Jon Green

Module 6.2: Strangler Fig Pattern

In 2011, Blackboard was a leader in technology for educational institutions. The company was producing a staggering annual revenue of approximately $650 million dollars. On the surface, Blackboard was dominating the space however, it had significant issues internally. One of the key issues stemmed from the company continuing to use legacy software in the codebase from 1997. The legacy coding began to create various issues such as increasing lead times due to complexity of integrating the current systems with older software. In some cases, it was taking upwards of 24-36 hours to get feedback from the integration process. The software developers were struggling to keep up with standard operations due to software compatibility issues. As a result of the software issues, the final product being deployed to customers was beginning to suffer.

In 2012, David Ashmen (chief architect) designed a new strategy that would alleviate the issues being caused by integrating new systems with legacy software. The plan was to use the strangler fig pattern to conduct a new code architecting project to resolve the issue. The strangler fig pattern is the strategy of creating a new system that expands over the top of the existing system. It continues to grow until the old system is “strangled” or rendered obsolete. When this occurs, the old system can be removed or replaced in its entirety without impacting the users or the product.

Ashmen’s system was referred to as “Building Blocks” and was implemented by having the team establish the “new code” in smaller and more manageable sections while the old code was still in use. They would gradually move parts of the old code from the Blackboard code repository into the Building Blocks sections to produce new versions of the code. This allowed them to focus on these sections and work independently while not disturbing the existing code. Utilizing the Building Blocks areas also prevented catastrophic failures in the legacy code in the event that a problem occurred. Once all the individual sections were complete, they were combined to replace the old system that contained the legacy code. This modular approach allowed the developers to safely decommission the legacy code and work with more independence and freedom.

Sources:

Shopify. “Refactoring Legacy Code with the Strangler Fig Pattern - Shopify.” *Shopify*, 11 Mar. 2020, shopify.engineering/refactoring-legacy-code-strangler-fig-pattern. Accessed 13 Sept. 2024.

‌ Kim, Gene, et al. *The DevOps Handbook : How to Create World-Class Agility, Reliability, & Security in Technology Organizations*. Portland, Or, It Revolution Press, Llc, 2021.

‌