https://www.geeksforgeeks.org/5-best-practices-for-code-review/

5 Best Practices For Code Review

- Difficulty Level : Basic
- Last Updated : 08 Mar, 2022

Software Development Process refers to implementing the design and operations of software, this process takes place which ultimately delivers the best product. Do several questions arise after this process like whether the code is secure? Is it well-designed? Is the code free of error? **As per the survey, on average programmers make a mistake once at every five lines of the code.** To rectify these bugs Code Review comes into the picture. Reviewing a code typically means checking whether the code passes the test cases, has bugs, repeated lines, and various possible errors which could reduce the efficiency and quality of the software. Reviews can be good and bad as well. Good ones lead to more usage, growth, and popularity of the software whereas bad ones degrade the quality of software.

In this blog, we will discuss the **5 steps to a complete review code.** So let's get started.

1. Split the Code into Sections

For web development, several files and folders are incorporated. All the files contain thousands of lines of code. When you start reviewing them, this might look dense and confusing. So, the first step of **code review** must be splitting the code into sections. This will make a clear understanding of the code flow.
Suppose, there are 9 folders and each folder contains 5 files. Divide them into sections. Set a goal to review at least 5 files of the first folder in n no of days and once you complete reviewing it, go for the next folder. Like this, when you assign yourself a task for some time, you'll get sufficient time to review, and thus, you'll not feel bored or disinterested.

2. Ask Fellow Developers to Review

This is the second step of the **code review process**. You must seek advice or help from fellow developers as everyone's contribution is equally important. Experienced ones can identify the mistakes within a second and rectify them but the young minds come up with more simple ways to implement a task. So, ask your juniors as they have the curiosity to learn more. To make it perfect, they find other ways which will benefit in two ways –
a) *They'll get deeper knowledge.*
b) Solution can be more precise.

The below quote states the best outcome of teamwork. Thus, teamwork improves the performance of software and fosters a positive environment.

"Alone, we can do so little. Together, we can do so much"
– Helen Keller

3. Basic Principles: Naming Conventions, Usage of libraries, Responsiveness

There are some **principles and standards to follow while writing code.** There has to be followed to enhance the effectiveness and productivity. Make a note of those principles and check one-by-one whether they're followed or not. The below ones describes some of the standards every developer should follow. You can also check for more.

Naming Conventions: Use standard names for variables to assign values. The name should be meaningful, pronounceable, sound positive. Before naming, always keep in mind that whenever anyone reads it, it should be understandable.

Usage of Libraries: A library is a generalized file of code that acts as a resource used by programs often under **software development.** To avoid lines of code, we use the library, we import (call and use) several methods from libraries and use them in our code to reduce complexity.

Responsiveness: It creates dynamic changes on the website. Do check for the responsiveness of the website as to whether it works on all devices like mobile phones, tablets, laptops, etc. This also helps websites get **higher search engine results**.

4. Check For the Reusability of Code

Functions are reusable blocks of code. A piece of code that does a single task that can be called whenever required. Avoid repetition of codes. Check if you've to repeat code for different tasks, again and again, so there you can use these functions to reduce the repeatability of code. This process of using functions maintains the codebase.

For example, if you're building a website. Several components are made in which basic functionalities are defined. If a block of code is being repeated so many times, copy that block of code or function to a file that can be invoked (reused) wherever and whenever required. This also reduces the complexity level and lengthiness of the codebase**.**

