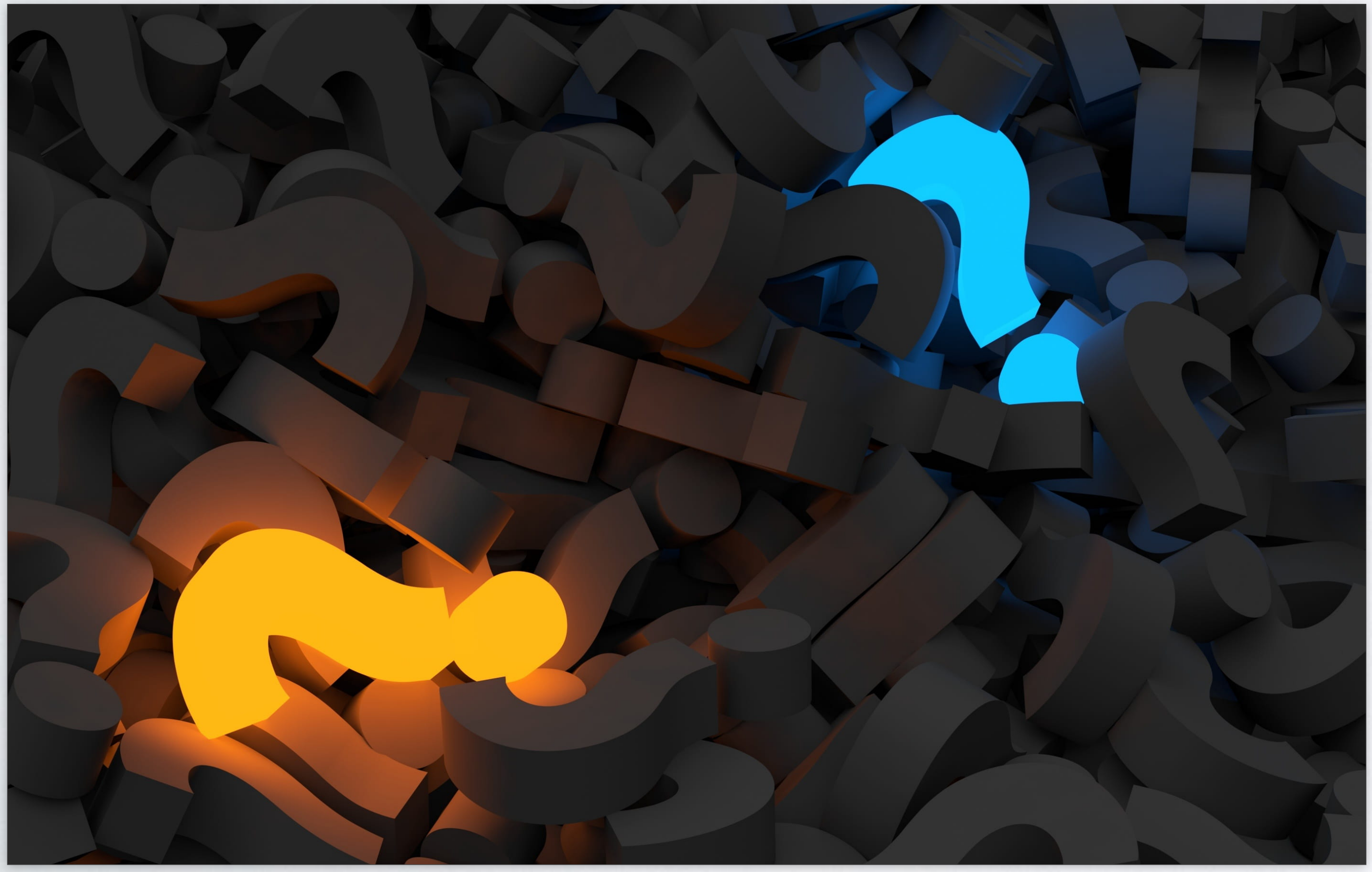
Ten Years Of Asking Questions On Code Review



Alberto Bacchelli

Associate Professor
Department of Informatics
University of Zurich

ZEST: Zurich Empirical Software engineering Team

http://zest.ifi.uzh.ch
zest@ifi.uzh.ch

Research

- Software quality
  - Peer code review
  - Software testing
  - Software security
- Fundamentals of Data Science for Software Engineering
  - Predictive Analytics
  - Data-driven Tools

Education

- Software Design & Construction
- Software Testing
- Software Analytics
  (aka Data Science for S.E.)

ZEST: Zurich Empirical Software engineering Team

http://zest.ifi.uzh.ch
zest@ifi.uzh.ch

Research

- Software quality
  - Peer code review
  - Software testing
  - Software security
- Fundamentals of Data Science for Software Engineering
  - Predictive Analytics
  - Data-driven Tools

Google

Software Improvement Group

ABB

moz://a

Microsoft

HUAWEI

AdNovum

Empirical Software Engineering?

Empirical software engineering involves the scientific use of quantitative and qualitative data to understand and improve software product, software development process and software management.

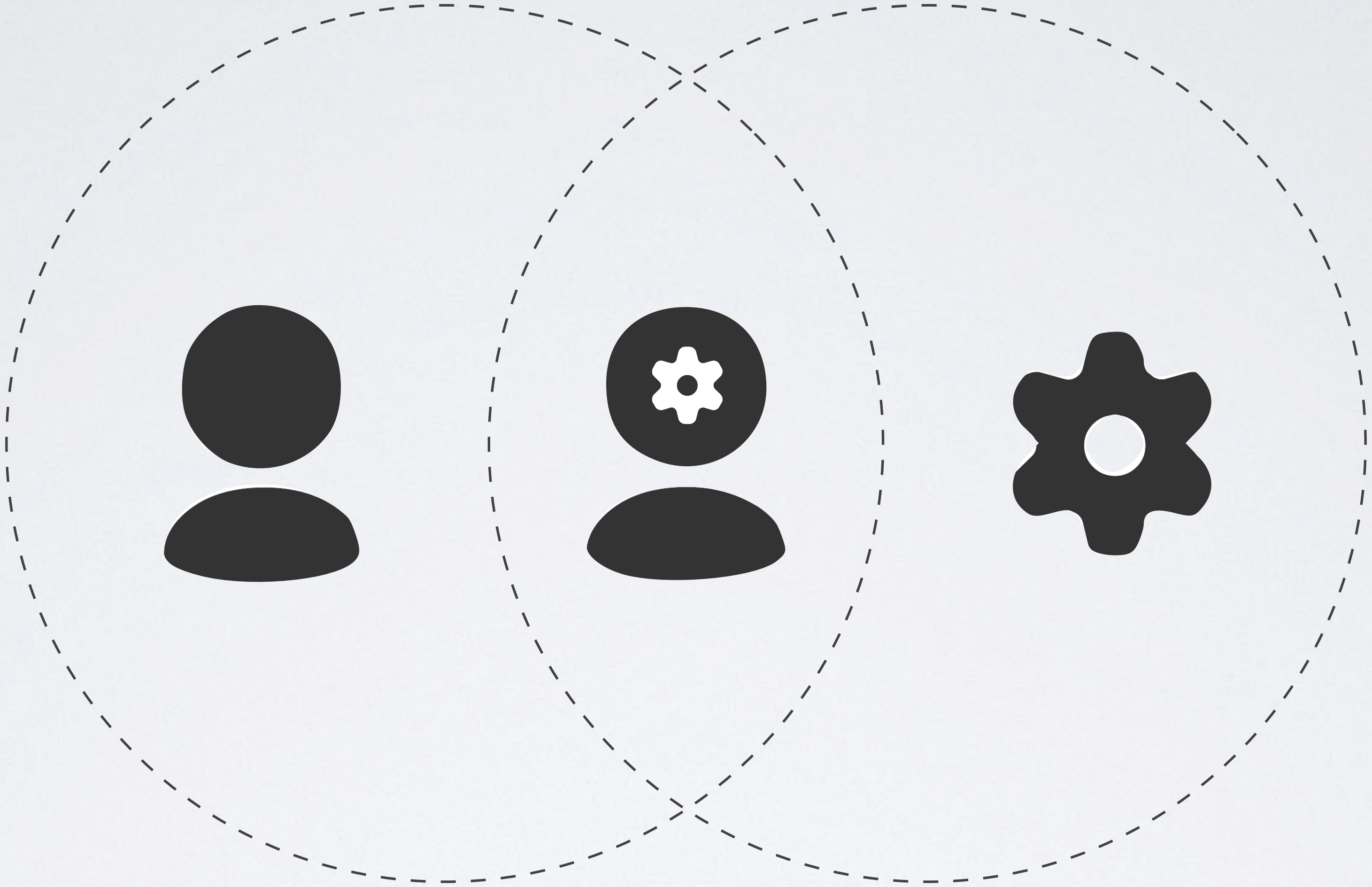
Empirical software engineering **starts with a good question**:

- Does pair programming work?
- Is static typing really good?
- What are the advantages of properly following continuous integration?
- How does using GitHub influence open-source projects?

Empirical software engineering **leads to actionable results**:

- The creation of new tools
- The improvement of existing tools
- The improvement of existing development and engineering processes
- More questions :)

Software Engineering as a Socio-Technical Space



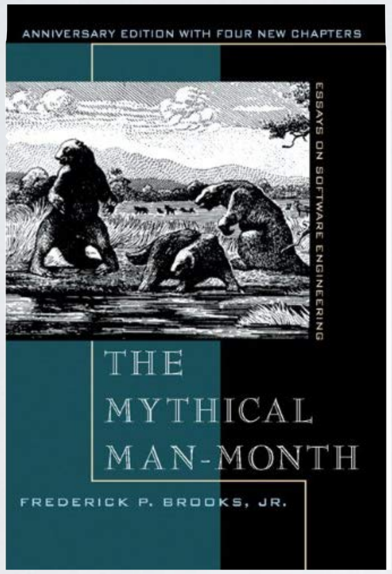
human / social aspects

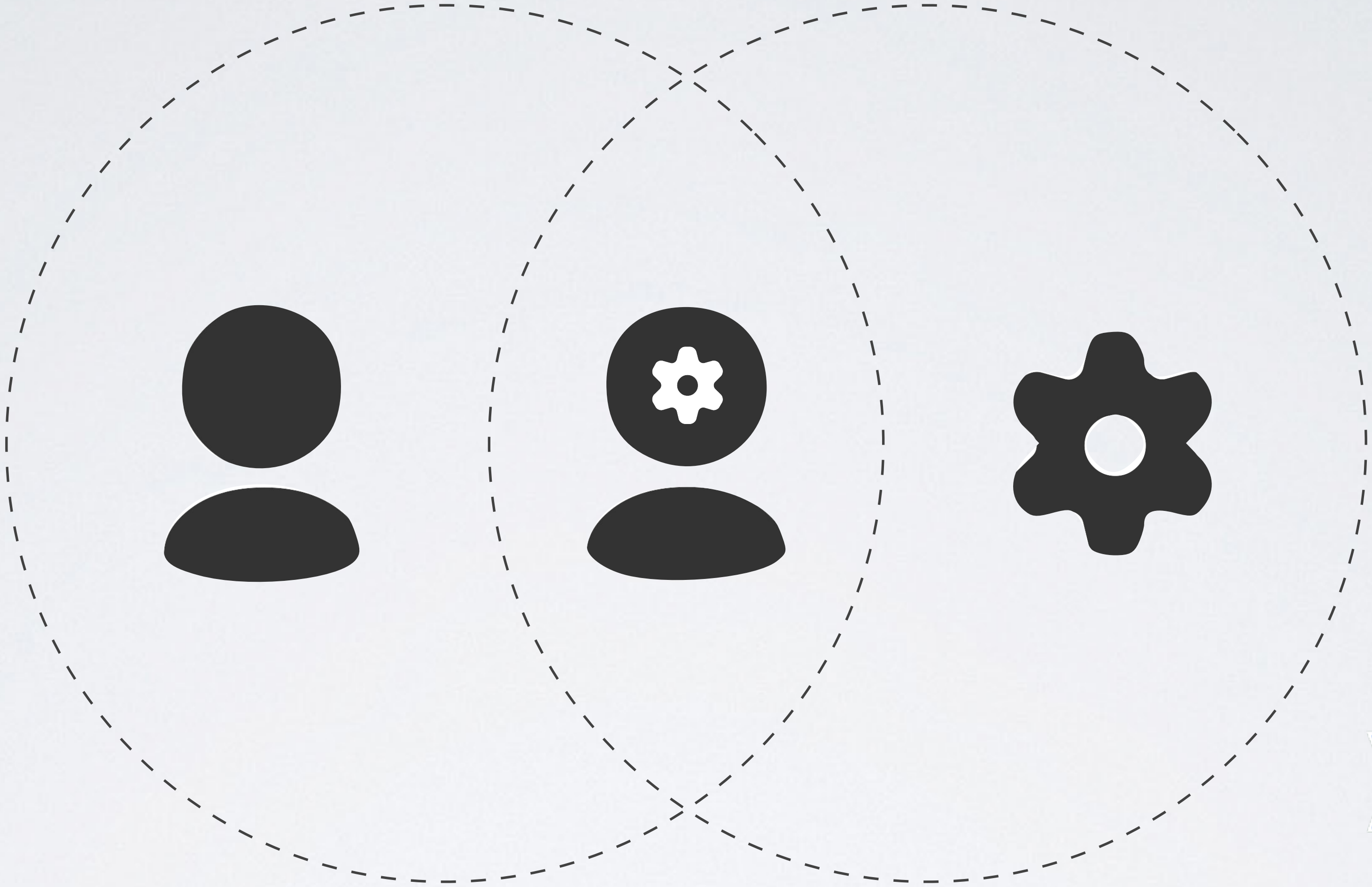
technical aspects

socio-technical aspects

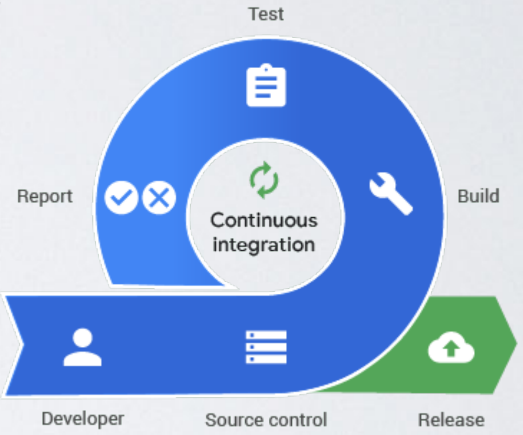
Software Engineering as a Socio-Technical Space

human / social


ANNIVERSARY EDITION WITH FOUR NEW CHAPTERS
ESSAYS ON SOFTWARE ENGINEERING
THE MYTHICAL MAN-MONTH
FREDERICK P. BROOKS, JR.

