

Noel Miranda

March 25, 2025

Module 2

Assignment 2

Case Study: Operation InVersion at LinkedIn (2011)

Summary

LinkedIn is the reason I got my internship this summer, and I use it frequently to connect with software developers for better career opportunities. I would have never thought the platform had such a complicated past. According to Kim et al. (2021), after LinkedIn's IPO in 2011, they faced serious infrastructure challenges, specifically because of their main application named Leo. Leo had become a bottleneck, slowing down deployments and making it difficult to release updates. Engineers struggled to troubleshoot issues, which eventually led to them working around them. It became the new norm, but the consequences included an unreliable and inefficient system that seemed to be a constant headache for both the developers and users.

To fix this, LinkedIn launched Operation InVersion, where they paused all new feature development for two months to focus entirely on fixing their core systems. As a current investor in IPOs, this news would have made me concerned, but in the end, it was a risk with a high return on investment. I wish I had been able to invest in LinkedIn during such a time, as it has clearly improved drastically 14 years later. According to Kim et al. (2021), this decision was necessary to address LinkedIn's growing technical debt, which had been accumulating from years of workarounds and outdated infrastructure. If they had not fixed it, the company would have continued to struggle or even collapsed.

The results proved the decision was worth it. Before the transformation, LinkedIn was deploying changes about once every two weeks. After Operation InVersion, they improved their

engineering process so much that major updates could be deployed three times a day. According to Kim et al. (2021), this shift allowed LinkedIn to grow from 150 separate services in 2010 to over 750 by 2015, making the platform more scalable and efficient.

Lessons Learned

This case study makes one thing clear, working around problems is never a good long-term solution. LinkedIn's experience shows the dangers of technical debt and the importance of addressing it before it spirals out of control. Kim et al. (2021) emphasize that companies must dedicate time to maintaining and improving their systems, at least 20% of work hours, instead of always prioritizing new features. LinkedIn, at the time of accumulating technical debt, is a counterexample of what *The DevOps Handbook* recommends. After realizing the root cause of their challenges, LinkedIn had to pause new development for two months just to catch up, proving how crucial it is to stay on top of technical debt before it becomes overwhelming.

Another key lesson is the necessity of constant innovation, especially when scaling up. A system that works for a small company may not be sustainable when millions of users rely on it. LinkedIn recognized that their old infrastructure could not keep up, and instead of patching problems, they made the bold decision to rebuild. According to Kim et al. (2021), this move set them up for long-term success and gave them the agility needed to grow rapidly.

This case study also highlights the power of visualizing an organization's value stream workflow, collaboration, and automation in modern software development. In other words, it demonstrates some of the crucial principles of DevOps in action. As a result, LinkedIn's slow and error-prone processes transformed into fast, reliable, and efficient ones.

References

Kim, G., Humble, J., Debois, P., Willis, J., & Forsgren, N. (2021). *The DevOps Handbook, Second Edition*. IT Revolution.