

Modern Code Reviews in Open-Source Projects: Which Problems Do They Fix?

Moritz Beller,
Alberto Bacchelli, Andy Zaidman
Delft University of Technology
The Netherlands
{m.m.beller, a.bacchelli, a.e.zaidman}@tudelft.nl

Elmar Juergens
CQSE GmbH
Germany
juergens@cqse.eu

MSR'14, May 31 – June 1, 2014, Hyderabad, India
ACM 978-1-4503-2863-0/14/05

背景

- インспекションの変化版であるMCR(モダンコードレビュー)は商業用とOSSのどちらのシステムにも使われている
 - インспекション: 実際にプログラムを動かすことなく、人間の目で見て不具合などを検証する作業
 - MCR: ツールを使うなど対面式でないレビュー、不具合の検出だけでなく情報の共有も対象とする
- インспекションは1970年代から研究されているが、MCRは最近の分野でまだあまり研究されていない

問題点

- MCRの先行研究
 - **Bacchelli & Bird**: MicrosoftでMCRの動機付けと結果を出した
 - **Mantyla & Lassenius**: 3つの商業的なソフトウェアシステムでレビューの中で見つかる不具合の種類を分類した
- 先行研究はレビューワーカーのコメントからMCRが見つかる不具合を調査しているが...
 - コメントがきっかけでないレビューがあるなど、コメントが軽視されていてデータに問題がある

目的

- MCRプロセスの成果の理解を深める
- RQ
 1. どの変更がレビューされるか
 - **Evolvability Changes vs Functional Changes**
 2. 何がレビューのきっかけになるか
 - コメントがあるかないか
 3. 何がファイル変更数に影響するか
 - 各要因の寄与率を調べる

データセット

- コードレビューは多くのプロジェクトで厳密に行われていない
 - 例: Tycho(Eclipseのサブプロジェクト)
 - “機能的な変更がなければGerritに投稿する必要はない”
- 対象: ConQAT、GROMACS
 - コミット毎に強制的にレビュープロセスがある

Table 1: Comparison of ConQAT and GROMACS.		
	ConQAT ⁸	GROMACS ⁹
Access to Code	Public releases	Public VCS
Access to TMS	On request	On the website
Development time	≥ 8 years	≥ 18 years
# of Developers	12 active, ~50 total	~16 active, ~44 total
# of Reviewers	5 active	13 active
Language	Java	C (mostly)
SLOC	260,465	1,449,440
# of Tasks	~2,500	~1,200
Code Reviews since	2007	2011
Review mandatory	Yes	Yes
Review tool	None (Eclipse)	Gerrit
# of Reviewers/Task	1	≥ 2
# of Review Rounds	[1; ∞[	[1; ∞[
Samples	ConQAT-rand, ConQAT-100	GROMACS-rand

