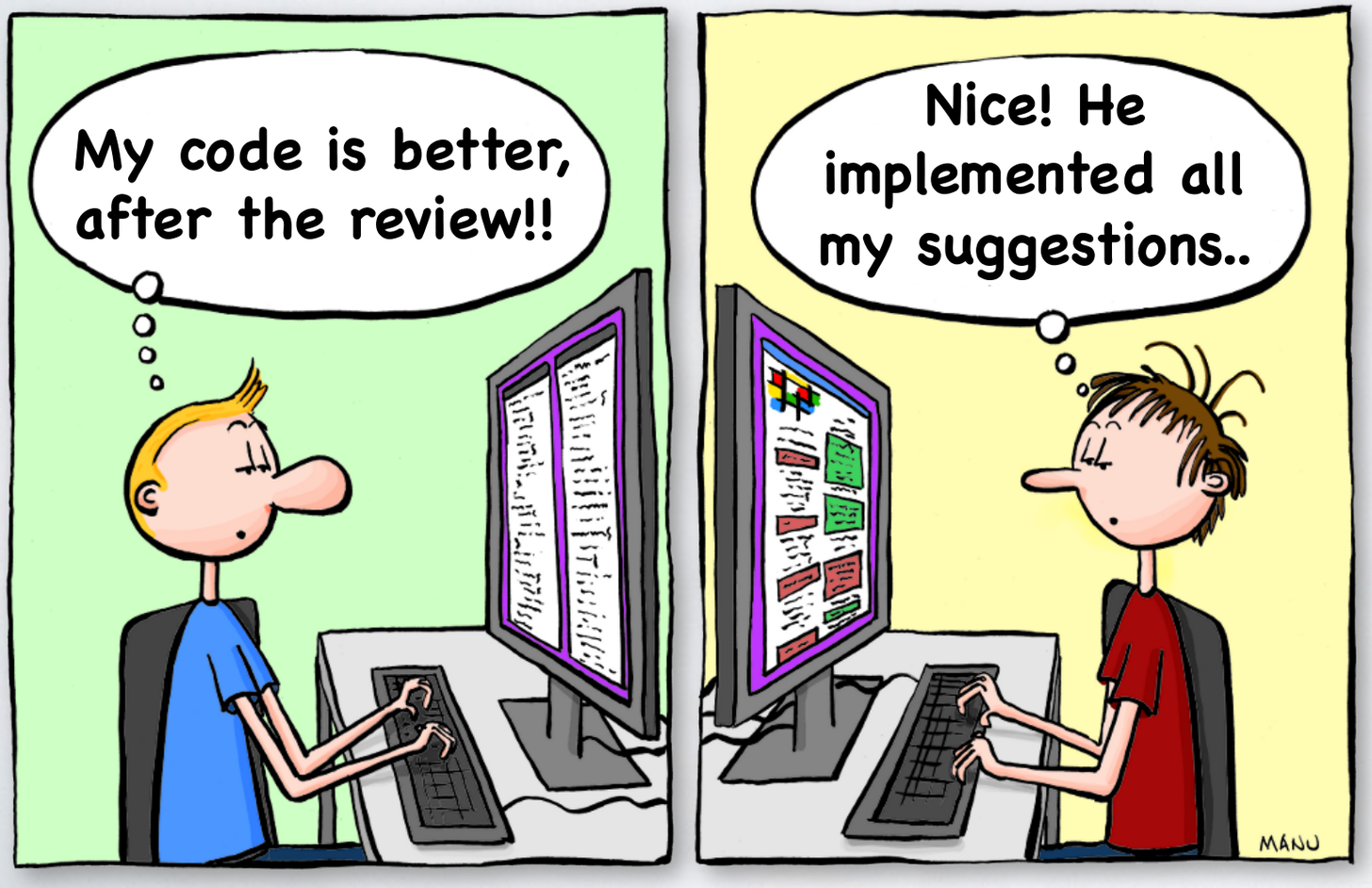
Alberto Bacchelli
University of Zurich,
Switzerland

Delft University of Technology,
The Netherlands

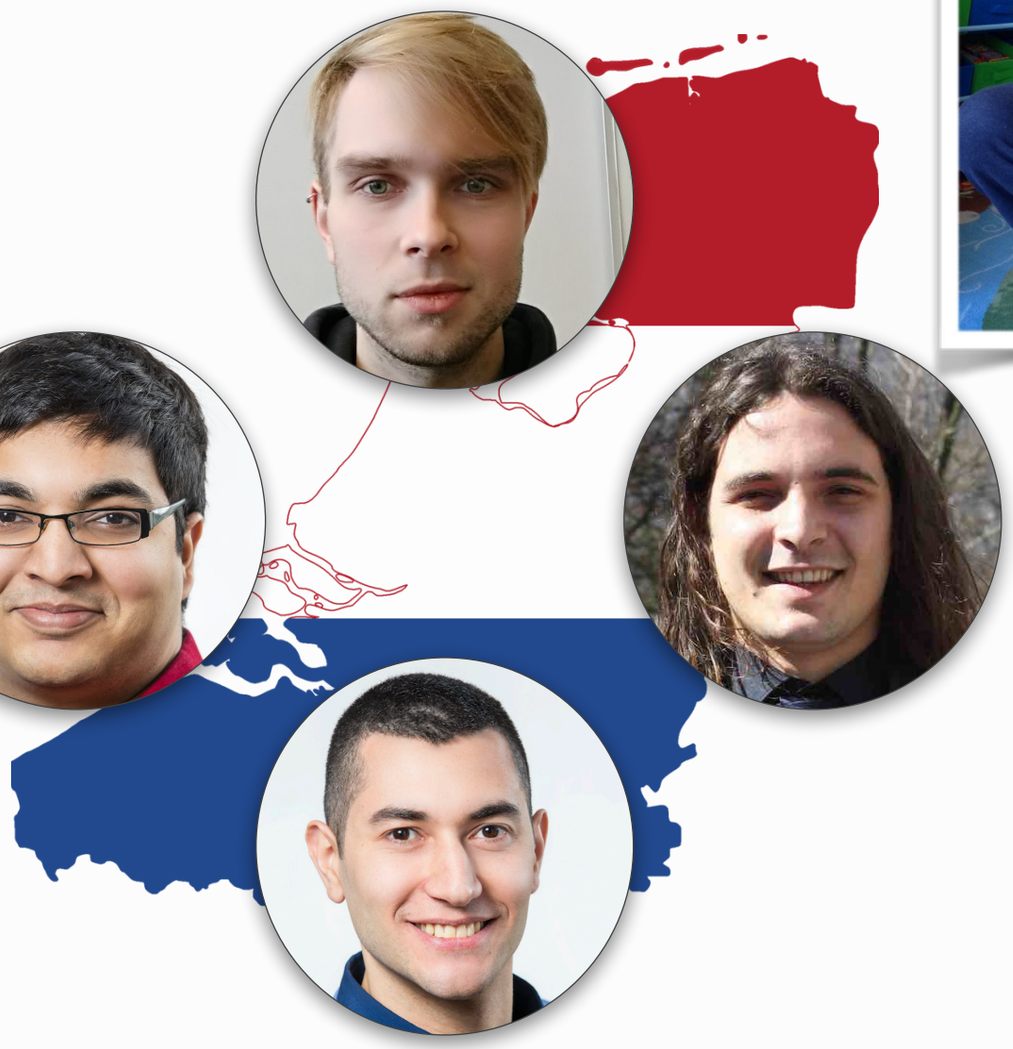
Evidence-based Code Review



My research team

SORCERERS

Delft University of Technology



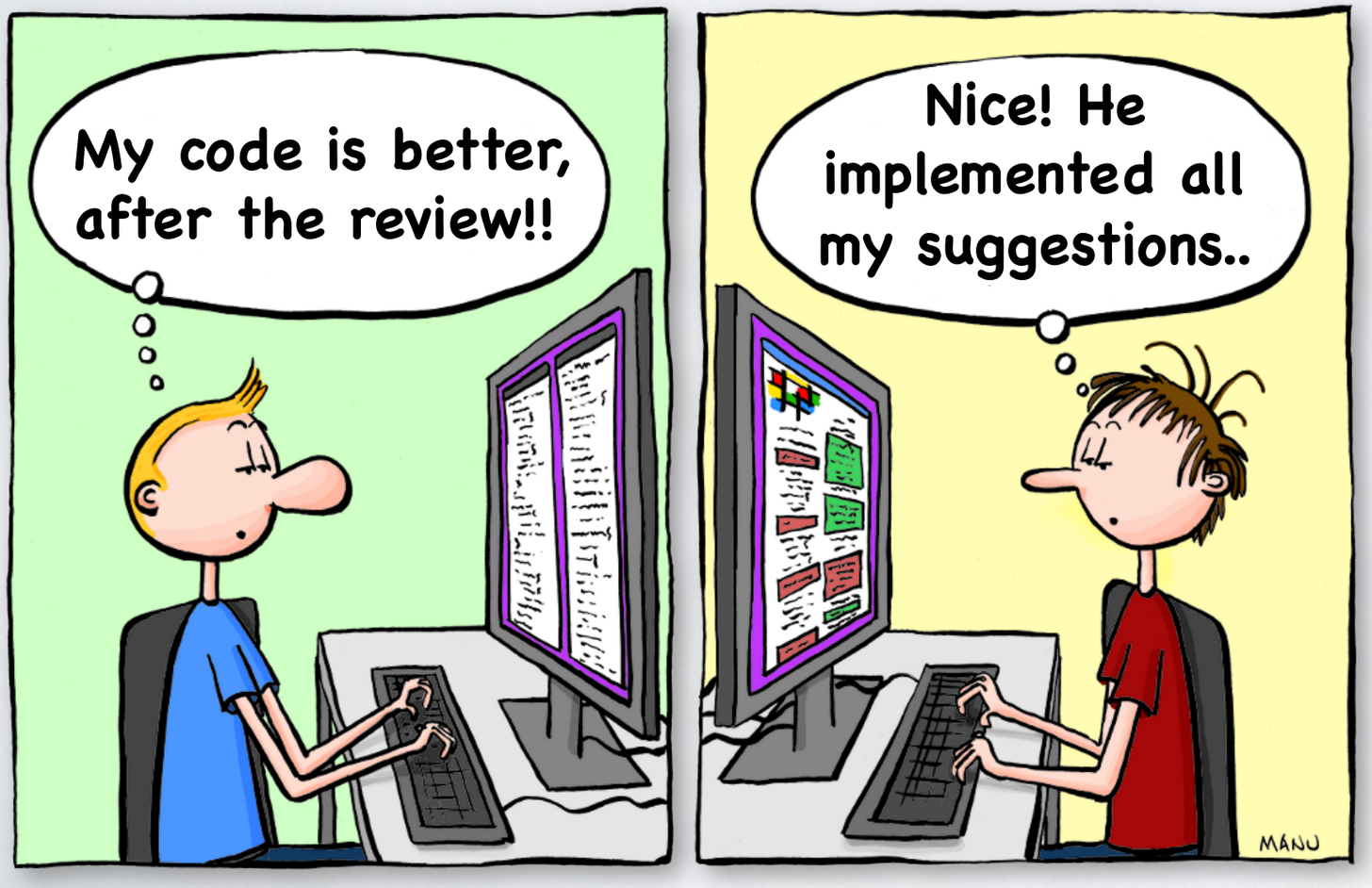
ZEST

University of Zurich

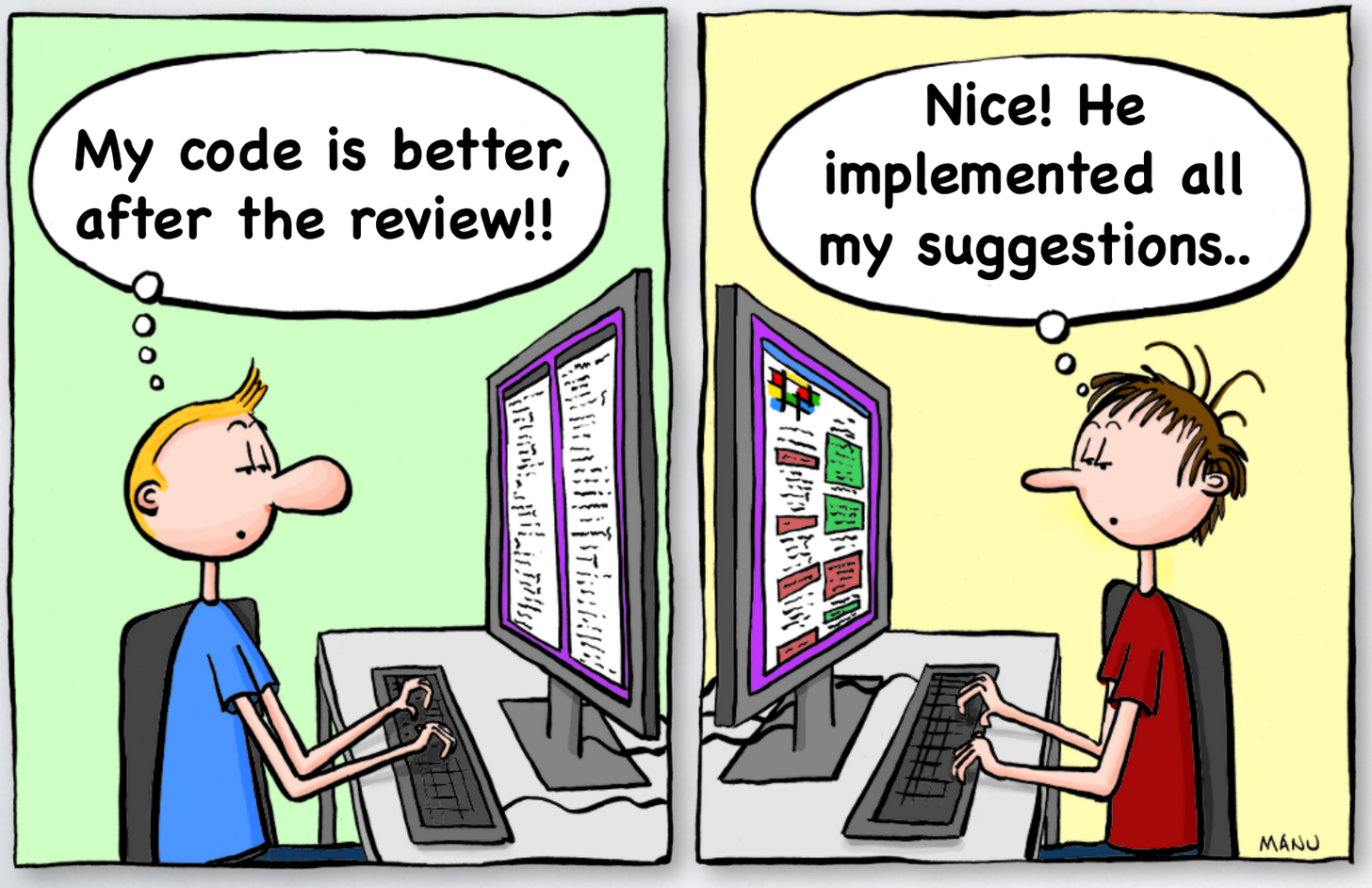
FNSNF
FONDS NATIONAL SUISSE
SCHWEIZERISCHER NATIONALFONDS
FONDO NAZIONALE SVIZZERO
SWISS NATIONAL SCIENCE FOUNDATION

