Click Here



Black Box Testing is a Software testing method in which the internal working of the application is not known to the tester. The Black Box Testing mainly focuses on testing the topics related to the Black Box Testing in detail. What is Black Box Testing? Black-box testing is a Type of Software Testing in which the tester is not concerned with the software's internal knowledge or implementation details but rather focuses on validating the functionality based on the provided specifications or requirements. Black Box TestingTypes Of Black Box knowing the internal code or structure, following are the various Types of Black Box Testing. Functional Testing is a type of Software Testing in which the system is tested against the functional Testing. properly satisfied by the application. This testing is not concerned with the source code of the application. Each functionality of the software application is tested by providing appropriate test input, and comparing the actual output with the expected output. This testing focuses on checking the user interface, APIs, database, security, client or server application, and functionality of the Application Under Test. Functional testing can be manual or automated. It determines the system's software functionality of the Application Under Test. Functional testing can be manual or automated. It determines the system's software functionality of the Application Under Test. everything still works as it should, even after updates or tweaks to the code. This ensures that the software remains reliable and functions properly, maintaining its integrity throughout its development lifecycle. Regression means the return of a bug. It ensures that the newly added code is compatible with the existing code. In other words, a new software update has no impact on the functional TestingNon-functional Testing is a type of Software Testing that is performed to verify the non-functional requirements of the application. It verifies whether the behavior of the system is as per the requirement or not. It tests all the aspects that are not tested in functional testing. It is as important as functional testing. It is also known as NFT. This testing is not functional testing of software. It focuses on the software's performance, usability, and scalability. Advantages of Black Box Testing The tester does not need to have more functional knowledge or programming skills to implement the Black Box Testing. It is efficient for implementing the tests in the larger system. Tests are executed from the user's or client's point of view. Test cases are easily reproducible. It is used to find the ambiguity and contradictions in the functional specifications. Disadvantages of Black Box Testing There is a possibility of repeating the same tests while implementing the testing process. difficult to execute the test cases because of complex inputs at different stages of testing. Sometimes, the reason for the test failure cannot be detected. It does not reveal the errors in the control structure. Working with a large sample space of inputs can be exhaustive and consumes a lot of time. Difference between Black Box Testing and White Box Testing Box Tes Testing is a way of testing the software in which the tester has knowledge about the internal structure or the code or the program of the software, ensuring that it meets the requirements and specifications. White box testing is mainly focused on ensuring that the internal code of the software is correct and efficient. Testing methods Black box testing uses methods like equivalence partitioning, boundary value analysis, and error guessing to create test cases. White box testing uses methods like control flow testing, data flow testing and statement coverage testing. Knowledge levelBlack box testing does not require any knowledge of the internal workings of the software, and can be performed by testers who are not familiar with programming languages. White box testing is generally used for testing the software at the functional level. White box testing is used for testing technique and the unit level, integration level and system level. Grey Box Testing technique and the White Box Testing technique and the unit level, integration level and system level. So the internal structure of the item being tested and in White Box Testing the internal structure is known to the tester. The internal structure is partially known in Gray Box Testing. This includes access to internal data structures and algorithms to design the test cases. Gray Box Testing is named so because the software program is like a semitransparent or gray box inside which the tester can partially see. It commonly focuses on context-specific errors related to web systems. Objectives of Gray Box Testing Done1. Syntax-Driven TestingSyntax-Driven Testing is a Software Engineering technique or approach that is used in functional automation testing. This type of testing is applied to systems that can be syntactically represented by some language. For example, language can be represented by context-free grammar. In this, the test cases are generated so that each grammar rule is used at least once. 2. Equivalence Partitioning (ECP). It is a software testing technique or black-box testing that divides input domain into classes of data, and with the help of these classes of data, test cases can be derived. The idea is to partition the input domain of the system into several Equivalence Classes such that each members of the class would also result in the same error. The technique involves two steps: Identification of equivalence class -Partition any input domain into a minimum of two sets: valid values and invalid values and invalid test cases. To each valid and invalid test cases. To each valid and invalid test cases are covering all valid and invalid test cases. considering that no two invalid inputs mask each other. The whole numbers which is a perfect square-output will be a decimal number. Positive decimals Negative numbers (integer or decimal). Characters other than numbers like "a","!",";", etc.3. Boundary value analysisBoundary Value Analysis is based on testing the boundaries are an area where testing is likely to yield defects. Boundaries are very good places for errors to occur. Hence, if test cases are designed for boundary values of the input domain then the efficiency of testing improves and the probability of finding errors also increases. For example - If the valid range is 10 to 100 then test for 10,100 also apart from valid and invalid inputs. called causes with corresponding actions called the effect. The causes and effects are represented using Boolean graphs. The following steps are followed: Identify inputs (causes) and outputs (effect). Develop a cause-effect graph. Transform the graph into a decision table. Convert decision table rules to test cases. For example, in the following causeeffect graph: It can be converted into a decision table like: Each column corresponds to a rule which will become a test case for testing. So there will be 4 test cases. 