Noel Miranda

May 13, 2025

DevOps

Module 12

# Overview of Case Studies in Chapter 23 from *The DevOps Handbook*

DevOps was originally created to bridge the gap between software development and IT operations. The goal was to help teams build, test, and release software faster and more reliably. According to *The DevOps Handbook* (Kim et al., 2021), over time this practice has evolved, revealing an important shift. DevOps is no longer just about developers and operations teams working better together. It now includes auditors, compliance officers, and others across the value stream. Chapter 23 of the book explains why involving these groups early in the process is not only helpful but necessary, especially when working in regulated industries. While the book also talks about bringing in security and other stakeholders, this paper will focus mainly on the lessons learned from the two case studies found in chapter 23, highlighting the importance of telemetry as well as the inclusion of auditors and compliance officers in the DevOps loop.

The first case study, "Proving Compliance in Regulated Environments (2015)," focuses on how teams in regulated industries can still meet compliance requirements while using modern cloud technologies. According to Kim et al. (2021), Bill Shinn from Amazon Web Services noticed that many traditional audit methods no longer fit how systems operate today. Instead of relying on outdated practices like screenshots, his team worked with auditors early on to identify the controls they needed and built those checks into their development process. By using real-time monitoring tools, they made it easier for auditors to verify compliance, which helped reduce delays and improve trust. This case study helped me realize that following rules does not have to slow down innovation. In fact, if the right tools are in place, it can even speed things up. I also

learned that compliance should not be treated as something separate from the development process. It should be part of it from the beginning. That way, one is not trying to prove safety after the fact but rather building it in from the start.

The second case study, “Relying on Production Telemetry for ATM Systems (2013),” shows how real-time monitoring, or telemetry, can be more effective than traditional security checks. In this example, a bank discovered that a developer had inserted a backdoor into ATM code, allowing unauthorized access to cash. Despite following standard processes like code reviews and separating development from operations, the fraud went unnoticed until someone spotted unusual ATM behavior during a routine meeting. According to Kim et al. (2021), it was the unexpected maintenance activity that raised red flags, not the formal checks. This case highlights how monitoring what actually happens in production can reveal issues faster than relying only on reviews or paperwork. It shows that no matter how careful a company is, it still needs real-time visibility into its systems because one cannot catch everything with just code reviews. Furthermore, sometimes problems do not show up until the software is running live. Therefore, telemetry gives teams a way to see those problems early and take action before things get worse.

Taking both case studies into consideration, what stood out to me the most is that following traditional rules and performing reviews is not enough on its own. The real impact comes from building systems that can be carefully monitored and clearly understood while they are running. Modern tools give us ways to detect problems faster and more clearly. I also learned that working closely with auditors and regulators is not a weakness. It is actually a forward thinking move because it creates a common understanding and helps prevent confusion later. According to Kim et al. (2021), building documentation, collecting useful evidence, and involving auditors early can all help reduce risks and avoid delays. This makes compliance feel

less like a burden and more like part of the development culture. That is something I had not considered before reading this book. Before this, I thought security and audits were always a last step check, as prior experiences have shown me that they are often bottlenecks. Now I see that with the right practices, they can actually support speed and safety at the same time.

In conclusion, the main message from these case studies is clear. Fast delivery and strong security do not have to compete. With the help of real-time monitoring, smart automation, and early collaboration with auditors, teams can meet their goals and still protect their systems. These lessons are not just about code or tools. They are about people working together, asking the right questions, and making better decisions based on what is really happening. That is what DevOps is all about.

# References

Kim, G., Humble, J., Debois, P., Willis, J., & Forsgren, N. (2021). *The DevOps Handbook, Second Edition*. IT Revolution.