Victor Gregory   
CSD380-H330  
Professor Sampson

Module 12.2 Assignment

To survive in the ever-changing tech landscape of today, it is a fine line as to where a company stands between speed, trustworthiness, and security. For decades, traditional IT organizations have depended on process-oriented change management to reduce risk to operations and security. In contrast, leaders in the industry including Google, Amazon, and Netflix have adopted DevOps practices that empower them to deploy code thousands of times each day while sustaining system stability.

Chapter 23 of The DevOps Handbook (Kim et al., 2016) outlines DevSecOps and its integration into change management processes. We will also explore some of the chapter highlights, such as, frequency of change as a new change classification, a better way of evaluating quality, moving ahead of segregation of duties mentality, and How production telemetry provide evidence to support compliance.

The traditional IT service management (ITSM) frameworks, for example ITIL, categorize the changes as standard, normal and emergency changes. Though this framework creates structure, it can cause bottlenecks and lower agility:

* Standard Changes: Changes that have low risk, are repeatable, and are either pre-approved or do not require time-consuming approval processes.
* Normal changes: (HL — High Risk) These changes need to be assessed by a change authority, and they can delay the deployment.

Changes of this type require rapid approval and documentation after completion.

A key factor creating inefficiency in the traditional change management process is the need for Change Advisory Boards (CABs) who may not have the technical know-how to evaluate sophisticated software deployments; however, it is placed in their hands, nonetheless. This dependency causes longer lead times and inhibits innovation and responsiveness. DevOps removes automation of non-value-added manual approvals and streamlined processes. This method enables organizations to reassess many lower risk changes as standard changes which can be processed more quickly, thereby enhancing efficiency and enabling security to grow yet remain cost-effective. Automated deployment tools and configuration management provide consistency and reduce errors.

Normal changes should not have an uninspiring process of approval as an obstacle. CABs can’t be a crutch that drags down the decision-making process. What organizations need to do instead is introduce context about changes and use historical data to show how deployments are low-risk and error-free. TPM creates traceability between change sets user stories and defects. This enables better transparency and accountability. The goal is to build confidence that changes will behave correctly in production, minimizing the need for manual hotfixes. The extra information that is already available before the change will allow organizations to actually process change requests quicker by following and processing them.

Separation of Duties (SoD) to prevent fraud and mistakes Traditionally IT organizations have focused on implementing SoD to prevent fraud and mistakes. However, strict SoD policies may hinder deployment processes. DevOps provides different controls, for example:

* Pair-programming and continuous code-review
* Automated testing for spotting jitters before deployment
* Monitoring Activities in Real Time for Early Detection of Anomalies

They are more secure and allow for faster detection of an issue than traditional isolation techniques. This calculation may sound counterintuitive for regulated environments — replacing manual approval gates with automated validation and peer collaboration means they can be agile without compromising security.

Auditors and compliance officers have usually worked with snapshots of system configurations (screenshots, static configuration files, etc.) at a moment in time. On the other hand, DevOps organizations bake compliance into the software development lifecycle through telemetry in real-time and automated audit logs. It gives auditors self-service access to logs and system health, so the disruptive manual reviews become less frequent. One powerful case in point is the use of real-time monitoring for fraud detection. An example is fraudulent activations of ATM maintenance mode, which utils scanned reported failed because traditional manual after-the-fact reviews were not able to detect it. Nevertheless, some of these patterns were identified through real-time telemetry and that exposed the fraud, further confirming the importance of continuous monitoring as a proactive security control.

However, as explained in The DevOps Handbook (Kim et al., 2016), change management processes need to change to support contemporary DevOps methodologies. By reclassifying lower-risk changes which can be implemented as necessary, streamlining the approval workflow based on context and impact, eliminating the need for separation of duties where possible, and leveraging real-time telemetry to ensure that changes don’t cause harm, organizations can move towards a risk-based, pragmatic approach. As enterprises scale DevOps practices, focusing on these principles will ensure increased agility, reliability, and compliance in a digital-first world.

**References:**

Kim, G., Debois, P., Humble, J., & Willis, J. The DevOps handbook: How to create world-class agility, reliability & security in technology organizations. IT Revolution Press.