

Culture is Key: Results of a Survey on Factors Influencing Code Review Adoption

Tobias Baum, Hendrik Leßmann and Kurt Schneider

Leibniz University Hannover

56th CREST Open Workshop

Research Context

- I'm a PhD student in Hanover, Germany
- My goal: Improving code review use in industry

Cognitive support review tools

Help the reviewer work more efficiently by moving... and to the computer...

Interested?
<http://tobias-baum.de>

Baum and Schneider, „On the Need for a New Generation of Code Review Tools“, PROFES'16

Baum et al., „On the Optimal Order of Reading Source Code Changes for Review“, ICSME'17

Industrial review practices

Why do/don't teams use reviews?
 How do teams in industry... (especially... code review...)

Baum et al., „A Taxonomy of Code Review Processes“, QRS'16

Baum et al., „Factors Influencing Code Review Processes in Industry“, FSE'16

Baum et al., „Comparing Pre Commit Reviews and Post Commit Reviews Using Process Simulation“, ICSSP'16

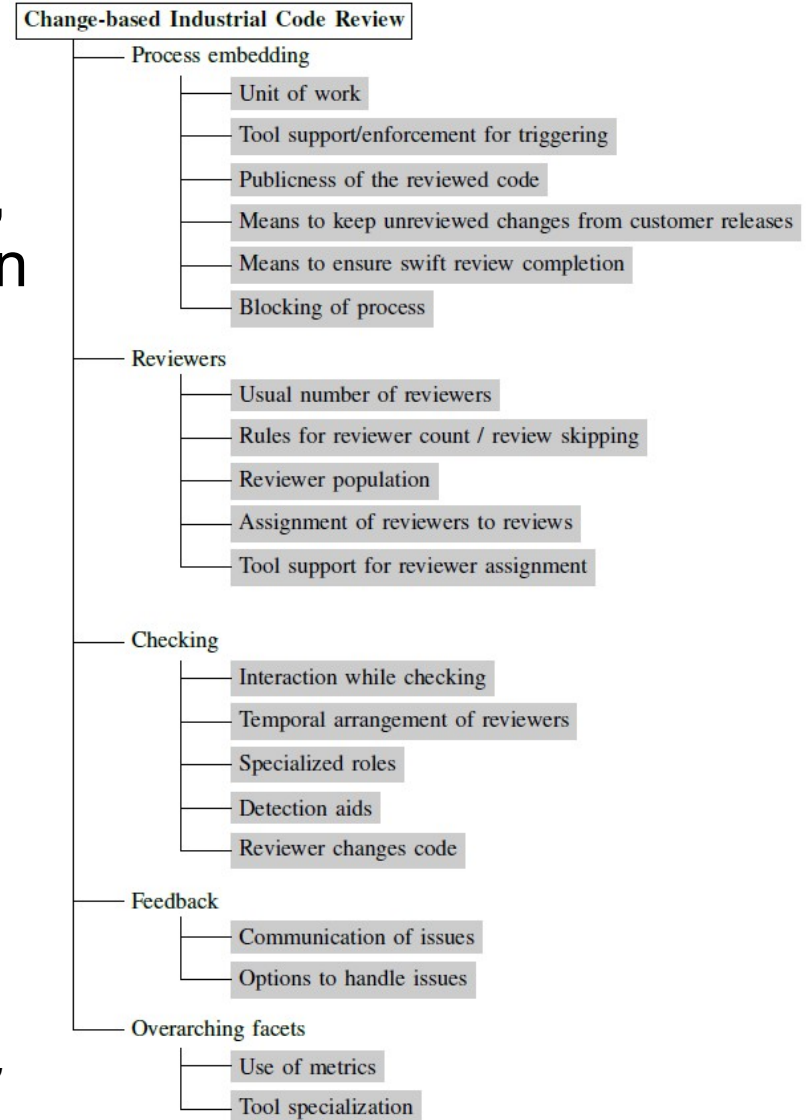
Focus of this talk

Code Review Survey

- Our (and others) previous research often analyzed code review processes in industry qualitatively
- The last large scientific survey on reviews in industry was conducted 15 years ago (Ciolkowski et al.)
- To gather recent quantitative data and to test some of the hypotheses put forward in our previous research, we conducted a large scale survey on industrial code review practices
- Outline of talk
 - Foundations and methodology of survey
 - Exploration of hypothesis on culture and review adoption
 - Presentation of further selected observations from the survey

