Brooks Roberts

01/19/25

2.2 Assignment

**Summary of the Case Study: Operation InVersion at LinkedIn (2011)**

Operation InVersion was LinkedIn's plan in 2011 to take on its technical debt and operational issues. After a successful IPO, LinkedIn had many deployment and performance issues because of its Leo java application. Even though it had small improvements, Leo had slow development cycles, scalability issues, and operational problems. This led to LinkedIn pausing all feature development for two months to focus on fixing all of its issues.

By 2010 LinkedIn introduced new services outside of Leo, but they deployed only biweekly and had many failures. Kevin Scott, LinkedIn's VP of Engineering, stopped all new feature development to focus on updating their engineering processes. This would involve creating new tools and services. It also would automate testing to improve the code and the speed of deployments.

Operation Inversion's ultimate goal was to create an atmosphere that reduced technical debt and gave stability to the system. This would lead to lasting innovation and growth. However, it was a risk since this plan would delay any new features during a critical time following LinkedIn's IPO. In the end, it resulted in a shift from biweekly deployments to daily, automated, bug-free releases. There was growth in developers' productivity and agility that helped grow its microservices from 150 in 2010 to over 750 by 2015. Additionally, there were fewer late-night emergencies that allowed engineers to focus on innovation.

**Lessons Learned:**

* Addressing technical debt proactively prevents costly, high-pressure overhauls in the future.
* Aligning engineering efforts with business needs ensures long-term stability and growth.
* A strong engineering culture values non-functional requirements like tooling and infrastructure as much as new features.