

## Non-functional testing includes:

- ✚ Reliability testing
- ✚ Usability testing
- ✚ Efficiency testing
- ✚ Maintainability testing
- ✚ Portability testing
- ✚ Baseline testing
- ✚ Compliance testing
- ✚ Documentation testing
- ✚ Endurance testing
- ✚ Load testing
- ✚ Performance testing
- ✚ Compatibility testing
- ✚ Security testing
- ✚ Scalability testing
- ✚ Volume testing
- ✚ Stress testing
- ✚ Recovery testing
- ✚ Internationalization testing and Localization testing

# Reliability testing

- ✚ Reliability Testing is about exercising an application so that failures are discovered and removed before the system is deployed. The purpose of reliability testing is to determine product reliability, and to determine whether the software meets the customer's reliability requirements.
- ✚ According to ANSI, Software Reliability is defined as: the probability of failure-free software operation for a specified period of time in a specified environment. Software Reliability is not a direct function of time. Electronic and mechanical parts may become “old” and wear out with time and usage, but software will not rust or wear-out during its life cycle. Software will not change over time unless intentionally changed or upgraded.
- ✚ Reliability refers to the consistency of a measure. A test is considered reliable if we get the same result repeatedly. Software Reliability is the probability of failure-free software operation for a specified period of time in a specified environment. Software Reliability is also an important factor affecting system reliability.

- ✚ Reliability testing will tend to uncover earlier those failures that are most likely in actual operation, thus directing efforts at fixing the most important faults.
- ✚ Reliability testing may be performed at several levels. Complex systems may be tested at component, circuit board, unit, assembly, subsystem and system levels.

Software reliability is a key part in software quality. The study of software reliability can be categorized into three parts:




1. Modeling
2. Measurement
3. Improvement

**Modeling:** Software reliability modeling has matured to the point that meaningful results can be obtained by applying suitable models to the problem. There are many models exist, but no single model can capture a necessary amount of the software characteristics. Assumptions and abstractions must be made to simplify the problem. There is no single model that is universal to all the situations.




**Measurement:** Software reliability measurement is naive. Measurement is far from commonplace in software, as in other engineering field. “How good is the software, quantitatively?” As simple as the question is, there is still no good answer. Software reliability can not be directly measured, so other related factors are measured to estimate software reliability and compare it among products. Development process, faults and failures found are all factors related to software reliability.

**Improvement:** Software reliability improvement is hard. The difficulty of the problem stems from insufficient understanding of software reliability and in general, the characteristics of software. Until now there is no good way to conquer the complexity problem of software. Complete testing of a moderately complex software module is infeasible. Defect-free software product can not be assured. Realistic constraints of time and budget severely limits the effort put into software reliability improvement.

# Usability testing

-  In usability testing basically the testers tests the ease with which the user interfaces can be used. It tests that whether the application or the product built is user-friendly or not.
-  Usability Testing is a black box testing technique.
-  Usability testing also reveals whether users feel comfortable with your application or Web site according to different parameters – the flow, navigation and layout, speed and content – especially in comparison to prior or similar applications.

## Usability Testing tests the following features of the software:

-  How easy it is to use the software
-  How easy it is to learn the software
-  How convenient is the software to end user

## Usability testing includes the following five components:

**Learnability:** How easy is it for users to accomplish basic tasks the first time they encounter the design?





**Efficiency:** How fast can experienced users accomplish tasks?

**Memorability:** When users return to the design after a period of not using it, does the user remember enough to use it effectively the next time, or does the user have to start over again learning everything?

**Errors:** How many errors do users make, how severe are these errors and how easily can they recover from the errors?

**Satisfaction:** How much does the user like using the system?

## Benefits of usability testing to the end user or the customer:

-  Better quality software
-  Software is easier to use
-  Software is more readily accepted by users
-  Shortens the learning curve for new users

Usability testing is a very wide area of testing and it needs fairly high level of understanding of this field along with creative mind. People involved in the usability testing are required to possess skills like patience, ability to listen to the suggestions, openness to welcome any idea, and the most important of them all is that they should have good observation skills to spot and fix the issues or problems.



# Advantages of usability testing:

- ✚ Usability test can be modified to cover many other types of testing such as functional testing, system integration testing, unit testing, smoke testing etc.
- ✚ Usability testing can be very economical if planned properly, yet highly effective and beneficial.
- ✚ If proper resources (experienced and creative testers) are used, usability test can help in fixing all the problems that user may face even before the system is finally released to the user. This may result in better performance and a standard system.
- ✚ Usability testing can help in discovering potential bugs and potholes in the system which generally are not visible to developers and even escape the other type of testing.

# Efficiency testing




Efficiency testing test the amount of code and testing resources required by a program to perform a particular function. Software Test Efficiency is number of test cases executed divided by unit of time (generally per hour).

It is internal in the organization how much resources were consumed how much of these resources were utilized.

