Journal Code Review

Hai Nguyen

CS 499 Computer Science Capstone

Professor Brook Goggin

31 Oct 2023

## Code Review:

The article “10 Common Software Security Design Flaws” of Kelly Jackson Higgins, emphasizes the paramount importance of addressing software security design flaws, highlighting that such flaws can be just as detrimental as security bugs, as evidenced by real-world breaches like the Target data breach. Addressing security at the design phase not only proves cost-effective but also prevents numerous issues down the development pipeline. A collaborative approach between security and development teams, implementing recommendations such as tamper-proof authentication mechanisms and explicit data validation, is instrumental in preventing design flaws and shifting the focus from bug hunting to risk mitigation. Ultimately, integrating security into the design process and early consideration of design flaws can eliminate entire classes of security vulnerabilities, fortifying the overall integrity of software applications (Higgins, 2014).

In a different resource by Trisha Gee, the author underscores the critical role of code reviews in upholding the quality, reliability, and maintainability of code by examining various facets of the development process. It distinguishes between automated and manual checks, emphasizing the invaluable human insight in assessing complex and subjective aspects of code. Beyond basic formatting, reviewers should delve into design, readability, maintainability, and functionality, ensuring code aligns with architectural principles and design paradigms while maintaining a balance between reusability and YAGNI (Gee, 2015). The blog also advocates for clarity, comprehensibility, and correctness in code and tests, addressing potential security, regulatory, performance, and documentation considerations. It highlights that code reviews encompass a comprehensive and multifaceted evaluation process, benefiting from the combined strengths of automated tools and human expertise.

## Journal Details:

## Code review is a vital and multifaceted process in software development that involves the thorough examination and assessment of code to ensure its quality, reliability, and security. It encompasses a range of activities, from evaluating code formatting and naming conventions to scrutinizing design, readability, maintainability, and functionality. The process aims to identify and rectify issues, align code with coding standards and best practices, and mitigate the risk of vulnerabilities and security flaws.

## Code review is a fundamental practice for computer science professionals because it serves several purposes. Firstly, it ensures the quality and reliability of code by identifying and rectifying issues early in the development process, reducing the likelihood of costly errors and bugs in the final product. Secondly, it enforces coding standards, best practices, and consistency in the codebase, which is essential for collaboration among team members and the long-term maintainability of the software. Additionally, code reviews play a pivotal role in enhancing security by identifying and mitigating design flaws and vulnerabilities, ultimately safeguarding against potential breaches and data leaks. Lastly, code reviews facilitate knowledge sharing and skills development within development teams, making it a valuable practice for fostering professional growth and expertise in the field.

Code reviews typically occur at various stages throughout the software development process. They can take place during and after the initial coding phase. In Agile development, for instance, code reviews may happen as part of a continuous integration process, with developers submitting their code for review as they complete tasks. In more traditional software development methodologies, code reviews may be scheduled at specific milestones or before major releases. The timing and frequency of code reviews can vary depending on the development process, team practices, and the critically of the code changes. The goal is to catch and address issues as early as possible in the development lifecycle to minimize the cost and effort of fixing problems later.

Readability, resilience, and reusability are indeed among the best practices in code reviews. Readable code is easier to understand, maintain, and debug. When code is reviewed, ensuring that it is well-formatted, properly documented, and follows consistent naming conventions is essential. Readable code reduces the likelihood of introducing errors during maintenance and helps new team members understand the code more quickly. In addition, code should be resilient, meaning it can gracefully handle unexpected or erroneous conditions. In code reviews, reviewers should examine error-handling mechanisms, input validation, and how the code responds to edge cases. Furthermore, encouraging code reusability helps prevent duplication and promotes efficiency. In code reviews, reviewers should assess whether the code can be abstracted into reusable components, functions, or libraries. By making code more modular and reusable, development teams save time and effort on future projects and ensure consistency across applications.

# References

Gee, T. (2015, July 22). *What to look for in a Code Review*. Retrieved from The Upsource Blog: https://blog.jetbrains.com/upsource/2015/07/23/what-to-look-for-in-a-code-review/

Higgins, K. J. (2014, August 27). *10 Common Software Security Design Flaws*. Retrieved from DARKREADING: https://www.darkreading.com/application-security/10-common-software-security-design-flaws