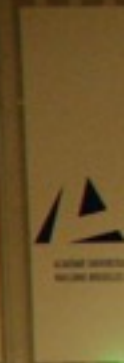


# What Do Code Reviews at Microsoft and in Open Source Projects Have in Common?





ULB  
UNIVERSITÉ LIBRE DE BRUXELLES





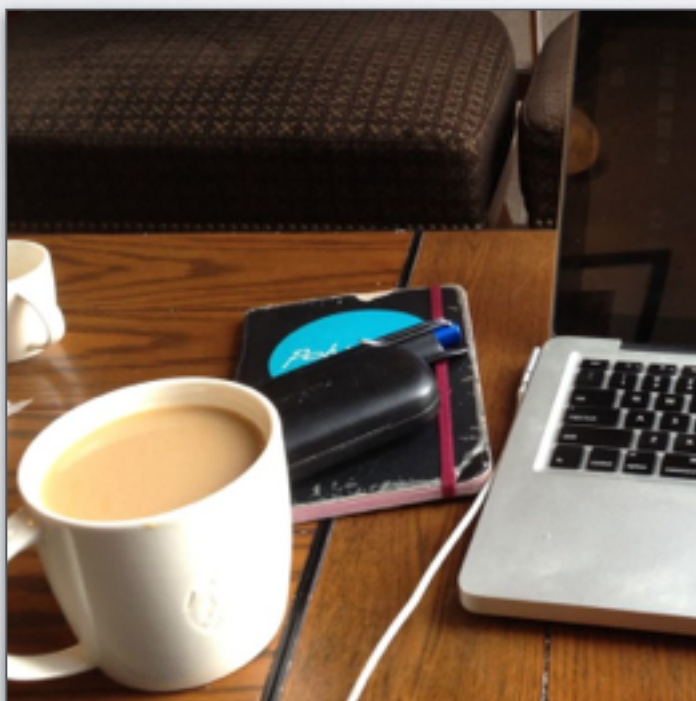
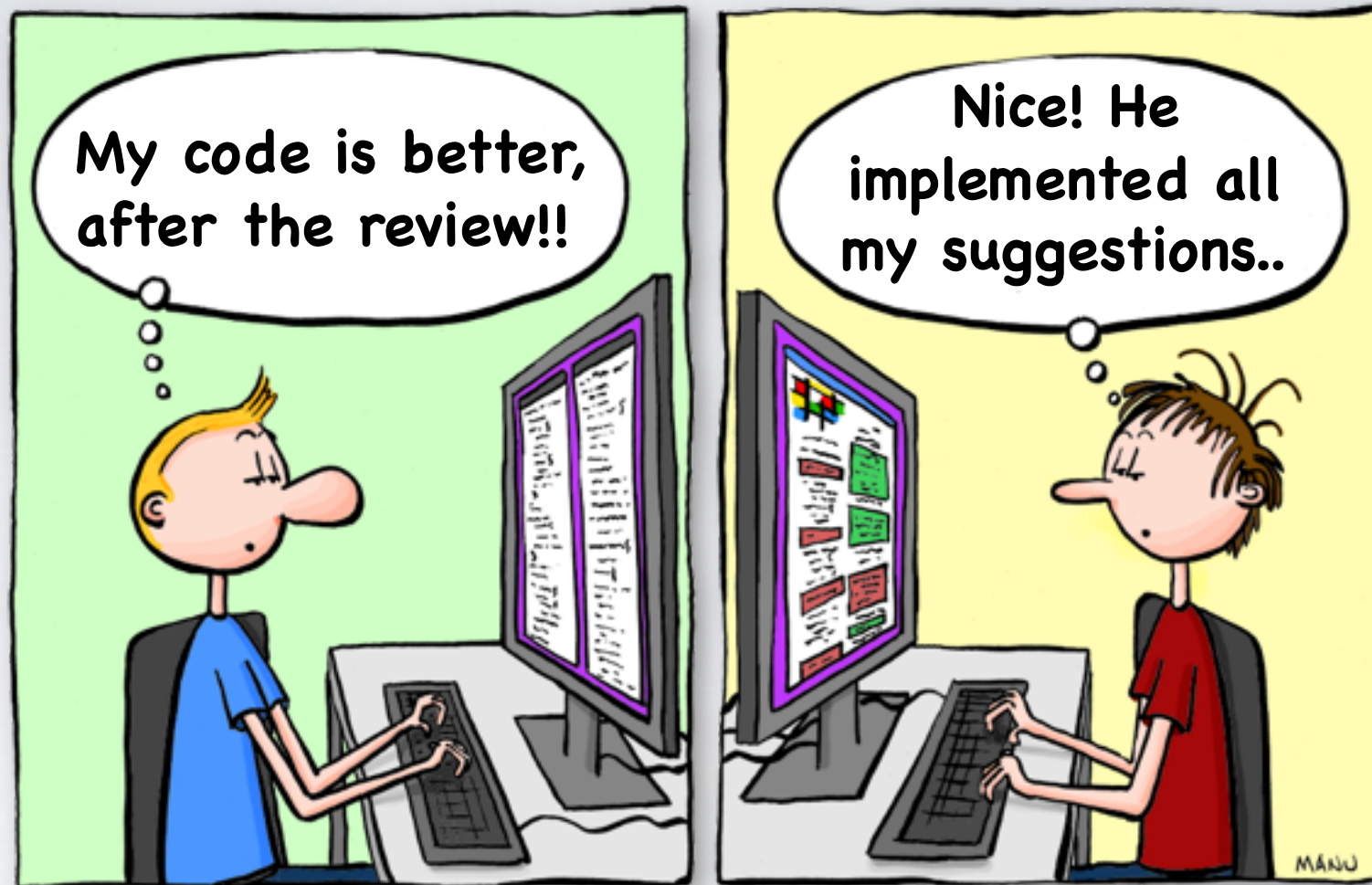




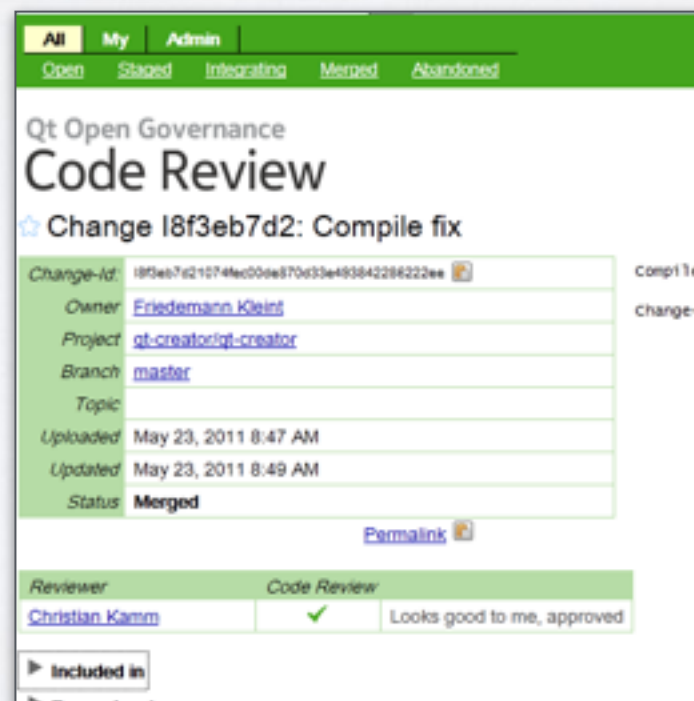




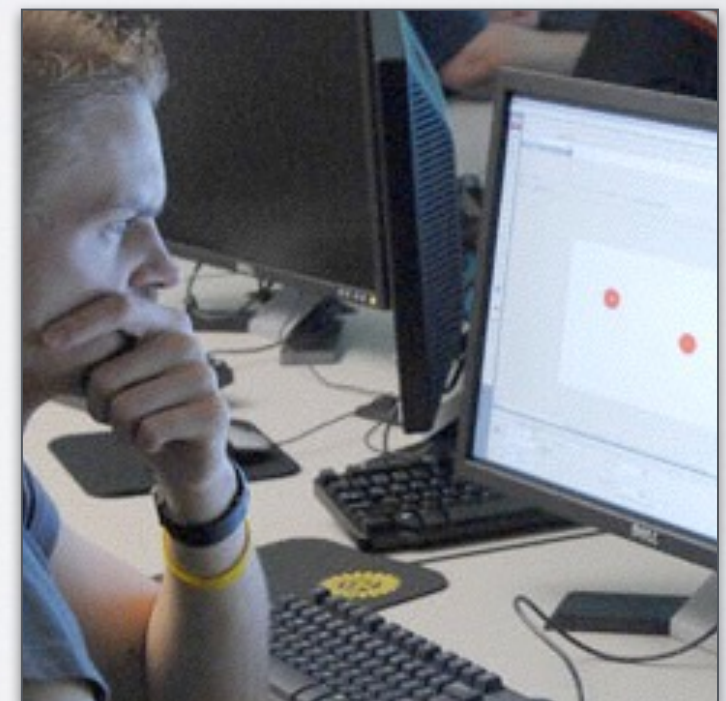
# Modern code review



informal



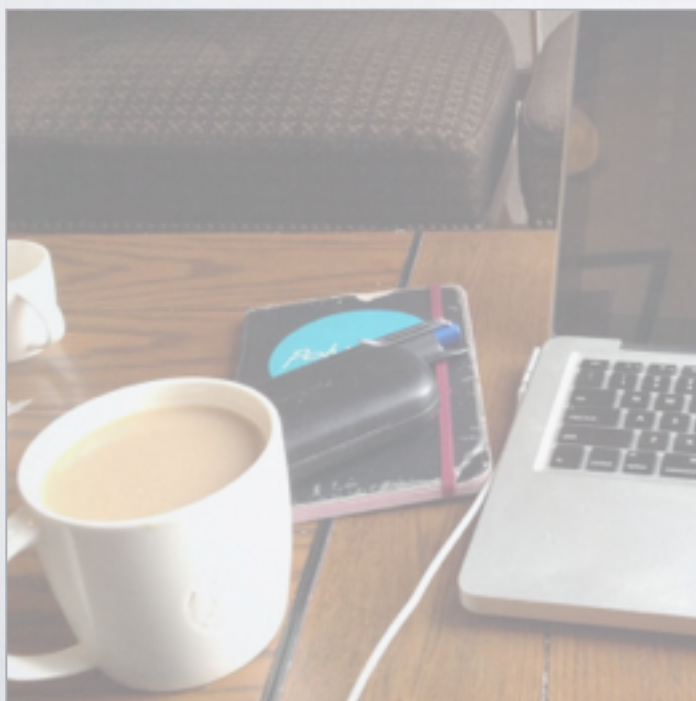
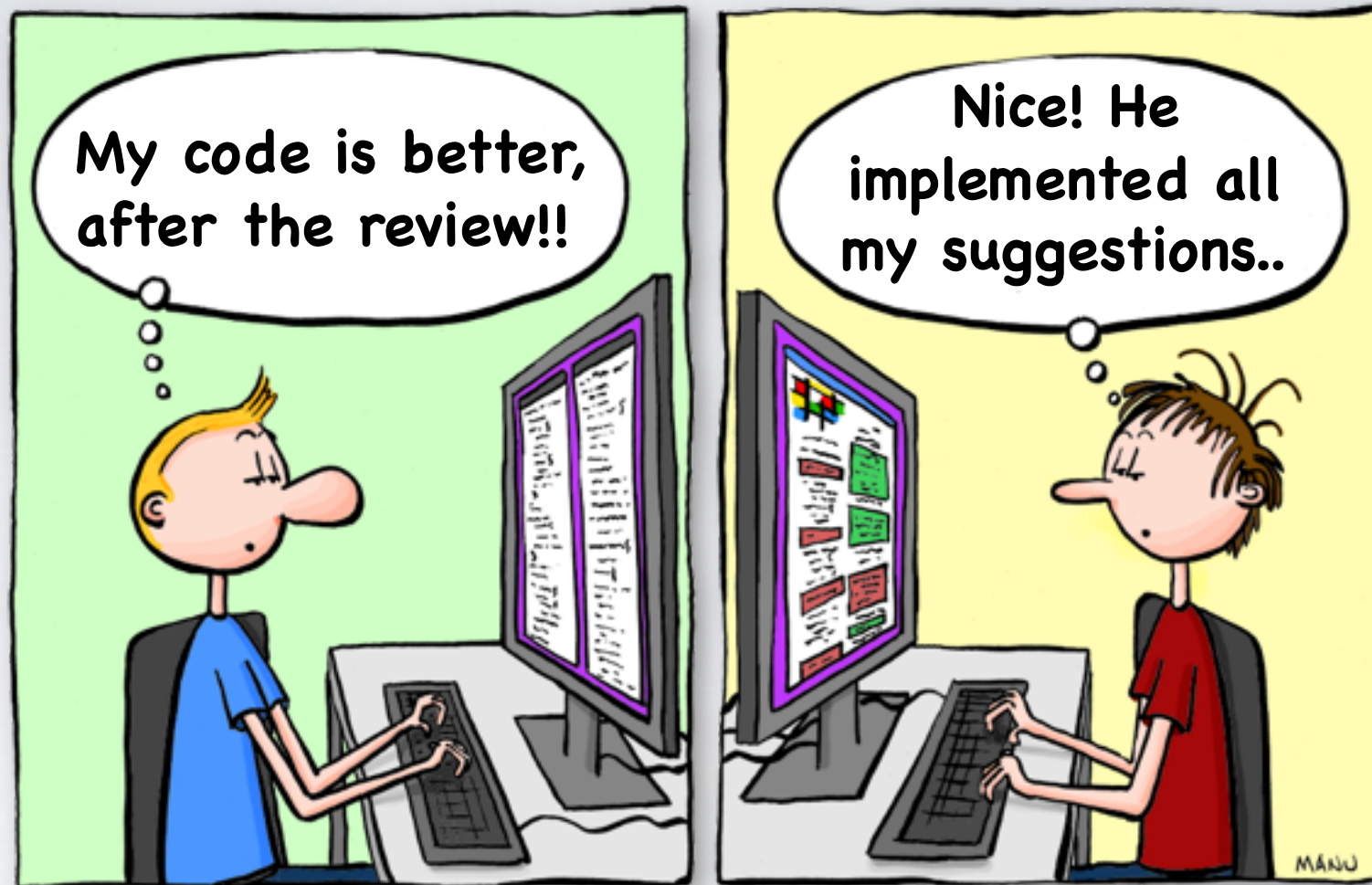
tool-based



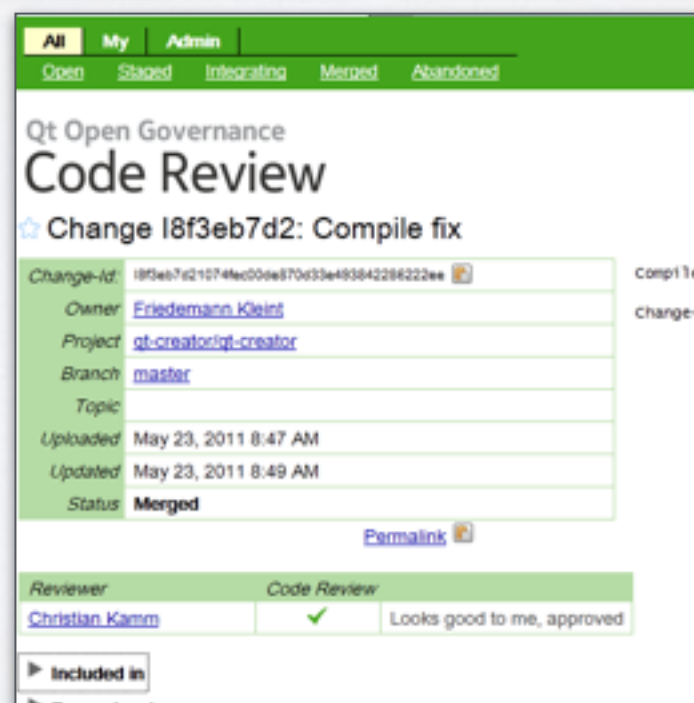
asynchronous



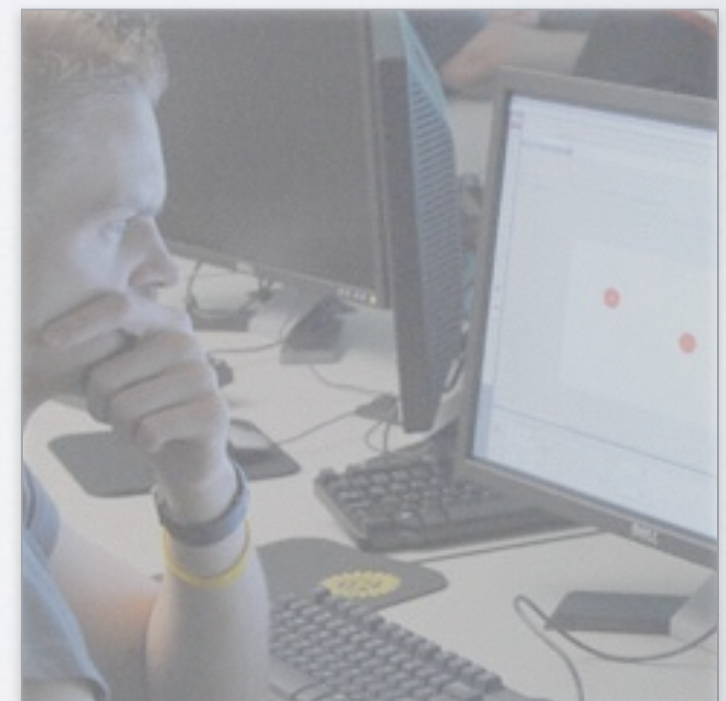
# Modern code review



informal



tool-based



asynchronous



# Code review tools: Gerrit

All

Projects

Documentation

Open

Merged

Abandoned

Change #, SHA-1, tr:Id or owner:email


Search

Sign In

android

open source project

Browse Repositories



Change-Id: I28cd0da83acc9df2162b6ec6174fc25712f68de2

Owner: jp abgrall

Project: kernel/common

Branch: android-3.4

Topic:

Uploaded: Feb 25, 2013 10:05 PM

Updated: Feb 25, 2013 10:12 PM

Status: Merged

Commit Message

Permalink

sock\_diag: Fix out-of-bounds access to sock\_diag\_handlers[]

Userland can send a netlink message requesting SOCK\_DIAG\_BY\_FAMILY with a family greater or equal then AF\_MAX -- the array size of sock\_diag\_handlers[]. The current code does not test for this condition therefore is vulnerable to an out-of-bound access opening doors for a privilege escalation.

