Paul Romer

CSD370

Module 11.2 Assignment

Operational Risk Management (ORM)

Requirements: Time for a little research. The Mesusa Corporation (MeCo) is not convinced that they have a good understanding of Operational Risk Management (ORM) during software deployment. Your task is to conduct research and find at least three (3) sources that provide information on deployment risk. Provide a summary of each source; does the source provide steps? does it provide best practices on reducing that risk? The last step would be to provide a comprehensive look at the sources. What do they have in common? Is there one source you would recommend to MeCo? Why or why not?

Software Deployment Security: Risks and Best Practices. devops.com

URL: https://devops.com/software-deployment-security-risks-and-best-practices/

Summary: This article focuses on the security risks of software development. It discusses the risks of code tampering, third-party vulnerabilities, configuration errors, inadequate environment isolation, and lack of monitoring. Some common strategies to mitigate this are canary deployment, where new software is rolled out to a small subset of users and then monitored for unexpected bugs or vulnerabilities. This limits exposure if something is found and if nothing is, the update it pushed to the broader userbase. Feature toggles allow developers to activate and disactivate features in the live environment. The main security risk with feature toggles is that if they are not correctly implemented, they can potentially expose unfinished or untested features to end-users, leading to potential vulnerabilities being exposed. Rolling deployment is a strategy of updating servers one by one, ensuring there is always a version available to the user. The primary security risk is that during the rollout, there is multiple active versions. If there is a vulnerability with one, it won't be remedied until the rollout is complete. A/B testing deployment is another method of operational risk management, where two versions of the software is run at the same time. These systems for deployment limit operational risk. Other best practices to limit operational risk are code reviews, automated security scans, environment hardening, principle of least privilege, secure configuration management, and immutable deployments.

AWS Operational Excellence:

URL: https://docs.aws.amazon.com/pdfs/wellarchitected/2022-03-31/framework/wellarchitected-framework-2022-03-31.pdf#operational-excellence

Summary: This resource is part of Amazon's Well Architected framework. It focuses on operational excellence and contains information about deployment risk monument. It stresses the importance of testing and validating improvements before deploying them to production. Implied by this, MeCo should have a testing environment and production environment. This gives you a place to validate changes don't introduce new bugs or break features before they are used by real customers with real data. They also suggest to 'test potential improvements using the minimum viable representative components. Deploy tested sustainability improvements to production as they become available.' Systems should be in place to make keeping operating systems, libraries, and applications up to date on the latest stable versions. This ensures that your software is using the most efficient technology and has the latest security patches.

Metric Stream

URL: https://www.metricstream.com/products/operational-risk-management.htm

Summary: The MetricStream Operational Risk Management (ORM) software is designed to help organizations enhance risk visibility, minimize loss events, and increase business resilience during operational activities. Their software enables organizations to define business objectives, processes, products, risks, and controls, mapping their relationships in a centralized library. It supports both top-down and bottom-up risk and control assessments, allowing for simple risk ratings or advanced assessments (like impact or likelihood) to improve visibility into deployment-related risks. It offers tools to measure and monitor key risk indicators, set thresholds to prevent threats, and automate issue fixes. Features like AI-driven issue help address deployment failures quickly. It provides real-time dashboards and analytics for monitoring deployment risks, increasing visibility into deployment.

Steps Provided:

1. Define and map risks and controls within a centralized framework.
2. Plan and perform risk assessments.
3. Monitoring KRIs and automating remediation workflows.
4. Leveraging analytics for real-time risk visibility and decision-making.

Comprehensive Summary and Recommendation:

I wouldn't recommend just one resource as the topics covered in these resources were somewhat distinct from one another. I'd recommend both the devops.com article and the metric stream article. The devops article give a great overview of different deployment strategies to reduce operational risk, and other best practices for secure software deployment. Metric Stream gives a great overview of what should be considered when implementing systems to reduce operational deployment risks. I recommend MeCo review and consider all of the recommendations from each source, establish secure development best practices, do a risk assessment to map all of the risks, and establish a deployment automation system, like Metric Stream, that provides control, visibility and automation into deployments.

Sources

1. https://docs.aws.amazon.com/pdfs/wellarchitected/2022-03-31/framework/wellarchitected-framework-2022-03-31.pdf#operational-excellence
2. https://devops.com/software-deployment-security-risks-and-best-practices/
3. https://www.metricstream.com/products/operational-risk-management.htm