

***Meta:** Understand the concept of code review, its impact on software quality, the challenges involved in the process, and best practices.*

What is Code Review? Definition, Importance, Types, and Tips

Efficient web development is characterized by low release time and superior code quality. Code review is a process that helps achieve both objectives. It's as integral to web development as editing is to book publication. That's why the knowledge of code review is essential for anyone trying to build a career in this field.

This article serves as a great starting point and provides a comprehensive overview of the concept, benefits, and pitfalls of code review. By the end of the article, you will learn the code review process and the advantages of choosing a well-structured [full stack web development program](#) to become an expert code reviewer.

What Is Code Review?

Code review refers to reviewing the code of other software developers, usually junior developers. It is a systematic process where you collaborate with your team members to ensure that you deliver high-quality, error-free code. Code review is similar to editing. While editing looks for areas to improve in a manuscript, code review reveals areas to enhance in a code.

The Importance of Code Reviews

No code is perfect from the get-go. Every code contains minor flaws, some of which may prove disastrous if they go undetected until the software is deployed. Hence, code review is crucial to ensure a streamlined software delivery and deployment.

Code review is important for several reasons, some of which are:

- **Enhanced quality:** Code review helps to narrow down areas where the code can be rewritten and improved. You can provide feedback regarding alternative options, best practices, and critical deletions or merges.
- **Elimination of bugs and flaws:** Prevention is better than cure. Code review identifies undetected bugs and issues, leading to immediate redressal and code modification.
- **Regulatory compliance:** Code review helps verify the code's compliance with the regulations applicable to the software in question. It irons out any errant terminologies, missing content, or restricted aspects.
- **Uniformity in style:** The form, practice, and code style can vary with the developer. Hence, even if the code is technically correct and operational, it may not align with the organization's usual coding practices. Thus, code review helps to bring consistency to the coding style so that it is followed by the developer in the future.

- **Responsible coding:** Code review enforces partial code ownership, implying that the code reviewer is accountable for the code's operation. Thus, it becomes a shared responsibility between the coder and the reviewer to ensure a clean and efficient code.
- **Collaborative learning:** Coding is essentially an isolating job, which is remedied by code review. Code review encourages collaboration and knowledge sharing, as the developer has to interact with the team members and receive their input. The developer gains more knowledge about how coding works in the real world, which helps reconcile their theoretical expertise with the street smartness required to make the coding work.
- **Quality inspection:** Besides improvement, code review is a quality assurance process that attests to functional and user requirements. It adds credence to the organization's claim of delivering high-quality work.
- **Continuous improvement:** Code review inculcates a habit of critical code evaluation among the team. Through regular code reviews at each step of code development, the team members work towards flaw-free code. They become more receptive to feedback and continuous improvement.
- **Documentation:** Code review records the discussions, modifications, and comments received during the coding development. Any future issues can be traced back to the code review to check if any critical part was missed.

Types of Code Review

While code review aims to evaluate the code thoroughly, several methods exist. Let's look at the different types of code review that software engineers choose from based on their project's needs.

- **Pair programming:** It's where two developers share a computer for their work. One of them writes the code, and the other simultaneously reviews it. This helps in real-time code review and correction.
- **Pull request (PR) review:** Pull request reviews are when developers working on version control systems like GitHub propose and create pull or merge requests. The code-developing team reviews the requests, and the code is modified after the changes are approved. This helps in a more collaborative code review.
- **Tool-assisted review:** Tool-assisted review is a type of code review in which the reviewers use tools and platforms such as Bitbucket, Crucible, Review Board, GitHub, GitLab, Gerrit, Space, Phabricator, Azure DevOps, etc. The tools enable a data-driven approach using metrics to assess the code and identify errors.
- **Over-the-shoulder review:** This type of review utilizes a traditional review method, where a team member sits with the developer and goes through each line of code on a single computer screen. They evaluate the code together, and the developer can make changes that can be reviewed immediately.
- **Email-based review:** This review does not involve a physical interaction between the developer and the reviewer. The developer sends the code to the reviewer via email, and the reviewer sends his comments back. A to-and-fro email transaction is used to review, modify, and finalize the code.
- **Ad hoc review:** An ad hoc review is a more casual code review process in which the developers ask the team members for a quick review. There is no formal email or intense physical review. These are typically unplanned and spontaneous. Such an ad

hoc review reduces the pressure on the team members to allocate time for a detailed review of the entire code.

