CS 4350: Fundamentals of Software Engineering
CS 5500: Foundations of Software Engineering

Lesson 7.3 Code Reviews

Jonathan Bell, Adeel Bhutta, Ferdinand Vesely, Mitch Wand

Khoury College of Computer Sciences

Learning Objectives for this Lesson

- By the end of this lesson, you should be able to:
  - Articulate what a *code review* is;
  - List the roles of people in code reviews;
  - Explain an appropriate time for code reviews;
  - Illustrate one way to hold a code review;
  - Describe the benefits of a culture of code review.

Linus's Law motivates Code Review

Given enough eyeballs, all bugs are shallow

- Coined by Eric Raymond in honor of Linus Torvalds.
- "Mantra" of Open-Source Movement
- (*pace* "Heartbleed", and others.)

Code Should be Read, not Just Run

- Requirements and design documents are usually read by multiple people - who often didn’t write them
- There are meetings, discussions
- Customers, managers, developers, Q&A provide input
- Should code get released having been read just by its author?
- Think of books: nothing is commercially published without scrutiny and input from editors

Code *Inspection* is Heavier Weight

- Formal process of reading through code as a group;
- Applied to all project documents;
- A 3-5 person team reads the code aloud and explains what is being done;
- Each person has a specific role (moderator, reviewer, reader, scribe, observer, author)
- Note, the author does not present their code
- Usually a 60 minute meeting;
- Less efficient (defects/cost) than modern review processes.
- Very waterfall.
- Traceable, measurable

Code Review: What

- A code review is the process in which the author of some code is asked to explain it to their peers:
  - What purpose the code has;
  - How the code accomplishes this purpose;
  - How the author is confident of this information,
    - E.g., show results of running tests.
- A code review often concerns a code *change.*
- A code review doesn’t assume anything is wrong.
- A code review isn’t “selling” the code.
- See Chapter 9 in SoftEng @ Google

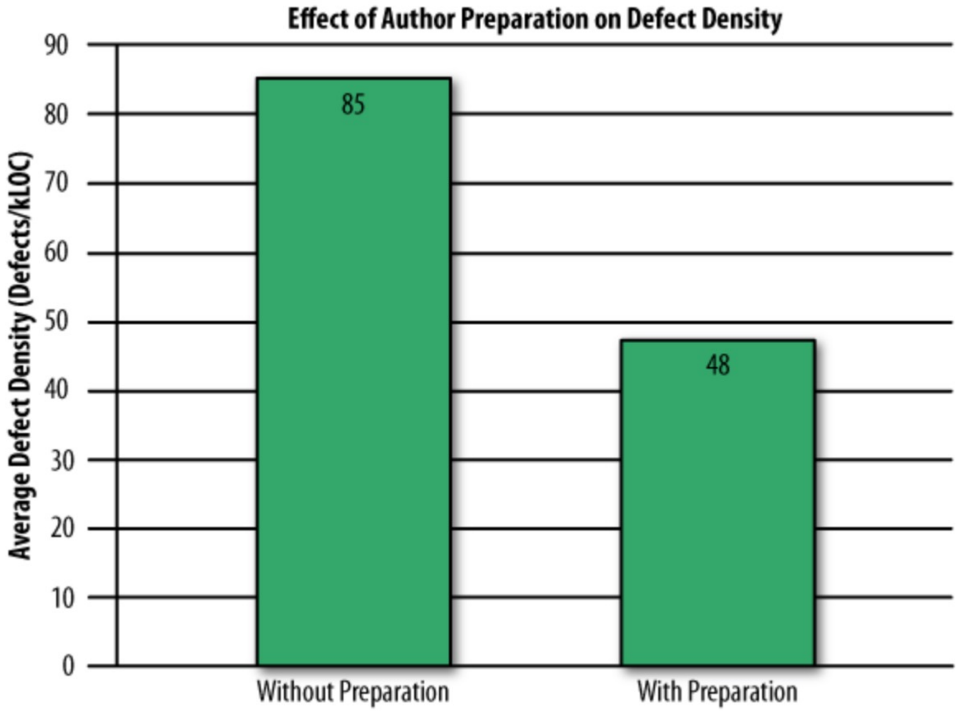
Code Review: Who

- The author of the code is the presenter.
- An owner of the code being changed or added to
  - May sometimes be the same person as presenter.
- Someone to verify that the code meets standards.
- Someone to ensure documentation is consistent.
- Other people:
  - Interested in this code base;
  - Experts in the code base.

