EXPECTATIONS, OUTCOMES, AND CHALLENGES OF MODERN CODE REVIEW

Alberto Bacchelli

University of Lugano, Switzerland

Alberto.bacchelli@usi.ch

Christian Bird

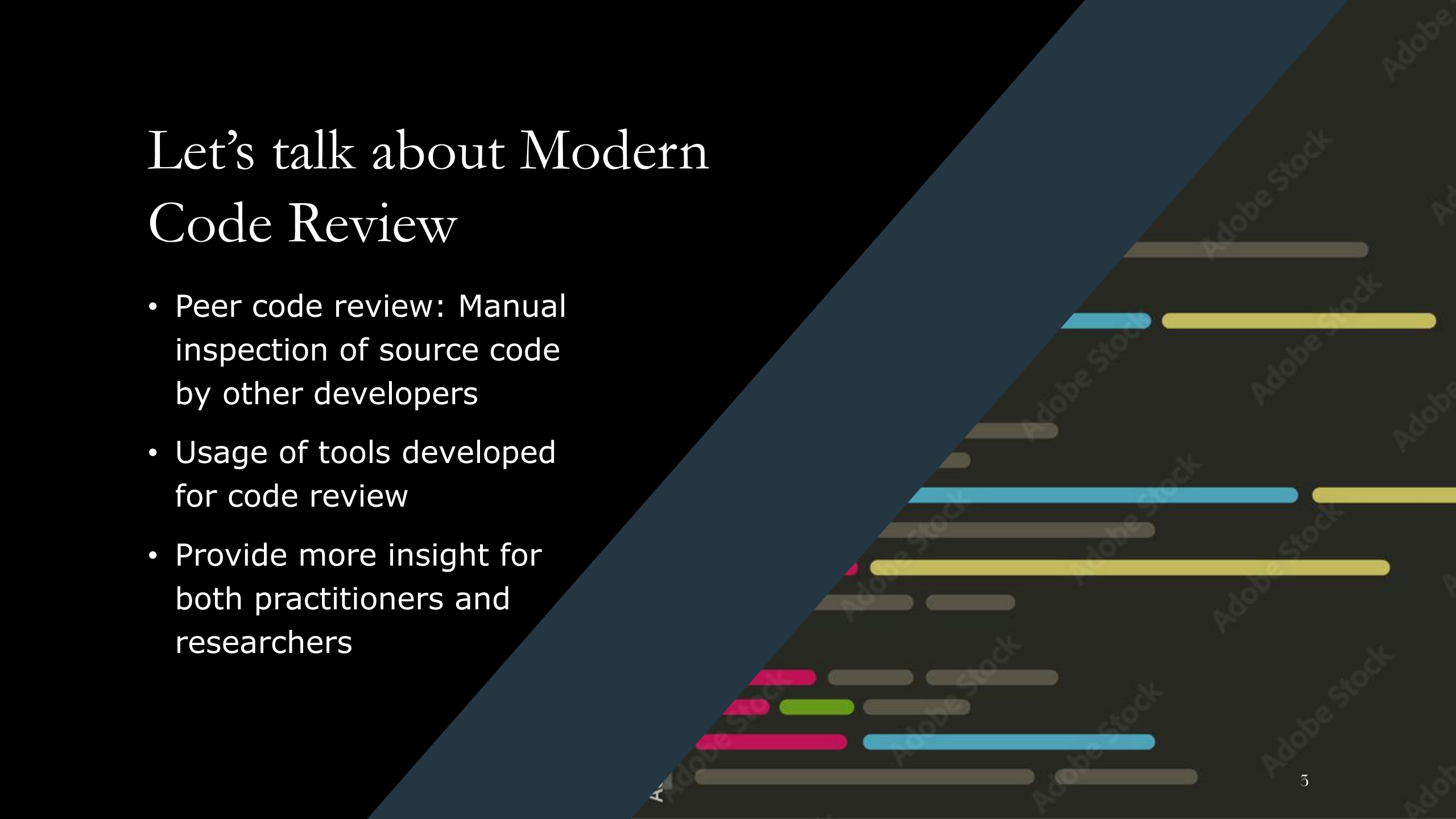
Microsoft Research

cbird@microsoft.com

THE PROBLEMATIC

Let's talk about Modern Code Review

- Peer code review: Manual inspection of source code by other developers
- Usage of tools developed for code review
- Provide more insight for both practitioners and researchers



THE PROBLEMATIC

GOAL OF THE STUDY

What's our goal?

