

# What are Software Test Types?

Test types are introduced as a means of clearly defining the objective of a certain level for a program or project. A test type is focused on a particular test objective, which could be the testing of the function to be performed by the component or system; a non-functional quality characteristics, such as reliability or usability; the structure or architecture of the component or system; or related to changes, i.e confirming that defects have been fixed (confirmation testing or retesting) and looking for unintended changes (regression testing). Depending on its objectives, testing will be organized differently. Hence there are four software test types:

1. [Functional testing](#)
2. [Non-functional testing](#)
3. [Structural testing](#)

## What is Functional testing (Testing of functions) in software?

In functional testing basically the testing of the functions of component or system is done. It refers to activities that verify a specific action or function of the code. Functional test tends to answer the questions like “can the user do this” or “does this particular feature work”. This is typically described in a requirements specification or in a functional specification.

The techniques used for functional testing are often specification-based. Testing functionality can be done from two perspective:

- **Requirement-based testing:** In this type of testing the requirements are prioritized depending on the risk criteria and accordingly the tests are prioritized. This will ensure that the most important and most critical tests are included in the testing effort.
- **Business-process-based testing:** In this type of testing the scenarios involved in the day-to-day business use of the system are described. It uses the knowledge of the business processes. For example, a personal and payroll system may have the business process along the lines of: someone joins the company, employee is paid on the regular basis and employee finally leaves the company.

## What is Non-functional testing (Testing of software product characteristics)?

In non-functional testing the quality characteristics of the component or system is tested. Non-functional refers to aspects of the software that may not be related to a specific function or user action such as scalability or security. Eg. How many people can log in at once? Non-functional testing is also performed at all levels like functional testing.

Non-functional testing includes:

- Functionality testing
- 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
- **Functionality testing:** Functionality testing is performed to verify that a software application performs and functions correctly according to design specifications. During functionality testing we check the core application functions, text input, menu functions and installation and setup on localized machines, etc.
  - **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.
  - **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 includes the following five components:

1. **Learnability:** How easy is it for users to accomplish basic tasks the first time they encounter the design?
  2. **Efficiency:** How fast can experienced users accomplish tasks?
  3. **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?
  4. **Errors:** How many errors do users make, how severe are these errors and how easily can they recover from the errors?
  5. **Satisfaction:** How much does the user like using the system?
- **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).
  - **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.
  - **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.
  - **Baseline testing:** It refers to the validation of documents and specifications on which test cases would be designed. The requirement specification validation is baseline testing.

- **Compliance 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.
- **Documentation testing:** 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.
- **Endurance 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.
- **Load testing:** A load test is usually 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
- **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.
- **Compatibility testing:** Compatibility testing 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:** Security testing is basically to check that whether the application or the product is secured or not. Can anyone come tomorrow and hack the system or login the application without any authorization. It is a process to determine that an information system protects data and maintains functionality as intended.
- **Scalability testing:** It is the testing of a software application for measuring its capability to scale up in terms of any of its non-functional capability like load supported, the number of transactions, the data volume etc.
- **Volume testing:** Volume testing refers to testing a software application or the product with a certain amount of data. E.g., if we want to volume test our application with a specific database size, we need to expand our database to that size and then test the application's performance on it.
- **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 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).
- **Recovery testing:** Recovery testing is done in order to check how fast and better the application can recover after it has gone through any type of crash or hardware failure etc. Recovery testing is the forced failure of the software in a variety of ways to verify that recovery is properly performed. For example, when an application is receiving data from a network, unplug the connecting cable. After some time, plug the cable back in and analyze the application's ability to continue receiving data from the point at which the network connection

got disappeared. Restart the system while a browser has a definite number of sessions and check whether the browser is able to recover all of them or not.

- **Internationalization testing and Localization testing:** Internationalization is a process of designing a software application so that it can be adapted to various languages and regions without any changes. Whereas Localization is a process of adapting internationalized software for a specific region or language by adding local specific components and translating text.

## What is functionality testing in software?

Functionality testing is performed to verify that a software application performs and functions correctly according to design specifications. During functionality testing we check the core application functions, text input, menu functions and installation and setup on localized machines, etc.

The following is needed to be checked during the functionality testing:

- Installation and setup on localized machines running localized operating systems and local code pages.
- Text input, including the use of extended characters or non-Latin scripts.
- Core application functions.
- String handling, text, and data, especially when interfacing with non-Unicode applications or modules.
- Regional settings defaults.
- Text handling (such as copying, pasting, and editing) of extended characters, special fonts, and non-Latin scripts.
- Accurate hot-key shortcuts without any duplication.

Functionality testing verifies that an application is still fully functional after localization. Even applications which are professionally internationalized according to world-readiness guidelines require functionality testing.

## What is reliability testing in software?

- 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

1. **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.

2. **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.

3. **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.

## What is Usability testing in software and it’s benefits to end user?

- 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:

1. **Learnability:** How easy is it for users to accomplish basic tasks the first time they encounter the design?
2. **Efficiency:** How fast can experienced users accomplish tasks?
3. **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?
4. **Errors:** How many errors do users make, how severe are these errors and how easily can they recover from the errors?
5. **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

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.

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.

## What is Efficiency testing in software?

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.

## What is Maintainability testing in software?

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.

