Code Review Journal

Tuyet Tran

11/09/2024

Code review, or peer review, is an essential part of the development process where other developers review code to ensure it meets quality and standards. Think of it like how a writer relies on editors and reviewers to catch mistakes before publishing—it helps avoid errors that could be embarrassing or costly. In organizations, this process is usually carried out by team members, senior developers, or managers within the department.

Code review plays a vital role in the professional growth of computer science practitioners. It’s not just about fixing mistakes—it’s an opportunity to learn from others’ coding styles and perspectives, which can help improve our own skills. Additionally, it sharpens our ability to identify errors, both in our own work and in others’. Beyond individual growth, code review fosters teamwork by encouraging communication and collaboration, ultimately improving the code and ensuring it’s the best version of itself. High-quality code has lasting benefits, providing a strong foundation for future improvements, scaling, and troubleshooting. On the flip side, poorly written code can harm a company’s reputation and waste valuable resources.

In my research on code review best practices, I found that setting clear goals and tracking metrics are particularly important. These practices keep the review process focused and productive. While I see value in all best practices, I believe having clear objectives and measurable outcomes is critical to staying on track. I also think code reviews should be an ongoing process throughout development. Catching mistakes early makes them easier and less costly to fix. However, there’s also a risk of becoming too familiar with the code and overlooking issues, so the timing of reviews should align with the team size and project scope. Regardless, every code review should happen before merging changes into the main codebase.

For my code review recording, I’ll use screen capture software to walk through my process. I’ll start by explaining the goal and purpose of the video, followed by outlining the checklist and the key points I’ll be targeting during the review. The flow of the video will cover three main areas: Software Design and Engineering, Algorithms and Data Structures, and finally, the Database. My aim is to provide a clear, structured review process that demonstrates best practices and thoughtful analysis of the code.