Reviewing Changes

Yegor Bugayenko

Lecture #4 out of 8
80 minutes

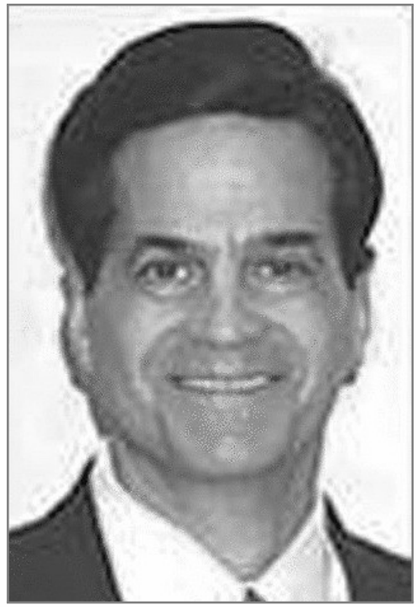
The slidedeck was presented by the author in this YouTube Video

Martin Fowler

"We should remember that pre-integration review grew out of an open-source context where contributions appear impromptu from weakly connected developers."

— Martin Fowler. Continuous Integration. http://martinfowler.com/articles/continuousIntegration.html, 2006. [Online; accessed 07-02-2024]

1. Raise issues, don’t resolve them! [Bugayenko, 2015a]



Michael Fagan

"The inspection is not intended to redesign, evaluate alternate design solutions, or to find solutions to errors; it is intended just to find errors!"

— Michael Fagan. Design and Code Inspections to Reduce Errors in Program Development. *IBM Systems Journal*, 38(3):258–287, 1999. doi:10.1147/sj.382.0258

Frank A. Ackerman

“Regardless of the application or the language, you can expect inspections to find from seven to 20 major defects per thousand noncomment lines of source code and to find major defects at a cost of one to five staff-hours.”

— A. Frank Ackerman, Lynne S. Buchwald, and Frank H. Lewski. Software Inspections: An Effective Verification Process. *IEEE Software*, 6(3):31–36, 1989. doi:10.1109/52.28121

Mateus Freira dos Santos

"In software projects with less than 34k lines of code, the number of developers that never contribute again after receiving a negative comment on the first pull request is 10.97%; this number more than doubles to 24.02% when evaluating projects with more than 197k lines of code."

— Mateus Freira, Josemar Caetano, Johnatan Oliveira, and Humberto Marques-Neto. Analyzing the Impact of Feedback in GitHub on the Software Developer's Mood, 2018

2. Educate the author

Andrew Sutherland

"The meat of the code review dialog, no matter what the medium, is the articulation of design rationale... Engineers find code review dialogs useful for a variety of purposes, but for understanding design rationale more than any other."

— Andrew Sutherland and Gina Venolia. Can Peer Code Reviews Be Exploited for Later Information Needs? In *Proceedings of the 31st International Conference on Software Engineering: Companion Volume*, pages 259–262. IEEE, 2009. doi:10.1109/ICSE-COMPANION.2009.5070996

Brendan Cleary

"'Raise issues, don't resolve them.' — this mentality limits a group's ability to collectively solve problems and mentor developers."

— Peter Rigby, Brendan Cleary, Frederic Painchaud, Margaret-Anne Storey, and Daniel German. Contemporary Peer Review in Action: Lessons From Open Source Development. *IEEE Software*, 29(6):56–61, 2012. doi:10.1109/MS.2012.24

Alberto Bacchelli

"Our results show that, although the top motivation driving code reviews is still finding defects, the practice and the actual outcomes are less about finding errors than expected: Defect related comments comprise a small proportion and mainly cover small logical low-level issues."

— Alberto Bacchelli and Christian Bird. Expectations, Outcomes, and Challenges of Modern Code Review. In *Proceedings of the 35th International Conference on Software Engineering*, pages 712–721. IEEE, 2013. doi:10.1109/ICSE.2013.6606617

