During the early years of LinkedIn in 2011, Its development was plagued by problems that often had developers working all nigh to correct one issue. This all happened 6 months after their first successful IPO. So bad in fact they had to stop all new production and start working to improved “their computer environments, deployments, and architecture.” In an operation they launched called InVersion.

LinkedIn actually ran on their own software called Leo application which was a massive Java application. It managed to serve each “page through servlets and managed JDBC connections to various back-end Oracle databases.” (Kim et al., 2016) Leo Application had two major components of their application. The first was a way to handle queries around the member connection, which was handled by memory. And the Second was a member search which happened to be layered over the first major component. (Kim et al., 2016) But the problem was that their application was difficult to troubleshoot.

Even though the company was scaling the application through the addition of memory and CPU, it kept having issues with production. “It was difficult to troubleshoot and recover, and difficult to release new code…” (Kim et al., 2016) Developers would often have to work all night on their application to fix it which is going to need to be changed. The VP wanted to “kill leo and break it up into many small functional and stateless services.” (Kim et al., 2016) That’s when the decision was made to stop all production until their application was fixed.

Kevin Scott, the VP of engineering made the decision to halt production and start fixing all their problems, this would last about 2 months. But by doing so they managed to create whole new software rather than the alternative to creating a new function and it takes weeks to deploy.

In conclusion, sometimes It may be best to just start new rather than dealing with a bunch of bugs. Making an application that is easier to troubleshoot, improving the tools they use is a lot better than dealing with the issues and taking weeks to deploy new features all while trying to keep the website from going down which could lead to tons of lost revenue.

Reference:

*Angular*. (n.d.-c). angular.io. Retrieved May 29, 2023, from https://angular.io/guide/router