**SOFTWARE TESTING**

**Fundamentals of Software Testing**

Software testing is a vast subject. There are software applications and system engineered for numerous domains and industries, and for a tester, every testing project is a new challenge because he has to understand the client’s point of view and the domain before moving on with testing activities. From project to project, a tester may have to change the testing methodologies as well. It is therefore very important to keep the fundamentals right. Getting the fundamentals right in the first place is biggest prerequisite to become successful in software testing.

**What is Software Testing?**

**Software Testing** is a method to check whether the actual software product matches expected requirements and to ensure that software product is[Defect](https://www.guru99.com/defect-management-process.html)free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

Some prefer saying Software testing as a [White Box](https://www.guru99.com/white-box-testing.html) and [Black Box Testing](https://www.guru99.com/black-box-testing.html). In simple terms, Software Testing means the Verification of Application Under Test (AUT)

**Why Software Testing Is Necessary?**

An error, defect or a bug can be caused by developers. It is not intentional but considering the complexity with which various software are being developed these days, it is quite possible for a developer to misunderstand and implement wrong logic and produce wrong code.

Testing is necessary because it helps us in identifying the faults in software. Once these defects have been detected they can be easily rectified, and quality of the software can be improved. So, software testing is necessary so that bug free applications can be developed and delivered. When a company decides to develop software for a client there are certain legal, contractual and industry-specific requirements based on the deal is made. A quality conscious company will definitely include software testing in its best practices.

It is difficult to say how much testing is enough but the fact is that if testing is planned carefully and good test cases are made then it is very much possible to deliver high quality software.

**Who Does The Software Testing?**

There is often a debate on who should actually test the software. People often question that why developers are not allowed to test. Well, a developer generally checks his code several

times before he submits it for testing and still in most cases it is never error free because a developer is generally blind to his own mistakes.

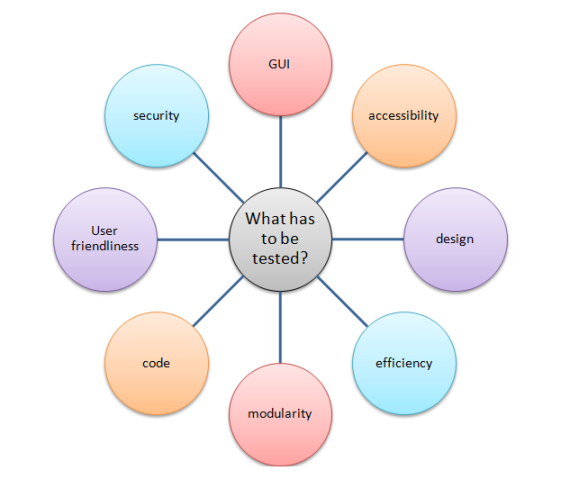
A tester on the other hand looks at software from the point of view of the client. He is unbiased and his focus is only on the specifications and the requirements. So, a tester is able to look into areas that a developer may have ignored. So, the testing should always be carried out by independent testers.

This approach does have some disadvantages. When the development and testing teams are different there is often a communication gap and sometimes, developers become careless towards coding and do not revise their code because they think that it is all a tester’s job thereby increasing the burden on the tester.

Many times developers share their work amongst each other and test each other’s work. This is known as buddy testing. Every development team should have dedicated testers and every project generally has at least one dedicated testing team. Some companies believe in having separate teams for different types of testing, this means different teams for usability, performance, security and other forms of testing. Some companies believe in outsourcing software testing work which means they hire a firm or independent testers or consultants to have a look at and test their project.

**What Has To Be Really Tested?**

The tester should have a good understanding about the project requirements. A fair idea about the real time scenario where the software will be implemented can help the tester understand how to carry out testing for the project. It is very important to know what has to be really tested in order to devise a testing strategy



**When Is The Software Testing Done?**

The earlier the testing team starts testing the software the easier it would be for the developers to complete the project on time and this would also save a lot of time, money and effort. Starting testing in the later stages of development can turn out to be an expensive matter as it is very difficult to rectify defects once the software has reached the final stages of development. Dividing software development into stages and then testing work done in every stage before moving on to the next stage helps in finishing the software development in time with good results. This also helps in better integration of different modules because you already know that every module has been tested independently and is working as per the given specifications.

**How Often Do We Need To Test?**

How often you need to test depends on how important the quality is for you. Ideally, testing should go hand in hand with development and a tester should focus on discovering maximum number of defects during the initial phases of software development so that if the design of the software requires any changes then it can be done early as it will be very difficult and expensive to make major changes in the project during the later stages of development.