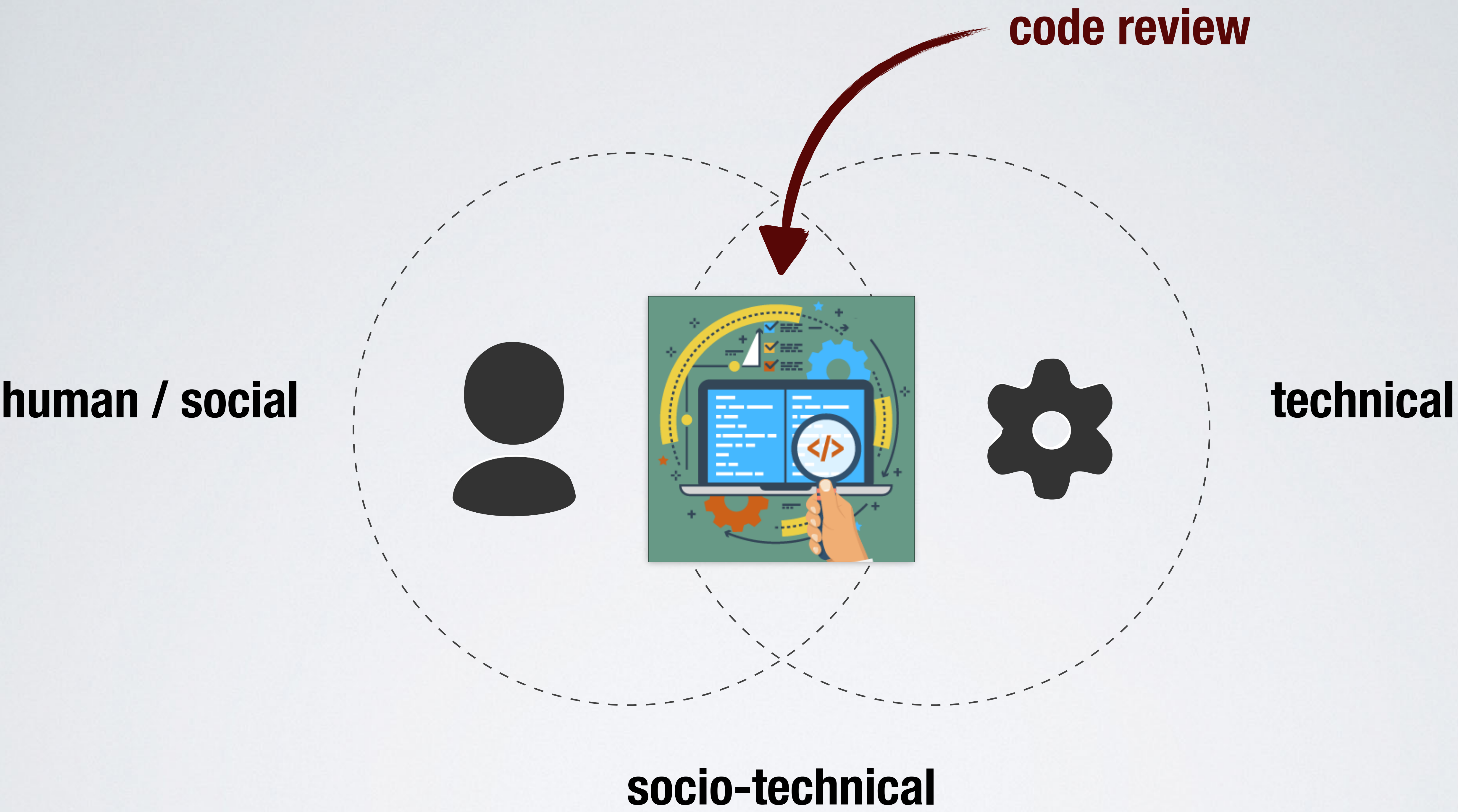


technical



socio-technical

Joint Space — Code Review



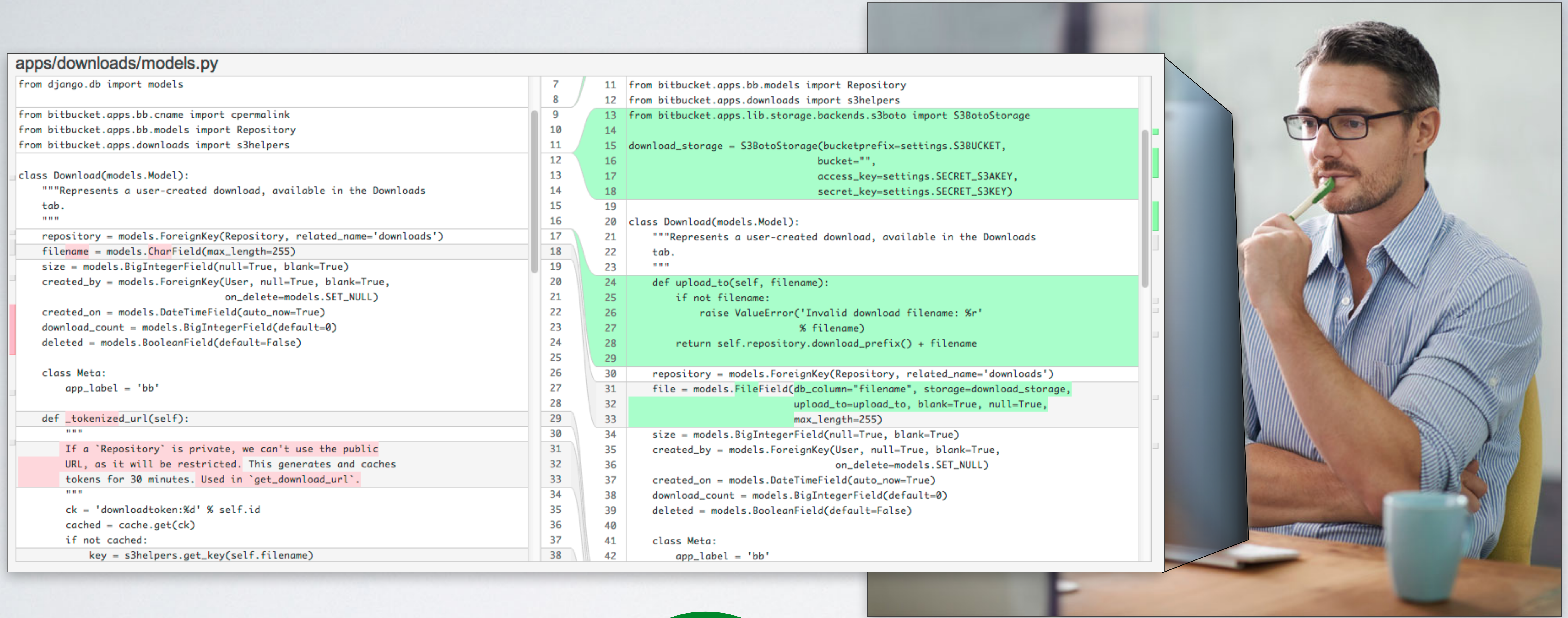
code review

human / social

technical

socio-technical

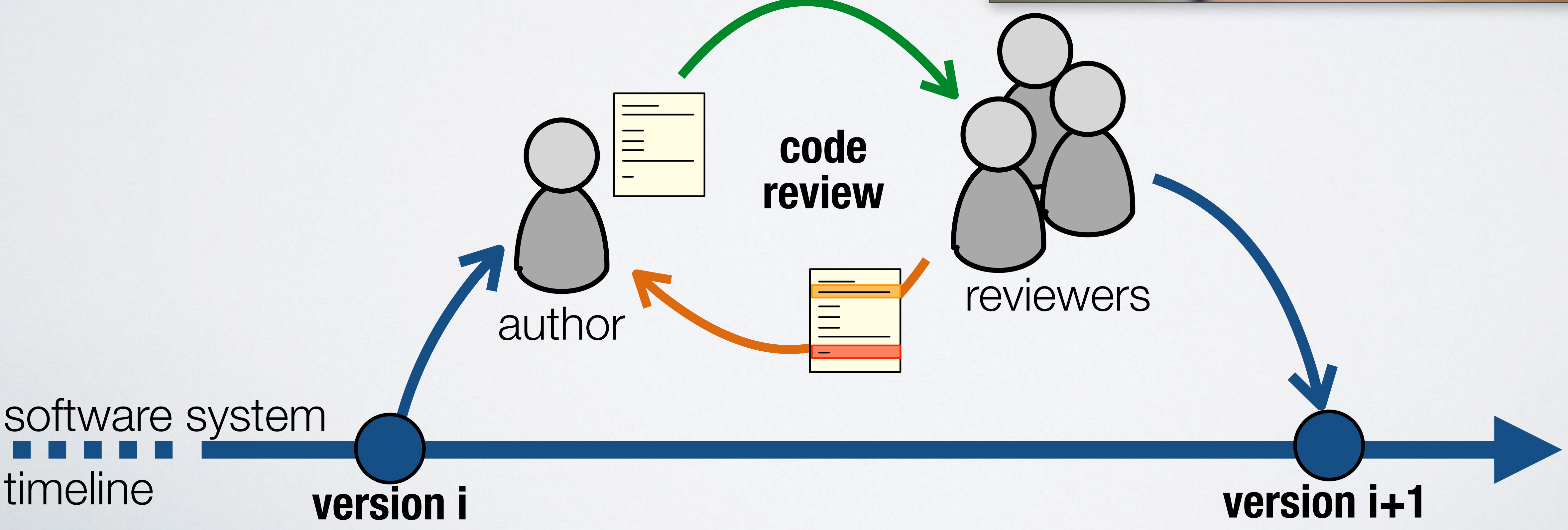
Joint Space — Code Review



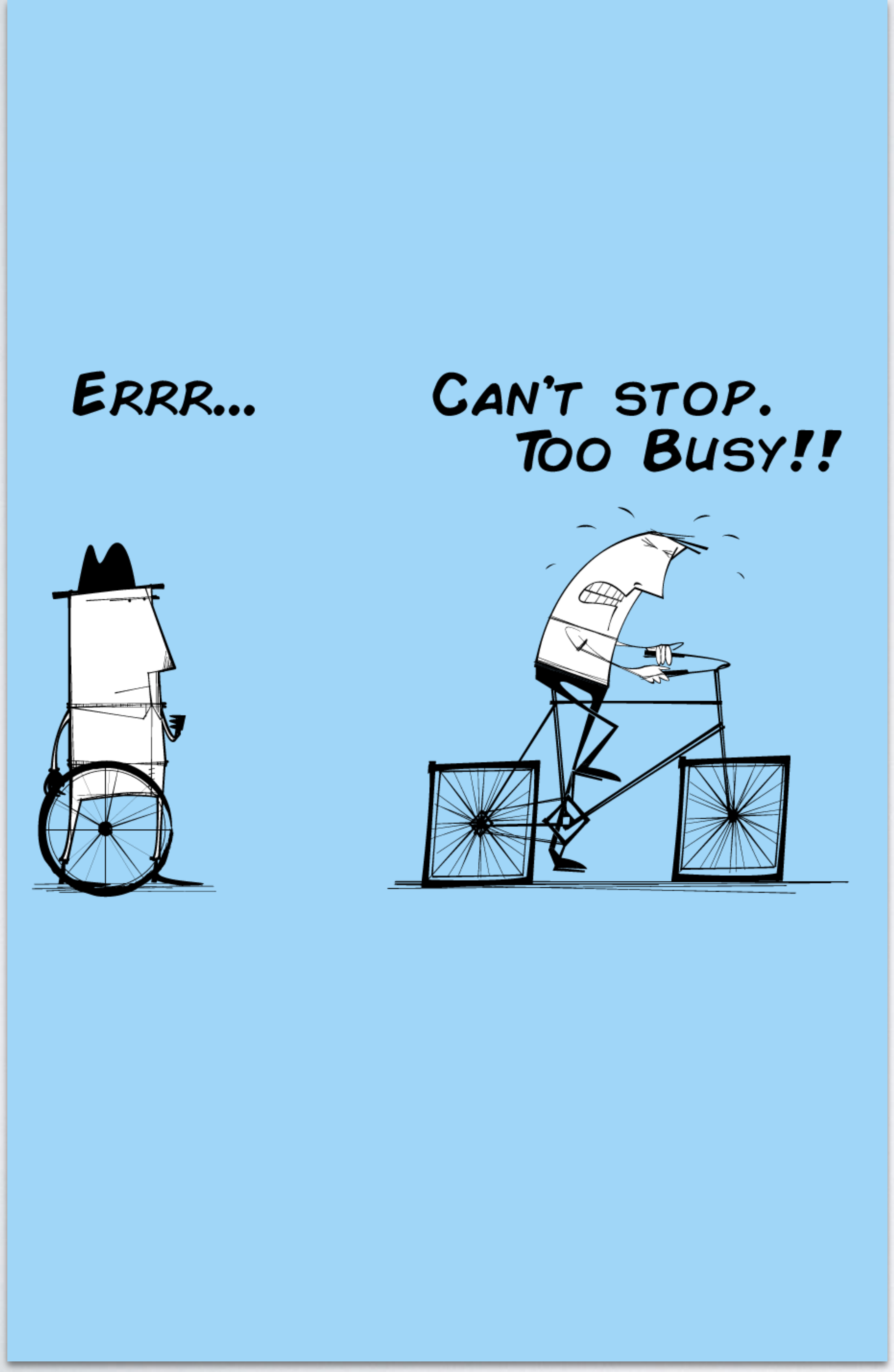
apps/downloads/models.py

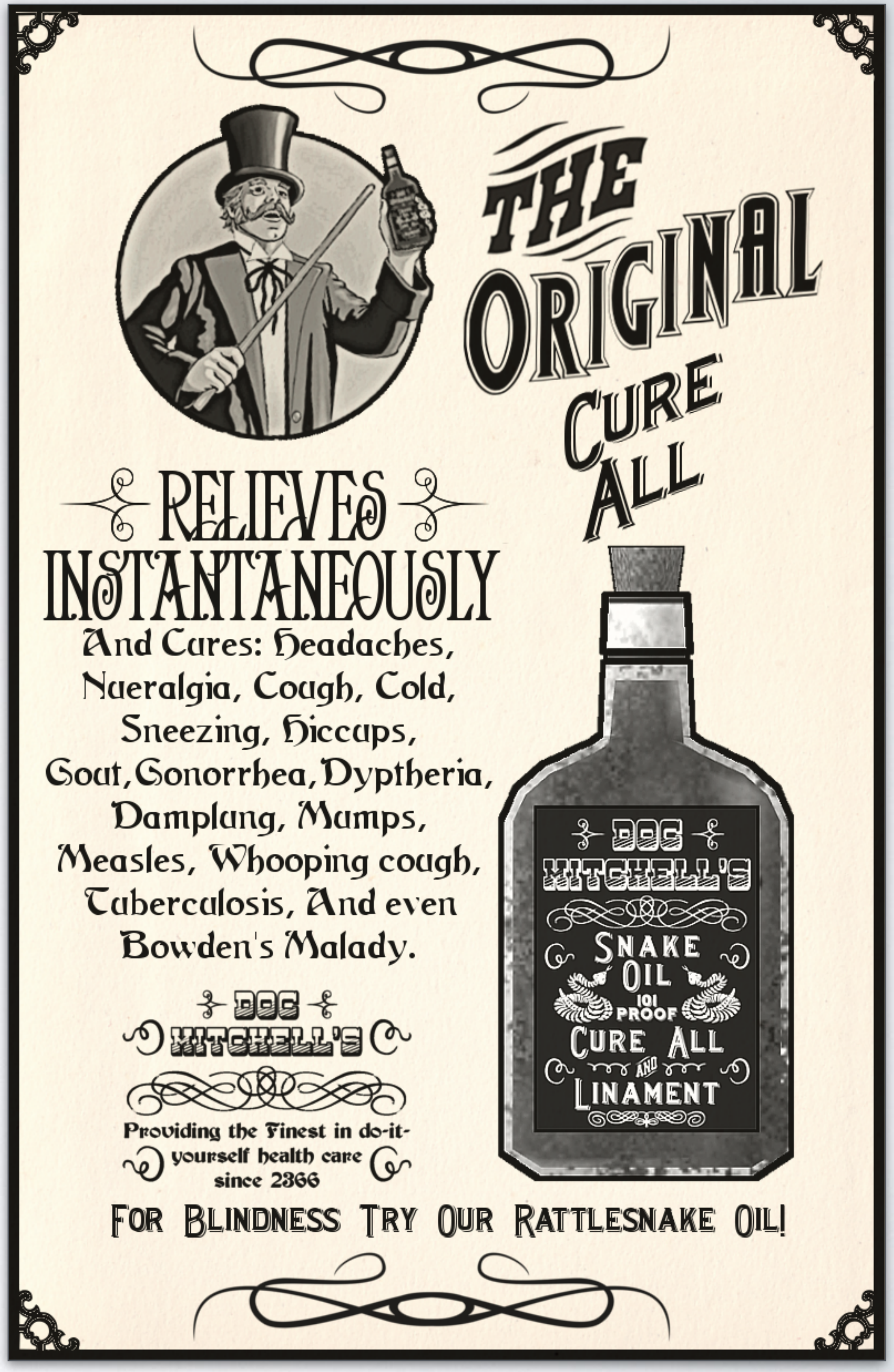
```
from django.db import models

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'
```



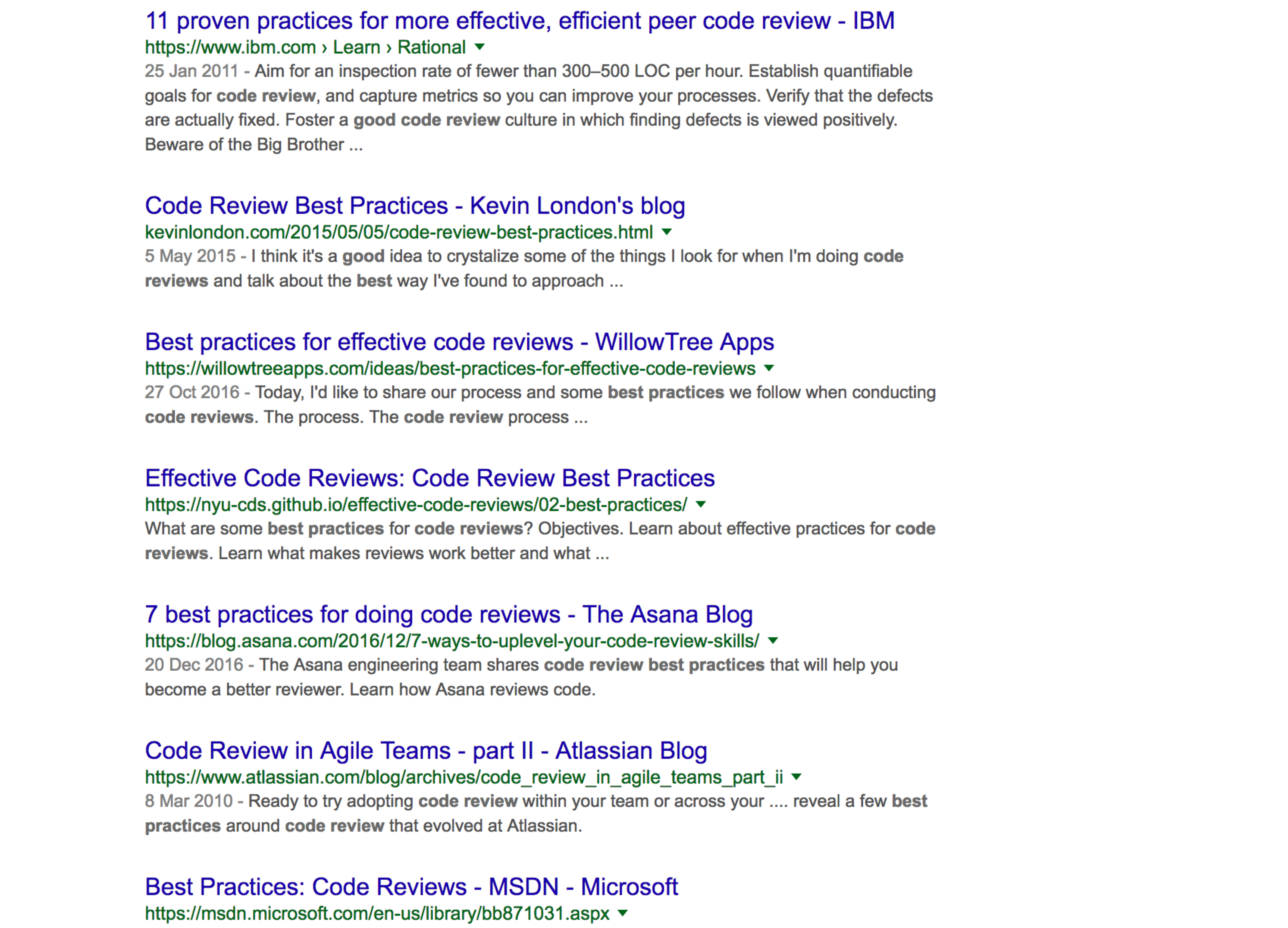
Why research on code review?




THE ORIGINAL CURE ALL
RELIEVES INSTANTANEOUSLY
And Cures: Headaches, Nueralgia, Cough, Cold, Sneezing, Hiccups, Gout, Gonorrhea, Dyptheria, Damplung, Mumps, Measles, Whooping cough, Tuberculosis, And even Bowden's Malady.
DOC MITCHELL'S
Providing the Finest in do-it-yourself health care since 2366
DOC MITCHELL'S
SNAKE OIL
101 PROOF
CURE ALL AND LINAMENT
FOR BLINDNESS TRY OUR RATTLESNAKE OIL!

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 IMPROVE 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

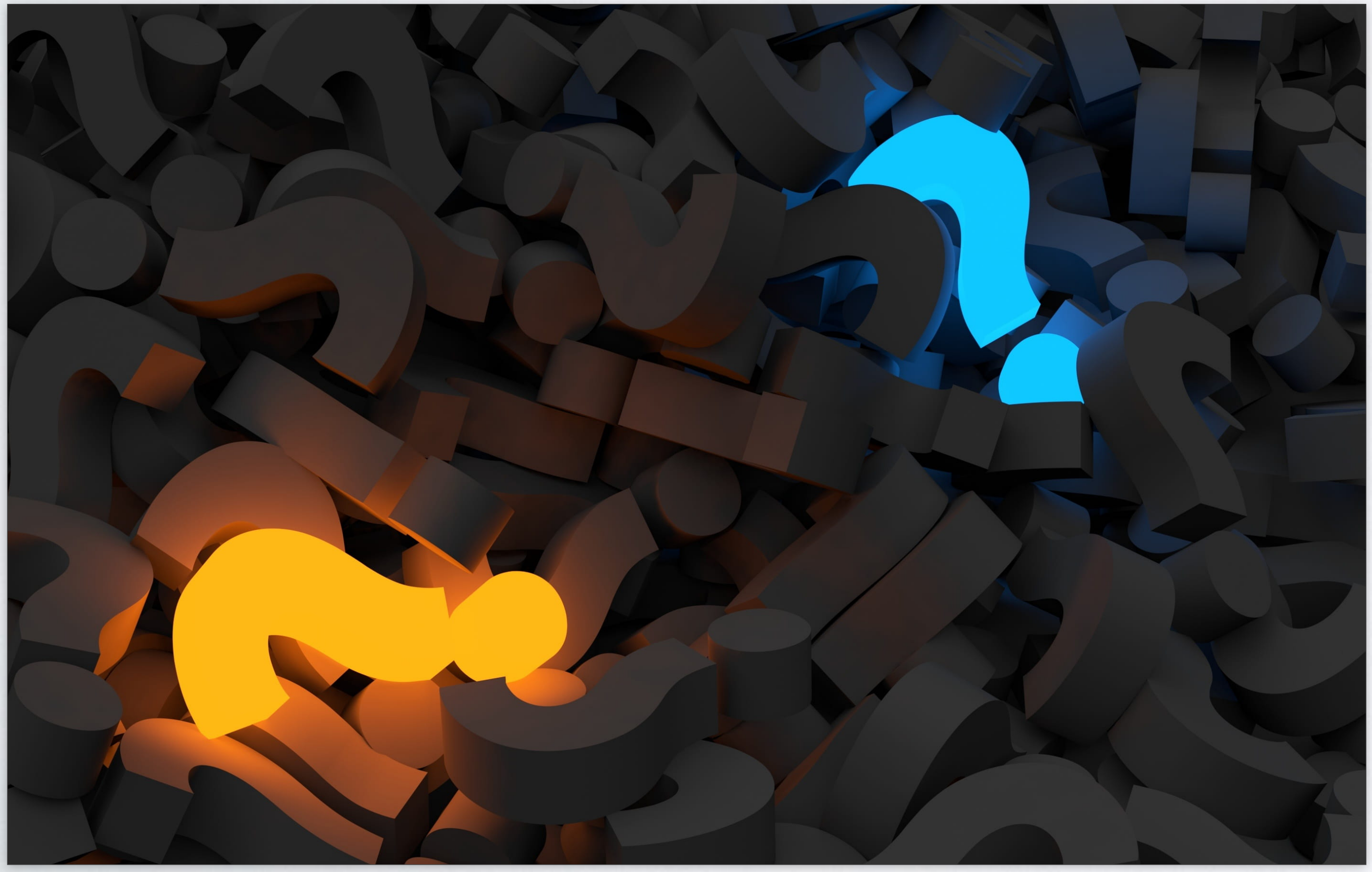
11 PROVEN PRACTICES FOR MORE EFFECTIVE, EFFICIENT CODE REVIEW

- 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 IMPROVE 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**

Ten Years Of Asking Uncomfortable Questions On Code Review



Alberto Bacchelli

Associate Professor
Department of Informatics
University of Zurich

Asking Uncomfortable Questions On Code Review

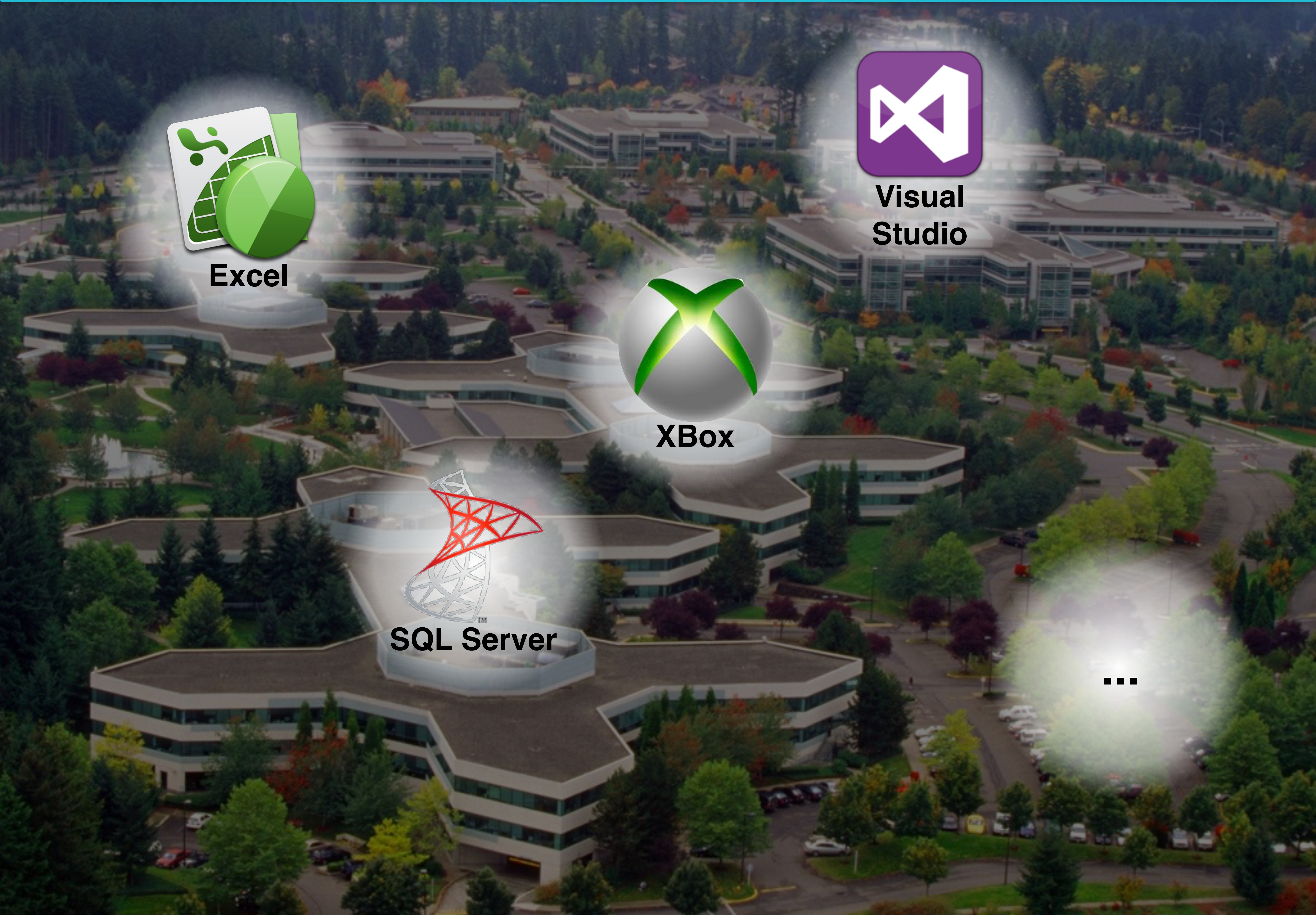
But.. why are we doing code review at all?

Why are managers and developers employing code review?

Dr. Christian Bird

Modern code review @ Microsoft

Modern code review @ Microsoft



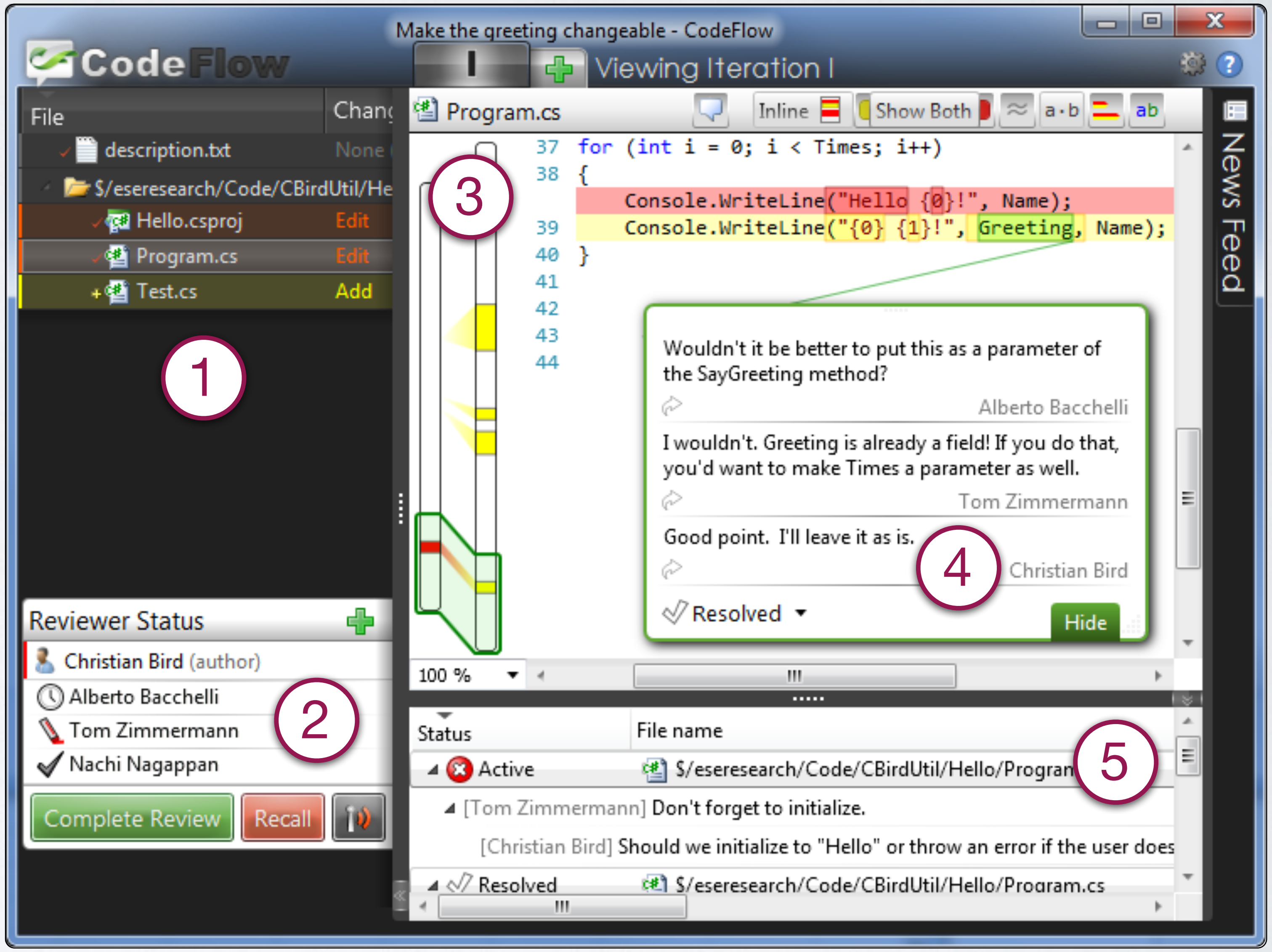
CodeFlow review tool

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

XBox

SQL Server

Visual Studio



Make the greeting changeable - CodeFlow

File	Chang
description.txt	None
$/eseresearch/Code/CBirdUtil/He	
Hello.csproj	Edit
Program.cs	Edit
Test.cs	Add

Reviewer Status

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

Program.cs

```
for (int i = 0; i < Times; i++)
{
    Console.WriteLine("Hello {0}!", Name);
    Console.WriteLine("{0} {1}!", Greeting, Name);
}
```

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.

