Ryan Norrbom

CSD380

Assignment 6.2

September 14th, 2024

Blackboard Learn’s Architecture

In this week’s case study, Blackboard Learn faced challenges due to reliance on a monolithic architecture built upon technology that was becoming increasingly outdated. The legacy applications included J2EE and even remnants of Perl code dating back to 1997. This legacy code was causing increasingly complex and error-prone build, integration, and testing processes. As a result, integration feedback times were 24-36 hours. These prolonged feedback loops, among other issues caused by the monolithic structure, made introducing new features or quickly addressing bugs challenging. Furthermore, these challenges were directly impacting customer satisfaction and operational efficiency.

By 2010, the development team at Blackboard determined that the system's complexity and growing technical debt were unsustainable. The application's nature, combined with its outdated technology stack, created bottlenecks in development processes. The chart in the study outlined how, as the organization expanded, lead times for development and testing increased, and the frequency of code commits decreased. This was one of the indicators that developers were struggling with the existing codebase, underscoring the urgency for change.

In response to these challenges, Blackboard's development team began work using the Strangler Fig Pattern, which allowed them to incrementally replace parts of the monolithic codebase with a more modular architecture for which they called Building Blocks. This shift to Building Blocks enabled developers to work in smaller, more autonomous teams, focusing on distinct modules decoupled from the main system. These modules communicated through APIs, drastically reducing interdependencies and streamlining the development process.

The impact of this transition was profound. As modules were moved out of the monolithic repository into the Building Blocks framework, the size of the legacy codebase began to decrease. In contrast, the number of commits to the new codebase increased significantly. This was a clear and early demonstration of improved developer productivity and engagement by the new framework.

Blackboard’s experience emphasizes modernizing legacy systems to keep pace with evolving business needs and technological advancements. By embracing a modular architecture and incrementally removing the monolithic solution using the Strangler Fig Pattern, Blackboard improved developer autonomy and productivity. In addition, it delivered a more resilient and responsive platform to its customers. Moving to this more modular solution, the Blackboard teams were able to work more independently, make quicker changes, and reduce the risk of system-wide failures.

**References**:

Kim, G., Debois, P., Willis, J., Humble, J., Forsgren, N., & Allspaw, J. (2021). *The devops handbook: How to create world-class agility, reliability, & Security in Technology Organizations*. IT Revolution Press, LLC.