Compliance: A Case Study Analysis

Kypton Lantz

Bellevue University

**CSD380-O307 DevOps (2261-DD)**

Prof. Darren Osier

October 12, 2025

Compliance: A Case Study Analysis

Organizations operating under heavy regulatory oversight often perceive DevOps transformations as risky, mainly due to the need for strict controls, audit trails, and separation of duties. Chapter 23 of *The DevOps Handbook (2nd ed.)* presents two case studies that challenge this assumption: “Providing Compliance in Regulated Environments” and “Relying on Production Telemetry for ATM Systems.” The first case examines how compliance can be integrated into agile and automated workflows, showing that DevOps automation and regulatory requirements can coexist. The second case illustrates the critical role of production telemetry in detecting system abuse or fraud that preventive controls alone might miss. Both cases emphasize the importance of evidence-based controls, continuous monitoring, and organizational alignment, providing practical insights for teams operating in regulated or mission-critical environments.

# Proving Compliance in Regulated Environments

The first case study focuses on reconciling DevOps automation with regulatory requirements. Traditional change management procedures, such as rigid change advisory boards (CABs), often dominate regulated environments. While these procedures are entrenched, the authors argue that DevOps teams should integrate with existing controls rather than bypass them. By classifying changes according to risk (standard, normal, and urgent), teams can automate low-risk, repeatable changes while retaining oversight for high-risk modifications.

Automation plays a central role in compliance. Machine-generated logs, audit-ready pipelines, and real-time dashboards provide reliable evidence for auditors, reducing reliance on manual documentation. The case also emphasizes that separation of duties and other control requirements must adapt to modern development practices. By demonstrating a strong track record of safe, automated deployments, organizations can justify reduced oversight on low-risk changes. Finally, cultural buy-in from compliance and security teams is critical; transparency and collaboration ensure that DevOps initiatives align with regulatory expectations rather than conflict with them.

From this case, the primary lesson is that compliance should not be viewed as an obstacle but as a design requirement for DevOps workflows. Incremental adoption, strong instrumentation, and stakeholder collaboration are key to transforming compliance processes without slowing delivery or reducing safety.

# Relying on Production Telemetry for ATM Systems

The second case study illustrates how preventive controls alone are insufficient for mission-critical systems. In an ATM network, a developer introduced unauthorized code that went undetected by standard code review processes. However, production telemetry, continuous monitoring of ATM operations, quickly revealed the anomaly, demonstrating the value of detective controls alongside preventive ones.

Telemetry is not only about performance monitoring but also about detecting policy violations, suspicious behavior, and unusual state transitions. Rapid feedback enables teams to respond to incidents quickly, reducing the window in which fraudulent activity can cause damage. The authors stress that telemetry should be designed with control and observability in mind, ensuring that developers, operators, and auditors have timely access to relevant metrics. This proactive approach emphasizes the importance of monitoring as a first-class component of system design, complementing traditional review and approval mechanisms.

The lessons from this case reinforce that organizations must assume that preventive controls can fail and that detection systems are critical for resilience. Telemetry should be comprehensive, accessible, and actionable, allowing teams to identify anomalies and respond effectively.

# Comparative Analysis and Implications

While the two case studies address different challenges (regulatory compliance versus fraud detection), they share common principles. Both highlight the balance between autonomy and control, emphasizing the importance of evidence-based practices and continuous monitoring. Preventive controls, such as change approvals or code reviews, are necessary but not sufficient. Systems must be designed with instrumentation and observability in mind to detect and respond to anomalies quickly.

Cultural alignment is another recurring theme. DevOps transformations succeed when compliance, security, and operations stakeholders are included in the process, and when transparency replaces rigid enforcement. Incremental adoption, starting with small, low-risk changes and expanding as confidence grows, is a recurring best practice in both contexts.

# Conclusion

These case studies demonstrate that DevOps and compliance need not be at odds. Organizations can operate efficiently by integrating automated, auditable processes with continuous monitoring while satisfying regulatory and security requirements. In regulated or mission-critical environments, the key takeaway is clear: assume that preventive controls may fail and design systems with robust detection, evidence-based oversight, and collaborative culture to maintain safety, resilience, and compliance.

References

Kim, G., Humble, J., DeBois, P., Willis, J., & Forsgren, N. (2021). *The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations* (2nd ed.). IT Revolution Press.