Here are some formulas to calculate **Software Test Efficiency** (for different factors):

- ✚ Test efficiency = (total number of defects found in unit+integration+system) / (total number of defects found in unit+integration+system+User acceptance testing)
- ✚ Testing Efficiency = (No. of defects Resolved / Total No. of Defects Submitted)\* 100

## Software Test Effectiveness covers three aspects:

-  How much the customer's requirements are satisfied by the system
-  How well the customer specifications are achieved by the system
-  How much effort is put in developing the system

# Maintainability testing

It basically defines that how easy it is to maintain the system. This means that how easy it is to analyze, change and test the application or product.

Maintainability testing shall use a model of the maintainability requirements of the software/system. The maintainability testing shall be specified in terms of the effort required to effect a change under each of the following four categories:

- ✚ **Corrective maintenance** – Correcting problems. The maintainability of a system can be measured in terms of the time taken to diagnose and fix problems identified within that system.

- ✚ **Perfective maintenance** – Enhancements. The maintainability of a system can also be measured in terms of the effort taken to make required enhancements to that system. This can be tested by recording the time taken to achieve a new piece of identifiable functionality such as a change to the database, etc. A number of similar tests should be run and an average time calculated. The outcome will be that it is possible to give an average effort required to implement specified functionality. This can be compared against a target effort and an assessment made as to whether requirements are met.
- ✚ **Adaptive maintenance** – Adapting to changes in environment. The maintainability of a system can also be measured in terms on the effort required to make required adaptations to that system. This can be measured in the way described above for perfective maintainability testing.
- ✚ **Preventive maintenance** – Actions to reduce future maintenance costs. This refers to actions to reduce future maintenance costs.

# Portability testing

It refers to the process of testing the ease with which a computer software component or application can be moved from one environment to another, e.g. moving of any application from Windows 2000 to Windows XP. This is usually measured in terms of the maximum amount of effort permitted. Results are measured in terms of the time required to move the software and complete the and documentation updates.

Being able to move software from one machine platform to another either initially or from an existing environment. It refers to system software or application software that can be recompiled for a different platform or to software that is available for two or more different platforms.

The iterative and incremental development cycle implies that portability testing is regularly performed in an iterative and incremental manner.

Portability testing must be automated if adequate regression testing is to occur.

## **The objectives of Portability testing are to:**

- ✚ Partially validate the system (i.e., to determine if it fulfills its portability requirements):
  - ✚ Determine if the system can be ported to each of its required environments:
    - ✚ Hardware ram and disk space
    - ✚ Hardware processor and processor speed
    - ✚ Monitor resolution
    - ✚ Operating system make and version
    - ✚ Browser make and version
- ✚ Determine if the look and feel of the webpages is similar and functional in the various browser types and their versions.

- ✚ Cause failures concerning the portability requirements that help identify defects that are not efficiently found during unit and integration testing.
- ✚ Report these failures to the development teams so that the associated defects can be fixed.
- ✚ Help determine the extent to which the system is ready for launch.
- ✚ Help provide project status metrics (e.g., percentage of use case paths successfully tested).
- ✚ Provide input to the defect trend analysis effort.



## Portability tests include tests for:

**Installability:** Installability testing is conducted on the software used to install other software on its target environment.

**Co-existence or compatibility:** Co-existence is the software product's capability to co-exists with other independent software products in a common environments sharing common resources.

**Adaptability:** Adaptability is the capability of the software product to be adapted to different specified environments without applying actions or means other than those provided for this purpose for the system.

**Replaceability:** Replaceability is the capability of the product to be used in place of another specified product for the same purpose in the same environment.

Examples of portability testing of an application that is to be portable across multiple:

- ✚ **Hardware platforms**

(including clients, servers, network connectivity devices, input devices, and output devices)

- ✚ **Operating systems**

(including versions and service packs)

- ✚ **Browsers**

(including both types and versions)

# Baseline testing

- ✚ It is one of the type of non-functional testing
- ✚ It refers to the validation of documents and specifications on which test cases would be designed. The requirement specification validation is baseline testing
- ✚ Generally a baseline is defined as a line that forms the base for any construction or for measurement, comparisons or calculations.
- ✚ Baseline testing also helps a great deal in solving most of the problems that are discovered. A majority of the issues are solved through baseline testing.

# Compliance testing

- ✚ It is a type of non-functional software testing.
- ✚ It is related with the IT standards followed by the company and it is the testing done to find the deviations from the company prescribed standards.
- ✚ It determines, whether we are implementing and meeting the defined standards.
- ✚ We should take care while doing this testing. Is there any drawbacks in standards implementation in our project and need to do analysis to improve the standards.
- ✚ Its basically an audit of a system carried out against a known criterion.

