Austen Rhyce Erickson
CSD380: DevOps
Assignment 8.2: The Dangers of Change Approval Processes
2/22/25

In DevOps, a change approval process is a set of steps that each proposed software change must go through before it is cleared for deployment. Change approval is like a system of checks and balances for quality control for the purpose of mitigating risks and maintaining system stability. It's important that code updates are rolled out in a controlled manner so that the outcome in the production environment can be predicted. This being said, with all the practical reasons for change approval, there are some lurking dangers to be aware of.

One such danger is the slowing down of the development cycle. Speediness is a main focus of DevOps and a change approval is, by definition, a speed bump or tollbooth on the road of deployment. Yes, a necessary evil as we don't want to skip change approval in the name of speediness but an overly cumbersome approval process can impede the ability to release new features or patches in an unnecessary way. For example, if approvals are required before making changes, the feedback loop becomes longer which increases lead time. Developers have to wait to assess the impact of their changes. A slow down in development like this could result in missed opportunities.

Another danger is bottlenecks. If an organization is overly bureaucratic, there may be complex approval chains. Delays accumulate into bottlenecks when waiting on multiple stakeholders to sign off on changes. A buildup of changes waiting to be launched is not in the spirit of DevOps. Additionally, if a specific person is responsible for an approval sign off, progress can be halted and deadlines can be missed if that person happens to be unavailable for some reason like being sick or on vacation. Depending on the nature of the software service, a fix not going out in a timely manner could have serious consequences. In addition to bottlenecks, it's worth mentioning that there is always the risk of human error. If changes are not

tracked accurately or approval statuses aren't properly updated in a sequential manner, changes may either be inadvertently delayed or worse, changes launched without approval.

Change approval can also cause issues by being seen as an annoying ordeal. In this way, developers may resist proposals of new approaches lest they contend with copious approval stepping stones. This can stifle innovation and open-mindedness to experimentation. Complacency becomes normalized and change is scrutinized. This is another strike against the spirit of DevOps and continuous improvement. A nasty alternative to disinterested developers is attempts on the part of developers to circumvent lengthy approval processes and sneakily deploy changes that are out of compliance with policies or regulations.

Another common frustration with change approval processes is miscommunication between leaders and development teams. There can be misunderstandings about why certain changes were or weren't made. This lack of transparency in decision-making can erode trust and morale. Miscommunication can also result in approval teams lacking the full context or awareness of technical situations. The outcome of this can be decisions that aren't aligned with the reality of the physical system.

To minimize the unpleasantries of change approval processes, organizations should look to streamline and automate the change approval processes. When you already have your ducks in a row, approval can march forth as intended without delay. Automated tests in a CICD pipeline execute in a consistent fashion. In this way, initial assessment can be trusted and reduce the need for manual approval while still maintaining quality standards. Governance should aim to find a balance between oversight and autonomy of developers. The approval process should be effective in a way that doesn't slow down workflows. Furthermore, when developers are included in the change approval conversations, there is greater trust and less resistance. Things happen more quickly – DevOps and CICD are all about quality speediness. Keeping this balance ensures the development process in relation to change approval avoids bottlenecks, inefficiencies, and frustration. The effort of keeping the balance is in itself a form of

engineering. It makes sense that DevOps is considered a specific engineering discipline in the sphere of software engineering and production.

Sources:

Nyo, Isabel. “Best Practices for Change Management in the Age of DevOps - Work Life by Atlassian.” *Work Life by Atlassian*, 13 May 2020, www.atlassian.com/blog/atlassian-engineering/best-practices-for-change-management-in-the-age-of-devops.

Long, Mike. “Change Control Doesn’t Work: When Regulated DevOps Goes Wrong.” *Dzone.com*, DZone, 18 May 2023, dzone.com/articles/change-control-doesnt-work.

“Change Management Is Broken: Here’s How to Fix It - DZone DevOps.” *Dzone.com*, dzone.com/articles/change-management-is-broken-heres-how-to-fix-it.

“Breaking the Change Management Barrier.” *IT Revolution*, 2022, itrevolution.com/articles/breaking-the-change-management-barrier/.

Kozlov, Serhiy. “Managing Change in DevOps: Strategies for Smooth Transitions.” *Romexsoft*, 28 Aug. 2023, www.romexsoft.com/blog/managing-change-in-devops/.