- In-depth study of practices in teams that use modern code review.
- Microsoft as case study
- Understand their expectations, benefits and the problem they face

What do we expect to learn?

- #1: Characterize the motivations of developers & compare them with actual outcomes
- #2: Relate the outcome to understanding needs & discuss how they achieve such needs

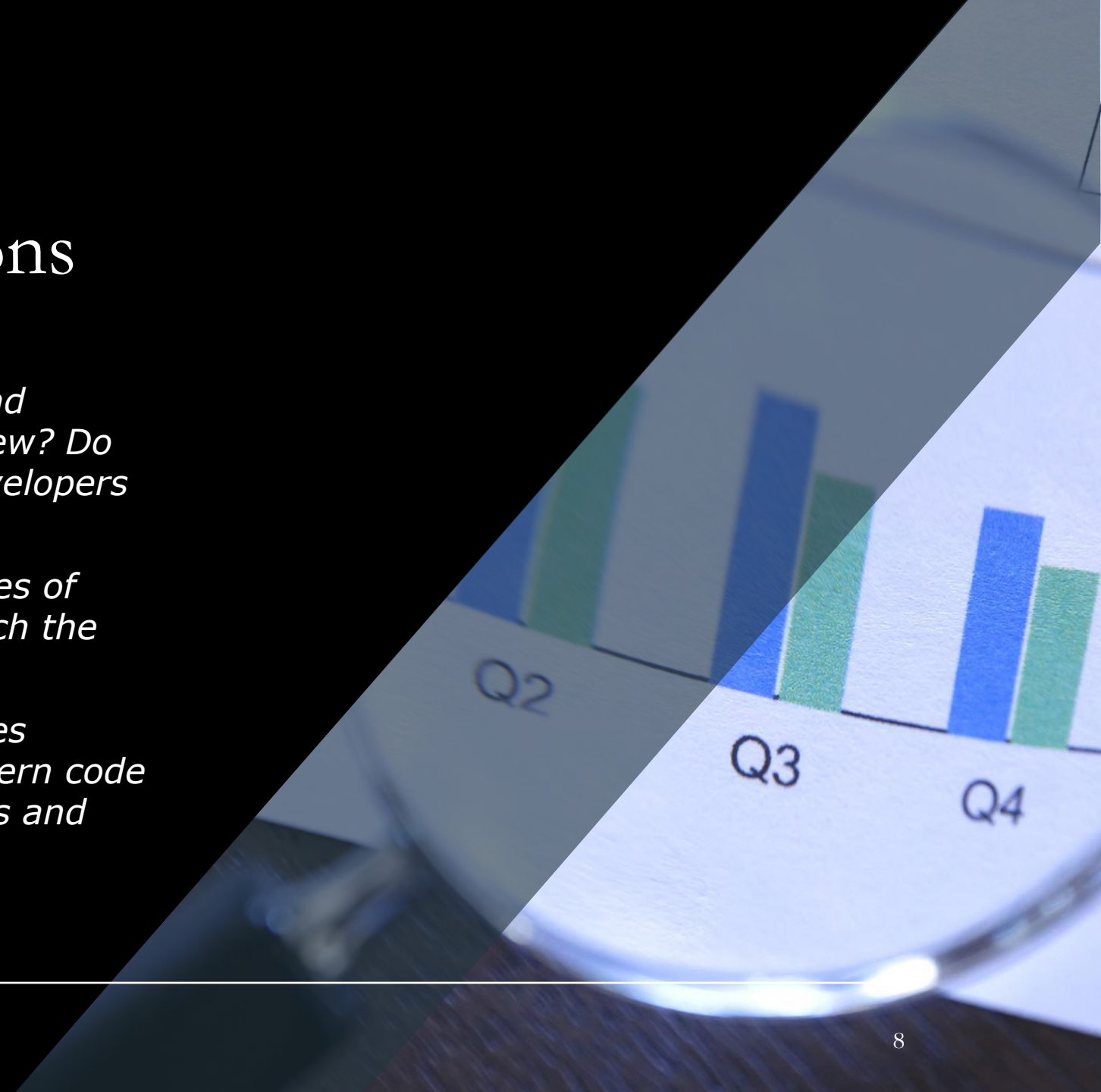
THE PROBLEMATIC

GOAL OF THE STUDY