SE@Google:
At least one person
other than author

Self-Review Less Effective than Peer Review

Study of 300 reviews at Cisco in 2006



Even if developers pre-review their code, many defects still found in peer review

“Best Kept Secrets of Peer Code Review”, Jason Cohen, SmartBear Software, 2006

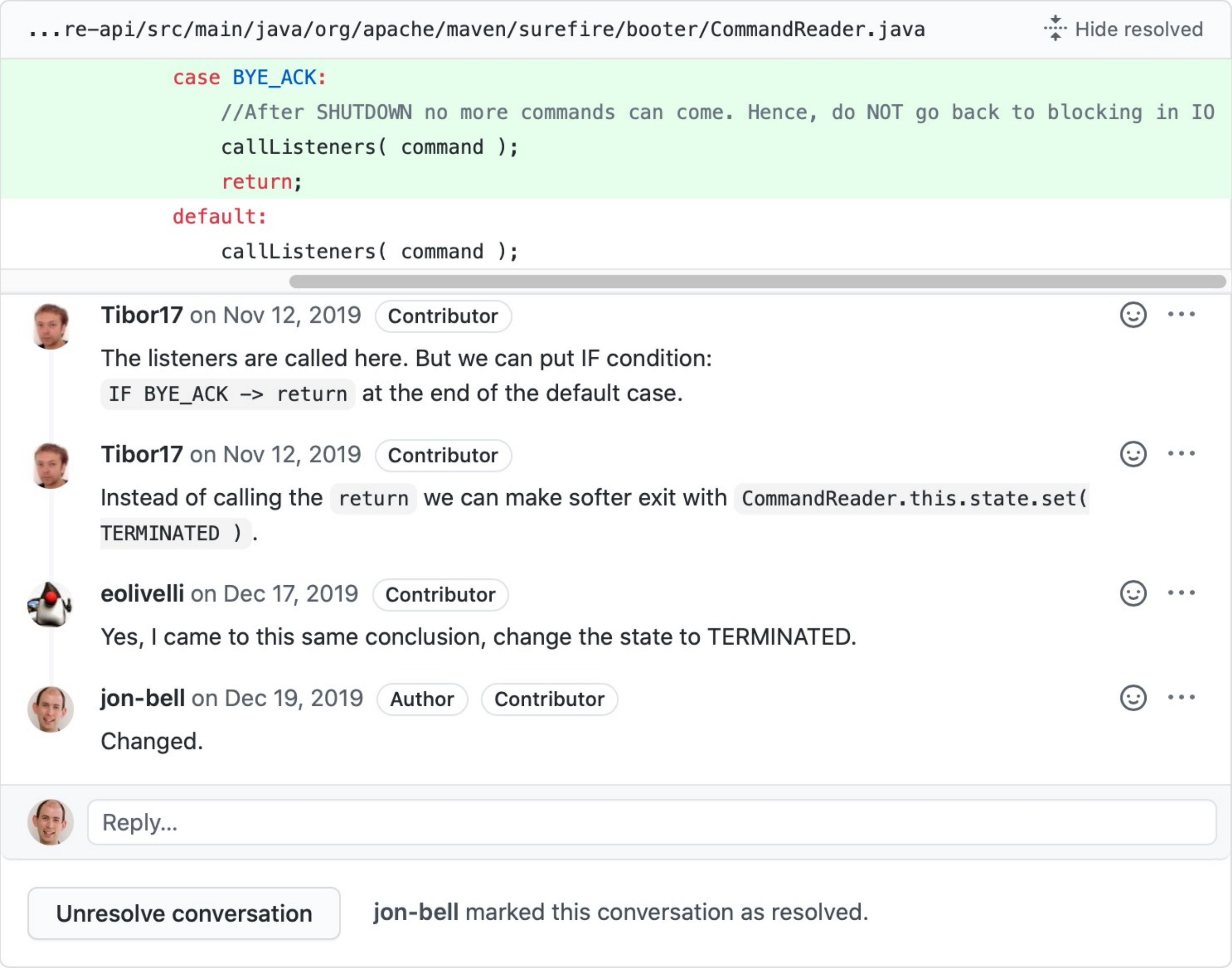
Code Review: When

- SE @ Google recommends reviews at commit:
  - **Every** commit must be reviewed;
  - Best time to ensure code is good:
    - Once code is in production, hard to justify;
    - Before code is ready to use, review superfluous.
  - Reviews need to be done quickly.
- Code review for new developers:
  - Helps them understand standards;
- Code review of established code:
  - Spread understanding of algorithms/techniques.

Code Review: How

- At Google, reviewers get access to changes, explanation and all relevant test results: review is asynchronous.
- Elsewhere reviews can be in person:
  - More heavyweight, cannot be as common.
- Review must be professional and impersonal:
  - No one is being “attacked” (or, no one *should* be).
- Don’t rehash design arguments (defer to author).
- All suggestions and criticisms must be addressed:
  - At least in the negative.

