Alisa Steensen 6/25/25

Module 6.2 DevOps

The DevOps handbook, provides a case study on how Blackboard Learn successfully used the Strangler Pattern to update and modernize their legacy systems. In 2010 their chief architect, Dave Ashman, decided to focus on the problems associated with their older monolithic system.

Over the years, Blackboard Learn had turned into what’s called a monolith where all the components are intermixed, like the user interface, business logic, and all the behind the scenes stuff. Because of this type of architecture, repairing bugs or changing parts of the system had become difficult. But as the software grew and more people used it, the program became harder and harder to manage. The team then faced longer development cycles and an increased risk when making changes. The team realized that rewriting the whole thing would take too long and could put the system at risk.

So, they tried a different plan: the Strangler Pattern. A strategy where parts of the legacy system are gradually replaced with new services. Because of this type of approach, the old and new systems can exist side-by-side. This allows for new features and components to be built as independently as possible. And over time, the legacy system becomes strangled out until it's entirely replaced.

The team at Blackboard Learn started by figuring out which parts of their big program were causing the most trouble and which ones would make the biggest difference if they were improved. They used graphs to show where the critical issues were and created a plan to address them. This ensured that the new code would be modernized through automated testing and continuous delivery in the pipelines. The old system and the new system would communicate with API’s. This allowed the platform to serve users seamlessly during the transition. Because of this gradual replacement, it meant that users could benefit from the improvements without waiting for the entire system to be replaced.

Here are some good lessons we can learn from what Blackboard Learn did. First, rewriting large complex systems are incredibly risky and can cause a delay in delivering to users. The strangler pattern produces the safer alternative by giving teams the ability to modernize systems piece by piece, while still delivering features and improvements to the user. Second, it’s important to design good APIs so the old and new systems can work together during the switch. This helps avoid breaking things while the changes are happening. Finally, communication is key. Everyone on the team needs to be on the same page; they should agree on the plan, the priorities, and the steps to take. Without this, things can get confusing.

In conclusion, Blackboard Learns situation demonstrates how the Strangler Pattern can help organizations to modernize legacy systems. All while managing the risks and confidently delivering value to the user. This approach shows gradual transformation, which makes it easier to scale, maintain, and improve large scale applications over time.

Reference:

Kim, G., Humble, J., Debois, P., Willis, J., & Forsgren, N. (2021). *The DevOps handbook: How to create world‑class agility, reliability, & security in technology organizations* (2nd ed.). IT Revolution.

Microsoft. (n.d.). *Strangler Fig Application pattern*. Microsoft Learn. <https://learn.microsoft.com/en-us/azure/architecture/patterns/strangler-fig>