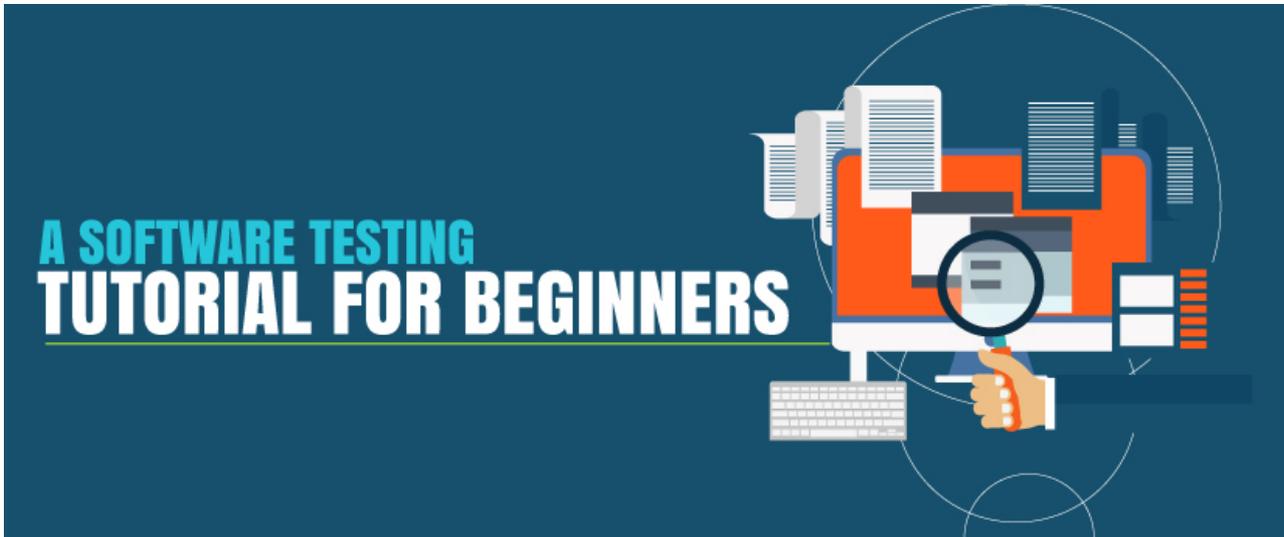


Software Testing Tutorial For Beginners

t testbytes.net/blog/software-testing-tutorial-for-beginners

November 26,
2019



We are sure you are here because you want to know what software testing tutorial is all about and how will it take you to the next level of your career. Before we start, software testing methodologies of verifying and validating that all the bugs/faults in a software product are resolved and is fit for use by the end user.

Software testing doesn't just stop at fixing all the bugs, but building a strategy that can improve the products usability, accuracy and reliability. In this blog, we'll help you understand the concepts and techniques involved in the process which will certainly help you to decide if this is the right profession you are looking for.



Software testing can be divided into two paces:

1. **Validation**
2. **Verification**

Validation is referring to various types of pre-specified tasks to ensure it matches with the customers' requirements. It involves tasks like automated testing, running scripts and checking functional dependencies.

Whereas, Verification refers to checking the tasks that makes the product actually run in its real time. It involves reviewing the test cases, inspections and documentation



Some key differences are as follows:

S. No	Validation	Verification
1	Does the final product actually meet what was proposed?	Is the product built with respect to the requirements?
2	<u>Automated & Manual Testing</u>	Document reviews, inspections and walk-throughs
3	Execution of application required	Does not require execution of software

Types of Software Testing

If you ever had to visit a training institute or an interview, you certainly must have heard the terms automation and manual testing. Well, that's nothing but types of testing. So let us classify these two:

Manual Testing:



Manual testing is the process undertaken by a tester himself to check if the systems performs as expected. In other words, the testing is done without using any automated tool or script.

The Manual Tests include a set of test cases defined by the tester and then each case is verified for their respective functions. To do this, the tester has to follow some steps:

Requirement Analysis

To successfully run manual tests, you should first know what the application is all about and what output is expected from its behaviours. For this, you need to analyse its requirements and study it thoroughly. By doing this you will know what you are testing and makes reaching your goal a lot easier.

Writing Test Cases:

A well written test case allows you to understand and run the tests with ease. It is like a guide which supports you during your testing process. There are many software's now that allows you to tip down those cases. While you write the tests, make sure you think like an end user. After-all you are testing it for them.

Regulate the Test Cases

Once you are done writing all test cases, it's time for executing each one of them. Jot down the "Passed", "Failed" and "Skipped" tests and leave the rest for the developers to fix.

