James Brown III

Module 2.2 Assignment

January 25, 2026

This case study explains how LinkedIn faced serious technical problems after its 2011 IPO due to an unstable system and slow, risky deployments caused by a large monolithic application called Leo. As traffic and user demand grew, the platform became harder to manage, frequently crashed and forced engineers into late night emergency fixes instead of productive development. To solve this, their leadership made the bold decision to pause all new feature development for two months and focus entirely on fixing infrastructure and paying down technical debt through an initiative called Operation InVersion. During this time, LinkedIn improved its deployment tools, automated testing processes, and broke the monolithic system into many smaller, stateless services that could scale more effectively. As a result, deployment frequency increased from once every two weeks to multiple times per day, system stability improved, and engineers became more productive and less stressed. The main lesson learned from this is that ignoring technical debt eventually slows innovation and damages reliability, while investing in core infrastructure creates long term business value. Although the short term pause in feature development was risky and uncomfortable, it enabled LinkedIn to build a stronger, more scalable platform, reduce operational failures and support faster innovation in the future. As a result, LinkedIn accepted short term pain to achieve long term stability, speed and growth.

References:

Grammarly, Inc. (2026). Grammarly [Writing assistance software]. https://www.grammarly.com

Kim, G., Humble, J., Debois, P., & Willis, J. (2021). The DevOps handbook: How to create world-class agility, reliability, & security in technology organizations (2nd ed.). IT Revolution Press.

OpenAI. (2025). ChatGPT (40 mimi). https://chat.openai.com/chat