Signed-off-by: Mathias Krause <minipli <at> googlegmail.com>

Reviewer	Code-Review	Verified
jp abgrall	✓	✓
Nick Kralevich	+1	

► Included in

► Dependencies

Old Version History: Base

▼ Patch Set 1 26cd0da83acc9df2162b6ec6174fc25712f68de2 (gitiles)

Author: Mathias Krause <minipli at> Feb 23, 2013 12:13 PM

Committer: JP Abgrall <jpa@google.com> Feb 25, 2013 9:57 PM


Parent(s): 1a6c74e0d58f6ca028f36c3d794fecff8543bbfb gpu: ion: Add support for sharing buffers with dma buf kernel handles

checkout checkout pull cherry-pick patch





# Code review tools: GitHub pull requests


## Sending a pull request #248


 Open


**cameronmcefee** wants to merge 1 commit into `octocat:master` from `cameronmcefee:master`

 Conversation 5

 Commits 1


 Files Changed 1


 4




cameronmcefee commented 2 years ago

I made some changes. Please review.


 **cameronmcefee** added a commit 2 years ago

 Made some changes for a pull request a4610fa




octocat commented 2 years ago

Awesome, thanks!




cameronmcefee commented 2 years ago


Why yes, of course.

 **cameronmcefee** closed the pull request 2 years ago





Labels

None yet

Notifications 


 Subscribe

4 participants






# Code review tools: Atlassian Crucible



Testing Project > TEST-75

## Blink Java Example Review

Author:  Edwin Dawson Created: 23 May 2010


Dashboard Source Projects People **Reviews**

☆ Edwin Dawson Search

+ Create Snippet ⚙ Tools


Click on source lines to add an inline comment.


```
01. import java.awt.*;
02. import java.util.*;
03.
04. public class Blink extends java.applet.Applet {
05.     private Timer timer;           // Schedules the blinking
06.     private String labelString;    // The label for the window
07.     private int delay;             // the delay time between blinks
08.
09.     public void init() {
10.         String blinkFrequency = getParameter("speed");
11.         delay = (blinkFrequency == null) ? 400 :
12.             (1000 / Integer.parseInt(blinkFrequency));
13.         labelString = getParameter("lbl");
14.         if (labelString == null)
15.             labelString = "Blink";
16.         Font font = new java.awt.Font("Serif", Font.PLAIN, 24);
17.         setFont(font);
18.     }
19.
20.     public void start() {
21.         timer = new Timer();        //creates a new timer to schedule the blinking
22.         timer.schedule(new TimerTask() { //creates a timertask to schedule
23.             // overrides the run method to provide functionality
24.             public void run() {
25.                 repaint();
26.             }
27.         });
28.     }
29. }
```



**Brendan Humphreys:** 19:31


consider using HTML5


[Reply](#) 



**Seb Ruiz:** 19:34


Absolutely

[Reply](#) 



**Seb Ruiz:** 19:32

This will fail when blinkFrequency is an empty string. Consider pulling this out into an if statement for greater readability.

[Reply](#) **Defect** 



# Code review tools: Microsoft CodeFlow

The screenshot displays the Microsoft CodeFlow web interface for a code review session titled "Make the greeting changeable - CodeFlow". The interface includes a sidebar on the left, a central code editor, and a bottom status panel. Numbered callouts highlight the following features:

- 1**: File list in the sidebar showing `description.txt`, `Program.cs`, `Test.cs`, and `Hello.csproj`.
- 2**: Reviewer Status section listing participants: Christian Bird (author), Alberto Bacchelli, Tom Zimmermann, and Nachi Nagappan.
- 3**: The code editor showing a C# snippet with a review comment bubble overlaid.
- 4**: A detailed review comment bubble containing a suggestion and responses from reviewers.
- 5**: The bottom status panel showing a table of review items with their status and file names.
- 6**: The top navigation bar with the "Viewing Iteration 1" tab.

**Code Snippet:**

```
37 for (int i = 0; i < Times; i++)
38 {
39     Console.WriteLine("Hello {0}!", Name);
40     Console.WriteLine("{0} {1}!", Greeting, Name);
41 }
42
43
44
```

**Review Comment:**

Wouldn't it be better to put this as a parameter of the SayGreeting method?  
— Alberto Bacchelli

I wouldn't. Greeting is already a field! If you do that, you'd want to make Times a parameter as well.  
— Tom Zimmermann

Good point. I'll leave it as is.  
— Christian Bird

Resolved

**Review Items Table:**

Status	File name
Active	<code>\$/esereseach/Code/CBirdUtil/Hello/Program.cs</code>
[Tom Zimmermann] Don't forget to initialize.	
[Christian Bird] Should we initialize to "Hello" or throw an error if the user does	
Resolved	<code>\$/esereseach/Code/CBirdUtil/Hello/Program.cs</code>




**Code review, an interactive online survey!**

**<http://sback.it/fosdem.html>**

**<http://goo.gl/forms/hknZvi1YUo>**



# Why research on modern code review?



**THE ORIGINAL CURE ALL**

**RELIEVES INSTANTANEOUSLY**  
And Cures: Headaches,  
Nuralgia, Cough, Cold,  
Sneezing, Hiccups,  
Goat, Gonorrhea, Dyptheria,  
Dampiang, Mumps,  
Measles, Whooping cough,  
Tuberculosis, And even  
Bowden's Malady.

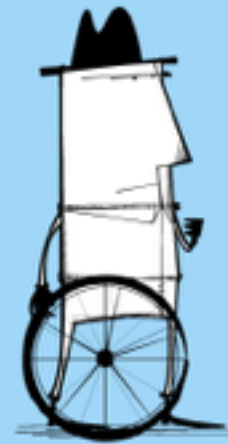
**MITCHELL'S**  
Providing the Finest in do-it-yourself health care since 2366

**FOR BLINDNESS TRY OUR RATTLESNAKE OIL!**

**MITCHELL'S SNAKE OIL 101 PROOF CURE ALL LINAMENT**

The advertisement features a central illustration of a bottle of Mitchell's Snake Oil Liniment. Above the bottle is a circular portrait of a man in a top hat holding a cane and a bottle. The text is arranged in a decorative, vintage style with ornate borders and a list of ailments cured by the product. At the bottom, it encourages trying the product for blindness.

ERRR...



CAN'T STOP.  
TOO BUSY!!





**“It engages developers and others in a formal process [...] that usually detects more defects in the product than does machine testing.”**

Advances in Software Inspections

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, V.

MICHAEL E. FAGAN, MEMBER, IEEE

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. SE-12, NO. 7,  
**Software Inspections**  
 FAGAN, MEMBER, IEEE

Software Ins  
MICHAEL E. FAGAN, MEMBER, IEEE

**Abstract**—This paper presents new studies and experiences that enhance the use of the inspection process and improve its contribution to development of defect-free software on time and at lower costs. Examples of benefits are cited followed by descriptions of the process and some methods of obtaining the enhanced results.

Software inspection is a method of static testing to verify that formal process of investigations. It engages the developers and product—and at lower cost—than does machine detection. Inspection method report very significant improvement in software development efforts. Excellent results have been achieved in all organizations in all phases of development. There is a need for inspection work.

... presents new studies and experiences that enable the methods of obtaining the enhanced results. Existing methods are cited followed by descriptions of the process and formal process of investigation. It engages the developers and others in a product—and at lower cost—than does machine testing. Users of the method report very significant improvements in quality that are accompanied by lower development costs and greatly reduced maintenance efforts. Excellent results have been obtained by small and large organizations in all aspects of new development as well as in maintenance. There is some evidence that developers who participate in the inspection of their own product actually create fewer defects in future work. Because inspections formalize the development process, productivity and quality enhancing tools can be adopted more easily and rapidly.

**Index Terms**—Defect detection, inspection, project, software quality assurance, software development, software project management, software quality, testing, walkthrough.

**BENEFITS:**

THE software...  
in IBM...  
In M...

Index Terms—Defect detection, inspection, project management, quality assurance, software development, software engineering, software quality, testing, walkthrough.

process to documentation, and documentation, in each instance, being found to improve product effectiveness, i.e., it saved more than it cost. The effectiveness of inspections are improved by applying them in many new and different ways to improve software quality and reduce costs.

**BENEFITS: DEFECT REDUCTION, DEFECT PREVENTION, AND COST IMPROVEMENT**

In March 1984, while addressing the annual meeting of the Software Engineering Society of America, the author reported on software service companies that had adopted the M Program for software quality control.

...REDUCTION, DEFECT PREVENTION,  
AND COST IMPROVEMENT  
...on software service, L. H. Fent  
...M Programming Systems, m  
...on quality improvement  
...Our goal is a  
...ect in

...addressing the IBM SHARE User  
...L. H. Fenton, IBM Director  
...Systems, made an important state-  
...quality improvement due to inspections [1].  
"Our goal is to provide defect free prod-  
...product information, and we believe  
...do this is by refining and enhanc-  
...software development process.  
...Since we introduced the  
...74, we have  
...quality

Since we have achieved the inspection process by refining and enhancing our existing code shipped for System/360 lines of code, while the back

IBM has nearly doubled the number of lines of code shipped for System/370 software products since 1976, while the number of defects per thousand lines of code has been reduced by two-thirds. Feedback from early MVS/XA and VM/SP Release 3 users indicates these products met and, in cases, exceeded our ever increasing quality expectations."

Observation of a small sample indicated that early experience injected in the products met and exceeded our ever increasing quality requirements."

INTRODUCTION  
Section pro  
NY

**INTRODUCTION**

THE software inspection process was created in 1972, in IBM Kingston, NY, for the dual purposes of improving software quality and increasing programmer productivity. Its accelerating rate of adoption throughout software development and maintenance industry is an acknowledgment of its effectiveness in meeting its goals. Outlined in this paper are some enhancements to the inspection process, and the experiences of some of the many companies and organizations that have contributed to its evolution. The author is indebted to and thanks the many people who have given their help so liberally.

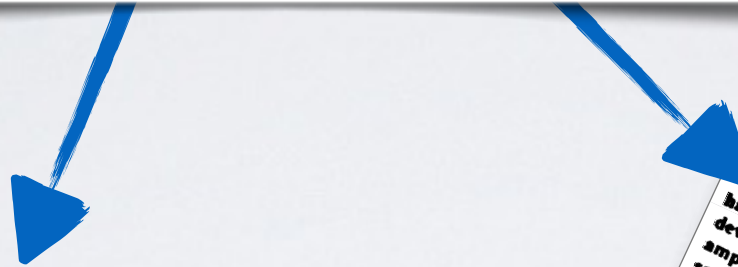
Because of the clear structure the inspection process has brought to the development process, it has enabled study of both itself and the conduct of development. The latter has enabled process control to be applied—a much earlier point in the requirements are inspected—from the point at which the requirements are inspected—a much earlier point in the process than ever before—and throughout development. Inspections provide data on the performance of individual developers, thus making it possible to foster improvement of its performance. At the same time, studies of inspection have injected a very high defect detection rate into the software development process, thus making it possible to be conducted routinely.



# Code inspections

# Can you trust these results apply to modern code review?

**“It engages developers and others in a formal process [...] that usually detects more defects in the product than does machine testing.”**



# Advances in Software Inspections

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, V.

MICHAEL E. FAGAN, MEMBER, IEEE

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. SE-12, NO. 7,  
**Software Inspections**  
 FAGAN, MEMBER, IEEE

Software Ins  
MICHAEL E. FAGAN, MEMBER, IEEE

**Abstract**—This paper presents new studies and experiences that enhance the use of the inspection process and improve its contribution to development of defect-free software on time and at lower costs. Examples of benefits are cited followed by descriptions of the process and some methods of obtaining the enhanced results.

Software inspection is a method of static testing to verify the formal process of investments. It engages the developers and meets its requirements. It engages the developers and meets its requirements. It engages the developers and meets its requirements.

Product—and at lower cost—than does static testing to verify the formal process of investments. It engages the developers and meets its requirements. It engages the developers and meets its requirements.

Method report very significant improvement in the software development process. It engages the developers and meets its requirements. It engages the developers and meets its requirements.

Companion efforts by lower development organizations in all phases of the software development process. It engages the developers and meets its requirements. It engages the developers and meets its requirements.

Inspection work.

**Index Terms**—Defect detection, inspection, project quality assurance, software development, software quality, testing, walkthrough.

development and design and code inspections prompted the adaptation of the principles of the inspection process to inspections of requirements, user information, and documentation, and test plans and test cases. In each instance, the new uses of inspection were found to improve product quality and to be cost effective, i.e., it saved more than it cost. Thus, as the effectiveness of inspections are improving, they are being applied in many new and different ways to improve software quality and reduce costs.

**BENEFITS: DEFECT REDUCTION, DEFECT PREVENTION, AND COST IMPROVEMENT**

In March 1984, while addressing the Software Engineering Group on software services, the Director of the Software Engineering Group, Mr. J. M. Program, stated:

on quality

...found to improve p...  
...effectiveness of inspections are imp...  
...being applied in many new and different wa...  
...improve software quality and reduce costs.

**BENEFITS: DEFECT REDUCTION, DEFECT PREVENTION,  
AND COST IMPROVEMENT**

In March 1984, while addressing the IBM S...  
...up on software service, L. H. Fent...  
...M Programming Systems, m...  
...on quality improvement.

Our goal is a...  
...ct inf...

DEFECTION REDUCTION, DEFECT PREVENTION, AND COST IMPROVEMENT

In March 1984, while addressing the IBM SHARE User Group on software service, L. H. Fenton, IBM Director of VM Programming Systems, made an important statement on quality improvement due to inspections [1]:

"Our goal is to provide defect free product information, and we believe this is by refining and enhancing software development and testing."

Since we introduced the IBM VM Development and Test Environment, we have

"Our goal is to provide defect free products and product information, and we believe the best way to do this is by refining and enhancing our existing software development process.

Since we introduced the inspection process in 1974, we have achieved significant improvements in quality. IBM has nearly doubled the number of code shipped for Systems Division products in the lines of 1976, while the number of defects per

Since we have achieved the inspection process in 1974, we have nearly doubled the number of lines of code shipped for System/370 software products since 1976, while the number of defects per thousand lines of code has been reduced by two-thirds. Feedback from early MVS/XA and VM/SP Release 3 users indicates these products met and, in many cases, exceeded our ever increasing quality expectations."

Observation of a small sample of these products met and, in many cases, exceeded our ever increasing quality requirements. The number of defects per thousand lines of software products has been reduced by two-thirds. MVS/XA and VM/SP Release 2 products are the best quality products we have ever produced. The early experience injected into the development of the new products has been a major factor in the success of the new products. The quality of the new products is a direct result of the early experience injected into the development of the new products. The quality of the new products is a direct result of the early experience injected into the development of the new products.

**INTRODUCTION**

THE software inspection process was created in 1972, in IBM Kingston, NY, for the dual purposes of improving software quality and increasing programmer productivity. Its accelerating rate of adoption throughout software development and maintenance industry is an acknowledgment of its effectiveness in meeting its goals. Outlined in this paper are some enhancements to the inspection process, and the experiences of some of the many companies and organizations that have contributed to its evolution. The author is indebted to and thanks the many people who have given their help so liberally.

