Andrew Stiles

CSD-380

Prof. Osier

Assignment 6-2

9/22/2025

**Case Study Summary**

Blackboard Learn ran into serious trouble by 2011. Their main product had been built on a J2EE codebase going back to 1997, and over the years it only became harder to manage. Old fragments of code, even Perl, were still buried inside. By 2010, the process of building, integrating, and testing the system had become painfully slow. Developers sometimes had to wait an entire day or more just to get feedback from integration. That kind of delay makes it hard to innovate, and it drags down morale.

The data told the story clearly. The number of lines of code in the system kept growing, but commits started to drop. What does that show? It shows that developers were struggling to add new features or fix problems. Each change became more complicated and riskier. In my view, this is a perfect example of how unchecked growth in a monolithic system eventually slows everything down.

To solve the problem, Blackboard’s team, led by Chief Architect David Ashman, chose to apply the strangler fig pattern. Instead of trying to rebuild the entire system at once, they introduced something called Building Blocks. These were modular components that worked independently of the old codebase and connected through stable APIs. Developers preferred working in the new structure because it gave them more freedom and much faster feedback. As code moved into the Building Blocks, the size of the monolith started to shrink, while activity in the new modules grew rapidly.

So what’s the takeaway? For me, the key lesson is that trying to force more code into a legacy monolith is a losing battle. Breaking it apart with a modular approach gave Blackboard a way forward. I think the success of the Building Blocks shows that autonomy and safe boundaries matter as much as technical design. When developers can work in smaller, decoupled pieces, they not only produce more, they also produce better. This case proves that re-architecting with the strangler fig pattern is not just a technical fix, but a cultural shift toward faster, more confident work.