Colton Kohler

Module 6.2 Assignment

6/29/2025

The case study about Blackboard Learn in 2011 illustrates how the company used the Strangler Pattern to transform a legacy monolithic system into a more sustainable, modular architecture. At that time, Blackboard Inc. was a major provider of educational technology, but its Learn product was burdened by an enormous J2EE codebase dating back to 1997. David Ashman, the chief architect, noted, “We still have fragments of Perl code still embedded throughout our codebase,” highlighting just how outdated parts of the system had become (Kim et al., 2016, p. 229).

By 2010, the scale and complexity of the application were causing severe slowdowns in development and testing. Ashman observed that “our build, integration, and testing processes kept getting more and more complex and error prone,” and that integration feedback often required “twenty-four to thirty-six hours” (Kim et al., 2016, p. 229). Data from the source code repository revealed that while the total number of lines of code was rising, the number of commits steadily declined, signaling that it was becoming increasingly difficult for developers to make changes safely.

In response, Blackboard began a re-architecture initiative in 2012 using the Strangler Pattern. The team introduced Building Blocks, a system of decoupled modules accessed through fixed APIs. This approach allowed developers to “work with more autonomy, without having to constantly communicate and coordinate with other development teams” (Kim et al., 2016, p. 230). Over time, developers migrated more of their work into Building Blocks, reducing the monolith’s size and improving productivity.

The benefits were significant. Developers could isolate changes to smaller components, making failures less likely to cascade throughout the application. Faster build processes also provided quicker feedback, leading to better overall quality. Ashman concluded that the approach resulted in “impressive improvements in code modularity,” which allowed teams to regain speed and confidence while modernizing their system incrementally (Kim et al., 2016, p. 230).

This case demonstrates how the Strangler Pattern can help organizations reduce risk, improve maintainability, and modernize legacy systems without disrupting customer service.

**Reference**

Kim, G., Humble, J., Debois, P., & Willis, J. (2016). The DevOps Handbook: How to create world-class agility, reliability, and security in technology organizations. IT Revolution Press.