Because of the clear structure the inspection process has brought to the development process, it has enabled study of both itself and the conduct of development. The latter has enabled process control to be applied—a much earlier point in the requirements are inspected—from the point at which the requirements are inspected—a much earlier point in the process than ever before—and throughout development. Inspections provide data on the performance of individual developers, thus providing a unique opportunity to evaluate new tools and techniques. At the same time, studies of inspection have fostered improvement of its techniques, thus resulting in very high defect detection rates. This paper describes how the inspection process can be conducted routinely.

**MADRID**



# Modern Code Review @ Microsoft



**Dr. Christian Bird**



# Modern Code Review @ Microsoft





# Modern Code Review @ Microsoft



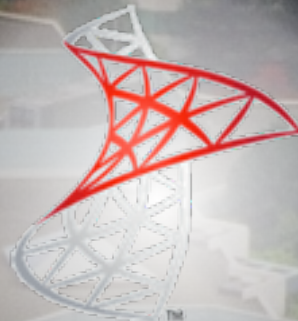
Excel



Windows  
Phone



XBox



SQL Server

...



# The CodeFlow review tool

Used across all Microsoft product teams  
by more than 70,000 developers, so far.



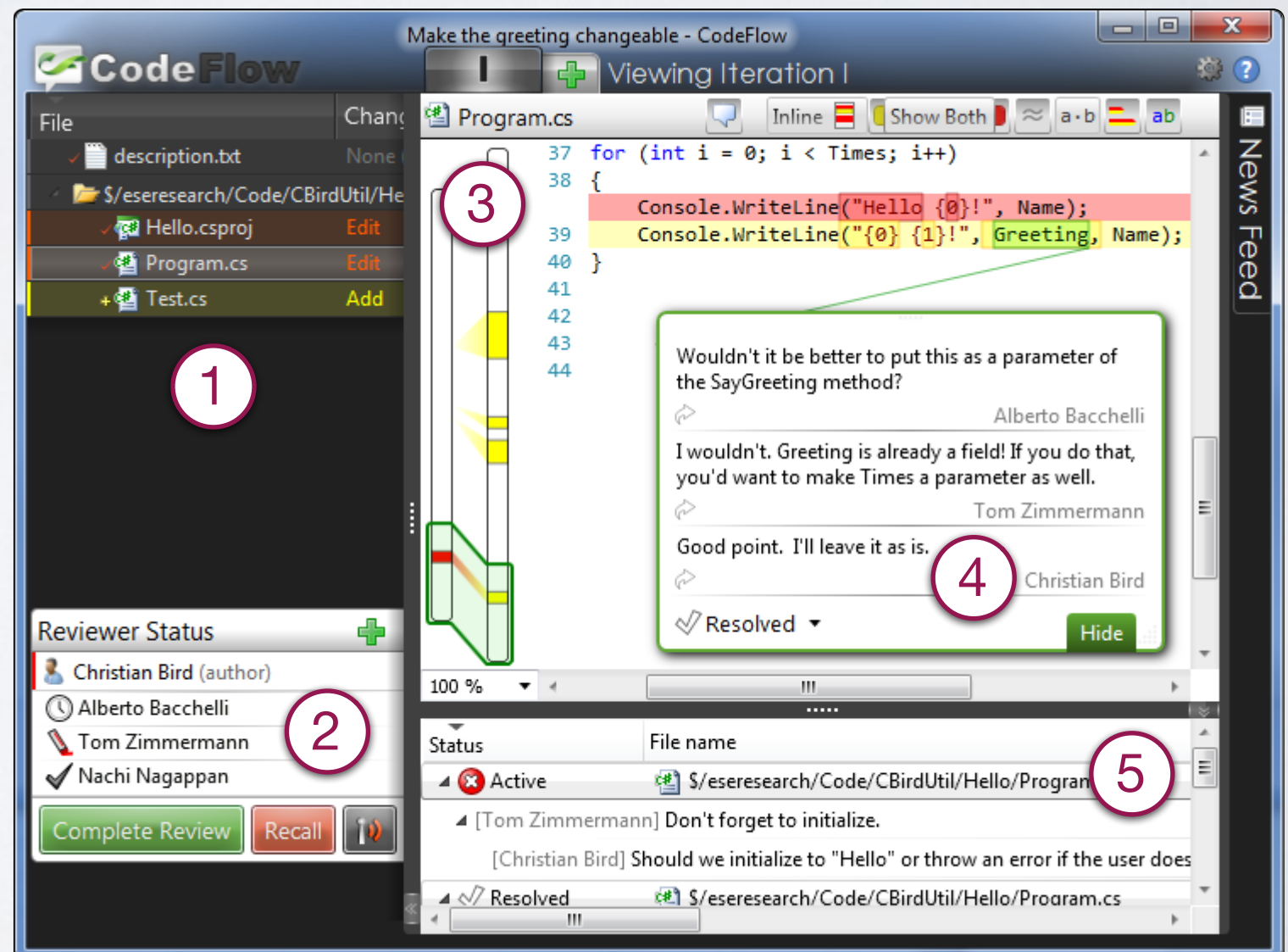
Excel



XBox

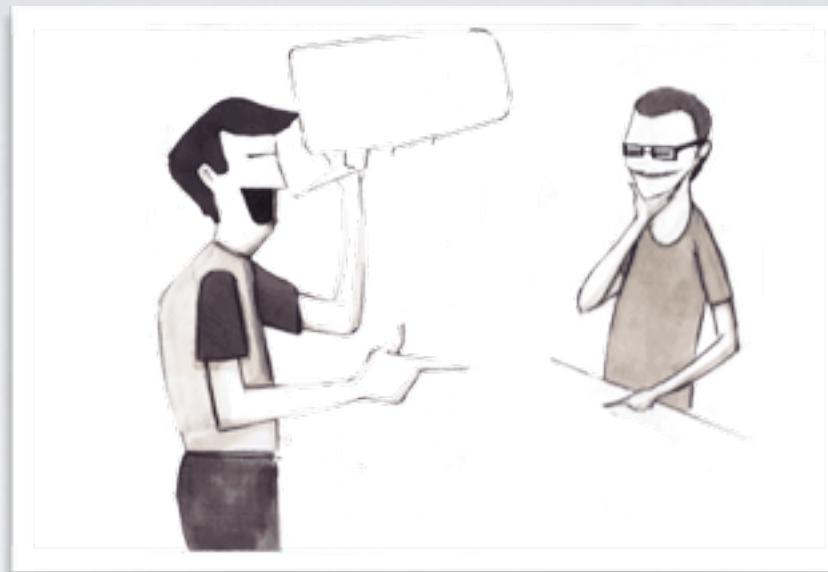


SQL Server





**observations**



**interviews**



**18 interviews with observations  
~40 minutes long  
developers, testers  
different roles  
signed off at least 50 reviews**





