**Case Study: Strangler Pattern at Blackboard Learn (2011)**

Larry Shane Tinsley

Bellevue University

CSD 380: DevOps 6.2

Dr. Joseph Issa

June 29, 2024

**Case Study: Strangler Pattern at Blackboard Learn (2011)**

**Summary of Main Points:**

1. **Legacy Code Challenges**:
   * Blackboard Inc., a major provider of educational technology, faced significant challenges with their legacy J2EE codebase dating back to 1997.
   * The complexity and lead times associated with the old system increased, negatively impacting developer productivity and customer outcomes.
   * Integration and testing processes were error-prone and lengthy, requiring 24 to 36 hours to provide feedback.
2. **Evidence of Productivity Decline**:
   * David Ashman, Blackboard's chief architect, identified a troubling trend where the number of code commits decreased while the lines of code in the monolithic repository grew.
   * This indicated the increasing difficulty of making changes to the codebase, leading to a need for a significant change in approach.
3. **Implementation of Strangler Pattern**:
   * In response to these challenges, Ashman initiated a re-architecting project using the strangler fig pattern in 2012.
   * The solution involved creating "Building Blocks," separate modules decoupled from the monolithic codebase and accessed through fixed APIs.
   * This approach allowed developers to work more autonomously without constant coordination with other teams.
4. **Impact of Building Blocks**:
   * The introduction of Building Blocks led to a decrease in the monolithic source code repository size as developers moved their code into the new, modular repositories.
   * Developers preferred working in the Building Block codebase due to its increased autonomy, freedom, and safety.
5. **Improved Developer Productivity and System Safety**:
   * The new codebase for Building Blocks enabled exponential growth in the number of lines of code and the number of code commits.
   * This shift allowed for safer and more productive development, with mistakes causing local failures rather than widespread system issues.
6. **Overall Improvements**:
   * The modularity achieved through the Building Blocks architecture led to impressive improvements in code quality and developer independence.
   * Updates to the build process provided faster and better feedback, enhancing code quality and development speed.

**Lessons Learned:**

1. **Need for Modernization**:
   * Legacy systems, while functional, can severely hinder productivity and innovation over time. Modernizing these systems is crucial for maintaining competitiveness and efficiency.
2. **Strangler Fig Pattern Effectiveness**:
   * The strangler fig pattern is an effective strategy for gradually replacing legacy systems with modern architectures, allowing continuous improvement without complete system overhauls.
3. **Modularity and Autonomy**:
   * Decoupling components into modular units, as seen with Building Blocks, significantly enhances developer productivity and reduces coordination overhead.
   * Increased autonomy leads to faster development cycles and improved developer job satisfaction.
4. **Importance of Feedback Loops**:
   * Faster and more accurate feedback mechanisms are essential for maintaining high code quality and efficient development processes.
   * Continuous integration and automated testing are vital components of a robust development workflow.
5. **Cultural and Organizational Change**:
   * Implementing such architectural changes requires technical solutions and a cultural shift within the organization.
   * Ensuring buy-in from all stakeholders and fostering a culture of continuous improvement are critical for successful implementation.

In summary, Blackboard Inc.'s transition to a modular architecture using the strangler fig pattern significantly improved their development process, enhancing productivity, better code quality, and greater developer satisfaction. This case study underscores the importance of modernizing legacy systems and the benefits of adopting modular, autonomous development practices. In fact, it may be time for Blackboard Inc. to revisit the process for the new needs of its clients.

**References**

Kim, G., Humble, J., Debois, P., & Willis, J. (2016). *The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations*. IT Revolution Press.