  
**How Often Do We Need To Test?**

**What Are Software Testing Standards?**

Software testing standards are of great importance from the consumer’s as well as producer’s point of view. A consumer invests in the software and if the software is of good quality then at the end of the day he is satisfied that he has purchased the right thing for himself.

All reputed companies ensure that the software quality of product is governed by some sets of standards that have been approved by the public. By abiding by these standards a company gives assurance about its products and it is able to give guarantee to its customers only when it has followed some standards and knows that the software will behave in a certain manner. So, consumer knows that he is buying the right thing if it is of right standard.

From a producer’s point of view standards help in improving the quality of the final product. Once a company has finalized the standard that it has to follow then it becomes easier for them to work on other software projects and whenever they start a new project they do not need start everything from scratch. There are many types of software testing standards defined for evaluating quality of software which can greatly improve the effectiveness of software testing however it is believed that till now no such standards have been made that can cover all aspects of software testing.

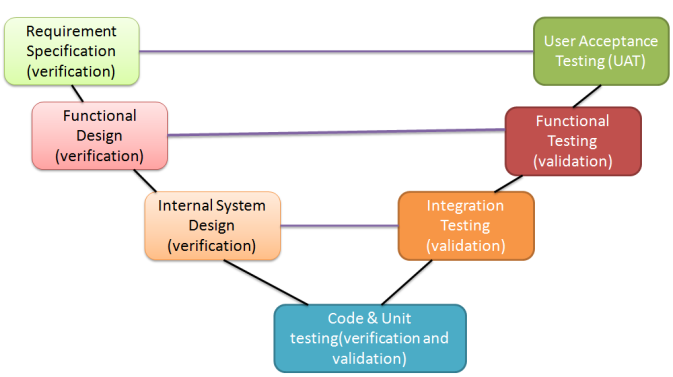
Standards that emphasize on having testing as part of larger requirement or standards supporting software testing are the ones that can be of use for software testing.

**Software Verification VS Software Validation**

Verification and validation are two very important terms in software testing. People often get confused between the two however these two terms are related to two different types of analysis. Validation is helps in building the right system. While carrying out validation we look at whether the system is in line with customer’s requirement or not.

Verification on the other hand helps in ensuring if the system is being developed in the right way. The focus of verification is on the quality of software that is being developed, whether it follows all the standards or not, is it well engineered?

So while validation checks if the specifications have been designed correctly to meet the customer’s requirement, the verification checks if the software has been developed as per the software quality standards and norms of the software engineering organization.

  
**Software Verification VS Software Validation**

Some theories suggest that verification is carried out in every phase of software development lifecycle but the same is not the case with validation. Validations are crucial in the beginning and towards the end of the project, i.e. during the requirement analysis and acceptance testing. This stactic is not fully correct and almost impossible to follow. The actual fact is that until today it has been observed that it is very difficult to capture the entire set of client requirements during the beginning of the project. Software requirements often undergo several changes even after the development has started. Many times the changes are requested by the development team itself. It is therefore important to carry out validation and verification processes in every phase of software development.

Today, testers usually consider verification and validation also known as V&V as a powerful way of looking at various aspects of the software.

**Software Testing VS Software Debugging**

This is another topic where people generally get confused. Software testing and debugging may sound like one and the same thing but that is not actually the case. To start with, the process of debugging starts when software testing gets over. While software testing uncovers defects, debugging removes defects from the system.

1. Software testing uncovers defects and debugging locates and corrects it.
2. Software testing is a very important aspect of software development cycle whereas debugging is a result of testing activities.
3. Testing begins soon after development starts. Debugging starts when testers start reporting defects.
4. Software testing involves verification and validation (V&V) of the software whereas debugging looks into actual cause behind the defect and corrects it.

## **What are the benefits of Software Testing?**

Here are the benefits of using software testing:

* **Cost-Effective:**It is one of the important advantages of software testing. Testing any IT project on time helps you to save your money for the long term. In case if the bugs caught in the earlier stage of software testing, it costs less to fix.
* **Security:**It is the most vulnerable and sensitive benefit of software testing. People are looking for trusted products. It helps in removing risks and problems earlier.
* **Product quality:**It is an essential requirement of any software product. Testing ensures a quality product is delivered to customers.
* **Customer Satisfaction:**The main aim of any product is to give satisfaction to their customers. UI/UX Testing ensures the best user experience.