# What makes a productive code review?

A code review refers to a step in the software development process where developers obtain a second opinion on their work from members of the software development team. Developing a strong code review process is paramount in ensuring that no unstable code is passed on to the customers. A productive code review offers clear objectives, focuses on key areas and also encourages collaborative discussions. This journal will proceed to highlight its importance for computer science professionals and discuss some of the best practices that should be followed to ensure a successful code review.

To computer scientists, a code review is an inevitable step in the software development process (Ivimey-Cook et al., 2023) which offers the opportunity to learn and share knowledge among team members. In so doing, it fosters a collaborative environment and also ensures that the developers take partial ownership of the code (collective responsibility to ensure the quality of the code) (Felice, 2023). Code reviews also serve as a form of documentation (Felice, 2023). In addition to this, performing a code review at the appropriate time during the software development process helps to maintain consistency in coding style across the board (Ivimey-Cook et al., 2023).

There are several different best practices that should be applied when conducting a code review to ensure maximum efficiency and effectiveness. Firstly, after determining why the code review is conducted in the first place, it is recommended that a code review checklist is created. Much like in many tasks of a similar nature, a code review checklist features a pre-determined set of questions and rules which will be followed throughout the process, providing a structured approach to ensure that the code meets all quality checks before it is introduced to the codebase (Ghani, 2024). Secondly, it is paramount that participants in the code review give constructive feedback (i.e. feedback that helps, not hurts) (Ghani, 2024). A code review in its essence is meant to create a positive environment, rather than one that does not encourage self-improvement and improvement as a team. It is also recommended that no more than 400 lines of code for 60 minutes of code review time is analyzed at any given point (Bellairs, 2019). The code review should always be conducted towards the end of the feature’s life cycle, right before it is released.

# Part 2

After reviewing the rubrics for this week’s code review, I have decided to conduct my code review using the open-source OBS Studio, which is available on Steam. Since my enhancements will be done on the same project for all three categories, I will use the provided checklist as a reference, while also ensuring that I thoroughly explain why each change is necessary to make the overall application a better one. The language I will use in my code review will be simple to understand, and tailored towards those who may be at different levels in understanding programming, from beginners to the experienced.

**References**

Bellairs, R. (2019, December 4). Code Review Best Practices. Perforce.

Felice, S. (2023, September 28). What is Code Review? Browser Stack. https://www.browserstack.com/guide/what-is-code-review

Ghani, U. (2024). 5 Code Review Best Practices. Atlassian. https://www.atlassian.com/blog/add-ons/code-review-best-practices

Ivimey-Cook, E. R., Pick, J. L., Bairos-Novak, K. R., Culina, A., Gould, E., Grainger, M., Marshall, B. M., Moreau, D., Paquet, M., Royauté, R., Sánchez-Tójar, A., Silva, I., & Windecker, S. M. (2023). Implementing code review in the scientific workflow: Insights from ecology and evolutionary biology. Journal of Evolutionary Biology, 36(10), 1347–1356. https://doi.org/10.1111/jeb.14230