DeJanae Faison

M2: Case Study

Due 8.24.25

Summary: Case Study-Operation InVersion at LinkedIn

Back in 2003, LinkedIn was launched as it created a platform for users to network and connect for better job opportunities. It was a rapid success as by the end of its first week they had 2,700 members and after a year had over 1 million members. With such fast growth, it created high demand for their back-end system. In its early start, LinkedIn’s architecture ran on their own Leo application. Leo was a monolithic Java app that handled every single one of LinkedIn’s pages and managed JDBC connections to multiple Oracle databases. With its rapid growth, two services were uncoupled from Leo: member connection graph and member search. By 2010, though hundreds of services were run outside of Leo, their homemade application was a massive problem. Even when adding memory and more CPUs, it was highly unstable and difficult to fix. The site would become a mess if too much was added at once and a lot of time was poured into fixing the mess that was made. When it can no longer be tolerated, some of their top engineers decided to stop producing new features, recalibrate the site’s core infrastructure. In 2013 Operation InVersion was launched where no new features would be developed until LinkedIn’s computing architecture was revamped. The outcome? Newly automated systems allowed for safer, and faster deployments as well as stability and massive scaling. They grew from having 150 separate services to more than 750.

From this case study, one take away is that technical debt (or implied cost of additional work in the future resulting from choosing a fast solution over a robust one) should be addressed immediately and often. This would save many late nights of debugging like the developers at LinkedIn had to go through. What may not seem as important later may be major in the future. For example, they did not prioritize scalability since they did not expect for the site to grow at such a huge and rapid pace. Due to this Leo was unstable. Another take away is that technical debt will eventually have ‘painful’ tradeoffs. One of the developers explained that when implementing Operation InVersion, it was scary since they had to put this new popular successful site on pause.

Resources:

Kim, G., Debois, P., Willis, J., Jez Humble, & Allspaw, J. (2021). The DevOps handbook: how to create world-class agility, reliability, & security in technology organizations. It Revolution Press, Llc.