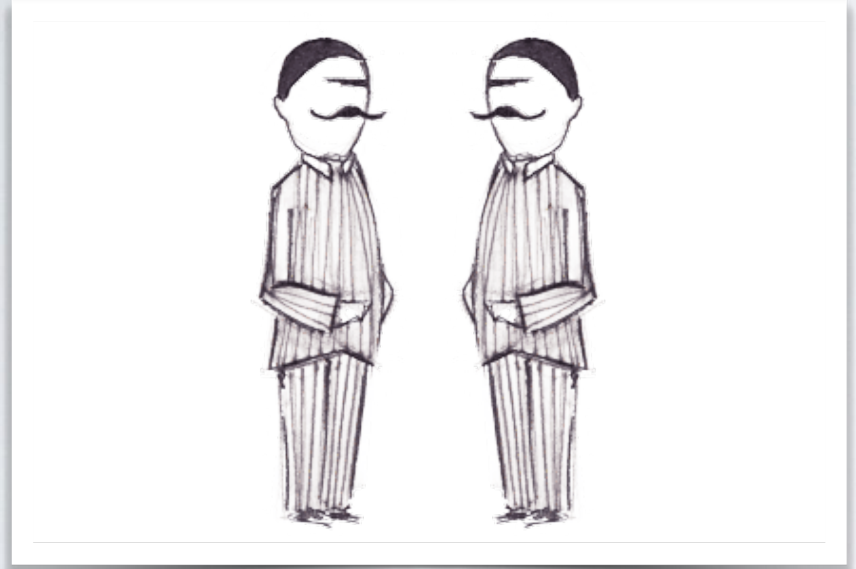
Christian Bird

✓ Resolved

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

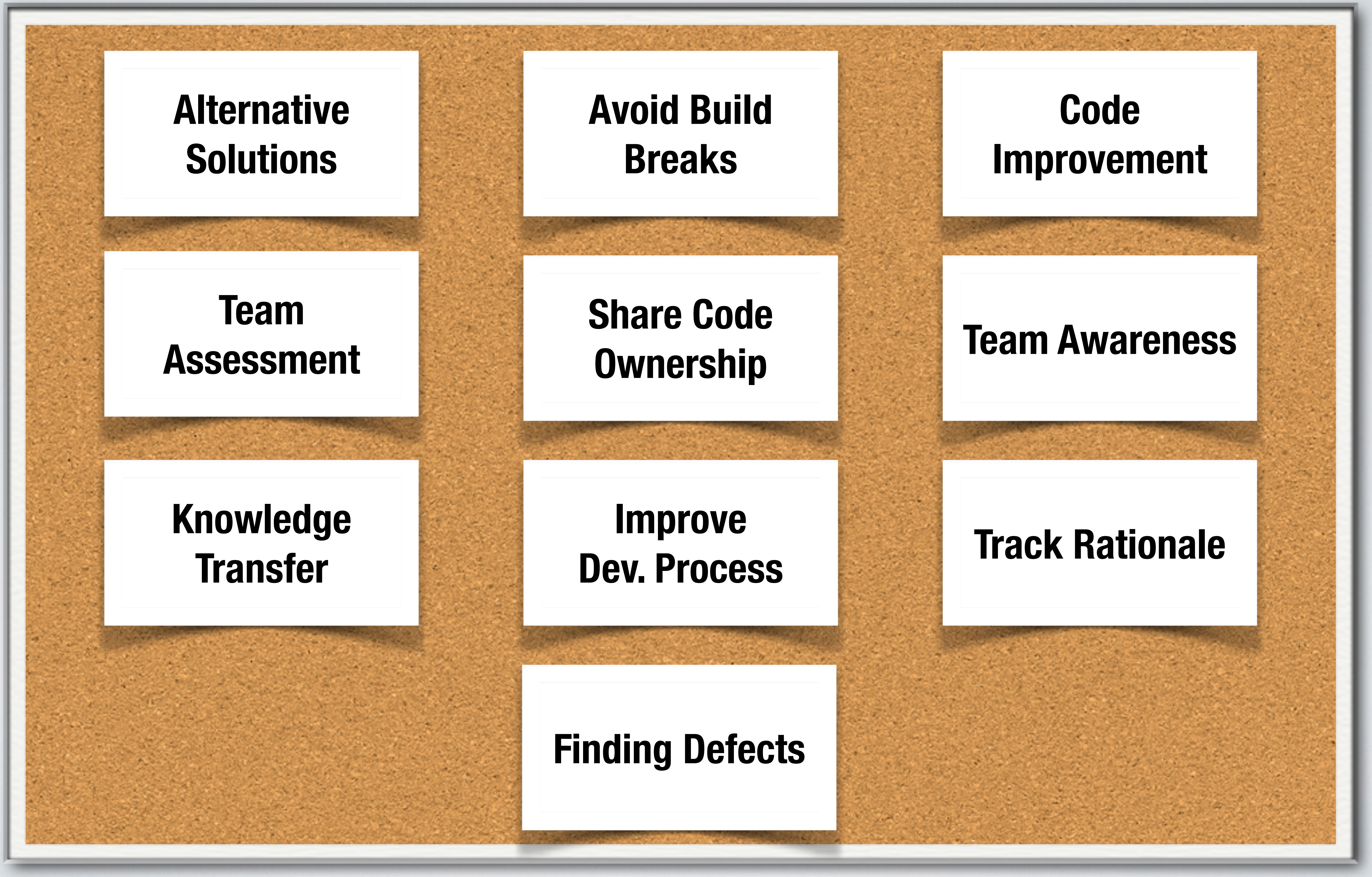
observations

18-20 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

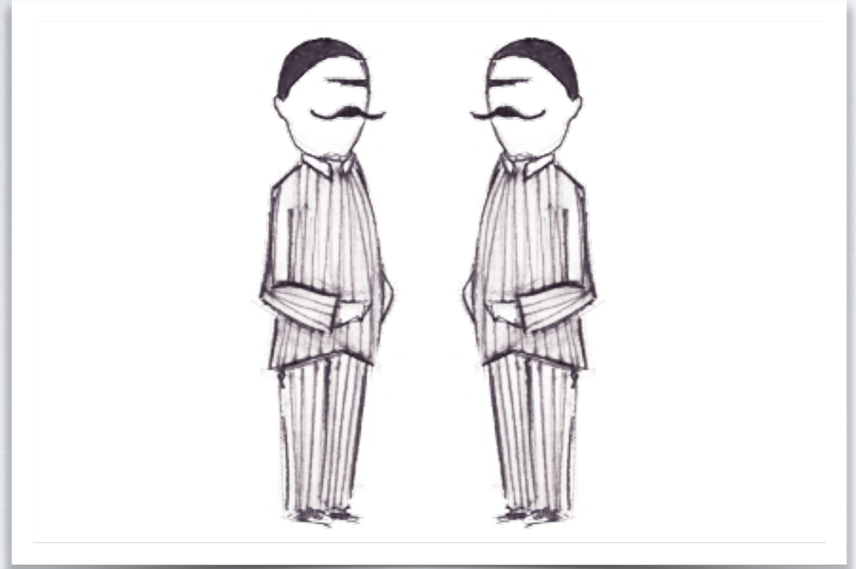
Improve Dev. Process

Track Rationale

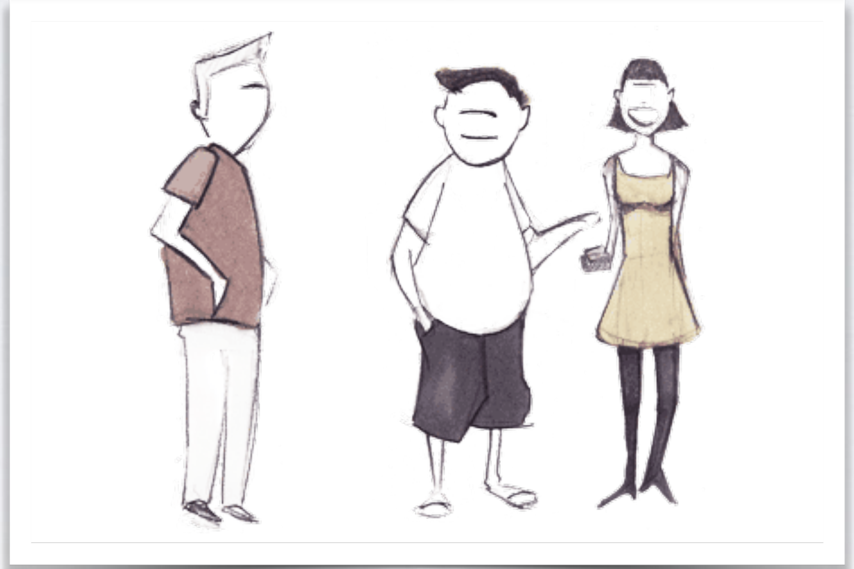
Finding Defects

observations

18-20 interviews

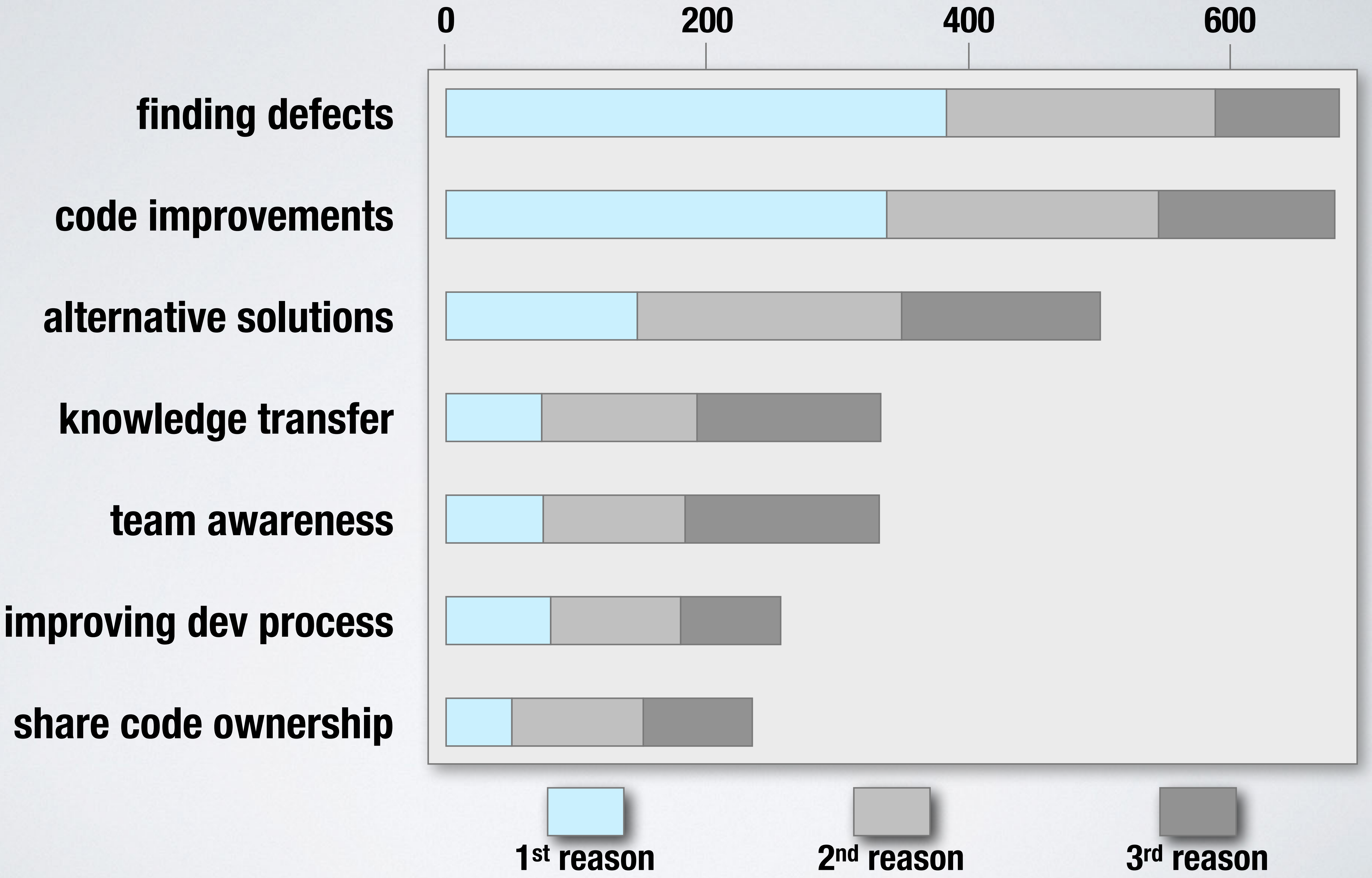


survey to 165 managers

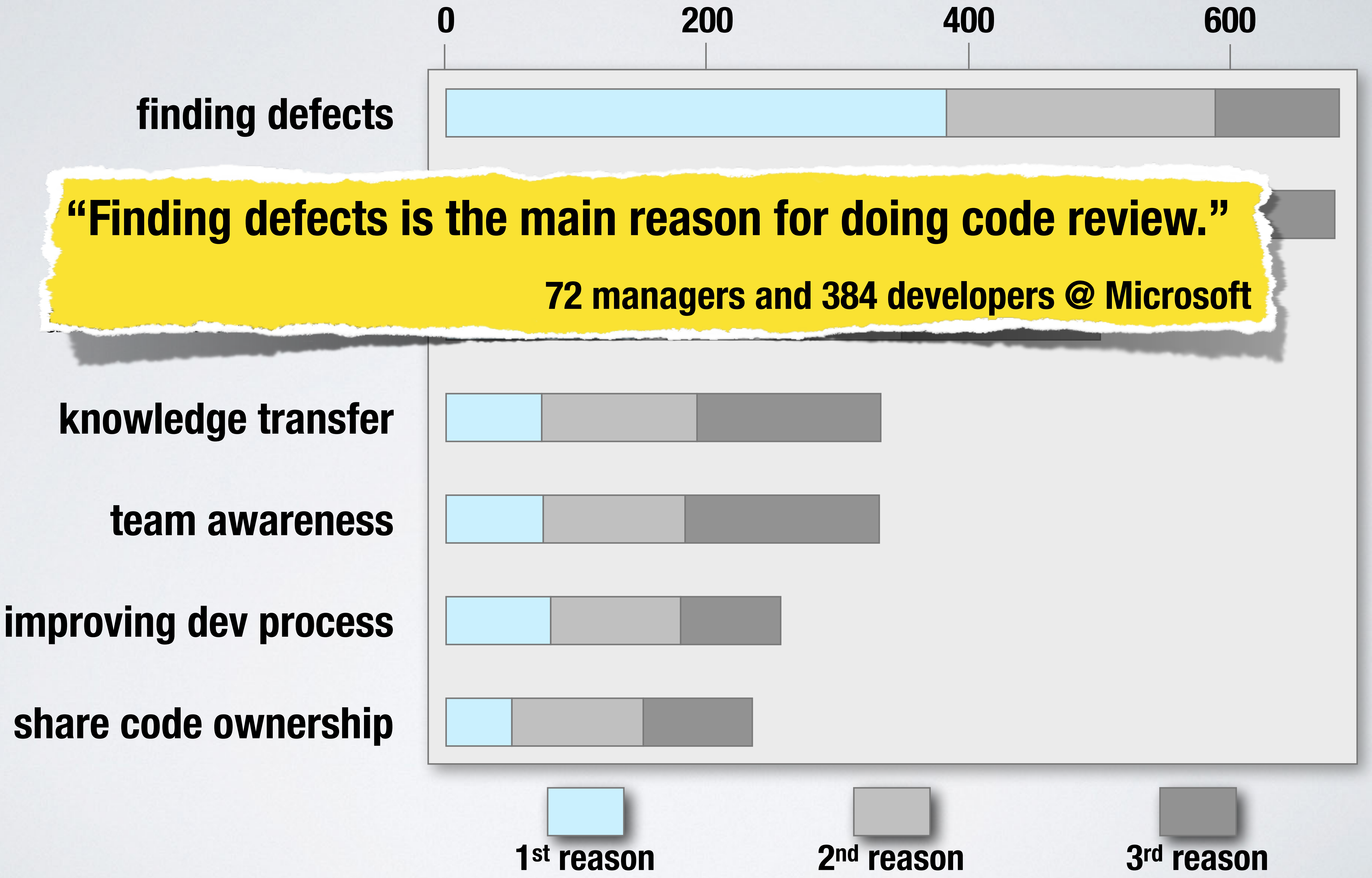


survey to 873 developers

Why do Microsoft developers do code reviews?



Why do Microsoft developers do code reviews?

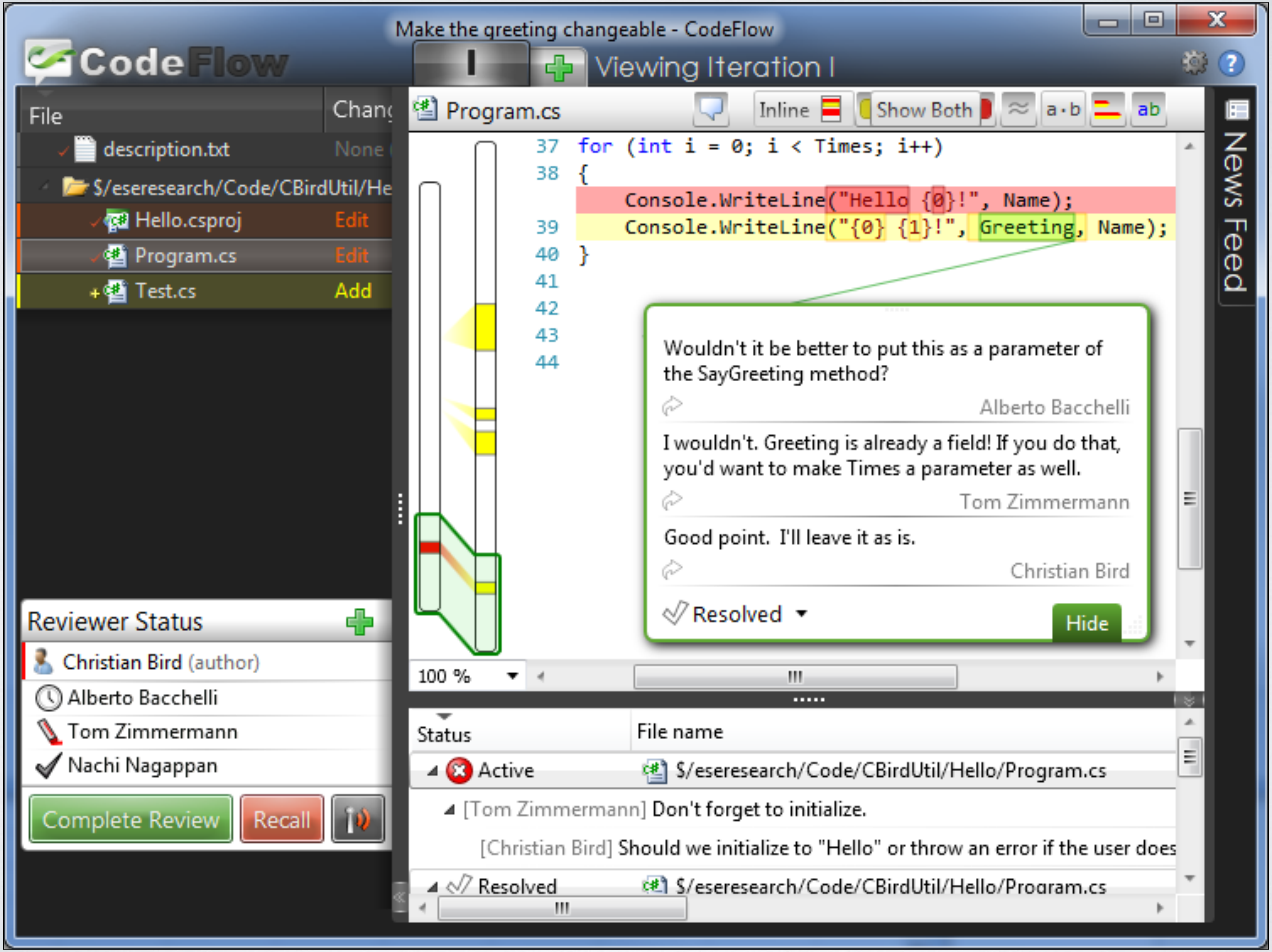


“Finding defects is the main reason for doing code review.”

72 managers and 384 developers @ Microsoft

What is the outcome of code review at Microsoft?

What is the outcome of code review at Microsoft?



File	Chang
description.txt	None
$/eseresearch/Code/CBirdUtil/He	
Hello.csproj	Edit
Program.cs	Edit
+ Test.cs	Add

```
for (int i = 0; i < Times; i++)
{
    Console.WriteLine("Hello {0}!", Name);
    Console.WriteLine("{0} {1}!", Greeting, Name);
}
```

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

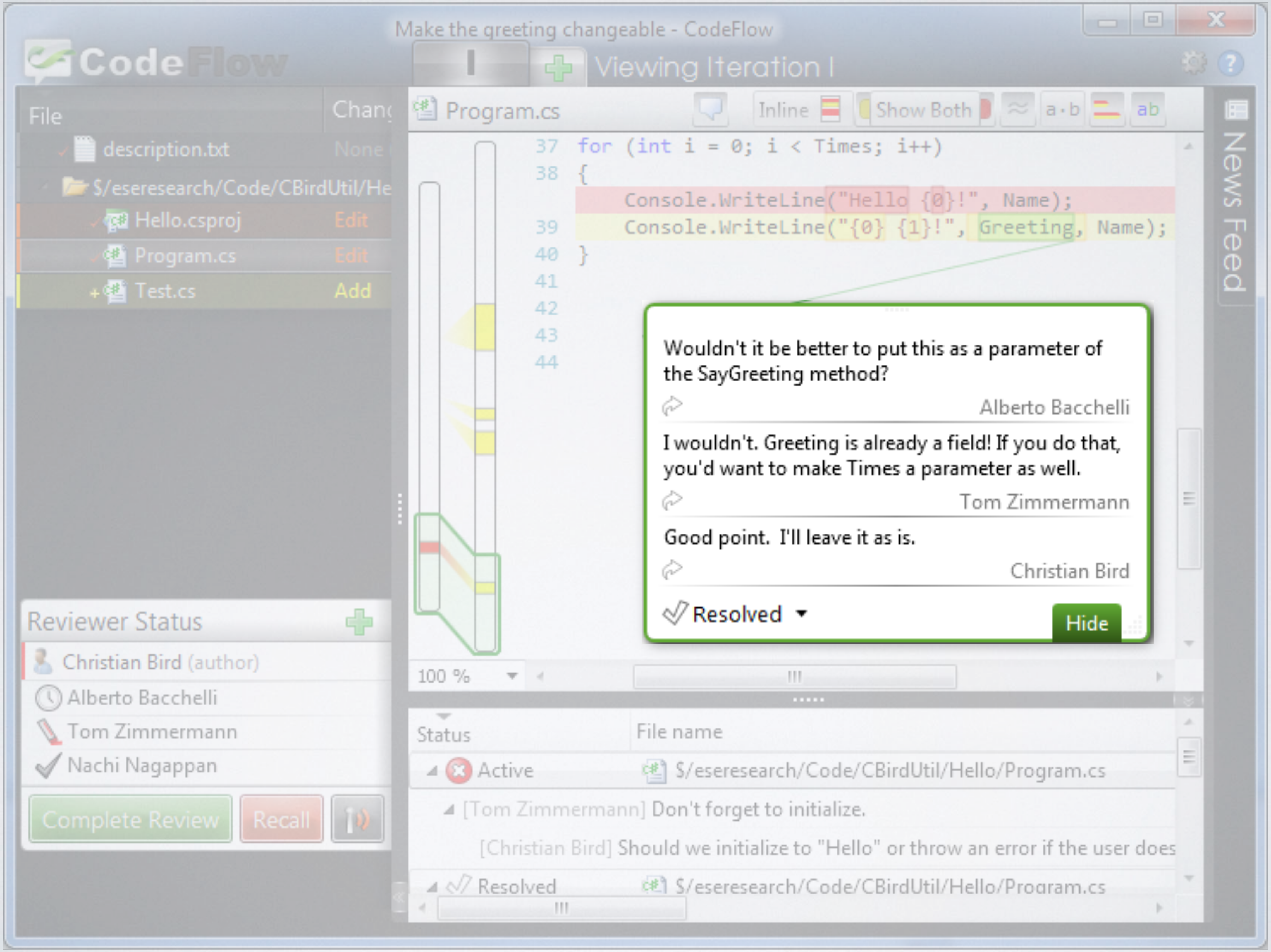
✓ Resolved ▾

Reviewer Status

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

Status	File name
Active	$/eseresearch/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	$/eseresearch/Code/CBirdUtil/Hello/Program.cs

Recorded code review comments



Make the greeting changeable - CodeFlow

```
for (int i = 0; i < Times; i++)
{
    Console.WriteLine("Hello {0}!", Name);
    Console.WriteLine("{0} {1}!", Greeting, Name);
}
```

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

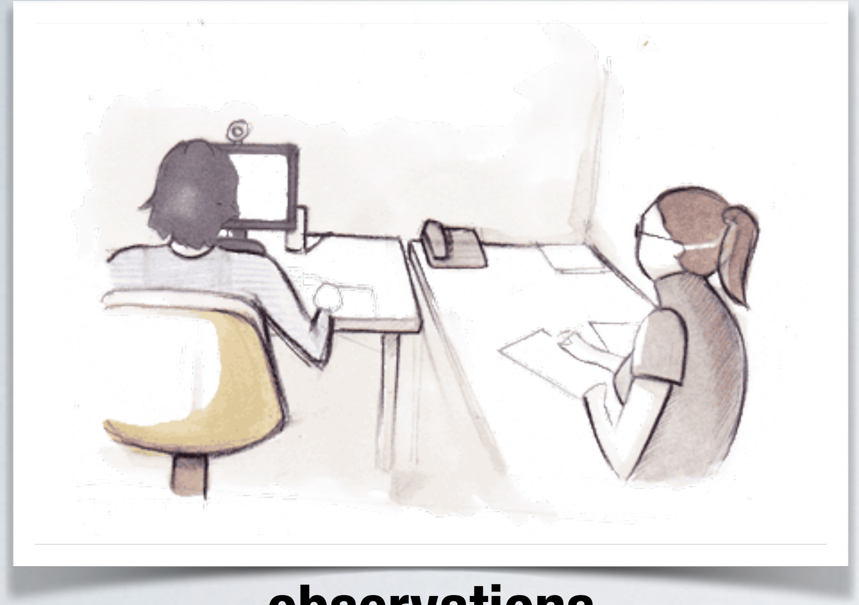
Good point. I'll leave it as is.

Christian Bird

Resolved

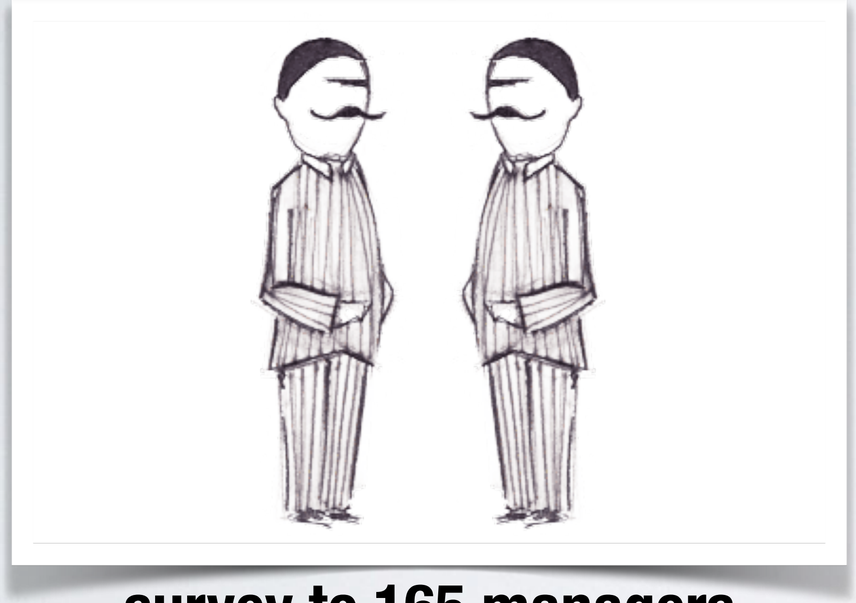
Christian Bird (author)
Alberto Bacchelli
Tom Zimmermann
Nachi Nagappan

