Brett Fuller

CSD-380 Assignment 6.2

6/29/2025

In 2010 David Ashman of Blackboard started to observe that there was a lot of complexity and legacy code present on their product Learn. They were seeing long lead times for and error prone deployments of that application. This was leading to a worse product for their customers and Ashman knew that they needed to do something. In studying the problem Ashman noticed that the more lines of code present in their product the less code commits they were seeing. Ashman perceived that this relationship was demonstrating that the structure of their application had produced a product that was too difficult to make changes to, and he believed that if they continued to iterate on the same design that the problem would continue to get worse rather than improve.

In order to address this issue, they introduced a code restructuring project utilizing the Strangler Fig pattern. Here they utilized what they called “Building Blocks” which was a process of breaking their Monolithic application up into smaller modules that could be worked on independently of the whole through APIs. Moving to this new paradigm saw an exponential increase in both lines of code and commits as the modular design allowed them to work with less risk and make more frequent updates, since any issues introduced with new code were generally isolated to their smaller module and could be easily addressed from there.

Key points for the Blackboard Learn Strangle fig pattern Case Study

* **Legacy System Issues:** The outdated codebase led to complex and error prone build, integration, and testing processes, resulting in longer lead times and poor customer outcomes.
* **Solution Implemented:** Blackboard adopted the strangler fig pattern, creating “Building Blocks” as separate modules decoupled from the monolithic codebase, enabling developers to work independently and access functionalities through fixed APIs.
* **Adoption:** Developers preferred working with Building Blocks for increased autonomy, leading to a decrease in the monolith repository size.
* **Impact:** Building Blocks codebase enhanced developer productivity, code modularity, and system safety by localizing failures.
* **Benefits:** Developers experienced faster and better feedback on their work, resulting in improved code quality.

The use of the Strangler Fig methodology allowed Blackboard to iterate on their product in a far more efficient manner. Utilizing micro services or “Building Blocks” allowed issues to be isolated to their own Block or service rather than directly integrated with a tightly coupled monolithic product. Making the change to their “Building Block” modular design allowed developers to dramatically increase deployments and provided a better experience for their customers.

**References**

Kim, G., Humble, J., Debois, P., Willis, J., & Forsgren, N. (2021). *The DevOps Handbook, 2nd Edition: How to Create World-Class Agility, Reliability, & Security in Technology Organizations*. IT Revolution.