Software Testing Types

-

Given below are few types of Software Testing.

#1) Alpha Testing

It is the most common type of testing used in software industry. The objective of this testing is to identify all possible issues or defects before releasing it into the market or to the user. Alpha testing is carried out at the end of the software development phase but before the Beta Testing. Still, minor design changes may be made as a result of such testing. Alpha testing is conducted at the developer's site. In-house virtual user environment can be created for this type of testing.

[image credit krify.co]

#2) Acceptance Testing

An acceptance test is performed by the client and verifies whether the end to end the flow of the system is as per the business requirements or not and if it is as per the needs of the end user. Client accepts the software only when all the features and functionalities work as expected. It is the last phase of the testing, after which the software goes into production. This is also called as User Acceptance Testing (UAT).

#3) Ad-hoc Testing

The name itself suggests that this testing is performed on ad-hoc basis i.e. with no reference to test case and also without any plan or documentation in place for such type of testing. The objective of this testing is to find the defects and break the application by executing any flow of the application or any random functionality.

Ad-hoc testing is an informal way of finding defects and can be performed by anyone in the project. It is difficult to identify defects without test case but sometimes it is possible that defects found during ad-hoc testing might not have been identified using existing test cases.

#4) Accessibility Testing

The aim of accessibility testing is to determine whether the software or application is accessible for disabled people or not. Here disability means deaf, color blind, mentally disabled, blind, old age and other disabled groups. Various checks are performed such as font size for visually disabled, color and contrast for color blindness etc.

#5) Beta Testing

-

It is a formal type of software testing which is carried out by the customer. It is performed in **Real Environment** before releasing the product into the market for the actual end users. Beta testing is carried to ensure that there are no major failures in the software or product and it satisfies the business requirements from an end-user perspective. Beta testing is successful when the customer accepts the software.

Usually, this testing is typically done by end-users or others. It is the final testing done before releasing an application for commercial purpose.Usually, the Beta version of the software or product released is limited to a certain number of users in a specific area. So end user actually uses the software and shares the feedback to the company. Company then takes necessary action before releasing the software to the worldwide.

#6) Back-end Testing

Whenever an input or data is entered on front-end application, it stores in the database and the testing of such database is known as Database Testing or Backend testing. There are different databases like SQL Server, MySQL, and Oracle etc. Database testing involves testing of table structure, schema, stored procedure, data structure and so on.

In back-end testing GUI is not involved, testers are directly connected to the database with proper access and testers can easily verify data by running few queries on the database. There can be issues identified like data loss, deadlock, data corruption etc during this back-end testing and these issues are critical to fixing before system goes live into the production environment

#7) Browser Compatibility Testing

It is a subtype of Compatibility Testing(which is explained below) and is performed by the testing team. Browser Compatibility Testing is performed for web applications and it ensures that the software can run with the combination of different browser and operating system. This type of testing also validates whether web application runs on all versions of all browsers or not.

#8) Backward Compatibility Testing

It is a type of testing which validates whether the newly developed software or updated software works well with older version of the environment or not. Backward Compatibility Testing checks whether the new version of the software works properly with file format created by older version of the software; it also works well with data tables, data files, data structure created by older version of that software. If any of the software is updated then it should work well on top of the previous version of that software.

#9) Black Box Testing

Internal system design is not considered in this type of testing. Tests are based on the requirements and functionality. Detailed information about the advantages, disadvantages, and types of Black box testing can be seen *here*.

#10) Boundary Value Testing

This type of testing checks the behavior of the application at boundary level. Testing is performed for checking if defects exist at boundary values. Boundary value testing is used for testing a different range of numbers. There is an upper and lower boundary for each range and testing is performed on these boundary values. If testing requires test range of numbers from 1 to 500 then Boundary Value Testing is performed on values at 0, 1, 2, 499, 500 and 501.

#11) Branch Testing

It is a type of white box testing and is carried out during unit testing. Branch Testing, the name itself suggests that the code is tested thoroughly by traversing at every branch.

#12) Comparison Testing

Comparison of a product's strength and weaknesses with its previous versions or other similar products is termed as Comparison Testing.

#13) Compatibility Testing

It is a testing type in which it validates how software behaves and runs in a different environment, web servers, hardware, and network environment. Compatibility testing ensures that software can run on a different configuration, different database, different browsers and their versions. Compatibility testing is performed by the testing team.

#14) Component Testing

It is mostly performed by developers after the completion of unit testing. Component Testing involves testing of multiple functionalities as a single code and its objective is to identify if any defect exists after connecting those multiple functionalities with each other.