…and awesome regular collaborators

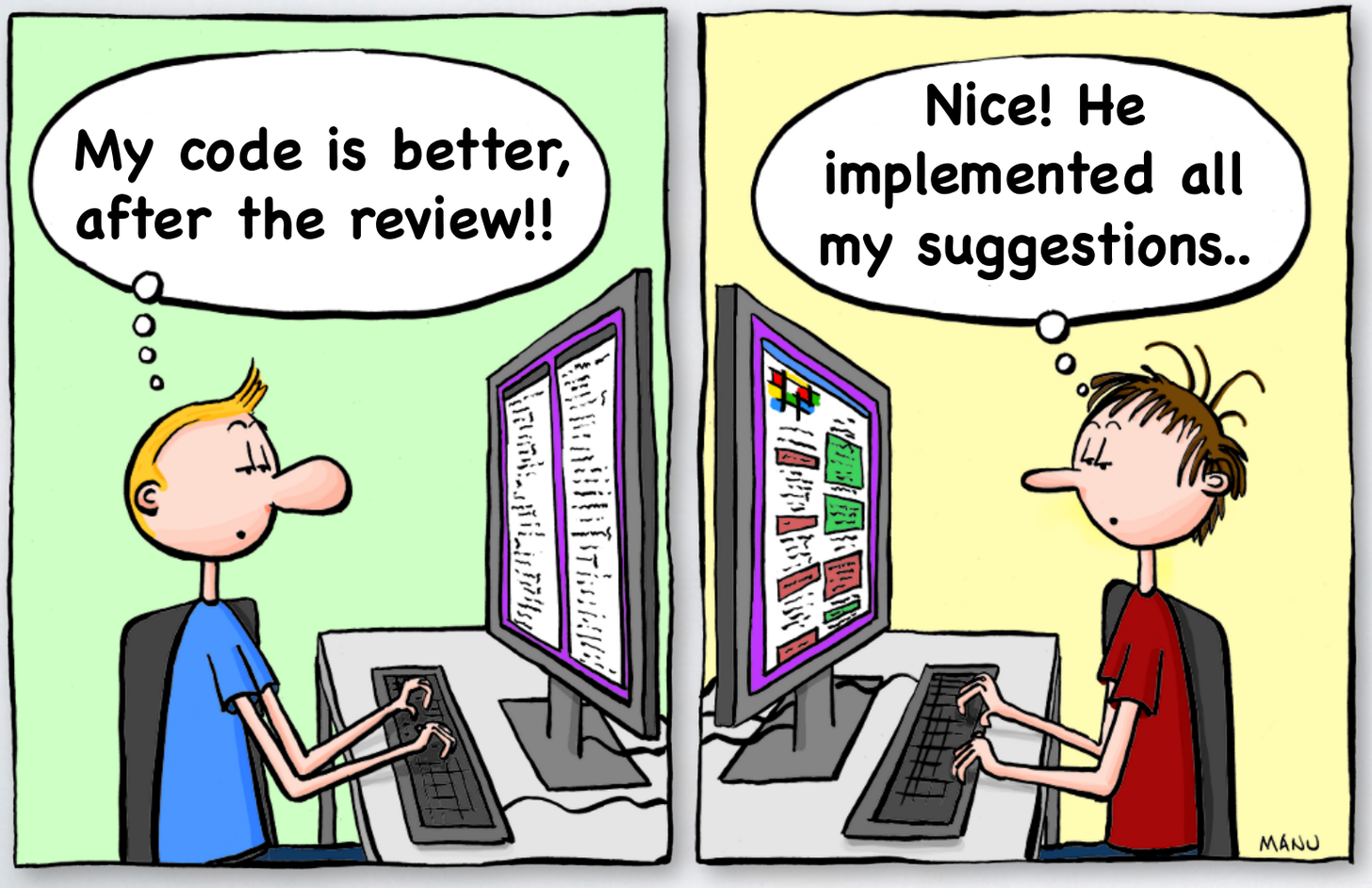
Evidence-based Code Review



Evidence-based
Modern Code Review



Evidence-based
Modern Code Review



1. informal

2. change-based

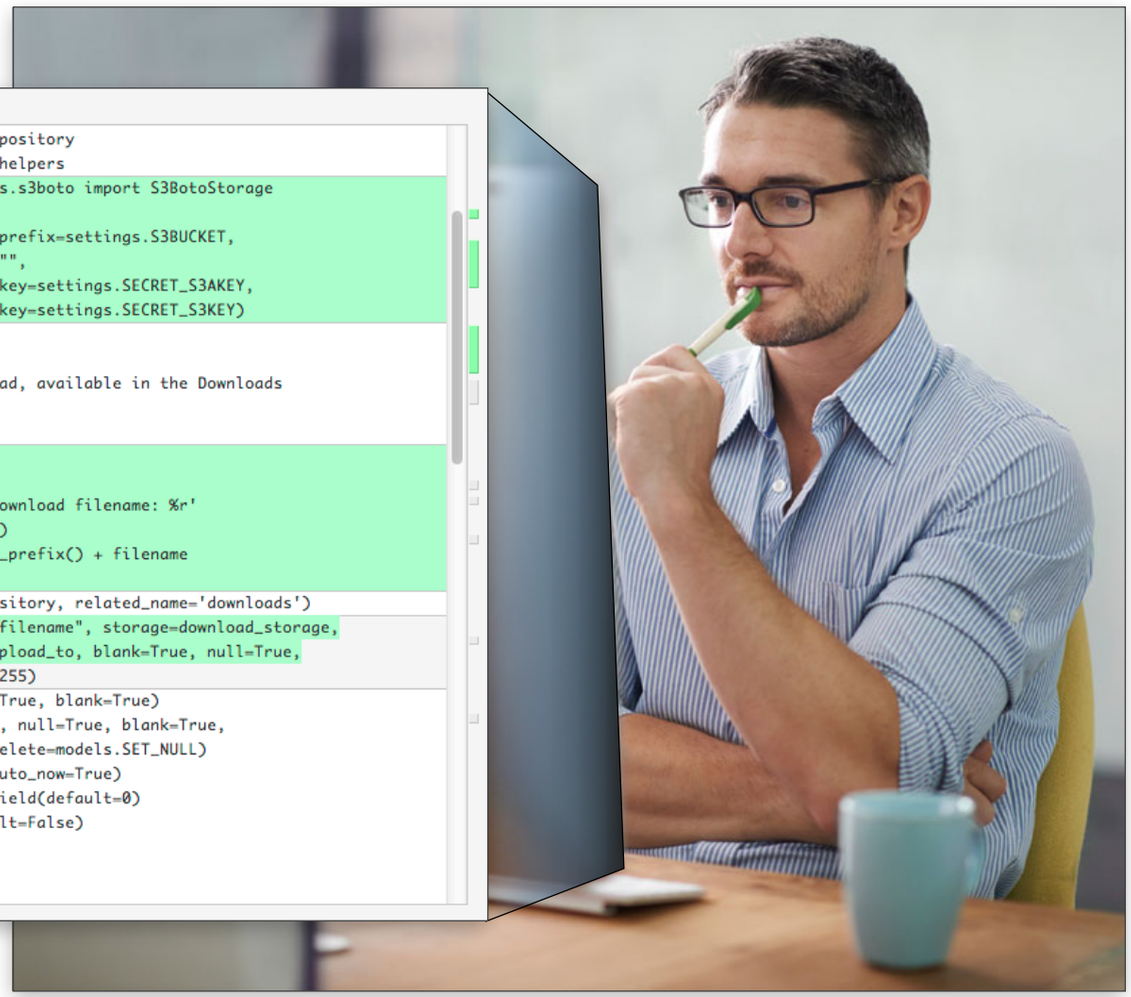
3. tool-based

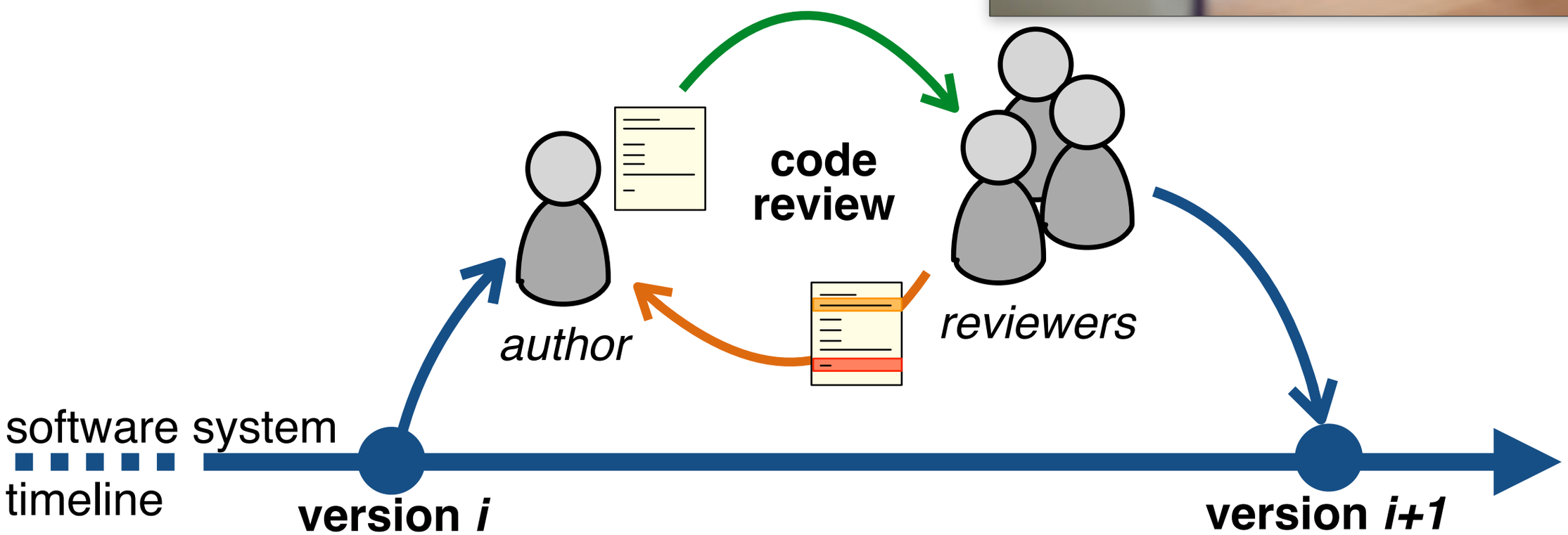
What is modern code review?

apps/downloads/models.py

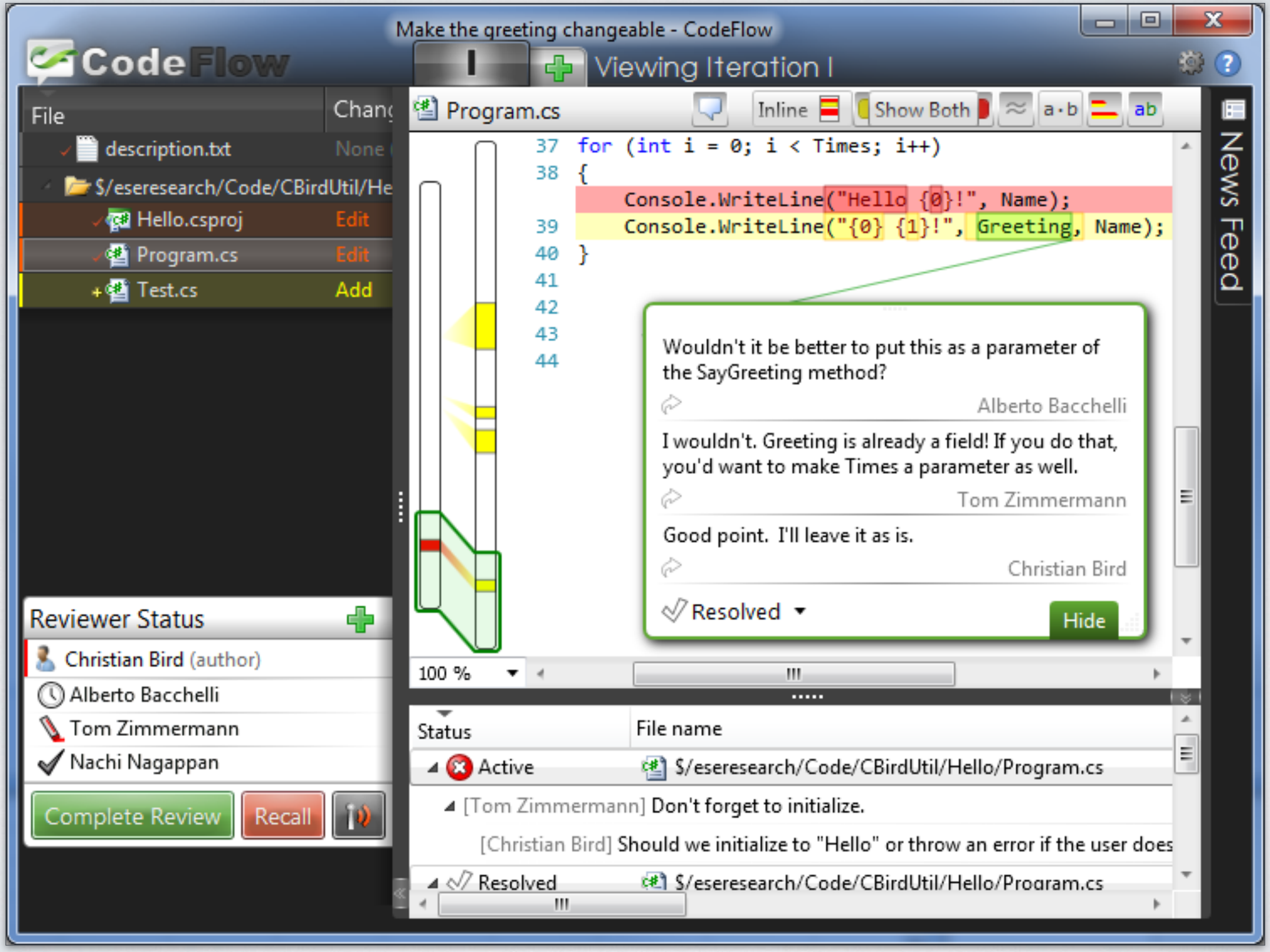
```
from django.db import models

from bitbucket.apps.bb.cname import cpermalink
from bitbucket.apps.bb.models import Repository
from bitbucket.apps.downloads import s3helpers

class Download(models.Model):
    """Represents a user-created download, available in the Downloads
    tab.
    """
    repository = models.ForeignKey(Repository, related_name='downloads')
    filename = models.CharField(max_length=255)
    size = models.BigIntegerField(null=True, blank=True)
    created_by = models.ForeignKey(User, null=True, blank=True,
                                   on_delete=models.SET_NULL)
    created_on = models.DateTimeField(auto_now=True)
    download_count = models.BigIntegerField(default=0)
    deleted = models.BooleanField(default=False)

    class Meta:
        app_label = 'bb'

    def _tokenized_url(self):
        """
        If a `Repository` is private, we can't use the public
        URL, as it will be restricted. This generates and caches
        tokens for 30 minutes. Used in `get_download_url`.
        """
        ck = 'downloadtoken:%d' % self.id
        cached = cache.get(ck)
        if not cached:
            key = s3helpers.get_key(self.filename)
```

```
from bitbucket.apps.bb.models import Repository
from bitbucket.apps.downloads import s3helpers
from bitbucket.apps.lib.storage.backends.s3boto import S3BotoStorage

download_storage = S3BotoStorage(bucketprefix=settings.S3BUCKET,
                                 bucket="",
                                 access_key=settings.SECRET_S3AKEY,
                                 secret_key=settings.SECRET_S3KEY)

class Download(models.Model):
    """Represents a user-created download, available in the Downloads
    tab.
    """
    def upload_to(self, filename):
        if not filename:
            raise ValueError('Invalid download filename: %r'
                             % filename)
        return self.repository.download_prefix() + filename

    repository = models.ForeignKey(Repository, related_name='downloads')
    file = models.FileField(db_column="filename", storage=download_storage,
                            upload_to=upload_to, blank=True, null=True,
                            max_length=255)
    size = models.BigIntegerField(null=True, blank=True)
    created_by = models.ForeignKey(User, null=True, blank=True,
                                   on_delete=models.SET_NULL)
    created_on = models.DateTimeField(auto_now=True)
    download_count = models.BigIntegerField(default=0)
    deleted = models.BooleanField(default=False)

    class Meta:
        app_label = 'bb'
```

