CHYNNA LEE

MODULE 8 ASSIGNMENT

DEVOPS

JULY 8, 2024

Change management is often disliked due to its bureaucratic approval processes that hinder development, resulting in inefficiencies and frustrations among developers who prioritize rapid changes. Traditional methods require lengthy approvals from disconnected individuals, leading to increased inefficiency, audit findings, and higher incident rates. Developers are frequently required to manually request changes for each deployment, which reduces productivity and undermines visibility and control. Standard change templates help, but developers still need to exit their workflow, adding to the inefficiency. The need for orthodox approval methods involving large committees, multiple levels of management, and predefined windows to make changes can demotivate teams and stall progress. Misaligned incentives between development and operations, a lack of standardized change procedures, poor communication, and siloed teams further exacerbate these challenges.

To address these issues, organizations should adopt strategies such as standardizing deployment practices, accelerating deployment systems, delivering smaller batches frequently, speeding up feedback cycles, automating infrastructure delivery, and focusing on self-service. Effective change management involves aligning people to processes, ensuring developers and change managers understand and agree on how to populate change templates, and determining when to invoke automation platforms. By incorporating risk-based approval steps and integrating tools like Jira with ServiceNow, organizations can automate the change request process, reducing manual tasks and improving efficiency. Automation with platforms like ServiceNow can also streamline deployments, trigger automated tests, and update documentation, keeping developers informed and reducing administrative burdens.

Research by DevOps Research and Assessment (DORA) and the 2020 State of DevOps Report highlights the benefits of continuous integration and delivery (CI/CD) tools, standardized and automated workflows, and real-time feedback. These practices enhance performance, reduce risks, and ensure changes are implemented smoothly. Emphasizing smaller, more manageable goals, frequent deployments, and minimizing manual reviews and approvals can lead to higher change management success rates. Adopting proven CI/CD systems, focusing on self-service capabilities, and treating the development platform as a product for fast feedback are crucial for effective change management. By shifting detailed code reviews to practitioners and automated methods, organizations can free up leadership to focus on strategic work, aligning with the practices of high-performing organizations. Through these approaches, change management can become a more streamlined, efficient, and effective process that supports both development and operations.

* DevOps Research and Assessment. (2019). Streamlining change approval. Retrieved from <https://dora.dev/capabilities/streamlining-change-approval/>
* Plat4mation. (n.d.). How to close the gap between DevOps and change management. Retrieved from <https://plat4mation.com/blog/how-to-close-the-gap-between-devops-and-change-management/>
* Stahnke, M. (2021, January 22). Change management is broken: Here’s how to fix it. DZone. Retrieved from <https://dzone.com/articles/change-management-is-broken-heres-how-to-fix-it>