The case study "Strangler Fig Pattern at Blackboard Learn (2011)" shows the challenges faced by Blackboard Inc., a major provider of educational technology, as they struggled with a legacy J2EE codebase dating back to 1997. By 2011, the company's flagship product, Learn, had become difficult to maintain due to outdated technology and complex code, which included remnants of Perl.

David Ashman, Blackboard’s chief architect, noted that the increasing complexity of their build, integration, and testing processes significantly slowed down development. As the product grew larger, lead times increased, and the quality of outcomes worsened. Integration feedback alone would take 24 to 36 hours. Ashman identified the problem through data showing that while the codebase size continued to grow, the frequency of code commits was declining, indicating that making changes was becoming increasingly difficult.

Recognizing this trend, Ashman realized that without intervention, the issues would only escalate, threatening developer productivity and product reliability.

This situation underscores the importance of recognizing technical debt early and implementing a strategy to modernize legacy systems before they become unmanageable.

In 2012, David Ashman led a code re-architecting project at Blackboard Learn using the strangler fig pattern to address the growing complexity of their legacy system. The development team implemented Building Blocks, which were modular components decoupled from the monolithic codebase and accessed through fixed APIs. This approach gave developers autonomy, allowing them to work independently without constantly coordinating with other teams.

As developers migrated their code into these Building Block modules, the size of the monolith codebase began to shrink, while productivity increased. Developers preferred working in the new, more flexible codebase, which enhanced freedom, safety, and efficiency. The smaller modules reduced the risk of catastrophic failures, as issues became localized instead of affecting the entire system. The exponential growth in both lines of code and code commits for Building Blocks showed the success of the approach in improving both code quality and developer productivity.

References:

Kim, Gene; Humble, Jez; Debois, Patrick; Willis, John; Forsgren, Nicole. The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations (p. 213). IT Revolution Press. Kindle Edition.

https://learn.microsoft.com/en-us/azure/architecture/patterns/strangler-fig