Status	File name
Active	$/eseresearch/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	$/eseresearch/Code/CBirdUtil/Hello/Program.cs



observations

18-20 interviews

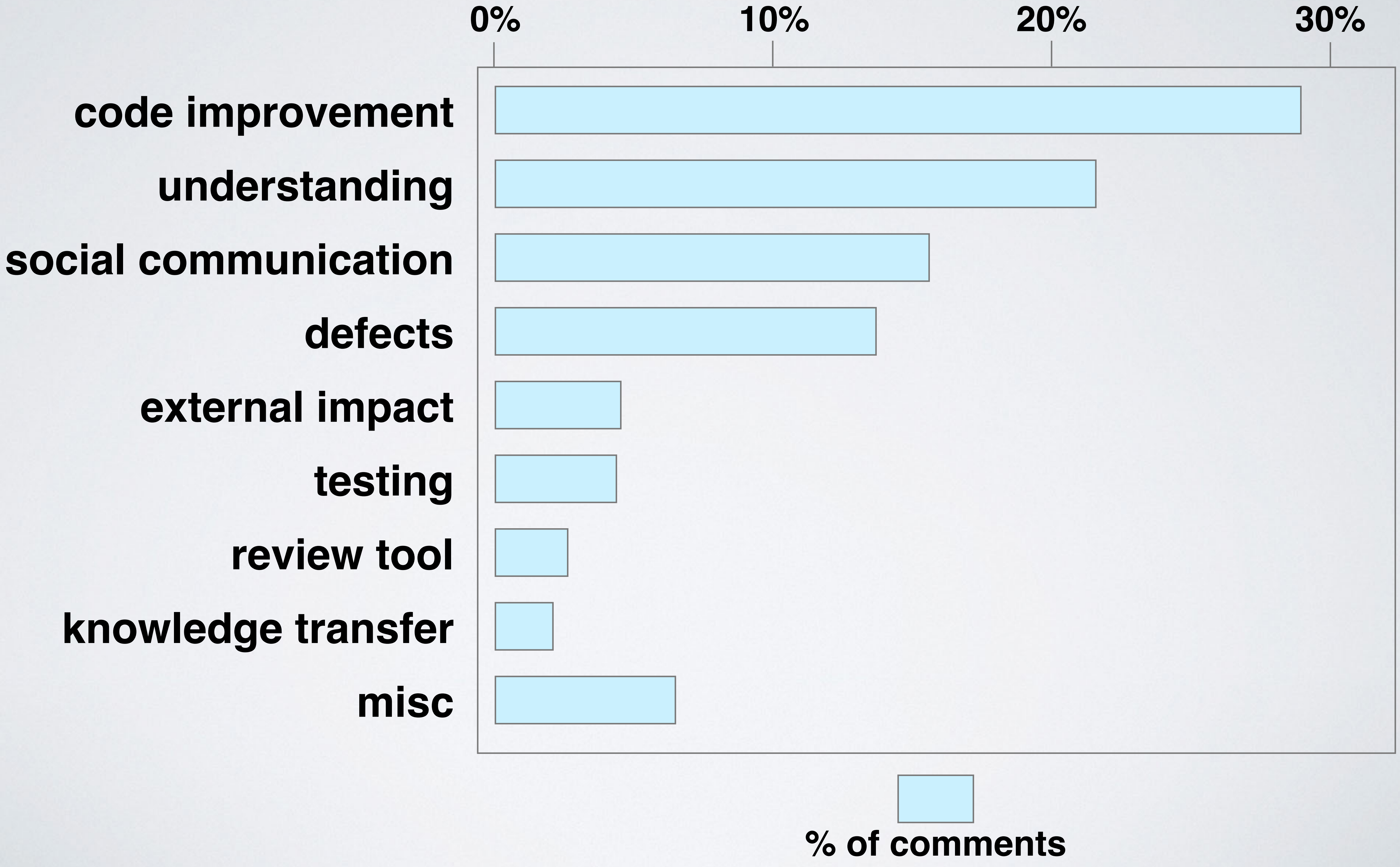


survey to 165 managers

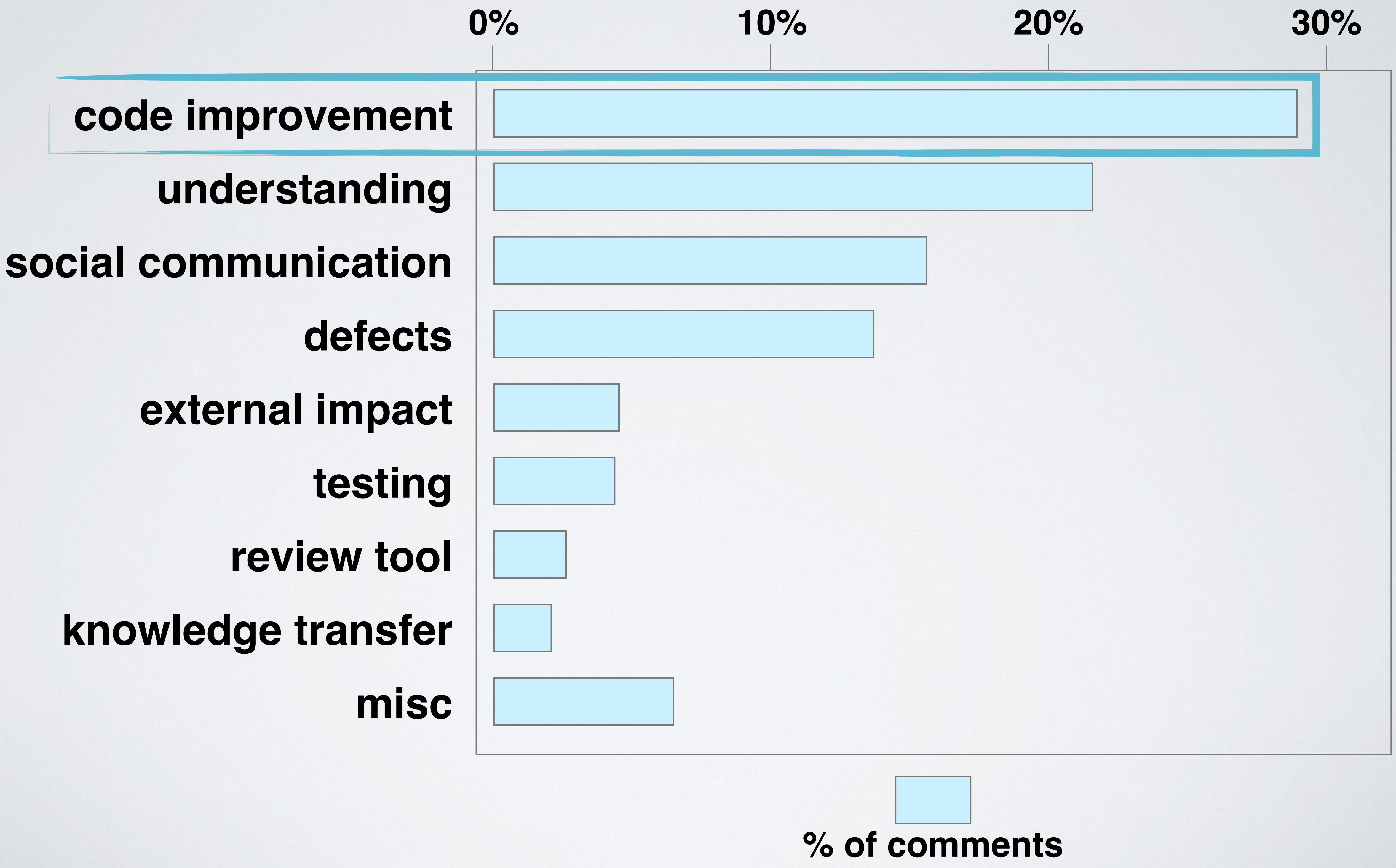
survey to 873 developers

classification of
570 review comments

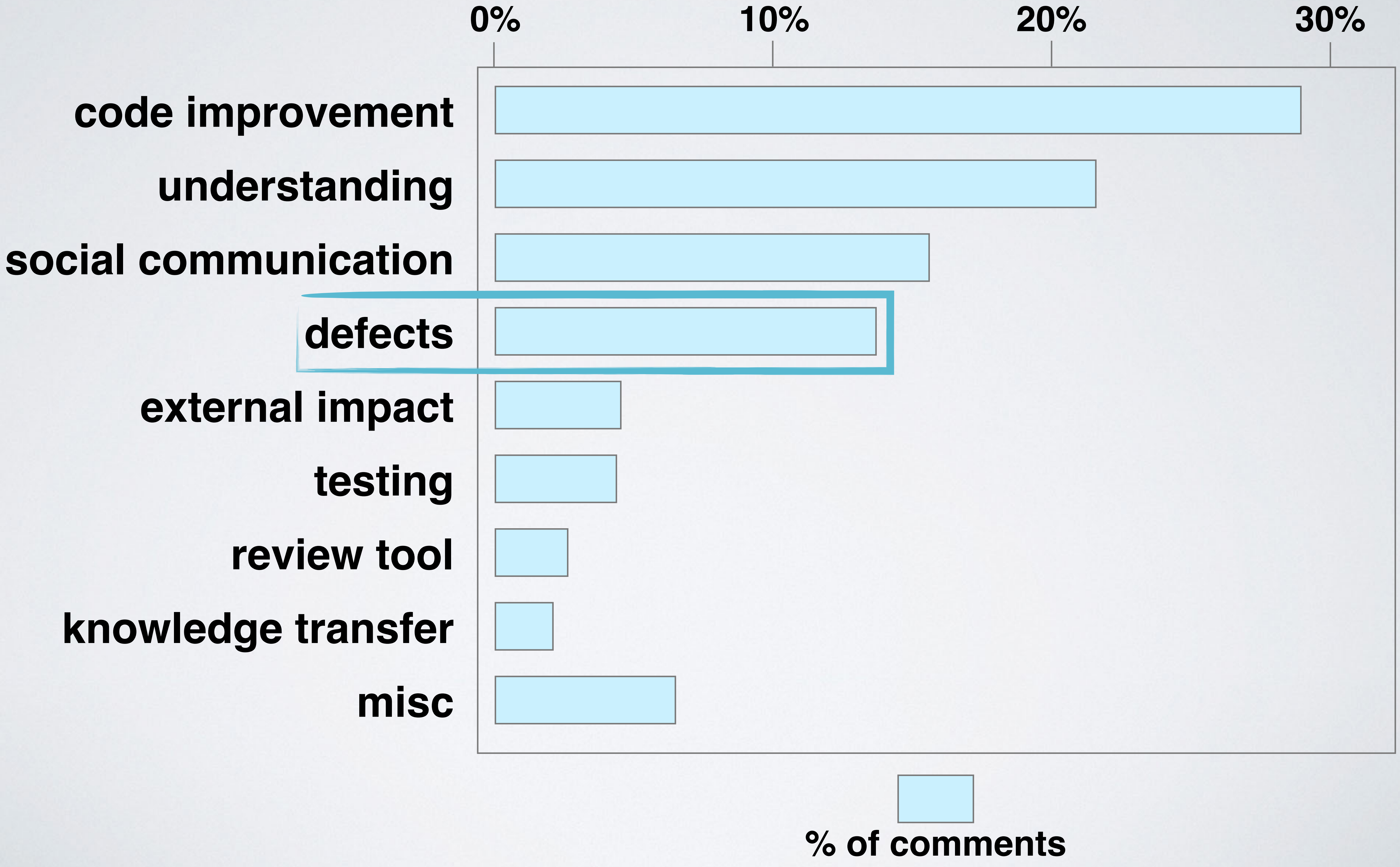
Results of review comments' analysis



Results of review comments' analysis



Results of review comments' analysis



Results of review comments’ analysis

“what if they are all used?”

“is it possible that this statement never match?”

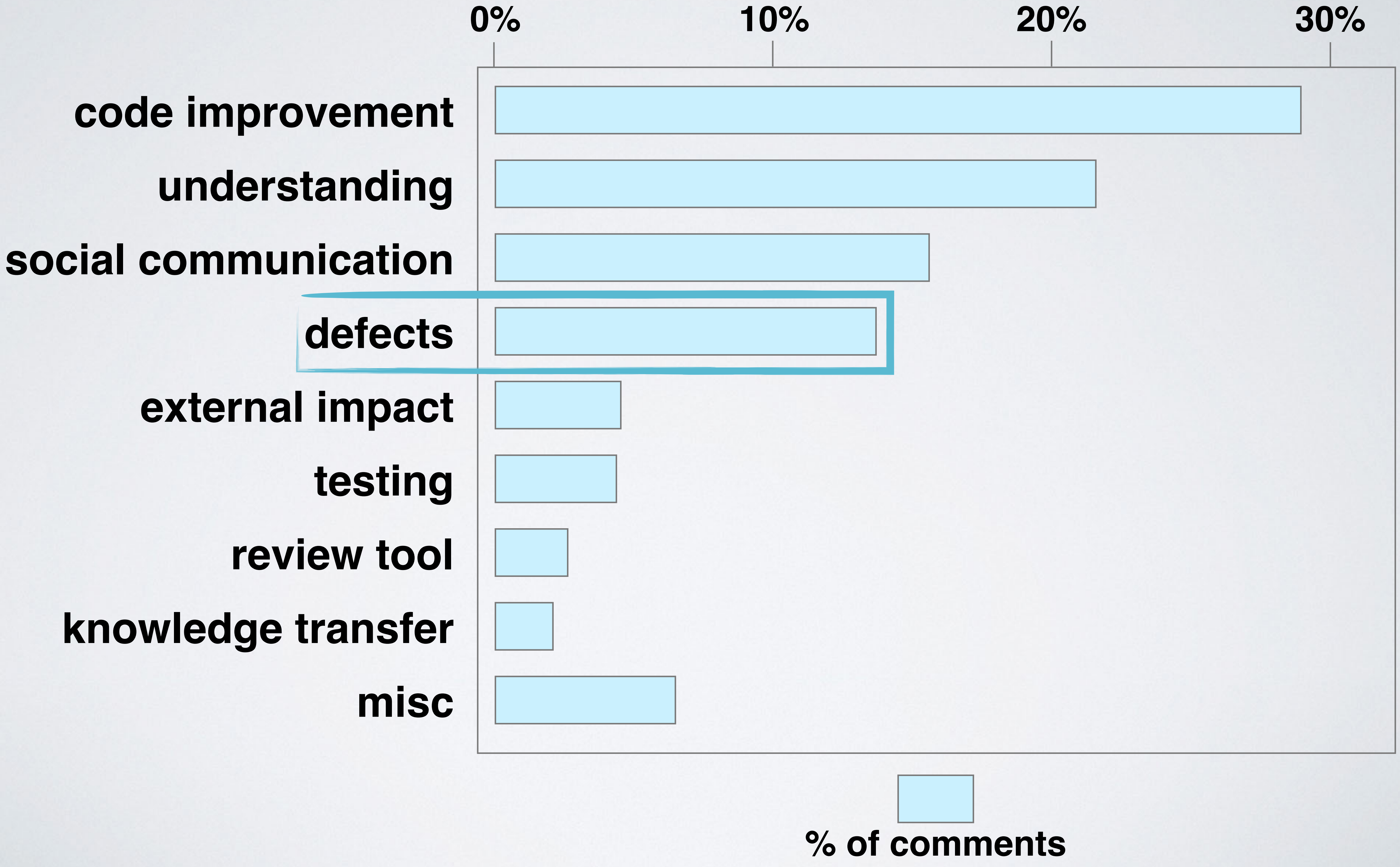
“should this end date be current date?”

“does it work if you put 0 here?”

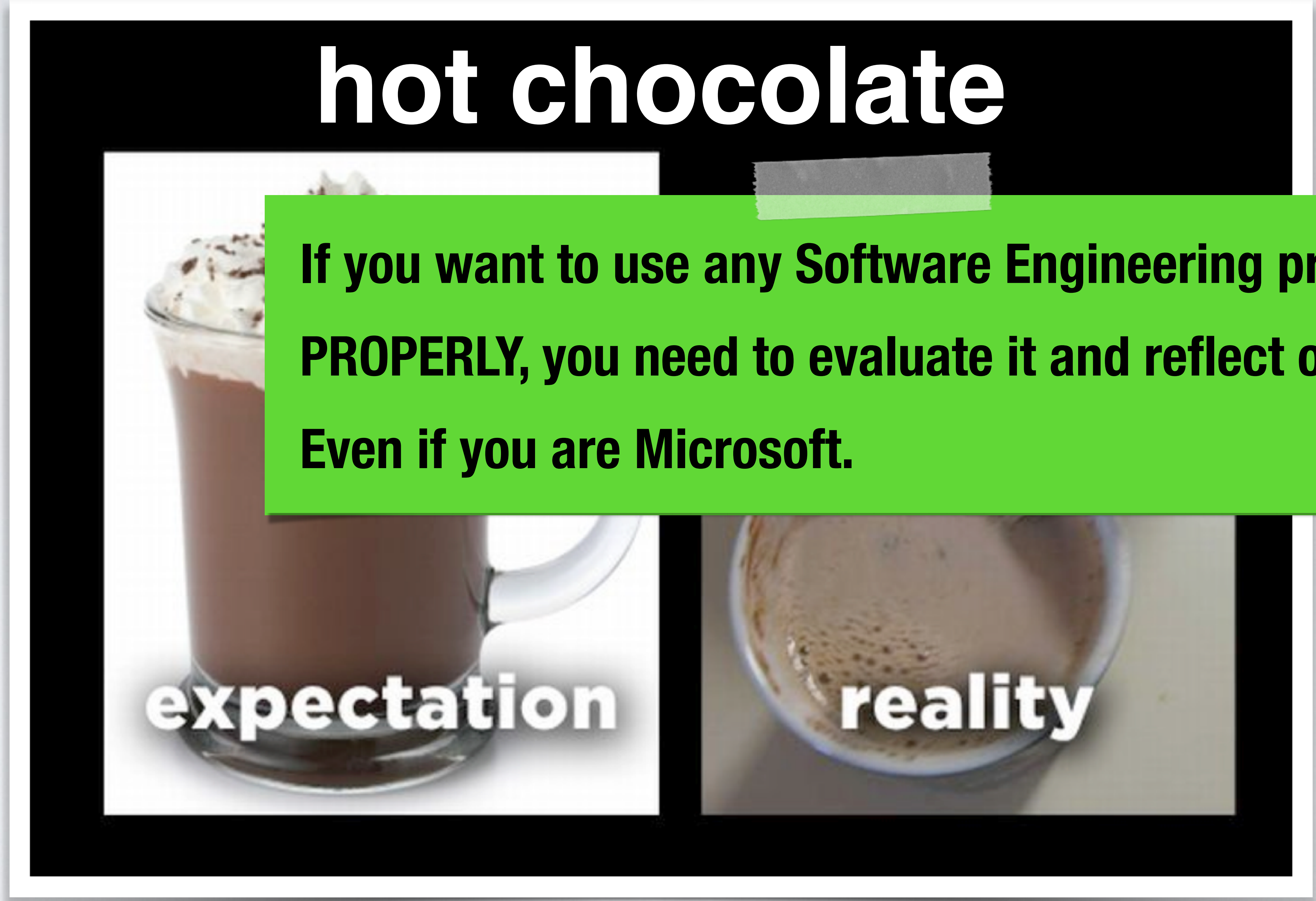
“any doubt about the precedence here?”

“should be &&?”

Results of review comments' analysis



Code review at Microsoft in 2012: Expectations vs. Reality



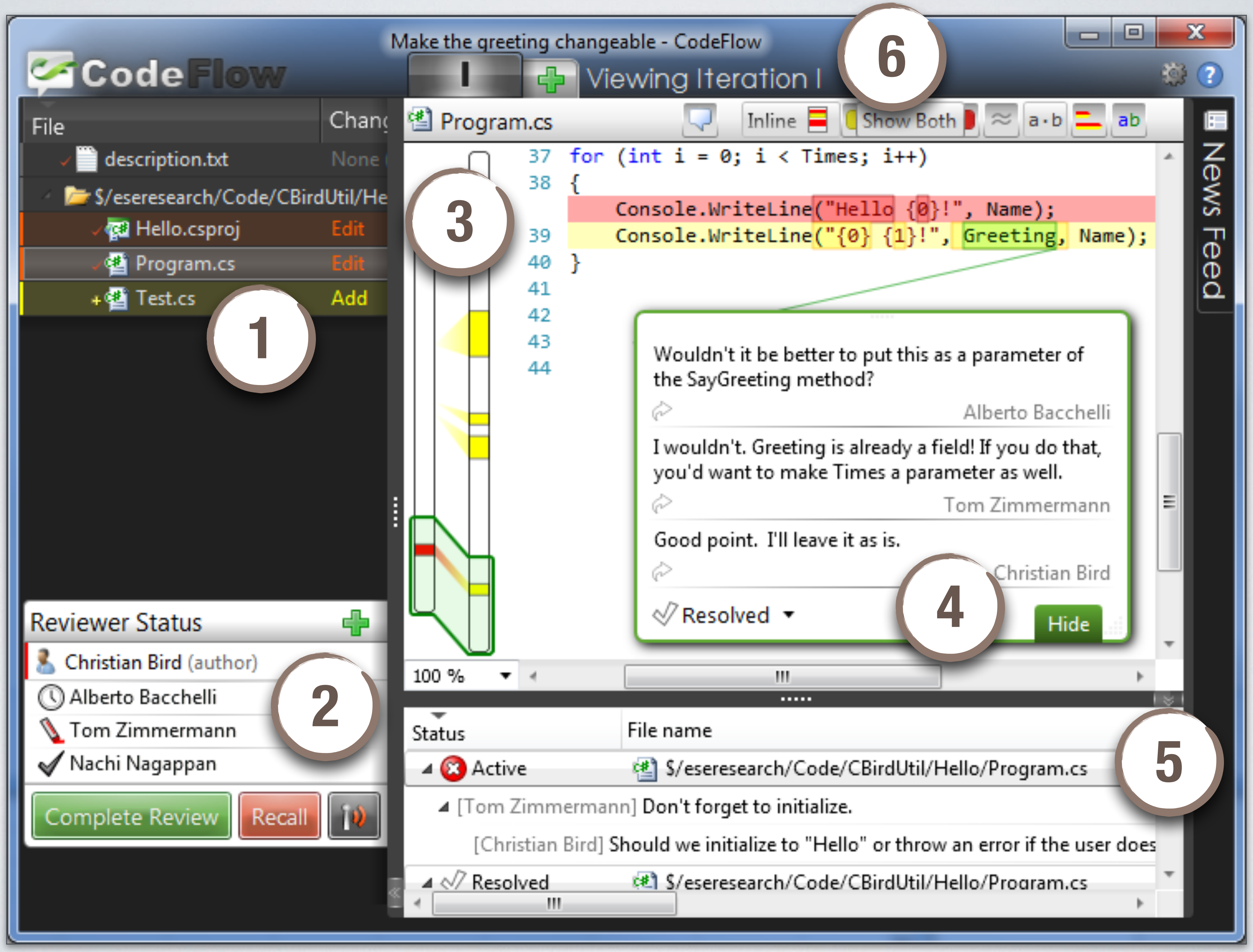
If you want to use any Software Engineering process PROPERLY, you need to evaluate it and reflect on it. Even if you are Microsoft.

Asking Uncomfortable Questions On Code Review

But.. why are we doing code review at all?

Our code review tools are great! Aren't they?

Modern code review tools: Microsoft CodeFlow



Make the greeting changeable - CodeFlow

Viewing Iteration I

File	Chang
description.txt	None
$/eseresearch/Code/CBirdUtil/He	
Hello.csproj	Edit
Program.cs	Edit
Test.cs	Add

Reviewer Status

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

Program.cs

```
for (int i = 0; i < Times; i++)
{
    Console.WriteLine("Hello {0}!", Name);
    Console.WriteLine("{0} {1}!", Greeting, Name);
}
```

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

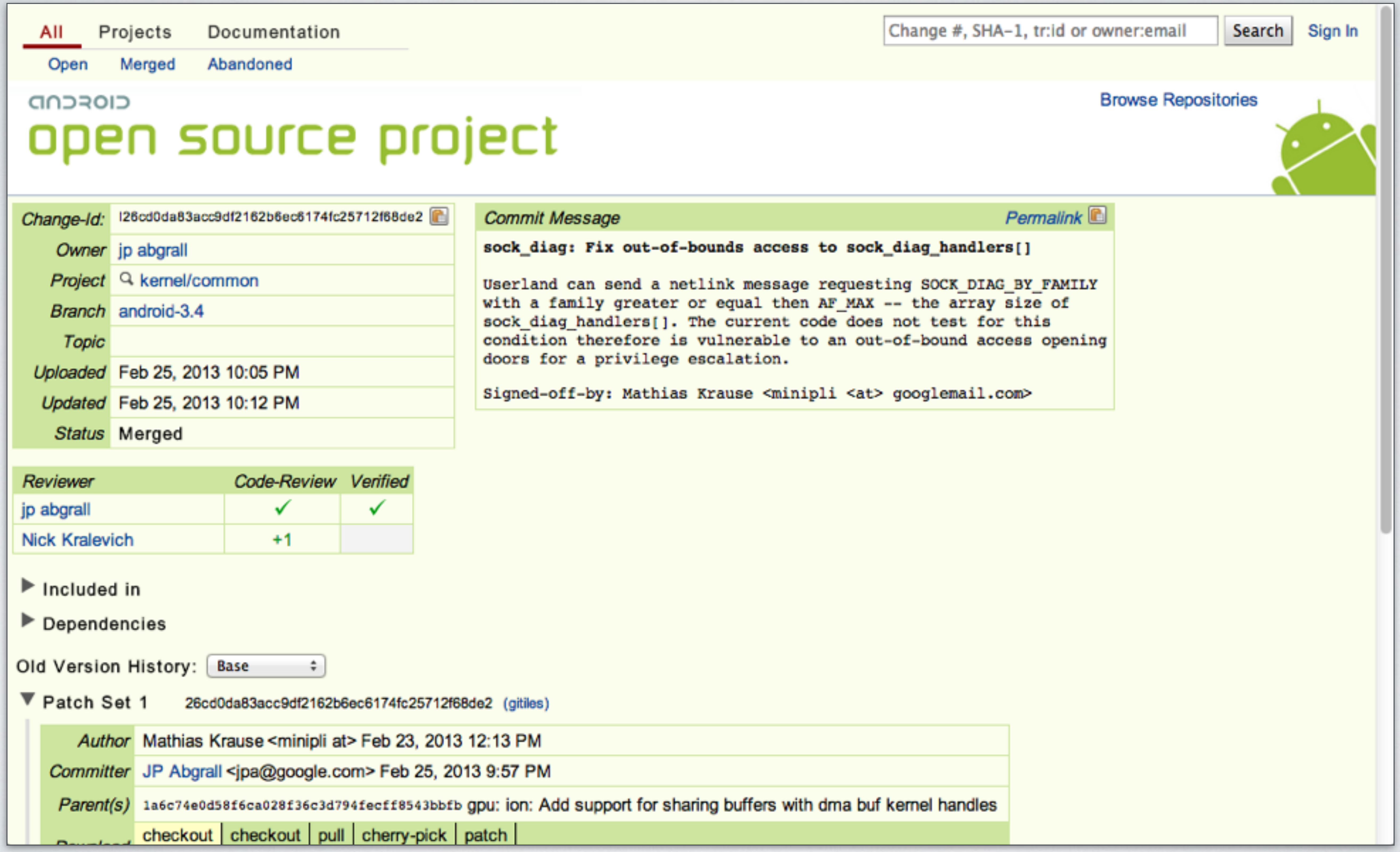
Good point. I'll leave it as is.

