CHYNNA LEE

MODULE 6 ASSIGNMENT

DEVOPS

JULY 1, 2024

In 2011, Blackboard Inc., a leading provider of educational technology, faced significant challenges with their legacy J2EE codebase, which dated back to 1997. The old system's complexity and increasing lead times were becoming unmanageable, as noted by David Ashman, Blackboard's chief architect. He highlighted that their build, integration, and testing processes were not only becoming more complex and error-prone but also extending feedback cycles to between twenty-four to thirty-six hours. This growing inefficiency was visibly impacting developer productivity, as evidenced by historical graphs from their source code repository. The data revealed a troubling trend: while the lines of code in the monolithic Blackboard Learn codebase continued to increase, the number of code commits was decreasing, indicating the growing difficulty of introducing code changes.

To address these issues, Ashman spearheaded a code rearchitecting project in 2012, implementing the strangler pattern. This approach involved creating "Building Blocks," which allowed developers to work in separate, decoupled modules accessed through fixed APIs. By moving code into these modular Building Blocks, developers could work with greater autonomy and without the constant need for coordination with other teams. This shift not only reduced the size of the monolithic codebase but also led to exponential growth in the number of code commits within the Building Blocks code repositories. Developers preferred working in this new environment, where they experienced more freedom, autonomy, and safety. The modular architecture ensured that mistakes resulted in localized failures rather than catastrophic impacts on the entire system.

The implementation of the strangler pattern at Blackboard Learn brought about impressive improvements in code modularity, allowing developers to work more independently and efficiently. The updates to their build process provided faster and better feedback, leading to higher quality outcomes. Ashman noted that the new architecture significantly improved developer productivity and reduced the risk of large-scale failures. This case study underscores the importance of adaptable architectures in software development, as highlighted in Puppet Labs’ 2015 State of DevOps Report, which identified architecture as a key predictor of engineering productivity and the ability to implement changes quickly and safely.

Overall, the case study of Blackboard Learn's use of the strangler pattern demonstrates how incremental migration from a monolithic to a modular architecture can effectively address legacy system challenges. It highlights the benefits of modular architectures in enhancing developer productivity, ensuring system reliability, and improving overall software quality. By adopting such techniques, organizations can better align their architectures with contemporary needs, facilitating smoother transitions and reducing the risks associated with large-scale changes. This approach is particularly relevant in today's rapidly evolving technological landscape, where flexibility, scalability, and resilience are paramount.