Improving Product Quality and Addressing Pitfalls: A Case Study of Blackboard Learn

Kristina Vasquez

Bellevue University

CSD370-J308 Secure Software Development (2247-DD)

Sue Sampson

11/24/2024

As DevOps engineers, our goal is to continuously improve the code and products we deliver to customers while identifying areas for improvement in our processes. One crucial practice in achieving these goals is continuous integration, a key element of DevOps. According to the article *What is Continuous Integration?*:

Continuous integration most often refers to the build or integration stage of the software release process and entails both an automation component (e.g., a CI or build service) and a cultural component (e.g., learning to integrate frequently). The key goals of continuous integration are to find and address bugs quicker, improve software quality, and reduce the time it takes to validate and release new software updates (Amazon Web Services [AWS], n.d.).

This concept is reflected in the case study from Chapter 13 of *The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations* (Kim, Humble, Debois, & Willis, 2016), which discusses how Blackboard, a leading learning management system, applied DevOps practices to modernize its legacy application.

**Key Points from the Case Study**

1. **Strangler Pattern Overview**  
   The case study focuses on Blackboard’s use of the *Strangler Pattern* to modernize its legacy system. This approach involves gradually building new features alongside the existing system, allowing the old system to be replaced incrementally over time. This method reduces the risks and costs associated with a full system rewrite (Kim et al., 2016).
2. **Problem at Blackboard**  
   Blackboard Learn’s legacy system had become monolithic and difficult to maintain, built on outdated technology. As demand grew, scaling the system became increasingly challenging. The organization struggled to innovate quickly, release new features, and maintain reliability.
3. **Approach**  
   Blackboard adopted the Strangler Pattern, focusing on replacing parts of the monolithic system incrementally. By identifying areas for improvement and re-architecting them gradually, Blackboard was able to modernize the system without disrupting ongoing operations.
4. **Implementation**  
   Blackboard’s migration strategy involved integrating new microservices and APIs that could coexist with the legacy system. The goal was to decouple new features from the old system, ensuring a smooth transition. This incremental approach allowed Blackboard to manage the risks associated with replacing its outdated infrastructure.
5. **Cultural Shift**  
   The transformation required more than just technical changes. It also necessitated a cultural shift toward continuous delivery, automation, and cross-functional collaboration. By adopting DevOps principles, Blackboard was able to deliver more reliable releases, increase agility, and respond more effectively to user needs.

**Lessons Learned from the Case Study**

1. **Incremental Change Reduces Risk**  
   The Strangler Pattern proved effective in managing the complexity and risks of transitioning from a legacy system. By replacing pieces of the system over time, Blackboard ensured stability and minimized the risk of large-scale failure.
2. **Continuous Delivery and Automation Are Key**  
   The success of Blackboard’s migration relied heavily on implementing continuous integration and continuous delivery (CI/CD) pipelines, along with automated testing and deployment. These practices enabled Blackboard to make changes quickly, safely, and with minimal downtime.
3. **Cross-Functional Collaboration**  
   To achieve successful modernization, Blackboard had to break down silos between departments. Adopting a DevOps culture—where teams work together to deliver code frequently and monitor systems continuously—was essential for achieving the desired results.
4. **Focus on Business Value**  
   Blackboard learned that technology migrations should align with business goals. The primary objective was not only technical improvement but also providing better value to customers and enhancing the user experience.
5. **Gradual Migration Over "Big Bang" Rewrites**  
   The case study underscores the dangers of attempting a full system rewrite all at once, which can introduce significant risk. By gradually migrating the system using the Strangler Pattern, Blackboard was able to manage complexity and reduce disruptions.
6. **Scalable Architecture and Agility**  
   Through this process, Blackboard learned the importance of building scalable, agile systems that could easily adapt to changing business needs. The new architecture enabled quicker innovation and more effective handling of future growth.

**Conclusion**

The Blackboard case study demonstrates the successful application of the Strangler Pattern and DevOps practices to modernize a legacy platform. By prioritizing incremental change, automation, collaboration, and alignment with business value, Blackboard achieved significant improvements in agility, reliability, and scalability. This approach minimized the risks of a full system overhaul while ensuring continued service delivery and system modernization.

**Citations:**

Amazon Web Services. (n.d.). *What is continuous integration?* Retrieved from <https://aws.amazon.com/devops/continuous-integration/#:~:text=Continuous%20integration%20is%20a%20DevOps,builds%20and%20tests%20are%20run>

Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps handbook: How to create world-class agility, reliability, & security in technology organizations* (2nd ed.). IT Revolution Press.