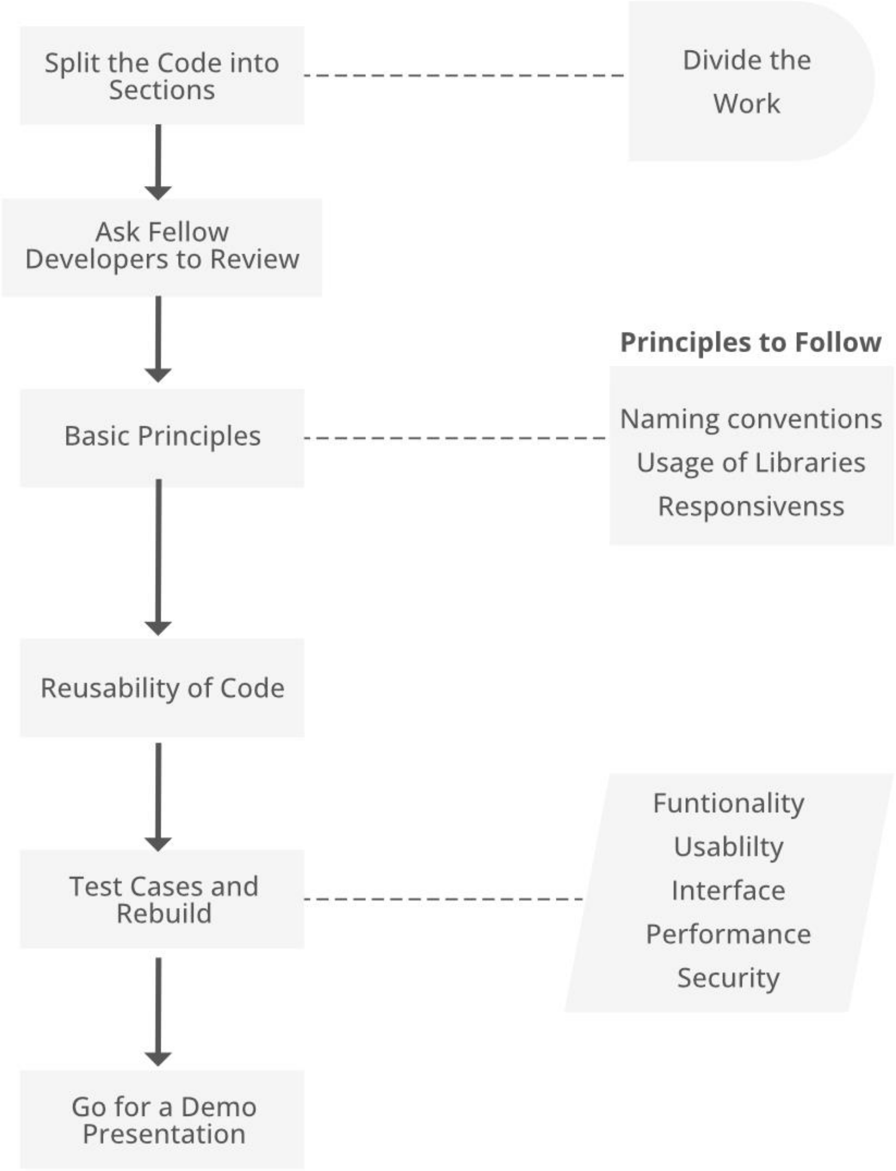
5. Check Test Cases and Re-Build

This is the final step of **the code review process**. When you have rectified all the possible errors while reviewing, check if all the test cases are passed, all the conditions are satisfied. There are various tests such as functionality, usability, interface, performance, and security testing.

- ***Functionality:*** *These tests include working of external and internal links, APIs, test forms.*
- ***Usability:*** *Checking design, menus, buttons, or links to different pages should be easily visible and consistent on all web pages.*
- ***Interface:*** *It shows how interactive the website is.*

- ***Performance:*** *It shows the load time of a website, tests if there's a crash in a website due to peak load.*
- ***Security:*** *Test unauthorized access to the website.*

Once all the test cases are passed, re-build the entire code. After this process is done, go for a look over the website. Examine all the working like buttons, arrow keys, etc.



Go For a Demo Presentation

When all the steps of the **Code Review** process stated above are done, go for a demo presentation. Schedule a flexible meeting and give a presentation to the team demonstrating the working of the software. Go through the operations of every part of a website. Tell them about the changes made. Validate your points as to why these changes have been done. See if all requirements are fulfilled and also the website doesn’t look too bulky. Make sure it is simple and at the complete working stage.

Things to avoid while reviewing code

1. ***Don’t take too many files at a time to review.***
2. ***Don’t go for continuous reviewing, take breaks.***
3. ***So many nested loops.***
4. ***Usage of too many variables.***
5. ***No negative comments to anyone in a team.***
6. ***Don’t make the website look too complex.***

So till now you must have got the complete picture of the **Code Review process**. It is a very tedious process in any modern development team’s workflow. It helps in giving a fresh start to identify bugs and simple coding errors before your product gets to the next step or deployment, making the process for getting the software to the customer more efficient. Before getting your prototype turned into a product, do a proper **code review** or scrutiny to get the best version of it