アプローチ

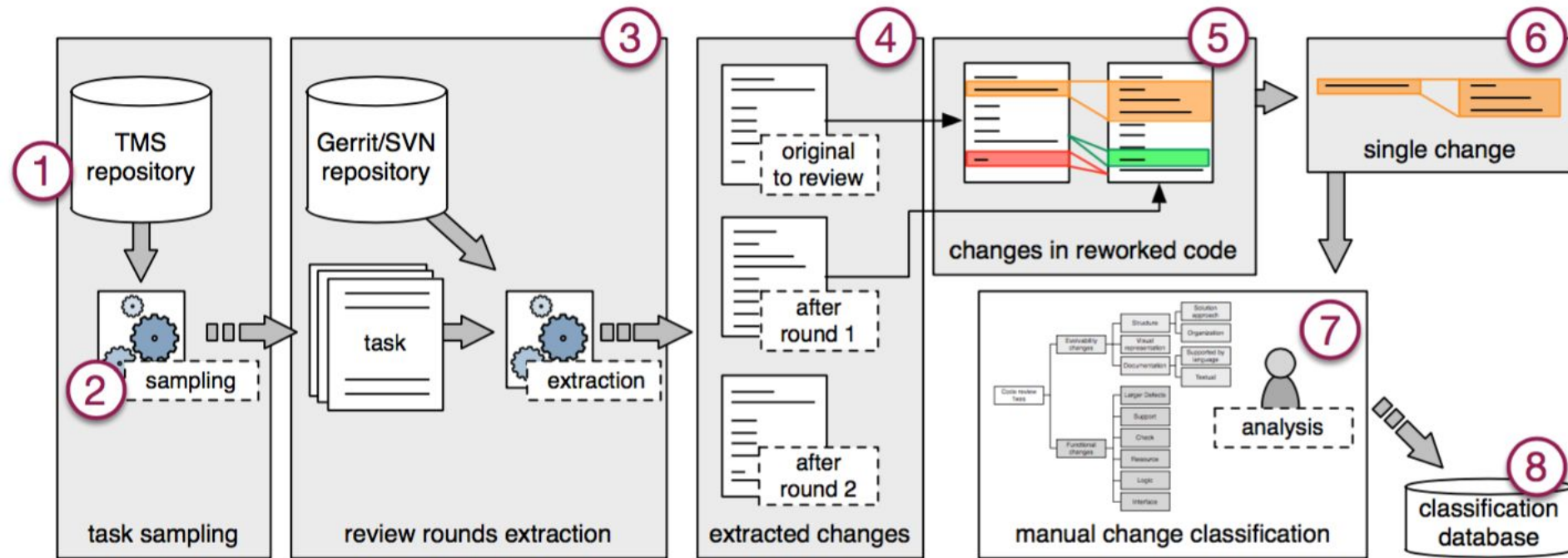
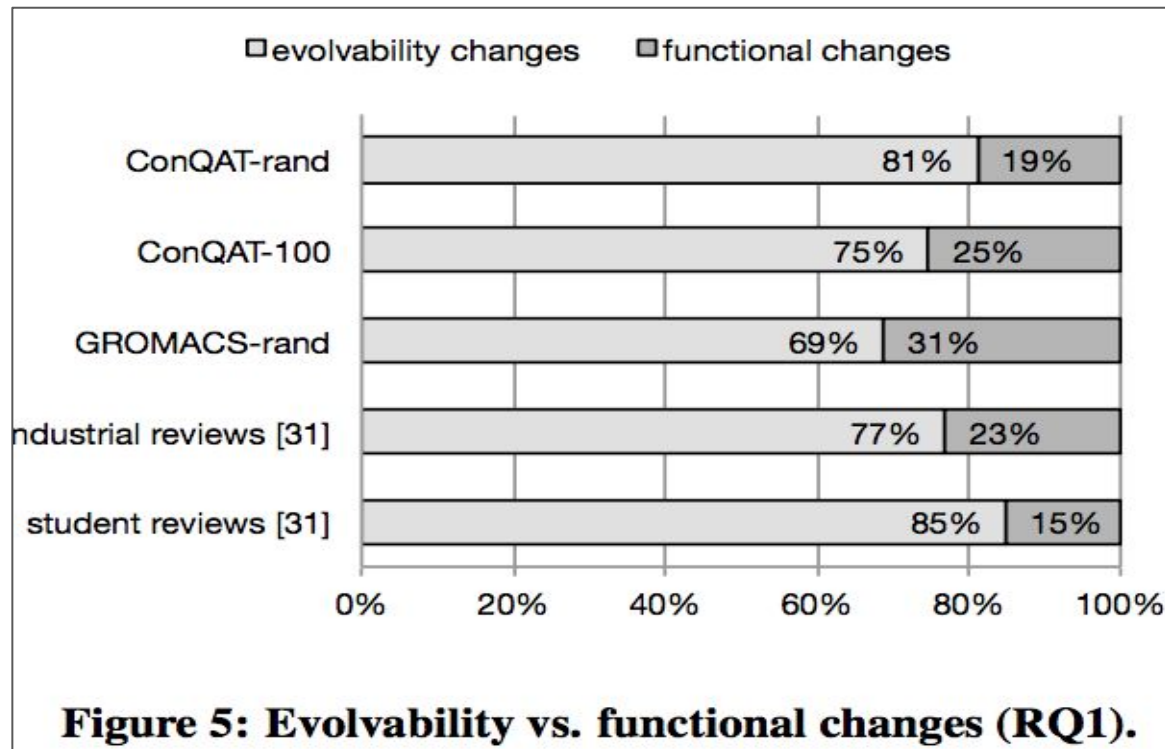


