**Teodoir O’Ceallaigh**

**CS-405**

**Southern New Hampshire University**

**June 2022**

**Journal**

**Module 8-2**

**Adoption of a secure coding standard, and not leaving security to the end:**

It’s absolutely vital to ad security into every step of the SDLC because each portion involves creating a longterm solution for future iterations of the software/code. By setting a standard and frequently reviewing best practices you can be sure that the code being produced follows a set standard and when new best practices or policies become available the updating or iteration of the software(s) can be easily implemented with minimal chaos and strong structuring.

**Evaluation and assessment of risk and cost benefit of mitigation:**

Understanding how to evaluate risk and cost benefits of both actioning versus deprioritizing is a great way to not only understand the security policies and best practices that have ben implemented, but how certain coding can be adjusted in future cycles of the SDLC. While it may not make sense to update code right now for this specific portion, as a new feature is being created a potentially easy fix in this current process will make security tighter and reduce risk of the future iteration while allowing time for the previous iterations to be fixed without impact to P1 project timelines.

**Zero trust:**

Zero Trust sounds alarming and negative but the reality is that situations can arise that we may not have considered in the original development of the software. A disgruntled employee, a hacker with someone’s log in information, etc - these things can happen so utilizing a zero trust policy and having that 2FA or those extra steps, that DiD set up, allows for those potential issues to be caught well in advance without sacrificing the system or believing that our software is better than anything else that’s already been created and subsequently hacked or destructed in some way or another.

**Implementation and recommendations of security policies:**

The implementation of security policies and recommending updates to this as necessary is a great way to really drive the longevity of a company and the software it creates. No one wants to be the headline of the latest data breach scandal, no one wants to lose their job because the company needs a scape goat, and no one wants to be responsible for a large scale breech of potentially vital financial/medical/etc information that can be utilized to harm others. Implementing an iterating is something that should continue endlessly through the SDLC and even after the final code has been written off as “done”. There are always ways to improve, new tactics being created, new threats rising. When in doubt, put up those walls!