METHODOLOGY

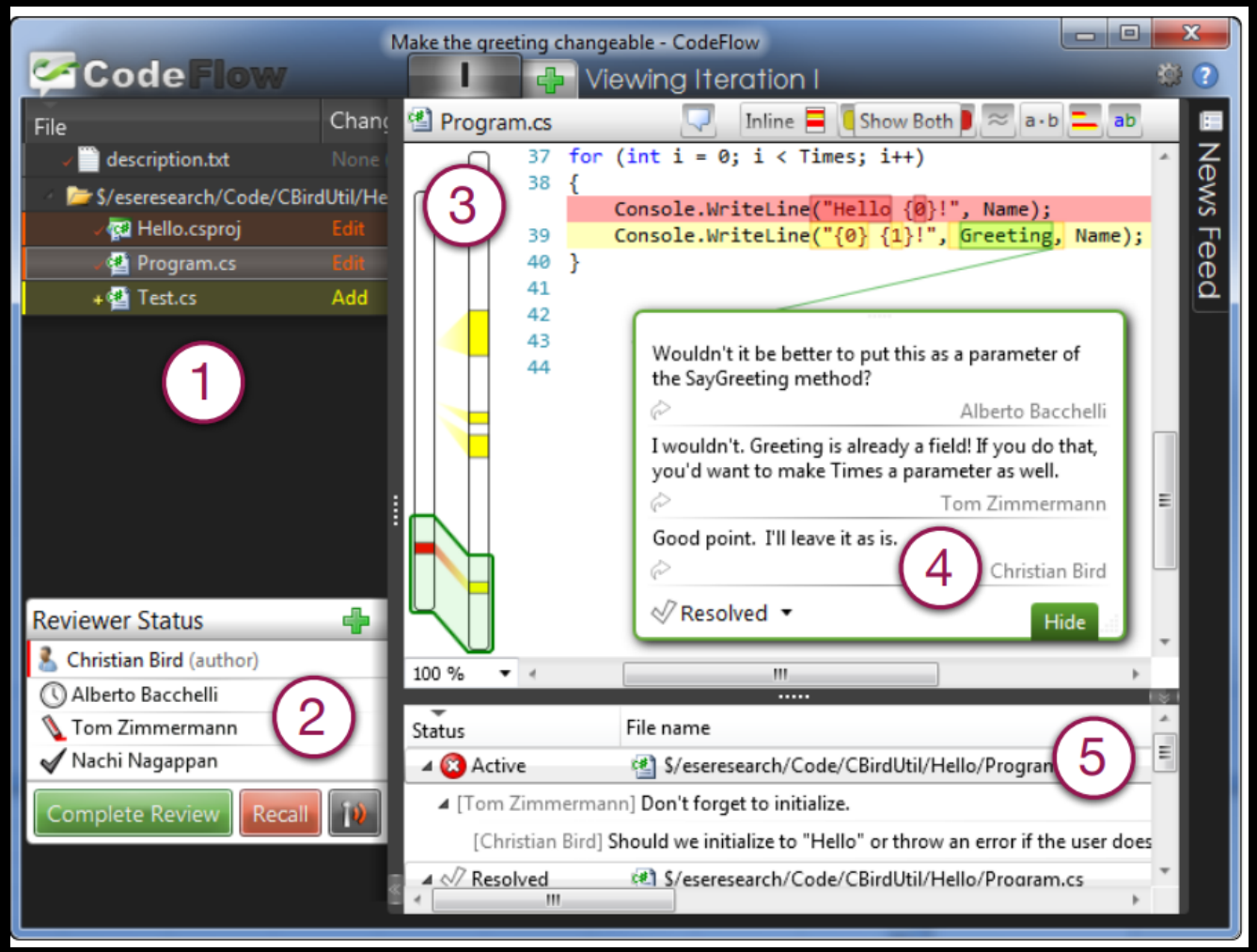
The research questions

- RQ1: *"What are the motivations and expectations for modern code review? Do they change from managers to developers and testers?"*
- RQ2: *"What are the actual outcomes of modern code review? Do they match the expectations?"*
- RQ3: *"What are the main challenges experienced when performing modern code reviews relative to the expectations and outcomes?"*



