**Assignment 6.2: Case Study: Strangler Pattern at Blackboard Learn (2011)**

Louis Capps

Bellevue University

CSD380 DevOps

Professor Darren Osier

30 June 2024

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

In “The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations (First edition.)”, the author discusses Blackboard Learn and how Blackboard Inc. used a strangler pattern to combat the issues they ran into with the monolithic program architecture. This paper will look at the problems initially caused by the monolithic program, the decision to do something about it, and the results of that decision. By understanding the entire situation and the benefits gained, we can better decide what type of architecture should be used for development and why the strangler pattern is an effective method to develop a decoupled architect system out of a monolithic program.

As the codebase grew, having a single monolithic program led to issues with scalability, particularly with an excessively long feedback time. On top of that, the identified issues only got worse as time went on. The lesson learned from these issues is that, while a monolithic program may work initially, it is unsustainable in the long term and does not allow for the scalability and growth a company may need.

With the scalability issues worsening, the number of code commits was decreasing, leading to the need for some action to be taken centered around replacing the monolithic program. It was decided to utilize a strangler pattern to replace the monolithic program architecture. They called the decoupled modules Building Blocks. The lesson learned from this situation and during this decision is that replacing a monolithic program is a challenging task. Utilizing a strangler pattern allows that monolithic program to eventually be replaced without completely freezing feature development.

Creating the Building Block modules and decoupling parts of the system from the monolithic program allowed developers to be much more autonomous. Additionally, the new modules were primarily used, and the old monolithic architecture gradually reduced in usage. The developer autonomy gained from this process allowed for exponential growth that would not have been possible before. The lesson learned from introducing this decoupled system is that having a system broken up into separate modules rather than a single massive program allows for a much greater scalability potential. Additionally, the developers themselves felt more free to code features into the system rather than fighting the system as they had before.

By utilizing a strangler approach to their problem, what began as a monolithic program causing major issues and delays turned into a much more reliable and scalable system, allowing Blackboard Inc. to grow exponentially as a company. Through this process that they went through, we can learn some valuable lessons. The first is that, while a monolithic program architecture isn’t always to be avoided, it isn’t easily scalable and may cause major issues in the future. The second is that it takes time to fix the issues and replace a monolithic program, and a strangler pattern can be an effective, if not longer, route to take. The third lesson learned is that having a decoupled development architecture in place allows for growth that would be extremely difficult otherwise.

**Sources**

Kim, G., Humble, J., Debois, P., Willis, J., & Allspaw, J. (2016). The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations (First edition.). IT Revolution Press.