Peter C. Rigby

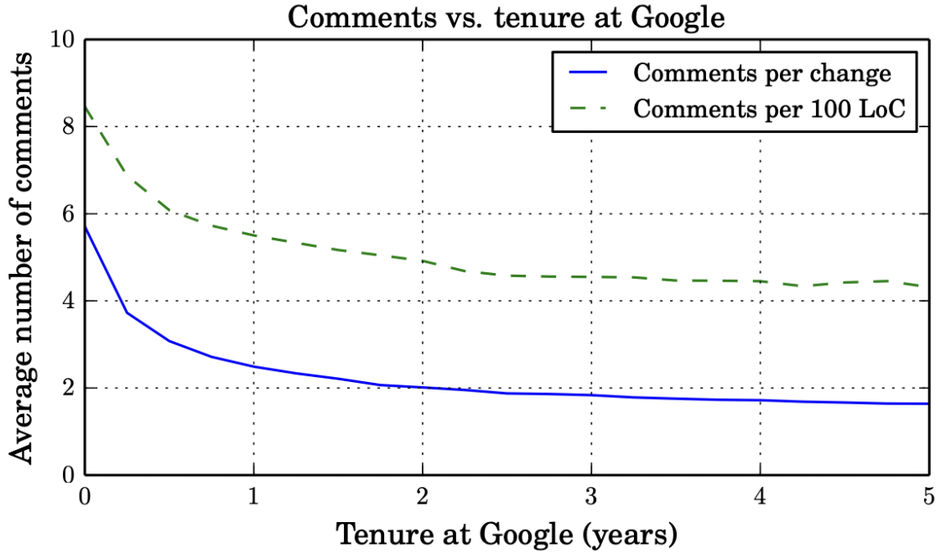
"Contemporary review is performed regularly and quickly just before the code is committed instead of when a larger work product is complete as in inspection. Contemporary reviewers prefers discussion and fixing code over reporting defects."

— Peter C. Rigby and Christian Bird. Convergent Contemporary Software Peer Review Practices. In *Proceedings of the 9th Joint Meeting on Foundations of Software Engineering*, pages 202–212, 2013. doi:10.1145/2491411.2491444

Caitlin Sadowski

"As developers build experience working at Google, the average number of comments on their changes decreases... Developers at Google who have started within the past year typically have more than twice as many comments per change."

— Caitlin Sadowski, Emma Söderberg, Luke Church, Michal Sipko, and Alberto Bacchelli. Modern Code Review: A Case Study at Google. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice*, pages 181–190, 2018. doi:10.1145/3183519.3183525



Source: Caitlin Sadowski, Emma Söderberg, Luke Church, Michal Sipko, and Alberto Bacchelli. Modern Code Review: A Case Study at Google. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice*, pages 181–190, 2018. doi:10.1145/3183519.3183525

3. Don’t run the code in the branch [Bugayenko, 2019].

4. Reject it, if it's too big [Bugayenko, 2015b].

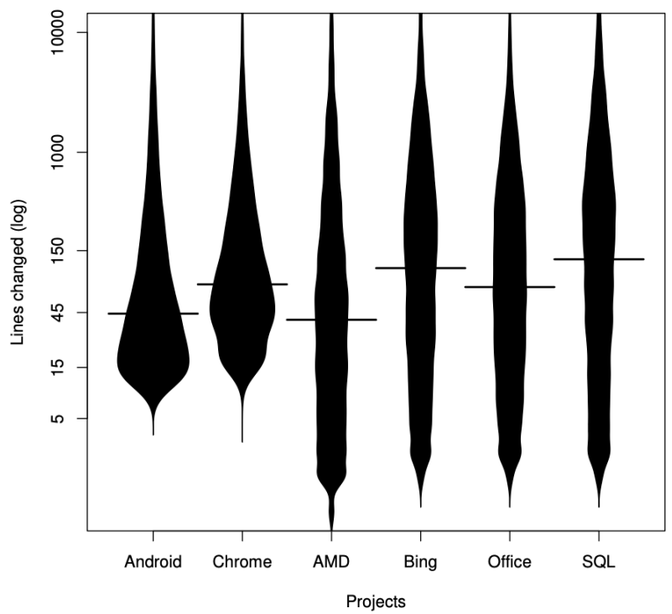
"Good programmers know what to write. Great ones know what to rewrite (and reuse)."

