Ashley Landin

CSD 380 DevOps

Module 6.2 Assignment

June 26, 2024

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

This case study analyzes how the Strangler Fig pattern was used with the Blackboard Learn online system in order to improve its scalability. The issues that were arising with the site were lead times increasing as the size of the source code increased. This was causing the several processes associated with the development of the site to lag significantly, and cause issues for users of the site. In order to begin to resolve this issue, they needed to find the direct cause, and data was pulled, and graphs were generated in order to display the trends of developer activity throughout the years.

The main issues were identified after viewing how despite the source code increasing in size, there weren’t enough recorded code commits to reflect that size increase. This meant that there were obstacles in the code committing process that prevented code changes to be implemented in a timely manner. From this point, the strangler fig pattern was referenced in order to start decoupling the legacy codebase that was causing the significant decline in developer productivity. Building Blocks were created to decouple the old system by placing sections of the legacy code into fixed APIs. Through these API’s the development team were able to work on restructuring the code by using modules created through the building blocks.

As the source code was beginning to be placed into the building blocks, the size of the source code began to decrease. These building blocks were starting to restructure the architecture of the entire system, and this was the change that needed to occur for the development team to see an increase in their overall productivity. Along with having more independence when it came to working on the system, there was also an increased level of system security provided by the building blocks. The parts of the legacy code remain protected within the fixed APIs, so when errors or bugs would occur, it wouldn’t lead to large scale failures. The system became safer to work with and easier to fix, since in the case of errors, the building blocks or arguments implemented within the system could be changed without affecting the pieces of the source code stored within the APIs. The main takeaway of the case study was that the architecture of the site was able to be updated to coincide with the scale of the site, and to also help the development team increase their productivity through a building blocks codebase that was achieved through decoupling the tightly-coupled source code.

References:

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* (Second Edition). IT Revolution Press.