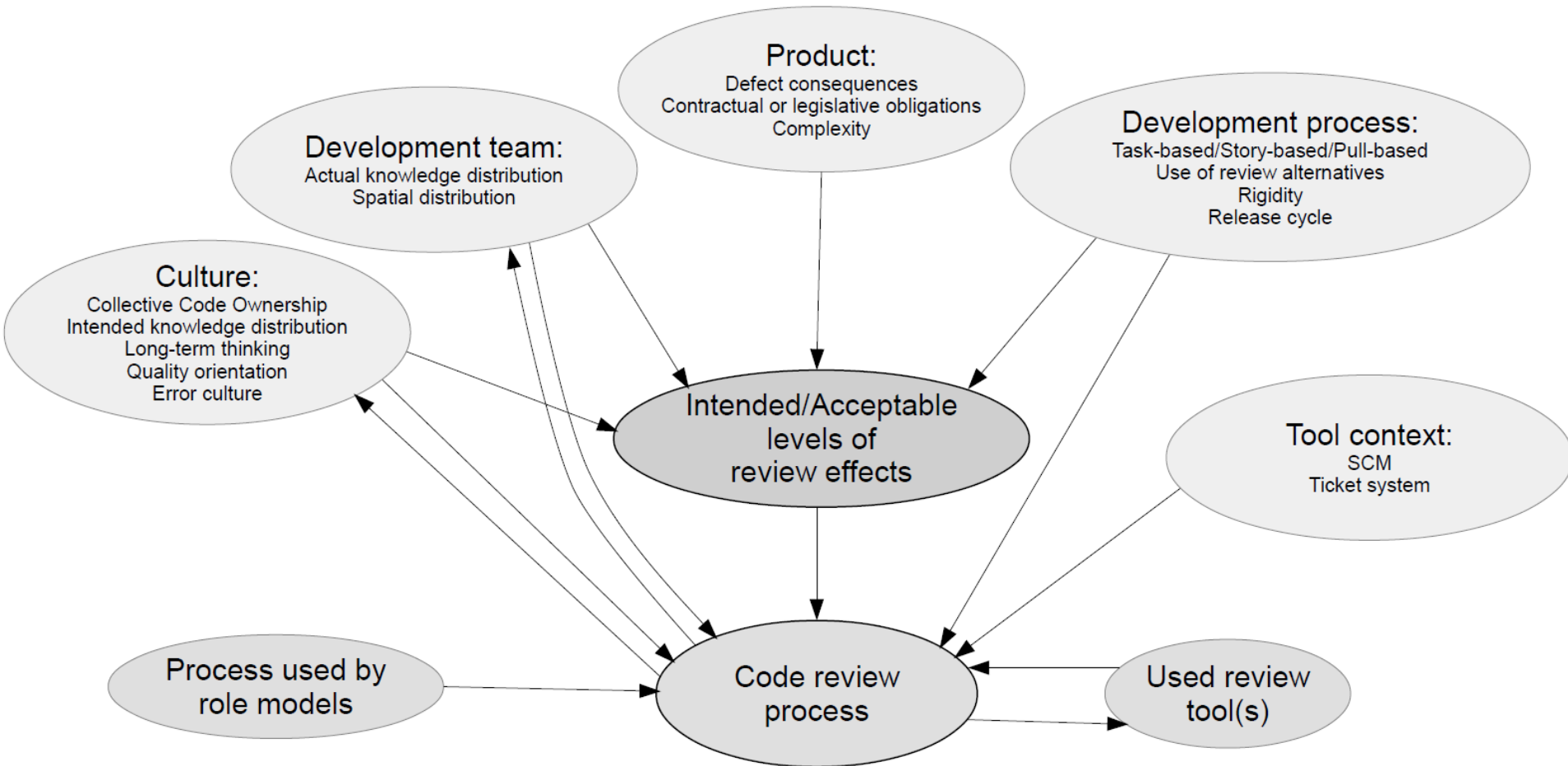
Background: Variation in Review Processes

- Change-based review processes in industry differ in many details
- To systematize these differences, we created a faceted classification scheme



Baum et al. "A Faceted Classification Scheme for Change-Based Industrial Code Review Processes.", QRS '16

