,9 %	212	50	262
▷ main.java.config		46,9 %	242	274	516
▷ main.java.importexport		93,2 %	2.081	153	2.234
▷ main.java.mandatsrechner		81,6 %	1.402	317	1.719
▷ main.java.model		84,9 %	2.583	458	3.041
▷ main.java.steuerung		15,2 %	25	140	165
▷ main.java.wahlgenerator		82,8 %	863	179	1.042
▷ main.java.wahlvergleich		65,8 %	438	228	666

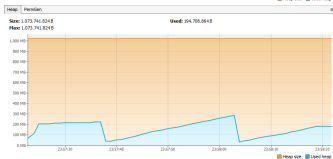
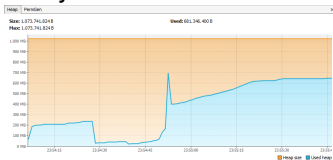
## Zweigüberdeckung

Element		Coverage	Covered Branches	Missed Branches	Total Branches
▷ main.java.chronik		72,7 %	16	6	22
▷ main.java.config		39,7 %	23	35	58
▷ main.java.importexport		77,7 %	213	61	274
▷ main.java.mandatsrechner		75,8 %	150	48	198
▷ main.java.model		81,7 %	299	67	366
▷ main.java.steuerung		12,5 %	3	21	24
▷ main.java.wahlgenerator		79,7 %	110	28	138
▷ main.java.wahlvergleich		39,2 %	29	45	74

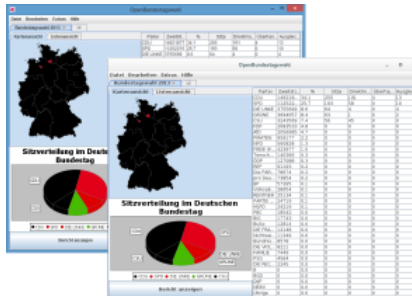
- Manuell durchgeführte Tests
- Anforderungen aus dem Pflichtenheft
  - Nicht-funktionale Anforderungen
  - Laufzeit, Bedienbarkeit ...

- GUI wurde manuell getestet
- Unterschiedliche Probleme wurden behoben
  - Layout
  - Oberflächenlogik
  - Fehlerbehandlung und Meldungen

- Performance Probleme
- Analyse mit JVisualVM war sehr hilfreich



- Unter Windows entwickelt
- Unter Windows und Linux getestet und lauffähig



- Simulation des negativen Stimmgewichts entfernt
- Hilfe Dialoge verwenden keine JavaFX Webviews mehr
- Handbuch wurde neu geschrieben
- Zusätzliche Features
  - Sortierung der Tabellen
  - Debug Klasse
- Name der Anwendung ist **OpenBundestagswahl**

- Es konnten viele Fehler gefunden und behoben werden
- Wir sind mit der Qualität der Software zufrieden
- Automatische Regressionstests sind uns wichtiger geworden
  - Wäre auch schon in der Implementierungsphase sinnvoll gewesen



- Entwicklung ist abgeschlossen
- In den letzten beiden Projektwochen ist Manuel Olk Phasenverantwortlicher

Vielen Dank für Ihre Aufmerksamkeit!