Vincent Ryan
02/09/2025
Module 6

There were two case studies in chapter 13, one on Amazon and one on Blackboard. Both of these case studies point towards the inevitability of long-standing companies. The longer you exist, the more pressure it puts on the system to change, but growth gets harder without planning for it. Both Amazon and Blackboard struggled with monolithic structures that were very tightly woven. This reaches a tipping point where new code continually breaks older code, and more time is spent repairing and having meetings about what could be done than actually getting anything done.

Both companies implemented the “strangler method” to transform monolithic code into smaller microservices. It took a great deal of analysis to find what components could be divorced from each other. During this analysis, they focus on the user, including the programmers, to find the best applications to split into their teams and components. Focusing on the user brings new insights and helps design where the splits should happen.

Old systems were carefully switched one piece at a time. To support this, better testing was designed and created so code commits could be trusted and encouraged to be committed more regularly. The lesson to take away from this is to learn from others and pay attention to where you’re going and what you’re working on. Always prioritize flexibility and support, and your coding will go further.

Kim, G., Humble, J., Debois, P., & Willis, J. (2016). The DevOps Handbook How to Create World-Class Agility, Reliability, & Security in Technology Organizations. IT Revolution Press.