Christian Bird

Resolved

Status	File name
Active	$/eseresearch/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	$/eseresearch/Code/CBirdUtil/Hello/Program.cs

Modern code review tools: Gerrit



Change-Id:	I26cd0da83acc9df2162b6ec6174fc25712f68de2
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> googlemail.com>
```

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

▶ Included in

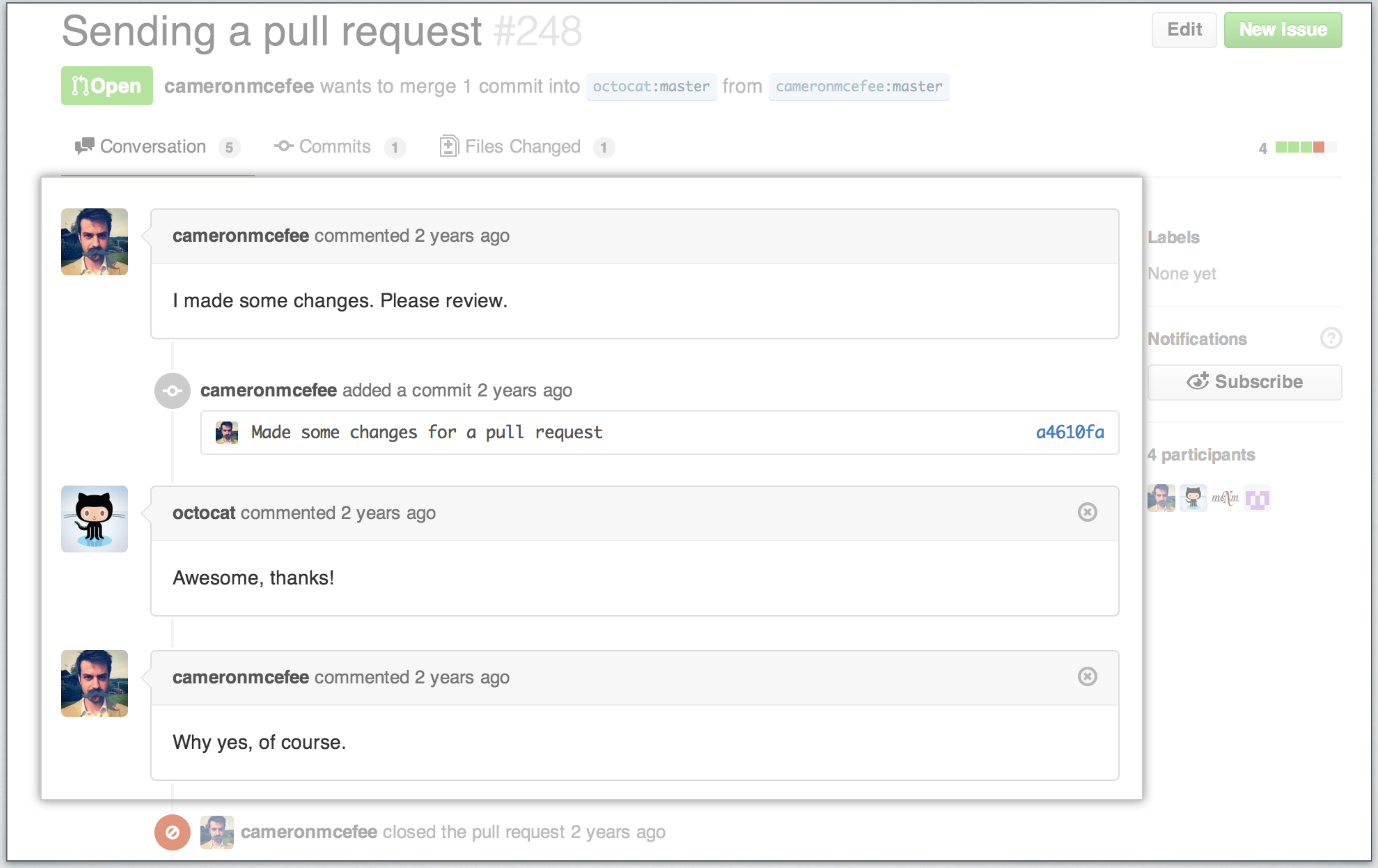
▶ Dependencies

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
Download	checkout | checkout | pull | cherry-pick | patch |

Modern code review tools: GitHub Pull Requests



Sending a pull request #248

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

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

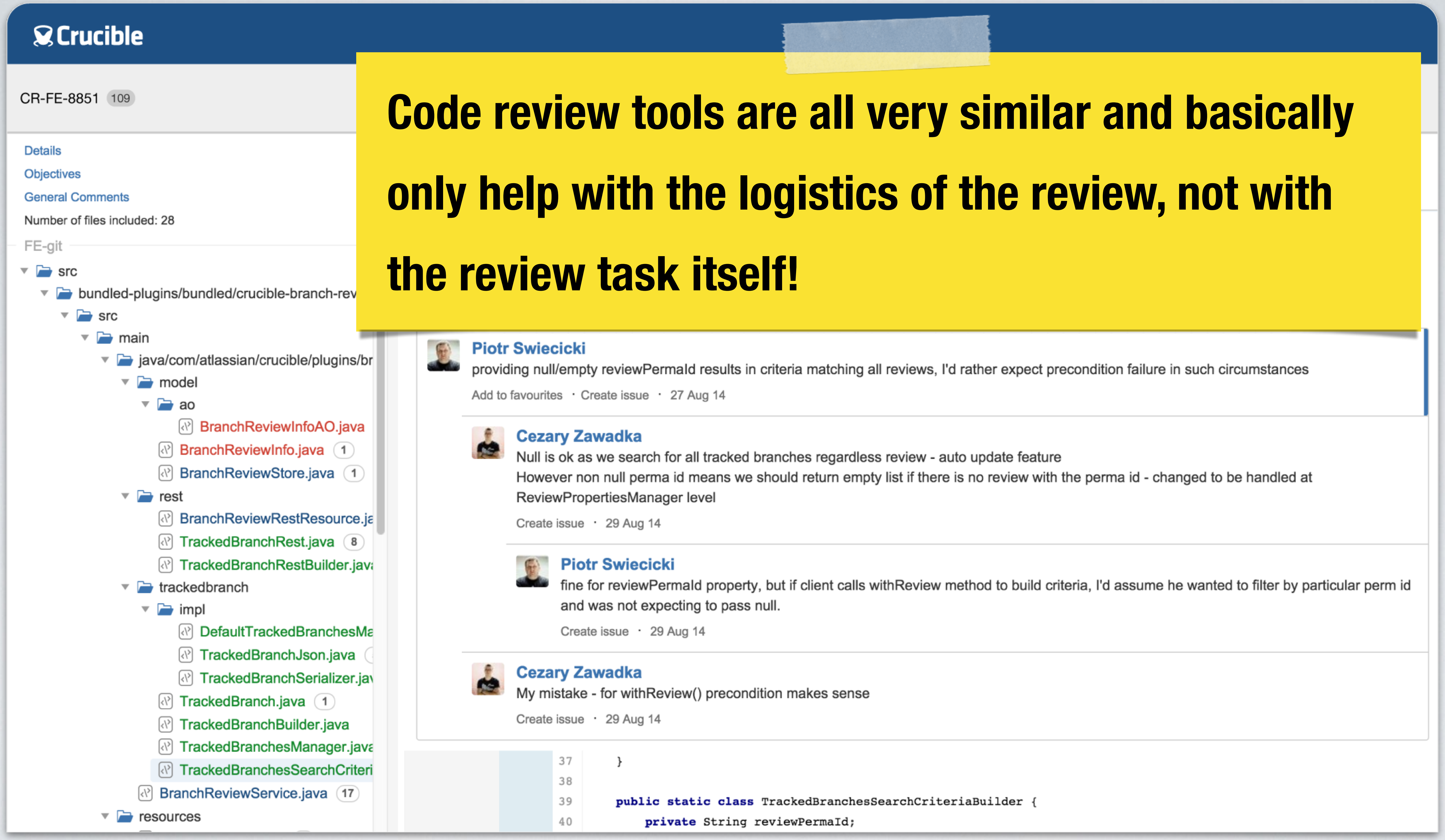
Labels

None yet

Notifications

4 participants

Modern code review tools: Atlassian Crucible



Crucible

CR-FE-8851 109

Details
Objectives
General Comments
Number of files included: 28

FE-git

- src
  - bundled-plugins/bundled/crucible-branch-rev
    - src
      - main
        - java/com/atlassian/crucible/plugins/br
          - model
            - ao
              - BranchReviewInfoAO.java
            - BranchReviewInfo.java 1
            - BranchReviewStore.java 1
          - rest
            - BranchReviewRestResource.ja
            - TrackedBranchRest.java 8
            - TrackedBranchRestBuilder.java
          - trackedbranch
            - impl
              - DefaultTrackedBranchesMa
              - TrackedBranchJson.java
              - TrackedBranchSerializer.jav
            - TrackedBranch.java 1
            - TrackedBranchBuilder.java
            - TrackedBranchesManager.java
            - TrackedBranchesSearchCriteri
          - BranchReviewService.java 17
      - resources

Piotr Swiecicki
providing null/empty reviewPermaId results in criteria matching all reviews, I'd rather expect precondition failure in such circumstances
Add to favourites · Create issue · 27 Aug 14

Cezary Zawadka
Null is ok as we search for all tracked branches regardless review - auto update feature
However non null perma id means we should return empty list if there is no review with the perma id - changed to be handled at ReviewPropertiesManager level
Create issue · 29 Aug 14

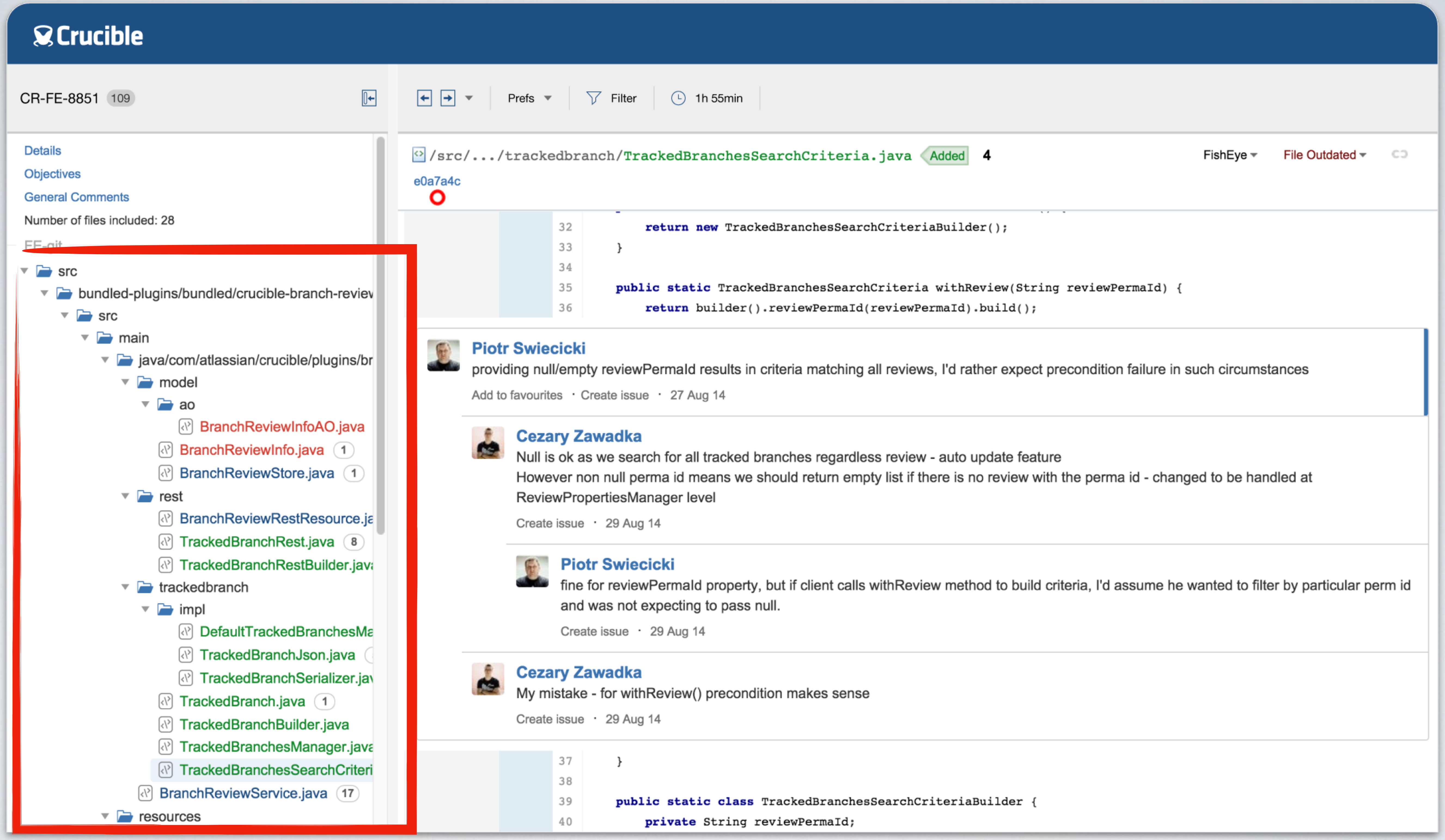
Piotr Swiecicki
fine for reviewPermaId property, but if client calls withReview method to build criteria, I'd assume he wanted to filter by particular perm id and was not expecting to pass null.
Create issue · 29 Aug 14

Cezary Zawadka
My mistake - for withReview() precondition makes sense
Create issue · 29 Aug 14

```
    }

    public static class TrackedBranchesSearchCriteriaBuilder {
        private String reviewPermaId;
```

Code review tools are all very similar and basically only help with the logistics of the review, not with the review task itself!

Modern code review tools: Atlassian Crucible



Crucible

CR-FE-8851 109

Prefs ▾ Filter 1h 55min

Details

Objectives

General Comments

Number of files included: 28

FE-git

- src
  - bundled-plugins/bundled/crucible-branch-reviev
    - src
      - main
        - java/com/atlassian/crucible/plugins/br
          - model
            - ao
              - BranchReviewInfoAO.java
            - BranchReviewInfo.java 1
            - BranchReviewStore.java 1
          - rest
            - BranchReviewRestResource.ja
            - TrackedBranchRest.java 8
            - TrackedBranchRestBuilder.java
          - trackedbranch
            - impl
              - DefaultTrackedBranchesMa
              - TrackedBranchJson.java
              - TrackedBranchSerializer.jav
            - TrackedBranch.java 1
            - TrackedBranchBuilder.java
            - TrackedBranchesManager.java
            - TrackedBranchesSearchCriteri
          - BranchReviewService.java 17
      - resources

/src/.../trackedbranch/TrackedBranchesSearchCriteria.java Added 4 FishEye ▾ File Outdated ▾

e0a7a4c

```
        return new TrackedBranchesSearchCriteriaBuilder();
    }

    public static TrackedBranchesSearchCriteria withReview(String reviewPermaId) {
        return builder().reviewPermaId(reviewPermaId).build();
```

Piotr Swiecicki
providing null/empty reviewPermaId results in criteria matching all reviews, I'd rather expect precondition failure in such circumstances
Add to favourites · Create issue · 27 Aug 14

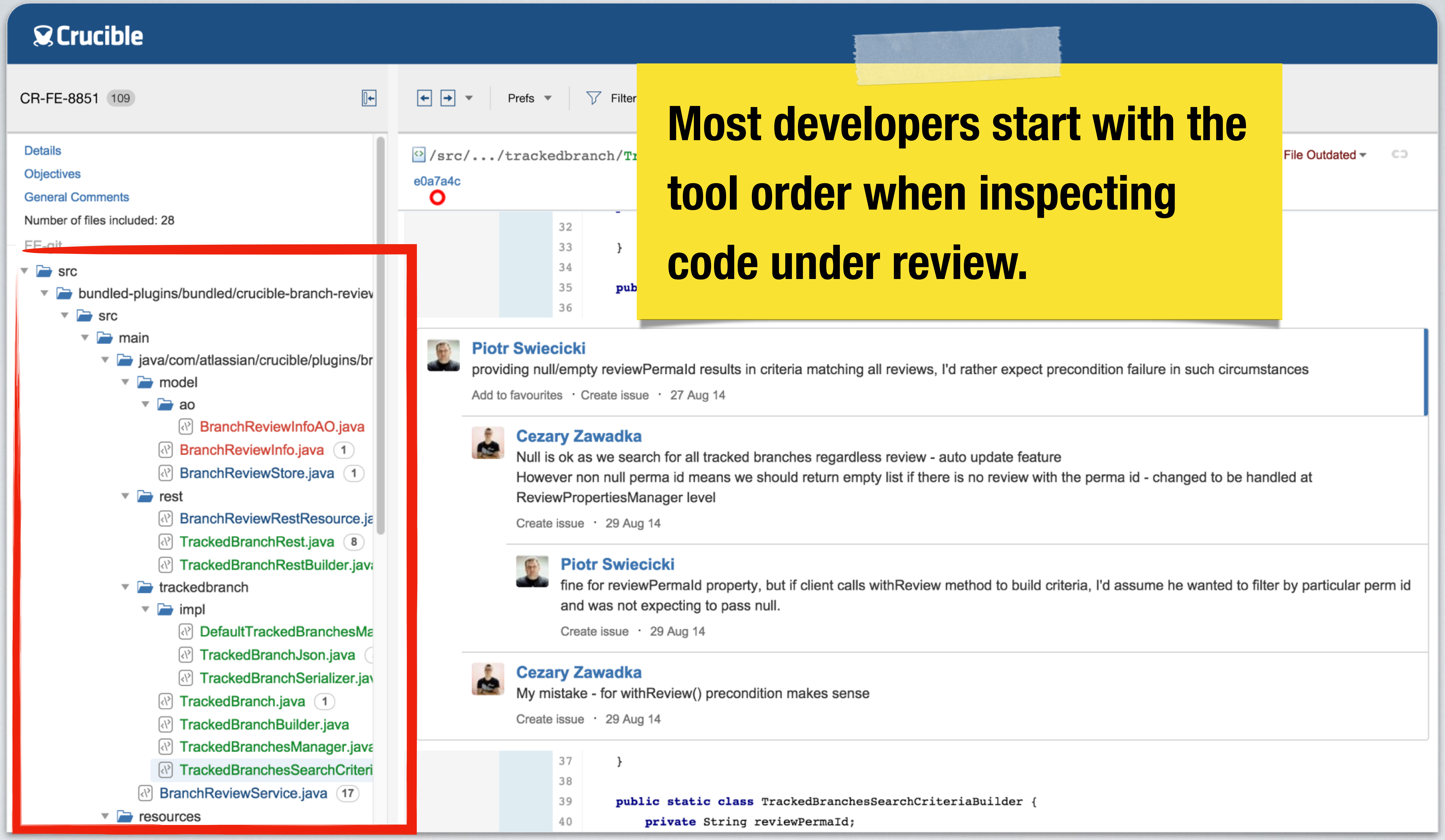
Cezary Zawadka
Null is ok as we search for all tracked branches regardless review - auto update feature
However non null perma id means we should return empty list if there is no review with the perma id - changed to be handled at ReviewPropertiesManager level
Create issue · 29 Aug 14

Piotr Swiecicki
fine for reviewPermaId property, but if client calls withReview method to build criteria, I'd assume he wanted to filter by particular perm id and was not expecting to pass null.
Create issue · 29 Aug 14

Cezary Zawadka
My mistake - for withReview() precondition makes sense
Create issue · 29 Aug 14

```
    }

    public static class TrackedBranchesSearchCriteriaBuilder {
        private String reviewPermaId;
```

Modern code review tools: Alphabetical Ordering Of Files



Details
Objectives
General Comments
Number of files included: 28

FE-git

- src
  - bundled-plugins/bundled/crucible-branch-review
    - src
      - main
        - java/com/atlassian/crucible/plugins/br
          - model
            - ao
              - BranchReviewInfoAO.java
            - BranchReviewInfo.java 1
            - BranchReviewStore.java 1
          - rest
            - BranchReviewRestResource.ja
            - TrackedBranchRest.java 8
            - TrackedBranchRestBuilder.java
          - trackedbranch
            - impl
              - DefaultTrackedBranchesMa
              - TrackedBranchJson.java
              - TrackedBranchSerializer.jav
            - TrackedBranch.java 1
            - TrackedBranchBuilder.java
            - TrackedBranchesManager.java
            - TrackedBranchesSearchCriteri
          - BranchReviewService.java 17
      - resources

/src/.../trackedbranch/Tr

e0a7a4c

```
    }

    pub
```

Piotr Swiecicki
providing null/empty reviewPermaId results in criteria matching all reviews, I'd rather expect precondition failure in such circumstances
Add to favourites · Create issue · 27 Aug 14

Cezary Zawadka
Null is ok as we search for all tracked branches regardless review - auto update feature
However non null perma id means we should return empty list if there is no review with the perma id - changed to be handled at ReviewPropertiesManager level
Create issue · 29 Aug 14

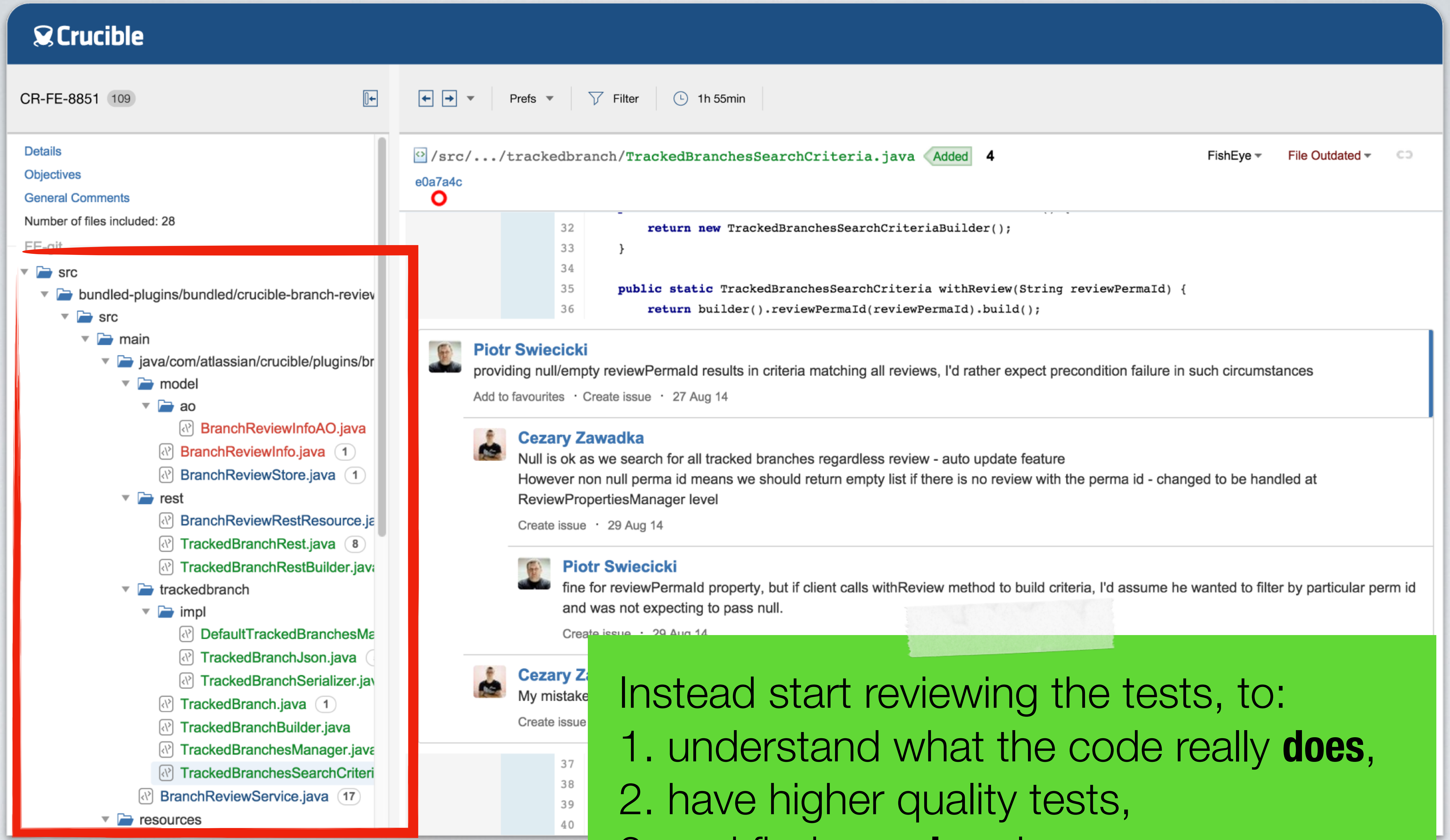
Piotr Swiecicki
fine for reviewPermaId property, but if client calls withReview method to build criteria, I'd assume he wanted to filter by particular perm id and was not expecting to pass null.
Create issue · 29 Aug 14

Cezary Zawadka
My mistake - for withReview() precondition makes sense
Create issue · 29 Aug 14

```
    }

    public static class TrackedBranchesSearchCriteriaBuilder {
        private String reviewPermaId;
```

Most developers start with the tool order when inspecting code under review.

Modern code review tools: Alphabetical Ordering Of Files



Number of files included: 28

FE-git

- src
  - bundled-plugins/bundled/crucible-branch-reviev
    - src
      - main
        - java/com/atlassian/crucible/plugins/br
          - model
            - ao
              - BranchReviewInfoAO.java
            - BranchReviewInfo.java 1
            - BranchReviewStore.java 1
          - rest
            - BranchReviewRestResource.ja
            - TrackedBranchRest.java 8
            - TrackedBranchRestBuilder.java
          - trackedbranch
            - impl
              - DefaultTrackedBranchesMa
              - TrackedBranchJson.java
              - TrackedBranchSerializer.jav
            - TrackedBranch.java 1
            - TrackedBranchBuilder.java
            - TrackedBranchesManager.java
            - TrackedBranchesSearchCriteri
          - BranchReviewService.java 17
        - resources

/src/.../trackedbranch/TrackedBranchesSearchCriteria.java Added 4

e0a7a4c

