



Abschlussvortrag Masterarbeit:
Operationalizing Architecture in an agile Software Project

Arsenij E. Solovjev
Freie Universität Berlin, Institut für Informatik

February 2, 2015

Arsenij E.
Solovjev

Einleitung Architektur Das Problem

Einleitung
Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Mein Ansatz: konstruktive Qualitätssicherung

Integration
in das
Prozess

Integration in das Prozess

Archnemesis

Archnemesis

Architecture
Compliance
Checking

Architecture Compliance Checking

Fazit

Fazit

Was ist Softwarearchitektur?

Softwarearchitektur

geht über die konkreten Algorithmen und Datenstrukturen hinaus und befasst sich mit der Gesamtstruktur eines Systems.

befasst sich mit:

- ▶ Globale Kontrolstrukturen
- ▶ Protokolle für Kommunikation, Synchronization und Datenzugriff
- ▶ Zuweisung von Funktionalität zu Entwurfselementen
- ▶ Komposition von Entwurfselementen
- ▶ physische Verteilung d. Systems usw.

[frei nach [Garland and Shaw, 1993]]

Arsenij E.
Solovjev

Einleitung
Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archnemesi

Architecture
Compliance
Checking

Fazit

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archnemesis

Architecture
Compliance
Checking

Fazit

“Emergent Architecture”

- ▶ keine explizite Planung
- ▶ Architektur entsteht durch *zeitgerechte* Refaktorisierung
- ▶ braucht Gewissenhaftigkeit vom Entwickler

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archnemesis

Architecture
Compliance
Checking

Fazit

Agile Entwicklung wertet das Reagieren auf Änderungen höher als das Befolgen eines Planes

⚡ zu traditionellen architekturellen Praktiken, welche viel Planung im Voraus brauchen

Das Problem: architekturelle Erosion

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archnemesi

Architecture
Compliance
Checking

Fazit

Architekturelle Erosion

bezeichnet den wachsenden Spalt zwischen
geplanter und tatsächlicher Architektur

Folgen

- ↓ Wartbarkeit
- ↓ Erweiterbarkeit
- ↓ Wiederverwendbarkeit
- ↓ Evolvierbarkeit

[frei nach [Terra, 2012]]

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archnemesis

Architecture
Compliance
Checking

Fazit

- ▶ Entwicklung einer *DSL* für Architekturen
- ▶ Modellierung der Architektur
- ▶ Implementierung von *Architecture Compliance Checking*
- ▶ *nahtloses* Einführen in den Prozess

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archnemesis

Architecture
Compliance
Checking

Fazit

“Operationalisierung”

Die Architektur wird selbst zum Artefakt der Entwicklung. Das Modell wird wie Quellcode behandelt, d.h. es wird unter Versionsverwaltung stehen, Änderungen daran werden durchgesichtet, und Änderungen daran haben konkrete Auswirkungen.

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archonemesis

Architecture
Compliance
Checking

Fazit

- ▶ *nahtloses* Einführen in den Prozess
- ▶ Entwicklung einer *DSL* für Architekturen
- ▶ Modellierung der Architektur
- ▶ Implementierung von *Architecture Compliance Checking*

Das Versuchskaninchen: Saros

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration in das Prozess

Archnemesis

Architecture
Compliance
Checking

Fazit



Jenkins & Gerrit

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration in das Prozess

Archnemesis

Architecture
Compliance
Checking

Fazit

Jenkins

- ▶ Continuous Integration
- ▶ Kompiliert und testet Patches

Gerrit

- ▶ Code Review
- ▶ Erlaubt das Submittieren eines Patches in die produktive Codebasis

Alter Workflow

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

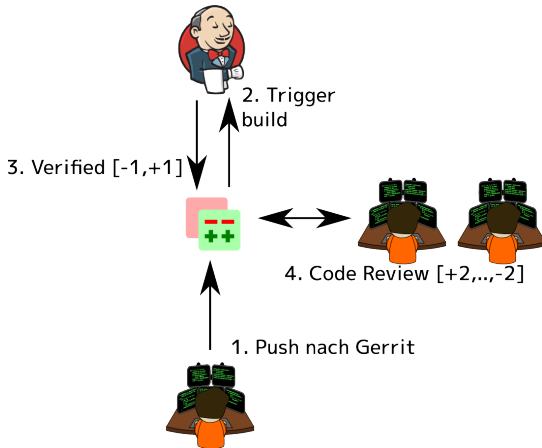
Mein Ansatz:
konstruktive
Qualitätssicherung