## What is Portability testing in software?

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).

## What is Baseline testing in software?

- 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.

## What is Compliance testing in software 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.

## What is documentation testing in software testing?

Documentation testing is a non-functional type of [software testing](#).

- It is a type of [non-functional 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.

## What is Endurance testing in software testing?

Endurance testing is a non functional type of [software testing](#).

- It is a type of non-functional 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.

## What is Load testing in software testing?

- Load testing is a type of [non-functional 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 hits 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.

## What is Performance testing in software?

- It is a type of non-functional 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

## What is Compatibility testing in software testing?

- It is a type of non-functional 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.

## What is Security testing in software 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.

## What is Scalability testing in software testing?

- It is a type of non-functional testing.
- It is a type of [software testing](#) that test the ability of a system, a network, or a process to continue to function well, when it is changed in size or volume in order to meet a growing need.
- It is the testing of a software application for measuring its capability to scale up in terms of any of its non-functional capability like load supported, the number of transactions, the data volume etc.

## What is Volume testing in software testing?

- It is a type of non-functional testing.
- Volume testing refers to testing a software application or the product with a certain amount of data. E.g., if we want to volume test our application with a specific database size, we need to expand our database to that size and then test the application's performance on it.
- "Volume testing" is a term given and described in Glenford Myers' *The Art of [Software Testing](#)*, 1979. Here's his definition: **"Subjecting the program to heavy volumes of data. The purpose of volume testing is to show that the program cannot handle the volume of data specified in its objectives"** – p. 113.
- The purpose of **volume testing** is to determine system performance with increasing volumes of data in the database.

## What is Stress testing in software testing?

- It is a type of [non-functional 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).

## Difference between Volume, Load and stress testing in software

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

Volume Testing = Large amounts of data

Load Testing = Large amount of users

Stress Testing = Too many users, too much data, too little time and too little room

## What is Recovery testing in software?

- It is a type of non-functional testing.

- Recovery testing is done in order to check how fast and better the application can recover after it has gone through any type of crash or hardware failure etc.
- Recovery testing is the forced failure of the software in a variety of ways to verify that recovery is properly performed. For example, when an application is receiving data from a network, unplug the connecting cable. After some time, plug the cable back in and analyze the application's ability to continue receiving data from the point at which the network connection got disappeared. Restart the system while a browser has a definite number of sessions and check whether the browser is able to recover all of them or not.

## What is Internationalization testing and Localization testing in software?

- It is a type of non-functional testing.
- Internationalization is a process of designing a software application so that it can be adapted to various languages and regions without any changes.
- Whereas Localization is a process of adapting internationalized software for a specific region or language by adding local specific components and translating text.

## What is Confirmation testing in software?

**Confirmation testing or re-testing:** When a test fails because of the defect then that defect is reported and a new version of the software is expected that has had the defect fixed. In this case we need to execute the test again to confirm that whether the defect got actually fixed or not. This is known as confirmation testing and also known as re-testing. It is important to ensure that the test is executed in exactly the same way it was the first time using the same inputs, data and environments.

Hence, when the change is made to the defect in order to fix it then confirmation testing or re-testing is helpful.

## What is Regression testing in software?

**Regression testing:** During confirmation testing the defect got fixed and that part of the application started working as intended. But there might be a possibility that the fix may have introduced or uncovered a different defect elsewhere in the software. The way to detect these '**unexpected side-effects**' of fixes is to do regression testing. The purpose of a regression testing is to verify that modifications in the software or the environment have not caused any unintended adverse side effects and that the system still meets its requirements. Regression testing are mostly automated because in order to fix the defect the same test is carried out again and again and it will be very tedious to do it manually. Regression tests are executed whenever the software changes, either as a result of fixes or new or changed functionality.

## What is Structural testing (Testing of software structure/architecture)?

- The structural testing is the testing of the structure of the system or component.
- Structural testing is often referred to as 'white box' or 'glass box' or 'clear-box testing' because in structural testing we are interested in what is happening 'inside the system/application'.

- In structural testing the testers are required to have the knowledge of the internal implementations of the code. Here the testers require knowledge of how the software is implemented, how it works.
- During structural testing the tester is concentrating on how the software does it. For example, a structural technique wants to know how loops in the software are working. Different test cases may be derived to exercise the loop once, twice, and many times. This may be done regardless of the functionality of the software.
- Structural testing can be used at all levels of testing. Developers use structural testing in component testing and component integration testing, especially where there is good tool support for code coverage. Structural testing is also used in system and acceptance testing, but the structures are different. For example, the coverage of menu options or major business transactions could be the structural element in system or acceptance testing.

## What is Maintenance Testing?

Once a system is deployed it is in service for years and decades. During this time the system and its operational environment is often corrected, changed or extended. Testing that is provided during this phase is called maintenance testing.

Usually maintenance testing is consisting of two parts:

- **First** one is, testing the changes that has been made because of the correction in the system or if the system is extended or because of some additional features added to it.
- **Second** one is regression tests to prove that the rest of the system has not been affected by the maintenance work.

## What is Impact analysis in software testing?

Impact analysis is basically analyzing the impact of the changes in the deployed application or product.

It tells us about the parts of the system that may be unintentionally got affected because of the change in the application and therefore need careful regression testing. This decision is taken together with the stakeholders.