Rogelio Orozco
July 10, 2024
Module 8.2
CSD 380
Professor Sampson

Dangers of Change Approval Process

In the realm of DevOps, the change approval process plays a pivotal role in managing and controlling modifications to software and infrastructure. This process is designed to ensure that changes are thoroughly reviewed, tested, and authorized before being implemented in production environments. While change approval processes have their merits, they are not without drawbacks. This paper explores what the change approval process entails, its benefits, and the common pitfalls associated with it.

The change approval process is a systematic approach used to evaluate, approve, and implement changes within a software or IT environment. It typically involves several stages:

1. **Change Request**: Initiating a request for change (RFC) which outlines the proposed modification, its purpose, and potential impacts.
2. **Review and Assessment**: Evaluating the change request to determine its necessity, benefits, risks, and the resources required.
3. **Approval**: Gaining authorization from relevant stakeholders or a change advisory board (CAB) to proceed with the change.
4. **Implementation**: Executing the approved change in a controlled manner.
5. **Verification**: Testing the change to ensure it works as intended without causing disruptions.
6. **Documentation**: Recording details of the change for future reference and compliance purposes.

The primary aim of the change approval process is to minimize risks associated with changes, ensure consistency, and maintain the stability of the production environment. The change approval process offers several advantages such as risk mitigation, compliance and accountability, coordination, and quality assurance.

By thoroughly reviewing changes before implementation, organizations can identify and address potential issues, reducing the likelihood of unexpected problems in the production environment. Documenting and approving changes ensures that organizations comply with regulatory requirements and can demonstrate accountability for their actions.

The process fosters better communication and coordination among teams, ensuring that everyone is aware of upcoming changes and their potential impacts. Rigorous testing and review help maintain high-quality standards, ensuring that changes do not degrade the system's performance or reliability.

Despite its benefits, the change approval process can pose significant challenges, particularly in a fast-paced DevOps environment.

The change approval process can introduce delays, especially when multiple layers of approval are required. This can slow down the development and deployment cycles, hindering the agility that DevOps aims to achieve. In some cases, the wait time for approval can be longer than the time required to implement the change itself.

The process can become cumbersome and complex, requiring significant administrative effort. This overhead can be particularly burdensome for small teams or organizations with limited resources. The need to fill out extensive documentation and attend multiple review meetings can distract from actual development work.

When the approval process is perceived as too rigid or bureaucratic, it can create resistance among developers. They might view it as an obstacle rather than a safeguard, leading to frustration and potential attempts to circumvent the process.

Approvers who are not directly involved in the project may lack the context needed to make informed decisions. This can result in unnecessary rejections or approvals of changes that do not align with the project's goals or technical requirements.

The need for formal approval can stifle innovation and experimentation. Developers may be hesitant to propose new ideas or improvements if they anticipate a lengthy and uncertain approval process. This can hinder the overall progress and adaptability of the organization.

In some cases, the change approval process may be inconsistently applied. Some changes might be fast-tracked while others are subjected to rigorous scrutiny, leading to confusion and potential disparities in quality and risk management.

Organizations can adopt several strategies to address these challenges. Establishing clear criteria for what types of changes require approval and which do not can help reduce unnecessary delays. For example, minor changes or bug fixes could be pre-approved, while major changes undergo thorough review.

Leveraging automation tools can streamline the approval process, reducing manual effort and speeding up decision-making. Automated testing and continuous integration/continuous delivery (CI/CD) pipelines can help ensure changes meet quality standards without extensive human intervention.

Empowering development teams with the authority to approve certain changes can enhance agility. This approach requires a high level of trust and accountability, ensuring teams follow best practices and maintain quality standards.

Enhancing communication channels between developers and approvers can help provide the necessary context for informed decision-making. Regular meetings and collaborative tools can facilitate better understanding and coordination.

Continuously reviewing and refining the change approval process based on feedback and outcomes can help address its shortcomings. Implementing a culture of continuous improvement ensures the process evolves to meet the organization's needs.

While the change approval process is essential for managing risks and ensuring quality, it can pose significant challenges in a DevOps environment. By understanding these pitfalls and adopting strategies to mitigate them, organizations can strike a balance between control and agility. Ultimately, the goal is to foster a culture of collaboration, accountability, and continuous improvement, enabling teams to deliver high-quality software efficiently and effectively.

Works Cited:

Atlassian. (n.d.). *Change management*. Retrieved July 10, 2024, from https://www.atlassian.com/itsm/change-management

DORA. (n.d.). *Streamlining change approval*. Retrieved July 10, 2024, from https://dora.dev/capabilities/streamlining-change-approval/

Stahnke, M. (2021, January 22). *Change management is broken: Here's how to fix it*. DZone. Retrieved July 10, 2024, from https://dzone.com/articles/change-management-is-broken-heres-how-to-fix-it