Caleb Stark, 3/9/2025, Module 12 Assignment 12.2

Auditing, a key part of the development lifecycle and one that can be easily overlooked if rushed. We looked at two case studies that talk about and get into details about how auditing should be performed in a moder code environment. Within the first case study we talk about having the right tools and systems in place to be able to effectively audit changes before they are implemented along with after the fact as well. The second case study is a great example as to why you should have more then just code reviews and a separation of duties when working within software development.

Proving compliance in regulated environments talks about the current development environment and how ensuring compliance can be quite challenging if not planned for properly. In our cloud based age where there may be servers spread all across the country hosting various parts of the code base and data collection for a company just checking a small part of that will not give good results and be incredibly costly. As an alternative listed in this case study we can see the idea of having more centralized storage of logs, proof of compliance screenshots, tests and any other object that could be used to show that compliance is being met. This is just part of it though because there is also the planning that goes into staying compliant and determining what compliance really means. Every situation is different so some instances may be more lax then others but regardless of the severity you do need to have some plan and structure to follow or you may find yourself in trouble down the road when you realize you lace the necessary compliance and have to redo the code base.

Relying on production telemetry for ATM systems is all about having the right set of audits and logs run not only when in the building and deployment phases, but also when in the production environment. Unfortunately there may be some bad actors that attempt to insert bad code into the production environment hopefully that malicious code can be found in code reviews prior to pushing to production but unfortunately that is most likely not the case. In this case study we see some malicious code get past the audits and tests and end up in production, fortunately for this bank in this scenario they have other audits that track many different parts of there operation. One of those audits determined something was amiss and the team was able to quickly find the bad code and implement a fix, but what If they did not have those tools? This may just be a story told on some YouTube podcast about the programmer who walked away with millions. But instead we get this case study because of those audits.

The lesson we can gleam from both of these case studies is one of acknowledgement around the tools that we have available to us. Understand what auditing is good for and what extent you need for your business, to me a good plan here is to go overboard and be prepared for growth instead of trying to skimp on some testing and audits to save a few dollars. Of course that’s my thoughts and takeaway from these case studies.

# References

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

Sorokopud, O. (2024, March 13). *introduction to software development audits*. Retrieved from SoftJourn: https://softjourn.com/insights/introduction-to-software-development-audits