Abigail Sibazeu

CS-499

01/19/2025

Journal Entry: What Makes a Productive Code Review?

**Part One-**

Code review is a systematic examination of code written by developers to identify bugs, improve code quality, and ensure adherence to best practices before it is merged into a project. It serves as a critical quality assurance step in software development.

For computer science professionals, code review is essential because it enhances code maintainability, improves team collaboration, and helps catch potential issues early in the development process. It also fosters knowledge sharing among team members and ensures that code follows industry standards and security guidelines.

Some best practices for code reviews include keeping reviews concise and focused, providing constructive feedback, and ensuring that the code adheres to coding standards. Reviews should occur frequently, ideally before merging code into the main branch, to prevent technical debt and reduce the cost of fixing issues later in the development cycle. Code reviews should be integrated into the development workflow through pull requests and automated tools to streamline the process.

**Part Two:**

For recording my code review, I have chosen to use **Bandicam and Capcut** due to their ease of use and ability to capture both video and audio effectively.

To ensure a structured code review, I will create an outline focusing on three key categories from the rubric:

1. **Code Functionality and Logic:** I will check whether the code meets the expected requirements, handles edge cases properly, and avoids unnecessary complexity.
2. **Code Readability and Maintainability:** I will assess if the code is well-documented, follows consistent naming conventions, and is easy to understand for future developers.
3. **Efficiency and Optimization:** I will review whether the code is optimized for performance, avoids redundancy, and follows best practices for resource management.

To prepare, I will draft a script to keep my review concise and ensure that I provide clear and actionable feedback. My approach will be to highlight strengths, pinpoint areas for improvement, and suggest solutions where necessary.

By following these steps, I aim to conduct a thorough and effective code review that aligns with industry best practices.

**References:**

1. Bosu, A., Corley, C. S., Heaton, D., Chatterji, D., Carver, J. C., & Kraft, N. A. (2017). Building reputation in StackOverflow: An empirical investigation. *Empirical Software Engineering, 22*(4), 1536–1573. https://doi.org/10.1007/s10664-016-9453-9
2. Rigby, P. C., & Bird, C. (2013). Convergent software peer review practices. *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*, 202–212. https://doi.org/10.1145/2491411.2491444
3. Bacchelli, A., & Bird, C. (2013). Expectations, outcomes, and challenges of modern code review. *Proceedings of the 2013 International Conference on Software Engineering*, 712–721. https://doi.org/10.1109/ICSE.2013.6606617
4. SmartBear. (2021). *Best practices for code review*. SmartBear Software. https://smartbear.com/learn/code-review/best-practices-for-peer-code-review/