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

Lesson 7.3 Code Reviews

Jon Bell, John Boyland, Mitch Wand Khoury College of Computer Sciences

© 2021 Jonathan Bell, John Boyland and Mitch Wand. Released under the <u>CC BY-SA</u> license

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.)

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.

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;
- Usually a 60 minute meeting;
- Less efficient (defects/cost) than modern review processes.

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 <u>Chapter 9</u> 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.

SEDGoogle: At least one person other than author

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: 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.



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

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.