Brittni Ray

CSD 380

Module 2.2

**Case Study Summary - Chapter Six of The DevOps Handbook**

This case study focuses on technical debt as it is related to the company we all know today as LinkedIn. It states that, in 2011, LinkedIn initiated “Operation InVersion,” which was a project that halted all feature development for a two-month time span in order to address the deeprooted technical debt that the company had garnered by heavily relying on a home-grown application called Leo. This application had caused them to face severe deployment challenges that stemmed from Leo’s monolithic architecture. As LinkedIn’s popularity grew, thereby increasing the user base and site traffic, its infrastructure suffered greatly, which led to frequent late-night emergencies for its software engineers. Seeing no end to this technical debt in sight, the senior engineer leaders presented “Operation InVersion” as a way to overhaul LinkedIn’s architecture and deployment processes.

“Operation InVersion,” specifically, aimed to deconstruct the monolithic Leo system into smaller functional services, automate deployment pipelines, and improve developer productivity by spending less time on fixing the outgrown Leo system. Though all feature development had been halted, their successful efforts created a robust and scalable infrastructure allowing LinkedIn to release updates multiple times daily. Although this had been a risky move, it paid off—giving developers time to innovate instead of focusing on constantly fixing an issue related to severe technical debt.

‘

**Lessons Learned**

1. Prioritizing technical debt can allow long-term stability and scalability.
2. Focusing on infrastructure instead of new features can be critical to a company/application's success, especially in periods of rapid growth.
3. A strong leader who is willing to take calculated risks is vital to making large organizational/architectural changes.
4. Regularly addressing technical debt as a part of daily work will help avoid severe disruptions.