Shuttle Connect

Microsoft

363

Shuttle  
Connect

363





**observations**



**interviews**



**survey to 165 managers**



# List of motivations for doing code review

**Alternative  
Solutions**

**Avoid Build  
Breaks**

**Code  
Improvement**

**Team  
Assessment**

**Share Code  
Ownership**

**Team  
Awareness**

**Knowledge  
Transfer**

**Improve  
Dev. Process**

**Track  
Rationale**

**Finding  
Defects**

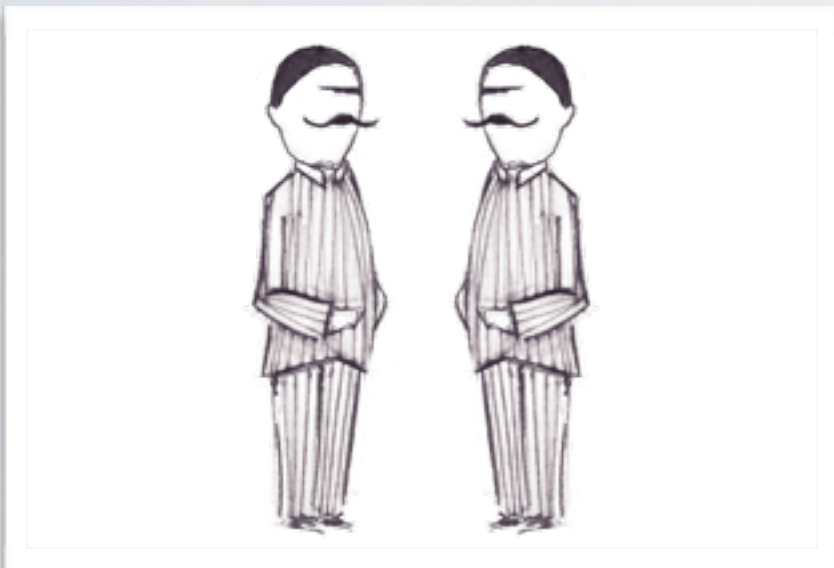




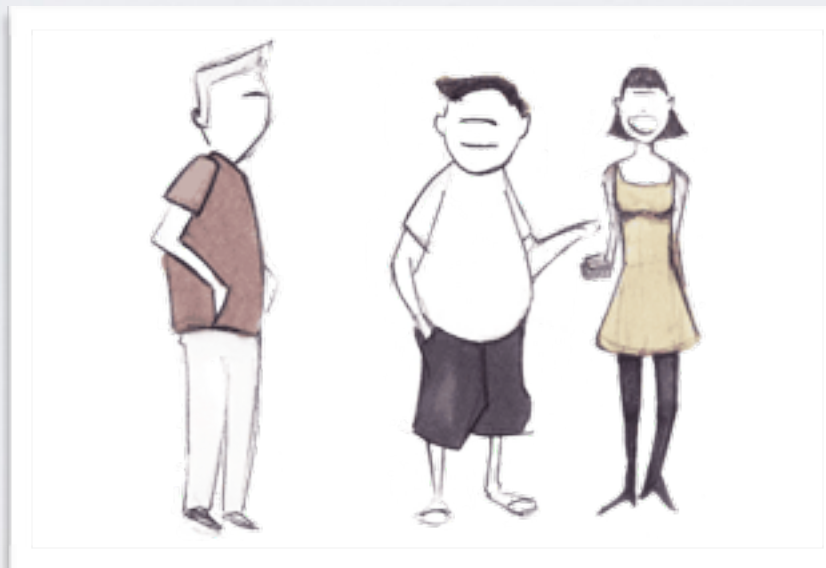
**observations**



**interviews**



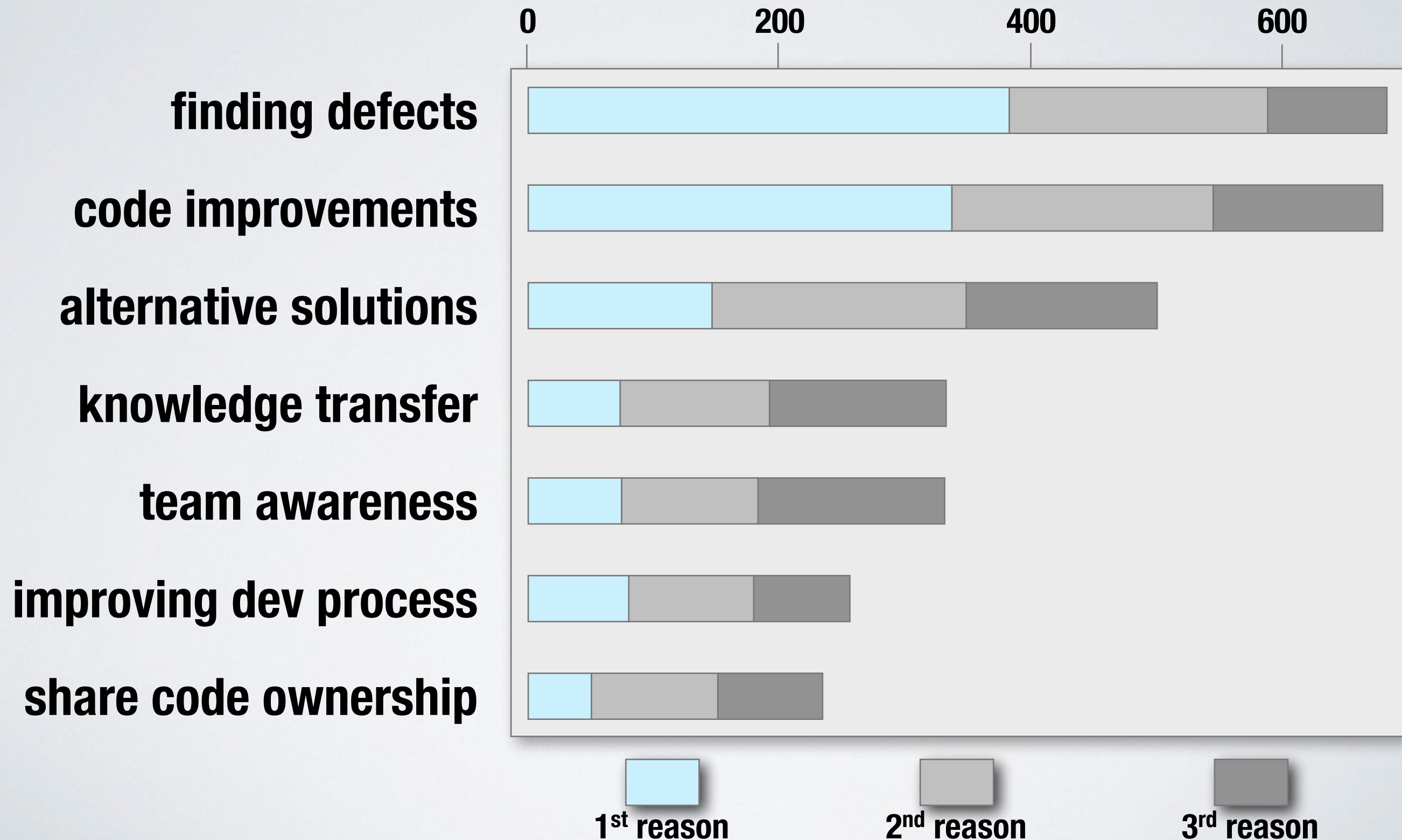
**survey to 165 managers**



**survey to 873 developers**



# Why do Microsoft developers do code reviews?

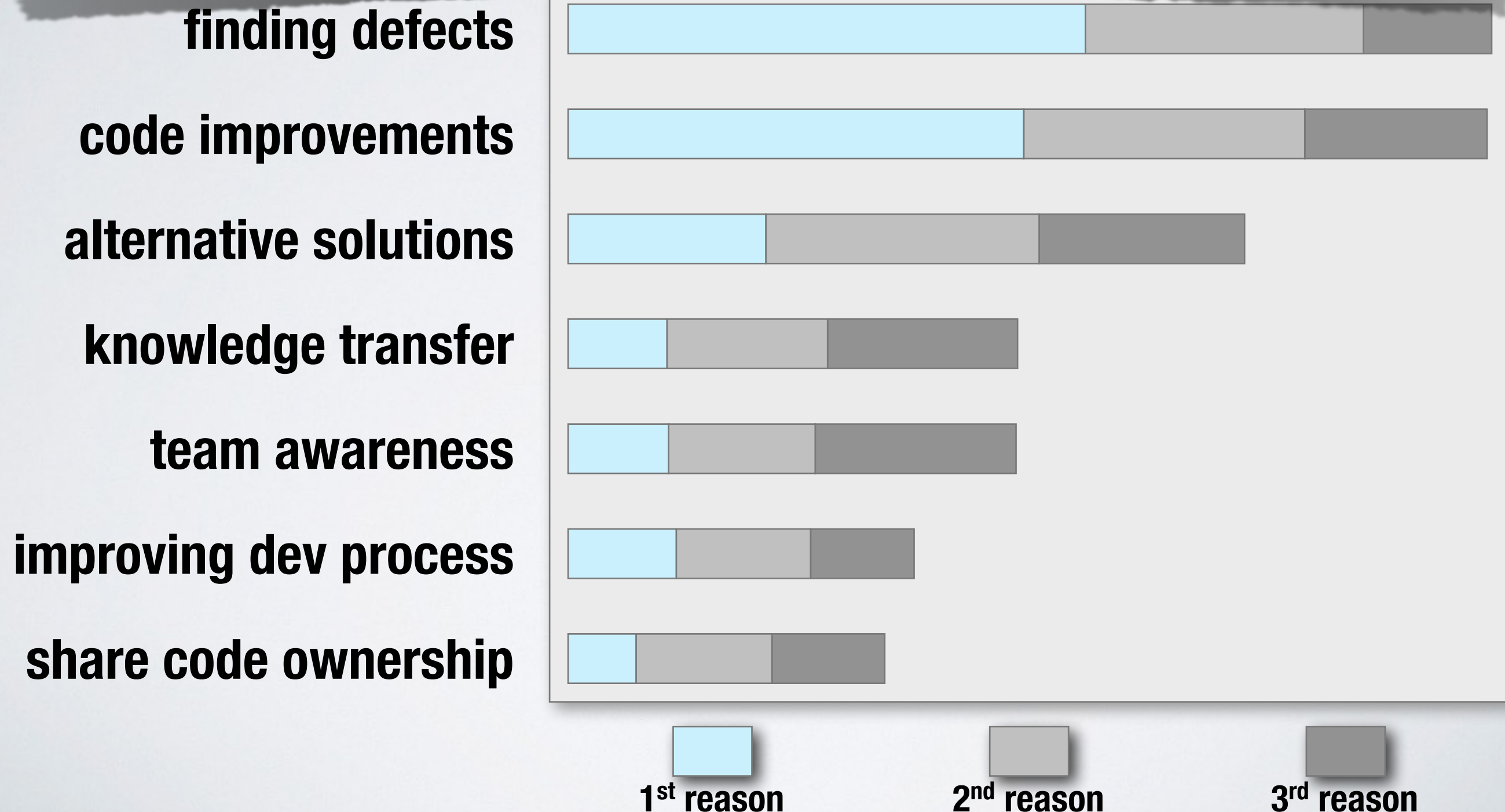




# Why do Microsoft developers do code reviews?

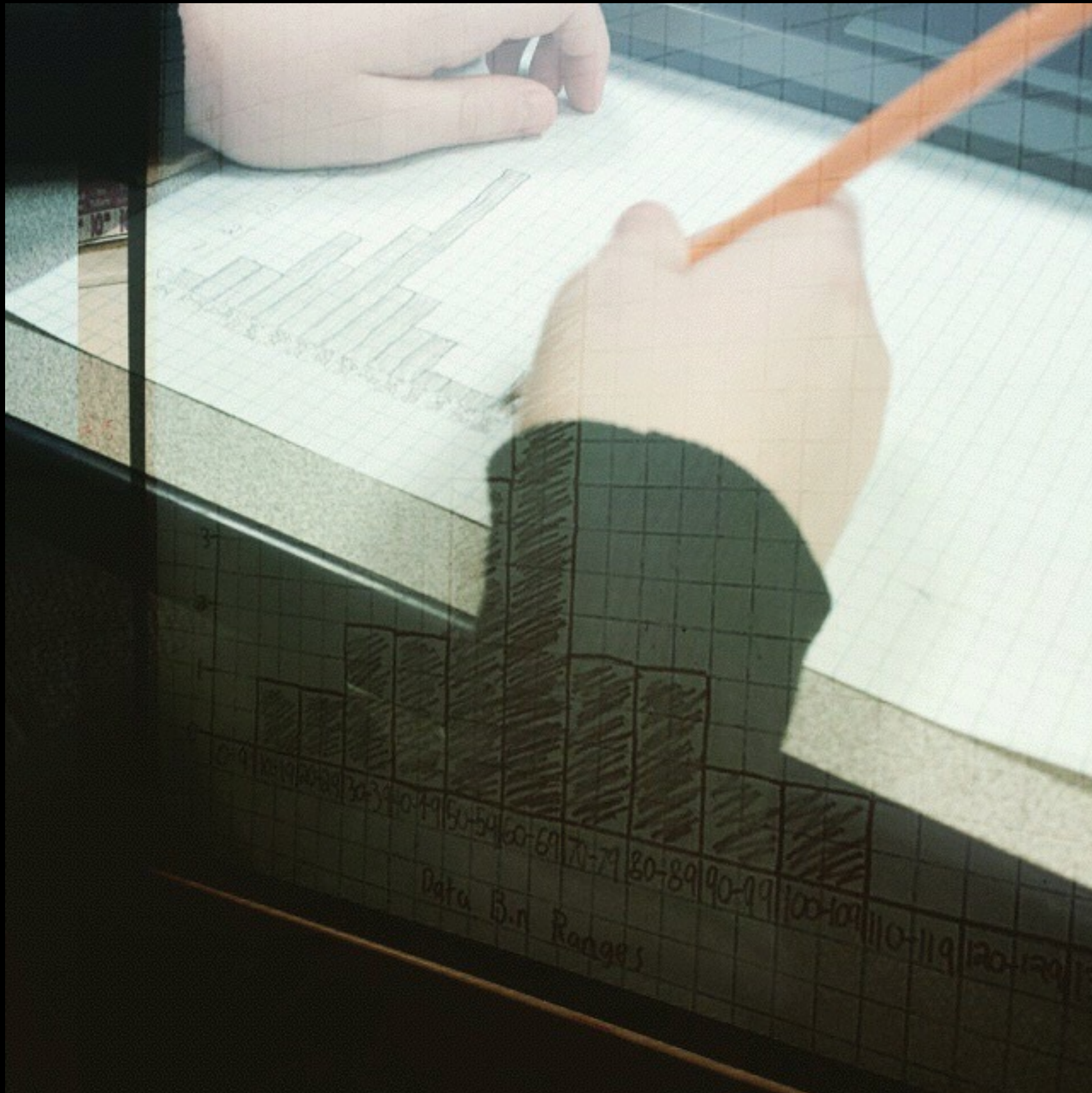
**“Finding defects is the main reason for doing code review.”**

72 managers and 384 developers @ Microsoft





# Let's look at the survey's answers!





# What is the outcome of code review at Microsoft?





# What is the outcome of code review at Microsoft?

The screenshot displays the CodeFlow application interface for a code review session titled "Make the greeting changeable - CodeFlow". The main window shows "Viewing Iteration 1" of the file "Program.cs".

**File Explorer (Left):** Lists files including "description.txt", "Hello.csproj", "Program.cs", and "Test.cs".

**Code Editor (Center):** Shows the following code snippet:

```
37 for (int i = 0; i < Times; i++)
38 {
39     Console.WriteLine("Hello {0}!", Name);
40     Console.WriteLine("{0} {1}!", Greeting, Name);
41 }
42
43
44
```

The code is highlighted with a red box for the first line and a yellow box for the second line. A green box highlights the "Greeting" parameter in the second line.

**Comment Thread (Right):** A comment box is open, showing a discussion:

- Alberto Bacchelli: "Wouldn't it be better to put this as a parameter of the SayGreeting method?"
- Tom Zimmermann: "I wouldn't. Greeting is already a field! If you do that, you'd want to make Times a parameter as well."
- Christian Bird: "Good point. I'll leave it as is."

The comment thread is marked as "Resolved" and has a "Hide" button.

**Reviewer Status (Bottom Left):** A panel showing the status of reviewers:

- Christian Bird (author)
- Alberto Bacchelli
- Tom Zimmermann
- Nachi Nagappan

Buttons for "Complete Review", "Recall", and a "Review" icon are present.

**Review Log (Bottom Right):** A table showing the status of the review:

Status	File name
Active	\$/esereseach/Code/CBirdUtil/Hello/Program.cs
[Tom Zimmermann] Don't forget to initialize.	
[Christian Bird] Should we initialize to "Hello" or throw an error if the user does	
Resolved	\$/esereseach/Code/CBirdUtil/Hello/Program.cs



# Recorded code review comments

The screenshot displays the CodeFlow application interface for a code review session titled "Make the greeting changeable - CodeFlow". The main window shows "Viewing Iteration 1" of the file "Program.cs".

**File Explorer (Left):**

- description.txt (None)
- S/esereseach/Code/CBirdUtil/Hello (Folder)
- Hello.csproj (Edit)
- Program.cs (Edit)
- Test.cs (Add)

**Code Editor (Center):**

```
37 for (int i = 0; i < Times; i++)
38 {
39     Console.WriteLine("Hello {0}!", Name);
40     Console.WriteLine("{0} {1}!", Greeting, Name);
41 }
42
43
44
```

**Comment Thread (Right):**

Wouldn't it be better to put this as a parameter of the SayGreeting method?

Alberto Bacchelli

I wouldn't. Greeting is already a field! If you do that, you'd want to make Times a parameter as well.

Tom Zimmermann

Good point. I'll leave it as is.

Christian Bird

Resolved Hide

**Reviewer Status (Bottom Left):**

- Christian Bird (author)
- Alberto Bacchelli
- Tom Zimmermann
- Nachi Nagappan

Complete Review Recall

**Review Log (Bottom Right):**

Status	File name
Active	S/esereseach/Code/CBirdUtil/Hello/Program.cs
[Tom Zimmermann] Don't forget to initialize.	
[Christian Bird] Should we initialize to "Hello" or throw an error if the user does	
Resolved	S/esereseach/Code/CBirdUtil/Hello/Program.cs

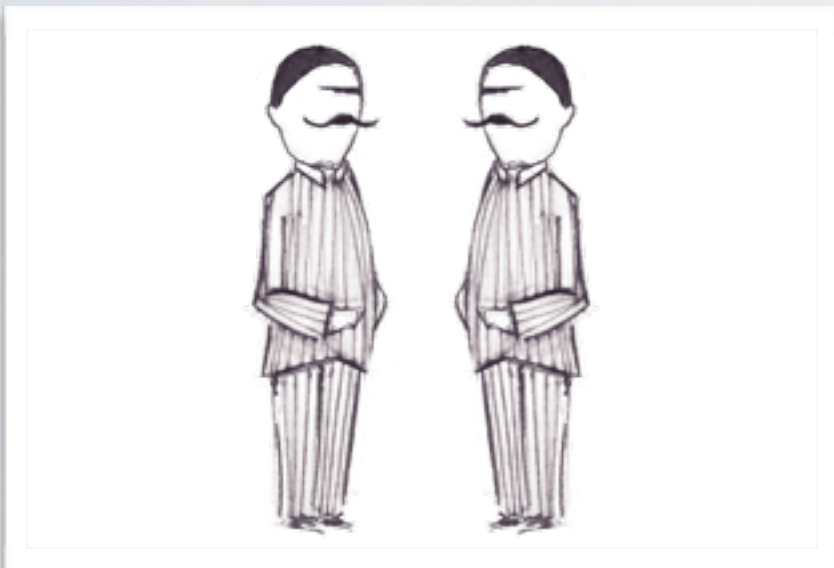




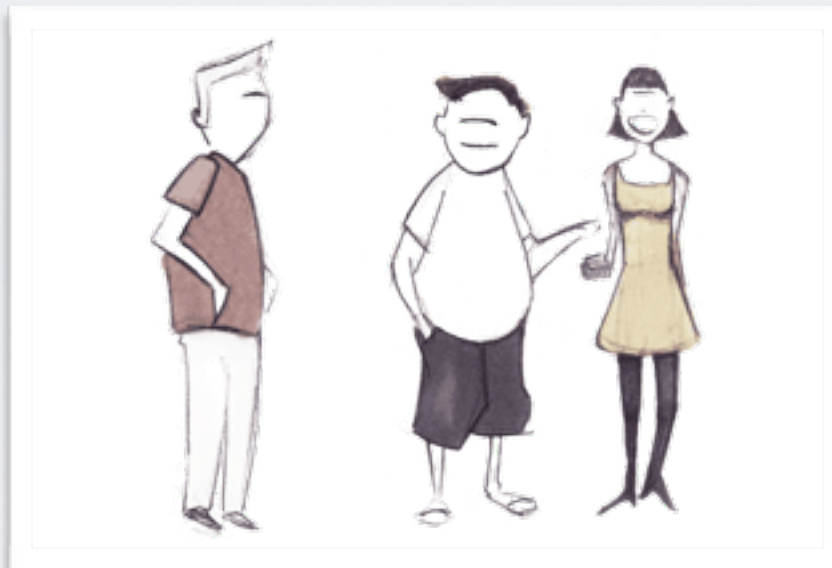
**observations**



**interviews**



**survey to 165 managers**



**survey to 873 developers**

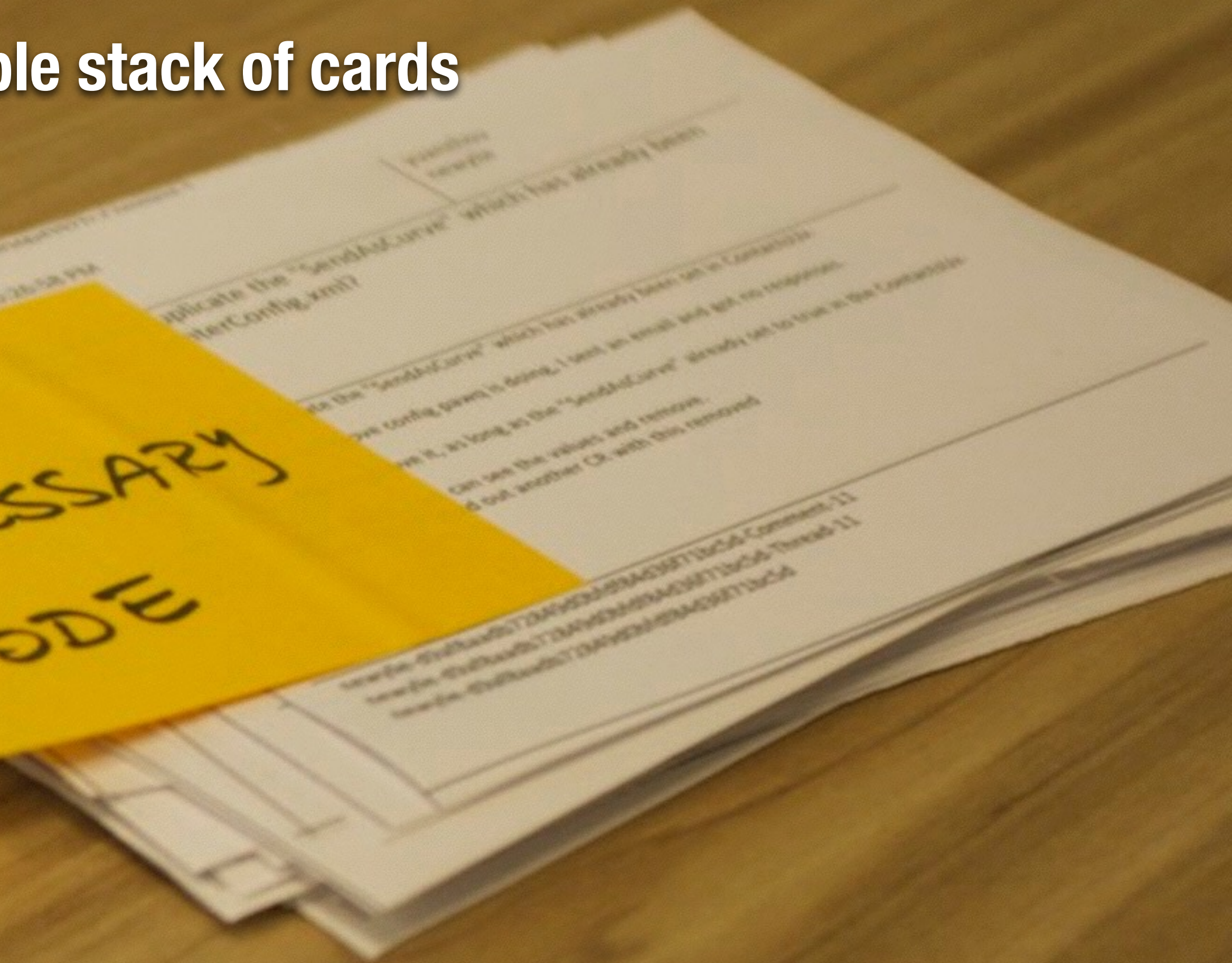
**classification of  
570 review comments**





