Jose Franco

06/24/25

Module 6.2 Assignment

Case Study: Strangler Pattern at Blackboard Learn (2011)

**Summary of main points**

In 2011, Blackboard Inc., a leading provider of educational technology, faced significant issues maintaining its flagship product, Blackboard Learn. The application was built on an aging J2EE monolithic codebase dating back to 1997, which included fragments of outdated Perl code. As noted by David Ashman, the company’s chief architect, growing system complexity caused lead times for builds, testing, and integration to increase dramatically. Integration feedback alone could take between 24 to 36 hours, severely impacting productivity and delivery speed.

Source control analysis from 2005 to 2012 revealed that while the number of lines of code steadily increased the number of code commits gradually declined. This trend indicated a growing difficulty in managing and safely integrating new changes within the monolithic system.

In response, in 2012 Ashman’s team adopted the Strangler Fig Pattern. This approach involved incrementally refactoring and replacing parts of the old system by creating a modular architecture called Building Blocks. These modules were decoupled from the monolithic system and accessed through fixed APIs, giving developers more autonomy and enabling faster, safer, and more isolated updates. This significantly improved code modularity, reduced risk, and allowed Blackboard to modernize its platform without a risky, large-scale rewrite.

**Lessons learned**

This case study demonstrates how large, legacy systems can be successfully modernized using the Strangler Fig Pattern. By isolating new modules from an aging monolith, teams reduce risks, increase development speed, and improve product stability. It also underscores the importance of measurable metrics, like code commit rates and codebase size, to guide architectural decisions.

The approach taken by Blackboard exemplifies best practices in continuous delivery and system re-architecture, proving that gradual, modular modernization can deliver better quality, quicker feedback, and safer production environments.

**Sources**

Kim, G., Humble, J., Debois, P., & Willis, J. (2021). *The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations.* (2nd ed.). IT Revolution Press. (Chapter 13, pp. 215–217)

Fowler, M. (2015). *Strangler Fig Application*. Retrieved from <https://martinfowler.com/bliki/StranglerFigApplication.html>