Reviewing Changes

Yegor Bugayenko

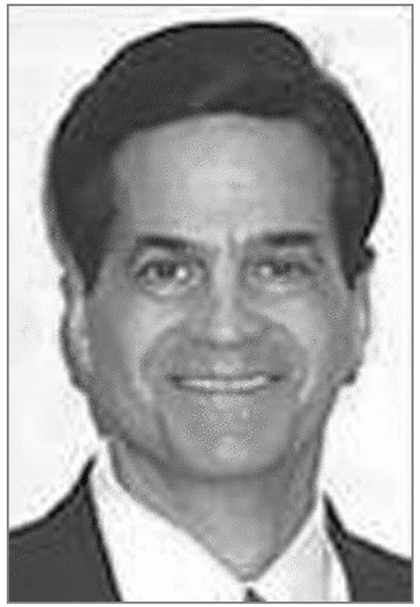
Lecture #4 out of 8
80 minutes

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]

.



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*, (3), 1976

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

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.”

—

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

Frank A. Ackerman

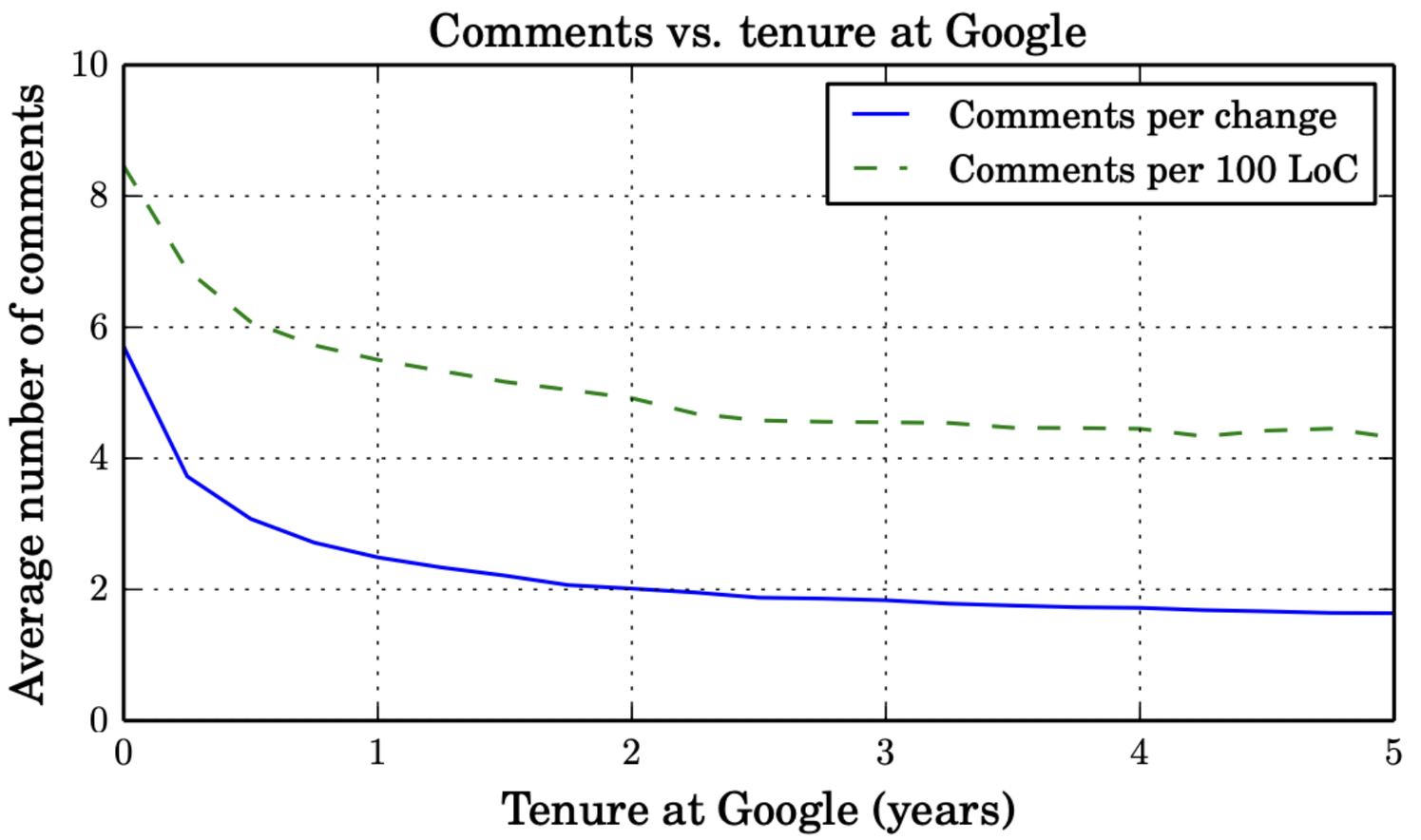
"Regardless of the application or the language, you can expect inspections to find from seven to 20 major defects per thousand noncomment lines of source code and to find major defects at a cost of one to five staff-hours."

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

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


Comments vs. tenure at Google
Comments per change
Comments per 100 LoC
Average number of comments
Tenure at Google (years)
0
2
4
6
8
10
0
1
2
3
4
5

"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

1. Don’t run the code in the branch[1]

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

2. 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

3. Make not compromises[2]

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

4. Have no fear[3]

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

5. Reject it, if it doesn't reproduce a bug[4]

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

6. Reject it, if it lowers code coverage[5]

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

References

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

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.

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

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

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

Michael Fagan. Design and Code Inspections to Reduce Errors in Program Development. *IBM Systems Journal*, (3), 1976.

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

Eric Raymond. The Cathedral and the Bazaar. *Knowledge, Technology & Policy*, 12(3):23–49, 1999.

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.

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.

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.