Cody Ferre

CSD380-O302

08/25/2024

For the case study of LinkedIn’s 2011 Operation InVersion was an operation that showcased the lesson of repaying technical debt. This was created due to LinkedIn’s own core system software Leo which would be updated every two weeks causing services for the website to go down. It's from this happening over a course of time that Operation InVersion came into play, this being the initiative to halt all of development and completely rebuild their framework, and systems. The main point of this would be that while they had continued to upgrade their system Leo over the years, too much was resolving a few issues and instead adding improvement building up a bunch of technical debt. By choosing to pay off the technical debt that had accrued, this allowed LinkedIn to grow further as a company and continue expanding, starting a new set of technical debt.

LinkedIn chose to pay this debt by killing their current software system Leo and taking it down into smaller services. This would be accomplished by having develop focus on creating separate systems that they can upgrade whenever they would like. The lesson to take away from all the work they pulled to do was that if proper scaling and scalability are taken new products can be increased and grown over time. (p. 125, Kim, Humble, Debois, Willis) Some other lessons that can be taken away from this is going from 150 separate services to 750 after the operation shows how many of the systems can be broken down into smaller components to have uncomplicated services that can be developed further and independently. Giving a good example of how we can change the scale of our designs and programs to create ease of use for not only us but others as programmers to also help reduce the debt that may arise from the problems also being simpler to resolve.

Citation

Gene Kim, Jez Humble, Patrick Debois, John Willis, (2016), The DevOps Handbook, https://www.google.com/books/edition/The\_DevOps\_Handbook/ui8hDgAAQBAJ?hl=en&gbpv=0