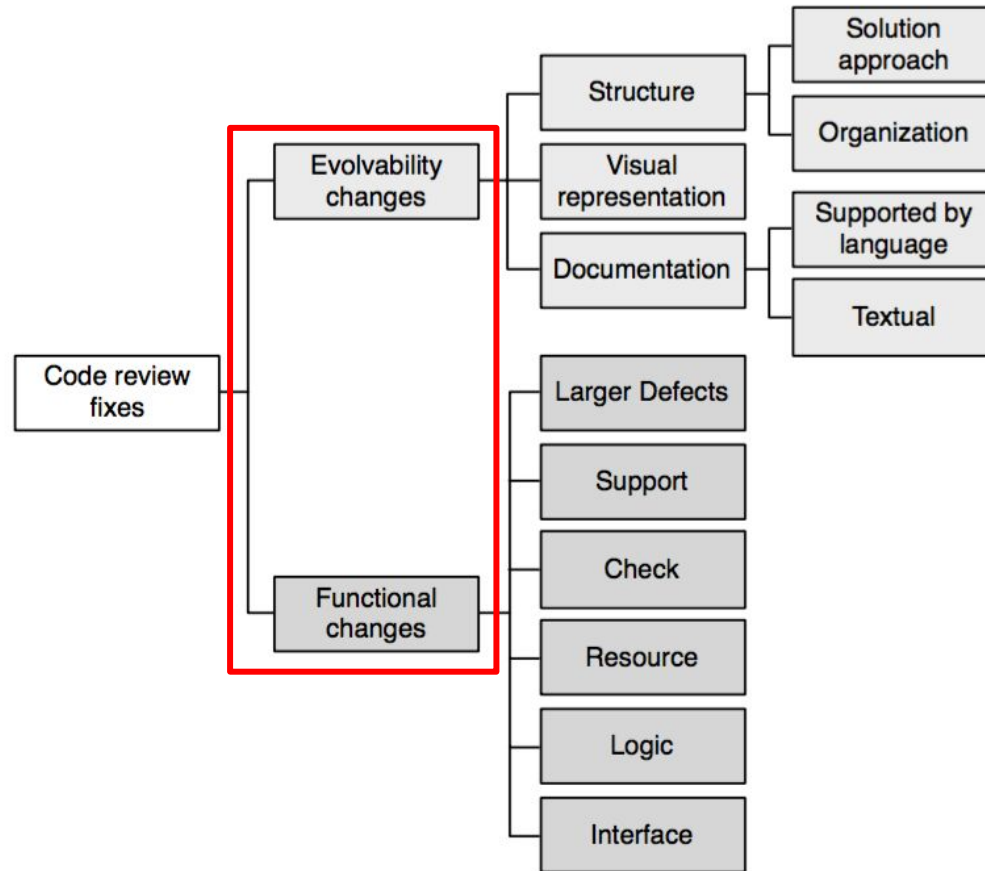
Figure 2: Research method applied to classify changes of code under review.

