



Praktikable Ansätze zur Verbesserung der Entwicklung eines Altsystems

Nataliia Lashchyk
Freie Universität Berlin

10. September 2015

Übersicht

Fallstudie

WELTN24

Motivation und Ziele

Bestehende Probleme

Ziele

Konzept

Infrastruktur

Defektanfällige Komponenten

Erkennung von Defektmustern

Wartungsaufwand für unerreichbare Codeblöcke

Testabdeckung

Ergebnisse

Auswertung im Überblick

Fazit

Bewertung des Systems

Ausblick

Weiterer Forschungsbedarf

Übersicht

Fallstudie

WELTN24

Motivation und Ziele

Bestehende Probleme

Ziele

Konzept

Infrastruktur

Defektanfällige Komponenten

Erkennung von Defektmustern

Wartungsaufwand für unerreichbare Codeblöcke

Testabdeckung

Ergebnisse

Auswertung im Überblick

Fazit

Bewertung des Systems

Ausblick

Weiterer Forschungsbedarf

WELTN24

- ▶ über 4 Jahre der Entwicklung
- ▶ Java, JavaScript
- ▶ über 155 000 LOC
- ▶ bis zu 20% Testabdeckung
- ▶ 8 - 12 Entwickler

Übersicht

Fallstudie

WELTN24

Motivation und Ziele

Bestehende Probleme

Ziele

Konzept

Infrastruktur

Defektanfällige Komponenten

Erkennung von Defektmustern

Wartungsaufwand für unerreichbare Codeblöcke

Testabdeckung

Ergebnisse

Auswertung im Überblick

Fazit

Bewertung des Systems

Ausblick

Weiterer Forschungsbedarf

- ▶ Altsystem (*engl.* legacy code)
 - ▶ kleine Testabdeckung (nach *Michael C. Feathers*)
- ▶ CMS im Hintergrund
- ▶ hohe Komplexität
- ▶ unzureichende Dokumentation
- ▶ hohe Kosten für Wartung und Weiterentwicklung

Übersicht

Fallstudie

WELTN24

Motivation und Ziele

Bestehende Probleme

Ziele

Konzept

Infrastruktur

Defektanfällige Komponenten

Erkennung von Defektmustern

Wartungsaufwand für unerreichbare Codeblöcke

Testabdeckung

Ergebnisse

Auswertung im Überblick

Fazit

Bewertung des Systems

Ausblick

Weiterer Forschungsbedarf

- ▶ Systemanalyse
 - ▶ Überblick des Systems
 - ▶ Verständnis des Systems
 - ▶ potenzielle Schwachstellen
 - ▶ Strategien für Weiterentwicklung
 - ▶ nützliche Produktpflege
 - ▶ Entwicklungsverbesserungen

Übersicht

Fallstudie

WELTN24

Motivation und Ziele

Bestehende Probleme

Ziele

Konzept

Infrastruktur

Defektanfällige Komponenten

Erkennung von Defektmustern

Wartungsaufwand für unerreichbare Codeblöcke

Testabdeckung

Ergebnisse

Auswertung im Überblick

Fazit

Bewertung des Systems

Ausblick

Weiterer Forschungsbedarf

Übersicht

Fallstudie

WELTN24

Motivation und Ziele

Bestehende Probleme

Ziele

Konzept

Infrastruktur

Defektanfällige Komponenten

Erkennung von Defektmustern

Wartungsaufwand für unerreichbare Codeblöcke

Testabdeckung

Ergebnisse

Auswertung im Überblick

Fazit

Bewertung des Systems

Ausblick

Weiterer Forschungsbedarf

- ▶ Ziel
 - ▶ Identifizierung von kritischen Modulen im System
- ▶ Forschungsfrage I:
 - ▶ Welche Module haben ein großes Fehlerpotenzial und durch welche Variablen wird dies beeinflusst?

Defektanfällige Komponenten

- ▶ Überführung des SVN-Repositories in MySQL Datenbank
 - ▶ CVSAAnalY
- ▶ Einlesen von JIRA Datenbank
- ▶ Erkennen von Bflinks (-> 755)
 - ▶ das Paar aus einem Fehlerbericht aus JIRA und einer entsprechenden Fehlerkorrektur
- ▶ Verifizierung von Bflinks
- ▶ Identifizierung der Module mit den meisten Korrekturen

- ▶ Überführung des SVN-Repositories in MySQL Datenbank
 - ▶ CVSAAnalY
- ▶ Einlesen von JIRA Datenbank
- ▶ Erkennen von Bflinks
 - ▶ das Paar aus einem Fehlerbericht aus JIRA und einer entsprechenden Fehlerkorrektur
- ▶ **Verifizierung von Bflinks**
- ▶ Identifizierung der Module mit den meisten Korrekturen