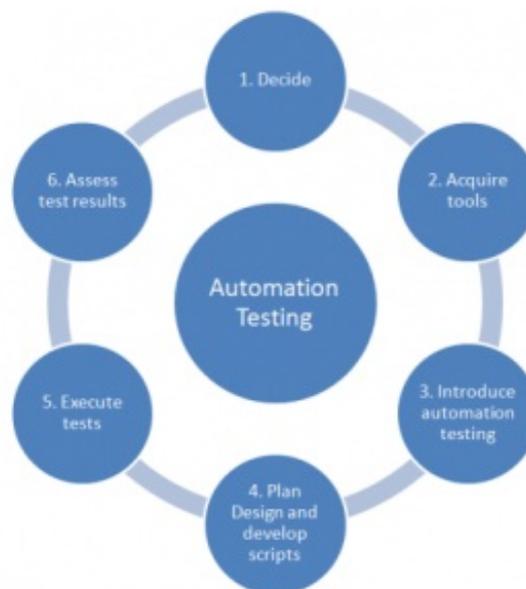
Logging of Bugs

Once you identify the state of the bugs, make sure you follow a proper medium to communicate them to the developing team. A lot of developers prefer using the [Mozilla Bugzilla](#), which is an add-on to the browser.

Reporting

The final steps expects you to report any additional assumptions and conditions which couldn't be marked. Also mention suggestions to improve validations and processes.

Automation Testing



Automation testing, as the named suggests, is automated. It differs from manual testing in a way that uses automated scripts for running the test cases. It is a boon in today's testing market.

Automated tests are basically conducted on large projects, which otherwise would consume a lot of time in manual. Some companies also run automation after the manual to make sure all the test cases are covered. Follow the below steps before proceeding with your automation tests.

1. Learn and decide what tests to automate
2. Test as often as you can
3. Selecting the right tool for automation
4. Make sure each test has its own contribution towards the efforts.
5. Have quality data in hand
6. Create automated scripts that can accommodate changes in the UI

When should you opt for automation testing

Always go for test automation if:

1. The project is large and complex
2. The project requires continues testing for the same area
3. Time constraints
4. The software is stable with respect to manual tests.

Tools for automation tests:

The Techniques of Software Testing

Believing you now have a clear idea what software testing and its types are, let's see the techniques used. There are 2 techniques defined in software testing:

1. Black-Box Testing

Black Box Testing deals with the internal working of the system where only the system coder has access to the source code. Hence, while testing, the tester has to perform the tests under the guidance of the developing team for inputs and outputs.

Black box testing clearly separates the testers and developers point of view to look at the software. Hence is well-suited for large code segments of the project. But certainly, this complicates the designing of test cases.

2. White-Box Testing

White-Box testing deals with the program logic and structure of working. For this, you need to first understand the internal working of the code, which is easier than working with the code.

Since the user has an understanding of how the code functions, it's easier to identify the data to be tested. That's why it's also called as glass-box testing or open-box testing.

Levels Of Testing

During testing, you come across various levels of steps. In this section let's have a brief look at these.

The Levels are divided into 2

- Functional Testing
- Non-Functional Testing

Functional Testing

Functional testing is performed when the actual software development is taking place ensuring that all its requirements are met. Not to forget, the test is performed only when a piece of module is complete.

The process involves 5 steps:

1. Defining the functionality to be checked
2. Creating the test cases
- Comparing the output based on the test cases
1. Execution of test cases and writing scenarios
2. Comparing the actual and expected results

Functional testing can be performed by the following tests:

1) Unit Testing: The tests are conducted before the program is handed to the testing team for executing test cases

2) Integration Testing: This testing is done by putting together parts of the application to determine if they together functional properly.

3) System Testing: This testing tests the whole system as one. Once all the system components are integrated, it is checked for the working on specified quality standards.

4) Regression Testing: Many a times, when there is an unplanned bug in the system testing, there are possibilities that it might have affected or will affect other components in the system. In such a case, regression testing is carried out.

5) Acceptance testing: Considered as one of the important tests in functional testing phase, acceptance testing is done purely by the Quality Assurance team. They check it for the final specification meeting the requirements.

6) Alpha Testing: A test conducted by the QA team and the developers, which combines all of the above tests for a final review before its tested by the end users.

7) Beta Testing: The final test before its release, the beta test is done by a chosen audience. The audience may be any non-technical team, or a group of new users. The application is run in real time to check the application runs as expected.

Non-functional Testing

Non-functional testing deals with bugs that are caused by any external factors like network, database processing, client-side downtime etc. The non-functional testing types includes:

1) Performance Testing

Identifies any performance issues that affects the speed, stability and reliability of the application. Performance testing can be further divided into Load testing and Stress testing based on its qualitative and quantitative aspects.

2) Security Testing

As the name implies, security testing check for flaws related to the security of the program. It includes confidentiality, validations for inputs, attacks due to SQL insertions, scripting and virus attacks etc.

Conclusion



So the reason is much simpler now, you know what you deserve in a testers career. So if you are a beginner, make sure you understand all the aspects of testing well before jumping in. We will be happy if you are even half way through to deciding your career in testing. Take your first step towards being an excellent tester.