```
        return new TrackedBranchesSearchCriteriaBuilder();
    }

    public static TrackedBranchesSearchCriteria withReview(String reviewPermaId) {
        return builder().reviewPermaId(reviewPermaId).build();
```

Piotr Swiecicki
providing null/empty reviewPermaId results in criteria matching all reviews, I'd rather expect precondition failure in such circumstances
Add to favourites · Create issue · 27 Aug 14

Cezary Zawadka
Null is ok as we search for all tracked branches regardless review - auto update feature
However non null perma id means we should return empty list if there is no review with the perma id - changed to be handled at ReviewPropertiesManager level
Create issue · 29 Aug 14

Piotr Swiecicki
fine for reviewPermaId property, but if client calls withReview method to build criteria, I'd assume he wanted to filter by particular perm id and was not expecting to pass null.
Create issue · 29 Aug 14

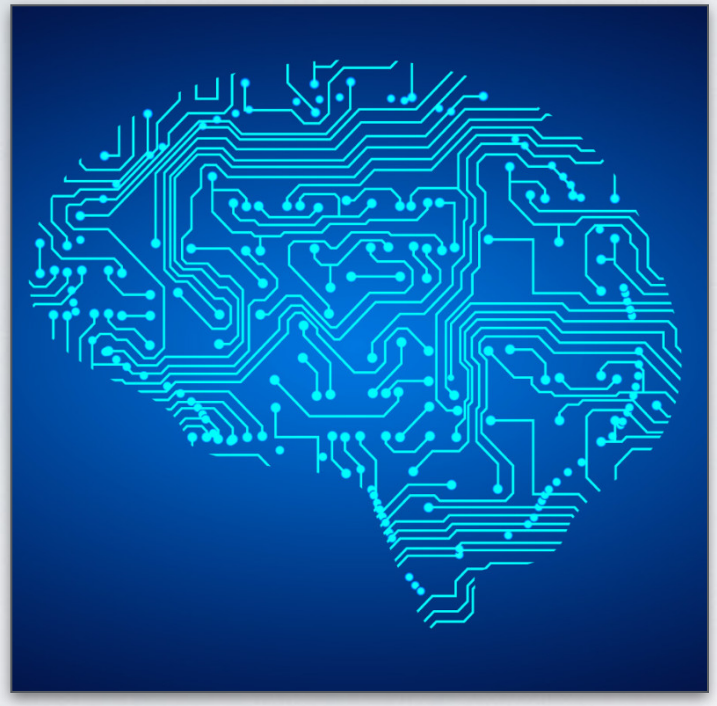
Cezary Za
My mistake
Create issue

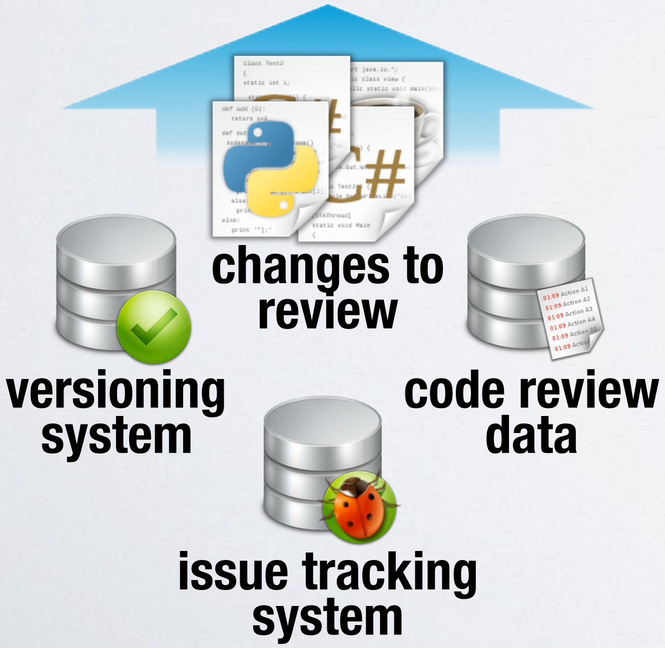
Instead start reviewing the tests, to:

1. understand what the code really **does**,
2. have higher quality tests,
3. and find **more bugs**!

Next generation code review tools: Risk-guided Code Review

risk detector



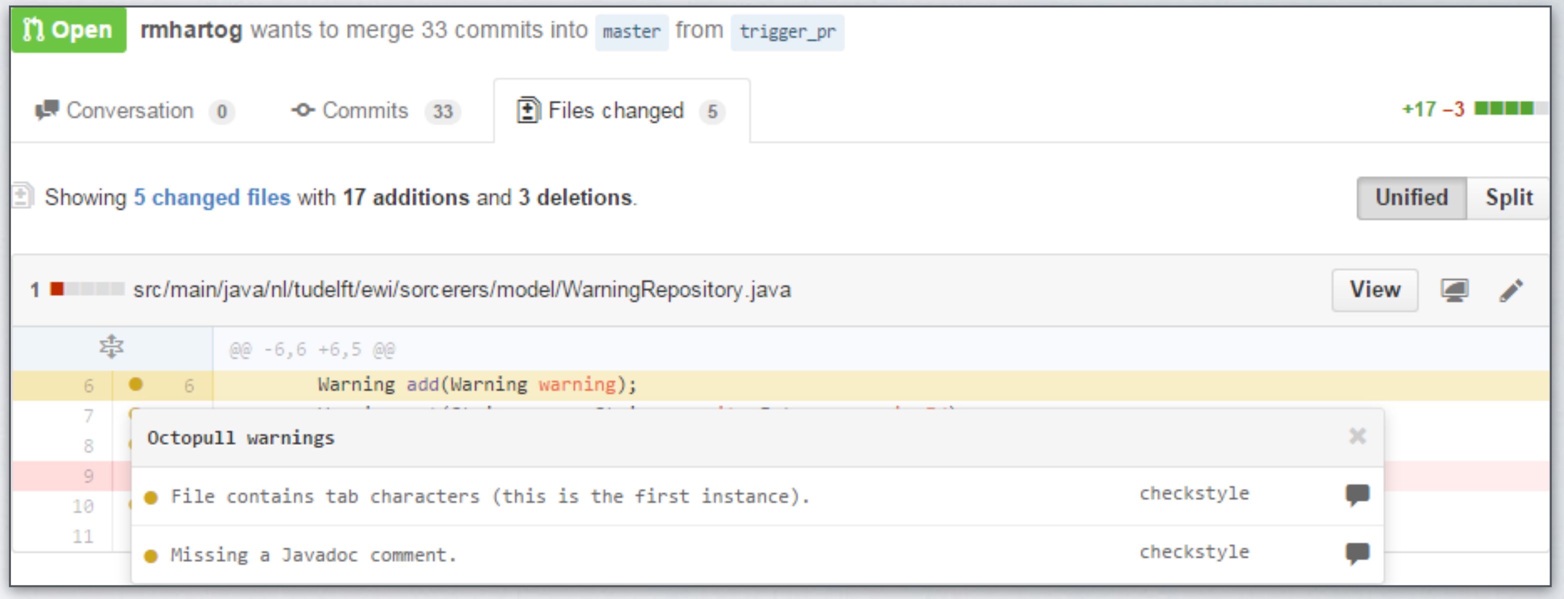


changes to review

versioning system

code review data

issue tracking system



rmhartog wants to merge 33 commits into master from trigger_pr

Showing 5 changed files with 17 additions and 3 deletions.

1 src/main/java/nl/tudelft/ewi/sorcerers/model/WarningRepository.java

```
@@ -6,6 +6,5 @@
        Warning add(Warning warning);
```

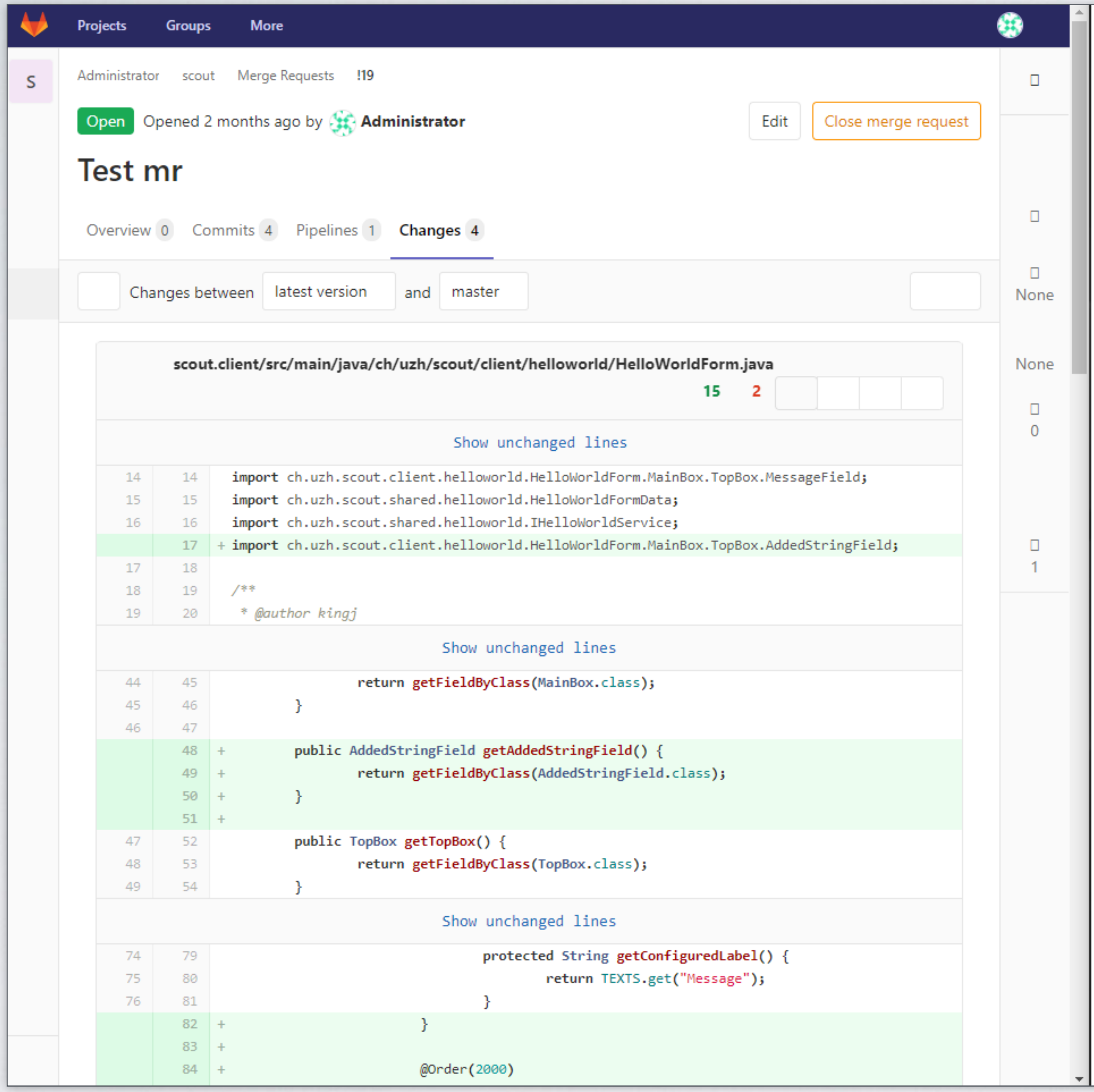
Octopull warnings

- File contains tab characters (this is the first instance). checkstyle
- Missing a Javadoc comment. checkstyle

In-line warnings

Re-ordered files

Next generation code review tools: Code Change Visualization



Open Opened 2 months ago by Administrator

Test mr

Changes between latest version and master

scout.client/src/main/java/ch/uzh/scout/client/helloworld/HelloWorldForm.java

15 2

Show unchanged lines

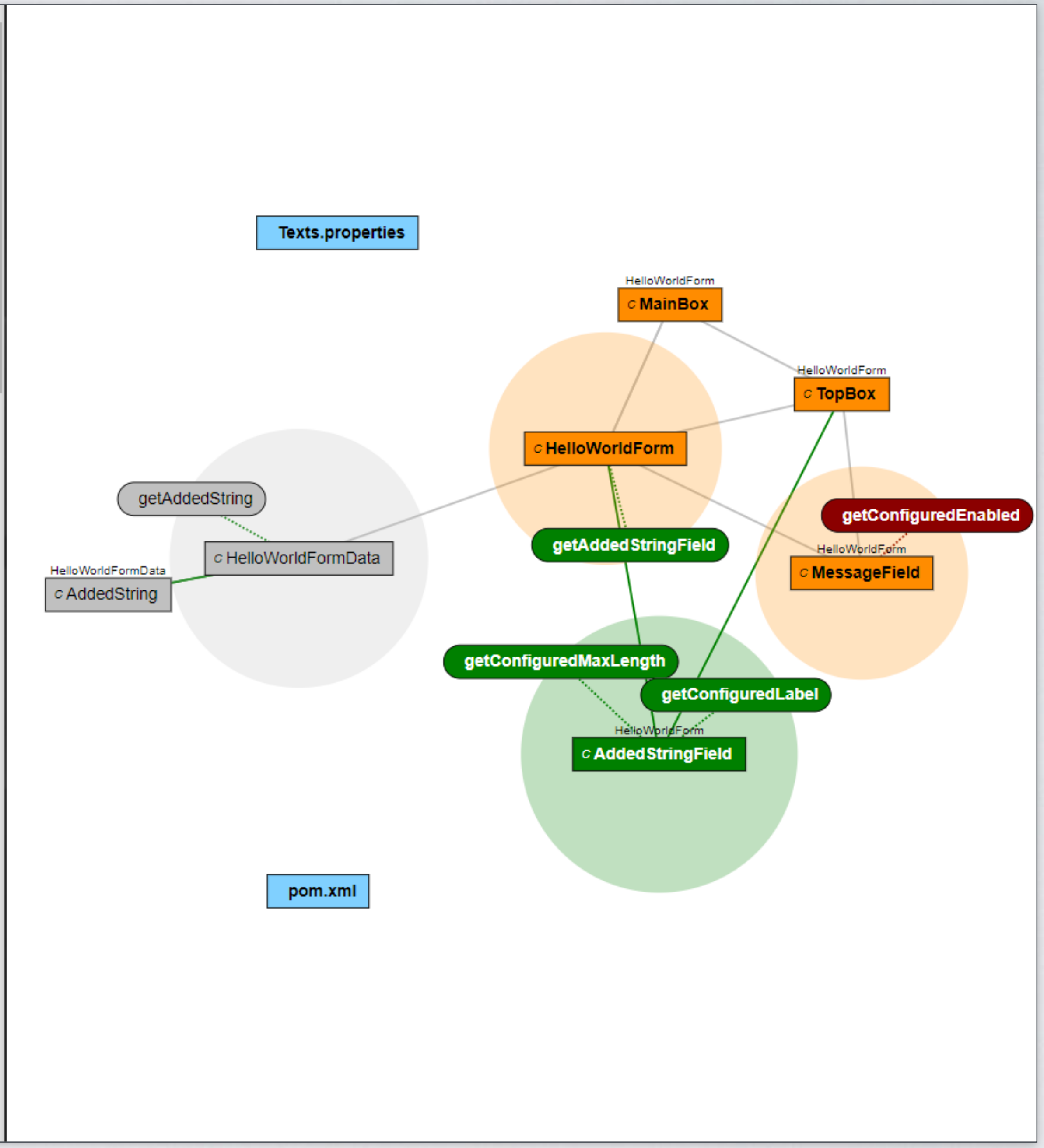
```
  import ch.uzh.scout.client.helloworld.HelloWorldForm.MainBox.TopBox.MessageField;
  import ch.uzh.scout.shared.helloworld.HelloWorldFormData;
  import ch.uzh.scout.shared.helloworld.IHelloWorldService;
+ import ch.uzh.scout.client.helloworld.HelloWorldForm.MainBox.TopBox.AddedStringField;

  /**
   * @author kingj
```

Show unchanged lines

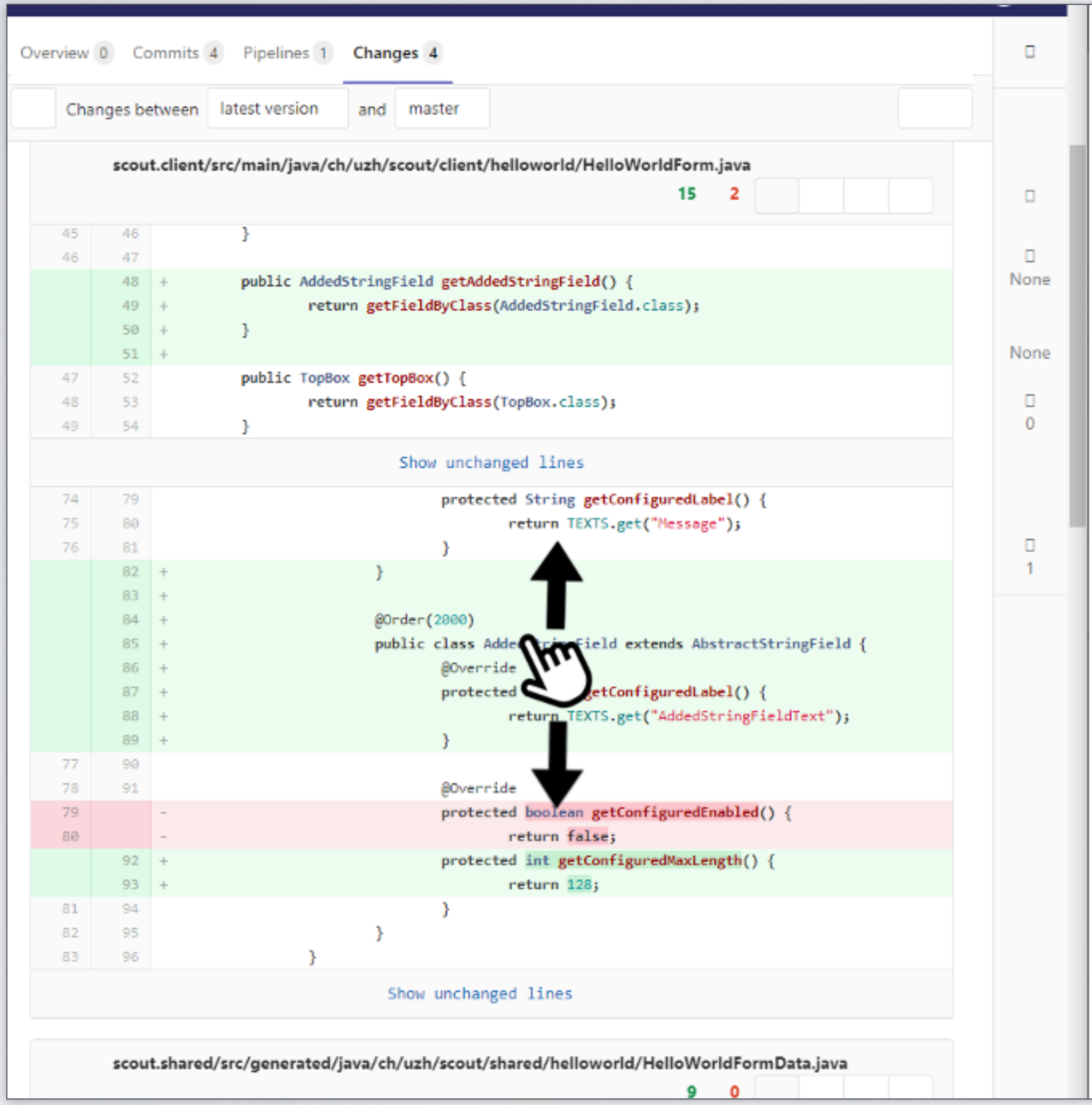
```
                return getFieldByClass(MainBox.class);
        }

+       public AddedStringField getAddedStringField() {
+               return getFieldByClass(AddedStringField.class);
+       }
+
        public TopBox getTopBox() {
                return getFieldByClass(TopBox.class);
        }
```

Show unchanged lines

```
                                protected String getConfiguredLabel() {
                                        return TEXTS.get("Message");
                                }
+                       }
+
+                       @Order(2000)
```



Next generation code review tools: Code Change Visualization



scout.client/src/main/java/ch/uzh/scout/client/helloworld/HelloWorldForm.java

15 2

```
              }

+             public AddedStringField getAddedStringField() {
+                     return getFieldByClass(AddedStringField.class);
+             }
+
              public TopBox getTopBox() {
                      return getFieldByClass(TopBox.class);
              }
```

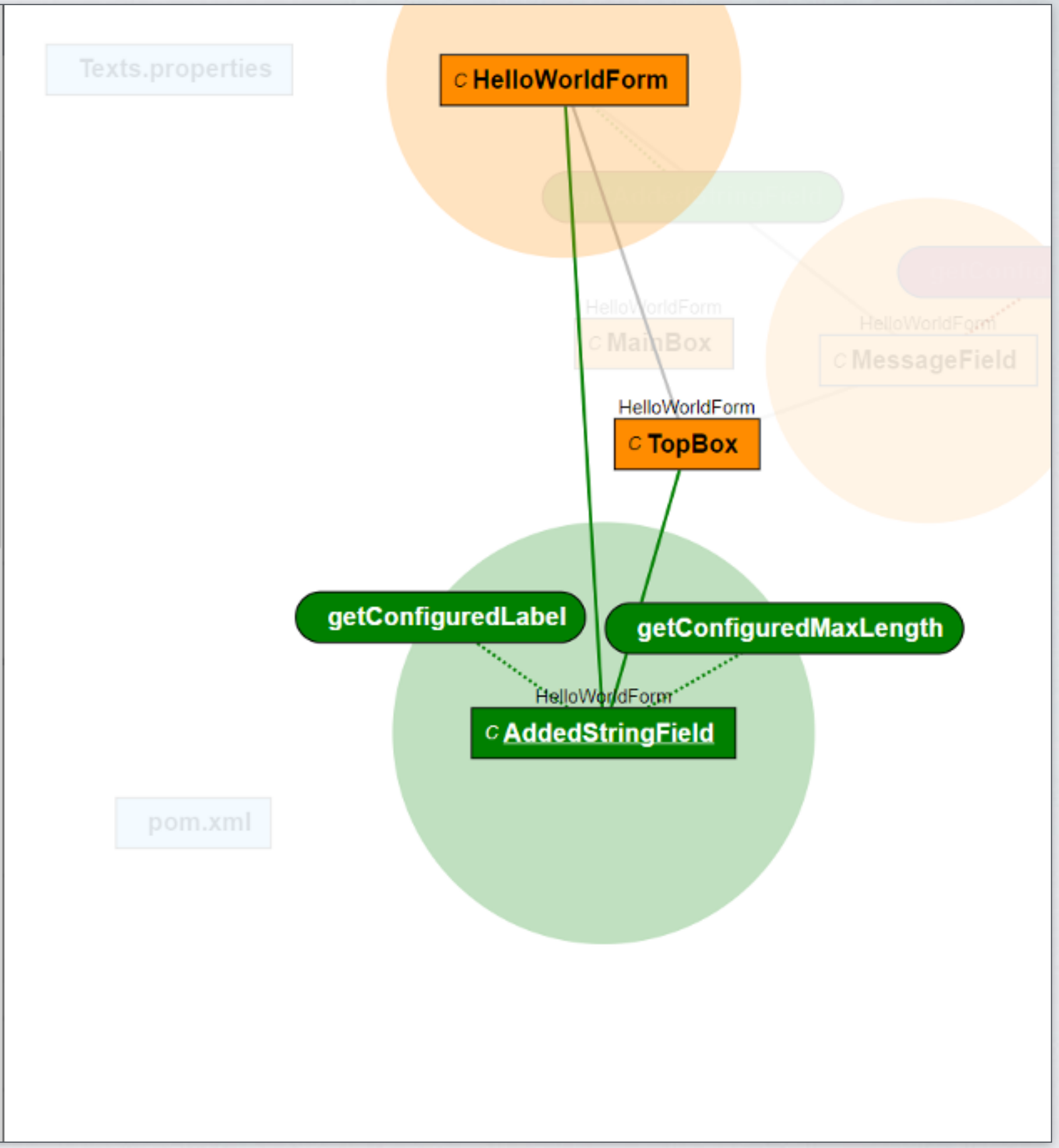
Show unchanged lines

```
                              protected String getConfiguredLabel() {
                                      return TEXTS.get("Message");
                              }
+                     }
+
+                     @Order(2000)
+                     public class AddedStringField extends AbstractStringField {
+                             @Override
+                             protected [illegible] getConfiguredLabel() {
+                                     return TEXTS.get("AddedStringFieldText");
+                             }

                              @Override
-                             protected boolean getConfiguredEnabled() {
-                                     return false;
+                             protected int getConfiguredMaxLength() {
+                                     return 128;
                              }
                      }
              }
```

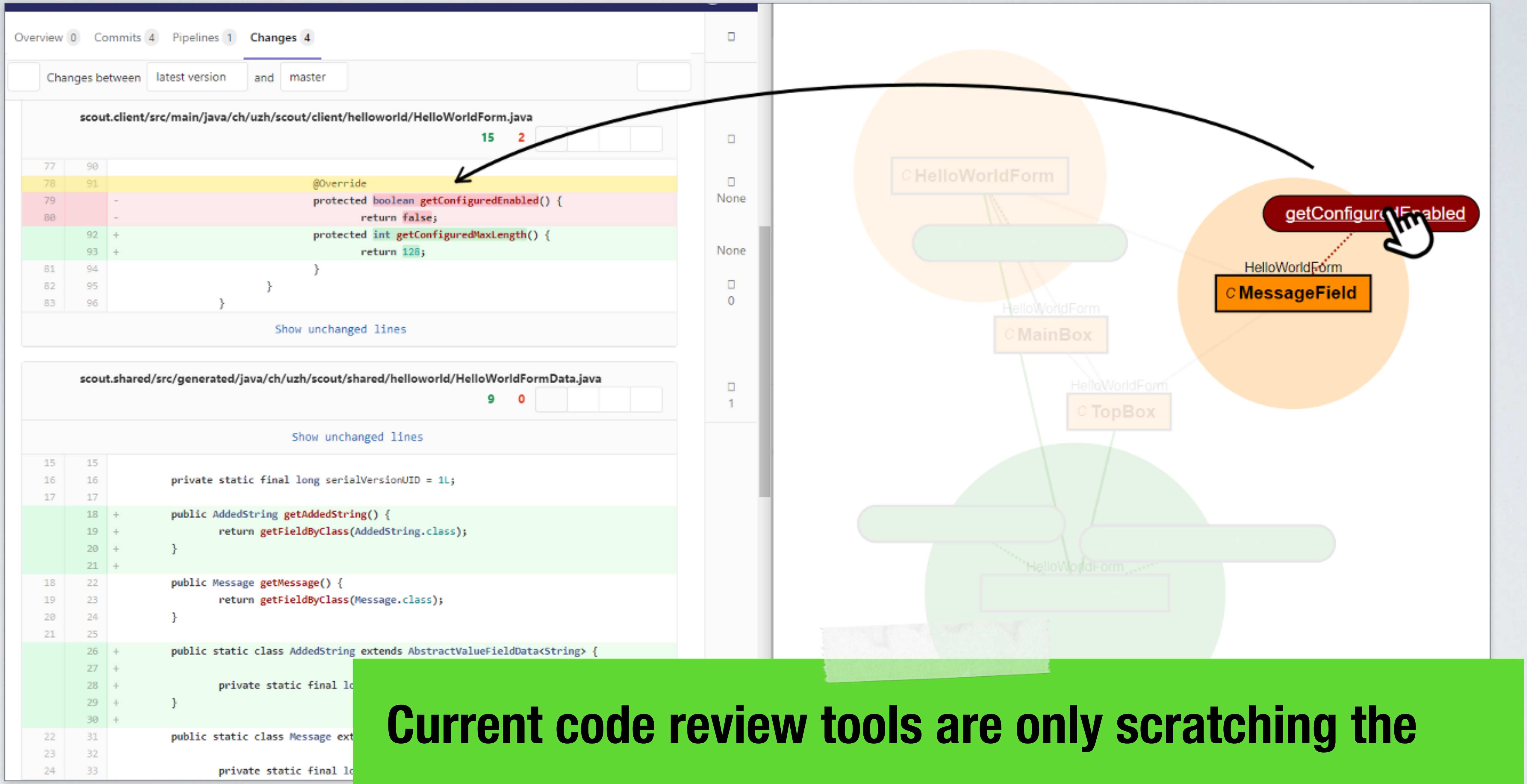
Show unchanged lines

scout.shared/src/generated/java/ch/uzh/scout/shared/helloworld/HelloWorldFormData.java

9 0



Next generation code review tools: Code Change Visualization



Overview 0 Commits 4 Pipelines 1 Changes 4

Changes between latest version and master

scout.client/src/main/java/ch/uzh/scout/client/helloworld/HelloWorldForm.java 15 2

```
          @Override
-         protected boolean getConfiguredEnabled() {
-             return false;
+         protected int getConfiguredMaxLength() {
+             return 128;
          }
      }
  }
```

Show unchanged lines

scout.shared/src/generated/java/ch/uzh/scout/shared/helloworld/HelloWorldFormData.java 9 0

Show unchanged lines

```

      private static final long serialVersionUID = 1L;

+     public AddedString getAddedString() {
+         return getFieldByClass(AddedString.class);
+     }
+
      public Message getMessage() {
          return getFieldByClass(Message.class);
      }

