Elmar Frerichs

Freie Universität Berlin

Komplexität von Refactoring im Kontext

Verteidigung und Diskussion

Übersicht

- Einführung: Begriffe und Ansätze
- Stand der Literatur
- Fragestellungen
- Vorgehen
- Ergebnisse
- Ausblick

Einführung: Refactoring

Eine Verbesserung der internen Struktur des Quellcodes, ohne das externe Verhalten zu ändern.

- Was ist extern, was intern?
- Gibt es Zusammenhänge zu Verhaltensänderungen?
- Wer veranlasst das Refactoring?

Einführung: Ansatz der Untersuchung

- Möglichkeiten nach Murphy-Hill 2008:
  1. Analyse der Commit-Nachrichten
  2. Analyse der Code-Änderungen
  3. Beobachtung von Programmierepisoden
  4. Mitschreiben der Verwendung von Werkzeugen

- Ansatz 2 erlaubt nachträgliche Analyse im Detail
  - Großer Datensatz erfordert starke Einschränkung der Suche
  - Weniger Abhängig von der Wahl der Projekte
  - Ich kann bisher nicht betrachtete Aspekte hervorheben

Stand der Literatur

- Diverse systematische Literaturübersichten (SLR), sogar tertiäre
  - Häufig Anwendung neuer Werkzeuge oder Metrik auf festen Softwarestand
- Analyse der Historie
  - Commit-Nachrichten: AlOmar et al. 2019-2021
  - Änderungen des Quellcodes
    - Werkzeuge: Ref-Finder (Kim et al. 2010) und RefactoringMiner (Silva et al. 2016)
    - Darauf aufbauend: Pantiuchina et al. 2020

Fragestellungen

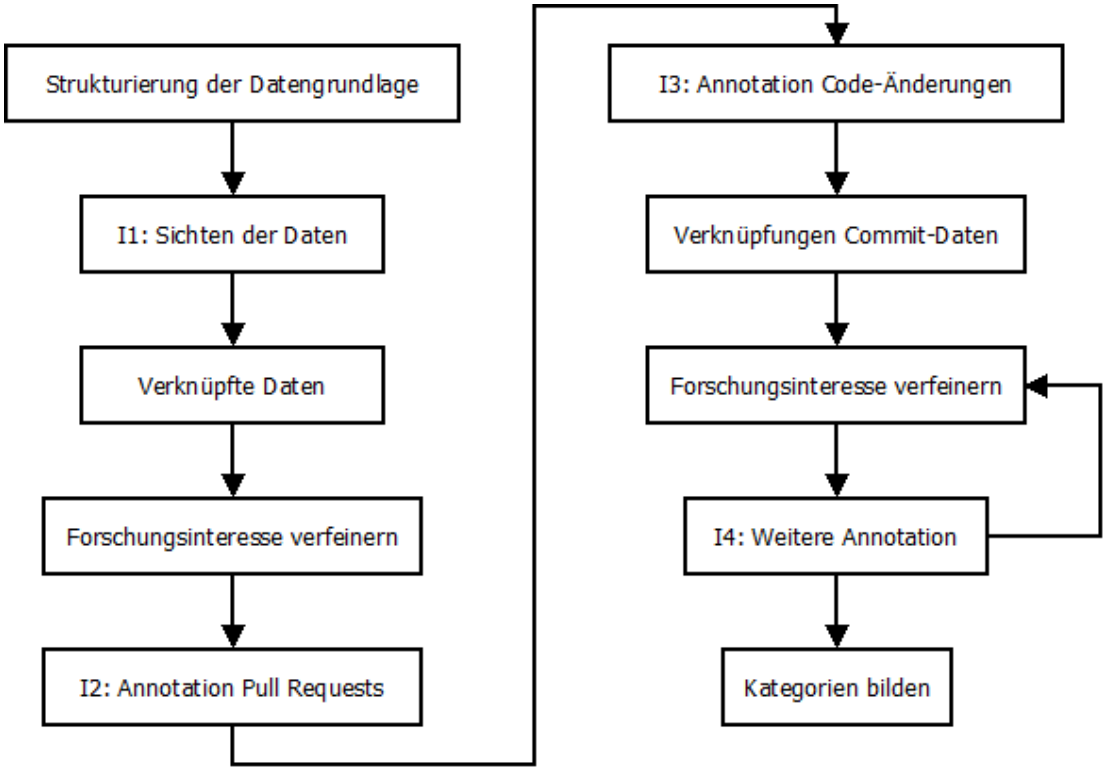
- Gibt es verschiedene Arten von Pull Requests, welche bisher nicht unterschieden wurden?
  - Welche Kriterien sind dafür relevant?
- Wie entwickelt sich die Diskussion in diesen Fällen?
- Welche Hindernisse offenbaren sich bei der Untersuchung?