The Setting

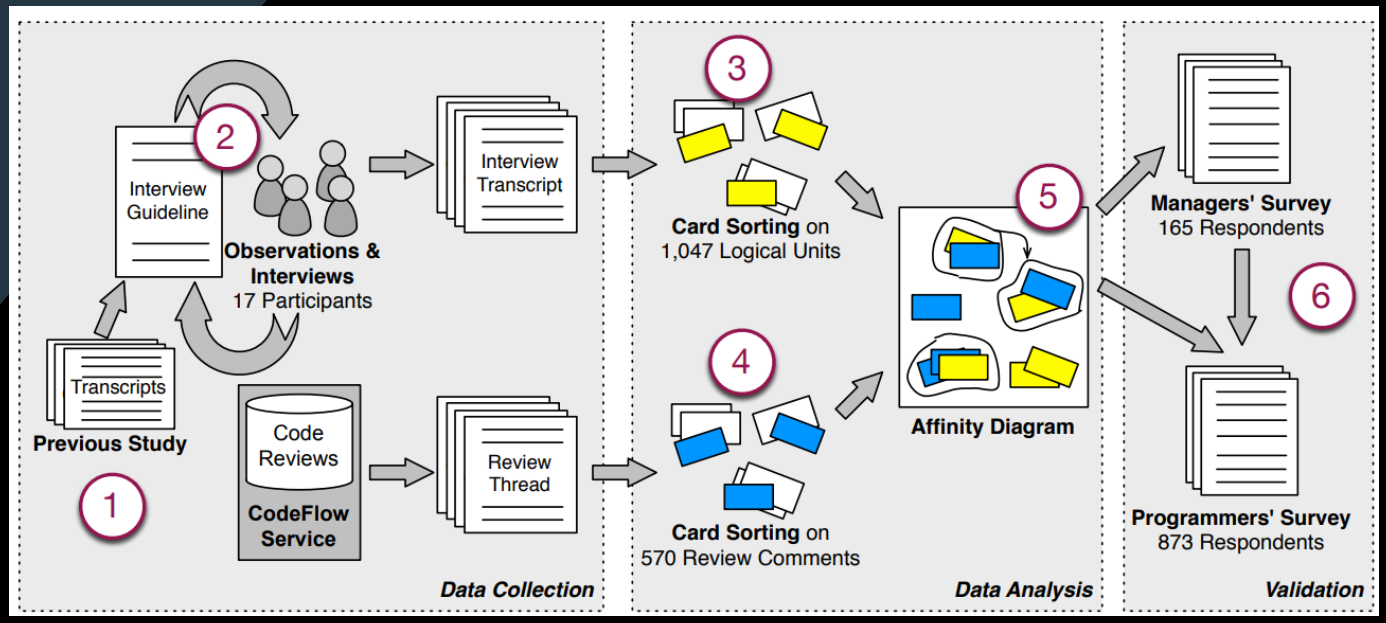
- Professional developers, testers, and managers within Microsoft
- SQL Server to mobile phone
- CodeFlow