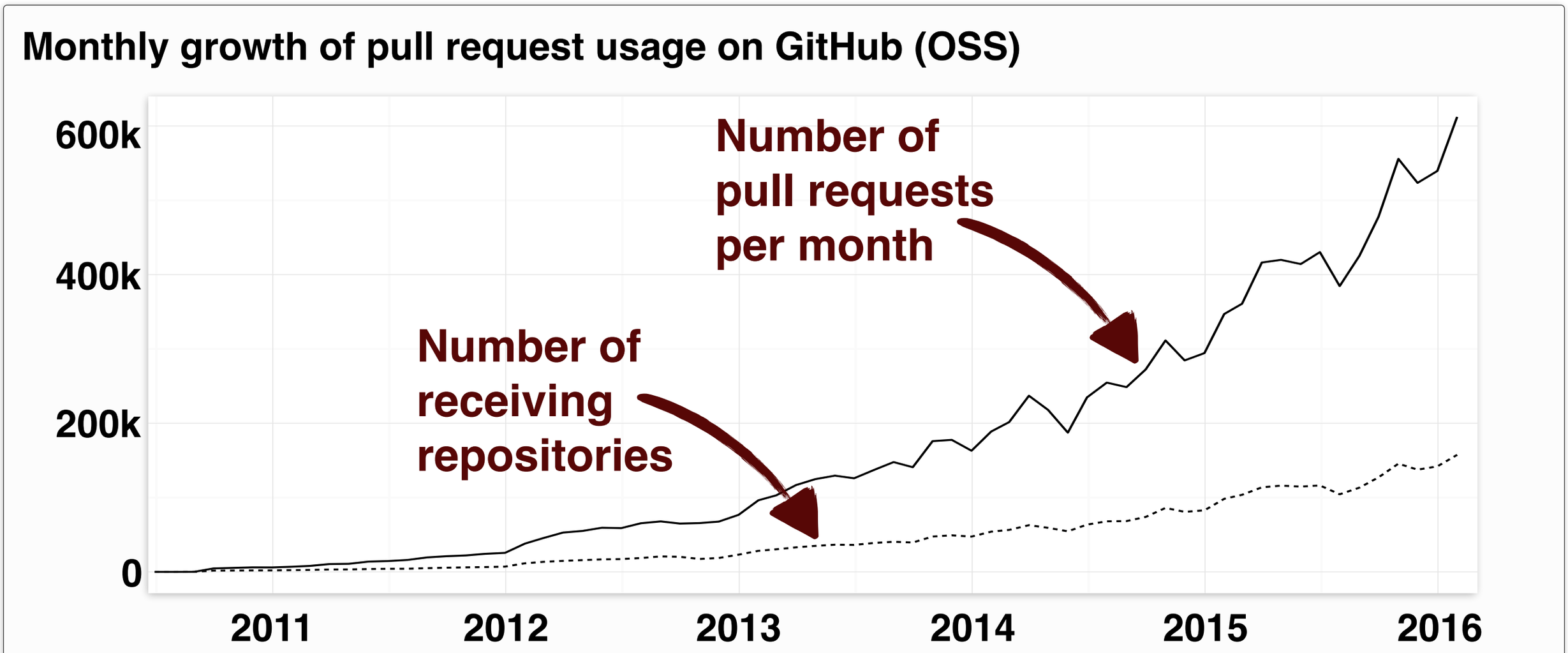




What is modern code review? A sample tool



How popular is it?



Popularity in industrial setting

Code review tool is used by more than 70,000 developers at Microsoft
[Czerwonka, Greiler, Tilford — ICSE 2016]

Each code change in the main repository at Google is reviewed
[Potvin, Levenberg — Communications of the ACM, 2016]

Most other companies have similar policies and big players then to develop their own review tools (e.g., Facebook)

Why code reviews? How? What is the outcome?

Let's look for "code review" on Google...

Google

code review

Sign in

All Images Videos News Maps More Settings Tools

About 230,000,000 results (0.45 seconds)

Code review is systematic examination (sometimes referred to as peer **review**) of computer source **code**. It is intended to find mistakes overlooked in software development, improving the overall quality of software.

Code review - Wikipedia
https://en.wikipedia.org/wiki/Code_review

About this result Feedback

People also ask

What is a security code review?

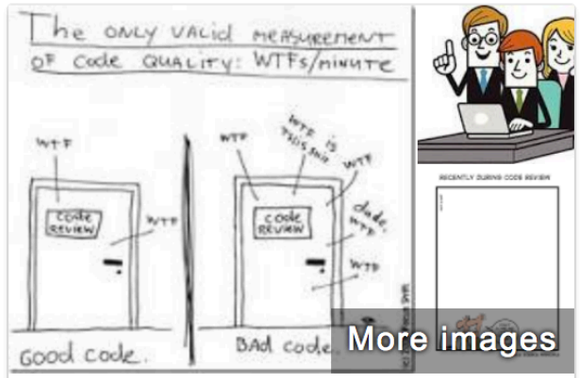
What is code review in software engineering?

What is the code inspection?

What do you mean by code coverage?

Feedback

Code review - Wikipedia
https://en.wikipedia.org/wiki/Code_review
Code review is systematic examination (sometimes referred to as peer **review**) of computer source **code**. It is intended to find mistakes overlooked in software development, improving the overall quality of software.
Introduction · Types



Code review

Code review is systematic examination of computer source code. It is intended to find mistakes overlooked in software development, improving the overall quality of software. Wikipedia

Feedback

Wikipedia page on code review — a stub

Code review

From Wikipedia, the free encyclopedia

This article **is written like a personal reflection or opinion essay** that states a Wikipedia editor's personal feelings about a topic. Please help improve it by rewriting it in an encyclopedic style. *(October 2017) (Learn how and when to remove this template message)*

Code review is systematic examination (sometimes referred to as peer review) of computer source code. It is intended to find mistakes overlooked in software development, improving the overall quality of software. Reviews are done in various forms such as pair programming, informal walkthroughs, and formal inspections.[1]

Contents [hide]

1 Introduction
1.1 Simple Definition
1.2 Artifacts
1.3 Details
2 Types
3 Criticism
4 See also
5 References
6 Further reading
7 External links

Introduction [edit]

Simple Definition [edit]

A code review is a process where two or more developers visually inspect a set of program code, typically, several times. The code can be a method, a class, or an entire program. The main code-review objectives are:

1. Best Practice ~ A more efficient, less error-prone, or more elegant way to accomplish a given task.

Software development

Core activities
Processes · Requirements · Design · Engineering · Construction · Testing · Debugging · Deployment · Maintenance

Paradigms and models
Agile · Cleanroom · Incremental · Prototyping · Spiral · Waterfall

Methodologies and frameworks
DevOps · DSDM · FDD · IID · Kanban · Lean SD · MDD · MSF · PSP · RAD · SAFe · Scrum · SEMAT · TSP · UP · V-Model · XP

Supporting disciplines
Configuration management · Documentation · Software quality assurance (SQA) · Project management · User experience

Practices
ATDD · BDD · CCO · CI · CD · DDD · PP · Stand-up · TDD

Tools
Compiler · Debugger · Profiler · GUI designer · Modeling · IDE · Build automation · Release automation · Infrastructure as Code · Testing

Standards and BOKs
CMMI · IEEE standards · ISO 9001 · ISO/IEC standards · SWEBOK · PMBOK · BABOK

V · T · E

Wikipedia page on code review — an embarrassing stub

The **main code-review objectives** are (1) **best practice**, (2) **error detection**, (3) **vulnerability exposure**, (4) **malware discovery** […]. Of the four objectives, **malware is the only one that requires human detection**.

Code **reviews can often find and remove common vulnerabilities** such as format string exploits, race conditions, memory leaks and buffer overflows, **thereby improving software security**.

Lightweight code review typically requires **less overhead than formal code inspections**, though it can be **equally effective when done properly**.[citation needed]

I love Wikipedia

several times. The code can be a method, a class, or an

are:

1. Best Practice ~ A more efficient, less error-prone, or more elegant way to accomplish a given task.

Code review “best practices”

Let’s look for “code review best practices” on Google…

11 proven practices for more effective, efficient peer code review - IBM
https://www.ibm.com › Learn › Rational ▾
25 Jan 2011 - Aim for an inspection rate of fewer than 300–500 LOC per hour. Establish quantifiable goals for **code review**, and capture metrics so you can improve your processes. Verify that the defects are actually fixed. Foster a **good code review** culture in which finding defects is viewed positively. Beware of the Big Brother ...

Code Review Best Practices - Kevin London's blog
kevinlondon.com/2015/05/05/code-review-best-practices.html ▾
5 May 2015 - I think it's a **good** idea to crystalize some of the things I look for when I'm doing **code reviews** and talk about the **best** way I've found to approach ...

Best practices for effective code reviews - WillowTree Apps
https://willowtreeapps.com/ideas/best-practices-for-effective-code-reviews ▾
27 Oct 2016 - Today, I'd like to share our process and some **best practices** we follow when conducting **code reviews**. The process. The **code review** process ...

Effective Code Reviews: Code Review Best Practices
https://nyu-cds.github.io/effective-code-reviews/02-best-practices/ ▾
What are some **best practices** for **code reviews**? Objectives. Learn about effective practices for **code reviews**. Learn what makes reviews work better and what ...

7 best practices for doing code reviews - The Asana Blog
https://blog.asana.com/2016/12/7-ways-to-uplevel-your-code-review-skills/ ▾
20 Dec 2016 - The Asana engineering team shares **code review best practices** that will help you become a better reviewer. Learn how Asana reviews code.

Code Review in Agile Teams - part II - Atlassian Blog
https://www.atlassian.com/blog/archives/code_review_in_agile_teams_part_ii ▾
8 Mar 2010 - Ready to try adopting **code review** within your team or across your .... reveal a few **best practices** around **code review** that evolved at Atlassian.

Best Practices: Code Reviews - MSDN - Microsoft
https://msdn.microsoft.com/en-us/library/bb871031.aspx ▾

Code review “best practices”

Two examples

11 PROVEN PRACTICES FOR MORE EFFECTIVE, EFFICIENT CODE REVIEW

- Review **fewer than 200–400 lines** of code at a time
- Aim for an inspection rate of **fewer than 300–500 LOC per hour**
- Take enough time for a proper, slow review, but **not more than 60–90 minutes**
- Be sure that **authors annotate source code before the review** begins
- Establish **quantifiable goals** […] and capture metrics [to] improve your processes
- **Use checklists**, because they substantially improve results
- **Verify** that the defects are actually fixed
- Foster a **good code review culture** in which finding defects is viewed positively
- Beware of the **Big Brother effect**
- **Review at least part of the code**, even if you can't do all of it, [for] The Ego Effect
- **Adopt lightweight, tool-assisted code reviews**

7 WAYS TO UPLEVEL YOUR CODE REVIEW SKILLS

- **Prioritize the goals** of code reviews with your team
- **Run the app** and try playing with the feature
- Visualize **method call hierarchies**
- Do code reviews **as soon as you see the request**
- **Imagine** how you would make this change before you read it
- Read the change in a **realistic development environment**
- **Always give approval**, unless you can prove that there is a bug

Code review “best practices”

Two examples

11 PROVEN PRACTICES FOR MORE EFFECTIVE, EFFICIENT CODE REVIEW

- Review **fewer than 200–400 lines** of code at a time
- **Aim for an inspection rate of fewer than 300–500 LOC per hour**
- Take enough time for a proper, slow review, but **not more than 60–90 minutes**
- Be sure that **authors annotate source code before the review** begins
- Establish **quantifiable goals** […] and capture metrics [to] improve your processes
- **Use checklists, because they substantially improve results**
- **Verify** that the defects are actually fixed
- Foster a **good code review culture** in which finding defects is viewed positively
- Beware of the **Big Brother effect**
- **Review at least part of the code**, even if you can't do all of it, [for] The Ego Effect
- **Adopt lightweight, tool-assisted code reviews**

7 WAYS TO UPLEVEL YOUR CODE REVIEW SKILLS

- **Prioritize the goals** of code reviews with your team
- **Run the app** and try playing with the feature
- **Visualize method call hierarchies**
- **Do code reviews as soon as you see the request**
- **Imagine** how you would make this change before you read it
- Read the change in a **realistic development environment**
- **Always give approval**, unless you can prove that there is a bug

