**Julio Pochet**  
CSD370 – Secure Software Development  
Module 11.2 Assignment – Operational Risk Management

**Introduction**

Operational Risk Management (ORM) is about planning ahead to reduce risks that might occur during software deployment. For Mesusa Corporation (MeCo), having a strong ORM approach can prevent issues like downtime, failed updates, or security vulnerabilities. ORM works by spotting risks early, evaluating their potential impact, and putting measures in place to prevent or minimize them. This paper reviews three current sources on ORM and software deployment, summarizing their key steps and best practices. It also compares their methods to decide which one MeCo should rely on for improving its deployment process.

**Source Summaries**

**Protecht – Strategies and Best Practices**  
Protecht (2025) describes ORM as a continuous and structured process designed to identify and control risks across systems, teams, and operations. It highlights the importance of embedding ORM into everyday activities and decision-making. Protecht suggests steps like risk assessments, creating controls, and constant monitoring. Its main strength is balancing high-level risk governance with practical tips that companies like MeCo can easily apply (Protecht Group, 2025).

**AuditBoard – ORM Process Overview**  
AuditBoard (n.d.) outlines a straightforward five-step ORM cycle: identify, assess, mitigate, implement controls, and monitor. Even though it is not specific to software, this framework is highly adaptable for deployment risks. AuditBoard recommends tools like risk matrices and scenario planning to visualize issues before they become major problems. The guide emphasizes proactive thinking and team collaboration, which is vital for managing deployments effectively (AuditBoard, n.d.).

**Configu – Software Deployment Guide**  
Configu (2025) focuses directly on software deployment risks. It provides hands-on advice like using CI/CD pipelines, testing in controlled environments, automating repetitive tasks, and preparing rollback plans in case something goes wrong. Configu’s guidance addresses common challenges, such as human errors and configuration drift, making it extremely practical for real-world software rollouts (Configu, 2025).

**Comparative Analysis**

All three sources agree that identifying risks early, planning controls, and monitoring systems are the keys to successful deployments. Protecht offers a governance-level view, AuditBoard delivers a clear ORM cycle, and Configu provides step-by-step deployment-specific practices. While Protecht and AuditBoard are strong in framework and strategy, Configu is the most actionable for day-to-day operations.

**Recommendation for MeCo:** Configu’s approach is best suited for MeCo because it speaks directly to deployment issues. Combining Configu’s technical tips with Protecht’s high-level risk strategy would give MeCo both structure and practical tools.

**Conclusion**

Operational Risk Management is crucial for reducing deployment problems and ensuring smooth, secure releases. Each reviewed source—Protecht, AuditBoard, and Configu—highlights similar best practices such as rollback planning, automation, and ongoing monitoring. Configu stands out as the most relevant for MeCo, offering real-world solutions that can be implemented immediately. Together with Protecht’s broader strategy, this approach would help MeCo build a solid ORM foundation.

**References**

AuditBoard. (n.d.). *Operational risk management: Overview and guide.* AuditBoard. <https://auditboard.com/blog/operational-risk-management>

Configu. (2025, May). *Software deployment: 2025 guide to process, strategies & tools.* Configu. <https://configu.com/blog/software-deployment-2025-guide-to-process-strategies-tools/>

Protecht Group. (2025, February). *A guide to operational risk management: Strategies and best practices.* Protecht Group. <https://www.protechtgroup.com/en-us/blog/comprehensive-guide-operational-risk-management>