# Example stack of cards

ST  
NECESSARY  
CODE





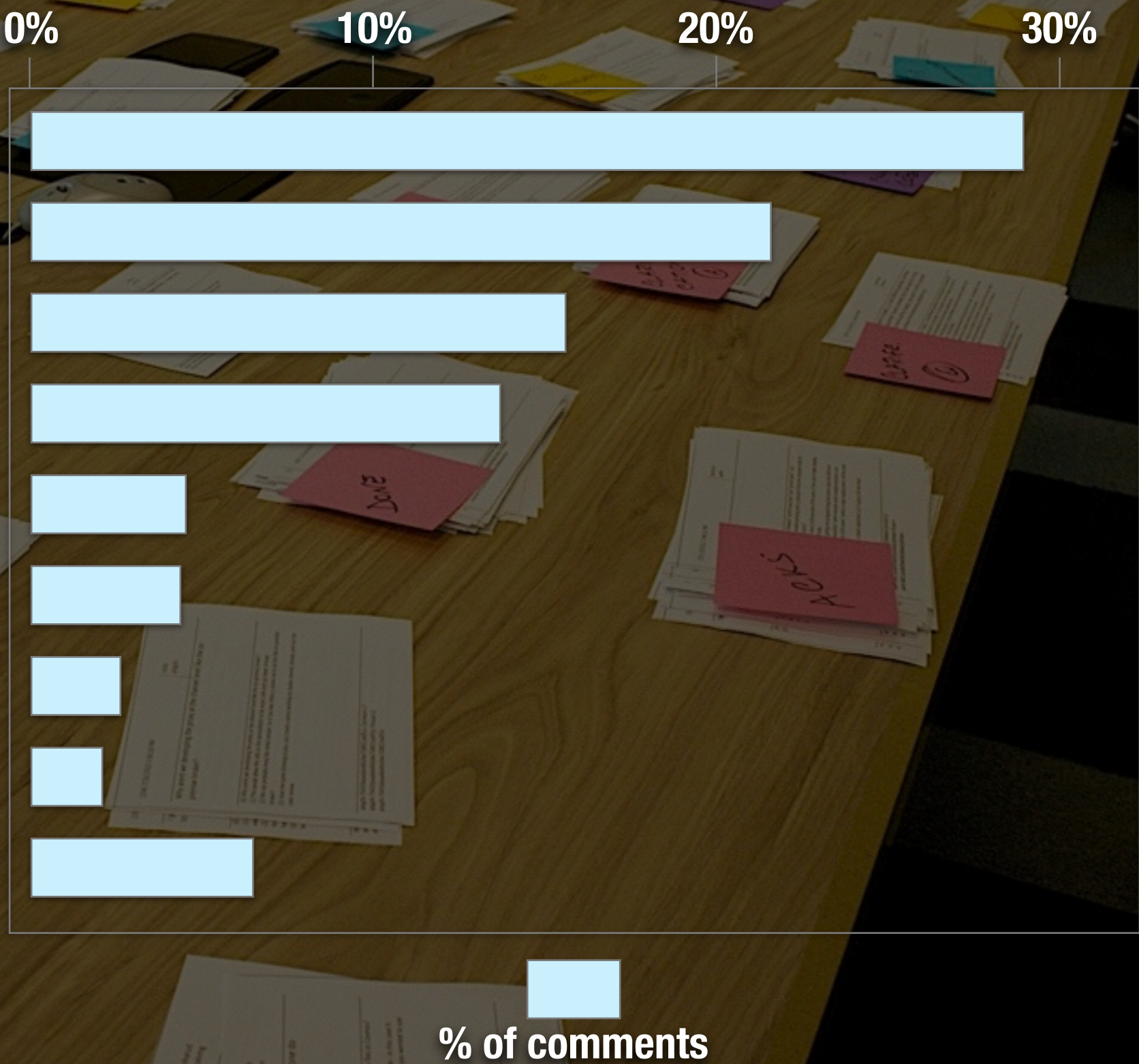
# Card sort almost completed





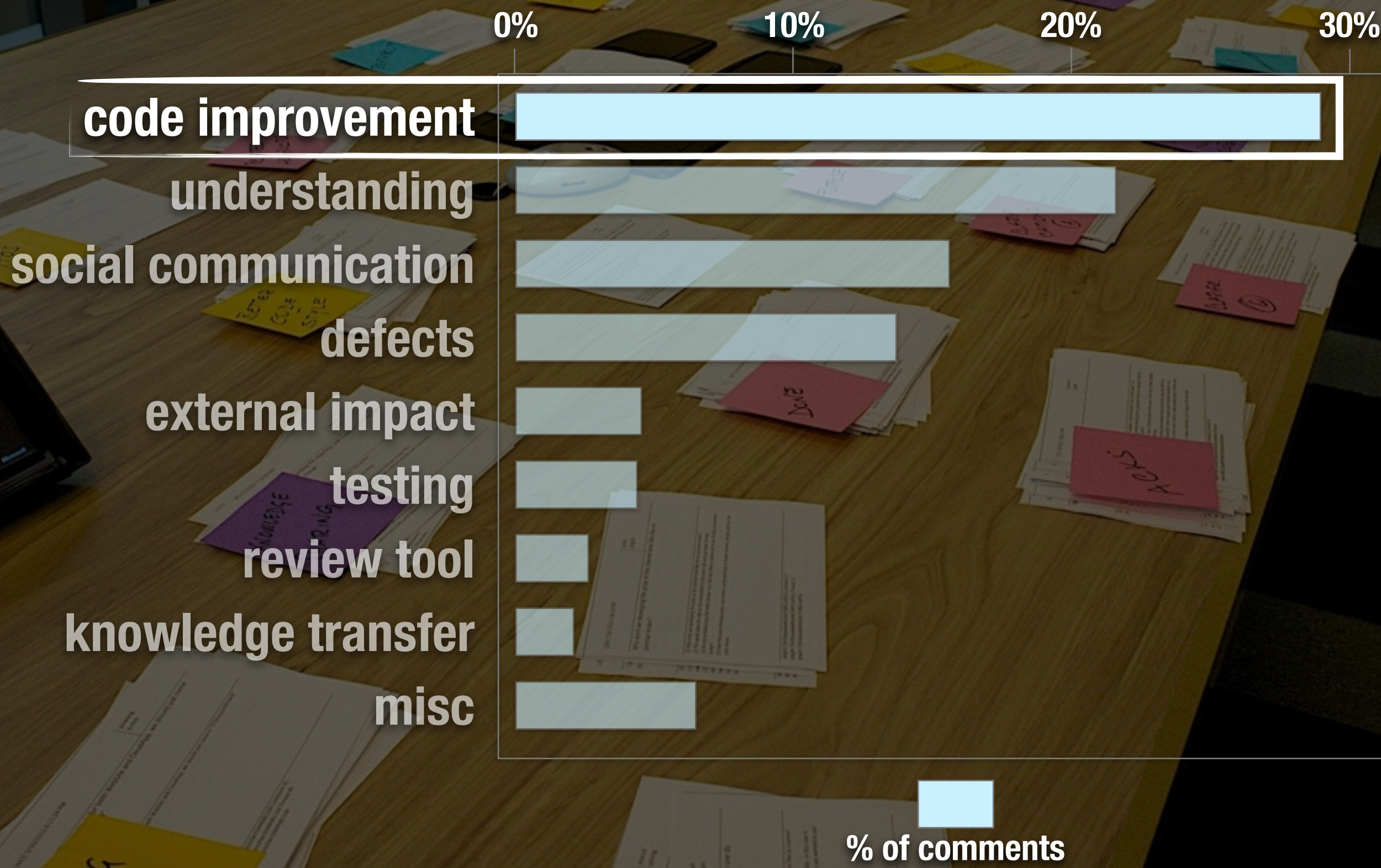
# Card sort results

**code improvement**  
**understanding**  
**social communication**  
**defects**  
**external impact**  
**testing**  
**review tool**  
**knowledge transfer**  
**misc**



