**Operation InVersion at LinkedIn**

The 2011 LinkedIn case study "Operation InVersion," highlighted in *The DevOps Handbook*, focuses on the challenges and strategies LinkedIn adopted to overhaul its system architecture and improve operational efficiency. Following LinkedIn’s IPO, technical debt and deployment issues had accumulated to the point of severely hampering productivity. In response, LinkedIn paused feature development for two months to concentrate solely on restructuring its infrastructure. This effort was led by Kevin Scott, then LinkedIn's VP of Engineering, who aimed to improve stability, reduce deployment failures, and increase release frequency by shifting from a monolithic to a microservices-based architecture.

During this overhaul, engineers developed tools and automated processes to detect and mitigate code issues before deployment. This transformation allowed LinkedIn to scale more effectively, moving from deploying bi-weekly to multiple times a day, significantly reducing the technical debt that had plagued their system. This initiative also demonstrated how prioritizing long-term operational stability over immediate feature delivery could drive sustainable growth and resilience, laying the groundwork for continuous delivery practices at LinkedIn.

The main lessons from this case study are the importance of periodically addressing technical debt and the need for cultural shifts within engineering teams to prioritize system reliability alongside feature development. By investing in non-functional improvements, LinkedIn was able to foster a more agile and robust infrastructure capable of supporting its rapid user growth and complex service demands.