Name: Ivan Pryymak

Date: 8/25/2024

Assignment 2.2 – Case Study: Operation InVersion at LinkedIn (2011)

This LinkedIn case study highlights the importance of addressing technical debt in a timely manner in order to keep the organization running the business normally, rather than suffering from decisions made in the past.

Technical debt can be defined as a cost of the bad programming and architectural decisions made in the past. Whether such decisions are made to meet the budget, the deadlines, or just for simplicity – technical debt will eventually catch up and will have to be paid.

LinkedIn case study shows how they had to deal with their technical debt that accumulated over a decade of bad implementational decisions. LinkedIn primarily ran on their application called Leo, which was an outdated Java application. As the number of customers started to grow (from 2700 members in the first week, to 350 million members 12 years later), Leo couldn’t handle the volume and would constantly crash in production. Realizing that if that process kept going the way it was, the company would not survive the overwhelming demand from the market, therefore, LinkedIn decided that a change is needed. Kevin Scott, one of the top engineers at LinkedIn, launched Operation InVersion, which basically constituted a halt to any new feature development for two months in order to have enough time to address the technical debt that accumulated over the years. The goal of the operation was to improve the productivity of the development team, allowing them to focus on new features, instead of finding was to work around old issues. As a result, it allowed LinkedIn to scale the product much more efficiently and deploy new features up to three times per day, instead of once every two weeks. Operation InVersion not only improved the reliability of the system, but also created a culture of continuous improvement.

The main takeaways from the case study:

* Paying Down Technical Debt – delaying this might lead to very costly consequences, therefore it should be addressed continuously.
* Strategic halt in development – instead of releasing new features on top of underlying architectural issues, fixing current issues might yield better results.
* Continuous improvement – culture of continuous improvement enhances company’s agility and ability to grow.
* Business goals – instead of working on new features, sometimes it’s important to take a step back and look at the big picture. This will allow us to align the goals of development team with the business goals of the company.