Lea Trueworthy

CSD 380 Secure Software Development

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

Nathan Braun

June 28, 2025

In 2011, Blackboard Learn, a widely used online education platform, faced significant challenges. Its outdated, monolithic architecture made maintenance difficult, and it struggled to meet growing user demands and performance expectations. The development team needed to improve the system's agility while ensuring continuous availability and reliability.

A complete system rewrite was deemed too risky and time-consuming, potentially delaying new features and increasing the chance of failure. Instead, as documented in *The DevOps Handbook*, the team chose the Strangler Pattern. This strategy involves gradually replacing parts of an existing system by building new functionalities around it. Much like a strangler fig tree envelops and eventually replaces its host, new services were incrementally developed to surround and take over the legacy components of Blackboard Learn. This approach allowed for modernization with minimal disruption and risk.

The modernization process started by pinpointing components of the system that could be separated and restructured as standalone services. New features were built using the modernized architecture, while legacy parts were gradually refactored or replaced. A key aspect of the transition was the implementation of a service layer that served as an intermediary. This layer routed user requests to either the legacy system or the new one, depending on where the functionality resided. This approach enabled both systems to function concurrently, ensuring a seamless experience for users during the transition.

By embracing the Strangler Pattern, Blackboard Learn avoided the risks associated with a complete overhaul. The team was able to release updates continuously, reduce technical debt incrementally, and move toward a more scalable, modular system. The strategy aligned with DevOps principles, emphasizing automation, collaboration, and rapid feedback loops.

Lessons Learned:

* Incremental change reduces risk: Migrating individual components allowed the team to avoid the dangers of a full rewrite.
* Service abstraction promotes flexibility: A service layer enabled seamless communication between legacy and modern systems, simplifying integration and transition.
* Continuous delivery keeps value flowing: Users benefited from regular updates instead of waiting for a major release.
* Patience and careful planning are crucial: Achieving success required a long-term vision, thoughtful strategy, and consistent execution.
* DevOps practices facilitated modernization: Automation and strong cross-team collaboration enabled efficient deployment and testing of new services.

In the end, the Strangler Pattern offered Blackboard Learn a practical and low-risk path for modernizing its legacy system into a service-oriented architecture. By incrementally replacing legacy components with modern services, the team preserved system stability and ensured a positive user experience throughout the transition. This case study highlights how intentional architectural strategies, combined with steady DevOps practices, can help organizations successfully modernize even the most deeply rooted legacy systems.

References:

ElHousieny, R. (2021, January 24). *The Strangler Pattern: Microservices Design Patterns In Practice Made Easy with AWS Serverless*. Medium; Nerd For Tech. https://medium.com/nerd-for-tech/the-strangler-pattern-microservices-design-patterns-in-practice-made-easy-with-aws-serverless-97437a82b052

Kalsi, H. S. (2024, September 6). *Understanding the Strangler Pattern in Software Development*. Medium; I am a dummy, enlighten me! https://medium.com/i-am-a-dummy-enlighten-me/understanding-the-strangler-pattern-in-software-development-4e4f58432535

Kim, G., Debois, P., Humble, J., Willis, J., & Forsgren, N. (2021). The DEVOPS Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations (2nd ed.) [Book]. It Revolution Press.