Class: CS499 Computer Science Capstone

Professor: Ramsey Kraya, PhD

Student: Anna Fisher

Date: 1/19/2025

**Part 1:**

**What is code review?**

During a code review, a developer submits their code to be reviewed and analyzed by a designated reviewer or team lead. A code review is an important quality assurance step in the software development life cycle and has multiple purposes. The primary purpose is to find and fix code errors prior to them making it into production. It’s much easier to fix a coding error in a lower testing environment than it is to fix production accounts that have been created or negatively impacted by bad code. The second purpose of a code review is to ensure the code design accurately and efficiently meets the requirements. If the customer has ordered or requested a code change to add functionality that is not included within the code, then it must be amended to meet the requirements gathered and approved by the customer. The third purpose of a code review is to ensure the code design is efficient and consistent with the protocols and existing codebase. The code should not have inefficient design that negatively impacts performance or unnecessarily increases database storage. The code should conform to the existing rules in place, and best practices for items such as readability, inline comments, security compliance, and preferred methods for naming conventions.

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

Code reviews have several important benefits. The most obvious benefit of a code review is assurance of code quality. Preventing code issues or errors before they make it to production saves time and money, and it also maintains and protects the reputation of the development team. If the user experience system errors and issues every time there is a production deploy, then there will be serious concerns and mistrust of the development and quality assurance team. Another important benefit of code reviews is knowledge transfer and training. Developers can learn a great deal from each other, we all have different strengths and weaknesses. Code reviews are a great way to communicate and learn from a developer with more knowledge and experience and share important coding practices, and occasionally better solutions.

**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.**

In the resources, I read that is it important to restrict your review of code to less than 400 lines of code at a time. Our brains can become overwhelmed when reviewing too many lines of code at a time, and this reduces the effectiveness of defect detection. I also read that it is important to take your time and be thorough with a code review. We all have time constraints and feel pressured to complete more tasks, but speeding through a code review increases the potential to miss defects. It’s also important to keep a checklist to effectively eliminate frequent errors and communicate expectations to the team. It’s important to make code reviews a positive experience and ensure team members continue to view this process as a way to gain and transfer knowledge. A code review should occur after a developer has completed their code and unit testing, but before the code is merged into the main or common codebase. It is vital to catch potential code errors prior to merging to avoid any negatively impacting the common codebase or other developers.

**Part 2:**

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

I have selected ScreenPal to record my code review.

**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 prepare myself for the recorded code review is to thoroughly review the artifacts, the three planned categories, and the rubric requirements. I will make a checklist of all the important points and items of discussion for each of the three categories.

* Current format and code functionality – I will discuss my existing code design features and functionality.
* Analysis of the existing code – I will discuss current design flaws, inefficiencies, and system vulnerabilities.
* Planned enhancements – I will discuss the changes I plan to include.

**References**

*Best practices for peer code review*. smartbear.com. (retrieved January 2025). Retrieved from: https://smartbear.com/learn/code-review/best-practices-for-peer-code-review/