+     public static class AddedString extends AbstractValueFieldData<String> {
+
+         private static final lo
+     }
+
      public static class Message ext

          private static final lo
```

Current code review tools are only scratching the surface of what can be done to support reviewers.

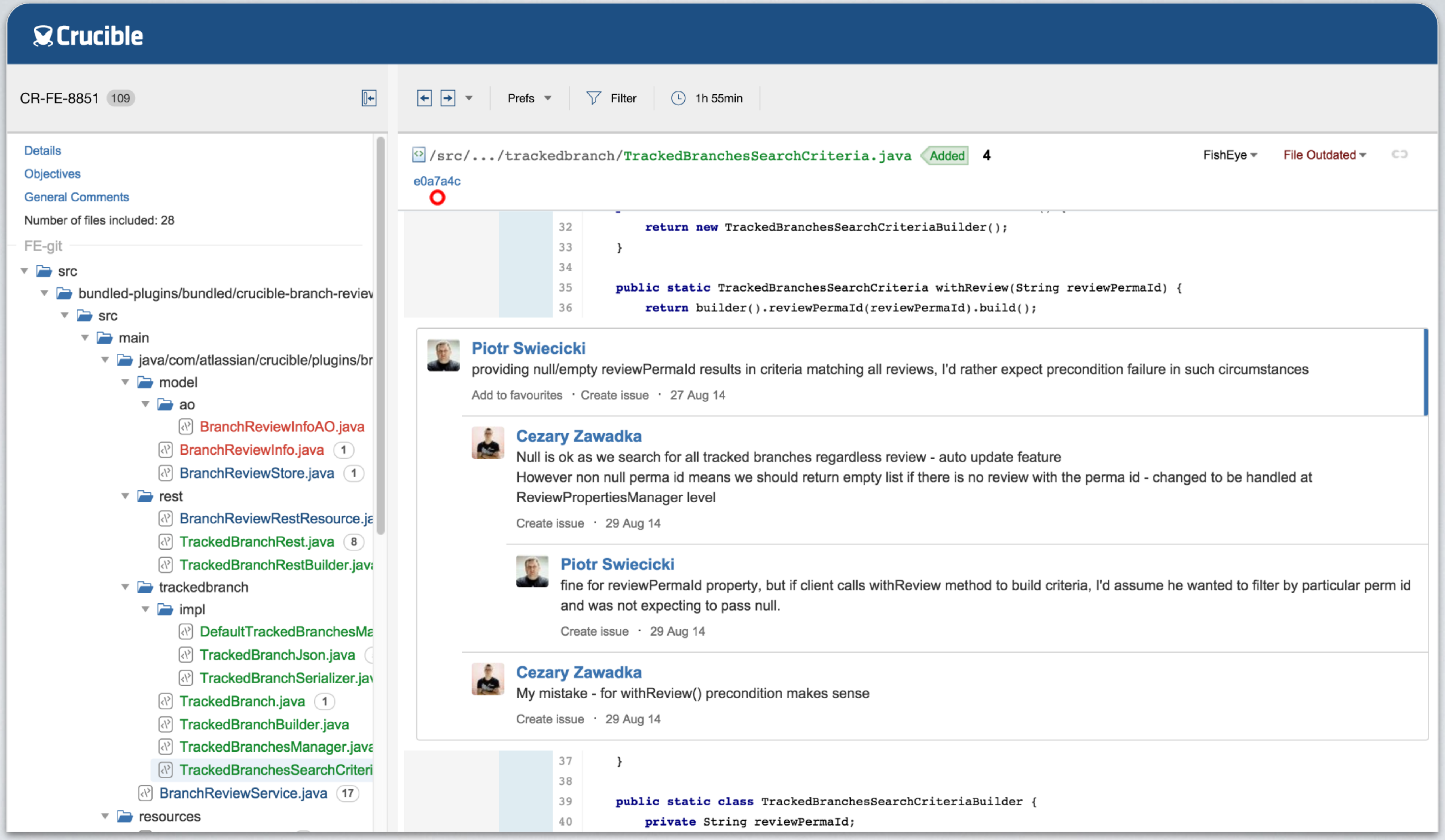
Asking Uncomfortable Questions On Code Review

But.. why are we doing code review at all?

Our code review tools are great! Aren't they?

Are we really in this together?

Who should review a change?



/src/.../trackedbranch/TrackedBranchesSearchCriteria.java Added 4

e0a7a4c

Details

Objectives

General Comments

Number of files included: 28

FE-git

- src
  - bundled-plugins/bundled/crucible-branch-reviev
    - src
      - main
        - java/com/atlassian/crucible/plugins/br
          - model
            - ao
              - BranchReviewInfoAO.java
            - BranchReviewInfo.java 1
            - BranchReviewStore.java 1
          - rest
            - BranchReviewRestResource.ja
            - TrackedBranchRest.java 8
            - TrackedBranchRestBuilder.java
          - trackedbranch
            - impl
              - DefaultTrackedBranchesMa
              - TrackedBranchJson.java
              - TrackedBranchSerializer.jav
            - TrackedBranch.java 1
            - TrackedBranchBuilder.java
            - TrackedBranchesManager.java
            - TrackedBranchesSearchCriteri
          - BranchReviewService.java 17
      - resources

```
        return new TrackedBranchesSearchCriteriaBuilder();
    }

    public static TrackedBranchesSearchCriteria withReview(String reviewPermaId) {
        return builder().reviewPermaId(reviewPermaId).build();
```

Piotr Swiecicki

providing null/empty reviewPermaId results in criteria matching all reviews, I'd rather expect precondition failure in such circumstances

Add to favourites · Create issue · 27 Aug 14

Cezary Zawadka

Null is ok as we search for all tracked branches regardless review - auto update feature
However non null perma id means we should return empty list if there is no review with the perma id - changed to be handled at ReviewPropertiesManager level

Create issue · 29 Aug 14

Piotr Swiecicki

fine for reviewPermaId property, but if client calls withReview method to build criteria, I'd assume he wanted to filter by particular perm id and was not expecting to pass null.

Create issue · 29 Aug 14

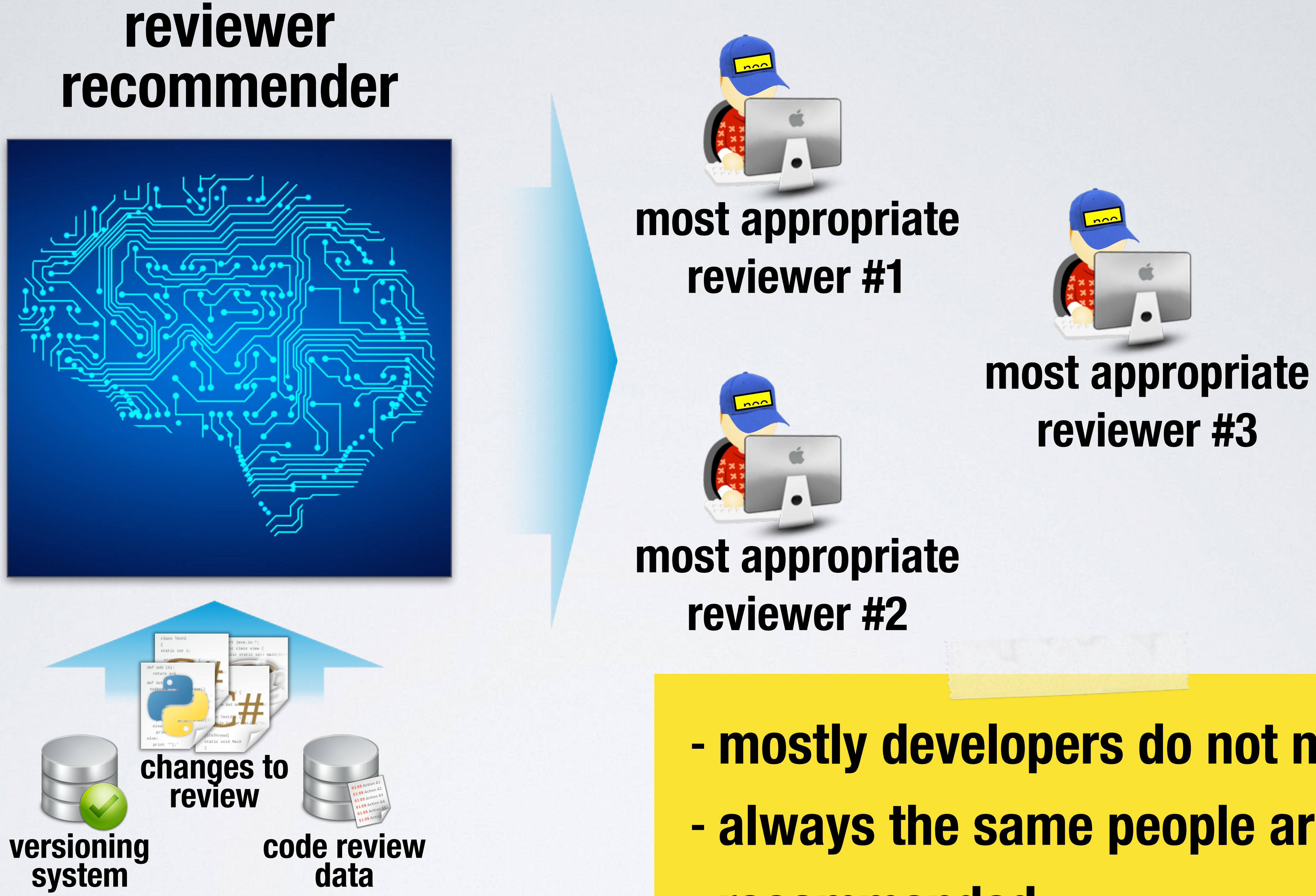
Cezary Zawadka

My mistake - for withReview() precondition makes sense

Create issue · 29 Aug 14

```
    }

    public static class TrackedBranchesSearchCriteriaBuilder {
        private String reviewPermaId;
```

Who should review the change? Reviewer recommender



reviewer recommender

changes to review

versioning system

code review data

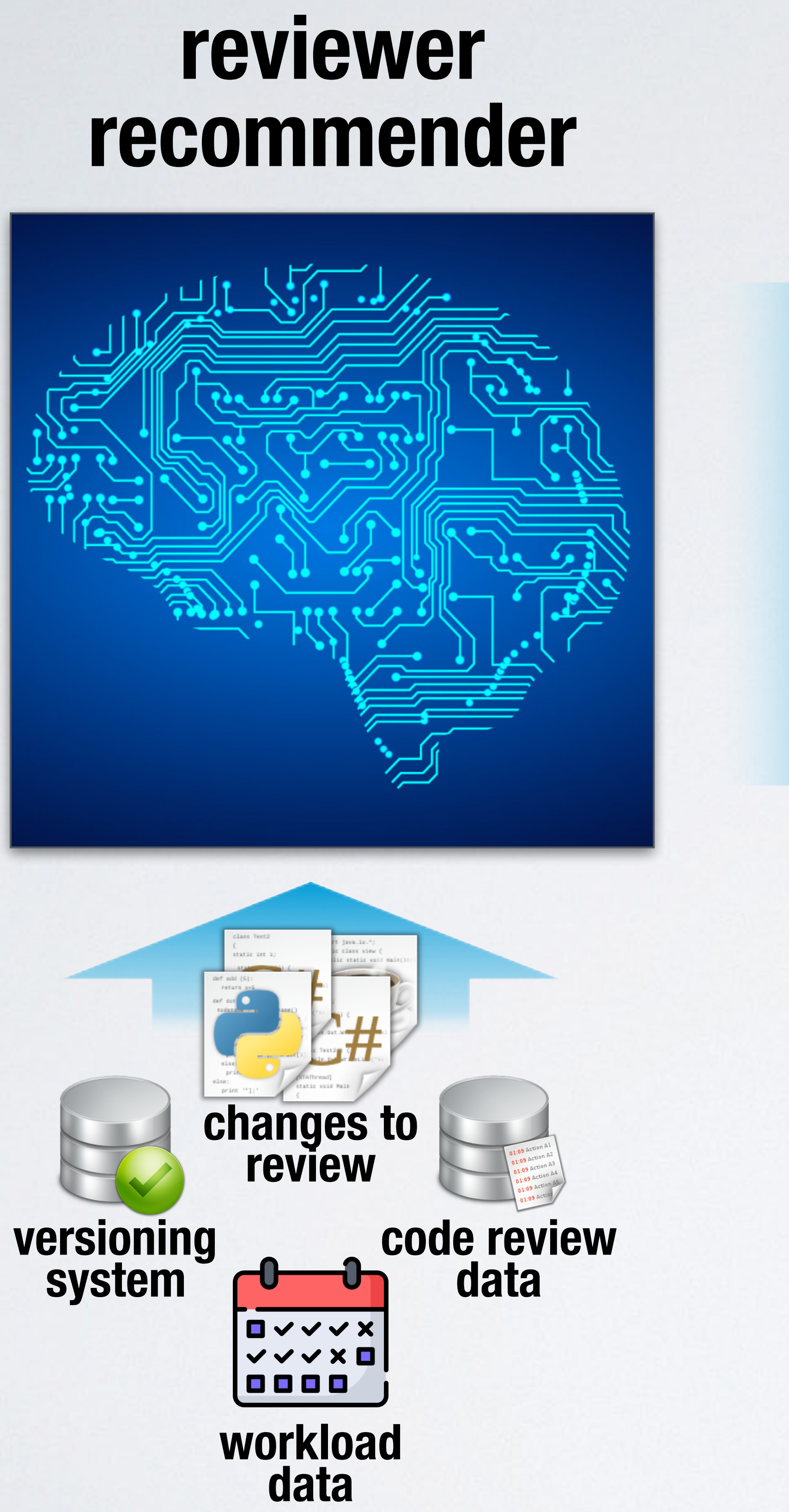
most appropriate reviewer #1

most appropriate reviewer #2

most appropriate reviewer #3

- mostly developers do not need it
- always the same people are recommended
- workload is not considered
- diversity is not considered

Who should review the change? Reviewer recommender



most appropriate reviewer #1
most appropriate reviewer #3
most appropriate reviewer #2

Tools and teams should be more mindful and supportive when it comes to reviewing code.

Asking Uncomfortable Questions On Code Review

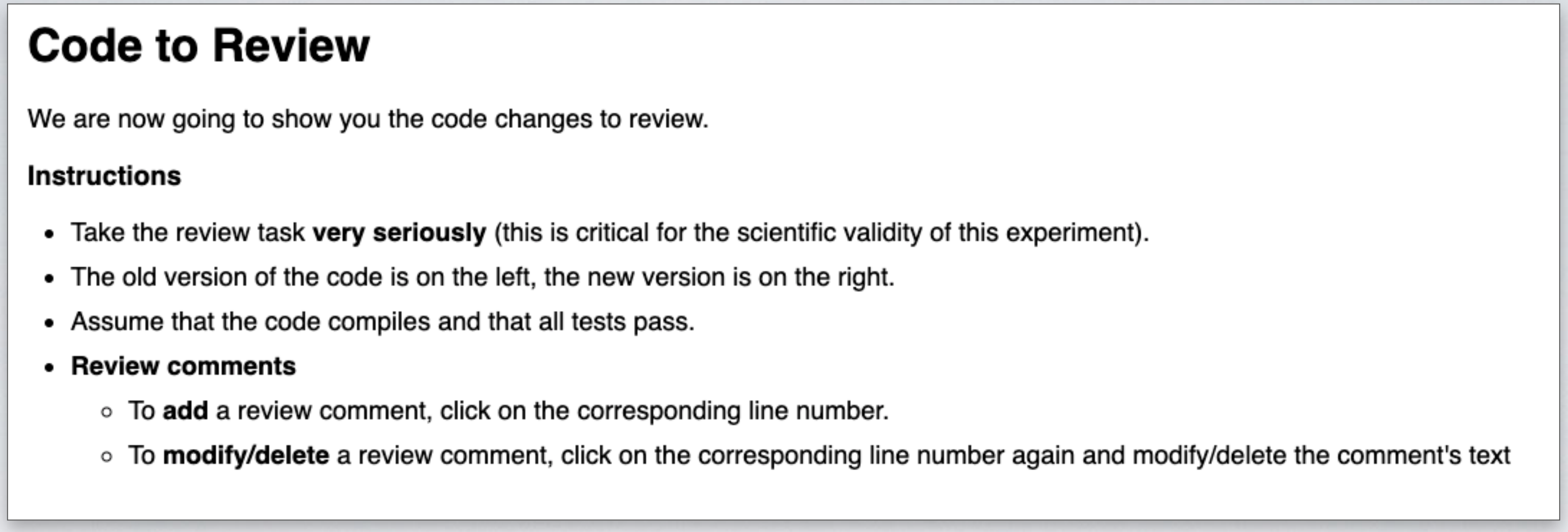
But.. why are we doing code review at all?

Our code review tools are great! Aren't they?

Are we really in this together?

Can we find security problems in code review?

Can we find security problems in code review?



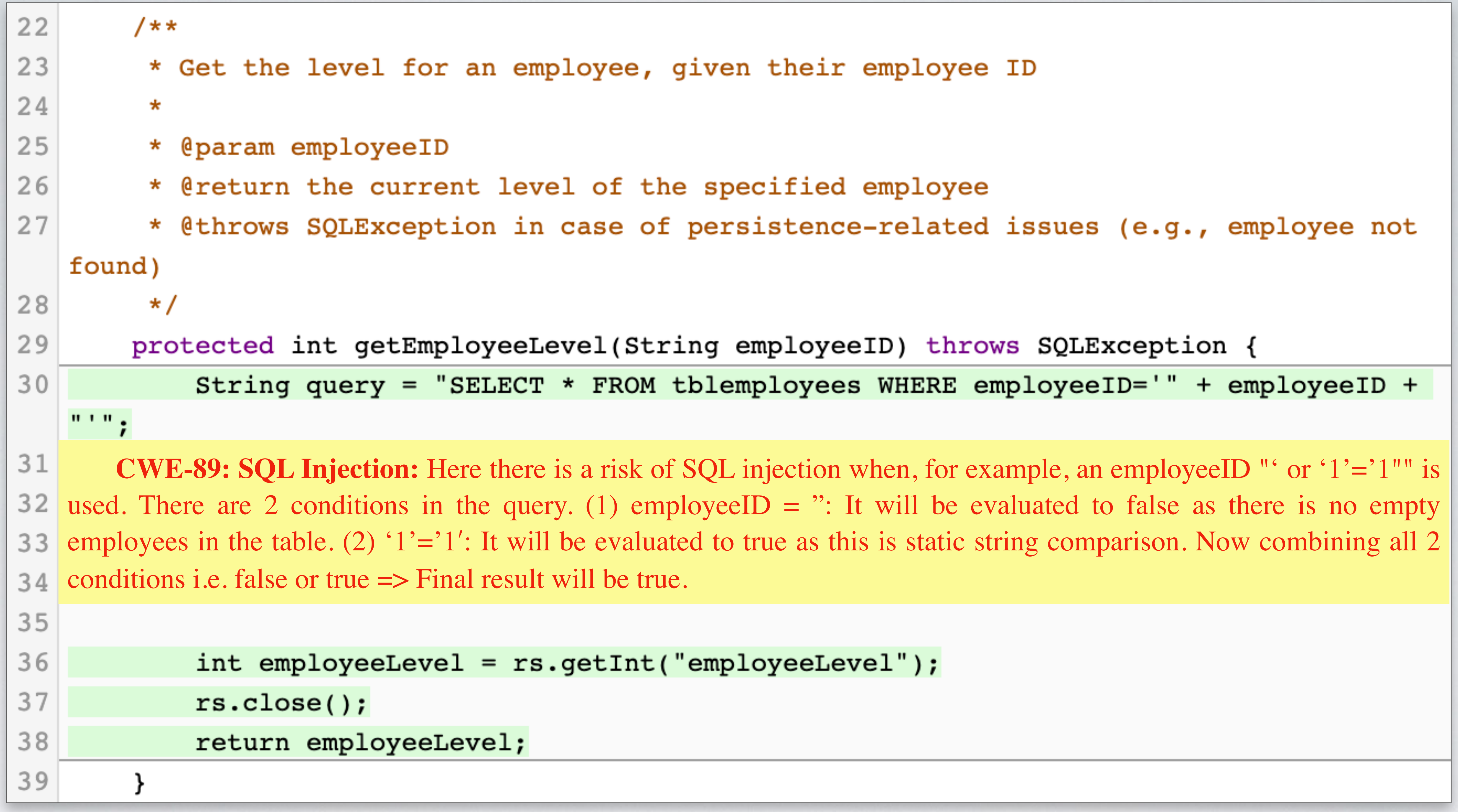
Code to Review

We are now going to show you the code changes to review.

Instructions

- Take the review task **very seriously** (this is critical for the scientific validity of this experiment).
- The old version of the code is on the left, the new version is on the right.
- Assume that the code compiles and that all tests pass.
- **Review comments**
  - To **add** a review comment, click on the corresponding line number.
  - To **modify/delete** a review comment, click on the corresponding line number again and modify/delete the comment's text

Can we find security problems in code review?

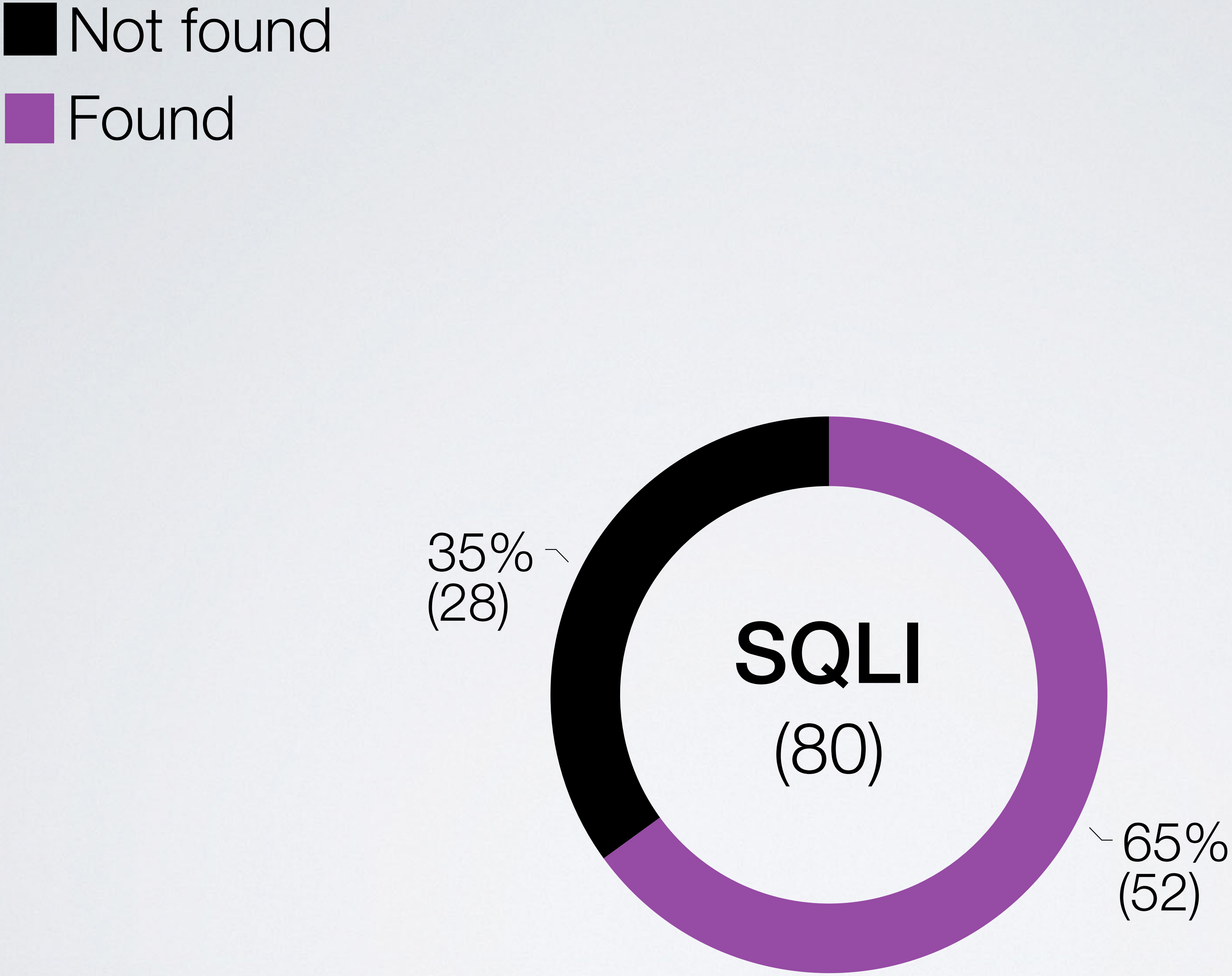


```java
    /**
     * Get the level for an employee, given their employee ID
     *
     * @param employeeID
     * @return the current level of the specified employee
     * @throws SQLException in case of persistence-related issues (e.g., employee not found)
     */
    protected int getEmployeeLevel(String employeeID) throws SQLException {
        String query = "SELECT * FROM tblemployees WHERE employeeID='" + employeeID + "'";
```

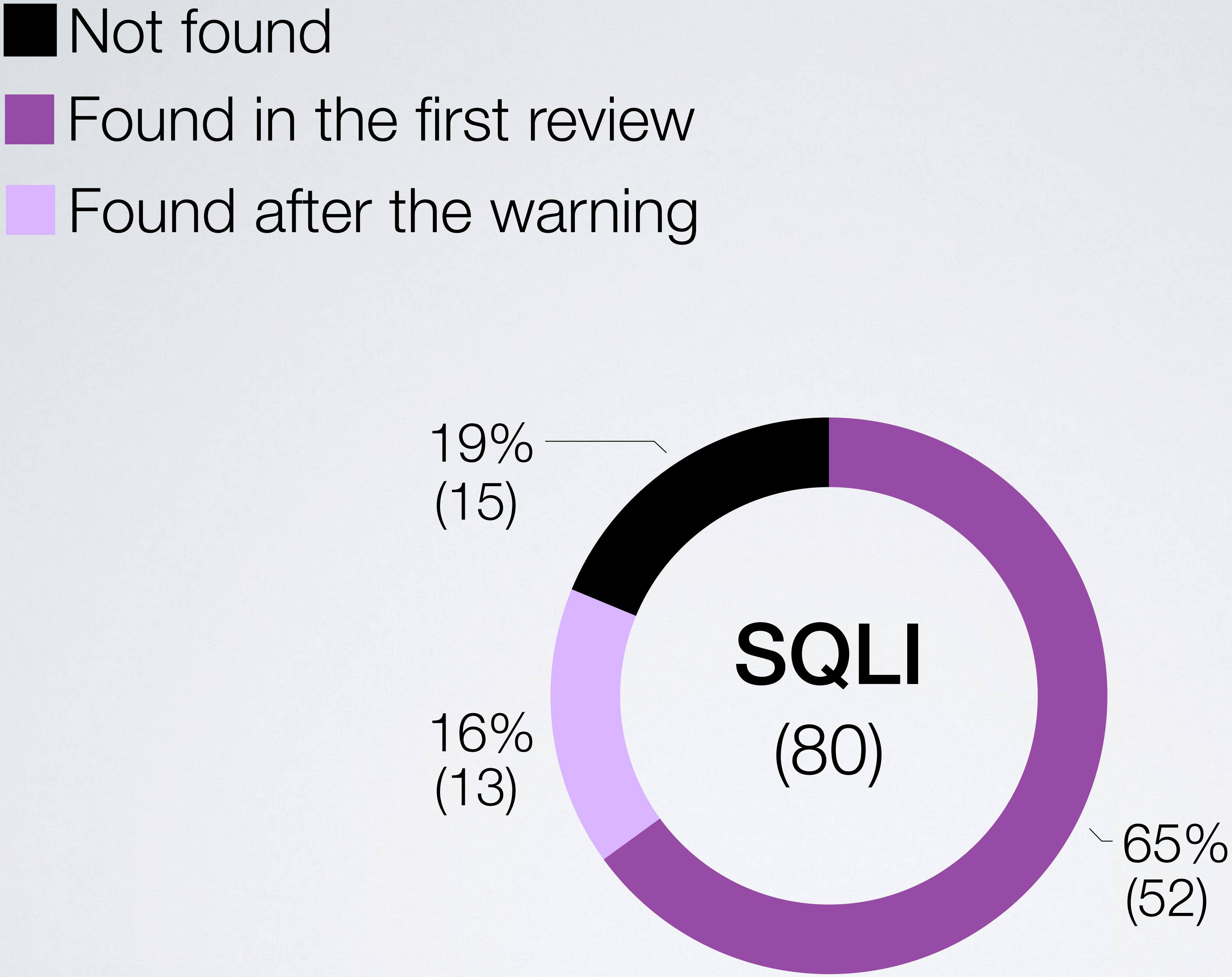
CWE-89: SQL Injection: Here there is a risk of SQL injection when, for example, an employeeID "‘ or ‘1’=’1"" is used. There are 2 conditions in the query. (1) employeeID = ”: It will be evaluated to false as there is no empty employees in the table. (2) ‘1’=’1′: It will be evaluated to true as this is static string comparison. Now combining all 2 conditions i.e. false or true => Final result will be true.

