Code Review [1]

Challenges of Large Code Base

- How to ensure…
  - Maintainable code?
  - DRY code?
  - Readable code?
  - Bug-free code?
- Average defect detection rate for various testing
  - Unit testing: 25%
  - Function testing: 35%
  - Integration testing:45%
- How can this be improved?

Code Review

- A constructive review of a fellow developer’s code. A required sign-off from another team member before a developer is permitted to check in changes or new code.
- Analogy: writing articles for a newspaper

  What is the effectiveness of…
  - Spell-check/grammar check
  - Author editing own article
  - People editing others’ articles

Mechanics of code reviews

- ***Who***:
  - Original developer and reviewer, sometimes together in person, sometimes offline.
- ***What***:
  - Reviewer gives suggestions for improvement on a logical and/or structural level, to conform to previously agreed upon set of quality standards.
    - Feedback leads to refactoring, followed by a 2nd code review.
    - Eventually reviewer approves code.

Mechanics of code reviews (cont’d)

- ***When***:
  - When code author has finished a coherent system change that is otherwise ready for checkin
    - change shouldn't be too large or too small
    - before committing the code to the repository or incorporating it into the new build

Why Bother ?

- > 1 person has seen every piece of code
  - Prospect of someone reviewing your code raises quality threshold.
- Forces code authors to articulate their decisions
- Hands-on learning experience for rookies without hurting code quality
  - Pairing them up with experienced developers

Why Bother ? (cont'd)

- Team members involved in different parts of the system
  - Reduces redundancy, enhances overall understanding
- Author and reviewer both accountable for committing code

Actual Studies [2]

- Average defect detection rates
  - Unit testing: 25%
  - Function testing: 35%
  - Integration testing:45%
  - Design and code inspections: 55% and 60%.

Actual Studies (cont'd)

- 11 programs developed by the same group of people
  - First 5 without reviews: average 4.5 errors per 100 lines of code
  - Remaining 6 with reviews: average 0.82 errors per 100 lines of code
  - Errors reduced by > 80 percent.

Actual Studies (cont'd)

- IBM's Orbit project: 500,000 lines, 11 levels of inspections. Delivered early and 1% of the errors that would normally be expected.
- After AT&T introduced reviews , study with > 200 people reported a 14% increase in productivity and a 90% decrease in defects.

Code reviews in industry

- Code reviews are a very common industry practice.
- Made easier by advanced tools that:
  - integrate with configuration management systems
  - highlight changes (i.e., diff function)
  - allow traversing back into history
    - E.g.: Eclipse, SVN tools

Code review variations

- **inspection**: A more formalized code review with:
  - roles (moderator, author, reviewer, scribe, etc.)
  - several reviewers looking at the same piece of code
  - a specific checklist of kinds of flaws to look for
    - possibly focusing on flaws that have been seen previously
    - possibly focusing on high-risk areas such as security
  - specific expected outcomes (e.g. report, list of defects)

Code review variations (cont'd)

- **walkthrough**: informal discussion of code between author and a single reviewer
- **code reading**: Reviewers look at code by themselves (possibly with no actual meeting)

Code Reviews at Google

- "All code that gets submitted needs to be reviewed by at least one other person, and either the code writer or the reviewer needs to have readability in that language. Most people use Mondrian to do code reviews, and obviously, we spend a good chunk of our time reviewing code."

-- Amanda Camp, Software Engineer, Google

Code reviews at Yelp

- "At Yelp we use review-board. An engineer works on a branch and commits the code to their own branch. The reviewer then goes through the diff, adds inline comments on review board and sends them back.

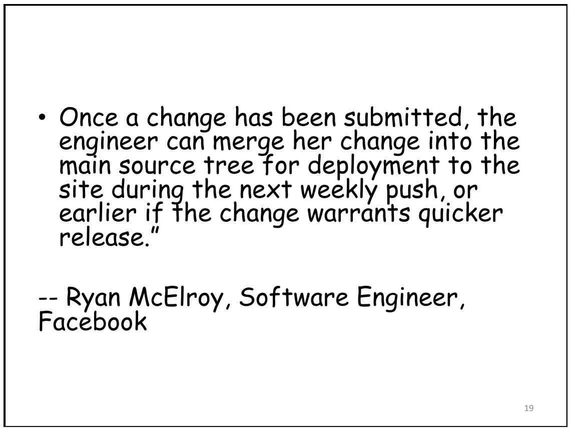
- The reviews are meant to be a dialogue, so typically comment threads result from the feedback. Once the reviewer's questions and concerns are all addressed they'll click "Ship It!" and the author will merge it with the main branch for deployment the same day."

-- Alan Fineberg, Software Engineer, Yelp

Code reviews at Facebook

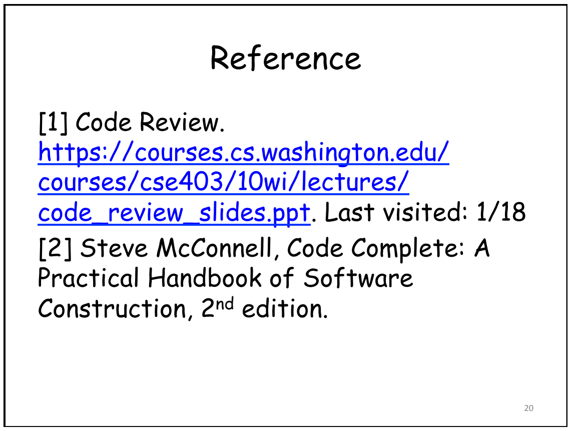
- "At Facebook, we have an internally-developed web-based tool to aid the code review process. Once an engineer has prepared a change, she submits it to this tool, which will notify the person or people she has asked to review the change, along with others that may be interested in the change -- such as people who have worked on a function that got changed.

- At this point, the reviewers can make comments, ask questions, request changes, or accept the changes. If changes are requested, the submitter must submit a new version of the change to be reviewed. All versions submitted are retained, so reviewers can compare the change to the original, or just changes from the last version they reviewed.



- Once a change has been submitted, the engineer can merge her change into the main source tree for deployment to the site during the next weekly push, or earlier if the change warrants quicker release."

-- Ryan McElroy, Software Engineer, Facebook



Reference

[1] Code Review. https://courses.cs.washington.edu/courses/cse403/10wi/lectures/code_review_slides.ppt. Last visited: 1/18

[2] Steve McConnell, Code Complete: A Practical Handbook of Software Construction, 2nd edition.