Mark McCormack

CS 499

Module 2 Journal: Code Review

Code review is a software development practice where the source code changes made by a developer are examined by a colleague or team lead before the changes are integrated into the main codebase. This process is intended to identify bugs, ensure consistency, improve the quality of the software, and enhance the skills of developers through collaborative critique. Code review is important for establishing quality assurance, consistency, mentorship, knowledge sharing, and security. Best practices for code review include timeliness, scope and size, using a checklist, automation, and integration with testing. The code review should be conducted soon after a code change is proposed but before it is merged into the main branch. Code reviews should not exceed 400 lines of code at a time or 1 hour at a time. Commonly used automation for code review are for mundane tasks like code formatting and linting. My favorite thing about code review is actually a psychological result of code review just being in place, and that is the subconscious implication. If an engineer knows for a fact that one of his peers or superiors will be conducting a code review on his merge request, he is more likely to carefully double check his own work before creating that merge request, resulting in higher quality code and consistency.

I’m going to use the ScreenPal tool to capture my narrated code review. My approach to creating an outline is to examine each of the bullets on the code review checklist, then basically answer the question of if and how it relates to the existing code for each of the 3 categories, as well as if our planned enhancements may address some of these items on the code review checklist..