Jeremiah Kellam

CSD380 Module 2.2 Assignment

6/7/2024

Case Study: Operation InVersion at LinkedIn (2011)

LinkedIn was launched in 2003 and went public in 2011. The rapid growth of the user base and the efforts of the developers to launch new content created a massive amount of Technical Debt where things were failing to work with its system based on Leo, a dated Java application. Kevin Scott joined as VP of Engineering right before the IPO and shortly after had to stop all development and launch Project InVersion in an effort to resolve LinkedIn’s repeated site crashes and inoperable feature additions.

The technical debt accrued with LinkedIn’s rapid growth is literally this textbook’s example of what additional rework cause by easy solutions almost cost the company. The debt was due to codebase complexities that didn’t allow additional features to be integrated smoothly, late night efforts to fix crashes, and time crunches for deadlines.

The effort of Operation InVersion allowed the company do make it modular, scalable, and modernized to current coding standards at the time. They refactored code, overhaul the architecture of the design to be service-oriented, created their own automation and suite of software to work with CI/CD practices, and relieved much of the stress on team members that needed to be heroes for the service to work.

Takeaways are to be proactive with recognizing what will create technical debt and prioritize what is necessary and correctly identify impact for any service. Modular design is needed to keep it simple, and everything needs to be scalable to allow future growth without integration issues. Automated testing is necessary for rapid releases. Code reviews and cross-focus education will allow collaboration between subject matter experts. Always consider end-user impact to determine if it is value added or if something else takes precedence.

LinkedIn was able to turn it around and is still successful today due to the efforts they put in during Operation InVersion to pay off their Technical Debt and revamp their service to be what they intended using the tools that would allow it to continue to grow.

References:

Kim, G., Debois, P., Willis, J., & Humble, J. (2015). The DevOps Handbook: How to Create World-Class Speed, Reliability, and Security in Technology Organizations. It Revolution Press.