Code Review: Example on Pull Request



...re-api/src/main/java/org/apache/maven/surefire/booter/CommandReader.java

```
        case BYE_ACK:
            //After SHUTDOWN no more commands can come. Hence, do NOT go back to blocking in IO
            callListeners( command );
            return;
        default:
            callListeners( command );
```

Tibor17 on Nov 12, 2019

The listeners are called here. But we can put IF condition: `IF BYE_ACK -> return` at the end of the default case.

Tibor17 on Nov 12, 2019

Instead of calling the `return` we can make softer exit with `CommandReader.this.state.set( TERMINATED )`.

eolivelli on Dec 17, 2019

Yes, I came to this same conclusion, change the state to TERMINATED.

jon-bell on Dec 19, 2019

Changed.

jon-bell marked this conversation as resolved.

Code Review: Sample Check-List

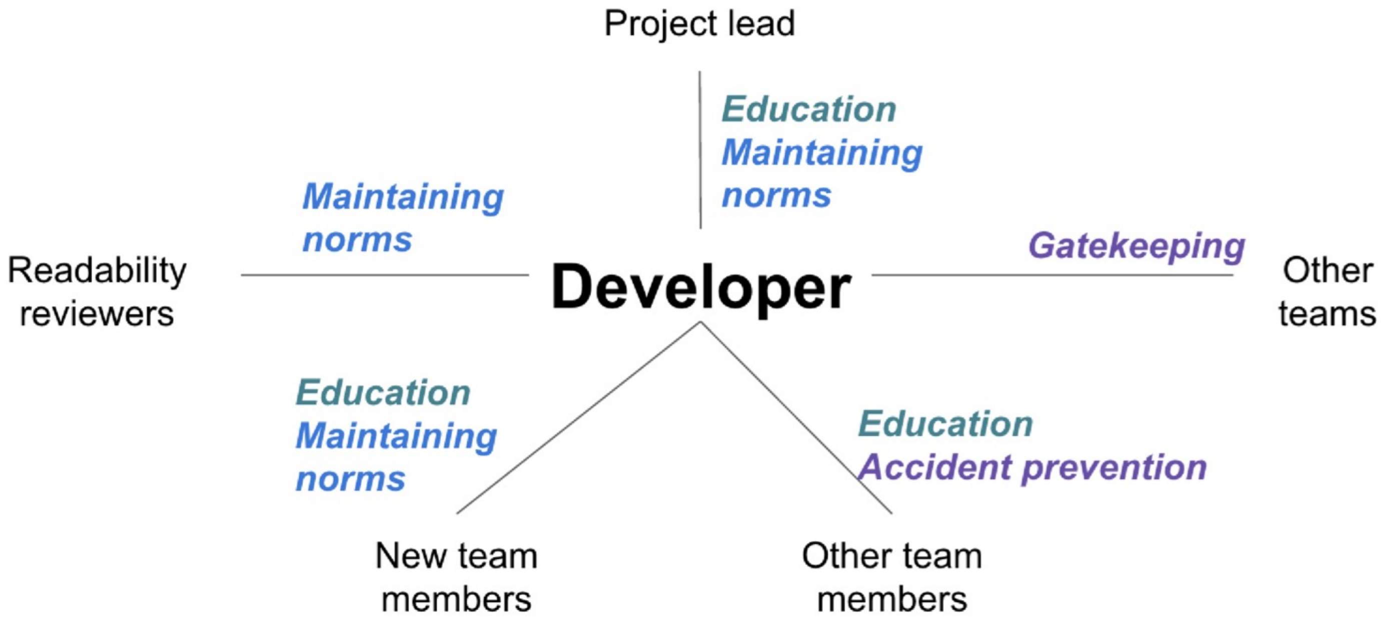
- Am I able to understand the code easily?
- Does the code follow our style guidelines?
- Is the same code duplicated more than once?
- Is this file (or change) too big?
- Does this code meet our non-functional requirements?
- Is this code maintainable?
- Does this code have unintended side-effects?

Code Review: Why

- Code review increases breadth of knowledge of code:
  - Other people ”know” the code;
  - Easier to handle someone cycling off project.
- Verbalizing decisions improves their quality:
  - The process of writing an explanation encourages critical thinking.
- Code reviews improve quality of code base:
  - Knowing code will be reviewed pushes developers to make code more presentable and understandable.

