In a code review, programmers check each other’s code for mistakes and give recommendations. Regardless of the methods a team chooses for code reviews, there are many important benefits that can be achieved by reviewing and examining code.

## Importance of Code Review

The key benefit of code review is mentorship. Every individual in the team possesses some bits of knowledge that the others don’t have. By showing code to others, one can learn new tips and tricks of the coding process. It can also expose one to different approaches of solving a problem and helps expand the ways to tackle similar issues in the future.

The most important reason to seek out code reviews, though, is for finding defects early on in the development process. It’s better to have two sets of eyes than one. With an extra pair of eyes to review the code, it is more likely to have lesser defects in the code before it goes to the repository.

## How to Review Code

Code reviews are all about collaboration, not about competition.

Follow these five best practices for conducting successful code reviews.

1. Look for Key Things in a Review

These include style, structure, performance, logic, design, functionality, readability, and test coverage. Some of these can be checked through automation, such as structure, but others such as functionality need a human to review.

1. Build and test before the review

Before conducting a manual review, it is important to build and test. Doing automated test helps cut down errors and saves time.

1. Review code for a maximum of one hour at one time

Going beyond this timeline tends to reduce attention-to-detail and performance. It is advisable to conduct frequent reviews with shorter intervals instead. By taking a break, the brain gets a chance to reset and helps perform better reviews.

1. Check 400 lines of code at the most

Again, reviewing too many lines of code can result in lesser probability to find defects. Each review should be kept for 400 lines or lesser. This limit is significant for the same reasons as setting time limits.

1. Give constructive feedback

Instead of being critical in feedback, a better approach is to be constructive. Point out the issues and suggest ways of improvement. Ask questions instead of making statements, and praise alongside the feedback.

Our code review team at Aardwolf Security consists of professionals who follow best practices and provide key recommendations. To get our help with code reviews, contact us today.