

Modern Code Review: A Case Study At Google

Sadowski, Caitlin, et al. "Modern code review: a case study at google." *Proceedings of the 40th international conference on software engineering: Software engineering in practice*. 2018.

PROBLEM BEING SOLVED

Code Review Processes and their corresponding research in history

Formal Code Inspections

Asynchronous Reviews

Tool Based Reviews

Pull-based reviews

Convergent Practices in Code Review ('13)

No focused or longitudinal perspective



id	Convergent Practice
CP ₁	Contemporary peer review follows a lightweight, flexible process
CP ₂	Reviews happen early (before a change is committed), quickly, and frequently
CP ₃	Change sizes are small
CP ₄	Two reviewers find an optimal number of defects
CP ₅	Review has changed from a defect finding activity to a group problem solving activity

SOLUTION

RQ1: What are the motivations for code reviews at Google?

12 face-to-face interviews

RQ2: What is the practice of code review at Google?

Analyze Critique Logs (9M reviews over 2 years)

RQ3: How do Google developers perceive code reviews?

Aforementioned Interviews + 44 surveys

RESULTS

Mainly group problem solving activity

Ensure code readability, maintainability + educational value

Lightweight and flexible

Also lightweight and flexible

Review speed, size and time invested

Reviews happen much quicker and have smaller changes, dev spend less time reviewing per week

Two reviewers are optimal

1 reviewer is deemed as sufficient in most cases

Breakdown reasons + satisfaction

Distance, Tone and Power, Mismatched expectations, Context

POSITIVE POINTS

Performs detailed interviews (12) to have qualitative insights + 9M logs for quantitative insights

A thorough picture of how code review happens in Google: CRITIQUE, Static Analysis

Insights about Breakdowns

Interview strategy

NEGATIVE POINTS

No quantitative insights about how well the code review strategy works

The study was lacking in longitudinal analysis

Self Selection Bias mitigation

FUTURE WORK

An automated tool to address the communication related issues in the code review process at Google

Use quantitative data to correlate review practices (e.g., review comment types, size of changes) with long-term code quality metrics (e.g., bug rates, maintainability) in Google

RATING

4 / 5

A very interesting read!

DISCUSSION POINTS

What are the pros and cons for using Snowball sampling for interviews? Could we make do with random sampling?

Can having questions on recent code changes (in the survey) lead to a recency bias?

Could different insights be gained from specialized code reviews (security/performance teams)?

Is breaking down reviews to smaller more isolated changes always helpful?