Die richtigen Fragen

- Herausforderung für mich: Überblick gewinnen
  - über bereits bearbeitete Fragestellungen
  - über mögliche Lücken
  - über angebrachte Systematik für das Vorgehen

- Es ist schwierig, die richtigen Fragen zu stellen, um die richtigen Antworten zu bekommen

Vorgehen: Prozess



Vorgehen: Kontext von Pull Requests

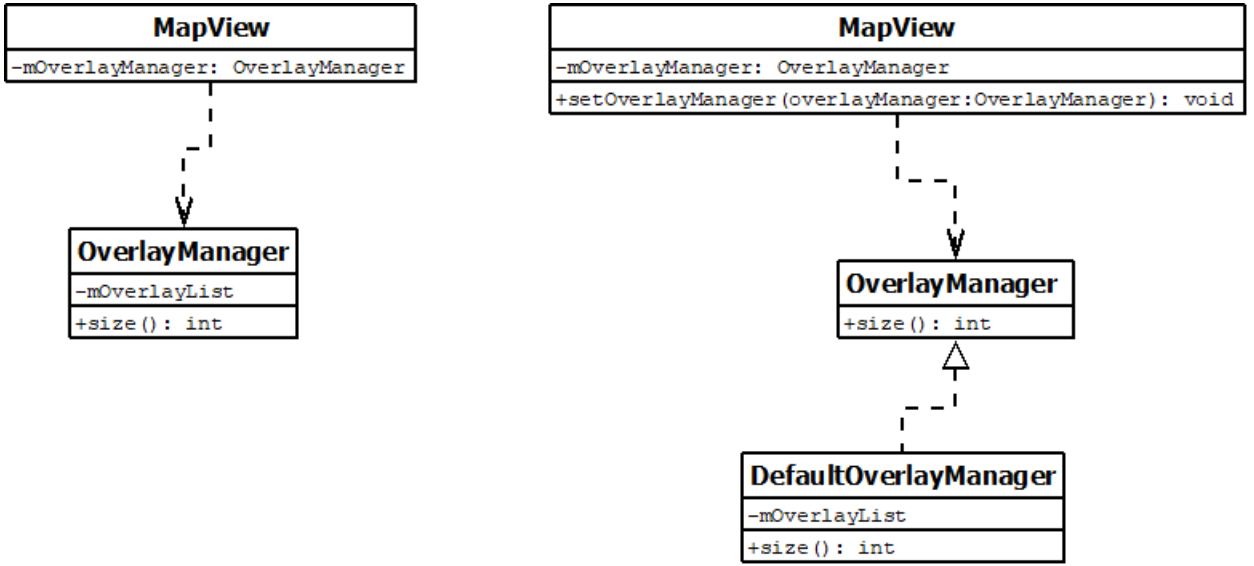
- Verknüpfte Issues (Titel, Commits, Verlinkung)
- Externe Dokumentation (z.B. Confluence)
- Inhalt und Interaktion der Diskussion (Beteiligte, Gegenstand)
- Baum der Änderungen in git
  (vorherige und nachlaufende Änderungen)
- Metadaten der Commits (Datum, Ersteller, Nachricht)