Code review “best pra very skeptical

Two examples

11 PROVEN PRACTI CODE REVIEW

- Review **fewer than 200**
- **Aim for an inspection**
- Take enough time for a **minutes**
- Be sure that **authors** s
- Establish **quantifiabl** r processes
- **Use checklists, bec**
- **Verify** that the defec
- Foster a **good code** d positively
- Beware of the **Big**
- **Review at least pa** The Ego Effect
- **Adopt lightweigh**

7 WAYS TO UP

- **Prioritize the go**
- **Run the app** an
- **Visualize meth**
- **Do code revie**
- **Imagine** how
- Read the char
- **Always give appro** ug

In-field study on modern code review at Microsoft

Dr. Christian Bird

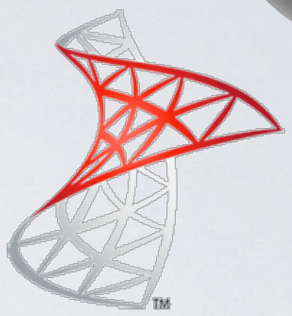
Microsoft setting

Microsoft setting: Several different separate “software organizations”

Same code review tool

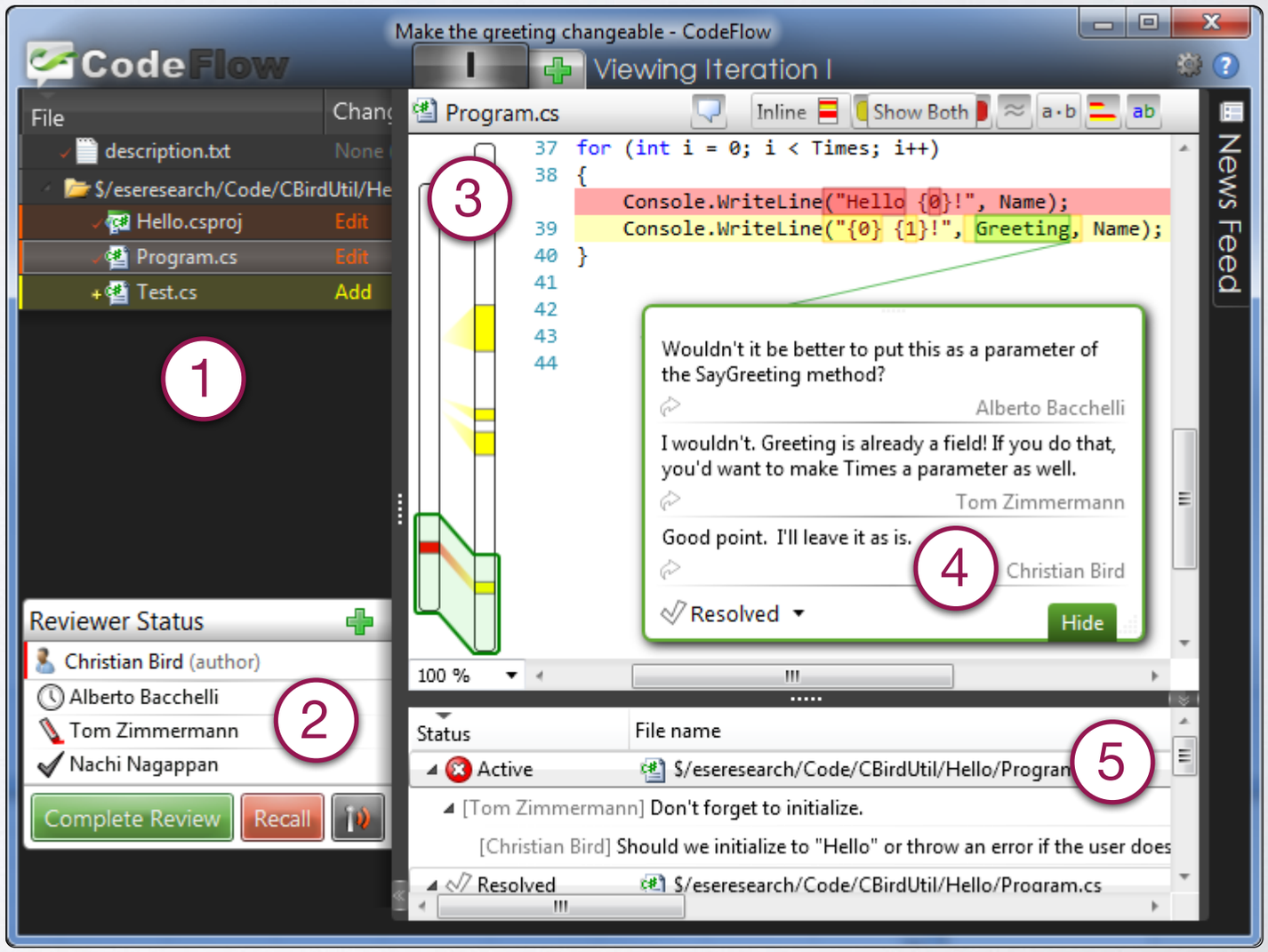
Excel

XBox



SQL Server

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



observations

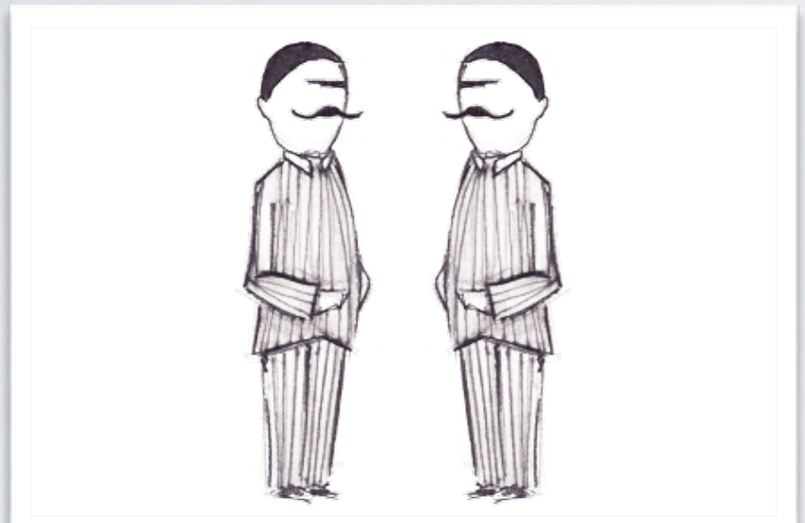
interviews

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


Shuttle Connect
SIENNA LE AWD
Microsoft
363
Shuttle
Connect
363

observations

interviews



survey to 165 managers

Why do programmers do code reviews?

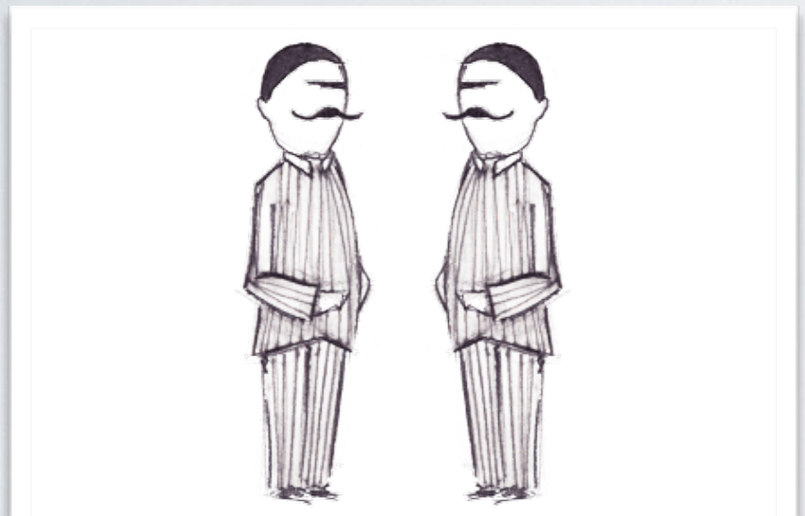
motivation

Why do programmers do code reviews? — List of Motivations

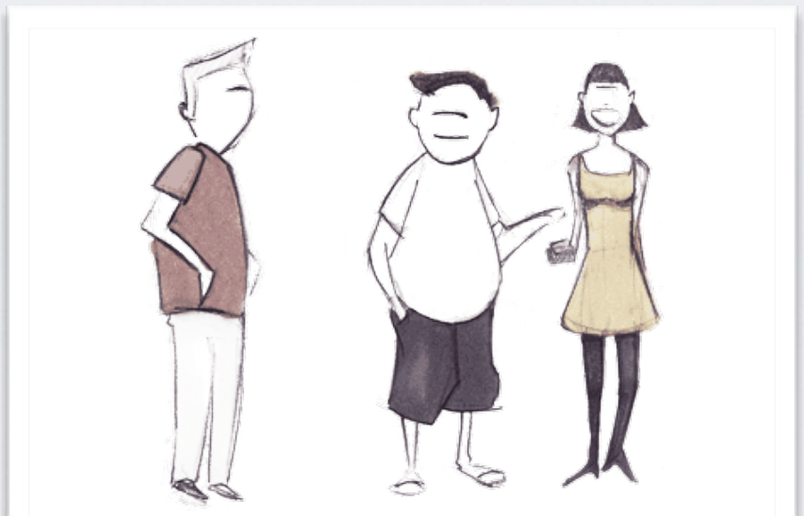
- Alternative Solutions
- Avoid Build Breaks
- Code Improvement
- Team Assessment
- Share Code Ownership
- Team Awareness
- Knowledge Transfer
- Improve Dev. Process
- Track Rationale
- Finding Defects

