

Jonah Aney 11/02/25 CSD380 Module-2.2 Assignment: Case Study: Operation InVersion at LinkedIn (2011)

In this assignment, we'll cover the case study of Operation inVersion at LinkedIn. I'll discuss the main points as well as the lessons learned through this operation.

Summary:

LinkedIn, the online professional networking platform we know today, went public in 2011 and had accumulated a large amount of technical debt since launching in 2003. It eventually got so bad, all feature development work was stopped for two months to focus on overhauling the computer environments, deployments, and architecture of LinkedIn.

LinkedIn initially ran on Leo, a monolithic Java app that was having problems by 2010. Leo was going down in production despite vertically scaling with memory and CPUs and it was difficult to recover and release new code. Another issue was that Leo was only being deployed every two weeks. It was decided to break Leo up into many functional stateless services for better scalability and resilience. ...

LinkedIn grew at a substantial rate. After their first year, the company had grown to over 1 million members and had over 350 million by 2015. With this massive growth came massive problems. When LinkedIn tried to add features, the site would break, causing engineers to work late to fix the issues. In the fall of 2011, the late-night work required to manage Leo was becoming too much to tolerate. Kevin Scott, VP of engineering, decided to stop all new feature work and focus on fixing the site infrastructure. He called it Operation InVersion. Not only was this needed for the business to become stable, but it also introduced a new way of thinking and culture within the company's engineering team. It was a risky decision as now they could no longer promise their customers any new features for the next two months during a time of extreme growth and popularity.

The case study explains how LinkedIn created an entire suite of tools and software to help with code and building a new site. Engineers were able to develop a new service, run automated testing to check for bugs and how it interacted with existing features, and add it to the main site. This work and added value enabled updates to happen more often and fewer late-night fixes. More time for developing features for the customers.

Lessons Learned:

Operation Inversion allowed engineering to focus on improving tooling and deployment, infrastructure, and productivity. This effort enabled the teams to have the agility needed to build and scale their products. Its success is evident as LinkedIn went from 150 separate services in 2010, to over 750 today. When I read this case study it made me

Jonah Aney 11/02/25 CSD380 Module-2.2 Assignment: Case Study: Operation InVersion at LinkedIn (2011)

think of the quote, “Sacrifice what you are for what you’ll become.” to me it means take a step back, accept goals won’t be met, and product wont be finished BECASUE we need to focus on the source of the issues and finally resolve it to better ourselves and move past it. This operation shows that when you pay down the debt and establish stability, teams are better prepared for future work and efficient growth. Find and fix problems to manage technical debt and avoid really bad days, weeks, or months. When these issues are proactively dealt with, the burden of daily workarounds is lifted, and teams can focus on delivering new features to customers.

Sources:

Kim, G., Humble, J., Debois, P., Willis, J., & Forsgren, N. (2021). *The DevOps handbook: How to create world-class agility, reliability, & security in technology organizations* (2nd ed.). IT Revolution Press.