William Colon

2-1

What Makes a Productive Code Review

**What is code review?**

Code review is the orderly examination of source code by one or more designers to distinguish mistakes, progress code quality, guarantee compliance with coding measures, and encourage information sharing among group individuals.

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

Code review is critical for computer science experts. It aids in the detection and resolution of issues such as poorly structured "spaghetti" code and functional flaws. In addition, the review.

**What are some code review best practices that you read about in the resources that are crucial to include in a code review? Include when a code review should occur in the development process with a rationale as to why.**

Establishing a precise definition of "done," promoting constructive criticism, concentrating on the code rather than the coder, and using checklists to maintain consistency are some essential code review best practices. In order to spot issues early, encourage team members to exchange information, and maintain high code quality—all of which contribute to a more manageable codebase—code reviews should ideally take place after a feature is finished and before merging into the main branch.

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

CS-320 Appointment originally coded in Java. changing a codebase from Java to C++ can offer several benefits, including improved performance due to C++'s closer-to-hardware memory management and optimized execution speed. It can also allow for greater control over system resources and low-level programming, which is beneficial for applications requiring high efficiency. Additionally, C++ provides more flexibility with object-oriented features and templates, enabling more complex data structures and algorithms. This transition can also enhance understanding of different programming paradigms and deepen skills in both languages, which is valuable for a computer science curriculum.

**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 rubric as well as the code review checklist.**

My approach to making a layout or script for a code survey includes three primary categories, readability, and performance.

1. Functionality:

- Start by summarizing the purpose of the code and its intended functionality.

- Outline specific features or requirements that the code should meet based on project specifications.

- Include test cases or scenarios to validate if the code behaves as expected and adequately handles edge cases.

2. Readability:

- Evaluate the code structure, including naming conventions, use of comments, and overall organization.

- Discuss the clarity of logic and flow, ensuring that the code is easy to follow for other developers.

- Highlight any areas where additional documentation might be beneficial for future maintainers.

3. Performance:

- Analyze the algorithmic efficiency, noting any possible bottlenecks or areas for optimization.

- Review resource management, including memory usage and processing time, especially in loops or recursive functions.

- Suggest best practices for scalability and maintainability, reflecting on how the code could perform under increased loads or varied inputs.

I will use the code review checklist to ensure a comprehensive evaluation of each category and facilitate a structured and productive discussion during the review meeting.