Jason Schriner

Module 2

6/7/25

In 2011 LinkedIn’s technical debt had become a serious problem. It was slowing down deployment of new features as developers had to create work around, and it was creating stability issues. To combat the issues that were occurring Kevin Scott, the VP of Engineering, decided that they should stop production of any new features and focus solely on rebuilding the infrastructure of the platform.

One of the largest issues the company was having was with Leo. Leo was a “monolithic Java application that served every page through serverlets and managed JDBC connections” (Kim et al., 2017). The drawback to this type of application is that it can make deployments riskier. Any small change means a whole redeployment of the entire app. This can also make it difficult to work on scaling an individual component due to its tightly coupled nature. To fix their problems, they shifted to smaller stateless services. This allowed them to decouple their infrastructure so that it was possible to make changes to a feature without it having the potential of causing an issue somewhere else in the code base. I also allowed the team to work independently on a single service to allow for faster deployment cycles. This rework of the system also allowed engineers to be more productive as they no longer had to try and create work arounds for existing problems.

LinkedIn took a major risk of stopping innovation to pay off their technical debt. This ended up working in their favor as they now have a more flexible platform that allows them to innovate faster.

LinkedIn took a radical approach to pay down their technical debt. While it is important, it is not always necessary to be this aggressive. In “The DevOPs Handbooks” (Kim et al., 2017) it explains that you could utilize twenty percent of your engineering capacity to this instead of performing a shutdown like they did. This way you can still offer new features while working on refactoring.

References:

Gene Kim (November, 2023) Case Study: LinkedIn’s 2011 Operation InVersion through lens of Slowify, Simplify, and Amplify. LinkedIn <https://www.linkedin.com/pulse/case-study-linkedins-2011-operation-inversion-through-gene-kim-dht2c/>

Kim, G., Debois, P., Willis, J., Humble, J., & Allspaw, J. (2017). The DevOps handbook : how to create world-class agility, reliability, and security in technology organizations. It Revolution Press, Llc.