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

Jackie Scott Module 6.2 9/9/2024

The case study in Chapter 13 highlights how Blackboard Inc., an organization specializing in educational technology, tackled the challenges of managing a legacy J2EE codebase for its flagship Learn product. By 2010, the development team was burdened with maintaining a system dating back to 1997, resulting in increased complexity, long lead times, and frequent errors. David Ashman, Blackboard’s chief architect, noted that as the product grew, “our build, integration, and testing processes kept getting more and more complex and error-prone,” leading to feedback delays of 24 to 36 hours and worsening outcomes for customers.

Ashman then decided to illustrate this issue by creating a graph that was generated by their course code repository that dated back to 2005. The graphic confirmed that their organization “needed to do something; otherwise, the problems would keep getting worse, with no end in sight.”

To address these issues, Ashman initiated a re-architecting project in 2012 using the *Strangler Fig Pattern*. The team created *Building Blocks*, a term the organization coined internally, modular components that were decoupled from the monolithic codebase and accessed through fixed APIs. This approach allowed developers to work independently with self-governance. As Ashman explained, “Every developer given a choice would work in the Building Block codebase, where they could work with more autonomy and freedom and safety.” Ashman also noted that, “In combination with the updates to our building process, they also got faster, better feedback on their work, which meant better quality.”

The move to a modular architecture brought significant improvements. The case study shows how this approach led to increased code commits, faster feedback, and overall better software quality. By decoupling code into independent modules, Blackboard reduced coordination overhead and encouraged developers to work more efficiently, resulting in faster delivery and fewer errors. The organization also discovered that applying the *Strangler Fig Pattern* allowed them to gradually transition parts of their code into new structures without disrupting daily operations, ultimately helping them regain control over their development process.