**Strangler Fig Pattern Blackboard Learn Assignment 6.2**

Robert Minkler

College of Science & Tech, Bellevue University

CSD 380 DevOps

Professor John Woods

April 20, 2025

**Strangler Fig Pattern**

Blackboard Learn was initially developed as a monolithic system to deploy on a client’s local system. As the codebase grew and aged, the project became difficult to maintain. The project’s build, integration, and testing phases grew more extensive and error-prone. Integration feedback was often slow, causing further delays. Due to the complex nature of the project, development lead time increased, and the results deteriorated. The time and effort invested in maintaining the system were not yielding the expected results.

They recognized a need for a radical shift in how they developed the system. In 2012, they re-architected the project using the Strangler Fig Pattern to replace the aging systems with new ones. The new systems would be easier to maintain to improve productivity and deliver value to their clients faster. They divided the project into building blocks or micro services that could slowly replace the aging monolithic system. Decoupling services and providing a fixed API allowed developers the freedom to work on an individual building block without the risk of damaging other code. The system would remain functional as long as the API remained intact. This gave the team more flexibility to improve the code and release new features without the concerns of breaking a large integrated system.

The reduced risk also allowed for greater autonomy. Teams were less reliant on other teams, allowing them to move faster. Productivity soared in response to the changes, and developers chose to work in the decoupled modules rather than the old monolithic codebase. In addition to the architectural adjustments, updates were made to the build process, improving feedback speed and detail.

Lessons learned:

* + There comes a time when monolithic codebases become too large to maintain.
  + Monolithic systems are harder to maintain as they scale because of their coupled dependencies.
  + Breaking the system into smaller, modular services can improve maintainability.
  + A gradual shift using the Strangler Fig Pattern allows developers to migrate services over time safely.
  + A build process integrating automated testing with rapid feedback allows a faster workflow with improved results.