**Module 6.2 – Case Study: Strangler Pattern at Blackboard Learn (2011)**

Dan Zhu

Software Department, Bellevue University

CSD380-O307 DevOps (2261-DD)

Professor Darren Osier

September 14th, 2025

**Case Study: Strangler Pattern at Blackboard Learn (2011)**

The Strangler Fig Pattern case study at Blackboard Learn (2011) introduces how Blackboard Inc., an educational institution, faced challenges caused by its legacy J2EE codebase, which dates back to 1997. By 2010, the chief architect, David Ashman, observed that their build, integration, and testing process had become more complex and error-prone. For large products, it has longer lead times and worse outcomes for customers. The time of integration will take twenty-four to twenty-six hours. Therefore, to solve those issues, Ashman implemented a code re-architecting project using the strangler fig pattern.

            The Strangler Fig Pattern is an exemplary implementation of the Strangler Application Pattern. Strangler Application Pattern is a strategy that replaces an old system with a modernized system. All functionalities will be replaced piece by piece in the new system. Both old and new systems will exist simultaneously until the transition is completed. In Blackboard Learn, the development team creates a façade or proxy, which they call Building Blacks internally. Building Blocks allows developers to create separate modules decoupled from the monolithic codebase. According to Ashman, developers can work more autonomously because they don't have to constantly communicate and coordinate with other development teams. After the implementation of the Strangler Fig Pattern, significant changes were observed. The lead time is reduced. Risk is lower and easy to maintain because modules are decoupled.

            The Blackboard Learn case demonstrates how effectively the Strangler Fig Pattern transfers a legacy system to a new, modern one. Our services architecture dictates how we test and deploy our code. According to The DevOps Handbook, the Strangler Fig Pattern can help us migrate between architectures incrementally, enabling us to adapt to the organization's needs.

**Reference**

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