Integration in das Prozess

Archnemesis

Architecture
Compliance
Checking

Fazit



Workflow mit Sonarqube

Arsenij E. Solovjev

Einleitung

Architektur
Das Problem

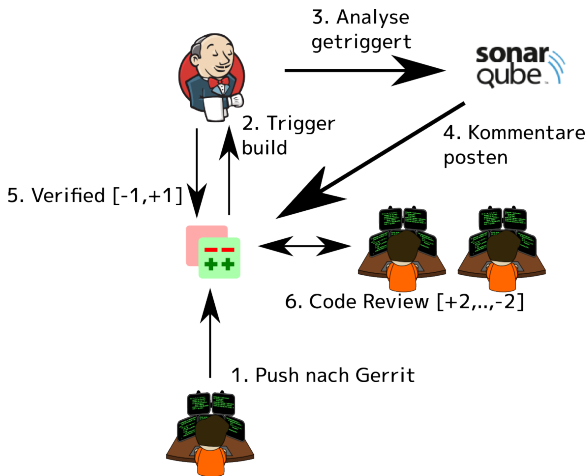
Mein Ansatz:
konstruktive
Qualitätssicherung

Integration in das Prozess

Archnemesis

Architecture
Compliance
Checking

Fazit



Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archnemesiis

Architecture
Compliance
Checking

Fazit

“open platform to manage code quality”

- ▶ statische und dynamische Analyse von Projekten
- ▶ Management der entdeckten “Issues”
- ▶ Erweiterbar und OpenSource (LGPLv3)

Implementierung der Integration

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archnemesiis

Architecture
Compliance
Checking

Fazit

“nahtlose” Integration

- ▶ keine Installation von neuen Werkzeugen
- ▶ kein zusätzlicher Aufwand um den Feedback von Sonarqube einzusehen
- ▶ schneller Feedback

Implementierung der Integration

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archnemesis

Architecture
Compliance
Checking

Fazit

Erweiterung der Jenkins Konfiguration

- ▶ Nach jedem Build für ein Patch werden pro Projekt Sonarqube Analysen sowie die Unittests durchgeführt
- ▶ Verified[+1, -1]

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archnemesis

Architecture
Compliance
Checking

Fazit

Scala Skript welches

- ▶ sammelt die Issues von der Analyse
- ▶ postet sie als Kommentare in das Gerrit-Review

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

**Integration
in das
Prozess**

Archnemesis

Architecture
Compliance
Checking

Fazit

- ▶ informell, da kleiner Kreis an aktiven Entwicklern
- ▶ positives Feedback
- ▶ im Groben wurde das Ziel der “Nahtlosigkeit” erreicht

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archnemesi

Architecture
Compliance
Checking

Fazit

- ▶ *nahtloses* Einführen in den Prozess
- ▶ Entwicklung einer *DSL* für Architekturen
- ▶ Modellierung der Architektur
- ▶ Implementierung von *Architecture Compliance Checking*

Architecture Description Languages

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archnemesisis

Architecture
Compliance
Checking

Fazit

ADL

formelle Notation zur Modellierung von Architekturen.