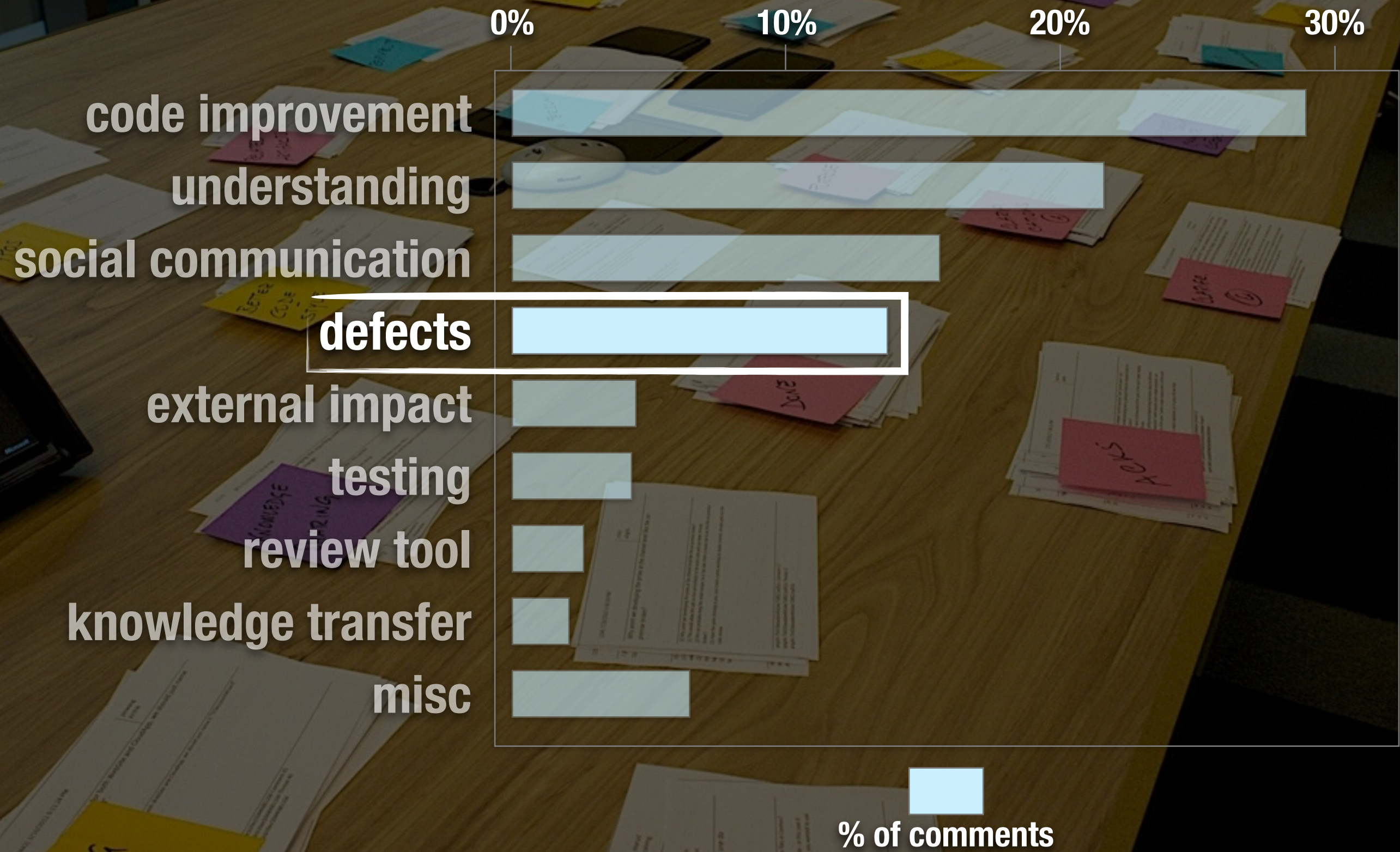


# Card sort results





# Card sort results





# Card sort results defects

**“what if they are all used?”**

**“is it possible that this statement never match?”**

**“should this end date be current date?”**

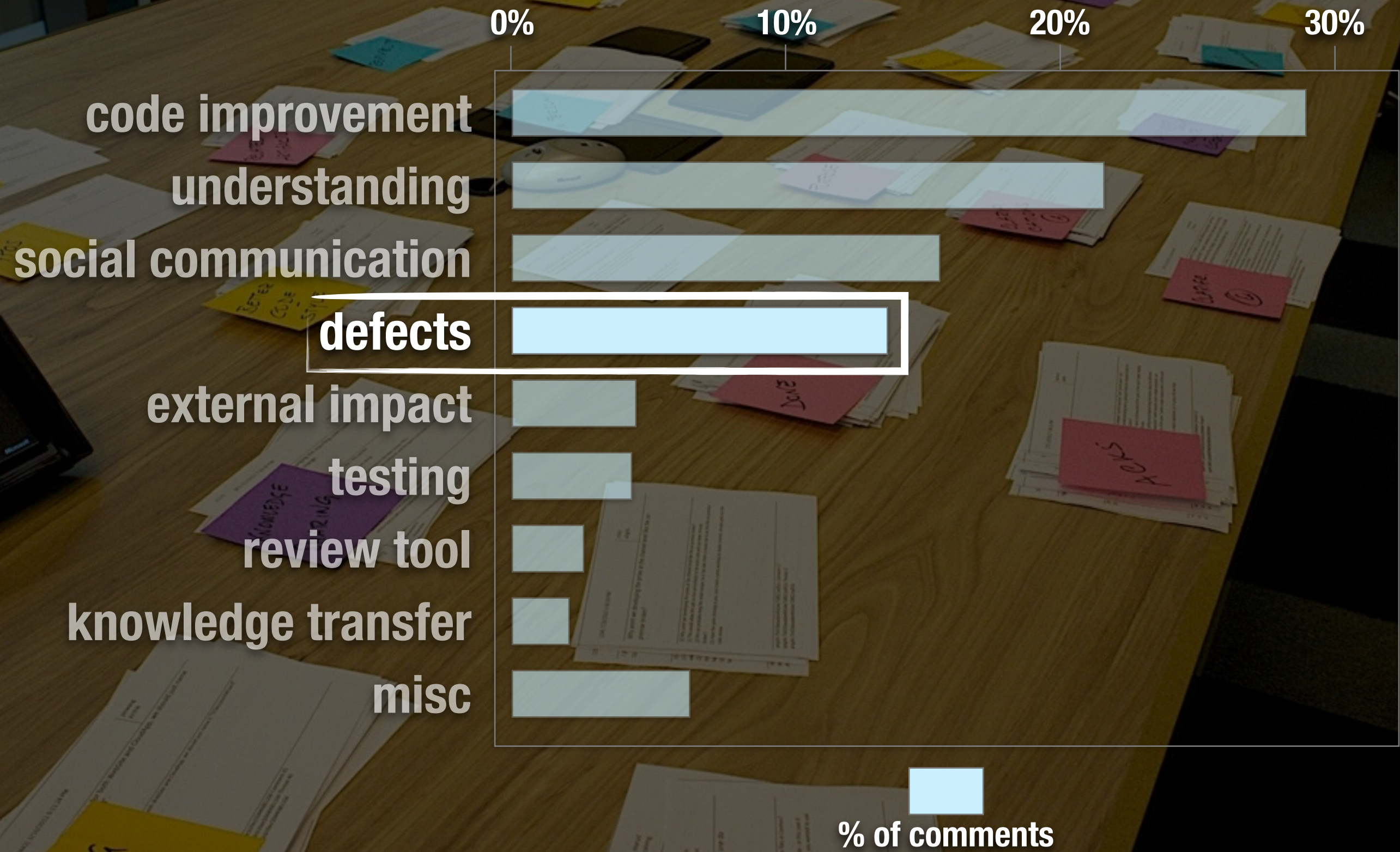
**“does it work if you put 0 here?”**

**“any doubt about the precedence here?”**

**“should be &&?”**



# Card sort results





# Code review at Microsoft: Expectations vs. Reality

## hot chocolate

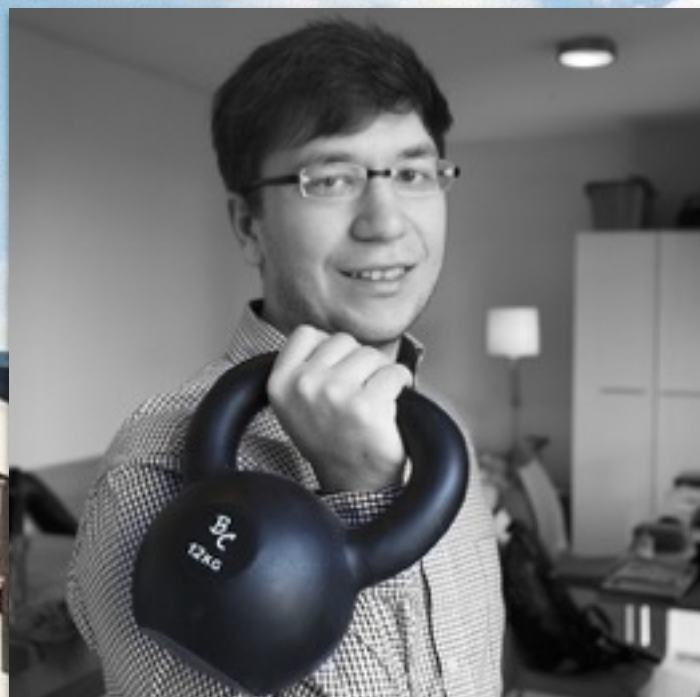




# What is the outcome of code review in OSS?







**Moritz Beller**

  
**TU Delft**





# What is the outcome of code review in OSS?

**GROMACS** FAST.  
FLEXIBLE.  
FREE.

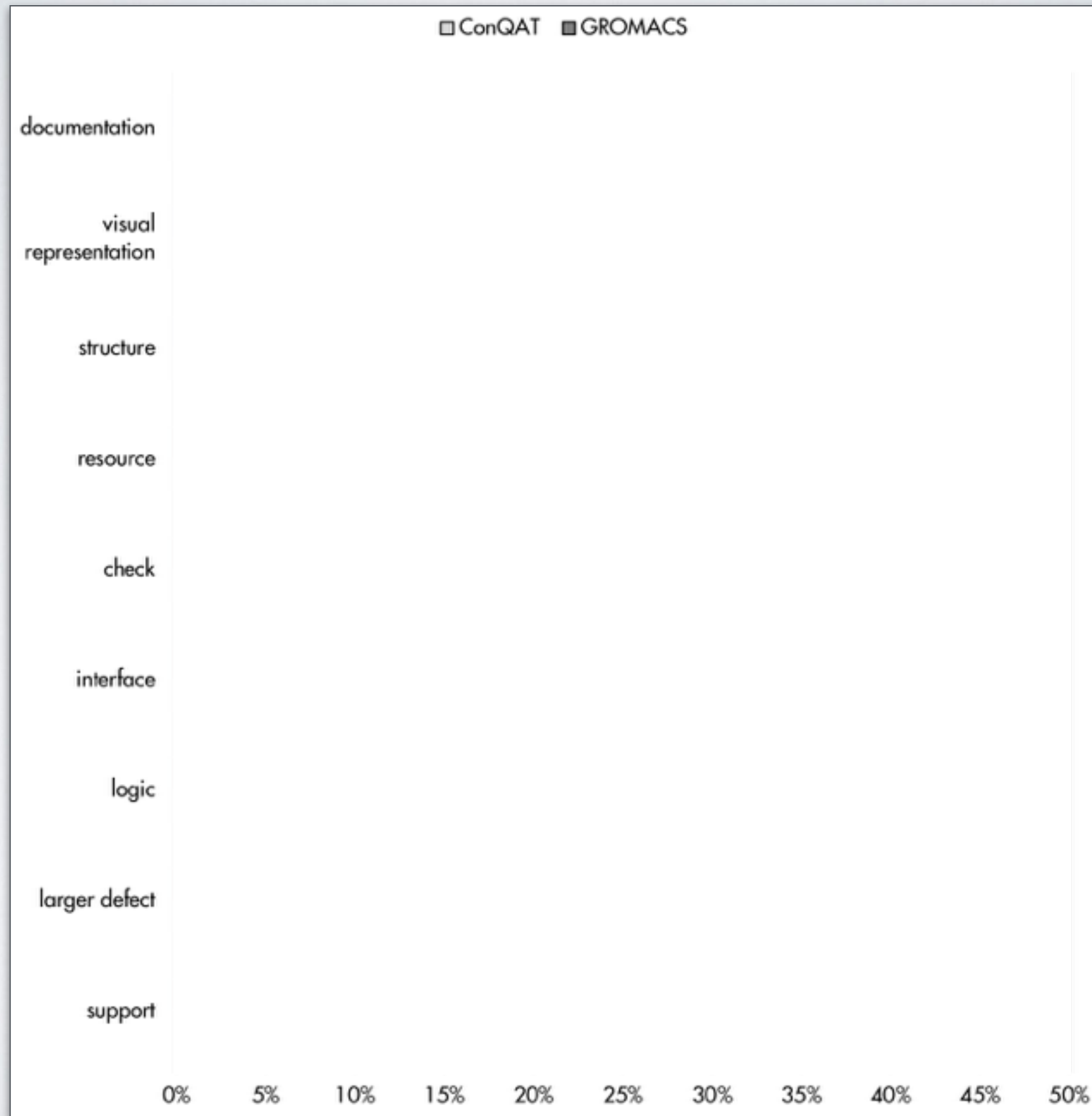
**conQAT**

We manually analyzed  
1,400 review-induced changes

108	/** Validity timer for concatenated SMS segment */	108	/** Max time value between concatenated SMS segments */
Wink Saville	Maybe reword as: Max time value between concatenated SMS seg...		
109	private static final int VALID_TIMER_CONCAT_SEGMENT = 256000;	109	private static final int MAX_CONCATENATED_SEGMENT_TIME =
Wink Saville	Rename to MAX_CONCATENATED_SEGMENT_TIME		
110	private static final int PDU_COLUMN = 0;	110	private static final int PDU_COLUMN = 0;
111	private static final int SEQUENCE_COLUMN = 1;	111	private static final int SEQUENCE_COLUMN = 1;
112	private static final int DESTINATION_PORT_COLUMN = 2;	112	private static final int DESTINATION_PORT_COLUMN = 2;
113	private static final int DATE_COLUMN = 3; //check validity time	113	private static final int DATE_COLUMN = 3;
Wink Saville	Remove comment.		
114		114	
	... skipped 574 common lines ...+108		... skipped 574 common lines ...+108
575	if (Arrays.equals(oldPdu, pdu)) {	575	if (Arrays.equals(oldPdu, pdu)) {
576	return Intents.RESULT_SMS_HANDLED;	576	cursor.close();
Wink Saville	Before returning we should call cursor.close().	577	return Intents.RESULT_SMS_HANDLED;
577	}	578	}

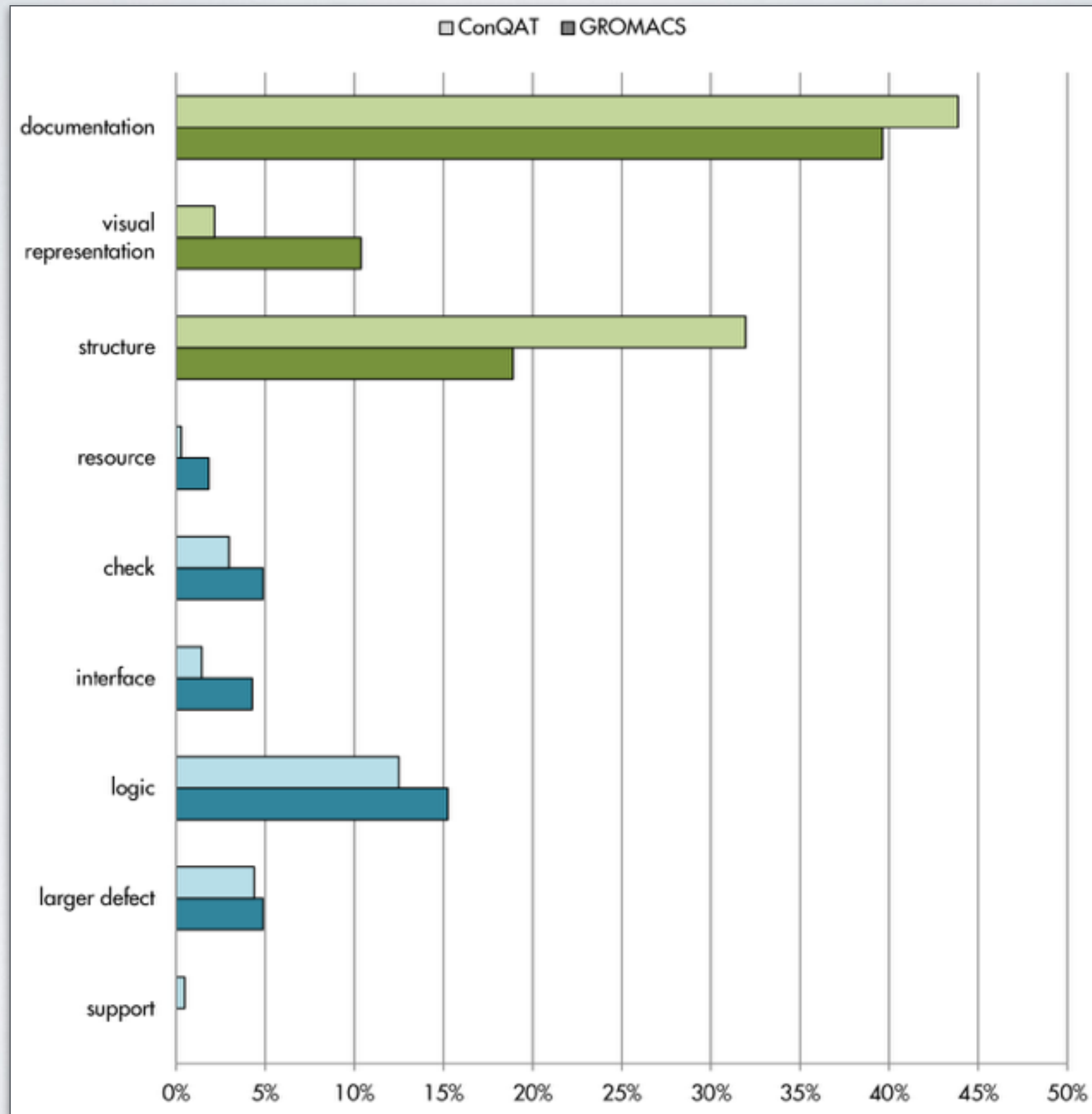


# What is the outcome of code review in OSS?



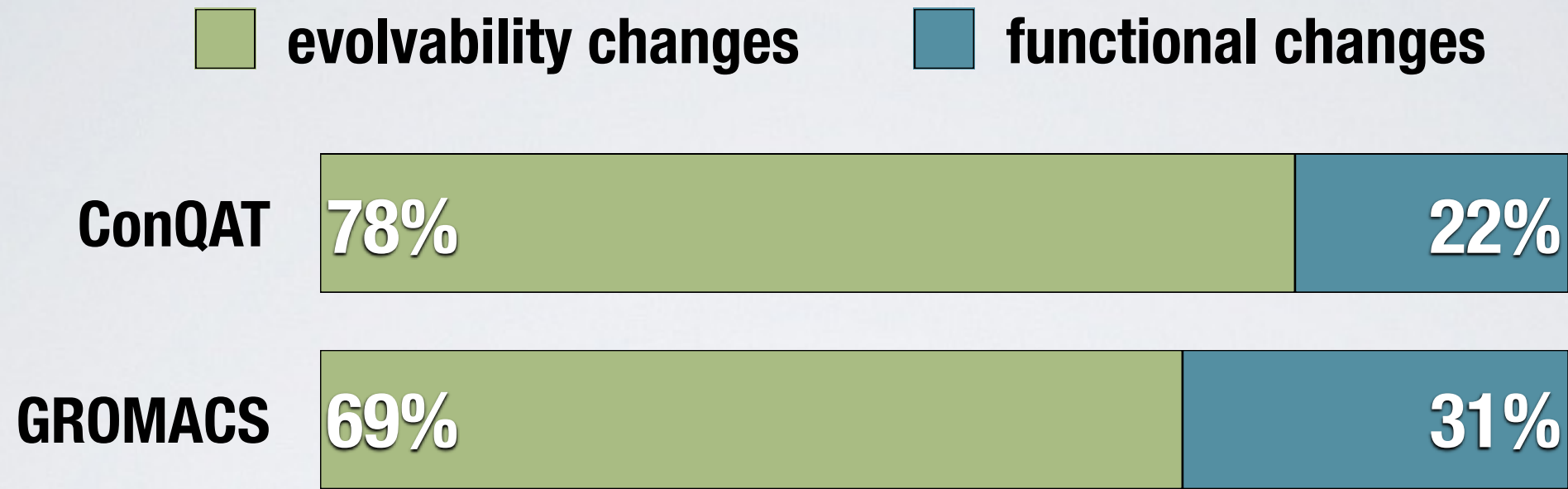


# What is the outcome of code review in OSS?





# What is the outcome of code review in OSS?







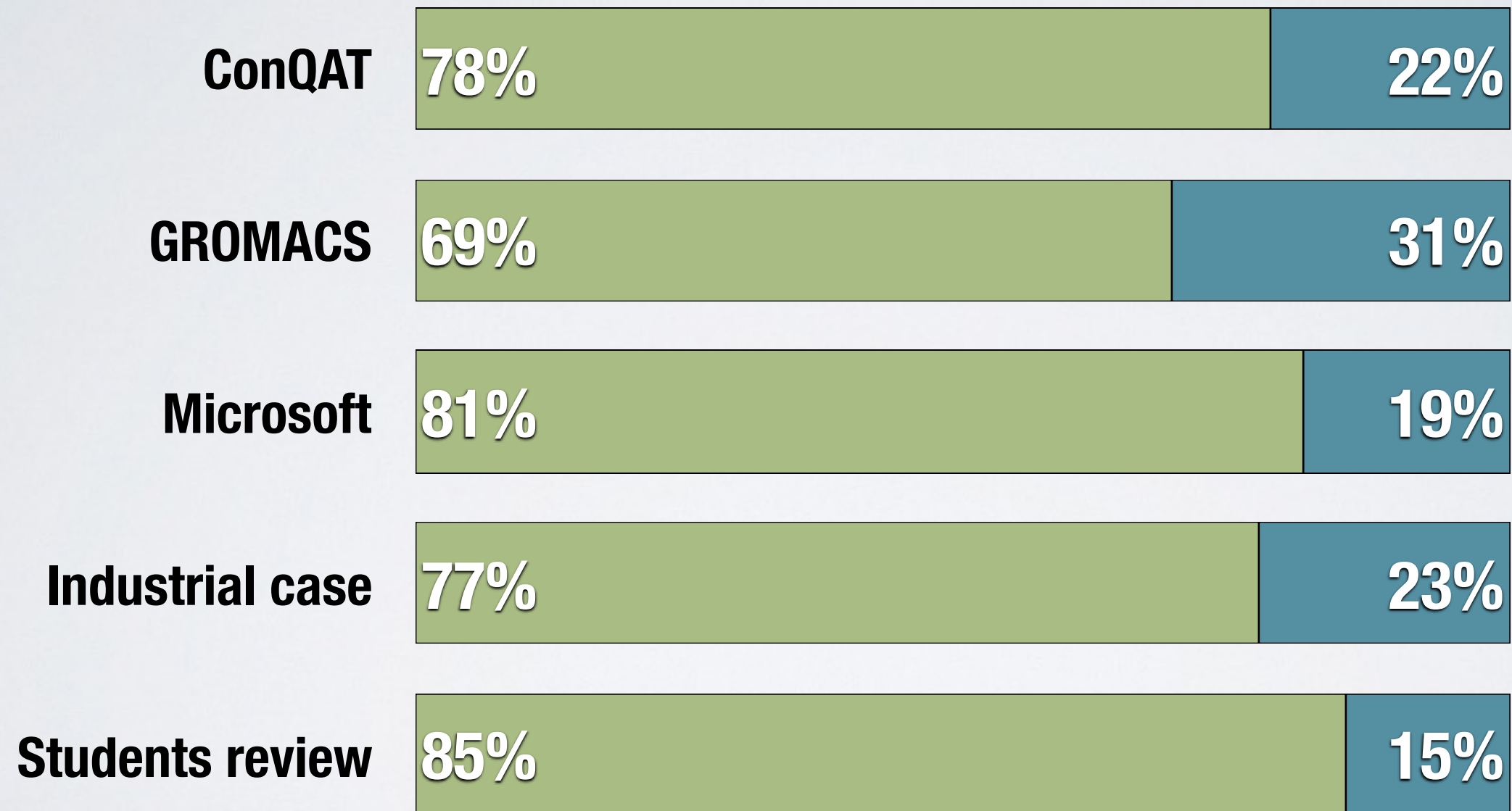
# What is the outcome of code review?





# What is the outcome of code review?

 **evolvability changes**       **functional changes**





# What Do Code Reviews at Microsoft and in Open Source Projects Have in Common?





# What Do Code Reviews at Microsoft and in Open Source Projects Have in Common?



**The outcome**

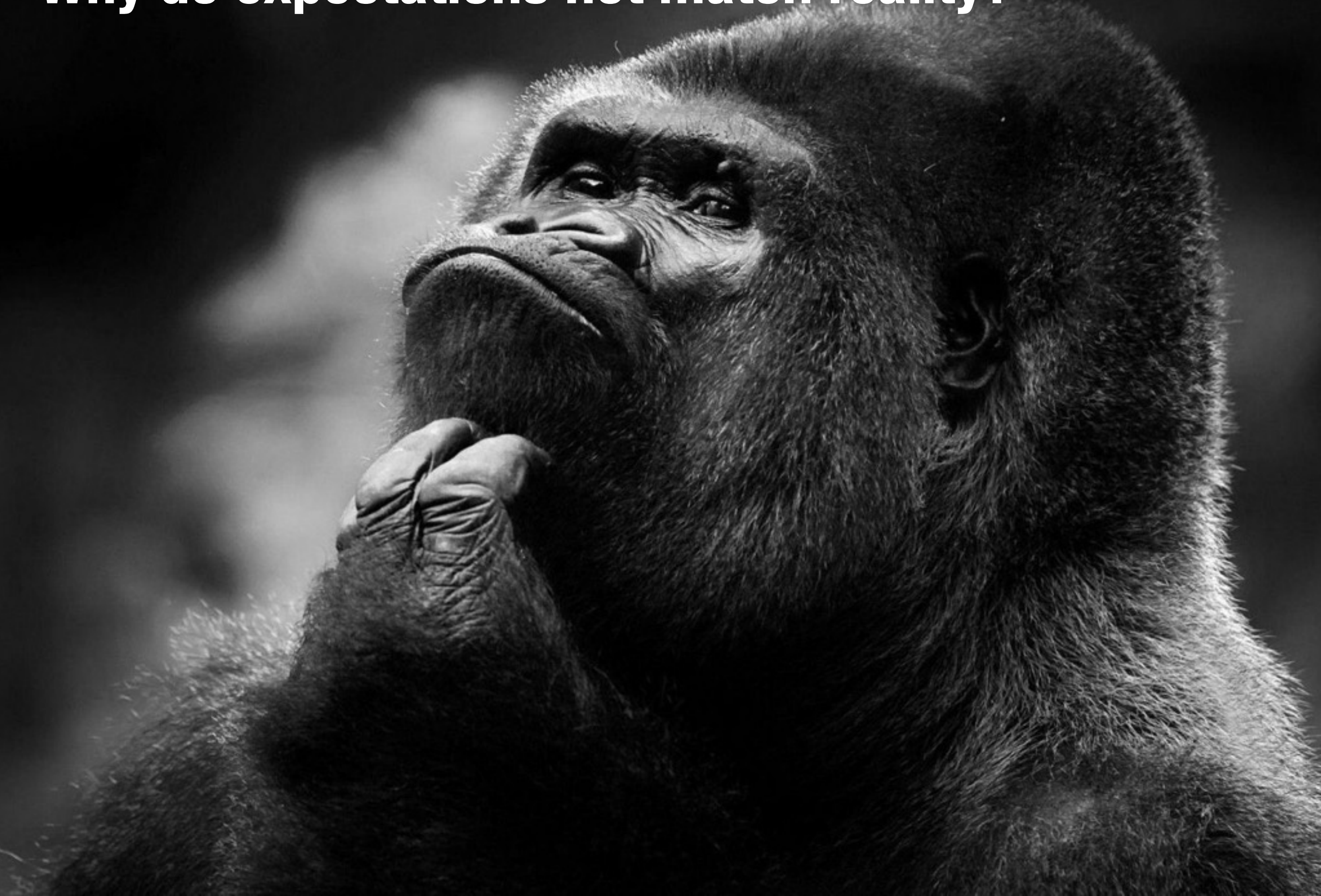








**Why do expectations not match reality?**





# Code reviews

**... “[if] executed properly, [they] find bugs faster and more effectively than testing or other known debugging techniques”**

— Jason Cohen, 2011





# Code reviews

... “[if] executed properly, [they] find bugs faster and more effectively than testing or other known debugging techniques—but when done inefficiently they can quickly become unproductive. ”

— Jason Cohen, 2011





**Code review  
is (still) a  
fully manual task**





# Tools only supports logistics of code review

The screenshot displays the CodeFlow application interface for a code review session. The window title is "Make the greeting changeable - CodeFlow". The main editor shows the file "Program.cs" with a diff view. The code is as follows:

```
37 for (int i = 0; i < Times; i++)
38 {
39     Console.WriteLine("Hello {0}!", Name);
40     Console.WriteLine("{0} {1}!", Greeting, Name);
41 }
42
43
44
```

Annotations in the code include a red highlight on line 39 and a yellow highlight on line 40. A comment thread is visible on the right side of the code editor, with a green box highlighting the comment:

Wouldn't it be better to put this as a parameter of the SayGreeting method?  
Alberto Bacchelli  
I wouldn't. Greeting is already a field! If you do that, you'd want to make Times a parameter as well.  
Tom Zimmermann  
Good point. I'll leave it as is.  
Christian Bird  
Resolved

The left sidebar contains a "Reviewer Status" panel with a list of reviewers: Christian Bird (author), Alberto Bacchelli, Tom Zimmermann, and Nachi Nagappan. Below the list are buttons for "Complete Review" and "Recall".

