

Module 2 – Journal

Kenneth Lockhart

Southern New Hampshire University

CS-499-13459-M01 Computer Science Capstone

Gene Bryant

March 16, 2025

A code review is exactly what it sounds like a review of code. However, it is not that simple, it is more of an analysis of code. As in a code review is the process of taking a very in-depth look at the create code and truly understanding it while also determining improvements. This is important as a code review acts as a reflection of a computer science professional. The process of code review allows you to reflection on code you created and learn how you can improve yourself and help others to improve as well. Additionally, it helps you focus on the important aspects of computer science and how you as a professional can help develop those skills further. I think the best resource for code review best practices is to create a checklist of key principles much like our milestone assignment. This checklist is something that keeps the brain on the prize and helps to remove blind spots during the review process. As part of this it is also important to know when a code review is good for the development process. A code review should always help before pushing anything to mainstream is one way to look at it. For example, in my GitHub projects a code review always occurs before anything gets pushed to main. Main is the release version it expects to be held up to a high standard as thus it is important to ensure that the changes being made are solid, well thought out, and most importantly in line with expectations. However, code reviews should occur more often, a good spot to add a code review after the addition of any new features even if a push to main is not happening right away. This will result in less work overall and will keep the principles of computer science fresh in the minds of the dev team.

I think I conducted my code review in the most simplistic yet complicated way possible. I used two softwares, the first being OBS for screen and audio capture, and the second being DaVinci Resolve to cut down the 30 minutes' worth of "umms", desk tapping, cat meowing, etc. This meant almost double the work overall to complete the task of the code review, but I think it was worth it as it helps keep it professional and focused. Something else to note about this

approach I was able to edit in clips of stuff that was missing during the initial take or edit in images on top of the video. My outline approach was a bit odd because if I missed something it does not like I couldn't just add it in post. As a result, I basically made a bullet list of details for each category and for the most part winged the actual code checklist part. Well not really all the actual wording was completely on the spot, but the ideas and focus was on the mind and on paper well before that point, so I knew exactly what to think and talk about without even seeing the code itself. Basically, I had no pressure easy flow to go with I forgot something I added it, I didn't want something I removed it. It did lead some waste of time clips like the 60 second intro take with a cat meowing in the background, but the quality was worth it.