

Manual Testing

Manual testing is a software testing process in which test cases are executed manually without using any automated tool. All test cases executed by the tester manually according to the end user's perspective. It ensures whether the application is working, as mentioned in the requirement document or not. Test cases are planned and implemented to complete almost 100 percent of the software application. Test case reports are also generated manually.

Manual Testing is one of the most fundamental testing processes as it can find both visible and hidden defects of the software. The difference between expected output and output, given by the software, is defined as a defect. The developer fixed the defects and handed it to the tester for retesting.

Manual testing is mandatory for every newly developed software before automated testing. This testing requires great efforts and time, but it gives the surety of bug-free software. Manual Testing requires knowledge of manual testing techniques but not of any automated testing tool.

Manual testing is essential because one of the **software testing** fundamentals is "100% automation is not possible."

Why we need manual testing

Whenever an application comes into the market, and it is unstable or having a bug or issues or creating a problem while end-users are using it.

If we don't want to face these kinds of problems, we need to perform one round of testing to make the application bug free and stable and deliver a quality product to the client, because if the application is bug free, the end-user will use the application more conveniently.

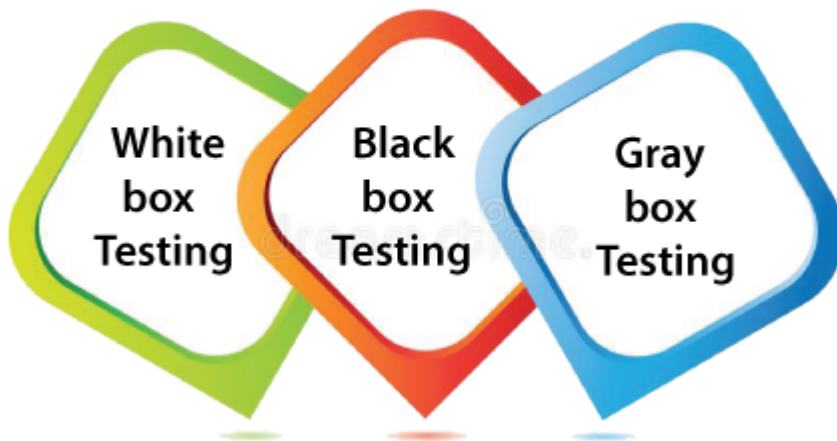
If the test engineer does manual testing, he/she can test the application as an end-user perspective and get more familiar with the product, which helps them to write the correct test cases of the application and give the quick feedback of the application.

Types of Manual Testing

There are various methods used for manual testing. Each technique is used according to its testing criteria. Types of manual testing are given below:

- White Box Testing
- Black Box Testing
- Gray Box Testing

Types of Manual Testing



White-box testing

The white box testing is done by Developer, where they check every line of a code before giving it to the Test Engineer. Since the code is visible for the Developer during the testing, that's why it is also known as White box testing.

For more information about white box testing, refers to the below link:

<https://www.javatpoint.com/white-box-testing>

Black box testing

The black box testing is done by the Test Engineer, where they can check the functionality of an application or the software according to the customer /client's needs. In this, the code is not visible while performing the testing; that's why it is known as black-box testing.

For more information about black-box testing, refers to the below link:

<https://www.javatpoint.com/black-box-testing>

Gray Box testing

Gray box testing is a combination of white box and Black box testing. It can be performed by a person who knew both coding and testing. And if the single person performs white box, as well as black-box testing for the application, is known as Gray box testing.

To get more details about gray box testing, refers to the below link:

<https://www.javatpoint.com/grey-box-testing>

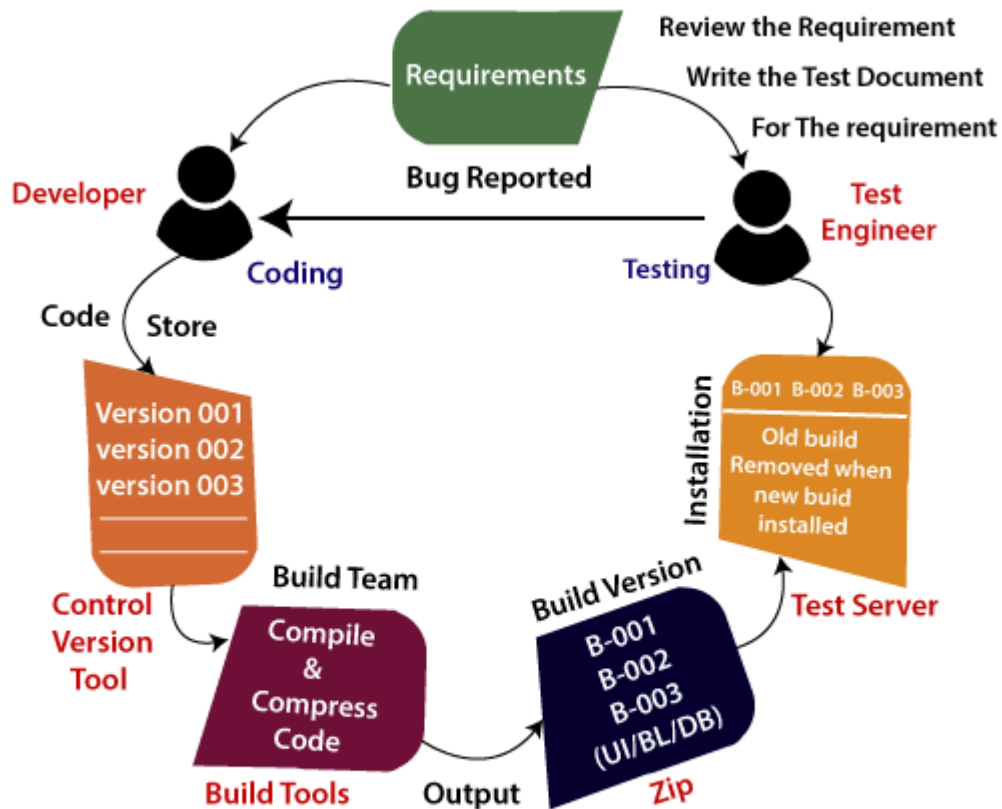
How to perform Manual Testing

- First, tester observes all documents related to software, to select testing areas.
- Tester analyses requirement documents to cover all requirements stated by the customer.
- Tester develops the test cases according to the requirement document.
- All test cases are executed manually by using Black box testing and white box testing.
- If bugs occurred then the testing team informs the development team.
- The Development team fixes bugs and handed software to the testing team for a retest.

Software Build Process

- Once the requirement is collected, it will provide to the two different team development and testing team.
- After getting the requirement, the concerned developer will start writing the code.
- And in the meantime, the test engineer understands the requirement and prepares the required documents, up to now the developer may complete the code and store in the **Control Version tool**.

- After that, the code changes in the UI, and these changes handle by one separate team, which is known as the **build team**.
- This build team will take the code and start compile and compress the code with the help of a build tool. Once we got some output, the output goes in the zip file, which is known as **Build** (application or software). Each Build will have some unique number like (B001, B002).
- Then this particular Build will be installed in the test server. After that, the test engineer will access this test server with the help of the Test URL and start testing the application.
- If the test engineer found any bug, he/she will be reported to the concerned developer.
- Then the developer will reproduce the bug in the test server and fix the bug and again store the code in the Control version tool, and it will install the new updated file and remove the old file; this process is continued until we get the stable Build.
- Once we got the stable Build, it will be handed over to the customer.



Software Build Process

Note1

- Once we collect the file from the Control version tool, we will use the build tool to compile the code from high-level language to machine level language. After compilation, if the file size will increase, so we will compress that particular file and dumped into the test server.
- This process is done by **Build team, developer** (if build team is not there, a developer can do it) or the **test lead** (if the build team directly handle the zip and install the application to the test server and inform the test engineer).
- Generally, we can't get a new Build for every bug; else, most of the time will be wasted only in creating the builds.

Note2

Build team

The main job of the build team is to create the application or the Build and converting the high-level language into low-level language.

Build

It is software, which is used to convert the code into application format. And it consists of some set of features and bug fixes that are handed over to the test engineer for testing purposes until it becomes stable.

Control version tool

It is a software or application, which is used for the following purpose:

- In this tool, we can save different types of files.
- It is always secured because we access the file from the tools using the same login credentials.
- The primary objective of the tools is to track the changes done for the existing files.