übliche Elemente

- ▶ Components
- ▶ Connectors
- ▶ Constraints
- ▶ Rationale

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archnemesis

Architecture
Compliance
Checking

Fazit

- ▶ Sprache für Grammatiken
- ▶ generiert Parser (wichtig für Architecture Compliance Checking)
- ▶ generiert Eclipse-Editor

Die Grammatik

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archnemesiis

Architecture
Compliance
Checking

Fazit

```

1 grammar de.fu_berlin.inf.archnemesiis.Archnemesiis with org.eclipse.xtext.common.Terminals
2
3 generate archnemesiis "http://www.fu_berlin.de/inf/archnemesiis/Archnemesiis"
4
5 Architecture:
6     rootComponent=RootComponent
7     (components+=Component)+
8     (connectors+=Connector)*;
9
10 Element:
11     Component | Connector | Constraint;
12
13 RootComponent:
14     'root' name=ID ':' namespace=PackageName;
15
16 Component:
17     'component' name=ID ':' namespace=PackageName;
18
19 Connector:
20     UniDirectional | BiDirectional;
21
22 UniDirectional:
23     from=[Component] '>' to=[Component];
24
25 BiDirectional:
26     comp1=[Component] '<>' comp2=[Component];
27
28 //Placeholder
29 Constraint:
30     name=ID;
31
32 PackageName:
33     ID ('.' ID)*;

```

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archnemesis

Architecture
Compliance
Checking

Fazit

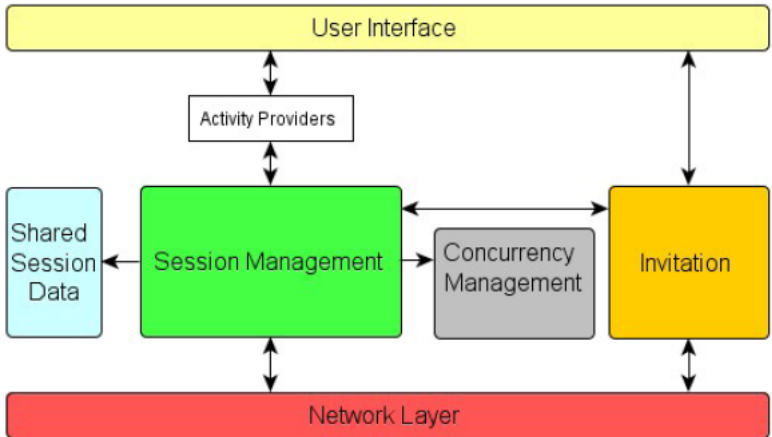


Figure : Patrick Schlott: Analyse und Verbesserung der Architektur eines nebenläufigen und verteilten Softwaresystems

Die Beschreibung

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archnemesis

Architecture
Compliance
Checking

Fazit

```

1  root saros: de.fu_berlin.inf.dpp
2
3  component ui: de.fu_berlin.inf.dpp.ui
4  component invitation: de.fu_berlin.inf.dpp.negotiation
5  component session_management: de.fu_berlin.inf.dpp.session
6  component operational_transformation: de.fu_berlin.inf.dpp.concurrent
7  component network: de.fu_berlin.inf.dpp.net
8
9  session_management => ui
10 session_management => operational_transformation
11 session_management => network
12 session_management => invitation
13 ui => invitation
14 invitation => network
    
```


Literate DSL

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archnemesisis

Architecture
Compliance
Checking

Fazit

Literate Programming

Programme in diesem Stil bestehen aus zwei Elementen.

- ▶ Essay
- ▶ “tangling” Code

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

- ▶ Evaluation fand im Reviewprozess auf Gerrit statt.
- ▶ schneller Feedbackzyklus

Integration
in das
Prozess

Archnemesis

Architecture
Compliance
Checking

Fazit

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archnemesi

Architecture
Compliance
Checking

Fazit

- ▶ *nahtloses* Einführen in den Prozess
- ▶ Entwicklung einer *DSL* für Architekturen
- ▶ Modellierung der Architektur
- ▶ Implementierung von *Architecture Compliance Checking*

Sonarqube Plugin

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archnemesiis

Architecture
Compliance
Checking

Fazit

- ▶ Erste Version als Erweiterung von dem PMD-Plugin
- ▶ Zweite Version als Erweiterung einer Sonarqube Java Rule
- ▶ Beide analysieren auf Ebene des abstrakten Syntaxbaumes

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archnemesiis

Architecture
Compliance
Checking

Fazit

PMD Rule Designer (v 5.1.3-SNAPSHOT)