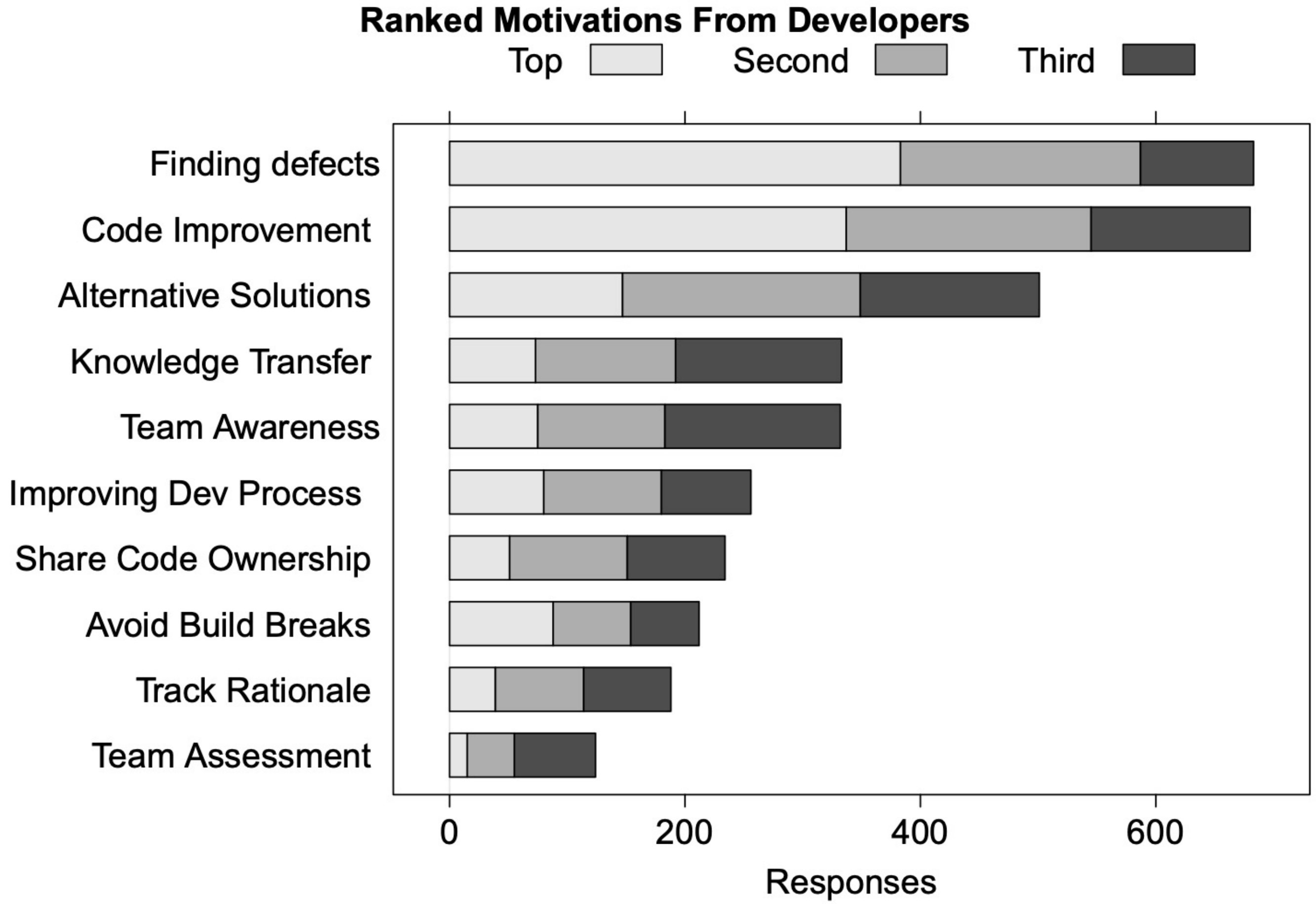
Code Review: Why (Google)

Different team members have different motivations and bring different benefits



“Modern Code Review: A Case Study at Google”, Sadowski et al, ICSE 2018

Code Review: Why (Microsoft)



“Expectations, Outcomes, and Challenges of Modern Code Review”, Bacchelli & Bird, ICSE 2013

A Note on Programmer’s Ego

- Someone looking over your work
- Probably some attachment to it
- Criticisms: sometimes hard not to take personally
- Acknowledge a criticism and move on
- Acknowledgment doesn’t imply that the author agrees with the content of the criticism
- Author should not try to defend the work under review

Review: Learning Objectives for this Lesson

- You should now be able to:
  - Articulate what a *code review* is;
  - List the roles of people in code reviews;
  - Explain an appropriate time for code reviews;
  - Illustrate one way to hold a code review;
  - Describe the benefits of a culture of code review.

Looking Forward...

- In our next lesson, we'll discuss other approaches for ensuring software quality: Analysis and verification.