## Brian Preston

## CSD-380

## Module 3

## Case Study :Operation InVersion at LinkedIn (2011)

## When reading Chapter 6, I learned that after LinkedIn’s IPO in 2011, the company faced serious deployment issues due to mounting technical debt. Their primary application, a massive Java-based monolith called Leo, was unstable, hard to troubleshoot, and slow to release updates. This led to the launch of Operation InVersion, where LinkedIn halted all new feature development for two months to focus solely on infrastructure improvements.

## This initiative allowed LinkedIn to break apart the Leo application into smaller, stateless services and create a safer, more reliable deployment process. As a result, engineers could deploy updates several times a day rather than once every two weeks. The project ultimately laid the groundwork for a more scalable and efficient development process.

## “LinkedIn created a whole suite of software and tools to help it develop code for the site…LinkedIn’s engineering corps [now] performs major upgrades to the site three times a day.” (Kim et al., 2016, p. 71)

## Lessons I learned first were to Pay Down Technical Debt Regularly: Ignoring it leads to severe instability. Fixing the root causes, even at the cost of short-term feature delays, improves long-term agility and reliability. Also, Creating Safe Systems of Work: When deployments are easy and safe, engineers can focus on innovation and prevent burnout.

## When reflecting on this, I feel you sometimes need to stop and move forward. Personally, as scary as it may be, a complete pause to fix internal problems can result in major performance boosts and fewer disruptions later on.

## “By allowing LinkedIn to pay down nearly a decade of technical debt, Project InVersion enabled stability and safety, while setting the next stage of growth for the company.” (Kim et al., 2016, p. 72)

## I relate to this story in my own life. Sometimes, I push forward even when my routine, energy, or workflow are completely out of sync. But everything runs smoother when I take the time to step back, organize, and improve my workflow. Just like LinkedIn paused to fix its infrastructure, I’ve found that pausing to fix my “personal systems” helps me move forward with greater efficiency and less stress.

## References

Kim, G., Humble, J., Debois, P., & Willis, J. (2016). \*The DevOps handbook: How to create world-class agility, reliability, and security in technology organizations\*. IT Revolution.