observations

interviews

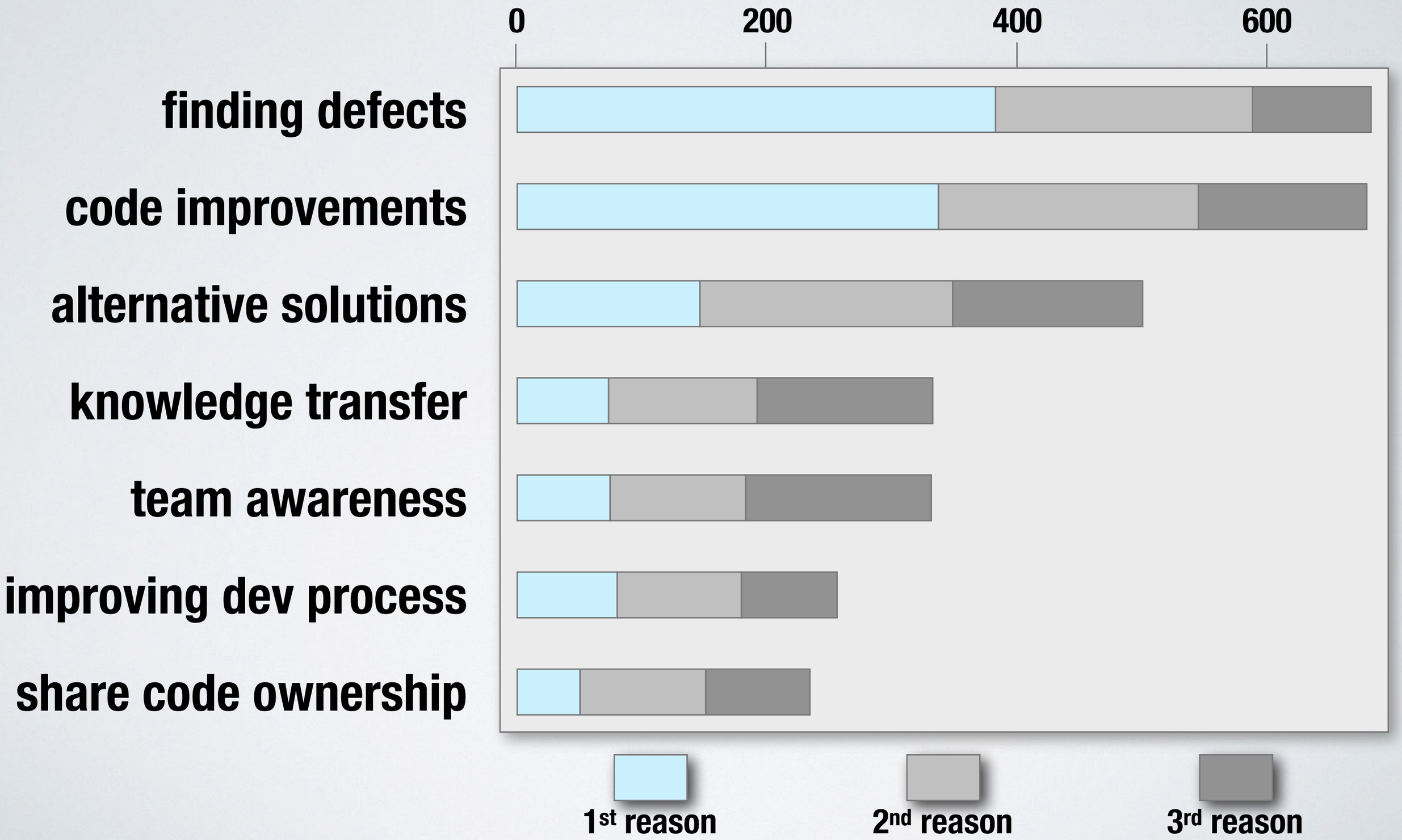


survey to 165 managers



survey to 873 developers

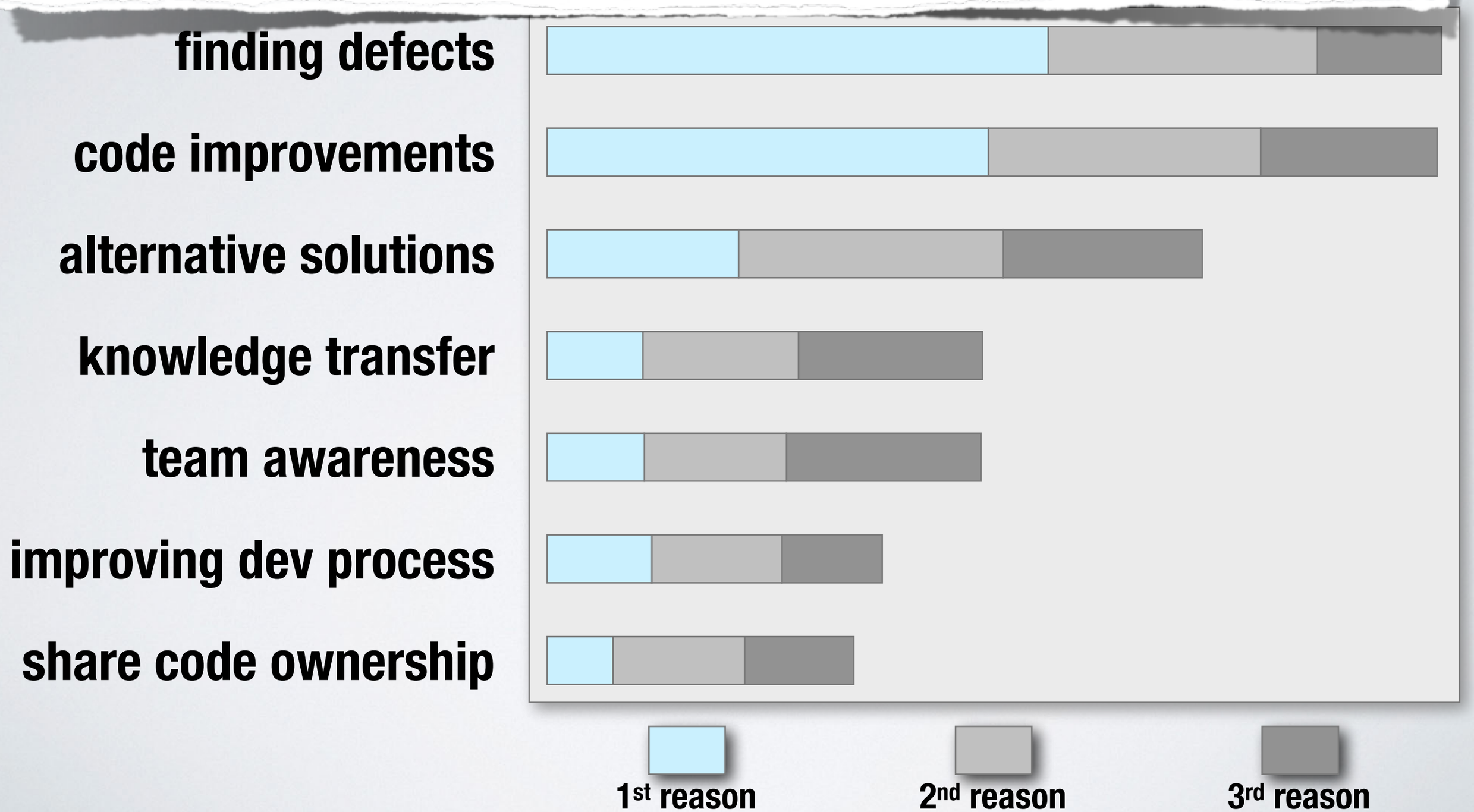
Why do programmers do code reviews? — Motivations' ranking



Why do programmers do code reviews? — Motivations' ranking

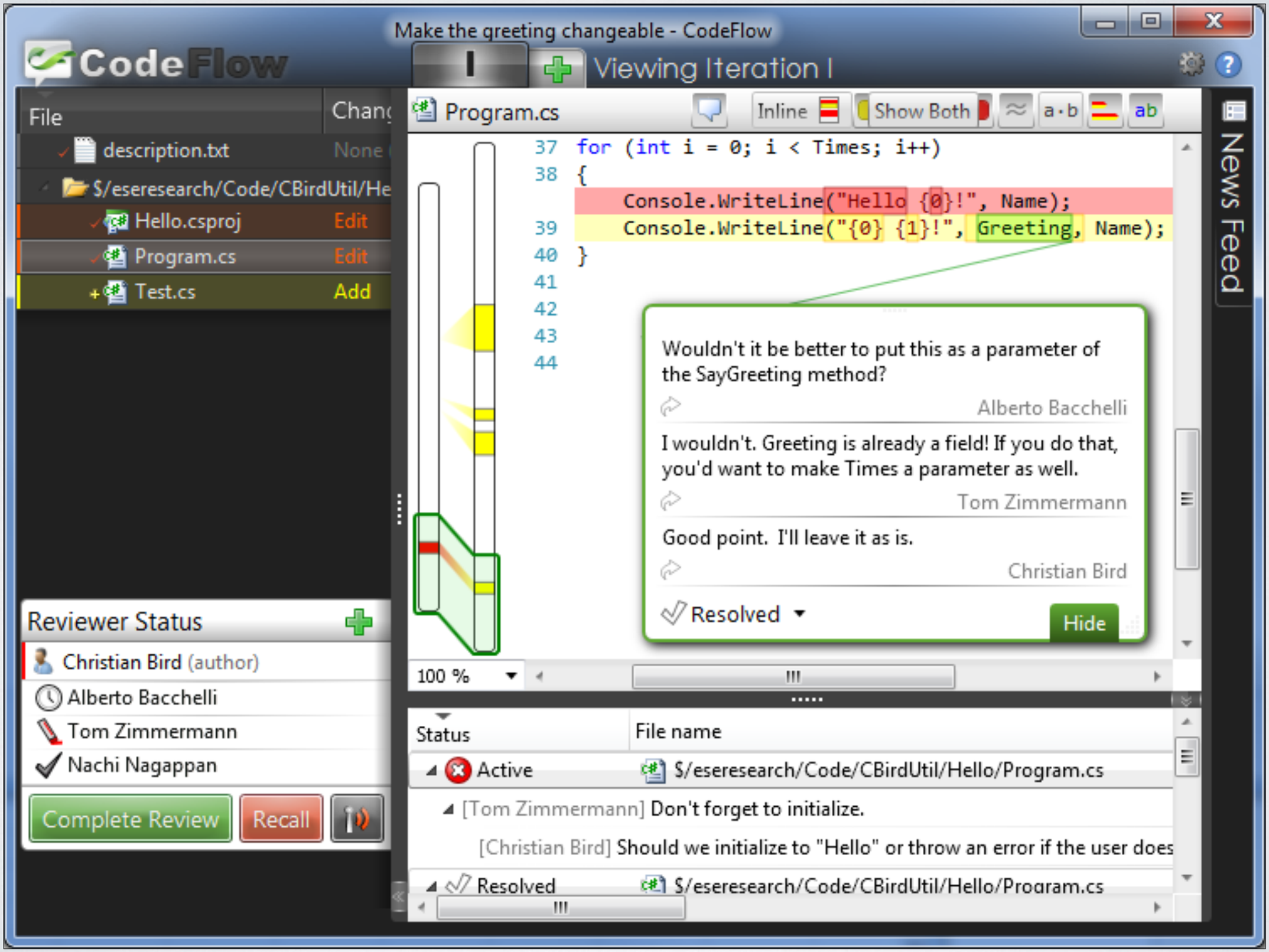
"Finding defects is the main reason for doing code review."

72 managers and 384 developers @ Microsoft

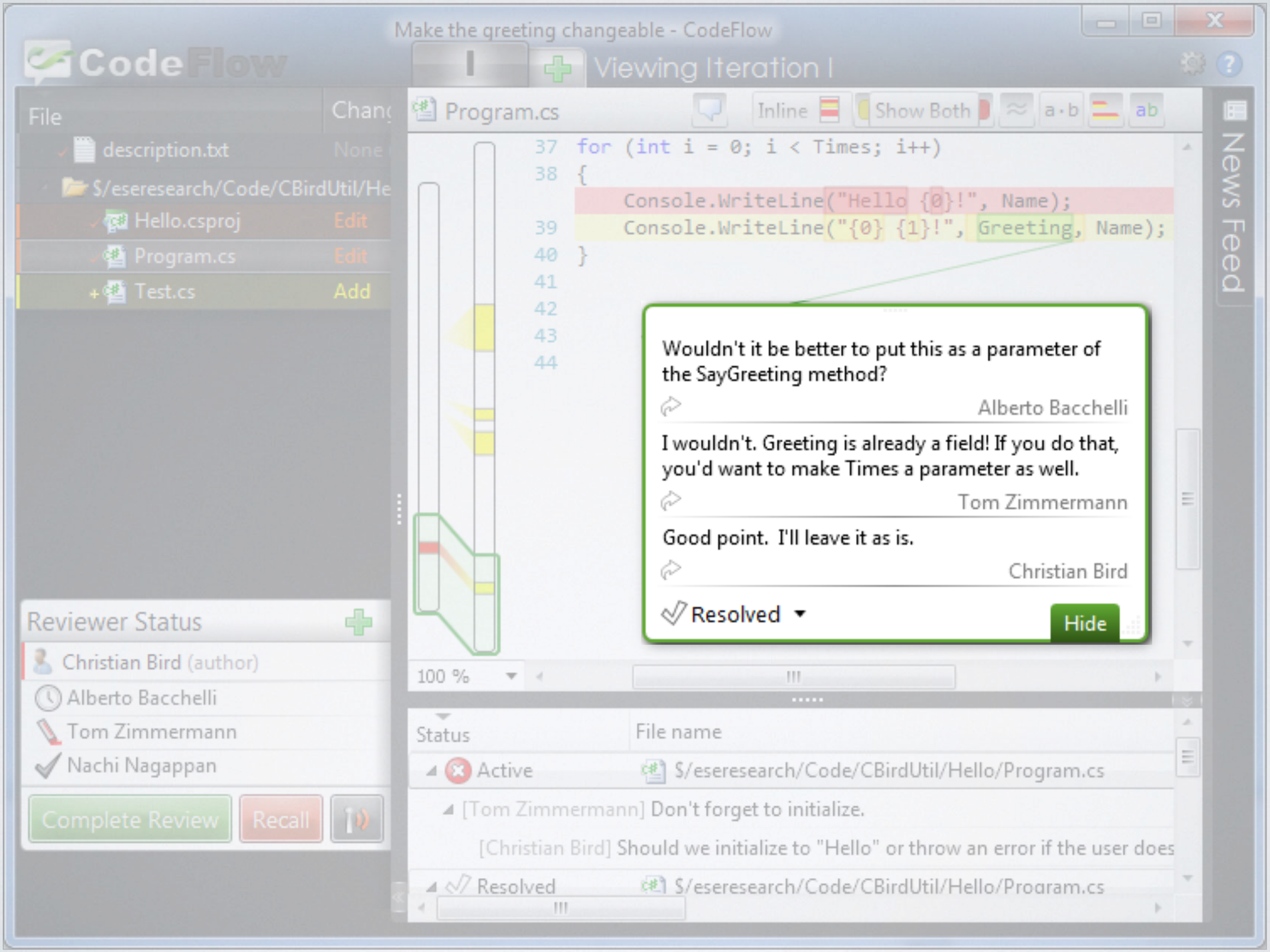


What is the outcome of code reviews?

What is the outcome of code reviews?