— Eric Raymond. The Cathedral and the Bazaar. *Knowledge, Technology & Policy*, 12(3):23–49, 1999. doi:10.1007/s12130-999-1026-0

Frederic Painchaud

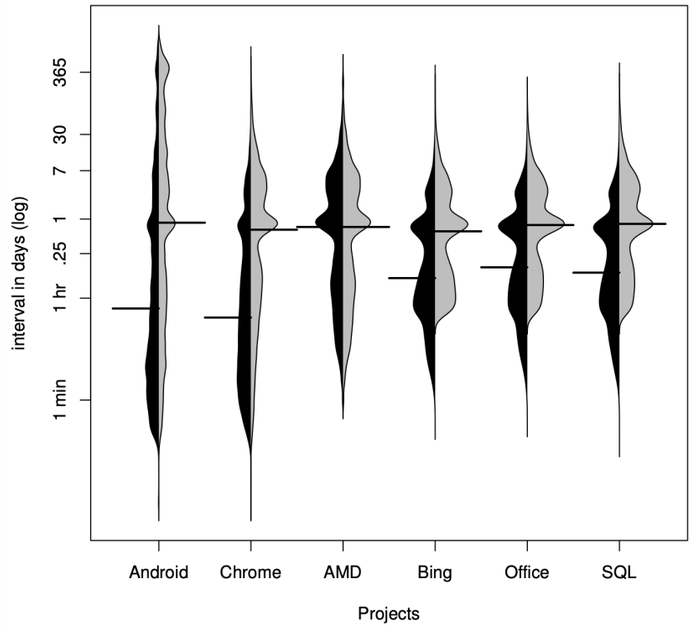
"To facilitate early and frequent feedback, OSS projects tend to review smaller changes than proprietary projects, ranging from 11 to 32 lines in the median case. The small size lets reviewers focus on the entire change, and the incrementality reduces reviewers' preparation time and lets them maintain an overall picture of how the change fits into the system."

— Peter Rigby, Brendan Cleary, Frederic Painchaud, Margaret-Anne Storey, and Daniel German. Contemporary Peer Review in Action: Lessons From Open Source Development. *IEEE Software*, 29(6):56–61, 2012. doi:10.1109/MS.2012.24



"Both Android and AMD have a median change size of 44 lines. This median change size is larger than Apache, 25 lines, and Linux, 32 lines, but much smaller than Lucent where the number of non-comment lines changed is 263 lines. Bing, Chrome's median change is 78 lines and includes 5 files."

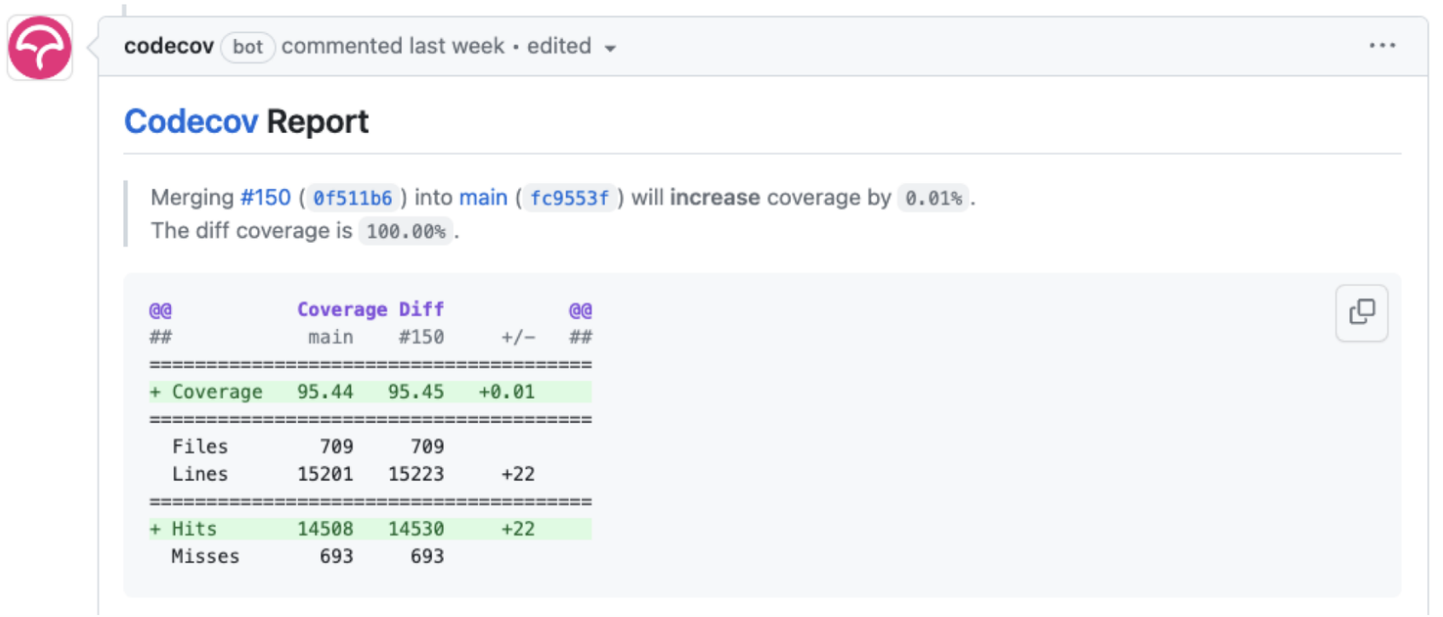
Source: Peter C. Rigby and Christian Bird. Convergent Contemporary Software Peer Review Practices. In *Proceedings of the 9th Joint Meeting on Foundations of Software Engineering*, pages 202–212, 2013. doi:10.1145/2491411.2491444