Background: Factors Influencing the Process

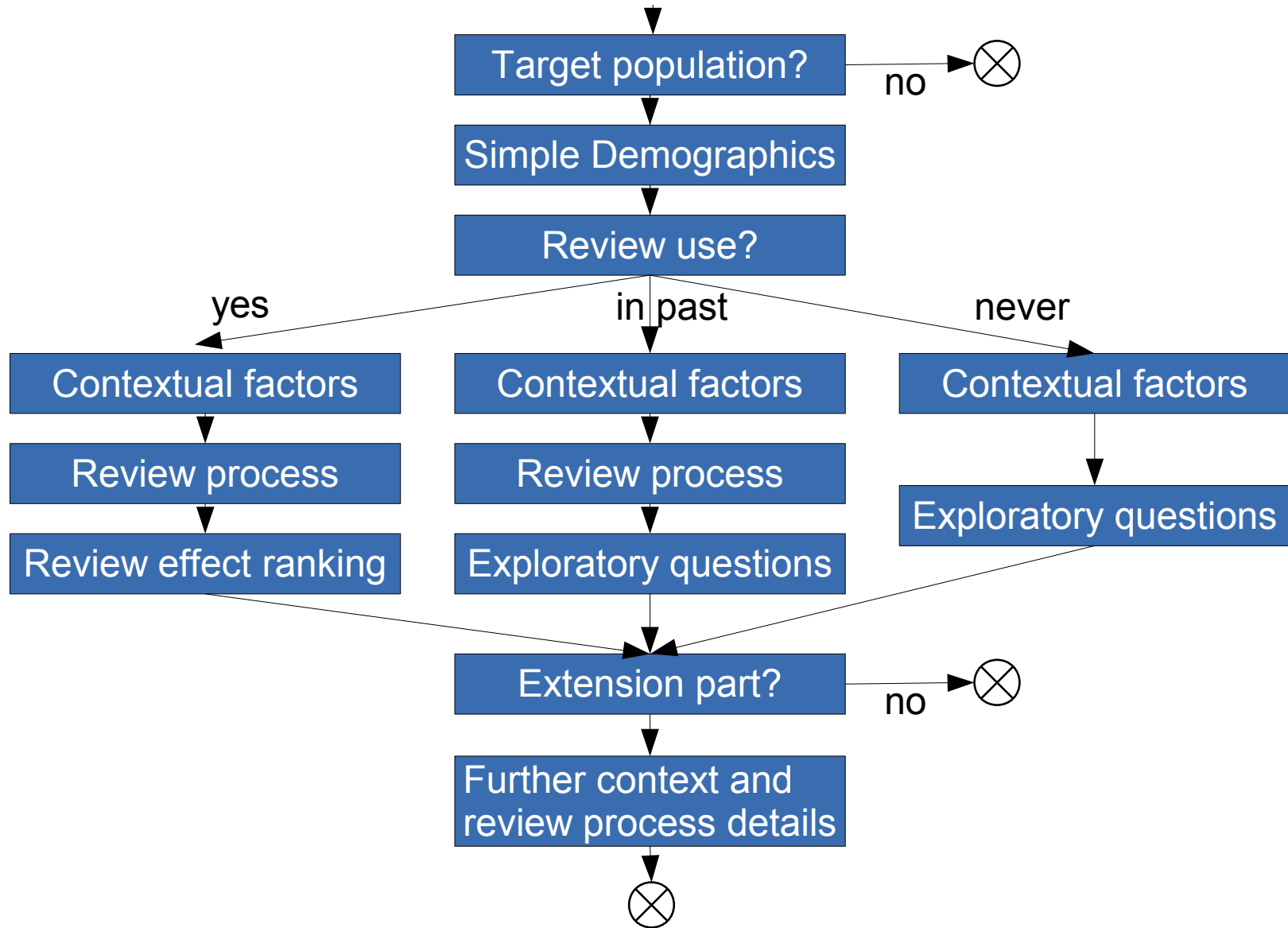


Baum et al. "Factors influencing code review processes in industry.", FSE '16

Methodology

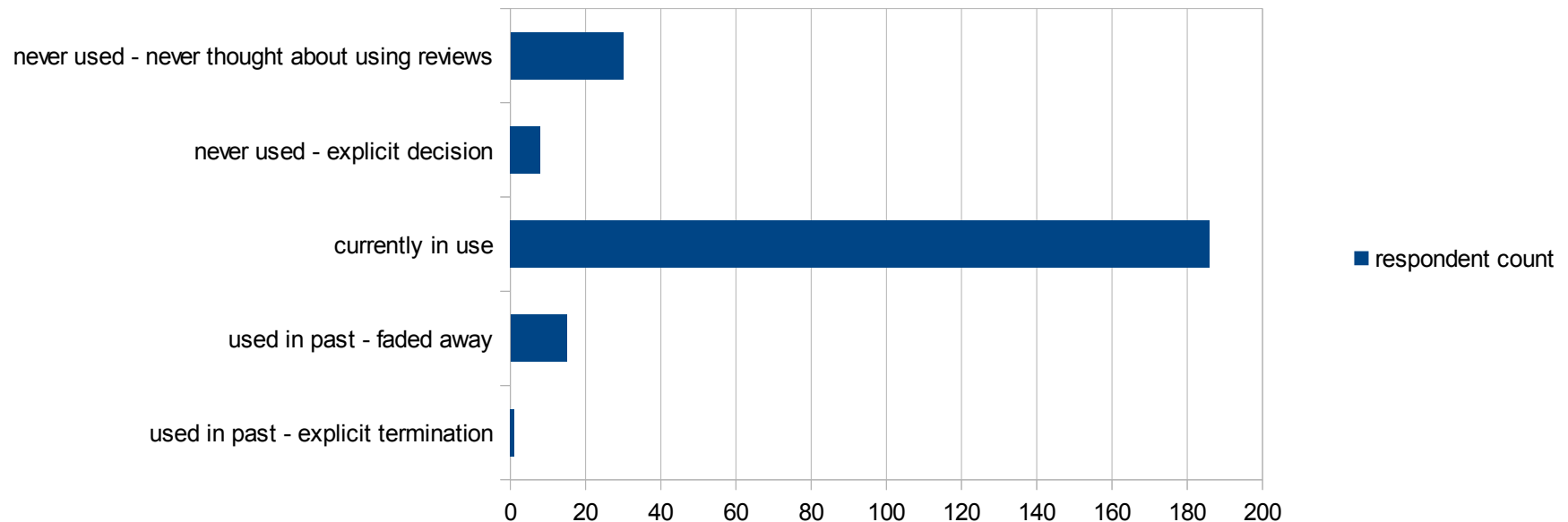
- Many questions based on our previous qualitative studies
- Online survey, conducted in February/March 2017
- Population: Commercial software development teams
- Sampling: Invitation of participants through ...
 - Mailing lists
 - Online communities (e.g. Facebook, Xing)
 - Personal contacts
 - We tried to limit bias by avoiding channels related to code review tools (e.g. GitHub, Gerrit)
- 240 valid responses from target population
- 76% from Germany, 24% from 18 other countries

Survey outline (simplified)



