Keanu Foltz
Module 8
7/13/25

# Dangers of the Change Approval Process

Change approval processes in Software development and IT were created to reduce risk and make deployments safer and more thought out. However with the acceleration of software delivery and competition between businesses in the constantly changing tech market, the practices of change approval processes in tech can cause more problems than they solve. Based off of research and notable incidents, rigid change controls have been found to slow delivery, increase failure, reduce business performance, and alienate developers. For this reason we must be wary of the effect change approval processes can have.

The most notable change management failure was mentioned in The DevOps Handbook. A flawed 15 minute software deployment caused $440 million in trading losses, causing the company to be sold so it would not go under. The causes were cut down to either a failure of change controls or testing but both have the common issue of poor feedback and slow detection. In situations like this adding more oversight and bureaucracy would cause more issues than they solve, controls meant to prevent failure were found to create distance between implementers and decision makers and would increase the chances of failure. This was especially true when the controls hindered agility and feedback.

The 2019 State of DevOps Report by DORA found that the best performers in software delivery tended to operate with lightweight and streamlined change approval processes. These organizations are more fast, resilient and aligned with business goals than teams that relied on approvals from C-level or external review boards. These teams that relied heavily on approval from high ups tended to have longer deployment times, lower productivity, and higher rates of failure. Similar findings were noted in other reports such as the report on the state of DevOps done by Puppet in 2014 which found that high performing teams relied more on peer reviews than centralized control boards.

Despite the evidence, many organizations still rely heavily on Change Advisory Boards or formal request for change procedures. CABs tend to operate at a high level, reviewing summaries and checklists rather than actual code. These safeguards have been found to be ineffective with predicting high risk changes. With how complex modern software is, there are many unintended effects caused by even changes that could be considered safe. With CABs, since the people authorizing changes are extremely far removed from the system, they have very little ability to anticipate the consequences of the changes they are approving. When the developers themselves do not have the authorization to make changes quickly or recover from failure fast, this becomes a big problem. When developers have to wait days or weeks for approvals, the whole process slows down and user feedback gets delayed. The lack of user feedback makes it much more difficult to understand what works and what doesn’t which causes resources to be wasted and opportunities to be missed. Other than the immediate dangers of inefficiency, heavy change approval processes can also hurt the moral and retention of developers. When developers feel burdened and held back by bureaucracy, they are likely to leave which hurts the stability of the business.

Change approval processes need to evolve or they will end up hurting the businesses that use them. When top level managers and review boards are in charge of approving changes that they are far removed from, we end up with a slower software delivery cycle, demotivated developers, and poor overall performance. Time gets wasted, flexibility is hurt, and we risk information loss or errors.

References:
https://www.cmwlab.com/blog/approval-management-benefits-hidden-dangers/

https://launchdarkly.com/guides/reconciling-change-management-and-continuous-delivery/the-downsides-of-heavy-change-management/

https://octopus.com/blog/change-advisory-boards-dont-work