Nathan Kintu

Southern New Hampshire University

March 16, 2024

1. **What is code review?**

Code review is a process where code written by one person is systematically examined by another person or a group of people to find and fix mistakes overlooked in the initial development phase, improve the overall quality of the code, and ensure that it aligns with coding standards and best practices.

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

Code review is a crucial practice for computer science professionals as it enables early detection of bugs and errors, fosters knowledge sharing among team members, enhances code quality by adhering to standards and best practices, promotes consistency in coding style, encourages collaboration and communication within teams, facilitates skill development through feedback and suggestions, and instills a sense of code ownership and accountability among developers.

1. **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.**

Code review best practices include ensuring reviews are conducted by multiple reviewers to catch diverse issues, providing constructive feedback focusing on the code and not the individual, setting clear objectives for the review, using tools to aid in the process, and following a checklist to cover all aspects. Code reviews should occur before merging code into the main branch in the development process to catch issues early, reduce the impact of bugs on other team members, and maintain code quality throughout the project, ultimately resulting in better software reliability and maintainability.

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

I have chosen to utilize GitHub for recording my code reviews. GitHub provides a robust platform for hosting repositories, managing code changes, and facilitating collaborative code review processes. With features such as pull requests, inline commenting, and code diff viewing, GitHub streamlines the code review process, enhances team collaboration, and enables easy tracking of feedback and changes. Additionally, GitHub's integration with version control systems like Git ensures transparency, accountability, and version history tracking, making it a preferred choice for efficient and effective code reviews.

1. **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.**

For software design, I will begin by evaluating the overall architecture, ensuring it adheres to design principles such as modularity, scalability, and maintainability. I check for the presence of design patterns, clean separation of concerns, and appropriate use of interfaces and abstractions. In terms of database management, I focus on the efficiency and correctness of database queries, normalization of data structures, and proper indexing strategies for performance optimization. I also assess data integrity constraints, error handling mechanisms, and security measures to protect against vulnerabilities. When reviewing Algorithms and Data Structures, I analyze the complexity of algorithms, their correctness, and their suitability for the problem at hand. I pay close attention to time and space complexity, algorithmic paradigms used, and consideration of edge cases. My approach combines a thorough understanding of the specific requirements outlined in the rubric with a comprehensive code review checklist tailored to each category, ensuring a systematic and detailed evaluation of the codebase.