Use of reviews in the sample

- 78% of the teams currently use code reviews
 - Beware: Likely biased by sampling (self-selection)
- Only one respondent indicated they explicitly decided to stop reviews



Hypothesis: Culture influences review adoption

What inhibits teams so that they don't start reviews?

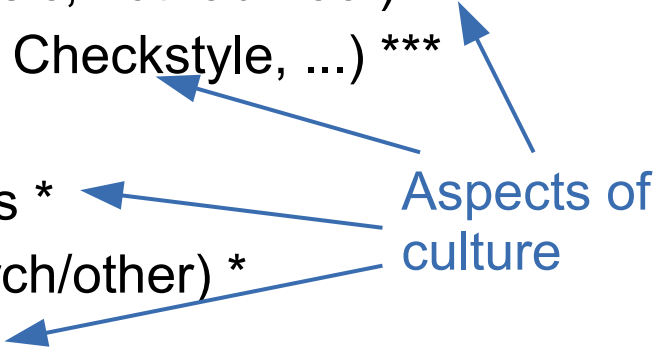
"With developers that are in the business for a long time, it's difficult. You often have the attitude that it's their code, it belongs to them, and you shouldn't meddle with it." (l. 18)

"Quality of the code is the least important" (l. 16)

Hypothesis: When code reviews are not used at all, this is mainly due to cultural and social issues. Needed time and effort are another important, but secondary, factor.

