Jessica LeDoux

CS-499 Computer Science Capstone

09/10/25

Module 2-1

Journal: What Makes a Productive Code Review?

Part 1:

1. What is code review?
   * They are the systematic assessment of code, usually by a peer, that act as quality assurance to find bugs, security flaws, ensure use of best coding practices, increase sustainability, to name a few aspects it can address.
2. Why is it an important practice for computer science professionals?
   * It provides the foundation to have a consistent way of producing code that by the time it reaches the conclusion or finish, it is polished and stable for whatever purpose it was completed for. It gives a buffer to make sure that all code goes through two sets of eyes before it reaches the hands of the client.
3. 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.
   * One practice that I think is super important is to take your time. I think like most things in life we try and go through things as fast as we can because we have so much to do and get done. Not to mention the fact that most of us trust our coworkers and think that what they are doing probably won’t have any issues. According to the data, going faster than 500 lines of code (LOC) an hour sees a significant drop in quality of the review. So, ensuring that the appropriate amount of time is given to the code review is extremely important, or the benefit of the review is wasted or diminished.
   * I know that all 10 practices are important individually, but it seems like the main idea for all of them is to take your time, limit the amount that you work on in one go, as well as the time, and have a consistent system set up. Overall, it should be a positive and sometimes even light experience of just a quick glance over the shoulder of a section or sending a quick email asking for a specific part to be looked at. It doesn’t always have to be a formal event and can be added in to aspects to try and aid along the way.
   * Code reviews can happen a different moments, they can be full formal events when an entire project part is finished and needs to be added to an existing code system or as stated above the can be more informal and be a email exchanges or even use tool assistant reviews but need to make sure to not only rely on these as the only review as the tool is not an actually code review. Using the best method will depend on the needs of the users and what the variables are for the situation at hand and what the requirements are but using code reviews will provide a greater chance of producing a clean, stable, and reliable code.

Part 2:

1. What software have you chosen to use to record your code review?
   * I started to try and mess with ScreenPal and use it as my software to record my code review
2. 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.
   * Write out the main points of what the code does (Keeping it to what is essential)
   * Work through the checklist and recognize and comment on any flaws I see
   * Even if not technically wrong make suggestions of improvement areas (efficiency, comments, or security)
   * Adrees the enhancements that I want to address and the skills that I think will show and declare and explain the outcomes that they support.

References

GitLab. (n.d.). *What is a code review?* about.gitlab.com. https://about.gitlab.com/topics/version-control/what-is-code-review/