2-1 Journal: What Makes a Productive Code Review?

Victoria Franklin

Southern New Hampshire University

CS-499 Computer Science Capstone

Professor Gene Bryant

Sunday, July 14, 2024

2-1 Journal

What is code review?

During a code review, one or more developers check another developer's work for specific quality requirements. Code reviews aim to find mistakes, possible defects, or issues the original developer could have overlooked. Raise the bar for code quality by checking that it is efficient, clean, and adheres to coding standards and best practices. Make it easier for developers to pick each other's brains and share what they know about the codebase and best practices; this will improve learning and knowledge. Identifying potentially exploitable security holes and vulnerabilities is essential in improving security. Ensure the code follows the project's style rules and standards to keep things consistent.

Code reviews significantly enhance software quality, a result that all developers can take pride in. An example of an over-the-shoulder code review would be having one developer look at the code while the author provides context. You can send the code to someone for review via email or another comparable platform. When two programmers sit at the same desk and critique each other's code as they go, they are engaging in pair programming. Code review systems such as Crucible, Review Board, and Phabricator allow for asynchronous comments, approvals, and reviews. These systems benefit distributed teams or teams working across different time zones, enabling developers to review code conveniently. Examples of such technologies include GitHub, GitLab, and Bitbucket (Radigan, 2024). A more informed development team, enhanced cooperation, and excellent software quality are all outcomes of well-executed code reviews, a practice that should be a source of pride and motivation for all involved.

Why is it an important practice for computer science professionals?

There are various reasons why code review is a crucial practice for computer science experts. It verifies that the code is error-free and follows all coding standards. Code that is easier to maintain and cleaner because of this. Having several people review the code helps find and correct problems before they reach production, which lowers the likelihood of defects. Developers can learn from each other and share their knowledge through code reviews. As a result, the team's expertise and abilities are enhanced through the distribution of skills. As a result, team members are more likely to work together and share ideas freely. The developers learn more about the codebase as a whole and how their work fits into it.

As a result, team members are more likely to work together and share ideas freely. The developers learn more about the codebase as a whole and how their work fits into it. Codebase consistency makes it easier to comprehend and manage by ensuring that coding standards and project-specific criteria are constantly followed. Building technical debt can hinder future progress and increase maintenance costs; identifying faults early reduces this. New team members can learn the ropes faster by participating in code reviews. As a result, they gain familiarity with the codebase and the norms and practices of the team's development. When developers know their code will be evaluated, they are more likely to generate high-quality code since their peers will scrutinize their work. When developers see that their code has been peer-reviewed and tested, they gain confidence in deploying it. (Keary et al., 2017).

What code review recommended practices do you read in critical code review resources?

SmartBear's article details some essential tactics for conducting peer code review. These include annotating code, using checklists to catch common errors, establishing a process for

fixing defects, reviewing no more than 400 lines of code at a time, limiting reviews to under 60 minutes, setting specific goals and tracking metrics, and encouraging lightweight, tool-assisted reviews. However, one of the most crucial aspects of code review is its role in fostering a positive review culture. This involves creating an environment where feedback is constructive, respectful, and focused on improving the code and the developer's skills. A positive review culture significantly impacts the effectiveness and outcomes of code reviews, leading to higher-quality software and a more cohesive development team. It also makes each team member feel valued and part of a supportive community, enhancing their sense of belonging and motivation. (*Best Practices for Peer Code Review*, 2024).

Code review software: what do you use? Explain your three-category code review blueprint or script using the criteria and checklist.

Microsoft's Screen Recorder for Windows 11 will record my code review, which I'll narrate. I outline by reviewing each item on the code review checklist and answering whether it applies to the current code in each of the three categories. I also make sure to mention whether any of the planned improvements could solve these issues.

References

Best Practices for Peer Code Review. (2024). smartbear.com. Retrieved July 14, 2024, from

    https://smartbear.com/learn/code-review/best-practices-for-peer-code-review/

Keary, E., Conklin, L., Robinson, G., Curiel, J., Mennouchi, I. A., Naderi, A., Pantelides, C.,

    Hidalgo, M., Chapman, J., Van Der Stock, A., Perego, P., Lowry, D., Rook, D., Cruz, D.,

    & Williams, J. (2017). *OWASP Code Review Guide*. https://owasp.org/www-pdf-

    archive/OWASP_Code_Review_Guide_v2.pdf

Muezza, & Muezza. (2024, January 25). *Tips for Effective Code Review: Enhance*

    Collaboration. Boydmillerwebdesign - Technology.

    https://boydmillerwebdesign.com/tips-for-effective-code-review-enhance-collaboration-

    and-code-quality.html

Radigan, D. (2024). *Why code reviews matter (and actually save time!)*. Atlassian. Retrieved

    July 14, 2024, from https://www.atlassian.com/agile/software-development/code-reviews