CS-499-11429  
Professor Brooke Goggin  
Mark Irwin  
9/11/2024

2-1 Journal: What Makes a Productive Code Review?

Code review at its core is the ability to look over code and ensure its quality while also keeping note of its potential issues. Breaking down the code to ensure it is functioning correctly while also potentially finding issues within the code. Ensuring the code is clear and concise is important and valuable to future developers who potentially work on or utilize the code. Code review is an extremely important practice for computer science professionals as it helps ensure the code is meeting all desired requirements while also allowing the ability to look back at the code and ensure there is no inconsistency within the code or potential security issues. Code review also helps other computer science professionals to understand your code, its intended goals, and its potential drawbacks. Some code review best practices that I read about include SMARTBEAR’s article named “Best Practices for Code Review” that I feel is crucial to include in a code review involve limiting how much code you review during a specific session. I feel reviewing too much code at one point could potentially defeat the purpose or hurt the benefits of a code review. Finally, fostering a positive code review culture is important to ensure a healthy coding environment and ensuring added confidence while coding. Code review should happen throughout the development process to ensure code is being properly maintained and the development team are all on the same page of understanding. Another benefit of code review is it allows others within a group to work together to fix potential problems that may not be initially noticed by the author of the code.

I have chosen to utilize Microsoft’s Clipchamp to record my code review. My approach to creating an outline/script for my code review for each of the three categories will involve making sure I fully understand the requirements of each category for the code review while also ensuring to utilize the review checklist throughout the process. Since I plan to utilize the same artifact for all three categories, making sure my understanding of the artifact is a priority. Software Design and Engineering, I plan to go through my code while utilizing the code checklist to ensure my understanding is strong before starting the code review. I plan to write key aspects of the code to utilize during the code review, my goal is to make sure to clearly explain the enhancement I am proposing. Algorithms and Data Structures is going to utilize a similar breakdown as my first category. Since it will be using the same artifact, the plan will be the same as category one, but I plan to focus more on how the code is structured towards algorithms and data structures and what the enhancement will entail. Databases, since my initial code does not actually support databases, I plan to talk about the enhancement and how I plan to work towards utilizing a database while also breaking down what benefits a database could have to my selected artifact.

References:

* *Best practices for peer code review*. smartbear.com. (n.d.). https://smartbear.com/learn/code-review/best-practices-for-peer-code-review/