THE ART OF REVIEWING CODE

Arjan van Leeuwen
Yep, I see your problem,
there's a dude passed out in your engine.

Also, your mechanic is a pony.
WHY REVIEW CODE?

• Code review could find and fix defects much faster than testing

• Some defects that are hard or impossible to find in testing can be found in code review

• Could catch more than 50% of defects
WHAT IS A CODE REVIEW?

• A Quality Control activity aimed towards detecting defects in code before the software is released by systematically examining changes to the source code
WHAT IS A DEFECT?

• A deviation from quality

• Seen from the viewpoint of the code reviewer

• If the code review team finds an issue it is a defect
WHEN TO DO CODE REVIEW?

• After automated checks have been done:
  • Code compiles
  • Existing automated tests pass
  • New automated tests have been created and pass (part of review)
WHO REVIEWS THE CODE?

• Peers, developers with knowledge of the code base
• Code owners
• Experienced developers
WHERE IS THE CODE REVIEW?

- Public
- Trackable
- Tools can help
TYPE OF DEFECTS

• Functional defects
• Non-functional or maintainability defects
• False positives
Distribution of Defects

based on research by Mika V. Mäntylä and Casper Lassenius, Helsinki University of Technology

- Maintainability: 71%
- Functional: 7%
- False positives: 21%
Maintainability Distribution

- Documentation: 35%
- Structure: 55%
- Visual representation: 10%
Structural Defects Distribution

- 52% Organization
- 48% Solution approach
USE OF CHECKLISTS
BASED ON RESEARCH BY GUOPING RONG, JINGYI LI, MINGJUAN XIE AND TAO ZHENG

• Can make it easier for beginning developers to review code
• Helps focus on the code, because there is something to look for
• People working with checklists do not find more defects
• Does not improve quality of defects found
• Does not improve efficiency
USE OF OTHER GUIDELINES

• Describe desired behavior of reviewers and developers whose code is under review

• Describe processes related to the review
  • What happens when a review is approved or rejected?
  • What happens when issues cannot be resolved?
SUBJECTIVITY IN NON-FUNCTIONAL DEFECTS

• Non-functional defects are often subjective in nature
• Nonetheless, research shows inter-rater agreement of 82%
• When in doubt add reviewers
• Long discussions between reviewers and authors should probably be moved offline, but summarize conclusions
DISAGREEMENTS: HAVING CODE REVIEWED

- Don’t take it personal. The review is of the code, not you
- Explain why code exists
- Seek to understand the reviewer’s perspective
- When disagreeing with a suggestion for improvement, make alternative suggestions
- Ask for a second opinion (in agreement with reviewer)
DISAGREEMENTS: REVIEWING CODE

• Seek to understand the author’s perspective

• Understand why the code is necessary

• Communicate whether you feel strongly about something or not

• Offer alternative implementations

• Ask for a second opinion (in agreement with author)
WHY DO CODE REVIEW

• Improve maintainability / quality of code

• Find defects that can not be found by testing (automated or manual)
OTHER EFFECTS OF CODE REVIEW

• Improve knowledge of code and changes in team
• Create a more uniform code base
WHAT CODE REVIEW DOES NOT DO

• Find all bugs before going into production
• Save time on testing / testers
• ‘Check on the new guy’
• Enforce check lists
ARGUMENTS AGAINST CODE REVIEW

• ‘I’m a good developer, I don’t need my code reviewed’

• Research shows defects are found by reviewers in all code, from experienced to non-experienced developers

• Reviews take time

• Maintainability issues can make adding new functionality 28% slower and fixing errors 36% slower (See Bandi, Vaishnavi, Turk: Predicting Maintenance Performance Using OO Design Complexity Metrics)
TAKE AWAY

• Code reviews find defects that cannot easily be found in other ways

• Even though defects are subjective, people mostly agree
Some Useful Links
Stuck To The End

- Critic (Git, weak code ownership): https://github.com/jensl/critic
- Gerrit (Git): https://github.com/jensl/critic
- Review Board (all): http://www.reviewboard.org

@avl7771
avlleeuwen@piwebs.com