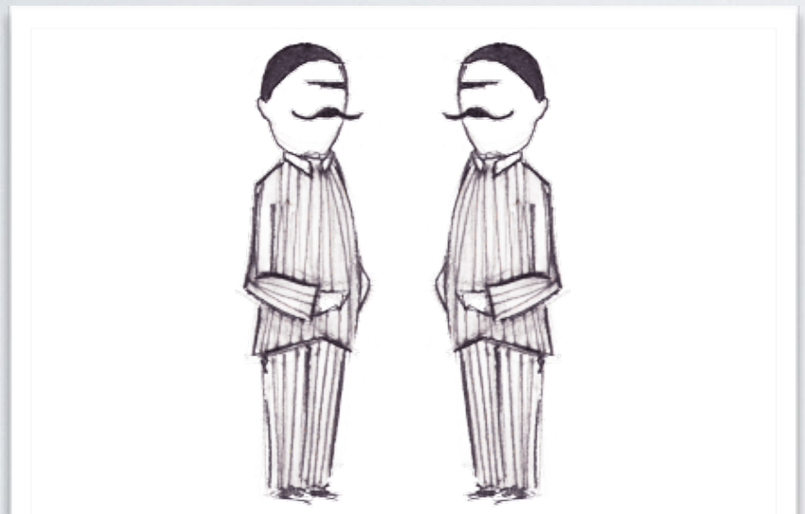


Recorded code review comments

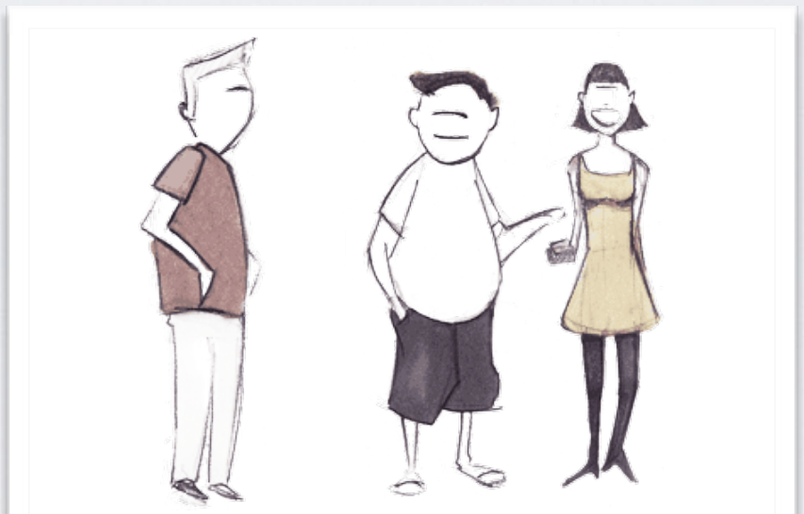


observations

interviews



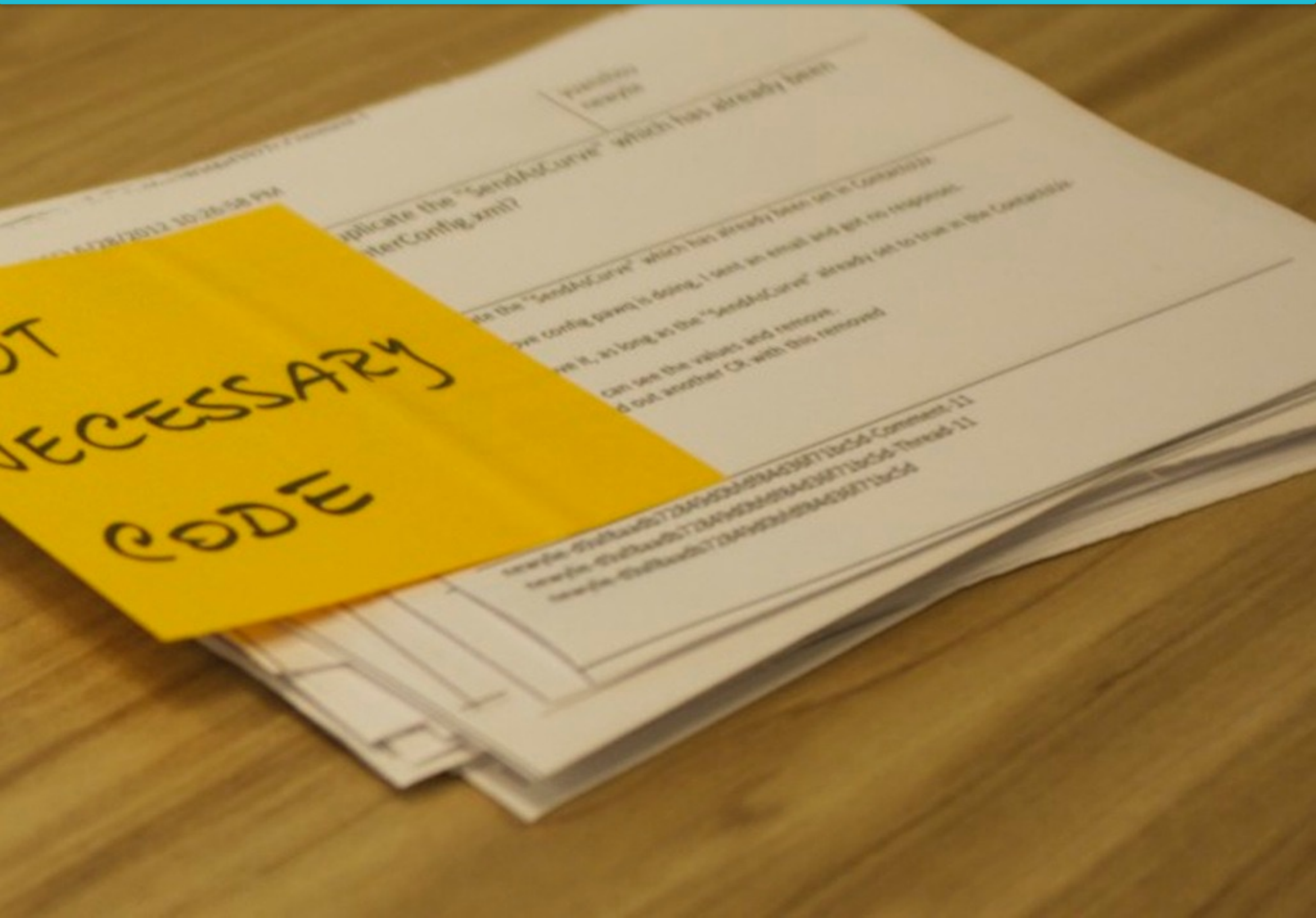
survey to 165 managers



survey to 873 developers

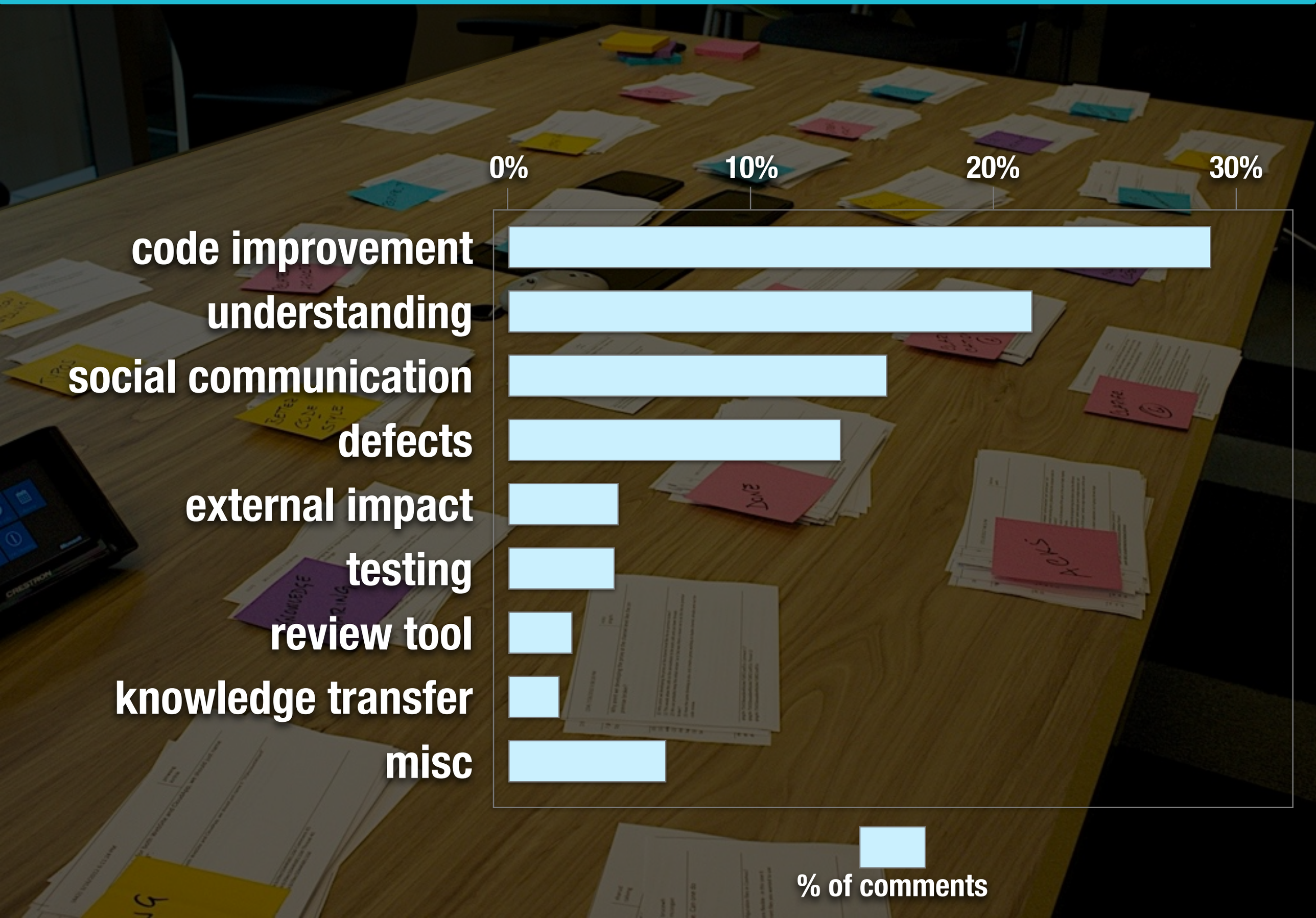
classification of
570 review comments

Example stack of cards

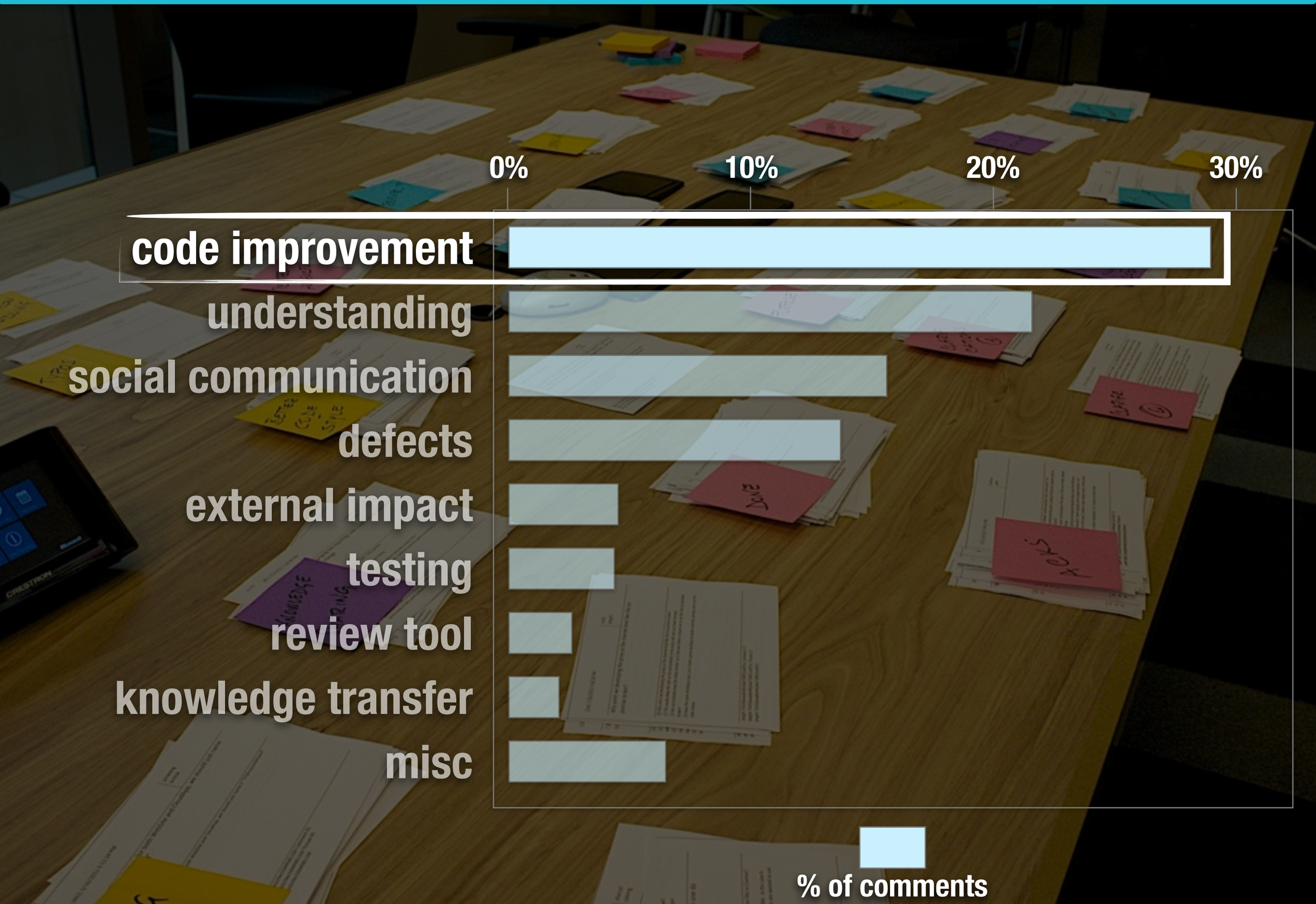


Card sort almost completed

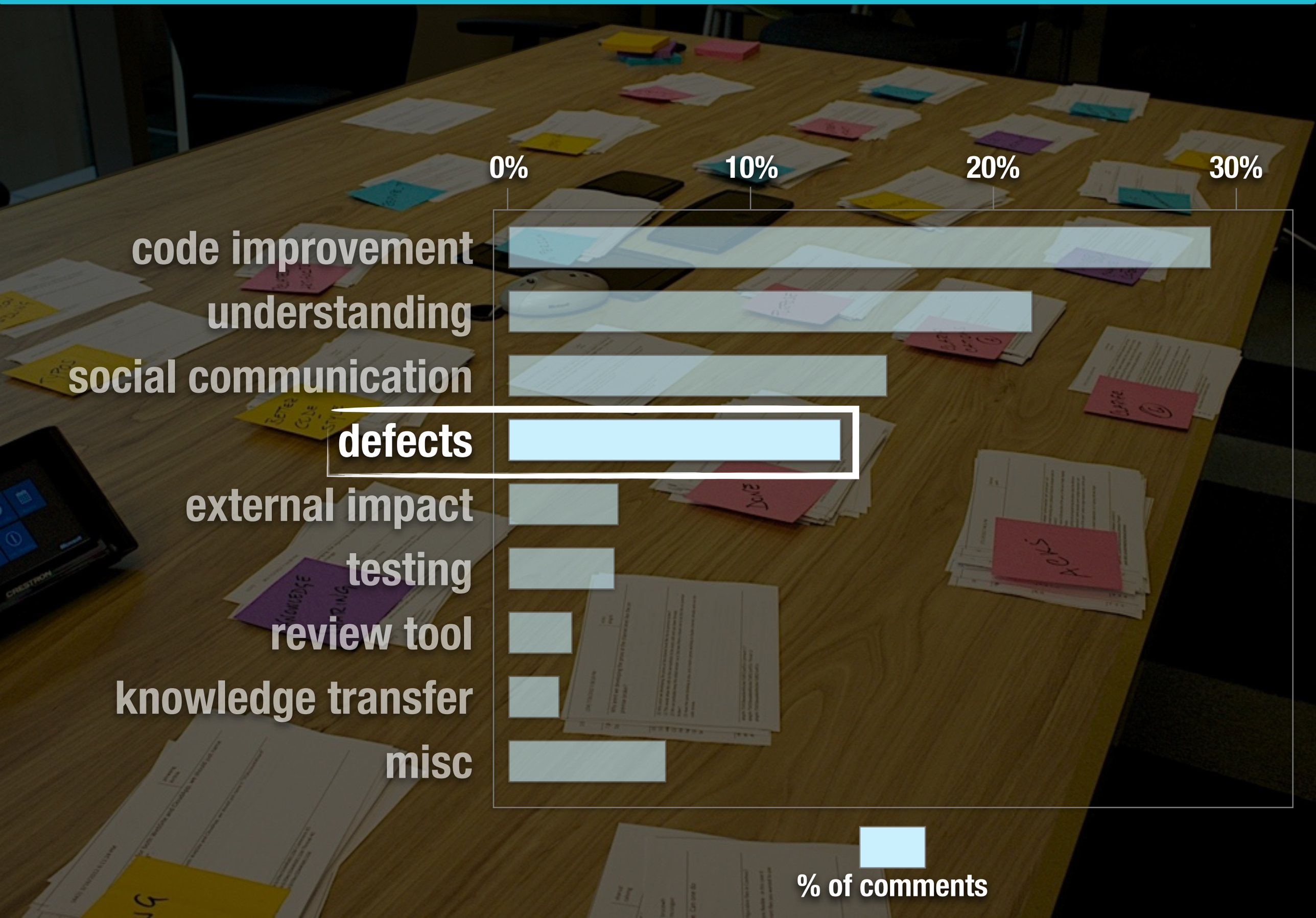
Card sort results



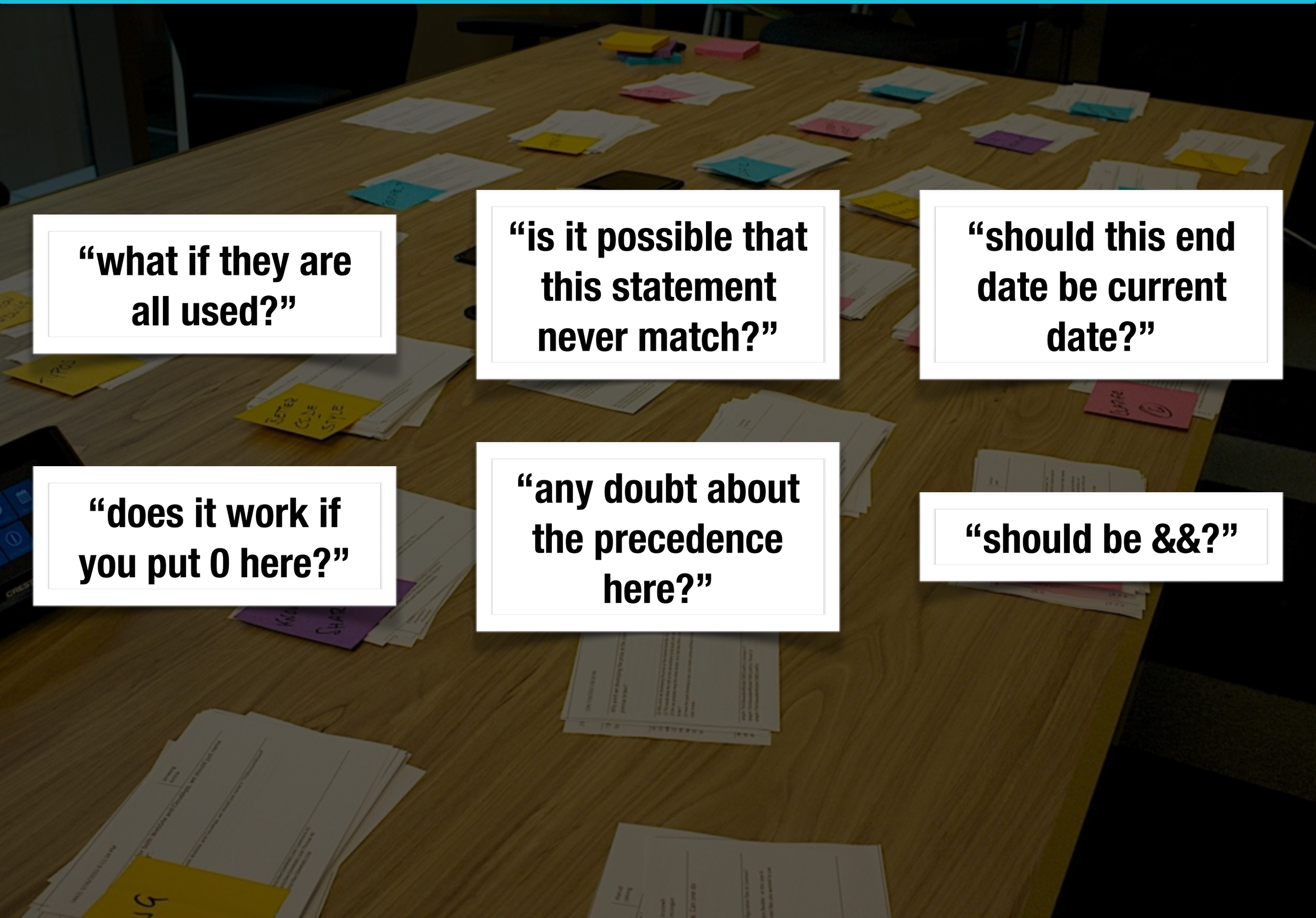
Card sort results



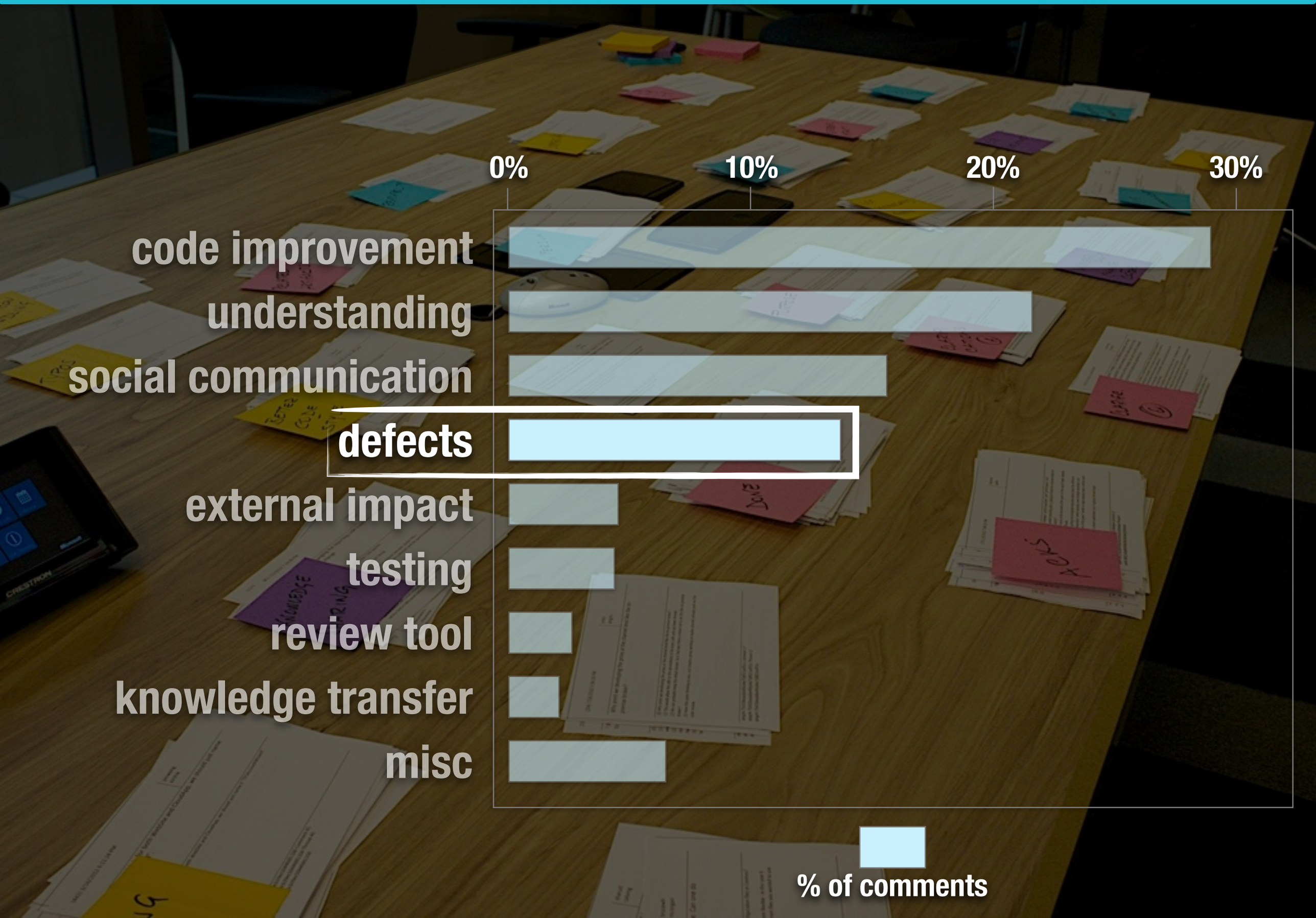
Card sort results



Card sort results **defects**



Card sort results



What is the outcome of code reviews? Reality ≠ expectations

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

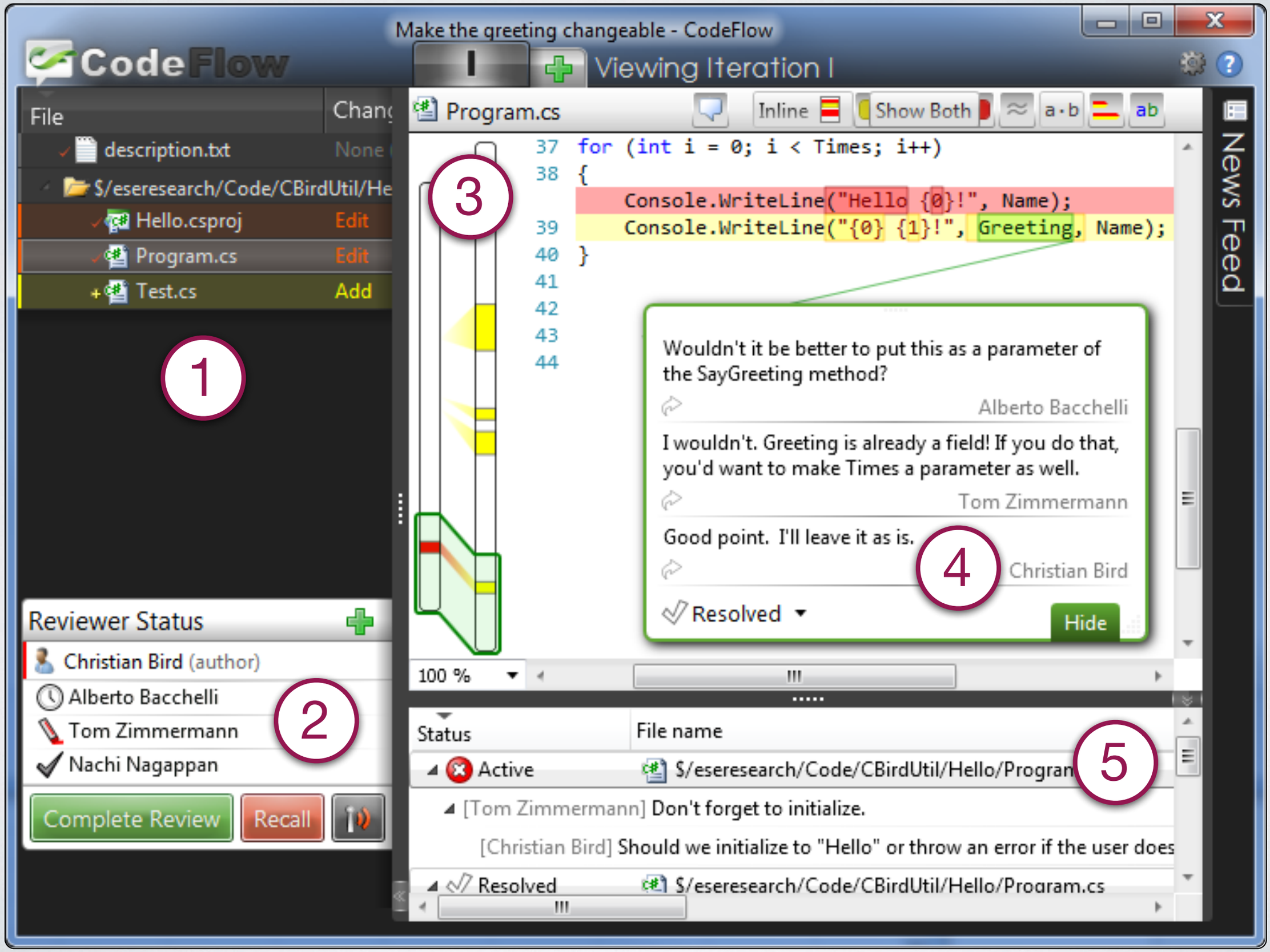
Code reviews

… “**[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



Code review “best practices”

Two examples

11 PROVEN PRACTICES FOR MORE EFFECTIVE, EFFICIENT CODE REVIEW

- Review **fewer than 200–400 lines** of code at a time
- **Aim for**
- Take en
- Be sure
- Establis
- **Use che**
- **Verify** th
- Foster a
- Beware
- **Review**
- **Adopt li**

7 WAYS

- **Prioritiz**
- **Run the**
- **Visualiz**
- **Do code reviews as soon as you see the request**
- **Imagine** how you would make this change before you read it
- Read the change in a **realistic development environment**
- **Always give approval**, unless you can prove that there is a bug

Call to Arms — Evidence-based modern code review

Research vision

**Transform code review into an engineering process,
to tap into its potential to develop high-quality software.**

What is an engineering process?

Call to Arms — Evidence-based modern code review

