**Brian Preston**

**CSD 380**

**July 27, 2025**

**Case Studies Reflection**

In this final assignment, I am reflecting on two impactful case studies from *The DevOps Handbook* (Kim et al., 2016), both found in Chapter 23. The first, "Proving Compliance in Regulated Environments", dives into the challenges of maintaining compliance in DevOps workflows. The second, "Relying on Production Telemetry for ATM Systems", highlights how real-time data and monitoring can detect and prevent fraud faster than traditional review methods. Both stories emphasize that DevOps is not just about speed but also visibility, accountability, and continuous improvement. As someone who works in a regulated retail environment, I found both case studies surprisingly relatable.

**Proving Compliance in Regulated Environments**

The first case study centers around Bill Shinn, a principal security solutions architect at Amazon Web Services. Shinn's work involved helping enterprise customers demonstrate compliance with strict regulatory standards, even while operating in a dynamic DevOps setting. I think this case really highlights how traditional audit expectations often clash with modern, automated infrastructure. Shinn points out, *“In audit fieldwork, the most commonplace methods of gathering evidence are still screenshots and CSV files filled with configuration settings and logs”* (Kim et al., 2016, p. 456). But in DevOps environments—where servers scale automatically and infrastructure is defined as code—those old methods quickly become outdated.

I feel the most powerful part of this story is how they tackled compliance proactively. Instead of scrambling to gather evidence after the fact, they integrated audit controls directly into the pipeline and used telemetry systems like Kibana to make compliance data self-service. I believe this forward-thinking approach turns compliance from a bottleneck into a strength. It reminds me of how in my job at Estée Lauder, we often have to maintain tight compliance when launching new promotions or limited editions. Our tracking systems for client outreach and sales events have to be accurate and auditable, especially when we are accountable for region-wide performance. While it is not the same as HIPAA or SOX compliance, the need for trustworthy data and visibility is very much the same.

**Relying on Production Telemetry for ATM Systems**

The second case study features “Mary Smith,” who leads a DevOps initiative in consumer banking. In this story, a developer had secretly inserted a backdoor into ATM software, allowing them to steal cash. Surprisingly, the fraud was not caught through code reviews or testing—it was caught by reviewing telemetry. Smith observed, *“We found the fraud even before the scheduled cash audit process, when they reconcile the amount of cash in the ATMs with authorized transactions”* (Kim et al., 2016, p. 457).

I think it is eye-opening to realize that even with all the right controls—separation of duties, change approval processes—things can still slip through. What matters is how quickly you can detect and respond. I feel telemetry is like always having eyes on your system. In my role, I have seen how data-driven insights help catch irregularities early—like tracking odd sales patterns that might suggest inventory issues, mispriced items, or duplicate client outreach. I believe relying on real-time data gives teams a much better chance of reacting before minor problems turn into major ones.

Taking this class has shifted the way I think about development, testing, and even collaboration. I have learned that DevOps is not just about deploying faster—it’s about embedding quality, security, and compliance into every step of the process. These two case studies perfectly summarize what we’ve explored throughout the course: visibility matters, automation matters, and shared responsibility matters. I think what stood out to me the most is the importance of treating security and compliance as a team effort, not as isolated responsibilities. Just like how my team at Estée Lauder succeeds when everyone is aligned—sales, operations, marketing—I believe DevOps works best when everyone is pulling in the same direction. This class gave me a new appreciation for how interconnected systems are and how vital it is to design for reliability, not just performance.

**References**

Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps handbook: How to create world-class agility, reliability, and security in technology organizations* (Chapter 23, pp. 456–457). IT Revolution.