Zachariah King

September 14, 2025

Case Study: Strangler Pattern at Blackboard Learn - Assignment 6.2

CSD 380-O301

Main Points

A. Blackboard Inc. (a highly successful and founding provider of educational technology systems) still had code in their current system from 1997 and this was in 2011.
    A. This came about from an inspection over growing lead times and number of errors that come from the old system.
B. Developer productivity was also being negatively affected with drastically less regular commits than before.
    A. The rate of growth for lines of code kept growing though, but the commits stayed at a low level.
C. In 2012, the Strangler Fig Pattern was brought in to redesign their system on command from Ashman.
    A. The team used 'Building Blocks' which gave freedom to developers to work on distinct modular codebases that enhance, but don't depend on the original monolithic codebase.
    B. This meant more autonomy and less lines of code within the original codebase.
    C. The monolithic lines of code were being divided and put into 'Building Block' modules based on their purpose which eventually meant more lines of code, but also more code commits.
    D. Productivity was high and mistakes weren't as drastic.
D. The Strangler Fig Pattern definitely helped the Blackboard Learn system architecture, specifically with overall code modularity, developer performance, and system maintainability/quality.

Lessons Learned

A. Even seemingly put-together, proficient systems have their codebase vulnerabilities, especially if they've been around for a while.

B. Growing lead times, growing error occurrences, and growing lines of code combined with less commits is a recipe for disaster and must be addressed before it's too late.

C. Feedback from integration processes must be obtained quickly.

D. Strangler Fig Pattern is all about modularity and using smooth APIs to update a system without totally demolishing it.

E. Not only does this pattern bring organization, it also brings exquisite morale among developers since they are happier with the increased autonomy, freedom, and safety.

F. More commits means faster productivity, more feedback, and more assurance in developer capability and code quality.