Research vision

**Transform code review into an engineering process,
to tap into its potential to develop high-quality software.**

What is an engineering process?

“Engineering enables ordinary people to do things that formerly required virtuosos.”

Mary Shaw (US National Medal of Technology and Innovation) — 2015

Call to Arms — Evidence-based modern code review

Research vision

**Transform code review into an engineering process,
to tap into its potential to develop high-quality software.**

What is an engineering process?

“Engineering enables ordinary people to do things
that formerly required virtuosos.”

“It creates cost-effective solutions…
…to practical problems…
…by applying scientific knowledge…
…building things…
…in the service of mankind.”

Mary Shaw (US National Medal of Technology and Innovation) — 2015

Call to Arms — Evidence-based modern code review

Research vision

**Transform code review into an engineering process,
to tap into its potential to develop high-quality software.**

Problem statement

There is no science of modern code review.

There are insufficient tools and evidence to answer the most fundamental questions and make it an engineering process.

Key goal for the next years

Make the first fundamental steps in establishing evidence-based modern code review, to make it a science.

Call to Arms — Evidence-based modern code review

Research vision

**Transform code review into an engineering process,
to tap into its potential to develop high-quality software.**

Key goal for the next years

**Make the first fundamental steps in establishing
evidence-based modern code review, to make it a science.**

Code review — What do we (really) know so far?