```java

        int employeeLevel = rs.getInt("employeeLevel");
        rs.close();
        return employeeLevel;
    }
```

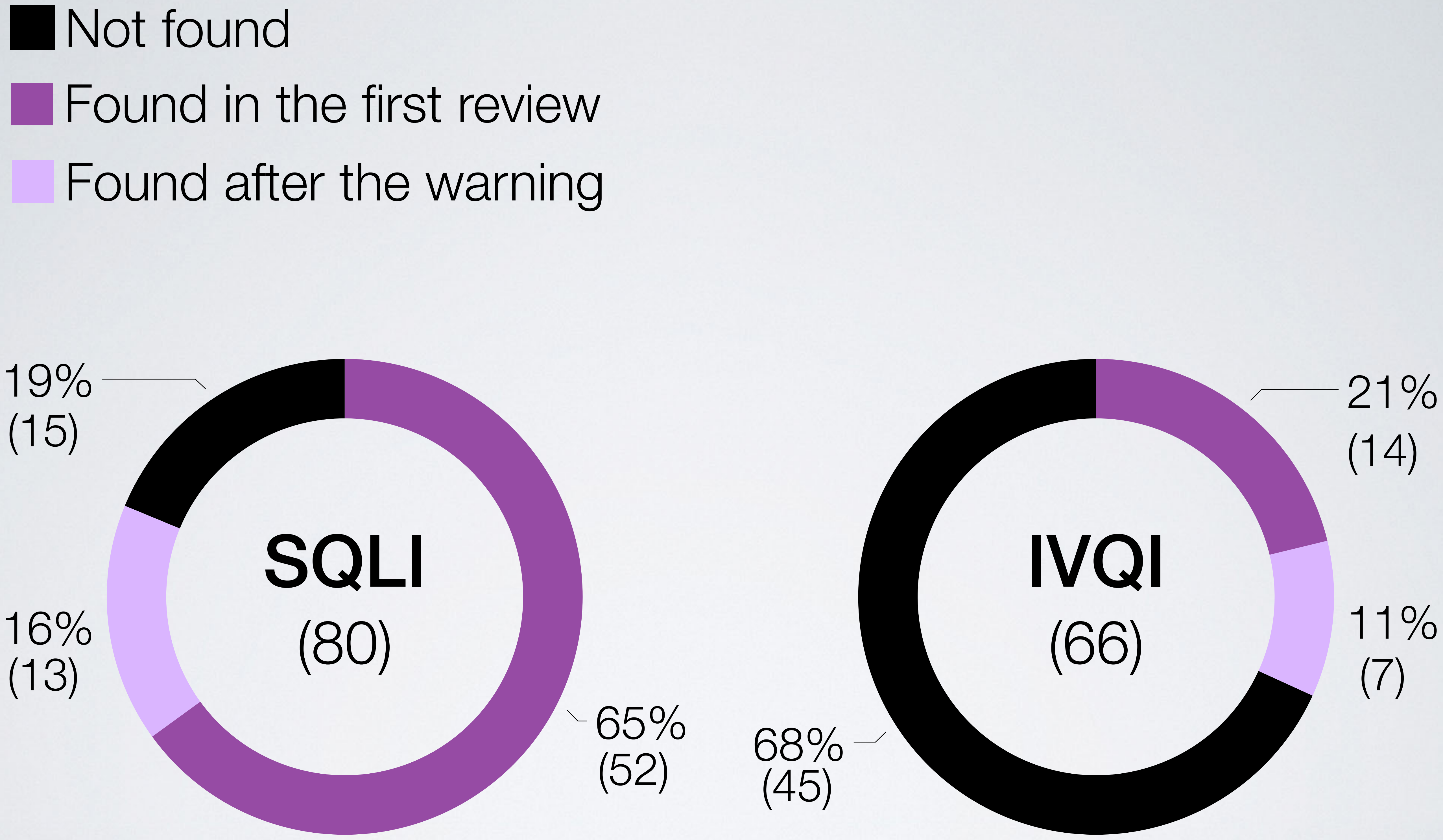
Can we find security problems in code review?



Can we find security problems in code review?



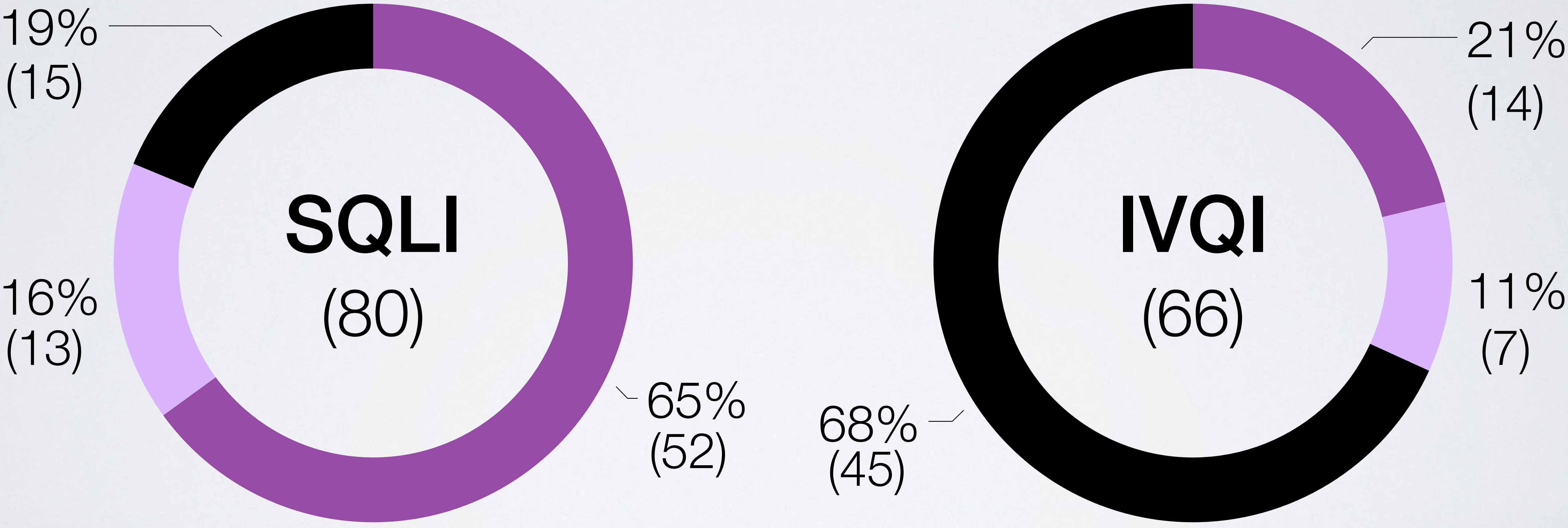
Can we find security problems in code review?



Can we find security problems in code review?

- ■ Not found
- 🟪 Found in the first review
- ▨ Found after the warning

Most factors related to <u>low knowledge</u> and <u>practice</u> contribute to missing vulnerabilities during code review, also after prompting.



Other Uncomfortable Questions On Code Review We Are Investigating

How should a change be **split for review**?

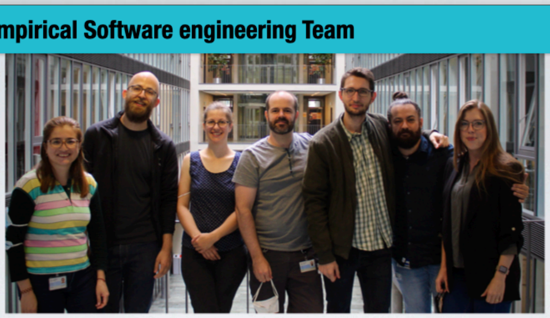
Can we **measure the effect** of code review?

What **cognitive biases** are influencing our review?

How do **top reviewers** go about reviewing code?

…

ZEST: Zurich Empirical Software engineering Team



Research

- Software quality
  - Peer code review
  - Software testing
  - Software security
- Fundamentals of Data Science for Software Engineering
  - Predictive Analytics
  - Data-driven Tools

Education

- Software Design & Construction
- Software Testing
- Software Analytics (aka Data Science for S.E.)

Empirical Software Engineering?

Empirical software engineering involves the scientific use of quantitative and qualitative data to understand and improve software product, software development process and software management.

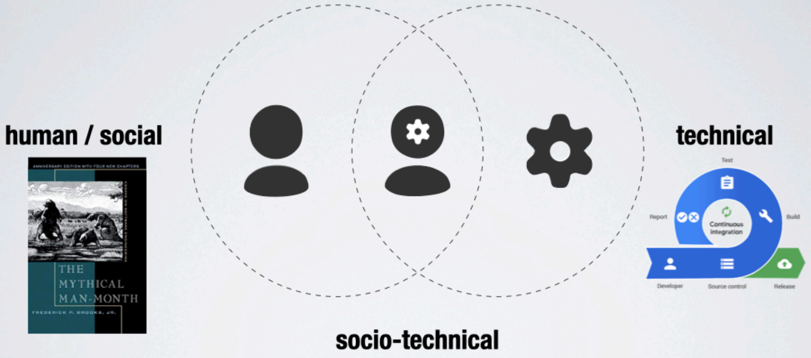
Empirical software engineering **starts with a good question**:

- Does pair programming work?
- Is static typing really good?
- What are the advantages of properly following continuous integration?
- How does using GitHub influence open-source projects?

Empirical software engineering **leads to actionable results**:

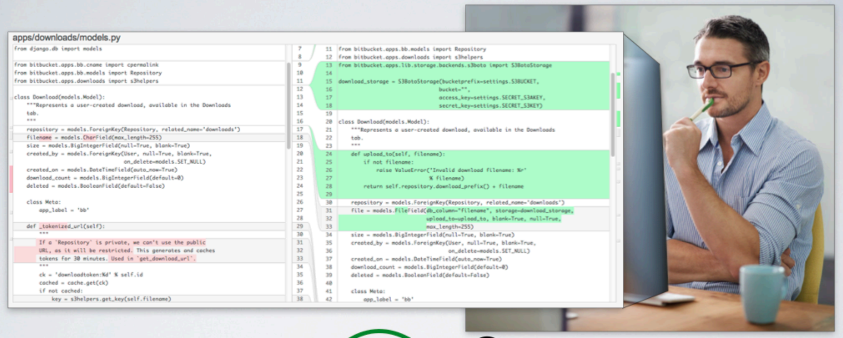
- The creation of new tools
- The improvement of existing tools
- The improvement of existing development and engineering processes
- More questions :)

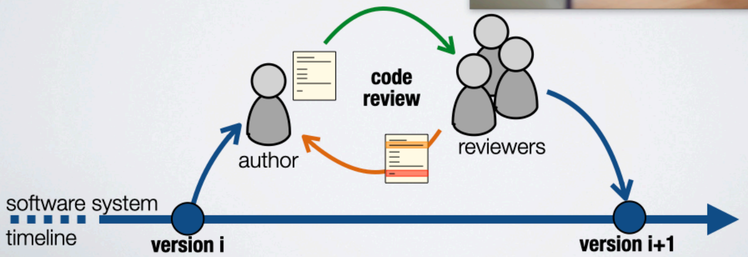
Software Engineering as a Socio-Technical Space



Productivity Paradox — Margaret-Anne Storey ICSE 2019

Joint Space — Code Review





Asking Uncomfortable Questions On Code Review

But.. why are we doing code review at all?

Our code review tools are great! Aren't they?

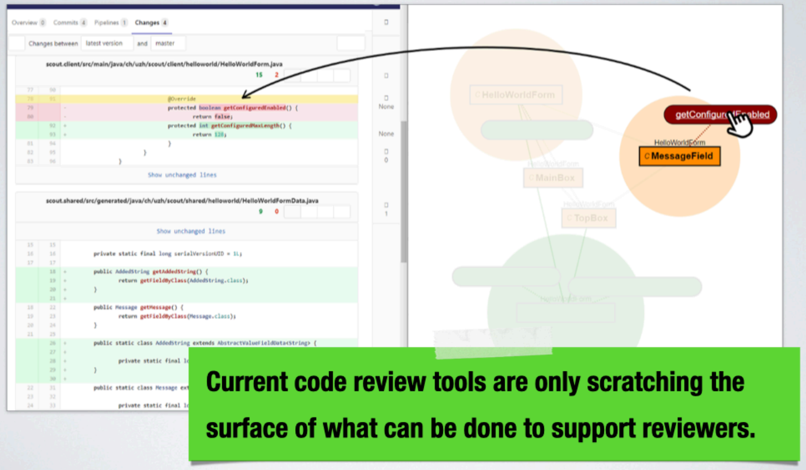
Are we really in this together?

Can we find security problems in code review?

Code review at Microsoft in 2012: Expectations vs. Reality

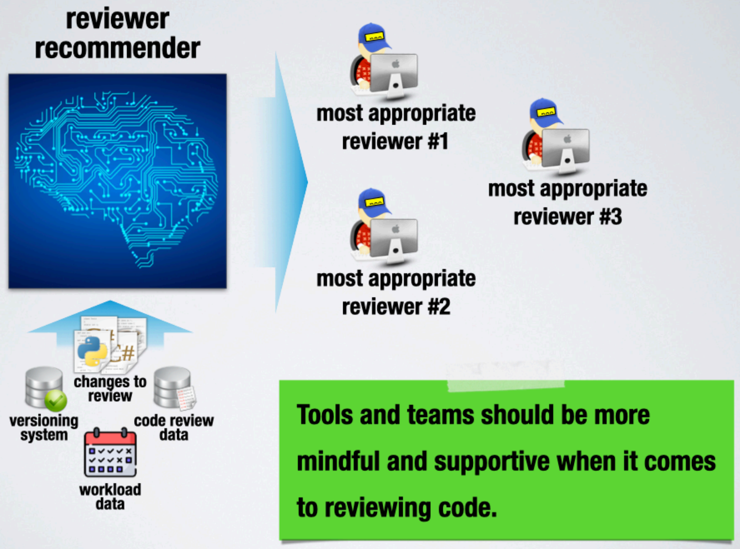
If you want to use any Software Engineering process PROPERLY, you need to evaluate it and reflect on it. Even if you are Microsoft.

Next generation code review tools: Code Change Visualization



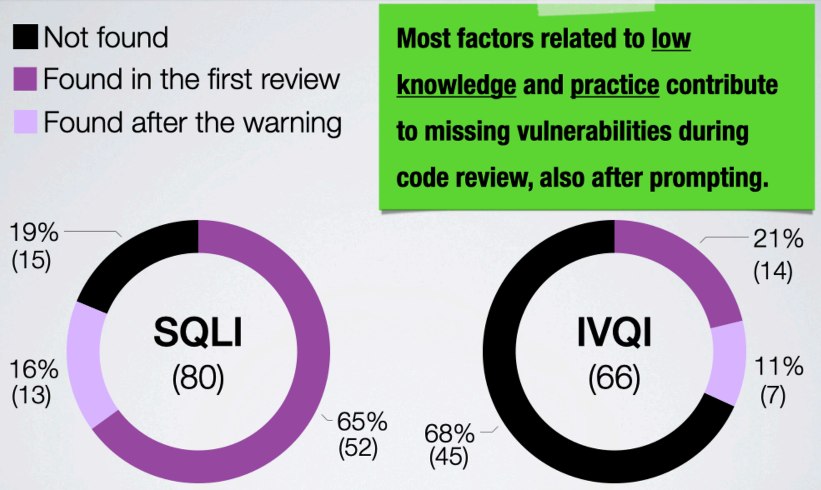
Current code review tools are only scratching the surface of what can be done to support reviewers.

Who should review the change? Reviewer recommender


reviewer recommender
most appropriate reviewer #1
most appropriate reviewer #2
most appropriate reviewer #3
changes to review
versioning system
code review data
workload data

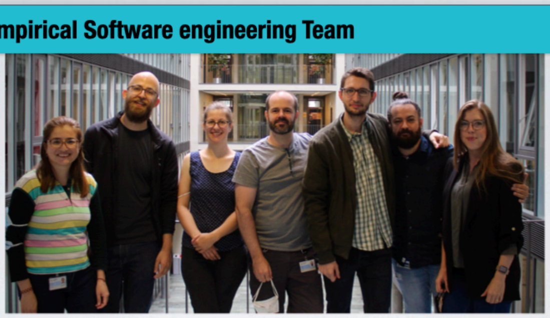
Tools and teams should be more mindful and supportive when it comes to reviewing code.

Can we find security problems in code review?



Most factors related to low knowledge and practice contribute to missing vulnerabilities during code review, also after prompting.

ZEST: Zurich Empirical Software engineering Team



Research

- Software quality
  - Peer code review
  - Software testing
  - Software security
- Fundamentals of Data Science for Software Engineering
  - Predictive Analytics
  - Data-driven Tools

Education

- Software Design & Construction
- Software Testing
- Software Analytics (aka Data Science for S.E.)

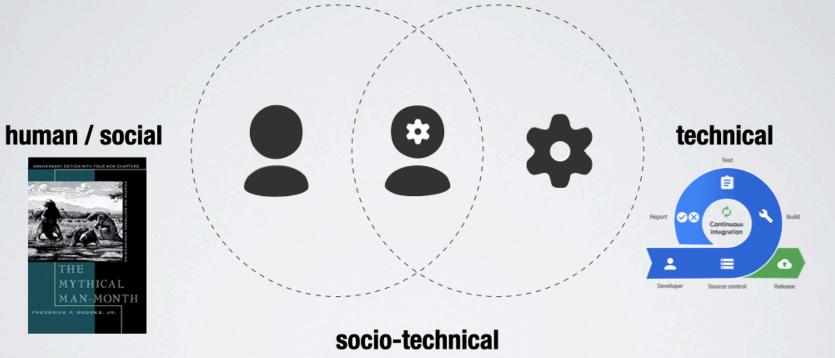
Thank you!

Alberto

http://zest.ifi.uzh.ch

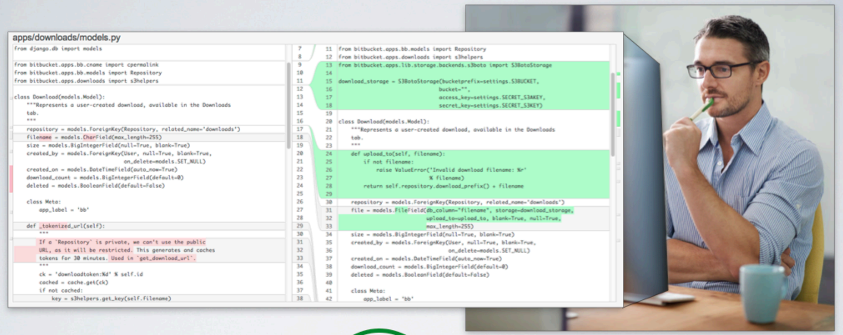
zest@ifi.uzh.ch

Software Engineering as a Socio-Technical Space



Productivity Paradox — Margaret-Anne Storey ICSE 2019

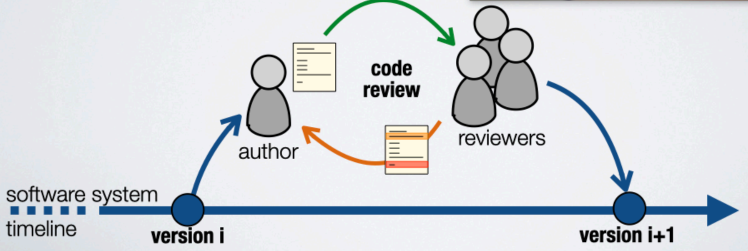
Joint Space — Code Review



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'
```



Asking Uncomfortable Questions On Code Review

But.. why are we doing code review at all?

Our code review tools are great! Aren't they?

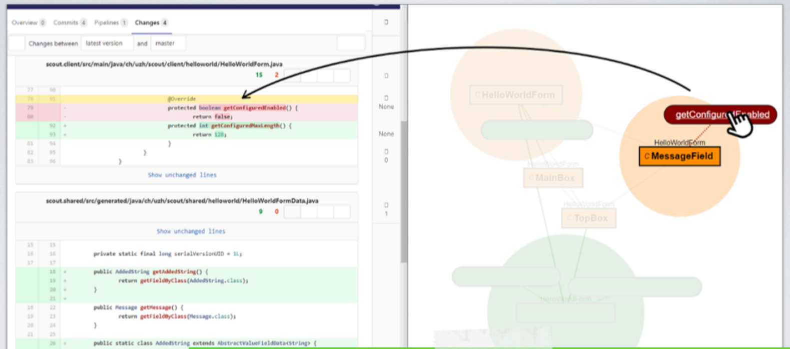
Are we really in this together?

Can we find security problems in code review?

Code review at Microsoft in 2012: Expectations vs. Reality

If you want to use any Software Engineering process PROPERLY, you need to evaluate it and reflect on it. Even if you are Microsoft.

Next generation code review tools: Code Change Visualization



Overview 0 Commits 4 Pipelines 1 Changes 4

Changes between latest version and master

scout.client/src/main/java/ch/uzh/scout/client/helloworld/HelloWorldForm.java 15 2

```
            @Override
-           protected boolean getConfiguredEnabled() {
-               return false;
+           protected int getConfiguredMaxLength() {
+               return 128;
            }
        }
    }
```

Show unchanged lines

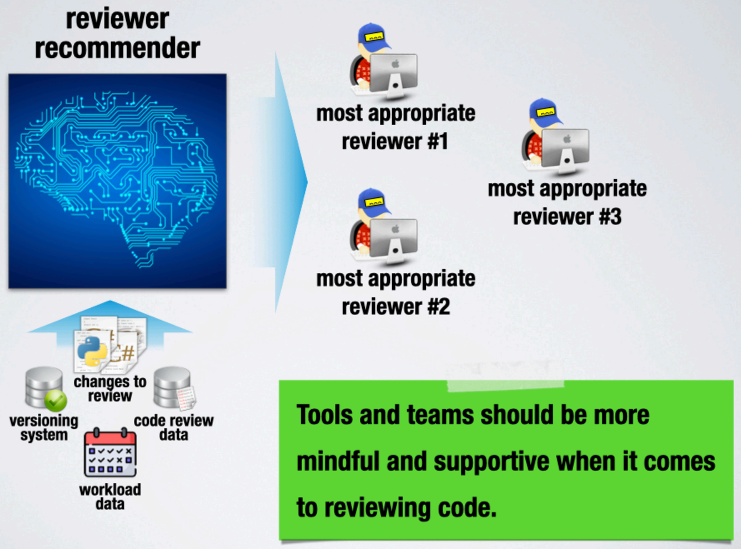
scout.shared/src/generated/java/ch/uzh/scout/shared/helloworld/HelloWorldFormData.java 9 0

Show unchanged lines

```
        private static final long serialVersionUID = 1L;

+       public AddedString getAddedString() {
+           return getFieldByClass(AddedString.class);
+       }
+
        public Message getMessage() {
            return getFieldByClass(Message.class);
        }

+       public static class AddedString extends AbstractValueFieldData<String> {
+
+           private static final l
+       }
+
        public static class Message ext

            private static final l
```

Current code review tools are only scratching the surface of what can be done to support reviewers.

Who should review the change? Reviewer recommender

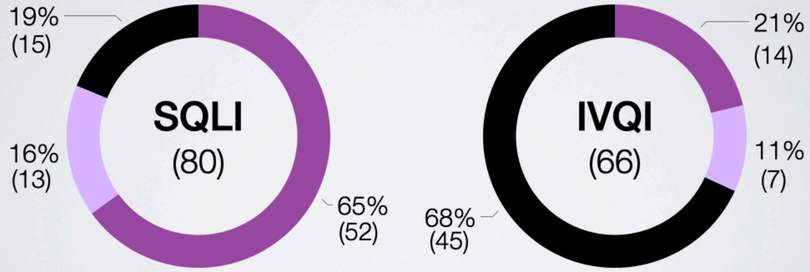

reviewer recommender
most appropriate reviewer #1
most appropriate reviewer #2
most appropriate reviewer #3
changes to review
versioning system
code review data
workload data

Tools and teams should be more mindful and supportive when it comes to reviewing code.

Can we find security problems in code review?

- ■ Not found
- ■ Found in the first review
- ■ Found after the warning

Most factors related to low knowledge and practice contribute to missing vulnerabilities during code review, also after prompting.


SQLI (80)
19% (15)
16% (13)
65% (52)
IVQI (66)
21% (14)
11% (7)
68% (45)