Candice Garcia
Module 2.2

This case study provided by LinkedIn was a great way to visualize how to handle the accumulation of technical debt within software. The case study was coined as Operation InVersion, which took place in 2011. When LinkedIn originally started, just like many companies, the level of online traffic the website might experience was still being determined. Luckily for LinkedIn, their company was doing well and quickly growing. However, this added stress to the company as they were probably unprepared for this many software users. This visualization helps me understand how they kept putting a Band-Aid on the situation and worked tirelessly to keep up with their database, which initially started as a monolithic system where everything was handled in the same place.

As the company grew, they kept trying to fix issues that came up and were working tirelessly to do so. Because of this, they decided to halt all new production, which was a great decision on their part because, in the end, they couldn't be sure if their system would crash or just cause a temporary inconvenience to users wanting specific updates. Overall, this company could've gone downhill quickly if it had yet to prove it could keep up with its offerings. The lesson learned from this is that LinkedIn took the time to fix the issues instead of temporarily fixing the situation, which would have needed to be more sustainable in the long term.

They also moved to a modular design, which allowed different areas to specifically handle different services so that an issue in one location wouldn't affect the whole system. That was huge because they had such a significant presence online. They also focused on what would be stable long-term rather than short-term, so it made sense that certain features were put off. Overall, the foundation needed to be strong to secure their future. As a result, engineers at LinkedIn were able to stop constantly restructuring and maintaining the system, which allowed it to become faster and more reliable for the products they were developing. Overall, it is an excellent example of how they saw the issue, fixed the foundational pieces before it was too late, and prioritized their company's long-term success over short-term solutions.

Source

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