- ▶ Bflinks mit den Fehler-IDs, welche in vielen Commits vorkommen
- ▶ Bflinks mit einem negativen Zeitabstand
- ▶ Bflinks mit einer invaliden Aktualisierung:
 - ▶ Der Commit nach der Aktualisierung
 - ▶ Die Aktualisierung nach einige lange Zeit
- ▶ Bflinks mit mehreren Fehler-IDs in einem SVN-Nachricht

- ▶ Überführung des SVN-Repositories in MySQL Datenbank
 - ▶ CVSAAnalY
- ▶ Einlesen von JIRA Datenbank
- ▶ Erkennen von Bflinks
 - ▶ das Paar aus einem Fehlerbericht aus JIRA und einer entsprechenden Fehlerkorrektur
- ▶ Verifizierung von Bflinks
- ▶ Identifizierung der Module mit den meisten Korrekturen

Übersicht

Fallstudie

WELTN24

Motivation und Ziele

Bestehende Probleme

Ziele

Konzept

Infrastruktur

Defektanfällige Komponenten

Erkennung von Defektmustern

Wartungsaufwand für unerreichbare Codeblöcke

Testabdeckung

Ergebnisse

Auswertung im Überblick

Fazit

Bewertung des Systems

Ausblick

Weiterer Forschungsbedarf

- ▶ Ziel
 - ▶ Analyse von Defekten anhand von Fehlerberichten und Defektkorrekturen
 - ▶ Fehlermuster
 - ▶ Ursprung von Defekten
- ▶ Forschungsfrage II:
 - ▶ Inwiefern können durch das Code-Review von Fehlerkorrekturen bestimmte Fehlermuster im System erkannt werden?

- ▶ Auswahl der Fehlerberichte
- ▶ Defekterkennung im Code
- ▶ Fehlerklassifizierung
 - ▶ Fehlermuster
 - ▶ Fehler oder Teilmenge von Fehlern mit gleichen Ausprägungen

```
Index: httpd/htdocs/www.welt.de/skins/welt/css/style.
less
=====
--- httpd/htdocs/www.welt.de/skins/welt/css/style.less
    (revision 96785)
+++ httpd/htdocs/www.welt.de/skins/welt/css/style.less
    (revision 96786)
@@ -184,7 +185,7 @@

    #topMenu {
        float: right;
-   margin: 12px 10px 0 190px;
+   margin: 12px 10px 0 0;
    }
```

Beispiel eines Diffs für eine CSS-Fehlerkorrektur in CVSAly

- ▶ Auswahl der Fehlerberichte
- ▶ Defekterkennung im Code
- ▶ Fehlerklassifizierung
 - ▶ Fehlermuster
 - ▶ Fehler oder Teilmenge von Fehlern mit gleichen Ausprägungen

Übersicht

Fallstudie

WELTN24

Motivation und Ziele

Bestehende Probleme

Ziele

Konzept

Infrastruktur

Defektanfällige Komponenten

Erkennung von Defektmustern

Wartungsaufwand für unerreichbare Codeblöcke

Testabdeckung

Ergebnisse

Auswertung im Überblick

Fazit

Bewertung des Systems

Ausblick

Weiterer Forschungsbedarf

- ▶ Ziel
 - ▶ Analyse des Wartungsaufwands und der Wirkung von unerreichbarem Code
- ▶ Forschungsfrage III:
 - ▶ Wie hoch ist der Wartungsaufwand bei unerreichbaren Codeblöcken?

- ▶ Ermittlung von unerreichbaren Codeblöcken
 - ▶ SonarQube
- ▶ Analyse des ermittelten Codes

Übersicht

Fallstudie

WELTN24

Motivation und Ziele

Bestehende Probleme

Ziele

Konzept

Infrastruktur

Defektanfällige Komponenten

Erkennung von Defektmustern

Wartungsaufwand für unerreichbare Codeblöcke

Testabdeckung

Ergebnisse

Auswertung im Überblick

Fazit

Bewertung des Systems

Ausblick

Weiterer Forschungsbedarf

- ▶ Ziel
 - ▶ Analyse der Weiterentwicklung von Unit Tests und deren Aufwand
- ▶ Forschungsfrage IV:
 - ▶ Wie können die Unit Tests für vorhandenen schwer testbaren Code geschrieben werden?

- ▶ Auswahl eines Moduls
 - ▶ eine kritische Komponente im System
- ▶ Schreiben von Unit Tests
 - ▶ Wie schwer ist der Code zu verstehen?
 - ▶ Wie viel Aufwand wird gebraucht, um den Code zu verstehen?