5. Requirement-based testing Requirement-Based Testing. This approach guarantees that the software aligns with documented specifications and meets the anticipated outcomes outlined during the initial phases of the project. Principles of Requirement-Based TestingTraceability: The approach aims to establish clear links between each test and its respective requirements, ensuring easy tracking. Early Engagement: Early involvement in testing allows teams to comprehend, validate, and clarify requirements, minimizing misinterpretation risks. Validation and Verification: This methodology focuses on both aspects to ensure software compliance with specified requirements, boosting testing reliability. 6. Compatibility testingCompatibility Testing is the test case results not only depends on the product but is also on the infrastructure for delivering functionality. When the infrastructure parameters that generally affect the compatibility of software are: Processor (Pentium 3, Pentium 4) and several processors. Architecture and characteristics of machine (32-bit or 64-bit). Back-end components such as database servers. Operating System (Windows, Linux, etc.) Black box testing focuses on verifying the functionality of a software application by evaluating its inputs and outputs without any knowledge of its internal workings. The tester interacts with the system as an end-user to ensure the software meets its requirements and performs tasks as expected. OA WolfMobotSelendroidWatirKatalonIBM Rational Functional Tester (RFT)AutoHotkeyRanorexSelenium IDETestCompleteFeatures of Black Box TestingIndependent testing: Black box testing is performed by testers who are not involved in the development of the application, which helps to ensure that testing is unbiased and impartial. Testing from a user's perspective: Black box testing is conducted from the perspective of an end user, which helps to ensure that the application meets user requirements and is easy to use. No knowledge of internal code: Testers performing black box testing do not have access to the application's internal code, which allows them to focus on testing the application's external behavior and functionality. Requirements, which helps to ensure that the application meets the required specifications. Different testing techniques: Black box testing can be performed using various testing techniques, such as functional testing, acceptance testing, and regression testing Easy to automate using various automation tools, which helps to reduce the overall testing time and effort. Scalability: Black box testing can be scaled up or down depending on the size and complexity of the application being tested, which helps to ensure that testing is more representative of how the end users will interact with the application. What Does black Box Testing Focus On?Black Box Testing focuses on assessing how an application performs based on its inputs and outputs, without the need to look internally in the code or internal structure. Here are the key areas Black Box Testing focuses on: Functional Requirements: verify the software performs the tasks it was designed for, according to the requirements, and meets user expectations. User Interface (UI): Tests the usability of the application by checking the interface is easy to navigate and works properly for users. Input Validation: Verifies that the system correctly handles both valid and invalid inputs, providing the right output and error messages when needed while running. Security: Checks that sensitive data is protected through encryption, proper access controls are in place, and there are no security vulnerabilities that could be exploited. Performance and Reliability: Assesses how the application behaves under stress, such as high user load or long periods of use, ensuring it remains stable and reliable. Compatibility: Verifies the software works across various platforms, including different operating systems, browsers, devices, and network configurations. Error Handling: Tests how the system handles unexpected issues or failures. The application and any external services (like APIs) are working well together, confirming the system functions as intended when components interact. ConclusionHere we learned the actual process of Black Box Testing, and we discussed the all topics related to the same in detail. If you want to learn more about the Software Testing types and all related to Software testing. Refer Software Testing Tutorial" which is really helpful for learning purposes. Functional Testing is a type of Software Testing in which the application properly satisfies the requirements or specifications. This type of testing is particularly concerned with the result of processing. It focuses on the simulation of actual system usage but does not develop any system structure assumptions. The article focuses on discussing function testing? Functional testing is defined as a type of testing that verifies that each function of the software application works in conformance with the requirement and specification. This testing is not concerned with the source code of the application. Each functionality of the software application is tested by providing appropriate test input, expecting the user interface APIs, database, security, client or server application, and functional testing is crucial for ensuring that a software application meets its specified requirements, having a solid understanding of how to effectively carry out this type of testing is essential for any software tester. If you're looking to deepen your expertise in functional testing and other key areas of software testing, helping you expertise in functional testing and other key areas of software testing. ensure that every functional testing mainly involves black box testing and can be done manually or using automation. The purpose of functional testing Flow Test each functional testing