Melissa Lawrence, CSD 380, Assignment 8, 12/08/2024

Change approval processes are formal procedures used to review and authorize changes before they are implemented in a system or project. These processes are crucial in ensuring that changes are necessary, well-planned, and do not introduce new risks or issues. In a DevOps context, change approval processes help maintain the stability and reliability of software systems while allowing for continuous improvement and deployment. However, these processes can also introduce significant risks if not managed properly.

When considering inefficiency and delays, traditional change approval processes often involve multiple levels of management and large committees. This can slow down the implementation of changes, leading to missed deadlines and increased costs. I have seen this in my career. The adage “too many cooks in the kitchen spoils the soup” usually comes to mind when there are too many managers for the same people. These delays can be particularly detrimental in fast-paced environments where timely updates are crucial.

Another significant danger is the overburdening of approvers. When too many change requests require approval, it can overwhelm the approvers. This can result in oversight, improper delegation, or less thorough reviews, potentially leading to costly outages and downtime. While having too many managers can be damaging, they still need support. Having a project lead to filter items as they head up the chain can be efficient when a manager oversees multiple project groups. However, without such support, the risk of errors and missed critical issues increases.

Moreover, a complex and rigid approval process can lead to resistance to change among employees. This resistance can hinder the adoption of necessary changes and negatively impact the overall efficiency of the organization. I have seen change initiatives fail due to poor communication of the hows, whys, and most importantly, the benefits to the hourly employees, leading to a lack of buy-in from those employees. This resistance can stall important updates and innovations.

In addition to resistance, a lack of flexibility in the change approval process can prevent organizations from quickly adapting to new circumstances or opportunities. This inflexibility can be detrimental in fast-paced environments where agility is crucial. The mindset of “my way or the highway” leads to a lack of group or business alignment, stifling creativity and responsiveness.

Furthermore, the complexity of the approval process can increase the risk of errors in the implementation of changes. These errors can have significant negative impacts on the organization, including system failures and security vulnerabilities. Streamlining change approval processes to increase efficiency, reduce risks, and bring consistency leads to a better understanding of the approval process. Otherwise, from my experience, it can feel like a coin flip each time a wishy-washy manager approves or disapproves things. This inconsistency can undermine the reliability of the system.

To illustrate how change approval processes can be managed effectively, consider the use of GitHub in DevOps. GitHub is a tool commonly used for managing change approval processes. When a developer wants to make a change, they create a pull request (PR) that includes the proposed changes and a description of what they do. Team members or specific approvers then review the PRs, providing feedback and ensuring the changes meet project standards. Once reviewers are satisfied, they approve the PR, which can be configured to require multiple approvals. Before merging, automated tests ensure the new code doesn't introduce bugs or issues. Approved changes are then merged into the main codebase and can trigger automated deployment processes.

Using GitHub for change approval processes offers several benefits, including transparency, collaboration, automation, and efficiency. However, it's essential to balance automation with human oversight to avoid new risks. Even with tools like GitHub, the dangers of inefficiency, overburdening, resistance, inflexibility, and errors must be carefully managed.

In conclusion, while change approval processes are vital for maintaining quality and stability in DevOps, they must be designed to support, rather than hinder, the rapid and reliable delivery of software. By understanding and mitigating the dangers of these processes, organizations can achieve a balance between necessary controls and agility.

Resources:

DORA. (n.d.). Streamlining change approval processes. Retrieved from https://www.devops-research.com/research/streamlining-change-approval-processes

GitHub. (n.d.). Reviewing deployments. GitHub Documentation. Retrieved from https://docs.github.com/en/actions/managing-workflow-runs-and-deployments/managing-deployments/reviewing-deployments

Myndbend. (n.d.). Best practices for IT change management approvals processes in help desk workflows. Myndbend. Retrieved from https://support.myndbend.com/hc/en-us/articles/200838923-Best-Practices-for-IT-Change-Management-Approvals-Processes-in-Help-Desk-Workflows

Microsoft. (n.d.). Approvals and checks in Azure Pipelines. Microsoft Learn. Retrieved from https://learn.microsoft.com/en-us/azure/devops/pipelines/process/approvals?view=azure-devops&tabs=check-pass