Journal Entry #1

Zaffar Shiekh

CS-499-10941-M01 Computer Science Capstone 2024 C-4

Federico Bermudez

07/13/2024

**Part 1:**

1. **What is code review?**

Code review is a process where developers check each other's code for mistakes, adherence to coding standards, and overall quality before it is integrated into the main project. It's a collaborative practice where team members review the code, provide feedback, and suggest improvements (Rigby & Bird, 2013).

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

Code review is crucial because it helps ensure code quality, find bugs early, and share knowledge among team members. This practice leads to better code maintainability, increased collaboration, and higher software quality. It also helps developers learn from each other and improve their coding skills (McIntosh, Kamei, Adams, & Hassan, 2016).

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.**

Some best practices for code reviews include:

* 1. **Keep Reviews Small:** Smaller code changes are more accessible and manageable to review thoroughly.
  2. **Review Regularly:** Regular code reviews help catch issues early and integrate feedback continuously.
  3. **Be Respectful and Constructive:** Provide feedback that is helpful and supportive.
  4. **Automate Where Possible:** Automate tools to catch basic issues before the manual review (Rigby & Bird, 2013).

Code reviews should occur after the code is written but before it is merged into the main codebase. This timing ensures that the code is reviewed for quality and correctness before becoming part of the leading project, preventing potential issues from propagating (Bacchelli & Bird, 2013).

**Part 2:**

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

I have chosen to use OBS Studio to record my code review. It is free and open-source software for video recording and live streaming, and it will allow me to capture my screen and audio as I review the code.

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 my code review, I will structure my script as follows:

* 1. **Introduction:**
     1. Briefly introduce the code and its purpose.
     2. Outline what will be covered in the review.
  2. **Software Engineering and Design:**
     1. Review the code structure and organization.
     2. Check for adherence to design principles and patterns.
     3. Ensure that the code is modular and maintainable.
  3. **Algorithms and Data Structures:**
     1. Evaluate the efficiency of the algorithms used.
     2. Ensure the appropriate use of data structures.
     3. Check for optimization opportunities and complexity management.
  4. **Databases:**
     1. Review database interactions and queries.
     2. Check for proper indexing and query optimization.
     3. Ensure data integrity and security measures are in place.
  5. **Conclusion:**
     1. Summarize the findings and provide recommendations.
     2. Highlight any areas for improvement and next steps.

I will also use a code review checklist to ensure that I cover all critical aspects and provide a comprehensive review.

**References**

Bacchelli, A., & Bird, C. (2013). Expectations, outcomes, and challenges of modern code review. In Proceedings of the 35th International Conference on Software Engineering (ICSE) (pp. 712-721). IEEE.

McIntosh, S., Kamei, Y., Adams, B., & Hassan, A. E. (2016). The impact of code review coverage and code review participation on software quality: A case study of the Qt, VTK, and ITK projects. In Proceedings of the 11th Working Conference on Mining Software Repositories (MSR) (pp. 192-201). ACM.

Rigby, P. C., & Bird, C. (2013). Convergent contemporary software peer review practices. In Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering (ESEC/FSE) (pp. 202-212). ACM.