restriction testing automation testing functional testin tests each function of the application by providing the appropriate input and verifying the output against the functional testing, the tester tests each entry function of the application. Test flow of the GUI screen: In functional testing, the tester tests each entry function of the application. flow of the GUI screen is checked so that the user can navigate throughout the application. What to Test in Functional Testing? The goal of functional testing is to make sure the app without any trouble. It's all about making sure the experience is smooth. Main Functions: Functional testing also looks at the app is a core features to verify they are working correctly, just as they're meant to. Accessibility: This ensures the app is a core features to verify they are working correctly. Handling: Lastly, it tests how the app handles errors. Are the right error messages shown when something goes wrong? This part verify users are informed when an issue arises. Functional testing involves the following steps: Step 1. Identify test input: This step involves the following steps: Step 1. Identify test input: This step involves the following steps: Step 1. Identify test input: This step involves identifying the functionality that needs to be testec This can vary from testing the usability functions, and main functions to error conditions. Functional Testing process Step 2. Compute expected outcomes: Create input data based on the specifications of the function and determine the output based on these specifications. Step 3. Execute test cases: This step involves executing the designed test cases and recording the output. Step 4. Compare the actual and expected output: In this step, the actual output obtained after executing the test cases is compared with the expected output to determine the amount of deviation in the results. This step reveals if the system is working as expected or not. Type of Functional Testing Techniques There are various types of functional Testing which are as follows: Functional Testing Techniques Unit testing: It is the type of functional testing techniques units are tested as a group and expose the individual units are tested. It ensures that each module is working correctly. Integration testing: faults in the interaction between the integrated units. Smoke testing: It is a type of functional testing technique where the basic functionality or feature of the application is tested as it ensures that the most important function works properly. as intended. It is the final phase of testing before the product release. Interface testing: is a type of software testing that is performed on the complete integrated system to evaluate the compliance of the system with the corresponding requirements. Regression testing: It is done to make sure that the code changes introduced are working as uncertained are working as uncertain expected. White box testing: It is a type of software testing that allows the tester to verify the internal workings of the software system. This includes analyzing the code, infrastructure, and integrations with the external system. the internal workings or structures of the software testing: It is a type of software testing that checks the schema, tables, etc of the database under testing. It also known as monkey testing or random testing. Recovery testing: It is a type of software testing that verifies the software 's ability to recover from failures, crashes, etc. Static testing: It is a type of software failures, crashes, etc. Static testing is a type of software failures, crashes, etc. Static testing is a type of software testing that is performed to check the defects in software failures, crashes, etc. Static testing is a type of software testing that is performed to check the defects in software failures, crashes, etc. Static testing is a type of software failures, crashes, etc. Static testing is a type of software failures in software testing that is performed to check the defects in software failures, crashes, etc. Static testing is a type of software failures, crashes, etc. Static testing is a type of software failures, crashes, etc. Static testing is a type of software failures, crashes, etc. Static testing is a type of software failures, crashes, etc. Static testing is a type of software failures, crashes, etc. Static testing is a type of software failures, crashes, etc. Static testing is a type of software failures, crashes, etc. Static testing is a type of software failures, crashes, etc. Static testing is a type of software failures, crashes, etc. Static testing is a type of software failures, etc. Static testing is a type of software failures, etc. Static testing is a type of software failures, etc. Static testing is a type of software failures, etc. Static testing is a type of software failures, etc. Static testing is a type of software failures, etc. Static testing is a type of software failures, etc. Static testing is a type of software failures, etc. Static testing is a type of software failures, etc. Static testing is a type of software failures, etc. Static testing is a type of software failures, etc. Static testing is a type of software failures, etc. Static testing is a type of software failures, etc. Static testing is a type of software failures, etc. Static testing is a type of software failures, etc. Static testing is a type of s software testing that includes black-box and white-box testing. In this, the test objects can be tested independently as a component without integrating with other components. Functional Testing vs Non-Functional Testing Below are the differences between functional testing and non-functional testing Non based on the expectations of the customer. Objective is to validate software actions. The objective is to validate software system Requirements Functional testing is carried out using the performance of the software system Requirements. product does. It describes how the product works. Example Unit testing. Integration testing. Sanity testing. Read More: Differences between Functional and Non-functional Testing Below are the tools for functional testing: 1. Selenium: It is an open-source umbrella project for a range of tools and libraries developed with the aim to support browsers. It provides a single interface that lets the tester write test across mos modern web browsers. 