#15) End-to-End Testing

Similar to system testing, End-to-end testing involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

#16) Equivalence Partitioning

It is a testing technique and a type of Black Box Testing. During this equivalence partitioning, a set of group is selected and few values or numbers are picked up for testing. It is understood that all values from that group generate the same output. The aim of this testing is to remove redundant test cases within a specific group which generates the same output but not any defect.

Suppose, application accepts values between -10 to +10 so using equivalence partitioning the values picked up for testing are zero, one positive value, one negative value. So the Equivalence Partitioning for this testing is: -10 to -1, 0, and 1 to 10.

#17) Example Testing

It means real-time testing. Example testing includes the real-time scenario, it also involves the scenarios based on the experience of the testers.

#18) Exploratory Testing

Exploratory Testing is an informal testing performed by the testing team. The objective of this testing is to explore the application and looking for defects that exist in the application. Sometimes it may happen that during this testing major defect discovered can even cause system failure. During exploratory testing, it is advisable to keep a track of what flow you have tested and what activity you did before the start of the specific flow.

An exploratory testing technique is performed without documentation and test cases.

#20) Functional Testing

This type of testing ignores the internal parts and focuses only on the output to check if it is as per the requirement or not. It is a Black-box type testing geared to the functional requirements of an application. For detailed information about Functional Testing click here.

#21) Graphical User Interface (GUI) Testing

The objective of this GUI testing is to validate the GUI as per the business requirement. The expected GUI of the application is mentioned in Detailed Design Document and GUI mockup screens. The GUI testing includes size of the buttons and input field present on the screen, alignment of all text, tables and content in the tables.

- It also validates the menu of the application, after selecting different menu and menu items, it validates that the page does not fluctuate and the alignment remains same after hovering the mouse on the menu or sub-menu.

#22) Gorilla Testing

Gorilla Testing is a testing type performed by a tester and sometimes by developer the as well. In Gorilla Testing, one module or the functionality in the module is tested thoroughly and heavily. The objective of this testing is to check the robustness of the application.

#23) Happy Path Testing

The objective of Happy Path Testing is to test an application successfully on a positive flow. It does not look for negative or error conditions. The focus is only on the valid and positive inputs through which application generates the expected output.

#24) Incremental Integration Testing

This is a Bottom-up approach for testing i.e continuous testing of an application when a new functionality is added. Application functionality and modules should be independent enough to test separately. This is done by programmers or by testers.

#25) Install/Uninstall Testing

Testing done on full, partial, or upgrade install/uninstall processes on different operating systems under different hardware or software environment is termed so.

#26) Integration Testing

Testing of all integrated modules to verify the combined functionality after integration is termed so. Modules are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed systems.

#27) Load Testing

It is a type of non-functional testing and the objective of Load testing is to check how much of load or maximum workload a system can handle without any performance degradation. Load testing helps to find the maximum capacity of the system under specific load and any issues that cause the software performance degradation. Load testing is performed using tools like JMeter, LoadRunner, WebLoad, Silk performer etc.

#28) Monkey Testing

-

Monkey testing is carried out by a tester assuming that if monkey uses the application then how random input, values will be entered by the Monkey without any knowledge or understanding of the application. The objective of Monkey Testing is to check if application or system gets crashed by providing random input values/data. Monkey Testing is performed randomly and no test cases are scripted and it is not necessary to

Monkey Testing is performed randomly and no test cases are scripted and it is not necessary to be aware of the full functionality of the system.

#29) Mutation Testing

Mutation Testing is a type of white box testing in which the source code of one of the program is changed and verifies whether the existing test cases can identify these defects in the system. The change in the program source code is very minimal so that it does not impact the entire application, only the specific area having the impact and the related test cases should able to identify those errors in the system.

#30) Negative Testing

Testers having the mindset of "attitude to break" and using negative testing they validate that if system or application breaks. A negative testing technique is performed using incorrect data, invalid data or input. It validates that if the system throws an error of invalid input and behaves as expected.

#31) Non-Functional Testing

It is a type of testing for which every organization having a separate team which usually called as Non-Functional Test (NFT) team or Performance team. NFT testing involves testing of non-functional requirements such as Load Testing, Stress Testing, Security, Volume, Recovery Testing etc. The objective of NFT testing is to ensure whether the response time of software or application is quick enough as per the business requirement.

It should not take much time to load any page or system and should sustain during peak load.

#32) Performance Testing

This term is often used interchangeably with 'stress' and 'load' testing. Performance Testing is done to check whether the system meets the performance requirements. Different performance and load tools are used to do this testing.