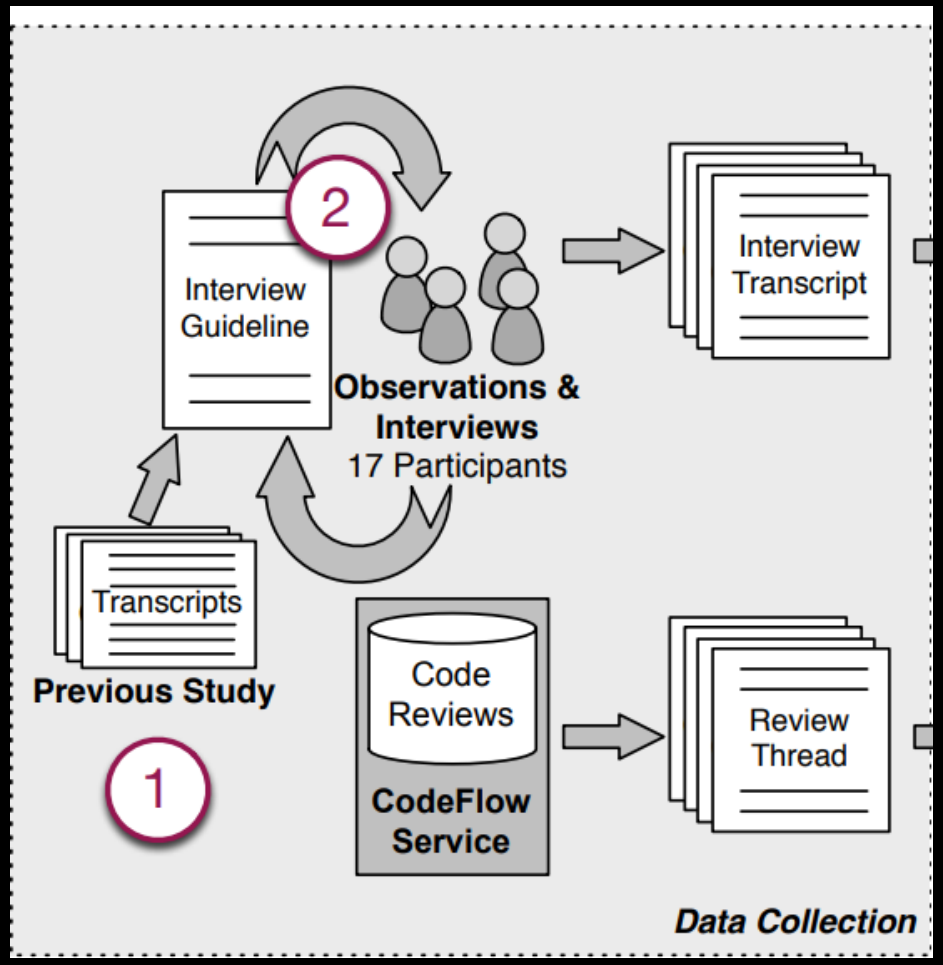
What is CodeFlow?

- Collaborative code review tool
- Create a package with changed files
- Select the reviewers
- Message to describe review
- Submit to CodeFlow

Let's talk about the real methodology



1. Analysis of previous study
2. Observations and interviews with developers
3. Cart sort (meetings)
4. Card sort (code review comments)
5. Affinity Diagram
6. Surveys



For the data collection part

1. Analysis of previous study

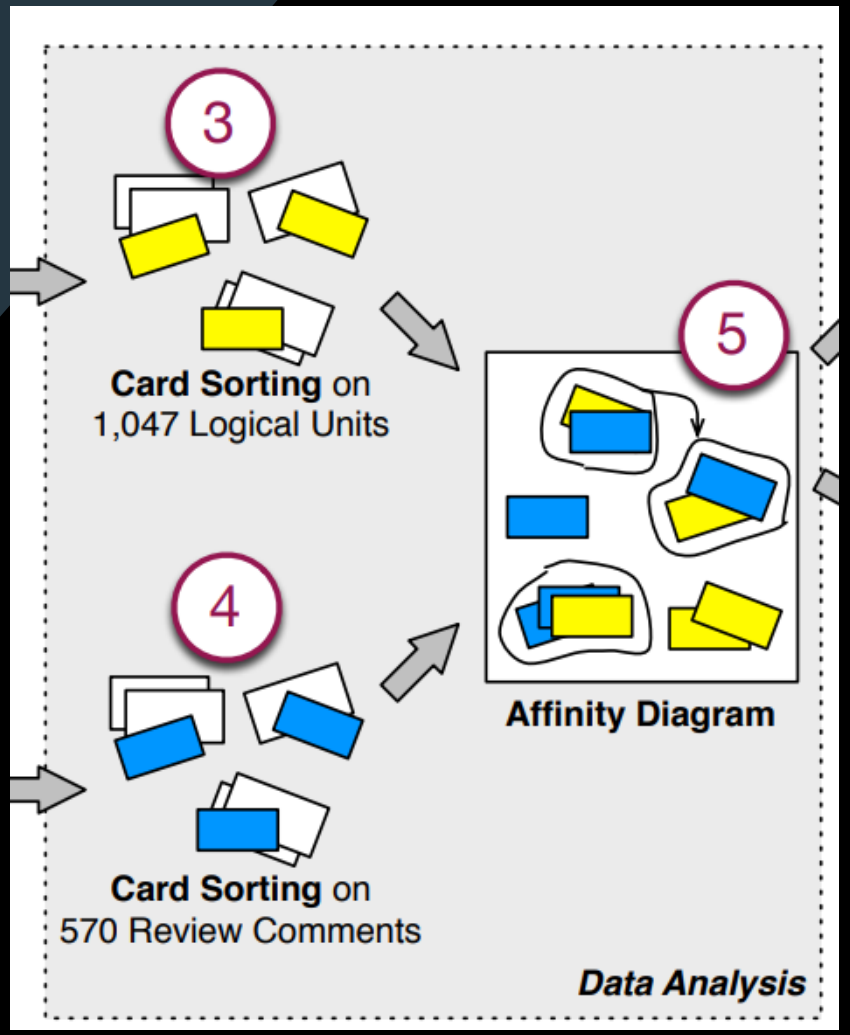
- *April and May 2012, 30 to 50 minutes*
- *4 motivations that emerged*
  - Finding defects
  - Maintaining team awareness
  - Improving code quality
  - Assessing high-level design