結果 RQ1:どんな変更がレビューされるか



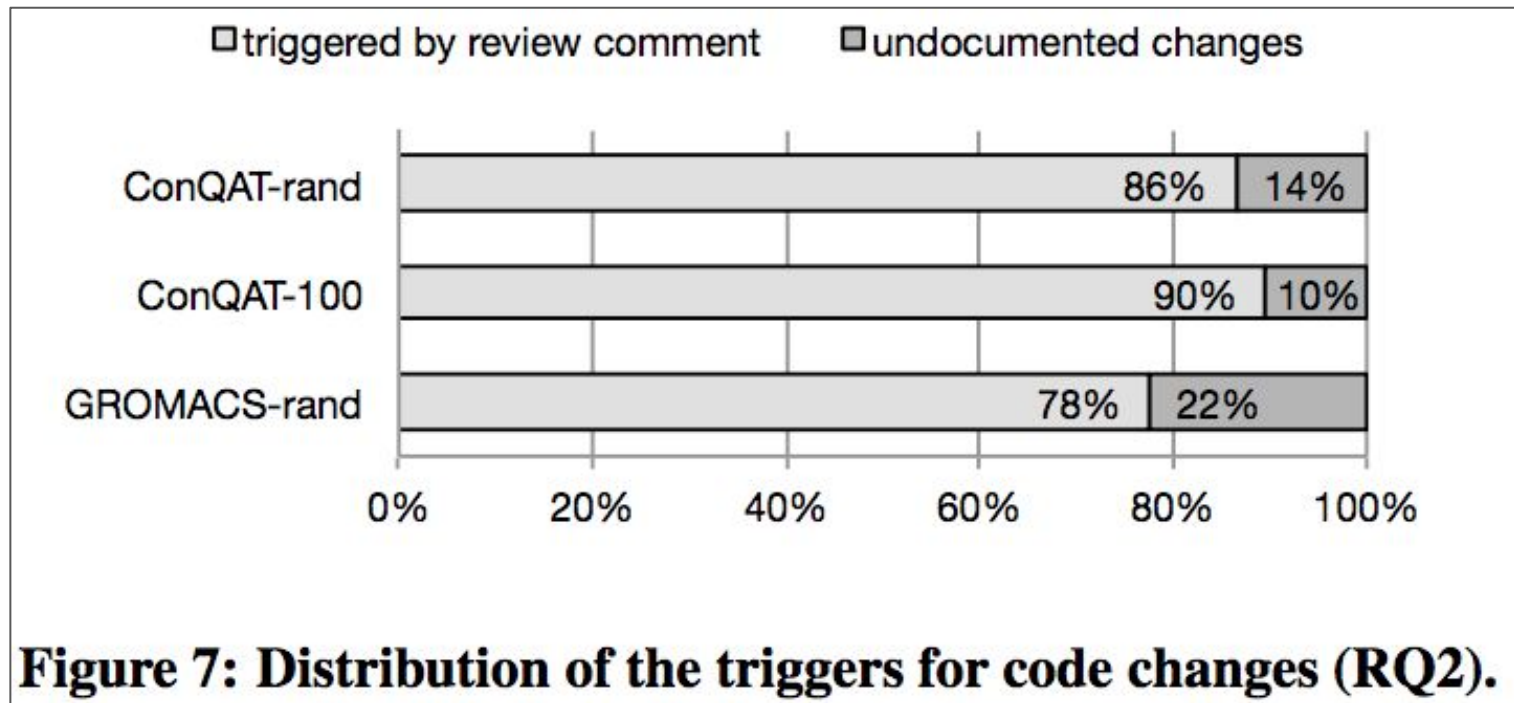
- 69-85%がEvolvability Changes
- ほとんど75:25の比に当てはまった
 - インспекションでの先行研究と同じような値

RQ1: 不具合の分類方法



- IBMとHPによって主に2つに分類されている
- サブ的な分類は今回は省略(数が少ないので)

結果 RQ2: 何がレビューのきっかけになるか



- **78-90%のレビューはコメントがトリガーになっている**
 - ドキュメントがないものは著者が自発的にレビューしていると判断
 - インタビューより

結果 RQ3:何がファイル変更数に影響するか

Table 3: Calculated model coefficients for RQ3 (see [8]).

Coefficient	Value	Significance Level
<i>Error Term ϵ</i> (Intercept)	-34.0042	1.0000
<i>Code Churn β_1</i>	0.0026	$2.34 \cdot 10^{-16}$ **
<i>Changed Files β_2</i>	0.0483	$< 2 \cdot 10^{-16}$ **
<i>Package</i>	β_3	
org.conqat.engine.blocklib	-36.9484	1.0000
org.conqat.engine.commons	0.7152	0.4649
org.conqat.ide.editor	0.1793	0.8633
org.conqat.ide.index.dev	-39.3108	1.0000
...		
<i>Task Type</i>	β_4	
adaptive	0.5277	0.1283
corrective	-0.6508	0.0496 *
perfective	0.7015	0.0138 *
preventive	-0.7289	0.1118
<i>Author</i>	β_5	
deissenb	32.4496	1.0000
juergens	32.7963	1.0000
...		

Significance codes: ** = 0.001, * = 0.05

まとめ

- 今回、MCRプロセスを実際に調べることで理解を深めた
- 結果として以下がわかった
 - レビューされるのは**69-85%**が**Evolvability Changes**
 - **78-90%**のレビューは**コメント**がきっかけ
 - ファイル変更数に寄与する要因は??