Example of Build process

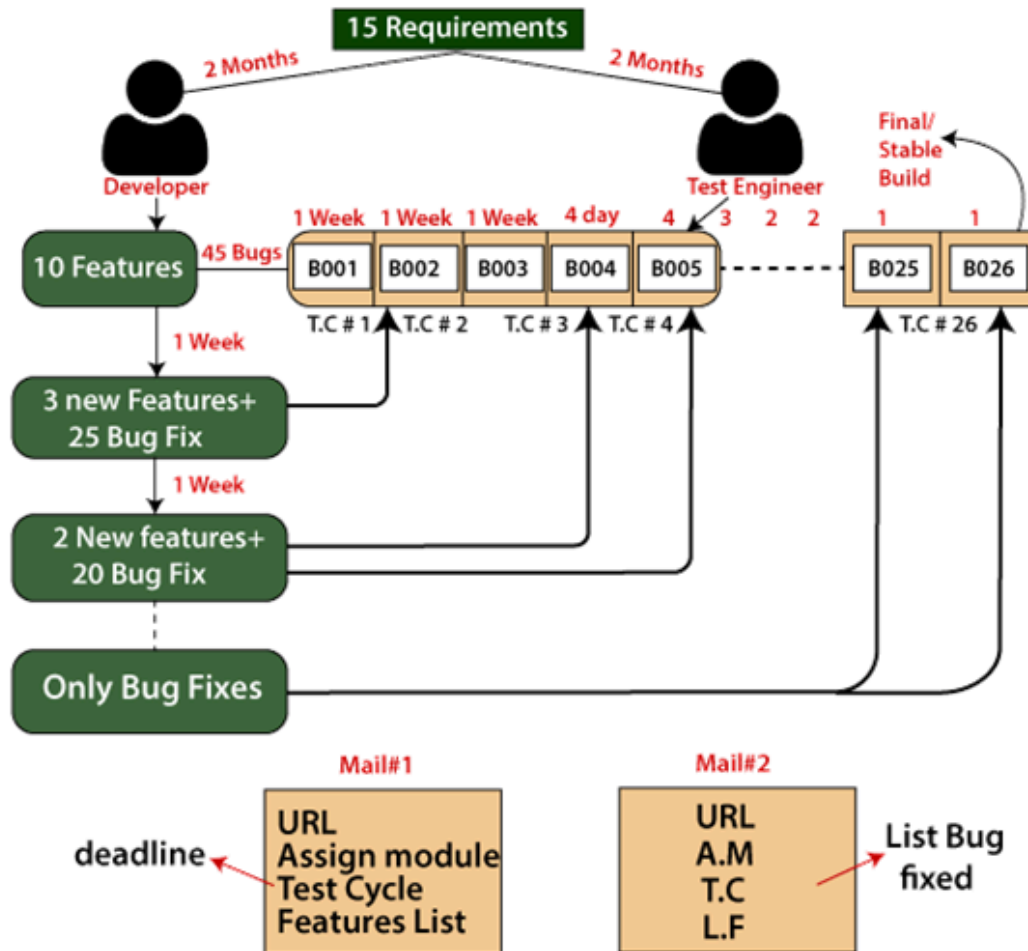
Let see one example to understand how to build process work on the real scenarios:

As soon as the test engineer gets the bug, they will send it to the developers, and they need some time to analyze; after that, he/she only fixes the bug (Test engineer can't give the collection of bug).

The developer is decided how many bugs he can fix according to their time. And the test engineer is decided, which bug should be fixed first according to their needs because the test engineers cannot afford to stop testing.

And the test engineer getting the mail, they can only know that which bug is fixed by the **list of the bug fixes**.

The time will increase because at the first Build, and developers should write the code in the different features. And at the end, he/she can only do the bug fixes and the numbers of days will be decreased.



Note3

Test cycle

The test cycle is the time duration given to the test engineer to test every Build.

Differences between the two build

The bugs found in one build and can be fixed any of the future Build, which depends on the test engineer's requirement. Each new Build is the modified version of the old one, and these modifications could be the bug fixes or adding some new features.

How frequently we were getting the new Build

In the beginning, we used to get weekly builds, but in the latest stage of testing, when the application was getting stable, we used to get the new Build once in 3 days, two days, or a daily basis as well.

How many builds we get

If we consider one year of any project duration, we got 22-26 builds.

When we get the bug fixes

Generally, we understand the bug fixes only after the test cycle is completed, or the collection of bugs is fixed in one build, and handover in the next builds.

Advantages of Manual Testing

- It does not require programming knowledge while using the Black box method.
- It is used to test dynamically changing GUI designs.
- Tester interacts with software as a real user so that they are able to discover usability and user interface issues.
- It ensures that the software is a hundred percent bug-free.
- It is cost-effective.
- Easy to learn for new testers.

Disadvantages of Manual Testing

- It requires a large number of human resources.
- It is very time-consuming.
- Tester develops test cases based on their skills and experience. There is no evidence that they have covered all functions or not.
- Test cases cannot be used again. Need to develop separate test cases for each new software.
- It does not provide testing on all aspects of testing.
- Since two teams work together, sometimes it is difficult to understand each other's motives, it can mislead the process.

Manual testing tools

In manual testing, different types of testing like unit, integration, security, performance, and bug tracking, we have various tools such as [Jira](#), [Bugzilla](#), Mantis, Zap, NUnit, Tessy, LoadRunner, Citrus, SonarQube, etc. available in the market. Some of the tools are open-source, and some are commercial.

For more information about testing tools, refers to the below link:

<https://www.javatpoint.com/software-testing-tools>

