Mihir Patel

CS 499

July 10, 2025

Prof. Aly

**What Makes a Productive Code Review**

Code review is the process of systematically going over source code that has been created by oneself or by others in order to find errors, make it easier to read, make sure that coding standards are being followed, and improve the overall quality of the software. Prior to the code being merged or deployed, it serves as a quality checkpoint. coding reviews are crucial for computer science professionals because they promote teamwork, lower technical debt, and enable teams to sustain consistent coding practices across time. They also facilitate the exchange of knowledge and the early detection of logical or design flaws in the development cycle.  
 The resources' best practices, which I thought were crucial, include using checklists to ensure consistency, giving constructive criticism, keeping evaluations brief and targeted, and ensuring that they are clear and maintainable. Code reviews should ideally happen after a developer completes a working module before merging it into the main branch. Doing this early ensures that potential issues are caught before they grow into larger bugs during integration or production stages.  
 I intend to record my screen and voice using OBS Studio for my code review. It is portable, user-friendly, and produces quality video. In the video, I'll demonstrate my Python, SQL, and Snowflake code artifacts.  
 I'm writing down the main points of discussion for databases, algorithms, and software engineering in order to get ready. I'll describe my modular design, logging, and error handling in terms of software engineering. In terms of algorithms, I'll describe the reasoning behind churn risk score and how I used NumPy to optimize feature generation. Regarding databases, I'll go over my SQL views, Snowflake schema, and Power BI data structure. To make sure I'm covering maintainability, functionality, efficiency, and clarity in each section, I'll also consult the code review checklist.