#33) Recovery Testing

It is a type of testing which validates that how well the application or system recovers from crashes or disasters. Recovery testing determines if the system is able to continue the operation after a disaster. Assume that application is receiving data through the network cable and suddenly that network cable has been unplugged. Sometime later, plug the network cable; then the system should start receiving data from where it lost the connection due to network cable unplugged.

#34) Regression Testing

Testing an application as a whole for the modification in any module or functionality is termed as Regression Testing. It is difficult to cover all the system in Regression Testing, so typically automation testing tools are used for these types of testing.

#35) Risk-Based Testing (RBT)

In Risk Based Testing, the functionalities or requirements are tested based on their priority. Risk-based testing includes testing of highly critical functionality, which has the highest impact on business and in which the probability of failure is very high. The priority decision is based on the business need, so once priority is set for all functionalities then high priority functionality or test cases are executed first followed by medium and then low priority functionalities.

The low priority functionality may be tested or not tested based on the available time. The Risk-based testing is carried out if there is insufficient time available to test entire software and software needs to be implemented on time without any delay. This approach is followed only by the discussion and approval of the client and senior management of the organization.

#36) Sanity Testing

Sanity Testing is done to determine if a new software version is performing well enough to accept it for a major testing effort or not. If an application is crashing for the initial use then the system is not stable enough for further testing. Hence a build or an application is assigned to fix it.

#37) Security Testing

It is a type of testing performed by a special team of testers. A system can be penetrated by any hacking way. Security Testing is done to check how the software or application or website is secure from internal and external threats. This testing includes how much software is secure from the malicious program, viruses and how secure and strong the authorization and authentication processes are.

It also checks how software behaves for any hackers attack and malicious programs and how software is maintained for data security after such hacker attack.

#38) Smoke Testing

Whenever a new build is provided by the development team then the software testing team validates the build and ensures that no major issue exists. The testing team ensures that build is stable and a detailed level of testing is carried out further. Smoke Testing checks that no show stopper defect exists in the build which will prevent testing team to test the application in detail.

If testers find that the major critical functionality is broken down at the initial stage itself then testing team can reject the build and inform accordingly to the development team. Smoke Testing is carried out to a detailed level of any functional or regression testing.

#39) Static Testing

Static Testing is a type of testing which is executed without any code. The execution is performed on the documentation during the testing phase. It involves reviews, walkthrough, and inspection of the deliverables of the project. Static testing does not execute the code instead of the code syntax, naming conventions are checked.

The static testing is also applicable for test cases, test plan, design document. It is necessary to perform static testing by the testing team as the defects identified during this type of testing are cost-effective from the project perspective.

#40) Stress Testing

This testing is done when a system is stressed beyond its specifications in order to check how and when it fails. This is performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to the system or database load.

#41) System Testing

Under System Testing technique, the entire system is tested as per the requirements. It is a Black-box type testing that is based on overall requirement specifications and covers all the combined parts of a system.

#42) Unit Testing

Testing of an individual software component or module is termed as Unit Testing. It is typically done by the programmer and not by testers, as it requires a detailed knowledge of the internal program design and code. It may also require developing test driver modules or test harnesses.

#43) Usability Testing

Under this testing, User-friendliness check is done. Application flow is tested to know if a new user can understand the application easily or not, Proper help documented if a user gets stuck at any point. Basically, system navigation is checked in this testing.

#44) Vulnerability Testing

The testing which involves identifying of weakness in the software, hardware and the network is known as Vulnerability Testing. Malicious programs, the hacker can take control of the system, if it is vulnerable to such kind of attacks, viruses, and worms. So it is necessary to check if those systems undergo Vulnerability Testing before production. It may identify critical defects, flaws in the security.

#45) Volume Testing

It is a type of non-functional testing performed by the performance testing team. The software or application undergoes a huge amount of data and Volume Testing checks the system behavior and response time of the application when the system came across such high volume of data. This high volume of data may impact the system's performance and speed of the processing time.

#46) White Box Testing

This testing is based on the knowledge about the internal logic of an application's code. It is also known as Glass box Testing. Internal software and code working should be known for performing this type of testing. Under this tests are based on the coverage of code statements, branches, paths, conditions etc.

For more information about white box testing and its types please click here.

Conclusion

The above-mentioned testing types are just a part of testing. However, there is still a list of more than 100+ types of testing, but all testing types are not used in all types of projects. So I have covered some common Types of Software Testing which are mostly used in the testing life cycle.

Also, there are alternative definitions or processes used in different organizations, but the basic concept is same everywhere. These testing types, processes, and their implementation methods keep changing as and when the project, requirements, and scope changes.

Happy Testing!!!

← PREV POST | NEXT POST →

516 comments ↓

#1 **namita**

it was valuable information about types of testing