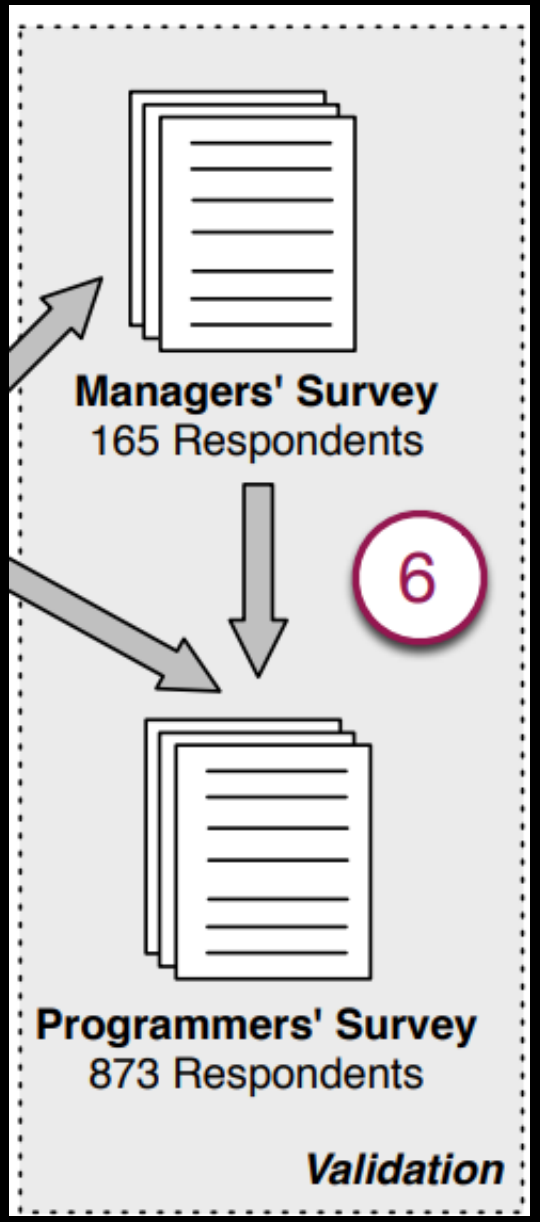
2. Observations and interviews with developers

- One-to-one meeting with developers
- 40 to 60 minutes interview, 100 candidates

For the data analysis part

3. Card sort (meetings)
   - *Organize code into hierarchies (Higher level of abstraction and common themes)*
4. Card sort (code review comments)
5. Affinity Diagram
   - *Record the categories on post-it*
   - *Spread them on walls*
   - *Sort the categories based on discussion until all are sorted and all participants agreed.*
   - *Name each group with a description*
   - *Capture and discuss the themes*





For the validation part

6. Surveys

- Validating the concepts that emerged from previous phases.
- 2 surveys
- 6 questions and sent to 600 managers
- 18 questions and sent to 2000 randomly chosen developers

THE PROBLEMATIC

GOAL OF THE STUDY

METHODOLOGY

RESULTS

Answering the first question

"What are the motivations and expectations for modern code review? Do they change from managers to developers and testers?"

