

Expectations, Outcomes, and Challenges of Modern Code Review

I went into the paper with the same intuition that the authors initially had, that code review was more about finding defects than anything else, but as I read on, I kept agreeing and changing my mind along with the authors as they presented their findings, which I found made the paper more enjoyable to read.

Microsoft has a lot of diverse teams so limiting the scope to just this company would help simplify their process while still having a diverse pool of people. However, I do see a potential flaw in this reasoning, in that each company has its own culture. Some companies might be more lax with their code reviews and some might be stricter, and each may have different goals when they do it, so restricting the research to one company might have ensured that all the interviewed people had very similar experiences with code review as compared to people at other companies.

Modern Code Reviews in Open-Source Projects: Which Problems Do They Fix?

I was quite surprised that out of the 30 popular repositories that the authors studied, only two followed the mandatory continuous code review that they all had claimed to do.

However, this made me think that the fact that these were the only two that actually adhered to doing what they had claimed to do, actually made them the outliers among the projects that were considered. They obsessively reviewed everything that was committed, when most of the other popular repositories did not. I feel this makes the other repositories more interesting to study since that is how I'd assume most modern code bases also would work.

I don't mean that the research done on these two were not valuable, I did find these to still answer the research questions to a large extent. I just think that these two represent that small subsection of projects that are already more likely to adhere to coding and reviewing standards.

The Impact of Adopting Modern Code Review on Software Projects

The few issues I might have had with the paper are all mostly addressed in the paper's Threats to Validity section, so I do find that the authors are aware of potential problems that might be there in the paper.

While the paper did good research and answered the research questions satisfactorily, I felt that at the end of the paper I didn't end up learning much. I don't feel like this paper affected my views on Modern Code Review in one direction or the other. That in itself is not a bad thing and research that does convince the user to think in a new way is by no means bad. I do feel like this paper in the long run will be quite valuable as a paper that future research refers to and builds on.