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

Kypton Lantz

CSD380-O307 DevOps (2261-DD)  
Module 6.2 Assignment

September 9, 2025

Legacy systems have a way of slowing down development without anyone noticing until it is a real problem. Blackboard Inc. ran into this in 2011 with its Blackboard Learn platform, which was a monolithic J2EE application still running bits of legacy Perl. Integration and testing could take more than a day, which made even small updates risky and slow (Kim et al., 2021). Instead of trying to rewrite the entire system all at once, Blackboard used the Strangler Fig Pattern. This approach allowed the team to gradually update parts of the system while keeping the old functionality running.

The company implemented “Building Blocks,” small modular components that developers could use through fixed APIs. New features were built in these modules, not in the old system. Over time, developers naturally moved toward using the Building Blocks, and the monolithic codebase started to shrink. At the same time, commits to the new modules increased, showing that development was both faster and safer (Kim et al., 2021). Because each module was isolated, any mistakes stayed contained, which made releases less stressful and easier to manage.

This approach matches what experts say about modular architecture. For example, Harrison Clarke (2023) points out that modular systems make it easier to add or change features without touching the whole system. That means faster development, less frustration, and simpler maintenance. Blackboard’s experience shows this in action: developers could work independently, focus on smaller chunks of code, and reduce the risk of breaking something else.

There are a few lessons that stand out. First, tackling legacy systems little by little is safer and more effective than a full rewrite. Second, breaking work into smaller modules gives teams more autonomy and keeps problems from affecting the whole system. Finally, the way you design architecture influences how people work. When a safer, easier path exists, developers naturally take it. Blackboard’s Building Blocks encouraged good habits without needing to enforce them.

Of course, moving to this kind of system is not just about technology. Leadership has to support the change, and teams need to understand the benefits. Starting small, defining clear API boundaries, and tracking progress with things like commits or shrinking legacy code can help keep momentum. This gradual approach builds confidence and keeps the system running while modernizing it.

In the end, the Blackboard Learn case shows how the Strangler Fig Pattern makes it possible to modernize safely. Moving features into modular Building Blocks sped up development, reduced risk, and improved maintainability. Combined with what experts say about modular design, this case makes it clear: gradual, thoughtful change is the best way to bring old systems into the modern era.

**References:**

Harrison Clarke. (2023). *Benefits of modular architecture: Moving from monolithic to modular*. Retrieved from [https://www.harrisonclarke.com/blog/benefits-of-modular-architecture-moving-from-monolithic-to-modular](https://www.harrisonclarke.com/blog/benefits-of-modular-architecture-moving-from-monolithic-to-modular?utm_source=chatgpt.com)

Kim, G., Humble, J., Debois, P., Willis, J., & Forsgren, N. (2021). *The DevOps Handbook: Second Edition* (Case Study: Strangler Fig Pattern at Blackboard Learn, pp. 215–217). IT Revolution.