1. Finding defects
   - *Top reason for 44% of the managers*
   - *First reason for 44% of developers, a second reason for 23%, and 11% for third*
2. Code Improvement
   - *First reason for 39% of developers, second for 24%, and 15% for third.*
   - *Top reason for 31% of managers*

Answering the first question

3. Alternative Solutions
   - *Improving the submitted code by adopting ideas that lead to better implementation*
   - *First reason for 17% of developers, a second reason for 23%, and 17% for third*
   - *2% of the managers only mention it*

4. Knowledge Transfer
   - *“One of the things that should be happening with code reviews over time is a distribution of knowledge.”*
   - *First reason for 8% of developers, a second reason for 14%, and 16% for third*

Answering the first question

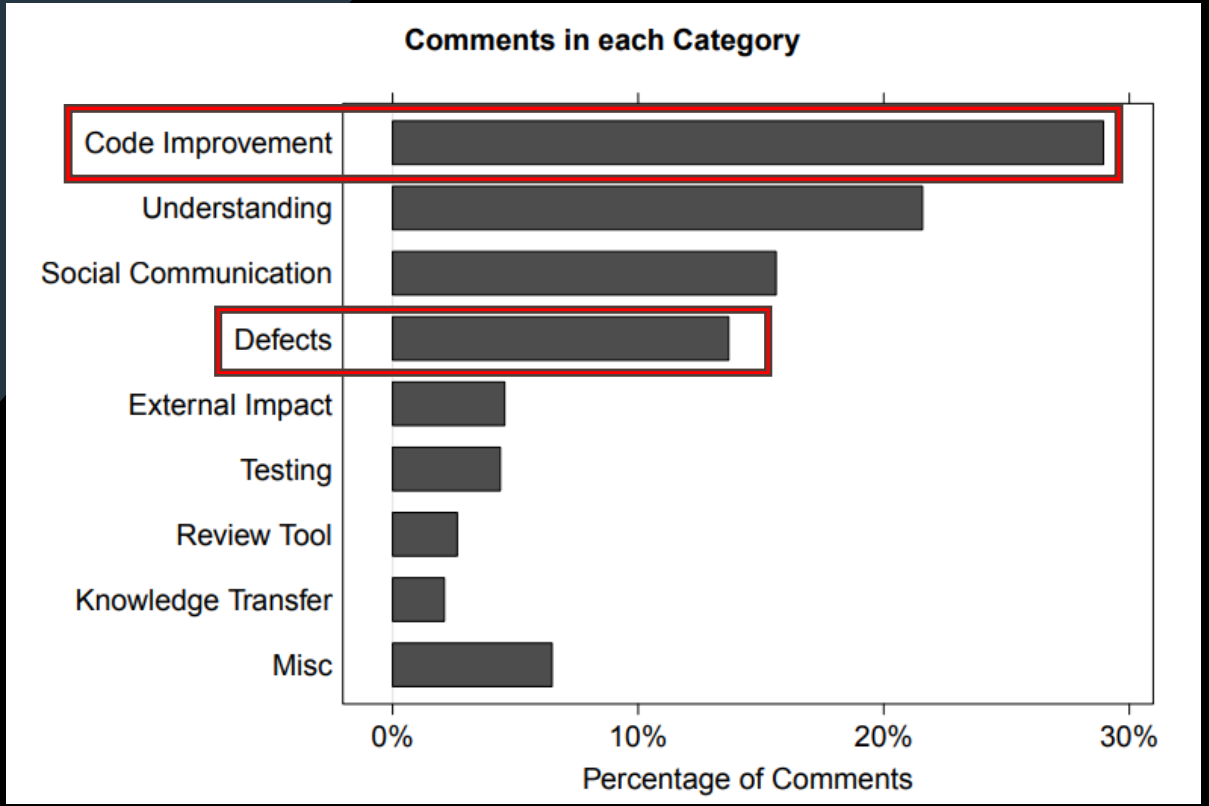
5. Team Awareness and Transparency
   - *Why changes were made*
   - *873 programmers rank it close to "knowledge transfer"*
   - *First reason for 9% of developers, a second reason for 23%, and 17% for third*

6. Share Code Ownership
   - *Mean to have more than one knowledgeable person about different parts of the code*
   - *First reason for 6% of developers, second for 11%, and 10% for third.*

