

The Dangers of Change Risk Approval

Code is a sprawling endeavor with thousands of lines and potentially hundreds of employees working on a product that millions of customers use. It's expected that not every code commit will be perfect, and the involvement of one change might have huge implications somewhere else in the code base. With so much at stake, it makes sense to have a review process before pushing those code changes into production. In what ways can reviewing these crucial changes be problematic?

"If your most talented developers endure inefficient processes long enough, their passion and productivity will wane. Many will leave. Failure to attract and retain talent puts your business at risk."
-LaunchDarkly

Code reviews can be cumbersome. Internal and external entities and long chains of approvals can be involved, depending on the impact of the code change. At the same time, other engineers are waiting for those adjustments to be committed to production and are working on old versions that don't include them. Behind that work, changes pile up, causing faulty reviews to speed through and things get missed. It's a bloated, inefficient system that stresses everyone involved.

To remedy this, step one is to reduce the overhead of a review. Coding in pairs or directly to one person consistently ensures better work in smaller commits. Smaller commits are more likely to succeed and cause fewer outages and problems in production. Having someone review your work regularly also puts a higher onus on the developer to test and code more rigorously. These commits will likely succeed due to the peer review process.

The second step is to categorize code changes better. How many customers will it impact, what is the security risk, etc? The lower-scoring items might not need such an official approval scheme. Approve locally for faster turnaround times, keeping up with the code structures and committing them to the trunk so all engineers working on the same base will work with recent code.

A third step is to find ways to automate the approval processes. Parts and pieces that can be automatically tested bring safety of mind to developers and the approvers. If the tests are built well and are far-reaching, the code changes can be trusted to be pushed into production.

Overall, code reviews aim to ensure that code is performing as well as it can. Designing a system around that goal helps every part of the process work more smoothly.

Citations

(n.d.). *Reconciling change management and continuous delivery*. Launch Darkly. Retrieved February 22, 2025, from <https://launchdarkly.com/guides/reconciling-change-management-and-continuous-delivery/the-downsides-of-heavy-change-management/>

(n.d.). *How to improve your IT support workflow*. Atlassian. Retrieved February 22, 2025, from <http://atlassian.com/itsm/change-management>

(2022, June 23). *Why your change process is too slow – and how to fix it*. Top Desk. Retrieved February 22, 2025, from <https://www.topdesk.com/en/blog/change-proces/>