"AMD has short review intervals, with the median review taking 17.5 hours. Bing, SQL, and Office: 14.7, 19.8, and 18.9 hours respectively. The median completion time is 15.7 and 20.8 hours, for Chrome and Android, respectively."

Source: Peter C. Rigby and Christian Bird. Convergent Contemporary Software Peer Review Practices. In *Proceedings of the 9th Joint Meeting on Foundations of Software Engineering*, pages 202–212, 2013. doi:10.1145/2491411.2491444

5. Reject it, if it lowers code coverage [Bugayenko, 2015b].



codecov bot commented last week • edited

Codecov Report

Merging #150 (0f511b6) into main (fc9553f) will **increase** coverage by 0.01%.
The diff coverage is 100.00%.

```
@@            Coverage Diff            @@
##             main    #150     +/-   ##
=========================================
+ Coverage    95.44   95.45   +0.01
=========================================
  Files          709     709
  Lines        15201   15223     +22
=========================================
+ Hits         14508   14530     +22
  Misses         693     693
```

https://docs.codecov.com/docs/pull-request-comments

6. Reject it, if it doesn't reproduce a bug [Bugayenko, 2015b].

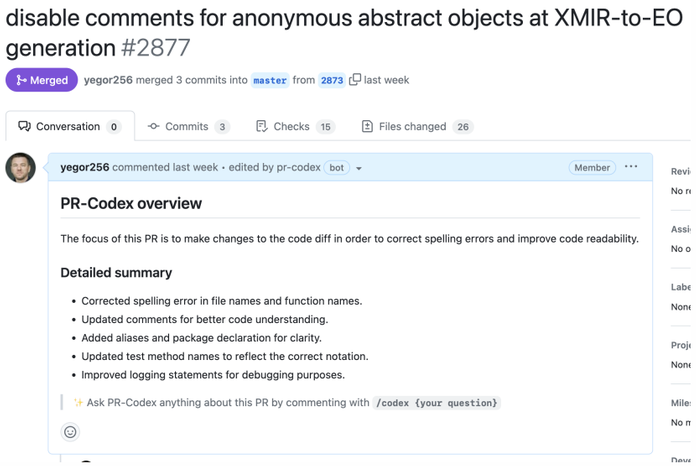
7. Rely on the CI status, but not too much

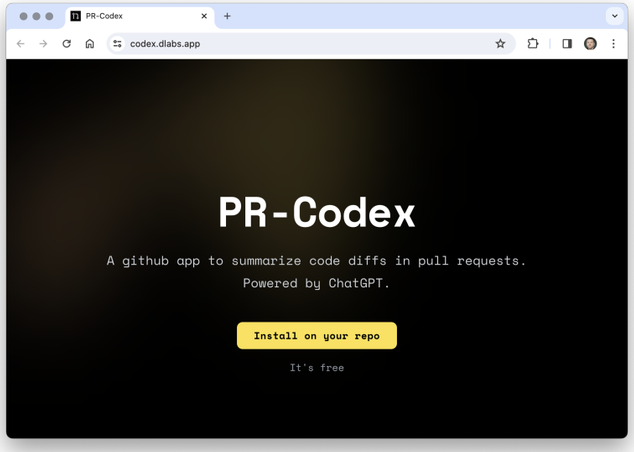
Mairieli Wessel

"Our findings also suggest that the adoption of GitHub Actions leads to more rejections of pull requests (PRs), more communication in accepted PRs and less communication in rejected PRs, fewer commits in accepted PRs and more commits in rejected PRs, and more time to accept a PR."

