Rachel Theis

Module 2 Assignment

CSD 380

6.6.25

**Chapter 6 Case Study: Summary**

The case study in Chapter 6 of the DevOps Handbook explores the Operation InVersion that occurred at LinkedIn in 2011 to catch up on extensive technical debt that was hindering the platform’s scalability and overall viability (Kim et al., 2015). The first decade of LinkedIn’s development was built upon a homegrown Leo framework that was unsuitable for the exponential growth and modern technical demands. To adapt the application to its growing user base, core functions were decoupled from Leo, ultimately leading to over one hundred services being handled outside of Leo (Kim et al., 2015). The result of this old framework and partial decoupling was problematic and painful deployments. In 2011, a senior engineer at LinkedIn, Kevin Scott, launched operation InVersion, an initiative that paused all work on new features and assigned the entire team to fixing and modernizing the site’s core infrastructure, lamenting this as a cultural and business need and top priority (Kim et al., 2015). “Leo was often going down in production, it was difficult to troubleshoot and recover, and difficult to release new code… it was clear we needed to ‘kill’ Leo and break it up into many small functional stateless services.”, said Scott (Kim et al., 2015). This project took two months to complete and was very successful, allowing LinkedIn to effectively scale its framework in response to company growth and innovate with new features that aren’t held back by old, broken code (Kim et al., 2015).

**Chapter 6 Case Study: Lessons**

**Lesson 1: Stay on top of technical debt and mitigate early**

The biggest lesson I took from this case study was the importance of paying technical debt as it arises. Our textbook encourages teams to reserve 20% of cycles for non-functional requirements and to reduce technical debt (Kim et al., 2015). When organizations fail to reduce technical debts incrementally, and as it arises, the technical debt can have a snowball effect and become unmanageable, making the software unusable. Of course, LinkedIn likely did not predict how large their user base would become and how quickly they would grow; however, if they had addressed their problems with scalability and incompatible code, Project Inversion would not have been necessary, and scaling would have been a much smoother process. Here, we can see the value of incorporating paying technical debt into daily routines to avoid future complex and resource-intensive fixes.

**Lesson 2: Build applications with scalability in mind**

The case study mentions that LinkedIn went from 2700 to over one million users in its second year of operation (Kim et al., 2015). This rapid growth demands extensive scalability in an application’s framework. This level of growth would be challenging to manage in the best of circumstances, so building scalable applications is crucial for being set up for success when growth happens, especially when growth happens rapidly.

**Lesson 3: The power of a shared goal**

This lesson is also mentioned in Chapter 6 of our textbook, which highlights the value of setting measurable and challenging yet achievable goals (Kim et al., 2015). One thing that stuck out to me in our case study was the “all in” attitude toward Project InVision. The senior leaders instilled a culture of importance and urgency surrounding Project InVision and set a clear goal: fix the existing architecture.

**Lesson 4: The importance of a solid foundation**

Here, we can see that, despite having a massive user base and becoming a publicly traded corporation, LinkedIn was struggling to meet the needs of its users and scale and grow the technical side of its business due to the state of its technical architecture. Companies should prioritize and invest in the fundamentals of their technical system so they can seamlessly integrate new features without sacrificing the application’s existing performance and reliability.

**References**

Kim, G., Debois, P., Willis, J., Humble, J., & Allspaw, J. (2015). *The Devops Handbook How to Create World-class Agility, Reliability, and Security in Technology Organizations.* It Revolution Pr.