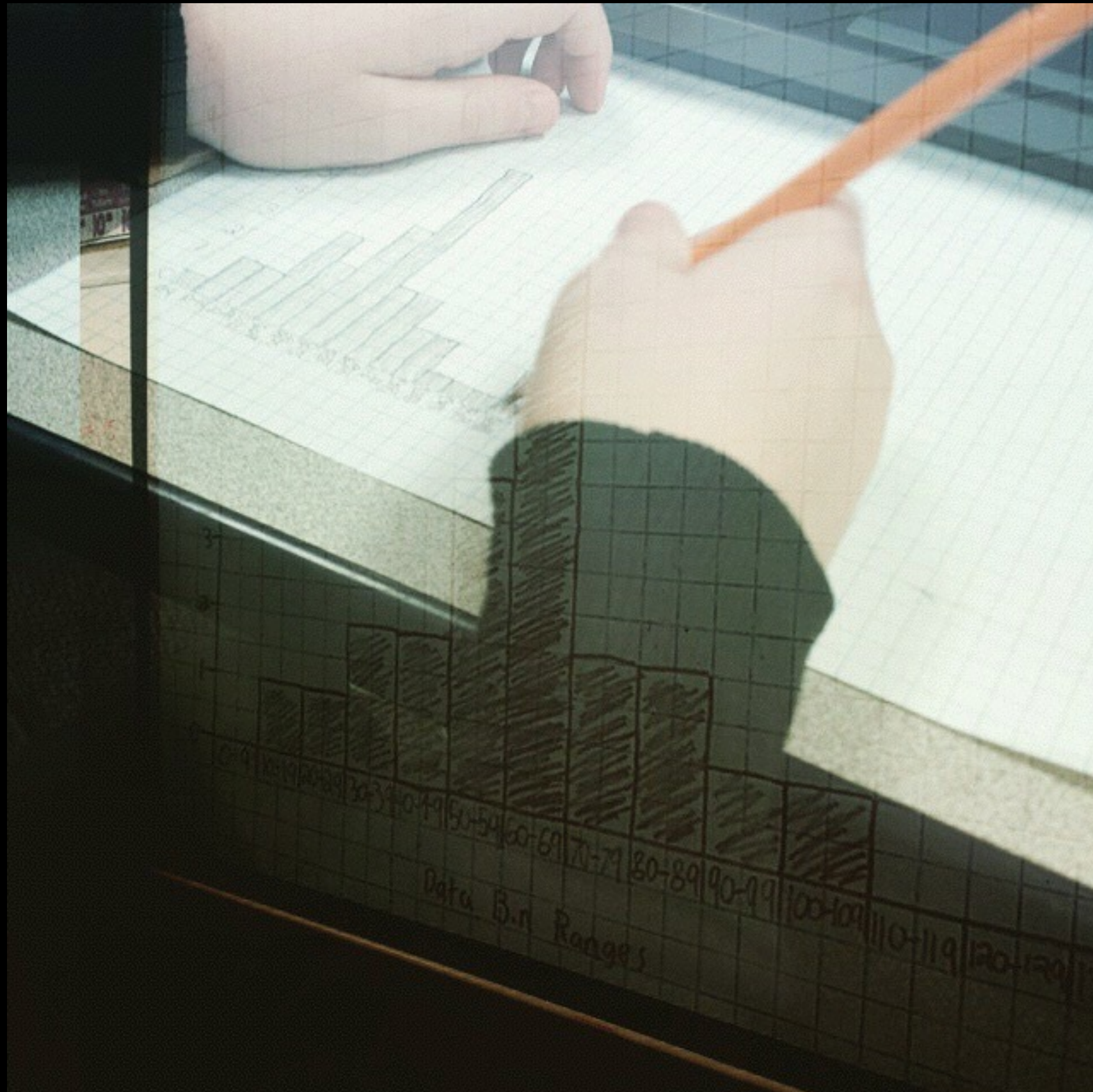
The bottom panel shows a list of review items with columns for "Status" and "File name". The items are:

- Active: \$/esereseach/Code/CBirdUtil/Hello/Program.cs
- [Tom Zimmermann] Don't forget to initialize.
- [Christian Bird] Should we initialize to "Hello" or throw an error if the user does
- Resolved: \$/esereseach/Code/CBirdUtil/Hello/Program.cs

Numbered callouts (1-5) highlight specific features: 1 points to the file explorer, 2 points to the reviewer status panel, 3 points to the code editor, 4 points to the comment thread, and 5 points to the review items list.



# Let's look at the survey's answers!





# **What are the challenges of code review at MSFT?**



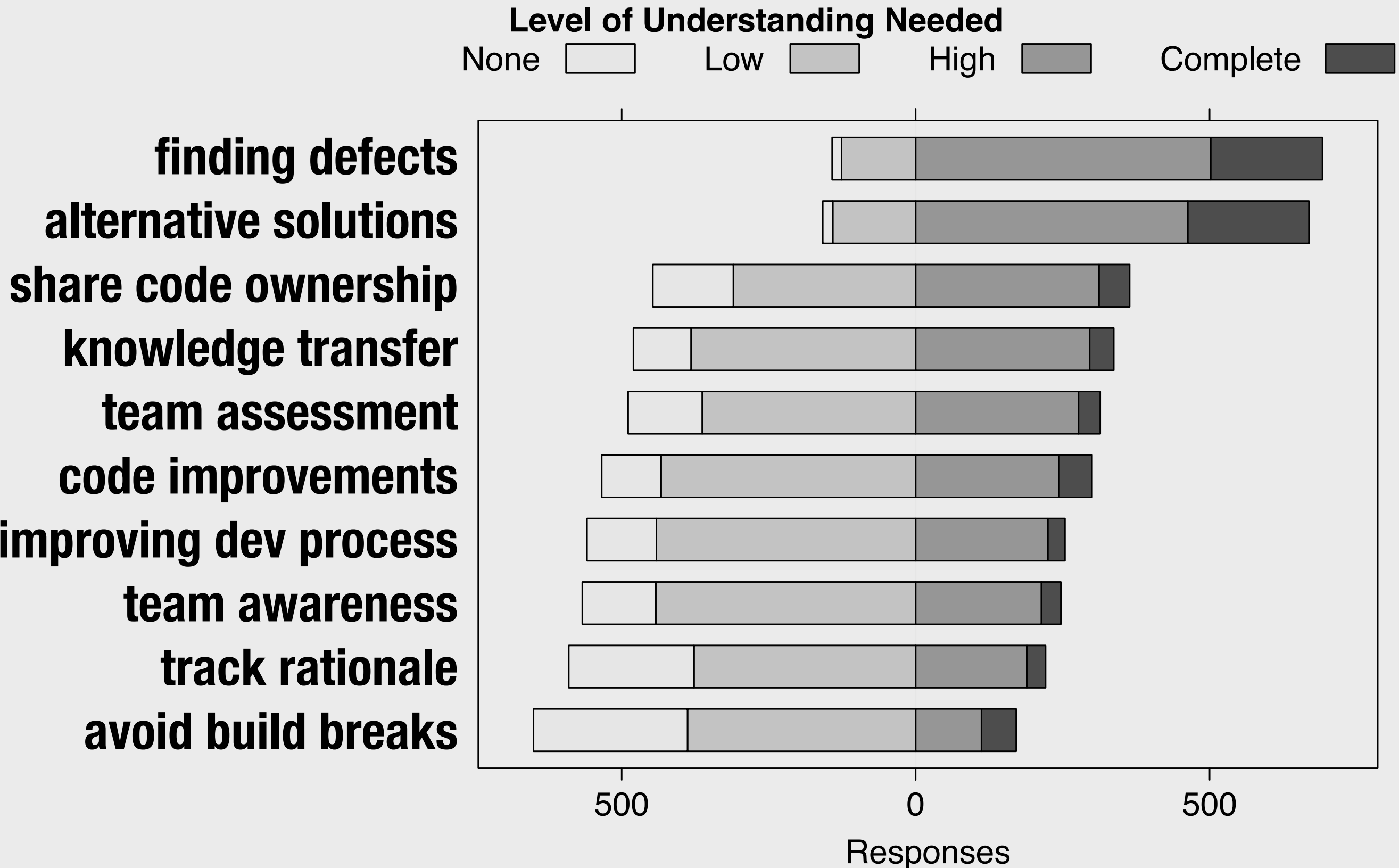
# What are the challenges of code review at MSFT?

*“understanding the code takes most of the reviewing time.”*



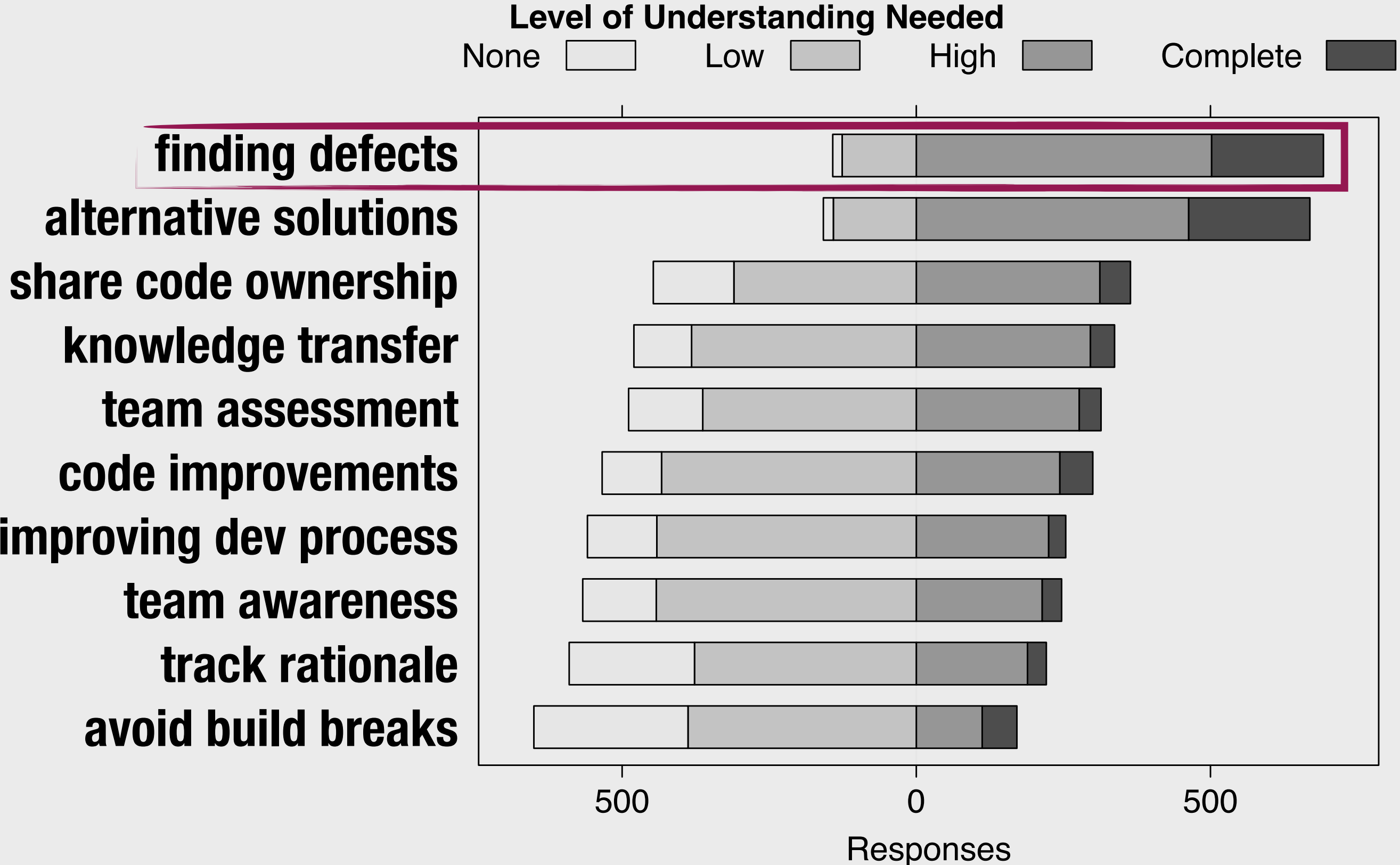


# Understanding needs, by outcome



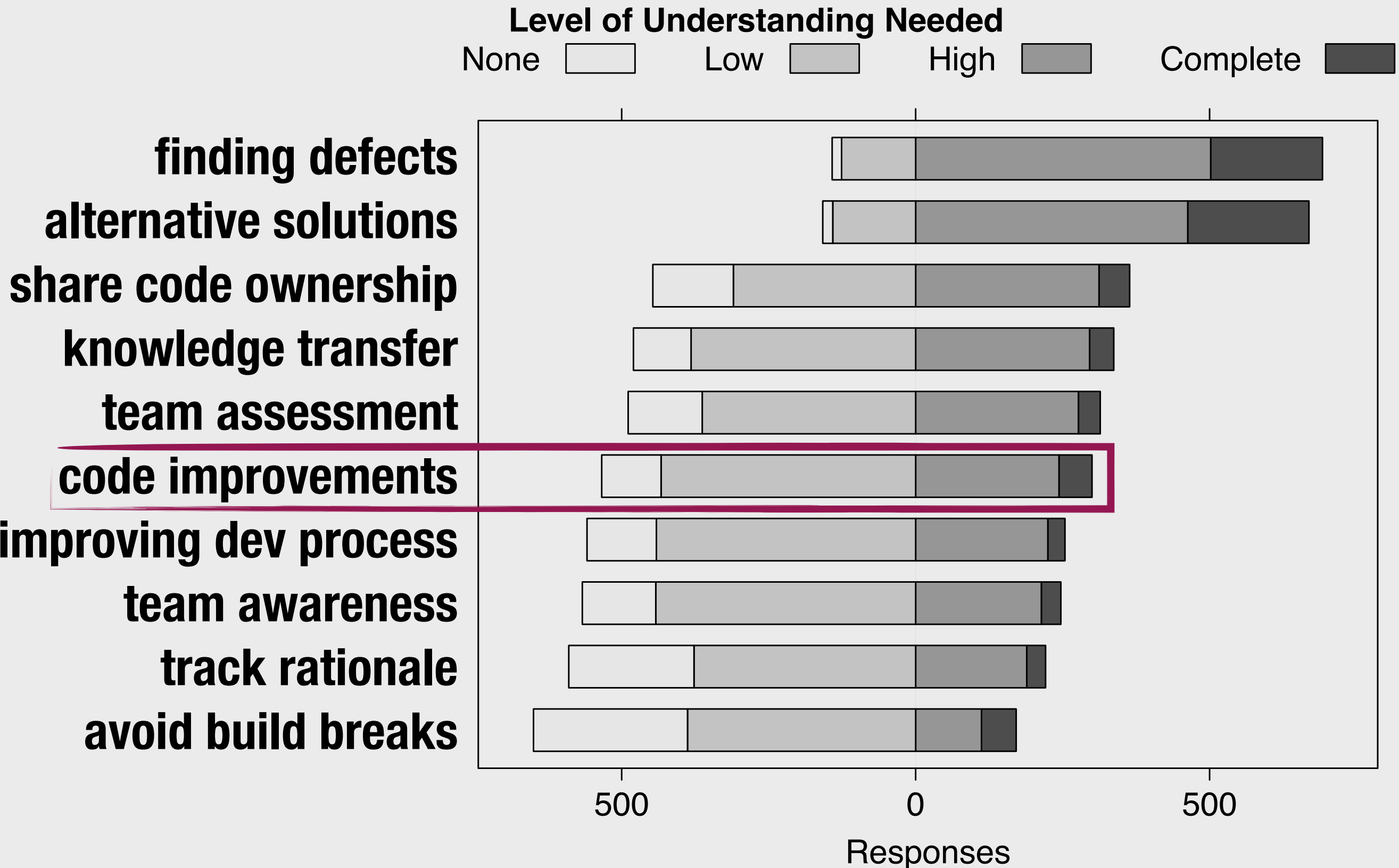


# Understanding needs, by outcome





# Understanding needs, by outcome





# Effect of code ownership on reviews

**Does it take longer to review files that you are not familiar with (or files that are new)?**

***“YES”***

**798 developers (91%)**

**Is there a difference in comments/feedback you receive when a reviewer is very familiar with or the owner of the files you changed in a code review?**

***“YES”***

**716 developers (82%)**



# Effect of code ownership on reviews

**Is there a difference in comments/feedback you receive when a reviewer is very familiar with or the owner of the files you changed in a code review?**

***“YES”***

**716 developers (82%)**

***“Comments reflect their deeper understanding – more likely to find subtle defects, feedback is more conceptual (better ideas, approaches) instead of superficial (naming, mechanical style, etc.)”***





# Code review

needs the right data and the right tools

---





# Software analytics

... “is analytics on software data for managers and software engineers with the aim of empowering software development individuals and teams to gain and share insight from their data to make better decisions.”