— Mairieli Wessel, Joseph Vargovich, Marco A. Gerosa, and Christoph Treude. GitHub Actions: The Impact on the Pull Request Process. *Empirical Software Engineering*, 28(6):131, 2023. doi:10.1007/s10664-023-10369-w

8. Employ ChatGPT


disable comments for anonymous abstract objects at XMIR-to-EO generation #2877
Merged
yegor256 merged 3 commits into master from 2873 last week
Conversation 0
Commits 3
Checks 15
Files changed 26
yegor256 commented last week • edited by pr-codex bot
Member
PR-Codex overview
The focus of this PR is to make changes to the code diff in order to correct spelling errors and improve code readability.
Detailed summary
Corrected spelling error in file names and function names.
Updated comments for better code understanding.
Added aliases and package declaration for clarity.
Updated test method names to reflect the correct notation.
Improved logging statements for debugging purposes.
Ask PR-Codex anything about this PR by commenting with /codex {your question}


PR-Codex
codex.dlabs.app
PR-Codex
A github app to summarize code diffs in pull requests.
Powered by ChatGPT.
Install on your repo
It's free

References

A. Frank Ackerman, Lynne S. Buchwald, and Frank H. Lewski. Software Inspections: An Effective Verification Process. *IEEE Software*, 6(3):31–36, 1989. doi:10.1109/52.28121.

Alberto Bacchelli and Christian Bird. Expectations, Outcomes, and Challenges of Modern Code Review. In *Proceedings of the 35th International Conference on Software Engineering*, pages 712–721. IEEE, 2013. doi:10.1109/ICSE.2013.6606617.

Yegor Bugayenko. Four NOs of a Serious Code Reviewer. `https://www.yegor256.com/150209.html`, feb 2015a. [Online; accessed 08-02-2024].

Yegor Bugayenko. A Few Valid Reasons to Reject a Bug Fix. `https://www.yegor256.com/150622.html`, jun 2015b. [Online; accessed 08-02-2024].

Yegor Bugayenko. Does Code Review Involve Testing? `https://www.yegor256.com/191203.html`, dec 2019. [Online; accessed 08-02-2024].

Michael Fagan. Design and Code Inspections to Reduce Errors in Program Development. *IBM Systems Journal*, 38(3):258–287, 1999. doi:10.1147/sj.382.0258.

Martin Fowler. Continuous Integration. `http://martinfowler.com/articles/continuousIntegration.html`, 2006. [Online; accessed 07-02-2024].

Mateus Freira, Josemar Caetano, Johnatan Oliveira, and Humberto Marques-Neto. Analyzing the Impact of Feedback in GitHub on the Software Developer's Mood, 2018.

Eric Raymond. The Cathedral and the Bazaar. *Knowledge, Technology & Policy*, 12(3):23–49, 1999. doi:10.1007/s12130-999-1026-0.

Peter Rigby, Brendan Cleary, Frederic Painchaud, Margaret-Anne Storey, and Daniel German. Contemporary Peer Review in Action: Lessons

From Open Source Development. *IEEE Software*, 29(6):56–61, 2012. doi:10.1109/MS.2012.24.

Peter C. Rigby and Christian Bird. Convergent Contemporary Software Peer Review Practices. In *Proceedings of the 9th Joint Meeting on Foundations of Software Engineering*, pages 202–212, 2013. doi:10.1145/2491411.2491444.

Caitlin Sadowski, Emma Söderberg, Luke Church, Michal Sipko, and Alberto Bacchelli. Modern Code Review: A Case Study at Google. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice*, pages 181–190, 2018. doi:10.1145/3183519.3183525.

Andrew Sutherland and Gina Venolia. Can Peer Code Reviews Be Exploited for Later Information Needs? In *Proceedings of the 31st International Conference on Software Engineering: Companion Volume*, pages 259–262. IEEE, 2009. doi:10.1109/ICSE-COMPANION.2009.5070996.

Mairieli Wessel, Joseph Vargovich, Marco A. Gerosa, and Christoph Treude. GitHub Actions: The Impact on the Pull Request Process. *Empirical Software Engineering*, 28(6):131, 2023. doi:10.1007/s10664-023-10369-w.