- **Checklist review:** This type of code review is based on a checklist that enlists the various components and requirements of the code. The reviewer goes through the code and verifies that each item on the checklist has been covered by the code. Such a review may be combined with one of the other detailed reviews, and the final comments may include the additions required to complete the checklist.
- **Formal inspection:** Formal inspection is an official code review process in which the code must comply with predefined inspection parameters and aspects. It is undertaken after the code has been completed. It is an intensive process in which a dedicated team undertakes the inspection and provides a detailed report of the required changes and the status of the code.

What is Code Review, and What Are Its Benefits and Challenges?

While code review may seem pretty straightforward, it has certain benefits and challenges you must consider. Code review is vital for the software development process as it:

- Identifies errors and resolves bugs during the development stage
- Enhances the quality of the code
- Improves the security of the code by detecting security vulnerabilities
- Inculcates a sense of collaboration among the team members
- Reveals opportunities for reusing the code
- Promotes sharing of skills and expertise among team members
- Encourages shared ownership of the code and reduces the dependency on a single person
- Creates a documented journey of code development
- Ensures regulatory compliance
- It helps develop a critical eye and a mindset for continuous improvement among the team members

However, certain challenges code reviewers face during the review, such as—

- Code review can become a time-intensive process if the code is long and complex, delaying delivery and deployment
- The feedback and comments are subjective and may differ with respect to the reviewer
- Unless the organization or the project has a predefined style, the reviewers may enforce their style for the code, leading to an inconsistent coding form
- Code reviews for minor modifications, more frequently, may lead to additional overheads and unnecessary expenditures
- Sometimes, reviewers may flag aspects that do not have any problem, resulting in unnecessary modifications and a waste of time on avoidable discussions
- Reviewers without sufficient experience may not be able to identify all the issues, may not be able to suggest the correct type of changes, or may overlook some flaws
- Code reviews conducted by multiple members may lead to confusion due to a mismatch or disagreement regarding the code
- Developers and reviewers may have to prioritize code review, delaying the completion of other tasks
- Frequent reviews may cause mental fatigue in reviewers and result in them missing important errors

Best Practices in Code Reviews

Code review is crucial in ensuring the high quality of new features or products. While a good code review definition can vary among companies, certain accepted guidelines can elevate the process. Here are some of those best practices:

- Establish the objectives of the code review, narrow down the goals, and define the exact tasks to be completed
- Determine the best practices, formats, style guides, and coding standards to be followed and make them accessible to all the team members
- Choose the correct reviewer so that it falls within the reviewer's scope and expertise
- Divide the code into small parts and provide comments and changes in manageable chunks
- Provide detailed description, background, purpose, and expectations of the changes to help the developer understand them first
- While providing feedback, prioritize code quality by ensuring that the developers work toward improving the quality of other parts while making the changes suggested by the reviewer
- Explore and use code review tools to automate the minor aspects of the review as well as obtain data-based feedback
- Maintain a timeline for the review and provide feedback within the deadlines to avoid delaying the process

How to Perform a Code Review?

Now that you know why we need a code review, let's discuss how to conduct one. Here are the major steps typically followed.

1. Code review begins when the developer notifies the team members that they have completed a portion of the code and would like to get it reviewed. They may create review requests flagging the code to be reviewed.
2. The code is then assessed, and the correct reviewer is selected based on their expertise and experience in conducting code reviews.
3. The reviewer then examines the code and chooses code review tools suitable for the code.
4. A checklist is created based on the coding standards, user requirements, functional objectives, and compliance regulations.
5. The reviewer then thoroughly inspects the code and analyzes it for issues such as standard non-compliance, security inefficiency, logic and syntax errors, unnecessary additions, and bottlenecks.
6. The developer then receives the comments from the reviewer. They may further discuss the feedback and resolve the doubts.
7. The developer further makes the changes and contacts the reviewer for another review. This process may be repeated until the code is modified within the project timeframe to the reviewer's satisfaction.
8. The reviewer gives the final approval, and the code is included in the primary codebase.
9. The reviewer revisits the code and assesses its performance. They monitor the effect of the code changes and may flag issues if required.

10. The code review process is officially closed after the documentation is completed.

Master the Skills to Launch a Successful Web Development Career

Code review is a critical process that streamlines web development. For a successful web developer, it's not enough to possess in-depth knowledge of coding but also the ability to note even those errors that may be missed by code review tools.

You can develop these skills by completing an industry-recognized [full stack web development bootcamp](#), which trains you in crucial concepts such as Agile and Scrum methodologies, GitHub version control systems, back-end technologies, and Spring Boot. Enroll today to learn from industry leaders, build your portfolio through capstone projects, and network to boost your career. Get started today!