

Seminar

Beiträge zum Software Engineering

Stephanie Hohenberg

27.06.2019

Roles and Impacts of Hands-on Software Architects in Five Industrial Case Studies

Inayat Rehman

Mehdi Mirakhorli

Rochester Institute of Technology

Meiyappan Nagappan

University of Waterloo

Azat Aralbay Uulu

Matthew Thornton

Rochester Institute of Technology

The Decision View's Role in Software Architecture Practice

Philippe Kruchten, *University of British Columbia*

Rafael Capilla, *Universidad Rey Juan Carlos*

Juan Carlos Dueñas, *Universidad Politécnica de Madrid*

Sollten Software-Architekten programmieren?



Sollten Software-Architekten programmieren?



aktive Teilnahme, besseres Verständnis der Software
mehr Erfahrung als Softwareentwickler



Abdelegieren an Softwareentwickler
Fokus auf eigene Verantwortlichkeiten



empirischer Beweis

Sollten Software-Architekten programmieren?

- 1 Programmieren Software-Architekten überhaupt?
- 2 Welche Art von Code schreiben Software-Architekten?
- 3 Profitieren Softwareprojekte davon?

Vorgehen

1

Einholen quantitativer Daten
industrielle Fallstudie von 5 OS Projekten
↳ Aufstellung von Hypothesen

2

Einholen qualitativer Daten
semi-strukturierte Interviews
↳ Evaluierung der Hypothesen

Auswahl der OS Projekte

Kriterien

- unterschiedliche Anwendungsfelder
- Verfügbarkeit von change logs und commit messages
- Verfügbarkeit von Design Dokumenten
- Anwendung von Architektur Taktiken
- Repräsentieren von großen industriellen Projekten



Sollten Software-Architekten programmieren?

- 1** Programmieren Software-Architekten überhaupt?
- 2 Welche Art von Code schreiben Software-Architekten?
- 3 Profitieren Softwareprojekte davon?

Programmieren Software-Architekten überhaupt?

- 0 Anpassung des Pew and Mavor's Human and Organizational Behavior Model zum Developer's Design Profile
- 1 Erstellen eines Developer's Design Profile für jeden Entwickler im Projekt
 - Skript zur Analyse von github Commits
 - manuelles social media mining
 - ↳ Profile Review
- 2 Gruppierung der Entwickler in Architekten und Nicht-Architekten
 - ↳ Profile Review
- 3 Interview mit Architekten
- 4 Architect's Code Contribution Matrix
 - code contribution extractor script

Sollten Software-Architekten programmieren?

1 Programmieren Software-Architekten überhaupt?

JA

NEIN

2 Welche Art von Code schreiben Software-Architekten?

3 Profitieren Softwareprojekte davon?

Sollten Software-Architekten programmieren?

1 Programmieren Software-Architekten überhaupt?

JA

NEIN

#Architekten = 2-7

%commits von Software-Architekten
27% - 47%

2 Welche Art von Code schreiben Software-Architekten?

3 Profitieren Softwareprojekte davon?

Sollten Software-Architekten programmieren?

- 1 Programmieren Software-Architekten überhaupt? 
- 2 Welche Art von Code schreiben Software-Architekten?
- 3 Profitieren Softwareprojekte davon?

Welche Art von Code schreiben Software-Architekten?

- 0 Code Contribution Extractor Script
- 1 Klassifikation der Commits
 - tactical vs functional
 - tactic classifier
 - ↳ manuelle Code Review
- 2 Analyse Code Contribution Matrix
- 3 Interview mit Architekten

Sollten Software-Architekten programmieren?

1 Programmieren Software-Architekten überhaupt? **JA**

2 Welche Art von Code schreiben Software-Architekten?

functional

tactical

3 Profitieren Softwareprojekte davon?

Sollten Software-Architekten programmieren?

1 Programmieren Software-Architekten überhaupt? **JA**

2 Welche Art von Code schreiben Software-Architekten?

functional

tactical

Software-Architekten
tactical und functional Code

Entwickler
functional Code > tactical Code

3 Profitieren Softwareprojekte davon?

Sollten Software-Architekten programmieren?