Ergebnisse: Reines Refactoring

- Pull Request ausschließlich mit Refactoring
  - Keine Änderung externen Verhaltens
  - Strukturverbesserung des Quellcodes

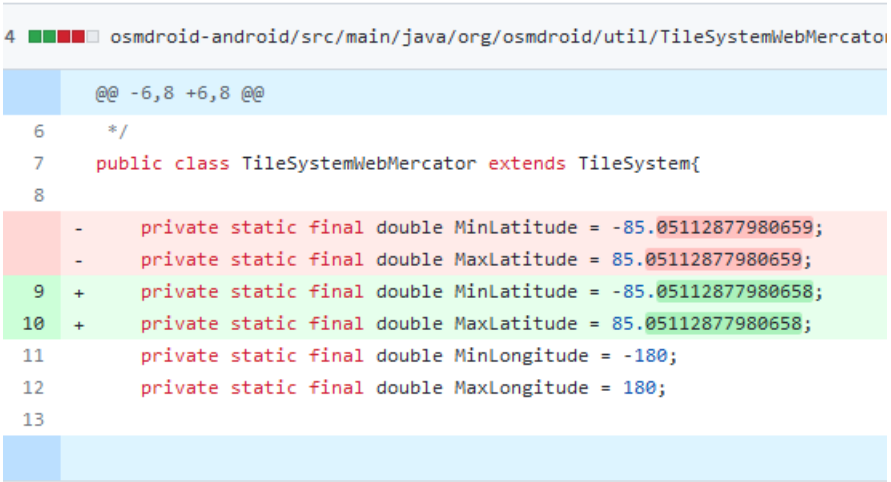
- Kann im Zusammenhang zu anderen externen Änderungen stehen

Reines Refactoring: Beispiel



Reines Refactoring?

- Vorheriges Beispiel: PR 179 aus Projekt osmdroid
  - Streng genommen ist MapView aber Teil der öffentlichen Schnittstelle -> externes Verhalten
- 3 weitere Beispiele mit kleineren Abweichungen



Reines Refactoring: Eigenschaften

- Wenig Diskussion zu den vorliegenden Fällen
  - 2x Nachfrage zur konkreten Umsetzung
  - 2x zusätzlicher, nicht themenverwandter Vorschlag
  - Struktur selbst nicht infrage gestellt

- Aufteilung der Pull Reqests ist zwischen Projekten unterschiedlich
  - Bei osmdroid werden die PR kleiner aufgeteilt
  - DSpace hat meist einen PR pro Jira-Ticket, damit alle Änderungen zusammen

Ausblick

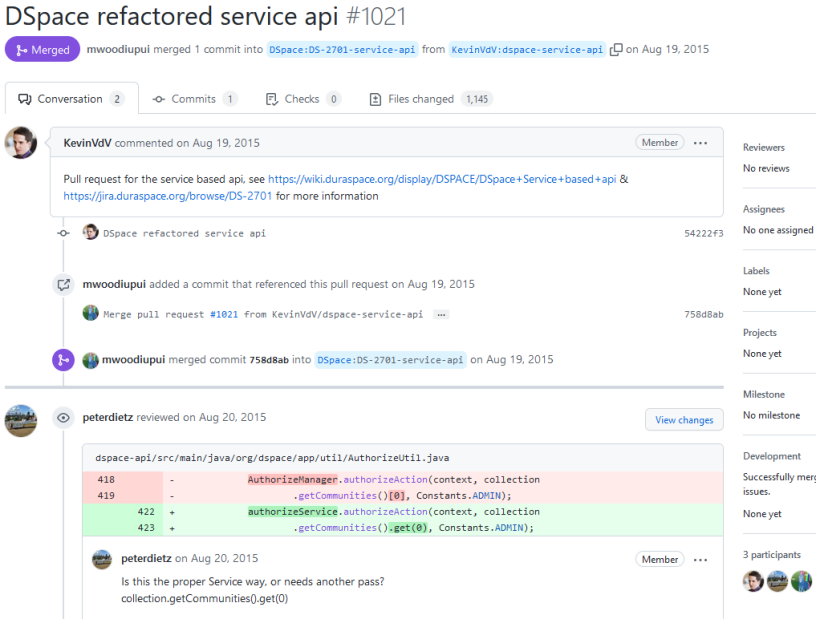
- Vertiefung Reine Refactorings
  - Breitere Datenbasis mit mehreren Projekten
  - Bessere Abgrenzung in Graubereichen
- Verbesserte Zuordnung von Kontext zu Pull Request
  - Automatisierte Verknüpfung der Issues
  - Berücksichtigung Erstellungsdatum von Commits
- Verwendung des Begriffs unter Entwicklern

