Joseph Silva Jr.

11/07/2021

SNHU

CS 499 – Journal 2-1: Productive Code review

* **What is code review?**

“A code review is a systematic examination of computer source code (1)”. The examination is supposed to find any mistakes in the computer code in order to fix these mistakes that were missed during the first development. The code can be reviewed by peers or by one’s own self depending on the situation of the app’s creation. The code review is like a student writing a first draft of a paper and then allowing the teacher to check the paper for any mistakes missed by the creator of the paper. I believe it is always best to allow a peer to review one’s computer code because a peer will more than likely catch the mistakes. When a peer reviews computer code, it is best to always get feedback from the peer because it allows a programmer to either better their skills by fixing the mistakes that were pointed out or it gives the programmer confidence in the work they are sending through before the code gets pushed to the main branch.

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

Code reviews are important because they allow programmers to either self-critic themselves or get feedback from their peers as I previously stated. Also, code review has been shown to be approximately 55% to 60% effective when the computer code is pushed to the main branch, while only testing the code can range from 25% to 45% effective according to “Code Complete” (1). Code review in a team setting is an important practice because it allows team members to share knowledge with each other and it also builds trust between members of a team.

* **When and how does it occur?**

According to *Code review guidelines* (2) and my own experience during my time at SNHU, code reviews should occur after the code has been self-evaluated by the programmer and automated testing by an IDE, but before the computer code gets pushed into the main branch. Especially in a team setting, it is important to make sure there are no mistakes or errors in the computer code during this time because many members of the team can be working or using this updated computer code in their local files by retrieving the code from the main branch. The self-review, testing, and peer review is how code review occurs before it is pushed.

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

I learned the best practices when conducting code review for a peer is to give honest feedback but be professional with the feedback. Another best practice is to review sections or blocks of code instead of the entire computer code because it allows the reviewer to concentrate on a section of code to find the mistakes or errors and it will be less likely in missing a mistake while looking at a smaller area. When I did code reviews for one of my classes, I broke the code down into sections for my classmates and found small mistakes within those sections. I then wrote comments about those sections for my peers to fix. Overtime, my classmates and I were able to fix large problems easier and quicker because we broke it down into smaller problems to be handled at a faster pace. Another practice which is important especially when if someone does not have a peer to review the code is to test blocks of code often. When I wrote my code, I always tested by code after completing a section of code to make sure each block of code worked hand in hand with each other. These are just a few of the practices that are important when conducting a code review.

**Optional: Are there any practices that may be currently uncommon that you believe would make code reviews more effective?**

I believe an uncommon practice in code reviews is outsource review. A programmer wanting to develop their skills in code review should help out the programming community by being available to review another’s code on sites like reddit and/or stackoverflow. Also, a programmer could also allow other outsource programmers on these sights to conduct review on the programmer’s work. These outsource reviews will let a programmer develop his own skills by being able to review another’s work, but it will also allow a programmer to receive good feedback from a programmer not connected to the project.

**CITATION:**

1. Ludovicianul. (2013, January 8). *Code review guidelines*. CodeProject. Retrieved November 6, 2021, from <https://www.codeproject.com/Articles/524235/Codeplusreviewplusguidelines>.
2. Palantir. (2020, November 21). *Code review best practices*. Medium. Retrieved November 6, 2021, from https://blog.palantir.com/code-review-best-practices-19e02780015f.