1 Programmieren Software-Architekten überhaupt? **JA**

2 Welche Art von Code schreiben Software-Architekten?

functional
& tactical

3 Profitieren Softwareprojekte davon?

Profitieren Softwareprojekte davon?

- 1** Analyse der Code Qualität
 - Analyse zur Defektdichte und Volatilität - change & defect extractor
 - statistischer Test - MWU Mann Whitney U Test
- 2** Interview mit Architekten
- 3** Erstellen einer Design Ownership Metrik

Sollten Software-Architekten programmieren?

1 Programmieren Software-Architekten überhaupt? **JA**

2 Welche Art von Code schreiben Software-Architekten?

functional
& tactical

3 Profitieren Softwareprojekte davon?

JA

NEIN

Sollten Software-Architekten programmieren?

1 Programmieren Software-Architekten überhaupt? **JA**

2 Welche Art von Code schreiben Software-Architekten?

functional
& tactical

3 Profitieren Softwareprojekte davon?

JA

NEIN

Profitieren Softwareprojekte davon?

1

tactical Code öfter geändert und mehr Bugs

↳ Hypothesen

H1 mehr Bugs wegen strengere Tests

H2 komplexer und schwieriger zu implementieren für einen Entwickler

2

weniger Tests (\leftrightarrow H1)

komplexer zu implementieren und zu warten

Unerfahrenheit

firefighting Situationen

↳ mehr Bugs

↳ Lösungen

reguläre Kommunikation

Reviews oder test-first development

3

tactical Code von Architekten haben weniger Bugs

Sollten Software-Architekten programmieren?

1 Programmieren Software-Architekten überhaupt? **JA**

2 Welche Art von Code schreiben Software-Architekten?

functional
& tactical

3 Profitieren Softwareprojekte davon? **JA**

Sollten Software-Architekten programmieren?



aktive Teilnahme, besseres Verständnis der Software
mehr Erfahrung als Softwareentwickler



Abdelegieren an Softwareentwickler
Fokus auf eigene Verantwortlichkeiten

Meine Kritik

Auswahl der Projekte

- OS Projekte != Industrie
- keine Analyse von Projekten mit Architekten, die nicht Coden

Design Profile

- binäre Bewertung in den Kompetenzfelder des Design Profiles
- keine Vergleich von Entwickler und Architekten Profilen
bzgl der Coding-Erfahrung

zu wenig Ursachenforschung, warum Entwickler mehr Bugs einführen

The Decision View's Role in Software Architecture Practice

Philippe Kruchten, *University of British Columbia*

Rafael Capilla, *Universidad Rey Juan Carlos*

Juan Carlos Dueñas, *Universidad Politécnica de Madrid*

Architekturentscheidungen

Sichten

- logisch, funktional
- Implementation
- Auslieferung
- Performanz
- Anwendungsfall

Herausforderungen

- Verwaltung und Wartbarkeit des Wissens
- Abstimmung von Code und Entscheidungen, vor Allem bei Änderungen



historische Evaluierung der
Repräsentation von Architekturentscheidungen

Repräsentation von Architekturentscheidungen

Motivation

- Wissenstransfer
- Verständnis der Architektur und vergangenen Entscheidungen
- Einflussanalyse
- weniger Wartungskosten

Herausforderungen

- Integration im Softwareentwicklungsprozess
- overhead im frühen Stadium des Projektes

Repräsentation von Architekturentscheidungen

Schritte

1. Angabe genereller Informationen
Name, Beschreibung, +, -, Status, Kategorie
2. Definition der Anforderungen
3. Evaluierung alternativer Entscheidungen
4. Definition von Beziehungen zwischen Entscheidungen
5. Verbindung mit Architektur
6. Kommunikation

Meine Hypothesen

Architekten müssen nicht zwingend programmieren,
wenn Sie ihre Entscheidungen ausreichend dokumentieren und kommunizieren

Architekten, die programmieren,
könnten die anderen Entscheidungsperspektive außer Acht lassen.

und wie sieht das ganze in der Praxis aus?

adesso

semistrukturiertes Interview mit einem Softwarearchitekten