Komplexität von Refactoring im Kontext

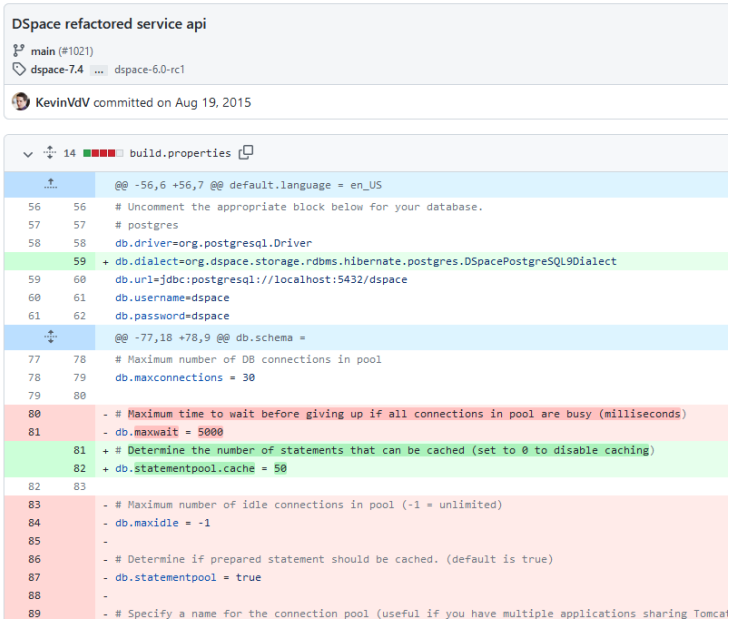
DISKUSSION

Backup: Fachwörter Gitlab

Pull Reqest



Commit



Backup: Datum eines Commits

Assume the following history exists and the current branch is "topic":

```
          A---B---C topic
         /
    D---E---F---G master
```

From this point, the result of either of the following commands:

```
git rebase master
git rebase master topic
```

would be:

```
                A'--B'--C' topic
               /
    D---E---F---G master
```

Search

Diff Old version New version Lines of context: 3 Ignore

```
Author: KevinVdV <kevin@mire.be>  2014-11-08 09:19:09
Committer: KevinVdV <kevin@mire.be>  2015-08-19 11:54:32
Parent: fcb3717aad396924f8ddbca069f98acd3c6dd414 (Merge pull re
Child:  758d8abdadee1979046d7ea6241956b4c2a23ec9 (Merge pull re
Branches: master,
```

Quelle: Git Dokumentation git-rebase

Quellen

- Bild Backup PullRequest: GitHub, URL: https://github.com/DSpace/DSpace/pull/1021
- Bild Backup Commit: GitHub, URL: https://github.com/DSpace/DSpace/pull/1021/commits/54222f3c1d3c63bd03af263703a950e9cfad7439
- Bild Backup git-rebase: git-rebase Dokumentation. URL: https://git-scm.com/docs/git-rebase