Baum et al. "Factors influencing code review processes in industry.", FSE '16

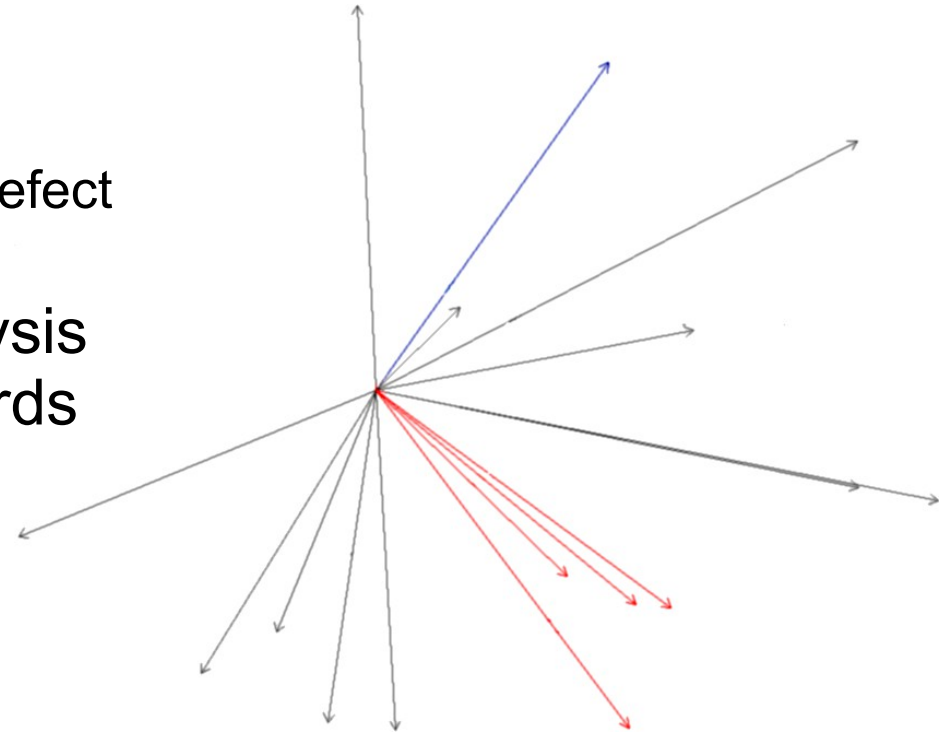
Inhibitors of review use

- To explore the hypothesis, we built a model to predict review use
 - Selected subset of factors (Weka CfsSubsetEval)
 - Built a logistic regression model
 - Factors with highest influence in the logistic regression model
 - defined development process (agile or classic, not “ad hoc”) ***
 - some use of static analysis (e.g. FindBugs, Checkstyle, ...) ***
 - team size > 4 *
 - intended knowledge distribution: generalists *
 - type of created software (not games/research/other) *
 - positive error culture *
- 

***: $p < 0.001$, *: $p < 0.5$

Static analysis is culture?

- Principal Component Analysis shows ...
 - Use of static analysis is related to
 - Intensity of testing
 - Long-term thinking
 - Collective code ownership
 - But it is orthogonal to expected defect consequences
- Hypothesis: Use of static analysis is a proxy for “orientation towards (source code) quality“



Qualitative survey data

The survey also adds further qualitative support for the hypothesis, e.g.:

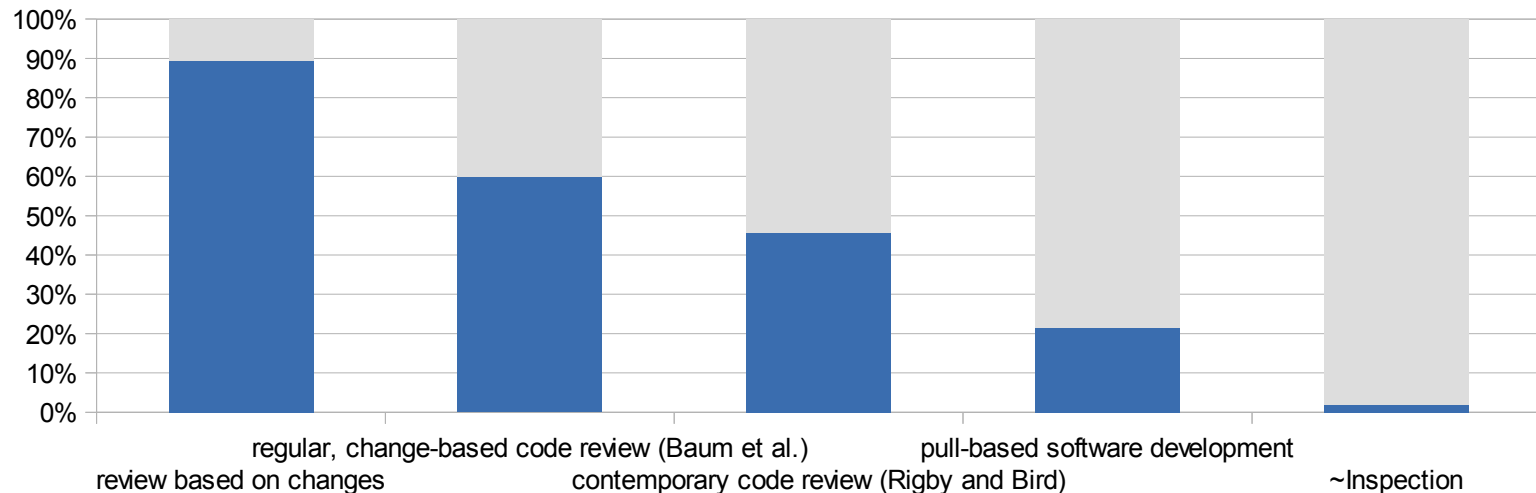
“It is an exhausting tasks, leads to many difficult conversations specially when people cannot handle criticism well or cannot give criticism in a constructive manner.”

