

Functional testing types

(1) Smoke testing.

Smoke Testing is a software testing process that determines whether the deployed software build is stable or not. Smoke testing is a confirmation for QA team to proceed with further software testing. It consists of a minimal set of tests run on each build to test software functionalities. Smoke testing is also known as “Build Verification Testing” or “Confidence Testing.”

For Example, a typical smoke test would be – Verify that the application launches successfully, Check that the GUI is responsive ... etc.

(2) Sanity Testing

Sanity Testing is a subset of regression testing. This is done prior to a release as final inspection and after applying minor patches. It normally includes a set of core tests of basic GUI functionality to demonstrate connectivity to the database, application servers, printers etc

Sanity testing helps in quickly identify defects in the core functionality. It can be carried out in lesser time as no documentation is required for sanity testing. If the defects are found during sanity testing, project is rejected that is helpful in saving time for execution of regression tests.

Example of Sanity Testing:

In an e-commerce project, main modules are login page, home page, user profile page, user

registration etc. There is a defect in the login page when the password field accepts less than four alpha numeric characters and the requirement mentions that this password field should not be below eight characters. Hence, the defect is reported by the testing team to the development team to resolve it. Then the development team fixes the reported defect and sends it to the testing team for clearance. Then the testing team checks whether the changes done are working fine or not. It is also determined if it does have an impact on other related functionalities. Now there is a functionality to update the password in the user profile page. As part of the sanity testing, login page is validated as well as the profile page to ensure that the checks are working fine at both the places.

Features of Sanity Testing:

-Subset of Regression Testing:

Sanity testing is a subset of regression testing and focuses on the smaller section of the application.

-Unscripted:

Most of the times sanity testing is not scripted.

Not documented:

Usually sanity testing is undocumented.

Narrow and deep:

Sanity testing is narrow and deep approach of testing where limited functionalities are covered deeply.

Performed by testers:

Sanity testing is normally performed by testers.

(3) Regression Testing

Regression Testing is a type of software testing executed to check whether a code change has not unfavorably disturbed current features & functions of an Application.

In other words, regression testing means re-executing test cases that have been cleared in the past against the new version to ensure that the app's functionalities are working correctly. Moreover, regression testing is a series of tests and not a single test performed whenever you add a new code.

(5) Retesting

Retesting is a process to check specific failed test cases that are found with bug/s in the final execution.

Generally, testers find these bugs while testing the software application and assign it to the developers to fix it. Then the developers fix the bug/s and assign it back to the testers for verification. This continuous process is called Retesting.

Where retesting differs from regression testing is that, instead of being designed to search through all the previous updates and features of the software to find unforeseen defects and bugs, retesting is designed to test specific defects that you've already detected (typically during your regression testing).

(6) Mutation Testing

Mutation Testing is a type of software testing in which certain statements of the source code are changed/mutated to check if the test cases are able to find errors in source code.

The goal of Mutation Testing is ensuring the quality of test cases in terms of robustness that it should fail the mutated source coding/Error Seeding.

(7) Adhoc Testing :

This testing requires no documentation or any specific procedure to be followed. Since this testing targets at discovering defects via a random approach, except any documentation, defects will no longer be mapped to check cases. You can use this testing to randomly test any part of the application.

(8) Exploratory Testing :

It is a kind of software testing that targets to optimize and enhance the software program in every viable way. In this kind of testing the tester is free to choose any feasible methodology to check the software. It is an unscripted strategy for software program testing. In exploratory testing, software program developers use their private learning, knowledge, abilities, and competencies to check the software program developed by means of themselves.

(9) Gorilla Testing

This is used to test the user interface of the software package. One can apply this technique to test the GUI, navigation sequences, message displays etc. If the software is given to a gorilla, it will play around the key board resulting in keys getting pressed randomly. This is the approach used in this technique.

Non-Functional Testing Types

(1) Performance Testing

It is a software testing process used for testing the speed, response time, stability, reliability, scalability and resource

usage of a software application under particular workload. The main purpose of performance testing is to identify and eliminate the performance bottlenecks in the software application.

The first step in performance testing is determining what factors matter most to the customer, like

Response-time of critical transactions

System throughput

Peak Loads

System performance is generally assessed in terms of response times and throughput rates under differing processing and configuration conditions.

(2) Volume Testing

it is a type of Software Testing, where the software is subjected to a huge volume of data. It is also referred to as flood testing. Volume testing is done to analyze the system performance by increasing the volume of data in the database.

With the help of Volume testing, the impact on response time and system behavior can be studied when exposed to a high volume of data.

