Sunshine Mason

SNHU

CS 499 Computer Science Capstone

09/10/2025

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

**What is a code review?**

A code review is the examination of the source code, doing this examination helps support finding any mistakes in the code so they can be fixed after they were not found on the original first development of the code. Depending on how the code was created this can be reviewed either by yourself or even be reviewed by peers that were associated with building the code. Doing a code review is equivalent to having to write a paper and doing the initial first draft, once its reviewed and graded by a teacher it shows any mistakes that were made in the process. I believe peer review is important when it comes to writing code because you’ll get feedback from an extra set of eyes that you could have initially missed or not realized. This allows the developer to learn and continuously improve on any mistakes they miss, as well as build confidence from a peer review before the code is pushed through.

**Why is it an important practice for computer science professionals?**

Doing a code review is important because it allows the programmer to get feedback or allow them to self-critic themselves on what can be done better or differently. Doing a code review has been proven to be more effective before fully submitting the code. Doing a code review with your peers though as a team allows everyone to share their knowledge and builds the trust of all the other individuals on the team while being able to weigh the pros and cons of each person for future developments.

**What are some code review best practices that you read about in the resources that are crucial to include in a code review?**

Some best practices when it comes to code review consist of setting clear realistic goals, doing this gives you the ability to define what the review should achieve in the areas of bug detection, performance improvement, and making sure everything is compliant. Utilizing a checklist that way any common issues that can occur don’t get overlooked. Doing reviews in smaller capacities that way smaller code changes can happen over time to ensure the best quality. Providing feedback on all areas and being able to discuss any improvements. Lastly another best practice would be to use automation tools when doing the review that way any style issues, syntax errors, or any other simple problem areas can be checked internally so the review can be more focused on any major issues.

**What software have you chosen to use to record your code review?**

The software I’ve found the best to use for a code review would be loom. Loom can screen record and is the best option when it comes to a video-based code review. I personally like that you can record your screen and yourself at the same time so complex issues can be provided. Loom also allows you a sharable link as soon as you finish recording. I also like loom because it is collaborated GitHub which is where my workflow has already resided from prior courses.

**Describe your approach to creating an outline or writing a script for your code review for each of the three categories that you will be reviewing based on the rubic as well as the code review checklist.**

The approach I would be taking on creating a script for my code review would be to follow the best practices and the code quality. Doing this would allow me to address the code standards needed, the maintainability, design patterns, as well as readability of the code itself. Next would be to do the functionality, this allows me to verify the code would meet all the functional requirements while being able to produce the correct outputs. Lastly would be security and performance, reviewing this will evaluate any potential risks within the code for security and the overall performance of the code.