

Austen Rhyce Erickson

CSD380: DevOps

Assignment 6.2: Case Study: Strangler Pattern at Blackboard Learn

2/07/25

In chapter 13 of our textbook is a case study exploration of Blackboard Learn's experience with the strangler pattern. It started with increased challenges with their legacy codebase. Their monolithic codebase had become increasingly complex and they experienced a slowing of development and more occurrences of errors. Their build and testing cycles took more than a day. Additionally, they noted a decrease in developer productivity. Commit history showed that while lines of code were increasing, the number of code commits themselves were decreasing. This was a culprit in the noted difficulty in making changes.

In the interest of curing their woes, the development team decided to make a move towards a process they called building blocks which were modular components that were decoupled from the monolith and accessed via fixed APIs which allowed developers to work independently. Although they didn't call it a strangler at the time, the building block strategy was, in effect, a strangler pattern.

The building block strangler pattern became popular as developers preferred working in the new modular system. There was an increase in code commits and a decrease in the size of the monolithic repository. Mistakes were contained thus preventing widespread failures. The overall effect of the modular approach and an improved build process was faster feedback loops and better code quality.

This case study emphasizes how modularization enhances productivity. Changes can be made faster when components are decoupled from a monolith. Independent work can be done concurrently. Another takeaway is that legacy systems need incremental refactoring. The strangler allows granular modernization without disrupting the entire system. The faster feedback loops that result from this pattern improves code quality and integration time is reduced. Isolating localized failures leads to a more stable development process. Lastly, giving developers autonomy increases morale, efficiency, and innovation. Yay stranglers!