For example, testing the music site behavior when there are millions of user to download the song

(3)Capacity Testing

Capacity Testing ensures that the application and environment can smoothly handle the maximum number of users or transactions according to the performance requirements defined in your Service-Level Agreement (SLA). Capacity Testing is aimed at testing the maximum capacity of your system in terms of traffic, while still being able to deliver optimal user experience.

(4)Scalability Testing

It is a non functional testing method that measures performance of a system or network when the number of user requests are scaled up or down. The purpose of Scalability testing is to ensure that the system can handle projected increase in user traffic, data volume, transaction counts frequency, etc. It tests system ability to meet the growing needs.

(5)Security Testing

Security Testing is a type of Software Testing that uncovers

vulnerabilities, threats, risks in a software application and prevents malicious attacks from intruders. The purpose of Security Tests is to identify all possible loopholes and weaknesses of the software system which might result in a loss of information, revenue, reputa at the hands of the employees or outsiders of the Organization.

Examples

- A password should be in encrypted format
- Application or System should not allow invalid users
- Check cookies and session time for application
- For financial sites, the Browser back button should not work.

(6)Compatibility Testing:

It is performed on an application to check its compatibility (running capability) on different platform/environments.

This testing is done only when the application becomes stable.Means simply this compatibility test aims to check the developed software application functionality on various software, hardware platforms, network and browser etc.

This compatibility testing is very important in product

production and implementation point of view as it is performed to avoid future issues regarding compatibility.

(7)Usability Testing

Usability Testing also known as User Experience(UX) Testing, is a testing method for measuring how easy and user-friendly a software application is. A small set of target end-users, use software application to expose usability defects. Usability testing mainly focuses on user's ease of using application, flexibility of application to handle controls and ability of application to meet its objectives. This testing is recommended during the initial design phase of SDLC, which gives more visibility on the expectations of the users.

(8)Recovery Testing

Recovery Testing is software testing technique which verifies software's ability to recover from failures like software/hardware crashes, network failures etc. The purpose of Recovery Testing is to determine whether software operations can be continued after disaster or integrity loss. Recovery testing involves reverting back software to the point where integrity was known and reprocessing transactions to the failure point.

Software Development Life Cycle (SDLC)

Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality softwares. The SDLC aims to produce a highquality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

SDLC is the acronym of Software Development Life Cycle.

It is also called as Software Development Process. SDLC is a framework defining tasks performed at each step in the software development process.



The stages of SDLC are as follows:

(1) Stage 1: Planning and requirement analysis

Requirement Analysis is the most important and necessary stage in SDLC. The senior members of the team perform it with inputs from all the stakeholders and domain experts or SMEs (Small and mid-size enterprises) in the industry.

Planning for the quality assurance requirements and

identifications of the risks associated with the projects is also done at this stage.

Business analyst and Project organizer set up a meeting with the client to gather all the data like what the customer wants to build, who will be the end user, what is the objective of the product. Before creating a product, a core understanding or knowledge of the product is very necessary.

For Example, A client wants to have an application which concerns money transactions. In this method, the requirement has to be precise like what kind of operations will be done, how it will be done, in which currency it will be done, etc.

Once the required function is done, an analysis is complete with auditing the feasibility of the growth of a product. In case of any ambiguity, a signal is set up for further discussion.

Once the requirement is understood, the SRS (Software Requirement Specification) document is created. The developers should thoroughly follow this document and also should be reviewed by the customer for future reference.

(2)Stage2: Defining Requirements

Once the requirement analysis is done, the next stage is to certainly represent and document the software requirements and get them accepted from the project stakeholders.

This is accomplished through "SRS"- Software Requirement Specification document which contains all the product requirements to be constructed and developed during the project life cycle.

(3)Stage3: Designing the Software

The next phase is about to bring down all the knowledge of requirements, analysis, and design of the software project.

This phase is the product of the last two, like inputs from the customer and requirement gathering.

(4)Stage4: Developing the project

In this phase of SDLC, the actual development begins, and the programming is built. The implementation of design begins concerning writing code. Developers have to follow the coding guidelines described by their management and programming tools like compilers, interpreters, debuggers, etc. are used to develop and implement the code.

(5)Stage5: Testing

After the code is generated, it is tested against the requirements to make sure that the products are solving the needs addressed and gathered during the requirements stage.

During this stage, unit testing, integration testing, system testing, acceptance testing are done.

(6)Stage6: Deployment

Once the software is certified, and no bugs or errors are stated, then it is deployed.

Then based on the assessment, the software may be released as it is or with suggested enhancement in the object segment.

After the software is deployed, then its maintenance begins.

(7)Stage7: Maintenance

Once when the client starts using the developed systems,

then the real issues come up and requirements to be solved from time to time.

This procedure where the care is taken for the developed product is known as maintenance.