

Romany Manriquez

In 2011, Blackboard Learn, a major player in educational technology, faced a growing challenge with its outdated codebase. Their flagship product, Learn, had been around since the late 90s, and by 2011, the legacy J2EE code had become a nightmare to work with. David Ashman, the company's chief architect, noticed that their code repository was not only growing in lines of code but also becoming harder to manage. The worst part? They still had bits of Perl code from the 90s hidden in the codebase, which slowed everything down. Ashman realized that the complexity of their system was leading to longer development times, with even simple integration tests taking up to 36 hours to process.

This bottleneck had a direct impact on developer productivity. Fewer code commits were being made, even though the number of lines of code kept increasing. The development team found it harder and harder to make changes without breaking something. Ashman knew that if nothing changed, the situation would only get worse.

In 2012, Ashman decided it was time for a radical solution: using the strangler fig pattern. This approach allowed Blackboard to slowly phase out parts of the old, monolithic codebase without rewriting the entire system from scratch—a costly and risky process. Instead, Ashman's team introduced "Building Blocks," which were small, modular pieces of code that were decoupled from the main system and could be accessed through fixed APIs.

With Building Blocks in place, developers could finally work with more independence. They were no longer forced to deal with the complex and fragile monolith every time they wanted to make a change. Instead, they worked on their own modules, moving their code into the Building Blocks repository. As a result, the size of the monolithic codebase started to shrink, while the new modular codebase grew. Developers preferred working in the Building Blocks environment because it gave them more freedom, flexibility, and safety. If they made a mistake, it only affected their small module, not the entire system.

The introduction of Building Blocks didn't just make the system easier to manage—it also sped up the development process. Developers received faster feedback on their work, which helped them catch issues earlier and improve the quality of their code. As a result, productivity increased, and the team could deliver updates more frequently and with fewer errors.

The Strangler Fig Pattern proved to be the perfect solution for Blackboard. It allowed them to transition from their outdated legacy system without a full rebuild. More importantly, it empowered the development team by giving them more autonomy and enabling them to work more safely. Ashman's decision to focus on modularity and independence transformed the way Blackboard's developers worked, resulting in a better, more efficient product.

The key lesson here is that using a modular approach like the strangler pattern can be an effective way to deal with aging, monolithic systems. It lets you tackle one problem at a time without overwhelming the team or risking the entire system. By breaking down the work into smaller, manageable pieces, teams can work faster, safer, and with more flexibility—all while improving the overall quality of the system.

#### References:

The DevOps Handbook

How to Create World-Class Agility, Reliability, & Security in Technology Organizations  
by Gene Kim; Jez Humble; Patrick Debois; John Willis; Nicole Forsgren