Übersicht

Fallstudie

WELTN24

Motivation und Ziele

Bestehende Probleme

Ziele

Konzept

Infrastruktur

Defektanfällige Komponenten

Erkennung von Defektmustern

Wartungsaufwand für unerreichbare Codeblöcke

Testabdeckung

Ergebnisse

Auswertung im Überblick

Fazit

Bewertung des Systems

Ausblick

Weiterer Forschungsbedarf

- ✓ grobe Unterteilung der Verzeichnisse
 - ▶ ein bestimmtes Backend-Verzeichnis auf der Modulebene
 - ▶ CSS-Dateien auf der Dateiebene
- ✓ Dateien mit Maximalzahl von Fehlerkorrekturen für Backend- bzw. Frontend-Module
- ✗ Erkennen von Fehlerquellen

- ✗ keine fassbare Klassifizierung von Fehlern
 - ▶ Inspektion von Fehler-IDs
 - ▶ 30 (CSS)
 - ▶ 40 (Java)
 - ▶ viele Refactoring-Maßnahmen statt Korrekturen
 - ▶ großer Aufwand durch Reviews
- ✗ Fehlerursachen

- ✓ 18 Codeblöcke
 - ▶ bis 24 LOCs
- ✗ gemeinsames Review mit Entwicklern

- ✓ Review von bereits geschriebenen Unit Tests
 - ▶ großer Zeitaufwand
 - ▶ viele Abhängigkeiten zu externen Quellen
- ✗ Implementierung von Unit Tests

Übersicht

Fallstudie

WELTN24

Motivation und Ziele

Bestehende Probleme

Ziele

Konzept

Infrastruktur

Defektanfällige Komponenten

Erkennung von Defektmustern

Wartungsaufwand für unerreichbare Codeblöcke

Testabdeckung

Ergebnisse

Auswertung im Überblick

Fazit

Bewertung des Systems

Ausblick

Weiterer Forschungsbedarf

- ▶ Forschungsfrage I: Welche Module haben ein großes Fehlerpotenzial und durch welche Variablen wird dies beeinflusst?
 - ▶ Ziel teilweise erreicht
 - ✓ Defektanfällige Komponenten bzw. Dateien
 - ✗ Auswertung der Defektanfälligkeit bzw. der Fehlerquellen
 - ▶ genügend hohe Relevanz für WeltN24
- ▶ Forschungsfrage II: Inwiefern können durch ein Code-Review von Fehlerkorrekturen bestimmte Fehlermuster im System erkannt werden?
 - ✗ Ziel nicht erreicht
 - ✗ Fehlermuster
 - ▶ niedrige Anzahl der Messergebnisse -> niedrige Aussagekraft

- ▶ Forschungsfrage III: Wie hoch ist der Wartungsaufwand bei unerreichbaren Codeblöcken?
 - ▶ Ziel teilweise erreicht
 - ✓ eine niedrige Anzahl von unerreichbarem Code (Java, JavaScript), kein bedeutender Einfluss auf das System
 - ✗ statische Analyse von CSS-Dateien
- ▶ Forschungsfrage IV: Wie können die Unit Tests für vorhandenen schwer testbaren Code geschrieben werden?
 - ✗ Ziel nicht erreicht
 - ✓ Review der bestehenden Unit Tests
 - ✗ neue Unit Tests

Übersicht

Fallstudie

WELTN24

Motivation und Ziele

Bestehende Probleme

Ziele

Konzept

Infrastruktur

Defektanfällige Komponenten

Erkennung von Defektmustern

Wartungsaufwand für unerreichbare Codeblöcke

Testabdeckung

Ergebnisse

Auswertung im Überblick

Fazit

Bewertung des Systems

Ausblick

Weiterer Forschungsbedarf

- ▶ mehrere Defizite im Prozess und Produkt aufgedeckt und Empfehlungen für eventuelle Optimierungen des Produkts und Prozesses in der Zukunft vorgelegt
 - ▶ feste Darstellungsvorgabe für Elemente
 - ▶ Vermeiden von hoher Konfigurierbarkeit
 - ▶ Erfassung der Zeit für Fehlerkorrekturen
 - ▶ Wohlgeformte Commit-Nachrichten in SVN
 - ▶ Fehlerpriorisierung, Fehlerlösung, Labeling für Konfigurationsfehler in JIRA
- ▶ Identifizierung von Fehlerquellen durch das Erkennen von Fehlermustern im Code manuell ist nicht lohnend
 - ▶ schwierige und umfangreiche Aufgabe
 - ▶ Mitwirkung eines erfahrenen Entwicklers wäre hilfreich
- ▶ eine manuelle Untersuchung, Verifizierung von BFlinks sowie das Code-Review erfordern einen hohen Zeitaufwand

Übersicht

Fallstudie

WELTN24

Motivation und Ziele

Bestehende Probleme

Ziele

Konzept

Infrastruktur

Defektanfällige Komponenten

Erkennung von Defektmustern

Wartungsaufwand für unerreichbare Codeblöcke

Testabdeckung

Ergebnisse

Auswertung im Überblick

Fazit

Bewertung des Systems

Ausblick

Weiterer Forschungsbedarf

- ▶ komplexe und breite Analyse durch wenig Aufwand
- ▶ automatische Ermittlung von Fehlerursachen

Vielen Dank
für Ihre Aufmerksamkeit!