**What Makes a Productive Code Review?**

Anderson Forestal

Science department, Computer Science

CS-499-T4250 Computer Science Capstone

Southern New Hampshire University

Brooke Goggin

March 12, 2023

What makes a productive code review?

As the name implies, code review is an essential best practice for computer scientists and others in the industry simply because it maximizes the value of innovative perspectives by identifying poor design patterns, coding styles, and bugs from polluting the source code. Incorporating code reviews in the development lifecycle enhances code quality, ensuring new features merge to source code, and are created using high-quality code. Moreover, it helps the development lifecycle be more proactive by delivering high-quality code instead of resolving defects at the end.

**What is code review?**

Code review is an excellent approach to discovering bugs from modifying branches. It is a process for a developer to analyze a team member’s code by identifying bugs, logic errors, style, test cases, consistency, performance, and unreachable call from breaking the source code. However, code review can be tedious, but incorporating it into the development lifecycle can result in positive outcomes. This approach helps a team to engage and deliver the best quality codes that adhere to code best practices.

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

For a computer science professional, code review is an important best practice because it helps developers to develop the skills required to analyze other's code to find errors and provide feedback to fix the problem early on the development lifecycle before reaching the source code. It is an opportunity for growth, not criticism. Therefore, it is imperative for computer science professionals to have this skill under their belt to add value to their portfolios. Moreover, a code review is the most used procedure for validating the design and implementation of features. It helps developers maintain consistency between design and implementation across many team members and between various projects on which the team works.

**When and how does it occur?**

A code review occurs when a developer develops codes for a new program, implements an existing code, or adds new features or branches for instance, the change must go under review by an experienced developer to ensure the features are not affecting the code performance, and if so, provide feedback by suggesting a fix and explain the reason the team member should make the change. Usually, a code is done by using collaborative distributed tools that enable individuals from across the organization to work as one unified project team, regardless of which department they’re in.

**What are some code review best practices that you would advocate?**

To prevent hindering collaboration and stalling overall progress, code review is a process that forces developers to adhere to a team's coding best practices throughout the development lifecycle. As team members continue to add new features to the project, the team coding best practices approach standardizes the source code by making it convenient for individuals across the team to maintain and understand the code.

Some of the code best practices that I would advocate are:

1. Always follow the code review checklist before approving implementation in the codebase.
2. Provide feedback on what needs to be improved and why the change is necessary.
3. Feedback is an opportunity for growth, do not take it personally.
4. Review others' code like you are reviewing your code.

References

Oliveira, D “Recommending code understandability improvements based on code reviews” Nov 2021

Mantyla, M. Lassenius, C. "What type of defects are discovered in code reviews?" 2009.

Dario, M. “Code review: What is it and why it important?” Sept 2022.

<http://thinkapps.com/blog/development/what-is-code-review/>

Usman, G. “Five code review best practices” Feb 3, 2022.

<https://www.atlassian.com/blog/add-ons/code-review-best-practices>