— Menzies and Zimmermann, 2013

FOCUS: GUEST EDITORS' INTRODUCTION

## Software Analytics: So What?

Tim Menzies, West Virginia University

Thomas Zimmermann, Microsoft Research

**IN THIS SPECIAL** issue of *IEEE Software*, we invited submissions that reflected the benefits (and drawbacks) of software analytics. The response was overwhelming. Software analytics is an area of explosive growth, and we had so many excellent submissions that we had to split this special issue into two volumes—you'll see even more content in the September/October issue. We divided the articles on conceptual grounds, so both volumes will feature equally excellent work.

To better frame these articles, we offer some definitions and historical perspectives on software analytics. Specifically, we describe where the field was, where it is, and where it might be going.

### What Is Software Analytics?

Thanks to the Internet and open source, there's now so much data about software projects that it's impossible to manually browse through it all:

- As of late 2012, our Web searches show that Mozilla Firefox had 800,000 bug reports, and platforms such as Sourceforge.net and GitHub hosted 324,000 and 11.2 million projects, respectively.

# Software analytics

... “is analytics on software data for managers and software engineers with the aim of empowering software development individuals and teams to gain and share insight from their data to make better decisions.”

— Menzies and Zimmermann, 2013

FOCUS: GUEST EDITORS' INTRODUCTION

## Software Analytics: So What?

Tim Menzies, West Virginia University

Thomas Zimmermann, Microsoft Research

**IN THIS SPECIAL** issue of *IEEE Software*, we invited submissions that reflected the benefits (and drawbacks) of software analytics. The response was overwhelming. Software analytics is an area of explosive growth, and we had so many excellent submissions that we had to split this special issue into two volumes—you'll see even more content in the September/October issue. We divided the articles on conceptual grounds, so both volumes will feature equally excellent work.

To better frame these articles, we offer some definitions and historical perspectives on software analytics. Specifically, we describe where the field was, where it is, and where it might be going.

### What Is Software Analytics?

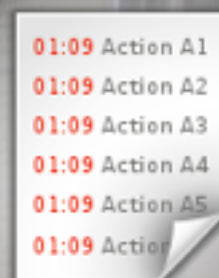
Thanks to the Internet and open source, there's now so much data about software projects that it's impossible to manually browse through it all:

- As of late 2012, our Web searches show that Mozilla Firefox had 800,000 bug reports, and platforms such as Sourceforge.net and GitHub hosted 324,000 and 11.2 million projects, respectively.











# Software analytics' workflow

**software engineering  
tasks helped**



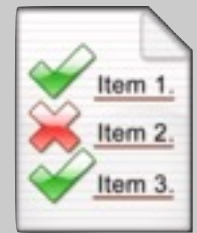
**programming**



**debugging**



**maintenance**



**testing**

...

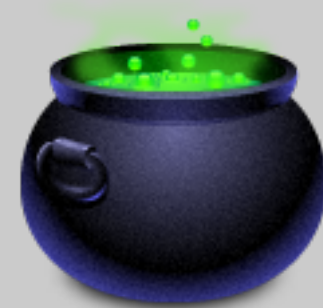
**data mining and  
software analysis  
techniques**



**classification**



**patterns**



**clustering**

...

**software data**



**source  
code**



**versioning  
system**



**issue tracking  
system**

...



# Software analytics for code reviews

**software engineering  
tasks helped**



**code review**

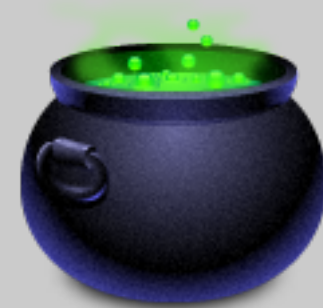
**data mining and  
software analysis  
techniques**



**classification**



**patterns**



**clustering**

...

**software data**



**source  
code**



**versioning  
system**



**issue tracking  
system**



**review  
data**

...



# My research: Data-supported code review

**recommender  
for reviewers**



**change  
untangler**



**automatic  
risk detection**



# My research: Data-supported code review

**recommender  
for reviewers**



**change  
untangler**



**automatic  
risk detection**





# Who should review my code changes?

**recommender  
for reviewers**



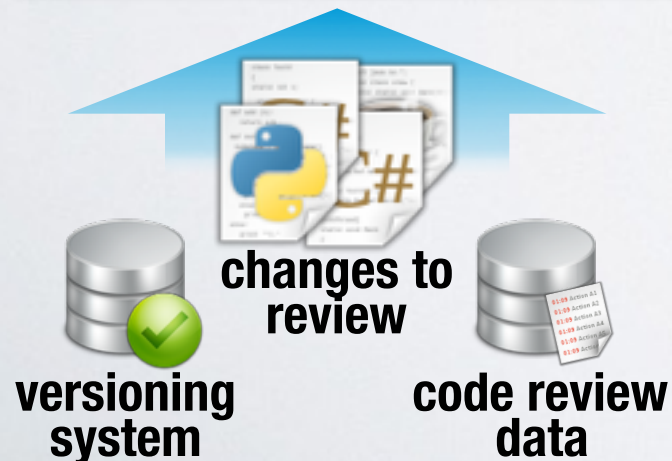
**most appropriate  
reviewer #1**



**most appropriate  
reviewer #3**



**most appropriate  
reviewer #2**



# My research: Data-supported code review

recommender  
for reviewers



change  
untangler



automatic  
risk detection





# How should I split my code for easier review?

**change  
untangler**



**changes to  
review**



**self-contained  
change**



**self-contained  
change**



**versioning  
system**

# My research: Data-supported code review

recommender  
for reviewers



change  
untangler



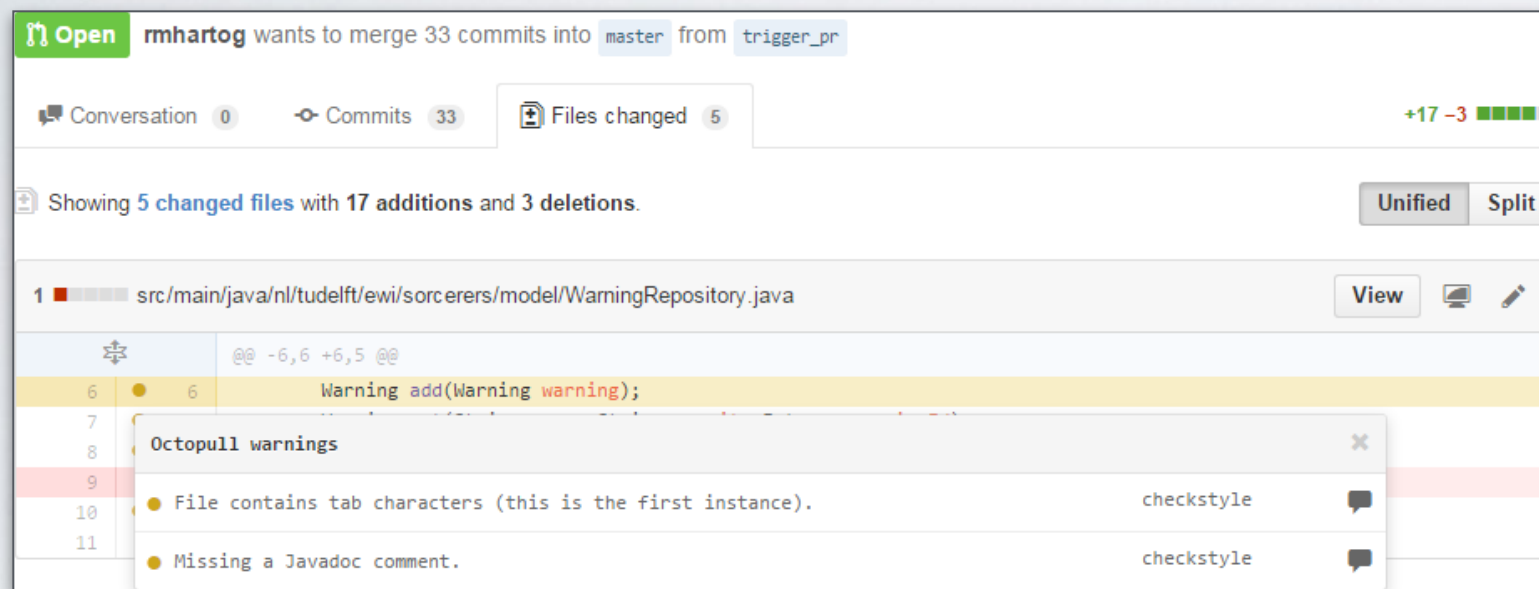
automatic  
risk detection



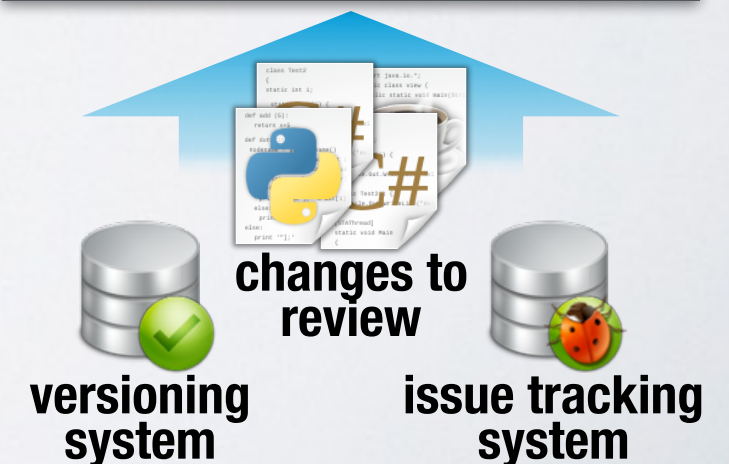


# Which changes should I review more carefully?

**automatic  
risk detection**



**Octopull**



# My research: Data-supported code review

**recommender  
for reviewers**



**change  
untangler**




**automatic  
risk detection**





# Add code review analytics support to GitHub

 **hubot / Spoon-Knife**  
forked from octocat/Spoon-Knife

Watch 0Star 0Fork 34,641

octocat:master ... hubot:masterEdit

Create pull requestDiscuss and review the changes in this comparison with others.

1 commit1 file changed0 commit comments1 contributor

Commits on Sep 10, 2014

hubotChange description of the repository8b7afff

Showing 1 changed file with 1 addition and 1 deletion.UnifiedSplit

2 README.mdView

...	...	@@ -1,6 +1,6 @@
1	1	### Well hello there!
2	2	
3		-This repository is meant to provide an example for <i>forking</i> a repository on GitHub.
	3	+This is my fork of the octocat/Spoon-Knife repository.
4	4	
5	5	Creating a <i>fork</i> is producing a personal copy of someone else's project. Forks act as a sort of bridge between the original repository and your personal copy. You can submit <i>Pull Requests</i> to help make other people's projects better by offering your changes up to the original project. Forking is at the core of social coding at GitHub.
6	6	

# Software analytics for code reviews

software engineering  
tasks helped



code review

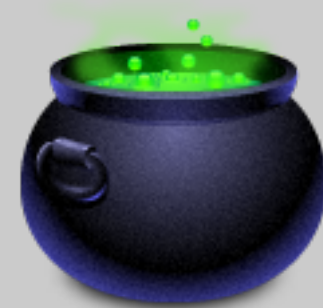
data mining and  
software analysis  
techniques



classification



patterns



clustering

...

software data



source  
code



versioning  
system



issue tracking  
system



review  
data

...



## What Do Code Reviews at Microsoft and in Open Source Projects Have in Common?



## Why research on modern code review?



## Modern Code Review @ Microsoft



## Code review at Microsoft: Expectations vs. Reality



## What Do Code Reviews at Microsoft and in Open Source Projects Have in Common?



## Code review is (still) a fully manual task



## My research: Data-supported code review

recommender  
for reviewers



change  
untangler



automatic  
risk detection





# Would you like to work on these topics (with me)?



**We have 3 fully funded 4-year  
PhD (or postdoc) positions!**

**And we are always looking for  
great students to work  
on fantastic Master theses!**

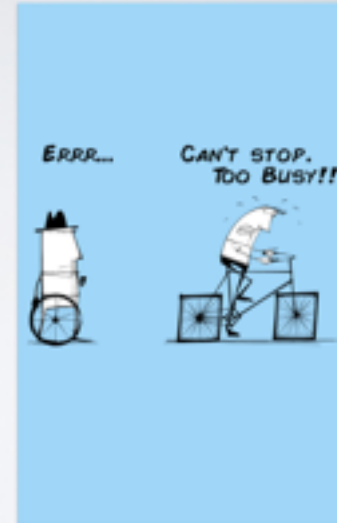
**Find me at the end of the talk,  
at [a.bacchelli@tudelft.nl](mailto:a.bacchelli@tudelft.nl) ,  
or at @sback\_**



## What Do Code Reviews at Microsoft and in Open Source Projects Have in Common?



## Why research on modern code review?



## Modern Code Review @ Microsoft



Dr. Christian Bird

Microsoft  
Research

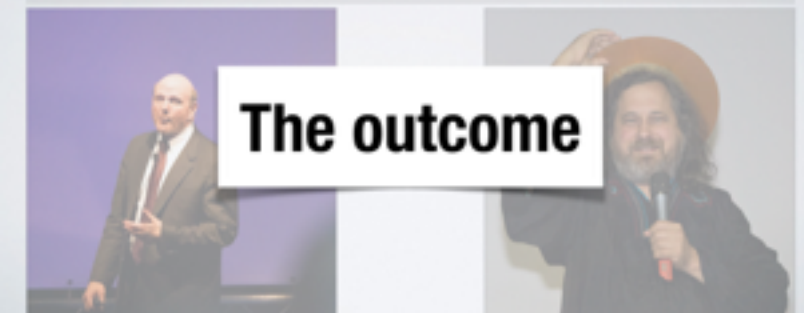
## Code review at Microsoft: Expectations vs. Reality



Moritz Beller

TU Delft

## What Do Code Reviews at Microsoft and in Open Source Projects Have in Common?

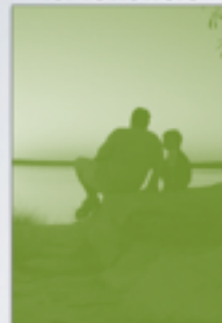


Code review  
is (still) a  
fully manual task



## My research: Data-supported code review

recommender  
for reviewers



change  
untangler



automatic  
risk detection



## Would you like to work on these topics (with me)?



We have 3 fully funded 4-year  
PhD (or postdoc) positions!

And we are always looking for  
great students to work  
on fantastic Master theses!

Find me at the end of the talk,  
at [a.bacchelli@tudelft.nl](mailto:a.bacchelli@tudelft.nl) ,  
or at @sback\_