Operation InVersion at LinkedIn (2011)

Jackie Scott Module 2.2 August 19,2024

Operation InVersion was launched in 2011 by Kevin Scott, the VP of engineering at the time. The purpose of Operation Inversion was “a way to ‘inject the beginnings of a cultural manifesto into his team’s engineering culture.” The idea was that no new developments would be planned until LinkedIn revamped its entire computing architecture.

LinkedIn experienced rapid growth in end users in the first year. The organization reached well over one million members. In the beginning, LinkedIn ran a Java application they created known as Leo. As the organization and its members grew, the Leo application became more problematic. LinkedIn members were generating tens of thousands of requests per second. This would result in millions of queries per second on the organization's back end—a stress Leo was unprepared to handle. Even after LinkedIn decoupled the two main services from their Leo application, issues were increasing. These issues would cause frequent downtimes and difficulties in troubleshooting and deploying any code features or fixes.

Finally, in 2011, LinkedIn decided that its best course of action was Operation InVersion. This operation would essentially halt any new feature development, so their main focus could be on fixing the infrastructure issues. The operation would also help the organization address any technical debt that had been accumulated over the course of trying to mend the Leo application. In the two-month span during which Operation InVersion took place, LinkedIn was able to develop new software and tools that assisted in streamlining code development and deployment. LinkedIn engineers were now able to deploy new services live up to three times daily, something that had never been possible with Leo. The overhaul that Operation InVersion did help in a decline of emergency fixes.

The author of this textbook really puts importance on the point of technical debt management. Before LinkedIn implemented Operation InVersion, they essentially hindered company growth by implementing quick fixes and poor code quality. In other words, racking up major tech debt. The author makes a point to mention that a lesson learned from this specific case study is that while it did take two months of focus, in the end, LinkedIn created a “stable and safe environment” while also lifting the “burdens of daily workarounds….and the team was able to once again focus on delivering new features to delight their customers.”

In conclusion, the case study of LinkedIn’s Operational InVersion demonstrates how managing technical debt and focusing on system stability enhances productivity and increase the system’s performance. This operation was key to LinkedIn’s long-term success, and it is still ongoing today.