Journal: What Makes a Productive Code Review?

By Zach Chantara

Part 1:

Code review is the process of evaluating and assessing code written by another developer to ensure it meets the quality, security, and functionality standards before merged into the main codebase. It is important because it helps identify bugs early, improve code quality, increase security, and promote shared knowledge across teams. According to GitLab reviews should happen after a developer completes their code before it is merged into a feature or main branch to prevent unstable or vulnerable code from being released and to allow fresh insight while the code is still top of mind (GitLab, n.d.). Best practices include keeping reviews small, focusing on one area at a time, using a checklist, and reviewing for readability, security, efficiency, and adherence to coding standards.

Part 2:

For my code review, I will break it down by category and use the appropriate development environments. I will use Visual Studio for Category 1 (Software Design and Engineering), Github for Category 2 (Algorithms and Data Structures), and Android Studio for Category 3 (Databases).

In Software Design and Engineering, I’ll describe what the code does, then analyze structure, logic, and security using the checklist to highlight needed improvements, such as better error handling or documentation. In Algorithms and Data Structures, I’ll evaluate the logic and efficiency of the code, targeting opportunities to optimize or improve security. In Databases, I’ll assess how data is stored and accessed, checking for SQL injection vulnerabilities, normalization issues, or performance bottlenecks. Each section will include practical enhancements and note the skills demonstrated in alignment with the course outcomes.

Reference:

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