Language Actions

source code:

```
private Viewer viewer;

private TreeContentProvider additionalContentProvider;

private HeaderElement sessionHeaderElement;
private HeaderElement contentHeaderElement;

private Roster currentRoster;
private ISarosSession currentSession;

private FollowingActivitiesManager followingTracker;

@Inject
private EditorManager editorManager;

@Inject
private AwarenessInformationCollector collector;

public SessionContentProvider(TreeContentProvider additionalContent) {
```

Abstract Syntax Tree / XPath / Symbol Table Data Flow Analysis

- ExtendsList
 - ClassOrInterfaceType: TreeContentProvider
- ClassOrInterfaceBody
 - FieldDeclaration: (private)
 - Type
 - ReferenceType
 - ClassOrInterfaceType: Viewer
 - VariableDeclarator
 - VariableDeclaratorId: viewer
 - ClassOrInterfaceBodyDeclaration
 - FieldDeclaration: (private)

AST Node: FieldDeclaration: (private)

- Scope: SourceFileScope
 - ClassNameDeclaration: Class SessionContentProvider
 - Scope: ClassScope
 - VariableNameDeclaration: Variable: image = 'viewer', line = 40
 - Name occurrence: viewer:70: class net.sourceforge.pmd.lang.java.ast.ASTName
 - Name occurrence: viewer:72: class net.sourceforge.pmd.lang.java.ast.ASTName
 - Name occurrence: viewer:79: class net.sourceforge.pmd.lang.java.ast.ASTName
 - Name occurrence: viewer:88: class net.sourceforge.pmd.lang.java.ast.ASTName
 - Name occurrence: viewer:91: class net.sourceforge.pmd.lang.java.ast.ASTName
 - Name occurrence: viewer:92: class net.sourceforge.pmd.lang.java.ast.ASTName
 - Name occurrence: viewer:106: class net.sourceforge.pmd.lang.java.ast.ASTName
 - Name occurrence: viewer:109: class net.sourceforge.pmd.lang.java.ast.ASTName

Ergebnisse im Überblick

Software

- ▶ DSL zur Beschreibung von Architekturen^a
- ▶ Eclipse Editor für die DSL^b
- ▶ Sonarqube Plugin für Architecture Compliance Checks^c
- ▶ Integration zwischen Issue Reports von Sonarqube und Gerrit

^a<https://github.com/saros-project/archnemesis>

^bEclipse Update Site:

<http://saros-build.imp.fu-berlin.de/update-archnemesis/>

^c<https://github.com/saros-project/archnemesis-sonar-rule>

Prozess von Saros

- ▶ zusätzliches Feedback von FindBugs, Squid etc.
- ▶ operationailisierte Beschreibung der Architektur

Arsenij E.
Solovjev

Einleitung
Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archnemesis

Architecture
Compliance
Checking

Fazit

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archnemesis

Architecture
Compliance
Checking

Fazit

Es gibt noch offene Stellen

- ▶ Schnittstellen könnten besser modelliert werden
- ▶ Sonarqube hat immer noch gelegentlich das “Read Timed out” Problem
- ▶ Das weitere Vorgehen um Constraints ist nicht ganz klar
- ▶ Man kann die Gerrit-Historie nach weiteren Architektur-Eigenschaften von Saros durchforsten, und diese in der DSL ausdrücken

Bibliographie

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Integration
in das
Prozess

Archnemesis

Architecture
Compliance
Checking

Fazit



Ricardo Terra and Marco Tulio Valente and Krzysztof Czarnecki and Roberto S. Bigonha
Recommending Refactorings to Reverse Software Architecture Erosion.
16th European Conference on Software Maintenance and Reengineering, 2012.



Garlan, D. & Shaw, M
An Introduction to Software Architecture,
Advances in Software Engineering and Knowledge Engineering Vol. 1, 1993.

Arsenij E.
Solovjev

Einleitung

Architektur
Das Problem

Mein Ansatz:
konstruktive
Qualitätssicherung

Vielen Dank für ihre Aufmerksamkeit!

Integration
in das
Prozess

Archnemesis

Architecture
Compliance
Checking

Fazit