We still know only a little, but we are making important steps:

CONVERGENT CONTEMPORARY SOFTWARE PEER REVIEW PRACTICES
[Rigby and Bird — FSE 2013]
- Contemporary peer review follows a lightweight, flexible process
- Reviews happen early (before a change is committed), quickly, and frequently
- Change sizes are small
- Two reviewers find an optimal number of defects
- Review has changed from a defect finding activity to a group problem solving activity

THE IMPACT OF MODER CODE REVIEW ON SOFTWARE QUALITY
[McIntosh, Kamei, Adams, Hassan — EMSE 2016]

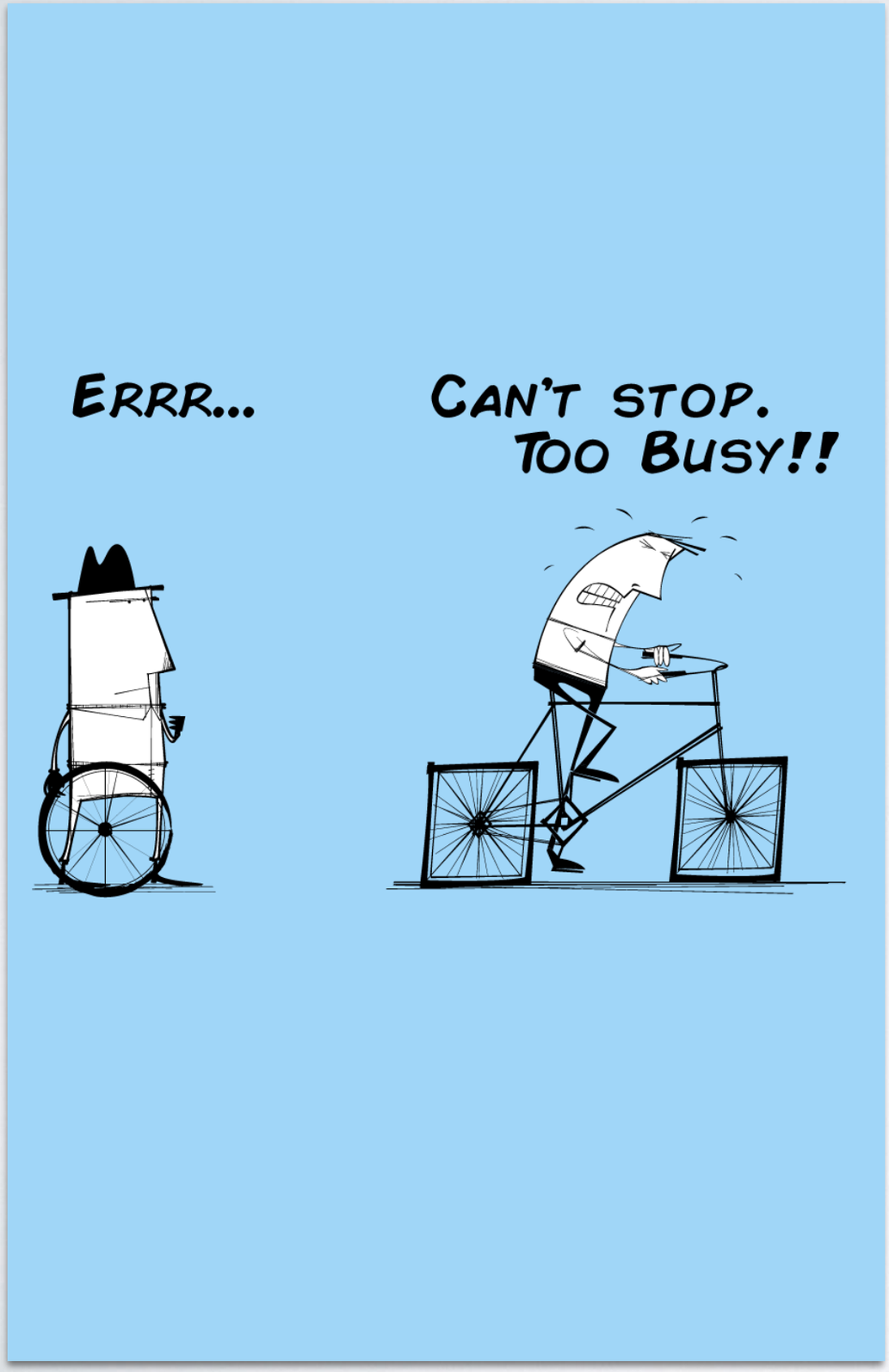
CODE OWNERSHIP IN THE SCOPE OF MODERN CODE REVIEW
[Thongtanunam, McIntosh, Hassan, Iida — ICSE 2016]

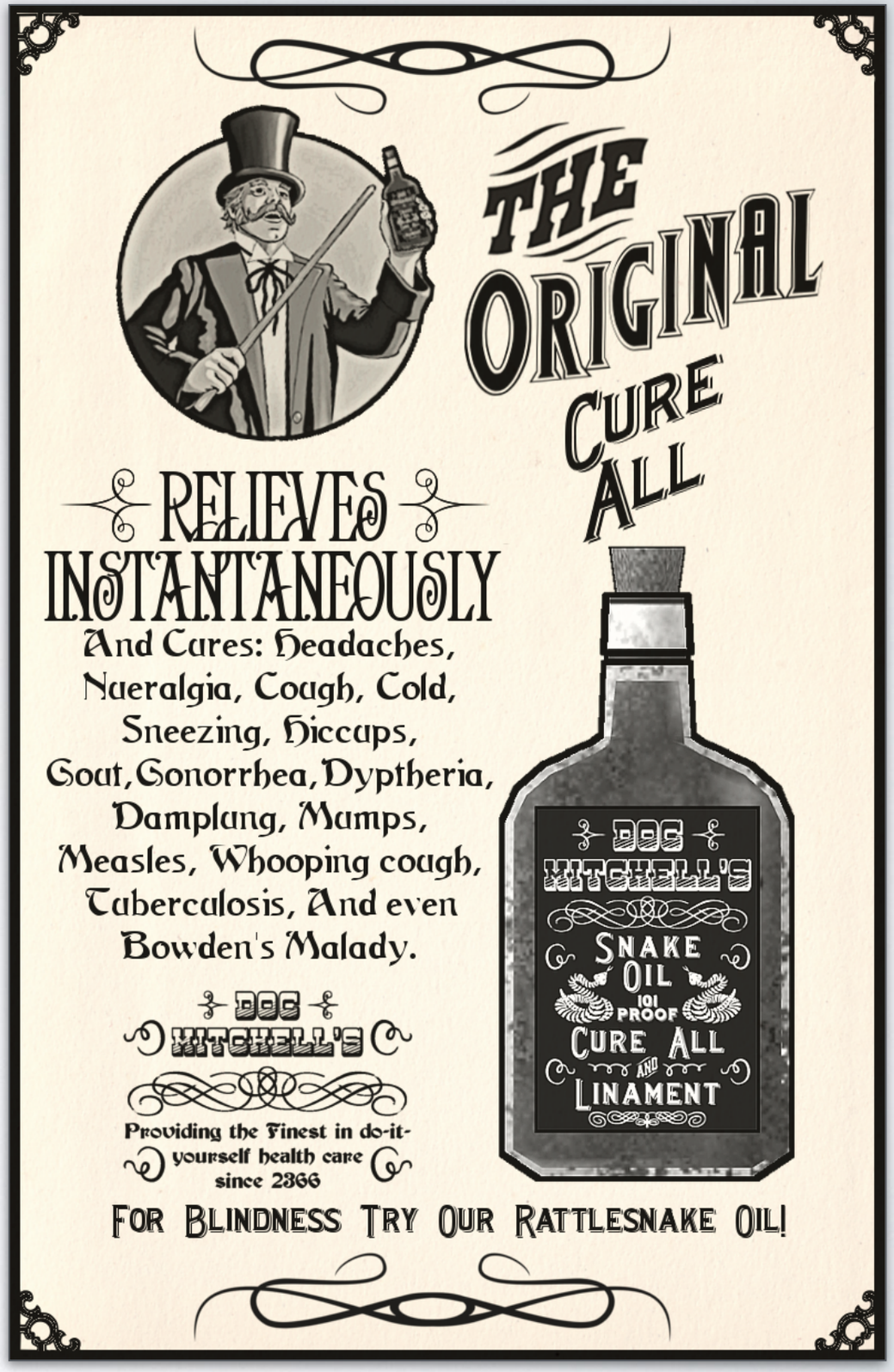
TECHNICAL AND NON-TECHNICAL FACTORS IN MODERN CODE REVIEW
[Baysal, Kononenko, Holmes, Godfrey — EMSE 2016]

CODE REVIEW QUALITY: HOW DEVELOPERS SEE IT
[Kononenko, Baysal, Godfrey — ICSE 2016]

…

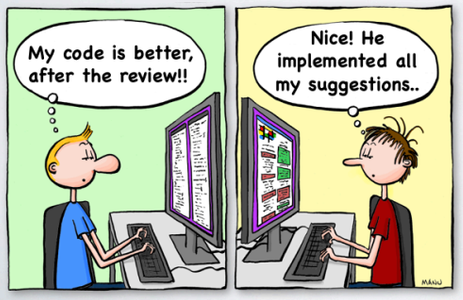
Call to Arms — Evidence-based modern code review





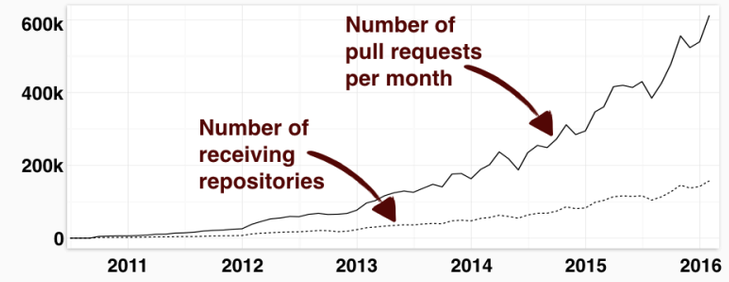
Alberto Bacchelli
University of Zurich, Switzerland
Delft University of Technology, The Netherlands

Evidence-based Code Review



How popular is it?

Monthly growth of pull request usage on GitHub (OSS)



Popularity in industrial setting

Code review tool is used by more than 70,000 developers at Microsoft
[Czerwonka, Greiler, Tilford — ICSE 2016]

Each code change in the main repository at Google is reviewed
[Potvin, Levenberg — Communications of the ACM, 2016]

Most other companies have similar policies and big players then to develop their own review tools (e.g., Facebook)

Wikipedia page on code review — an embarrassing stub

The **main code-review objectives** are (1) **best practice**, (2) **error detection**, (3) **vulnerability exposure**, (4) **malware discovery** […]. Of the four objectives, **malware is the only one that requires human detection**.

Code **reviews can often find and remove common vulnerabilities** such as format string exploits, race conditions, memory leaks and buffer overflows, **thereby improving software security**.

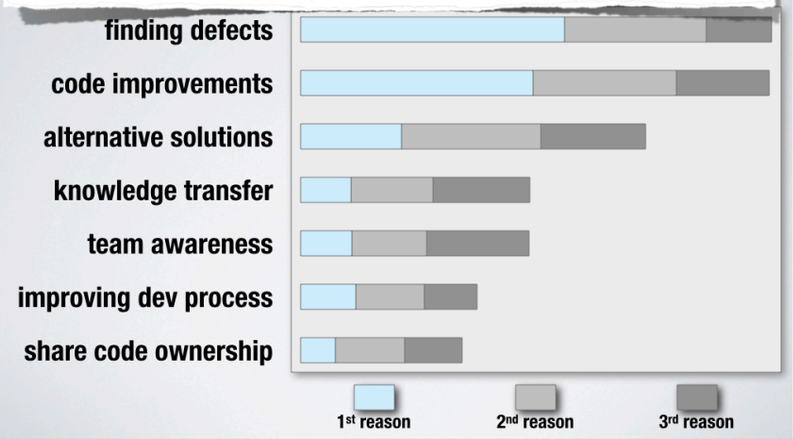
Lightweight code review typically requires **less overhead than formal code inspections**, though it can be **equally effective when done properly**.[citation needed]

In-field study on modern code review at Microsoft

Why do programmers do code reviews? — Motivations' ranking

"Finding defects is the main reason for doing code review."

72 managers and 384 developers @ Microsoft



What is the outcome of code reviews? Reality ≠ expectations

Call to Arms — Evidence-based modern code review

Research vision

Transform code review into an engineering process, to tap into its potential to develop high-quality software.

Key goal for the next years

Make the first fundamental steps in establishing an evidence-based modern code review, to make it a science.

Would you like to work on this vision (with us)?

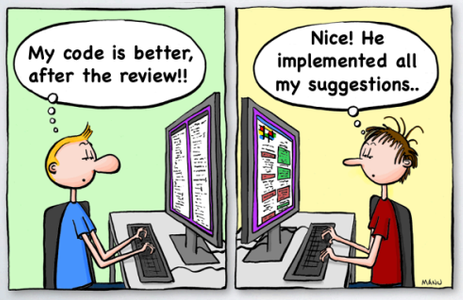
My research group (ZEST) at UZH has 2 fully funded 4-year PhD positions!

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

Find me at the end of the talk, at bacchelli@ifi.uzh.ch , or at @sback_

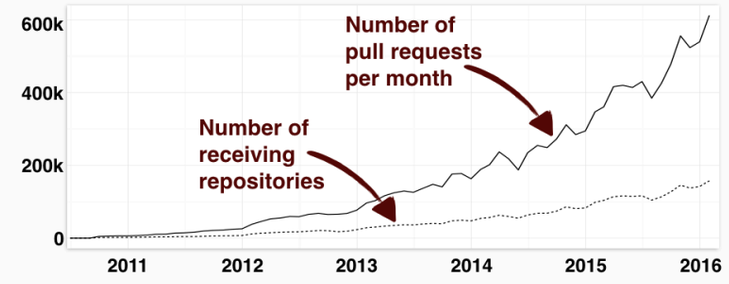
Alberto Bacchelli
University of Zurich, Switzerland
Delft University of Technology, The Netherlands

Evidence-based Code Review



How popular is it?

Monthly growth of pull request usage on GitHub (OSS)



Popularity in industrial setting

Code review tool is used by more than 70,000 developers at Microsoft
[Czerwonka, Greiler, Tilford — ICSE 2016]

Each code change in the main repository at Google is reviewed
[Potvin, Levenberg — Communications of the ACM, 2016]

Most other companies have similar policies and big players then to develop their own review tools (e.g., Facebook)

Wikipedia page on code review — an embarrassing stub

The **main code-review objectives** are (1) **best practice**, (2) **error detection**, (3) **vulnerability exposure**, (4) **malware discovery** […]. Of the four objectives, **malware is the only one that requires human detection**.

Code **reviews can often find and remove common vulnerabilities** such as format string exploits, race conditions, memory leaks and buffer overflows, **thereby improving software security**.

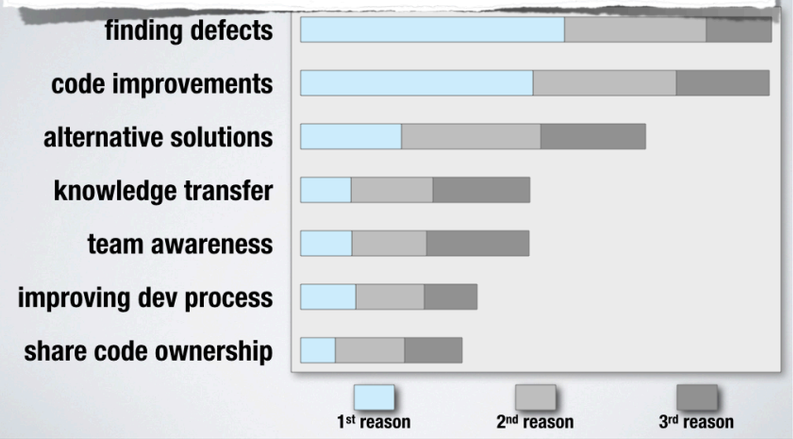
Lightweight code review typically requires **less overhead than formal code inspections**, though it can be **equally effective when done properly**.[citation needed]

In-field study on modern code review at Microsoft

Why do programmers do code reviews? — Motivations' ranking

"Finding defects is the main reason for doing code review."

72 managers and 384 developers @ Microsoft



What is the outcome of code reviews? Reality ≠ expectations

Why do expectations not match reality?

Call to Arms — Evidence-based modern code review

Research vision

Transform code review into an engineering process, to tap into its potential to develop high-quality software.

Key goal for the next years

Make the first fundamental steps in establishing an evidence-based modern code review, to make it a science.

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

My research group (ZEST) at UZH has 2 fully funded 4-year PhD positions!

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

Find me at the end of the talk, at bacchelli@ifi.uzh.ch , or at @sback_