# Documentation testing

- ✚ Any written or pictorial information describing, defining, specifying, reporting, or certifying activities, requirements, procedures, or results'. Documentation is as important to a product's success as the product itself. If the documentation is poor, non-existent, or wrong, it reflects on the quality of the product and the vendor.
- ✚ As per the IEEE Documentation describing plans for, or results of, the testing of a system or component, Types include test case specification, test incident report, test log, test plan, test procedure, test report. Hence the testing of all the above mentioned documents is known as documentation testing.
- ✚ This is one of the most cost effective approaches to testing. If the documentation is not right: there will be major and costly problems. The documentation can be tested in a number of different ways to many different degrees of complexity. These range from running the documents through a spelling and grammar checking device, to manually reviewing the documentation to remove any ambiguity or inconsistency.
- ✚ Documentation testing can start at the very beginning of the software process and hence save large amounts of money, since the earlier a defect is found the less it will cost to be fixed.

# Endurance testing

- ✚ It is also known as Soak testing.
- ✚ Endurance testing involves testing a system with a significant load extended over a significant period of time, to discover how the system behaves under sustained use. For example, in software testing, a system may behave exactly as expected when tested for 1 hour but when the same system is tested for 3 hours, problems such as memory leaks cause the system to fail or behave randomly.
- ✚ The goal is to discover how the system behaves under sustained use. That is, to ensure that the throughput and/or response times after some long period of sustained activity are as good or better than at the beginning of the test.
- ✚ It is basically used to check the memory leaks.

# Load testing

- ✚ A load test is type of software testing which is conducted to understand the behavior of the application under a specific expected load.
- ✚ Load testing is performed to determine a system's behavior under both normal and at peak conditions.
- ✚ It helps to identify the maximum operating capacity of an application as well as any bottlenecks and determine which element is causing degradation. E.g. If the number of users are increased then how much CPU, memory will be consumed, what is the network and bandwidth response time.
- ✚ Load testing can be done under controlled lab conditions to compare the capabilities of different systems or to accurately measure the capabilities of a single system.

- ✚ Load testing involves simulating real-life user load for the target application. It helps you determine how your application behaves when multiple users hit it simultaneously.
- ✚ Load testing differs from stress testing, which evaluates the extent to which a system keeps working when subjected to extreme work loads or when some of its hardware or software has been compromised.
- ✚ The primary goal of load testing is to define the maximum amount of work a system can handle without significant performance degradation.



## Examples of load testing include:

- ✚ Downloading a series of large files from the internet
- ✚ Running multiple applications on a computer or server simultaneously
- ✚ Assigning many jobs to a printer in a queue
- ✚ Subjecting a server to a large amount of traffic
- ✚ Writing and reading data to and from a hard disk continuously

# Performance testing

- ✚ Performance testing is testing that is performed, to determine how fast some aspect of a system performs under a particular workload
- ✚ It can serve different purposes like it can demonstrate that the system meets performance criteria
- ✚ It can compare two systems to find which performs better. Or it can measure what part of the system or workload causes the system to perform badly
- ✚ This process can involve quantitative tests done in a lab, such as measuring the response time or the number of MIPS (millions of instructions per second) at which a system functions

## Why to do performance testing:

- ✚ Improve user experience on sites and web apps
- ✚ Increase revenue generated from websites
- ✚ Gather metrics useful for tuning the system
- ✚ Identify bottlenecks such as database configuration
- ✚ Determine if a new release is ready for production
- ✚ Provide reporting to business stakeholders regarding performance against expectations

# Compatibility testing

- ✚ Compatibility testing is a type of software testing used to ensure compatibility of the system/application/website built with various other objects such as other web browsers, hardware platforms, users (in case if it's very specific type of requirement, such as a user who speaks and can read only a particular language), operating systems etc. This type of testing helps find out how well a system performs in a particular environment that includes hardware, network, operating system and other software etc.
- ✚ It is basically the testing of the application or the product built with the computing environment.
- ✚ It tests whether the application or the software product built is compatible with the hardware, operating system, database or other system software or not

# Security testing

- ✚ It is a type of non-functional testing.
- ✚ Security testing is basically a type of software testing that's done to check whether the application or the product is secured or not. It checks to see if the application is vulnerable to attacks, if anyone hack the system or login to the application without any authorization.
- ✚ It is a process to determine that an information system protects data and maintains functionality as intended.
- ✚ The security testing is performed to check whether there is any information leakage in the sense by encrypting the application or using wide range of software's and hardware's and firewall etc.
- ✚ Software security is about making software behave in the presence of a malicious attack.
- ✚ The six basic security concepts that need to be covered by security testing are: confidentiality, integrity, authentication, availability, authorization and non-repudiation

# Stress testing

- ✚ It involves testing beyond normal operational capacity, often to a breaking point, in order to observe the results.
- ✚ It is a form of software testing that is used to determine the stability of a given system.
- ✚ It put greater emphasis on robustness, availability, and error handling under a heavy load, rather than on what would be considered correct behavior under normal circumstances.
- ✚ The goals of such tests may be to ensure the software does not crash in conditions of insufficient computational resources (such as memory or disk space).

Very simply we can put the difference between Volume, Load and stress testing as:

- ✚ **Volume Testing** = Large amount of data
- ✚ **Load Testing** = Large number of users
- ✚ **Stress Testing** = Too many users, too much data, too little time and too little space