Case Study: Strangler Pattern at Blackboard Learn (2011)

Summary:

In 2011, the leading architect at Blackboard Inc., David Ashman, realized that the company’s rate of commits to the code were decreasing as a result of what was essentially technical debt. The company was running on a monolith code which dated back to 1997. This meant that with each successful commit, many more lines of code were required to keep the code in working order and it was becoming increasingly difficult to make updates. Ashman decided to utilize the strangler pattern to slowly phase out the old code by breaking the code into what he called “building blocks”. These building blocks were separate from the monolith code and gave developers more safety and autonomy for developing their own code, as they didn’t have to risk errors to the source code. The adoption of this system helped increase the rate of commits (Kim et al., 2021).

Lessons learned:

Lessons learned from this case study is that there are scenarios when decoupling deployments from releases is beneficial, technical debt happens to everyone, and the strangler pattern is a great way to manage technical debt related problems. Decoupling deployments from releases seems to be something put into practice at Blackboard Learn, as developers would work independently and then push to the main branch. It is unspecified if this push meant a release, but it almost certainly is not because they would need to do further testing to see if the code was compatible with other code updates made by other developers.

Technical debt also made its appearance into this case study, which seems to happen to many large and successful companies. Blackboard had extremely old code that likely could have been streamlined earlier on but wasn’t, and ended up requiring the use of the strangler pattern to phase it out. The strangler pattern was a great way to handle this issue and fit the needs of the company well. They were able to increase productivity as well as developer confidence, which is important if code is to be improved. The use of the strangler pattern also enabled them to slowly chip away at the monolith code that they were dealing with.

Source

Kim, G., Humble, J., Debois, P., Willis, J., Forsgren, N., & Allspaw, J. (2021). *The devops handbook: How to create world-class agility, reliability, & Security in Technology Organizations*. IT Revolution Press.