Amos Roland

Southern New Hampshire University

CS-499 Computer Science Capstone

Professor Akhil Gudivada

09 November 2025

CS 499 Module 2-1 Journal: What Makes a Productive Code Review?

Part 1:

1. What is code review?

A code review is the process of carefully examining computer code to find mistakes, improve structure, and ensure that it meets software and security standards. It is a way for developers to check each other's work to make sure the program runs correctly, follows requirements, and is easy to maintain. According to SmartBear (n.d.), code reviews are most effective when performed regularly with constructive feedback. Similarly, BrowserStack (Felice, 2025) emphasizes reviewing code before merging to catch logical and security issues early. In my experience as a student and as a service member, I see code review as teamwork. Just like double-checking equipment in the Navy, it helps make sure the final result is safe, accurate, and reliable before being used.

2. Why is it an important practice for computer science professionals?

Code review is important because it improves the quality of software and builds collaboration among developers. It helps to identify problems such as poor logic, performance

issues, and security risks before software is released. In my field of information security, reviewing code helps prevent issues like weak authentication, poor data handling, and other vulnerabilities. According to the OWASP Code Review Guide (OWASP, 2021), performing code reviews early in development helps discover vulnerabilities that could lead to major security problems later.

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

Some important best practices include reading the code line by line, checking for logical errors, ensuring consistency with coding standards, and verifying that comments and documentation are clear. The Best Practices for Code Review article by Atlassian (2023) explains that reviewers should focus on improving the code rather than criticizing the developer. Another best practice is to include security and performance checks in every review. A code review should happen after a feature or function is complete but before it is added to the main system. This timing allows the team to find mistakes early and make corrections without delaying deployment.

Part 2: Planning My Code Review

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

Now that I have reviewed what makes a code review effective, I will describe how I plan to conduct my own for this course.

For my code review videos, I will use ScreenPal because it allows me to record my screen and voice clearly while showing my code. I plan to record my work directly from Visual

Studio Code, Jupyter Notebook, and Android Studio, depending on the project I am reviewing. Each recording will be between five and ten minutes and will include my explanation of what the code does, what issues I found, and how I plan to fix them.

5. My approach to outlining my code reviews

I will follow the SNHU Code Review Checklist as my guide. Before recording, I will create an outline for each artifact to stay organized and to ensure I cover every part of the rubric.

1. Software Engineering and Design (Travlr Getaways Website)

- Review the overall architecture and structure.
- Check that the Model-View-Controller pattern is applied correctly.
- Look for logic errors, missing validations, and potential security issues.
- Plan to enhance it by integrating MongoDB, adding JWT authentication, and improving data validation.

2. Algorithms and Data Structures (Deep Q-Learning Cartpole)

- Review how the neural network uses experience replay and reward functions.
- Identify inefficiencies in the training loop and review the algorithm's stability.
- Plan to enhance it with Double DQN, prioritized experience replay, and improved parameter tuning for accuracy and learning speed.

3. **Databases (Android Weight-Tracking App)**
    - Review the SQLite database schema and query structure.
    - Check if the app properly validates and encrypts stored user data.
    - Plan to improve security by migrating to Firebase, using user authentication, and encrypting sensitive data.

This structured approach will help me identify problems early, especially those related to security and data handling. Code review is not just about improving performance; it is also about ensuring safety and reliability. Following this checklist will allow me to refine my artifacts and demonstrate strong problem-solving skills in my ePortfolio.

References

Atlassian. (n.d.). *What are code reviews and how they actually save time*. Atlassian. Retrieved November 5, 2025, from https://www.atlassian.com/agile/software-development/code-reviews

Best Practices for Peer Code Review. (n.d.). SmartBear. Retrieved November 5, 2025, from https://smartbear.com/learn/code-review/best-practices-for-peer-code-review/

Felice, S. (2025, March 25). What is Code Review? *BrowserStack*.

https://www.browserstack.com/guide/what-is-code-review