“It is currently not used due to our historic way of working, capacity and culture.”

“The project lead does not want them and is more concerned with 'Features'”

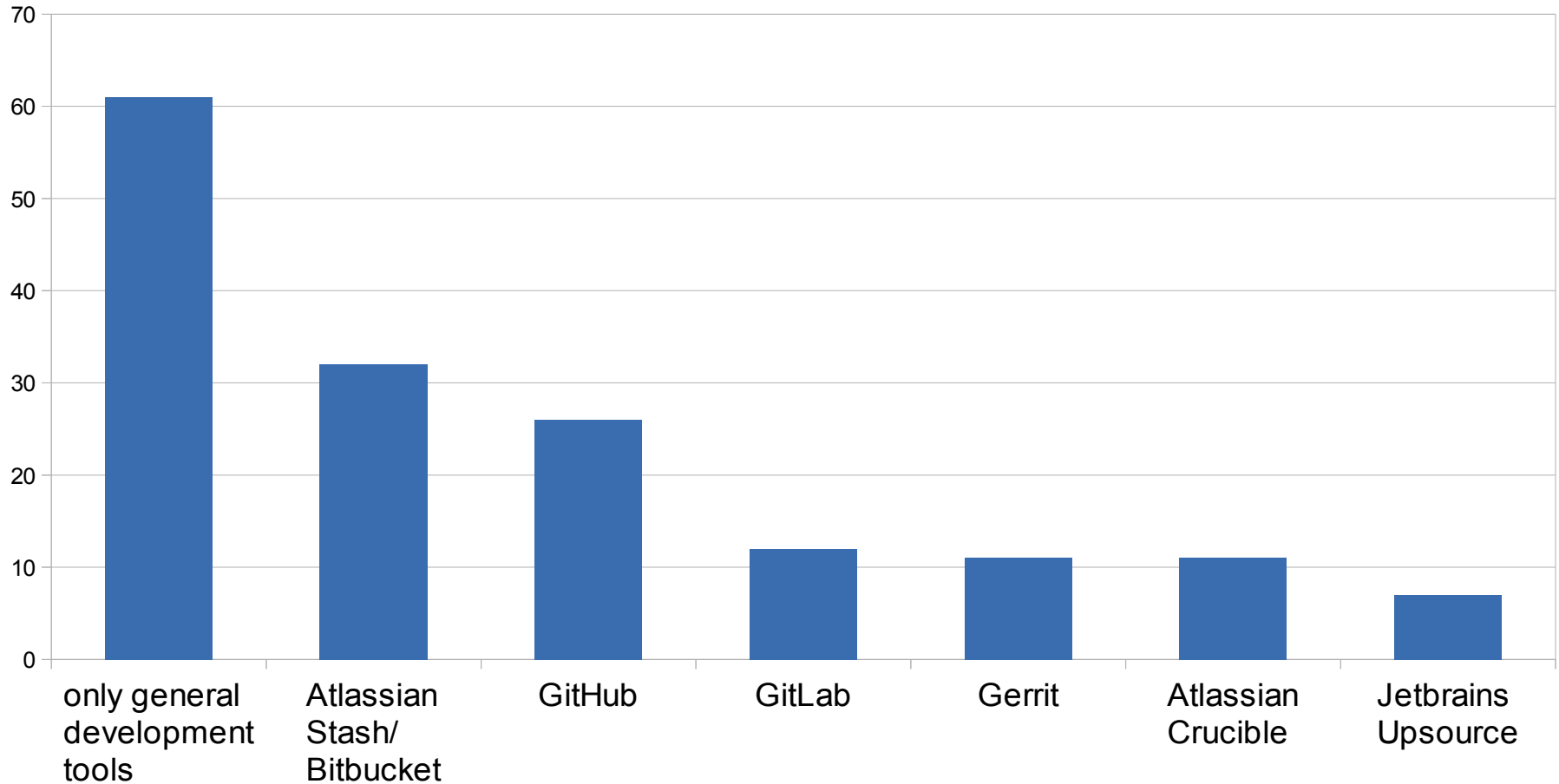
Further data 1: Change-based review prevalence

- 90% of the teams using reviews use reviews based on changes.
 - There is no statistically significant difference between agile and classic development processes
- Also checked for more specific processes from the literature:



Further data 2: Review tools

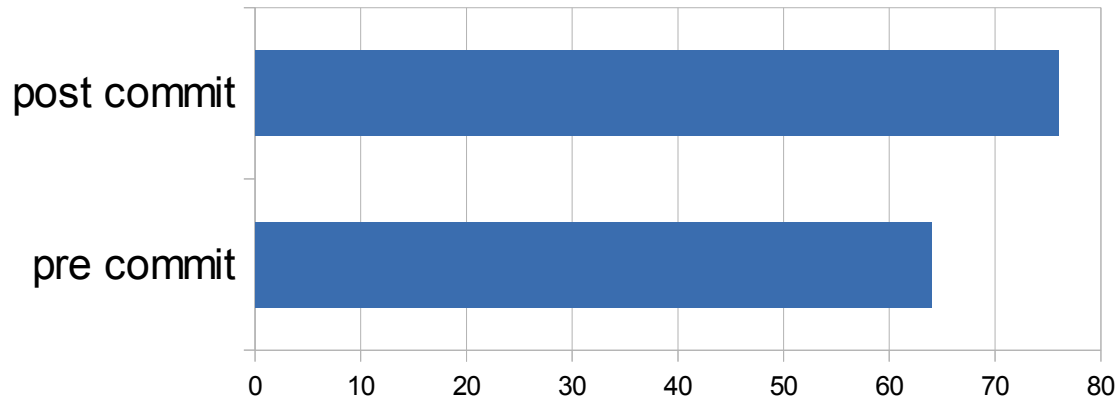
36% of respondents do not use a specialized code review tool



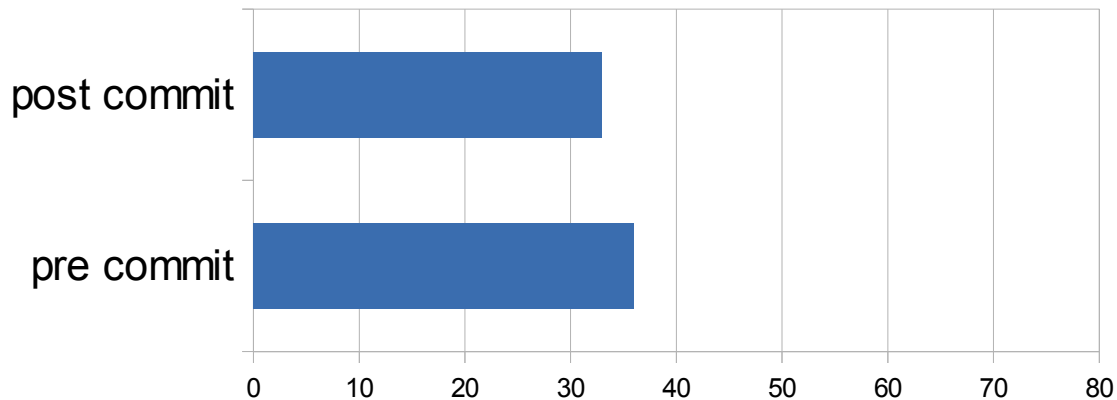
Further mentioned tools: TFS, Reviewboard, Codeflow, Collaborator, Phabricator, ReviewAssistant, proprietary tools

Further data 3: Pre/Post commit review

- A slight majority performs post commit reviews (commit then review)

