\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dangers of Change Approval Processes

Kimberly Orozco

CSD380-T301 DevOps

Bellevue University

October 25, 2025

Professor Vianelis Martinez

# Introduction

Change approval processes are supposed to keep systems safe and make sure people follow the rules. But sometimes these processes are too strict and confusing which leads to problems for teams who need to perform quickly. This paper talks about the risks of traditional change approval processes, what research says about better methods, and what teams can do to make change safer and faster.

# What approval processes are meant to do

Approval processes are supposed to stop risky changes from breaking things and make roles and responsibilities clear. Older systems like ITIL and CABs were made for big companies that had lots of teams and strict rules to follow (LaunchDarkly, 2025). These kinds of approvals can still help when it comes to staying compliant and managing large projects.

# Main problems with strict approval process

**1. They slow everything down**

When teams have to wait for central boards or special people to approve every change, work starts to pile up. Teams often wait days or weeks so they group big changes together to save time.

**2. Bigger changes mean bigger problems**

When teams release lots of changes at once, one mistake can break many things at the same time. This makes failures harder to fix and more expensive.

**3. Work gets stuck**

If approvals take too long or are confusing developers might try to skip them or a few people can end up holding all the power to approve things. This creates delays and sometimes leaves information missing or wrong. CMW Lab mentions problems like wasted time, lack of flexibility, and even lost information in older systems (Pavel, 2025).

**4. Developers can get stressed out**

Nobody likes to wait for approval for every small change. It makes work stressful and boring which can cause lower motivation or even make people quit.

# What research says works better

**Peer review and automation work better than one big approval board.**

In the DORA article, they explain that it’s more effective when review happens during development, like in code reviews or continuous integration (DORA, nd.). Peer reviews make sure no one pushes risky code alone and automated testing finds problems faster. This helps teams release updates more often. This also helps keep productivity levels steady.

**Feature Flags make changes safer.**

Feature flags let teams turn off or turn on new code without rolling back everything. This means teams can deploy changes but keep them hidden until they’re ready. If something breaks, it can be turned off right away. In the LaunchDarkly article they talk about how using feature flags helps teams reduce failures and fix problems faster (LaunchDarkly, 2025).

# Real examples

CMW Lab talks about how old approval systems caused a lot of problems like delays, lost information, and bad communication between teams. Sometimes changes got stuck waiting for one person to approve them or people used tools that didn’t work well together. This made it hard to keep track of everything and wasted a lot of time. Automation helped fix these problems by making the process faster and more organized. DORA also found that automated systems work better and have fewer mistakes. LaunchDarkly says using feature flags and simple workflow tools helps teams stay safe and follow the rules while still moving fast. It’s shown, clearly, in my three sources that too many manual approvals slow teams down and cause more problems than they solve.

# Conclusion

Approval processes are still important for safety and compliance, but when they are too slow or strict they cause more problems than they solve. Research shows that peer reviews, automation, and feature flags are better ways to stay safe while moving fast. Teams should track how their approvals affect time and failure rates and use automation to make things easier.

**References**

*Capabilities: Streamlining change approval.* DORA. (nd.) <https://dora.dev/capabilities/streamlining-change-approval/>

LaunchDarkly. (2025, October 24). *The downsides of heavy change management*. <https://launchdarkly.com/guides/reconciling-change-management-and-continuous-delivery/the-downsides-of-heavy-change-management/>

Pavel. (2025, April 3). *What is approval management? benefits, challenges*. CMW Lab Blog. <https://www.cmwlab.com/blog/approval-management-benefits-hidden-dangers/>