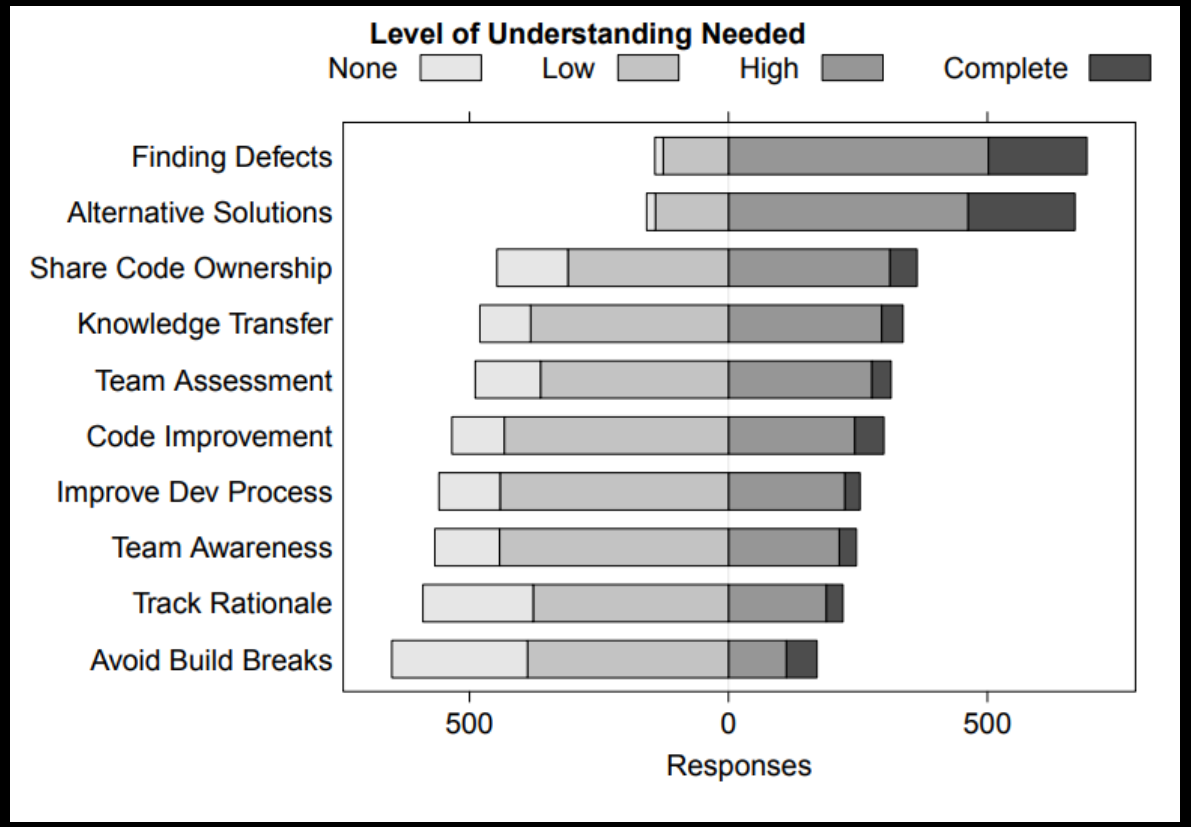
Answering the second question

"What are the actual outcomes of modern code review? Do they match the expectations?"

- 200 threads recorded by CodeFlow
- Data does not correlate with results in RQ1
- The sample too small to represent the population



Answering the third question



"What are the main challenges experienced when performing modern code reviews relative to the expectations and outcome?"

1. Understanding the code
   - *Understanding is the main challenge*
   - *Authors asked, "Does understanding need to change with the expected outcome of code review?"*
2. A Priori Understanding
   - *"Whether it takes longer to review files they are not familiar with and why" (91% positive)*
   - *"Whether reviewers are familiar with the changed files give different feedback and how" (82% positive)*
3. Dealing with Understanding needs
   - *Modern tools only show basic understandings for reviewers*

Recommendations and Limitations

Recommendations

- Quality Assurance
- Understanding
- Beyond Defects
- Communication

- Limitations
  - *Research within one company provides little value for the academic community*
  - *Representativeness of the results in other contexts*

Conclusion

- In-depth study to reveal what practitioners think.
- Wide range of motivations
- Outcomes do not always match motivations
- Recommendations to practitioners and researchers.