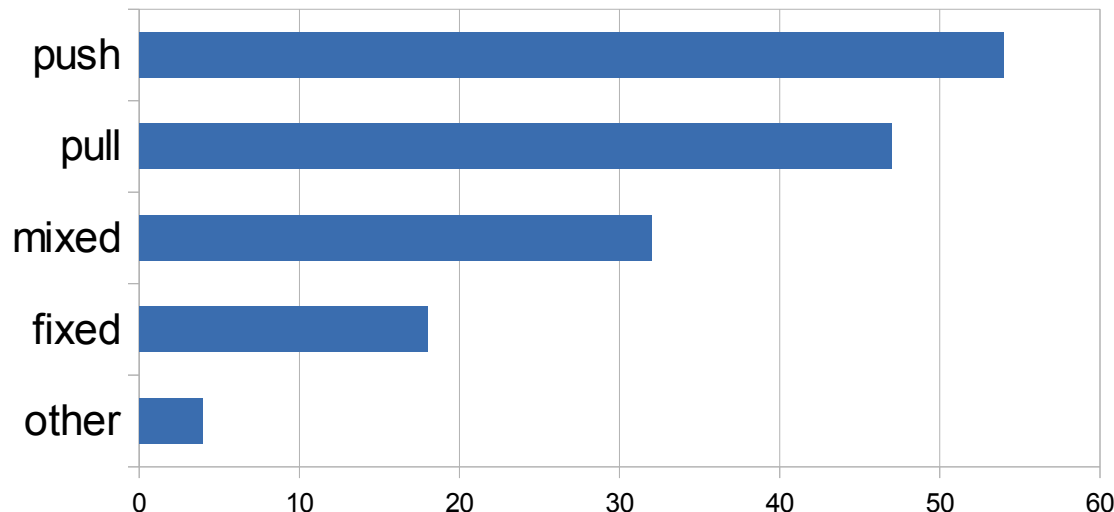


- For teams that recently introduced reviews, it's the other way around



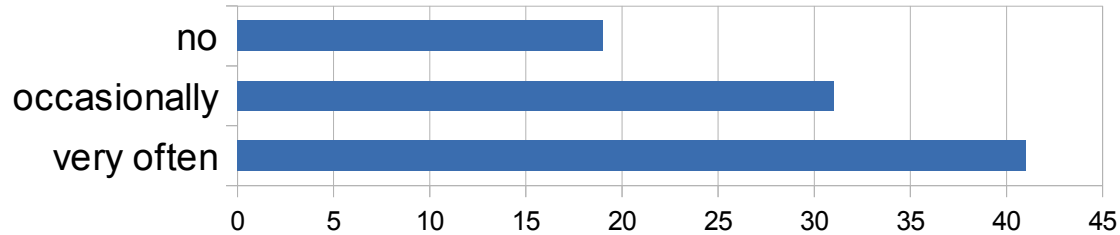
Further data 4: Pull vs Push

- There are different ways to determine who reviews what
 - Author selects a reviewer (“push“)
 - Reviewer selects a review (“pull“)
 - Author invites a group of potential reviewers (“mixed“)
 - There is a fixed assignment of reviewers, e.g. team lead or module owner
- Push is the most widespread method (contrary to common agile principles)

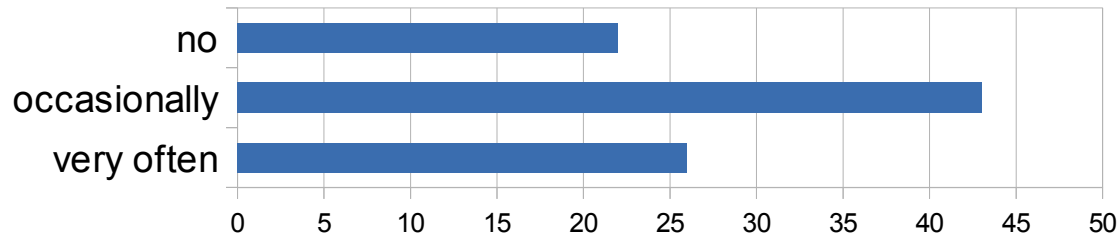


Further data 5: During the review

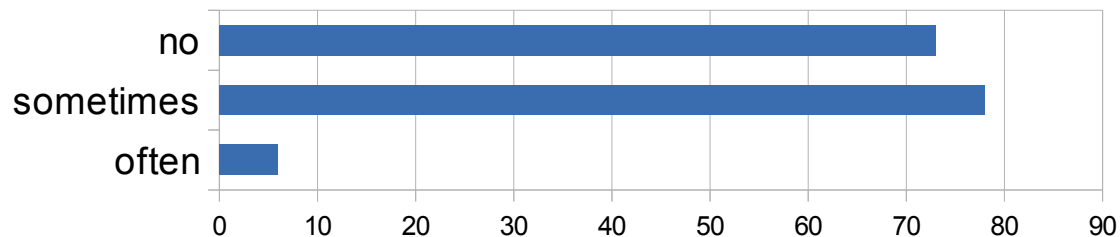
- Do you use static code analysis as support during the review?



- Do you test/execute code during the review?



- Do reviewers fix code on their own during the review?



Implications/Discussion

- Research on change-based code review is relevant :)
- Culture is an important factor for review adoption
 - Just increasing review efficiency will not necessarily increase adoption
 - Importance of company/team culture well known from management literature, but sometimes neglected in SE
 - Should we care more?
- There is a lot of variation in industrial review processes
 - How much is necessary? How much is accidental?
 - Could a “pattern language” of code review processes help?
- One size of process or tool will not fit all

Summary

- Large survey on code reviews in commercial development
 - 240 participants
 - Provides quantitative data on review process variants, team contexts, ...
- Further support that culture is an important factor for code review adoption
- Currently one publication based on the survey: *Baum, Leßmann and Schneider. “The Choice of Code Review Process: A Survey on the State of the Practice“. PROFES 2017*
- The full survey data set is available online and contains many details I had to leave out:
<https://doi.org/10.6084/m9.figshare.5104249.v1>
or follow the link on <http://tobias-baum.de>