Caleb Stark, 2/9/2025, Module 6 Assignment 2, Strangler Pattern at Blackboard Learn

This case study is one that focused on a company that utilized a monolithic design to development but found that issues were arising with that choice so an alternative had to be found. When it comes to these case studies, up to now most saw the developers shifting the code base from Monolithic to being multiple separate services. This case study sees a similar approach but without the large overhead that we would see if you completely redid the monolithic code. Instead this approach saw the team using building blocks. These building blocks were various chunks of functionality that could be removed from the current code base and becomes its own independent API or service.

The reason for this approach is that the team was having more and more issues with committing code to the codebase, and when they did add code it was averaging out to more and more lines of code then prior. This could easily lead the company further into technical debt if left unaddressed and see deadlines be missed on releases for shareholders. By switching over to just taking out parts of the code not only was the size of the main codebase dropped but developers were able to make many more commits without having to worry as much about a bug causing issues within the program because that portion of the code would be separate from the codebase.

To me the most important message from this document is the idea that you do not need to go with the first option that comes to mind when deciding how you want to fix your company’s code base. Every company will be different so knowing what your company can handle and what it can not is very important. I am sure that in this case if the developers could have completely redone the code base they could probably make an even larger impact on the overall development cycle but that would have been expensive. By finding an option that allowed them to make expansions to the code base without continuing to make contributions to there technical debt is enough to me to be warranted. (Gene Kim, 2021)

# References

Gene Kim, J. H. (2021). *The DevOps Handbook, 2nd Edition.* IT Revolution Press.