Joe Clancy

11/4/2024

CS499

2-1 Journal: What Makes a Productive Code Review

**Part 1**

Code review is a process in software development where peers examine and evaluate each other's code to identify potential issues, ensure adherence to coding standards, and improve the overall quality of the software (SmartBear Software, n.d.). This practice involves checking for logical errors, verifying functionality, and assessing how well the code aligns with best practices and project requirements.

Code reviews help identify issues early, improving the overall quality of the codebase. Through feedback from peers, developers learn from one another, which encourages a collective understanding of best practices. Reviewing code can also help uncover vulnerabilities that could lead to security breaches (OWASP Foundation, 2017), find and fix bugs early in the development process, and facilitate knowledge sharing across the team.

Commonly mentioned best practices for code reviews include limiting the scope of the review, both in terms of the number of lines reviewed and the duration of the review period. Recommendations include keeping reviews under 400 lines of code and no longer than 60 minutes to help maintain focus and effectiveness (SmartBear Software, n.d.). Other recommendations include using checklists to ensure the process is both rigorous and consistent and employing automated tools, when possible, to help identify issues with formatting, syntax, or other basic issues to maximize the effectiveness of the reviewer’s time.

Code reviews should occur after a developer completes a feature or a logical segment of code but before it’s merged into the main codebase. This allows issues to be identified before they are integrated into the larger project, reducing the likelihood that they will cause larger or more complex issues down the line.

**Part 2**

To record my code review, I have chosen to use the recommended ScreenPal software. In order to create an outline for my code review, I began by reviewing the rubric for the assignment. Then, for each category, I prepared a brief introduction explaining what the code is and how it is meant to work. Next, I listed out items from the checklist to review in the code for each of the categories. Finally, I prepared an overview of the enhancements I plan to make to the code and how they will integrate into the existing code. While I organized key points I want to address for the overall review and for each category, I tried to leave plenty of space to analyze the code organically as I reviewed it. Ideally, I want my code review to support the following course outcomes:

2. Design, develop, and deliver professional-quality oral, written, and visual communications that are coherent, technically sound, and appropriately adapted to specific audiences and contexts

4. Demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry- specific goals

**References**

OWASP Foundation. (2017). *OWASP code review guide* (v2.0). OWASP. Retrieved November 4, 2024, from <https://owasp.org/www-project-code-review-guide/assets/OWASP_Code_Review_Guide_v2.pdf>

SmartBear Software. (n.d.). *Best practices for peer code review*. SmartBear. Retrieved November 4, 2024, from <https://smartbear.com/learn/code-review/best-practices-for-peer-code-review/>

SmartBear Software. (n.d.). *What is code review?* SmartBear. Retrieved November 4, 2024, from <https://smartbear.com/learn/code-review/what-is-code-review/>