2. QTP: This tool now can UFT is a tool designed to perform automated functional testing without the need to monitor the system in intervals. It can be used to test web, desktop applications, and client servers. It is based on the VB scripting language. It is one of the widely used automation tools in the testing industry. 3. JUnit: It is a unit-testing open-source framework for the Java programming language. It is used by Java developers to write and execute automate tests for web applications. It provides several annotations to identify test methods. It has test runners to run tests. 4. SoapUI: It is one of the leading tools for SOAP and web service testing. It allows for easy and rapid creation and execution of functional, regression, and load tests. It has an easy-to-use graphical interface. It provides a code-free test environment where one can create and execute complex test cases with drag-and-drop options. It lets to dynamically analyze how well SOAP and REST service contract is covered by the functional tests. 5. Cucumber: It is an open-source testing tool written in Ruby language. This tool allows for easy reuse of code in tests due to the style of writing the tests. Best Practices for Functional Testing Automate Functional tests can be repetitive, time-consuming processes so the more the tests are automated the faster one can identify and correct defects, and the more savings can be achieved in time and costs. It may not be possible to automate all test cases, so automating important test cases can improve the test ROI. Dedicated automation team: Automation requires time, effort, and a special skill set. It is considered best to allocate automation tasks to those who are equipped to accomplish them. Create test cases later in the project developmen cycle. Pick the right tests: It is very important to pick the right tests that are prone to human error. Prioritize: Testers have finite time and budget, so it is not possible to test each and every feature in the application. Consider high-priority functions first to create test cases. Test frequently: Prepare a basic test automation bucket and create a strategy for frequently: Prepare a basic test automation bucket and create a strategy for frequently. product. Customer satisfaction: It ensures that all requirements are met and ensures that the customer usage. Proper working of application: This ensures that the application works as expected and ensures proper working of all the functionality of the application. Improves quality of the product: Functional testing ensures the security and safety of the product and improves the quality of the product. Limitations of Functional Testing Missed critical errors: There are chances while executing functional tests that critical and logical errors are missed. Redundant testing: There are high chances of performing redundant testing. Incomplete requirements: If the requirement is not complete then performing this testing will check that each function of a software application has been working as expected and focuses on the user interface, APIs, and functionality. While it improves product quality and customer satisfaction, it may miss the major errors and involve repeated testing. overall the important to deliver a bug-free and high-quality product as output. In this post 'Types of Software Testing', I would like to mention almost all the software testing types in one place. One challenge to learning about software testing is that there are many terms in the industry, and these terms often used inconsistently. While there is no universally-accepted definitions for all the testing terms, I think a good source is to refer ISTQB Certified Tester Foundation Level Syllabus. The Ultimate List of 100+ Software Testing TypesI would like to start with Software Testing before going to the actual post 100+ Software Test Types. Software Testing: It is a process, to evaluate the functionality of a software application with an intent to find whether the developed software application with an intent to find whether the developed software application with an intent to find whether the developed software application with an intent to find whether the developed software application with an intent to find whether the developed software application with an intent to find whether the developed software application with an intent to find whether the developed software application with an intent to find whether the developed software application with an intent to find whether the developed software application with an intent to find whether the developed software application with an intent to find whether the developed software application with an intent to find whether the developed software application with an intent to find whether the developed software application with an intent to find whether the developed software application with an intent to find whether the developed software application with an intent to find whether the developed software application with an intent to find whether the developed software application with an intent to find whether the developed software application with an intent to find whether the developed software application with an intent to find whether the developed software application with an intent to find whether the developed software application with an intent to find whether the developed software application with an intent to find whether the developed software application with an intent to find whether the developed software application with a softw Software Testing Definitions & Approaches. The Ultimate List of Types of Testing: In simple words, what the system actually does is functional testing. To verify that each function of the software application behaves as specified in the requirement document. Testing all the functionalities by providing appropriate input to verify whether the actual output is matching the expected output or not. It falls within the scope of black-box testing and the testers need not concern about the source code of the application. 2. Non-functionality testing. Non-functional testing are concernated output or not. It falls within the scope of black-box testing and the testers need not concernated output is matching the expected output or not. It falls within the scope of black-box testing and the testers need not concernated output is matching the expected output or not. It falls within the scope of black-box testing and the testers need not concernated output or not. It falls within the scope of black-box testing and the testers need not concernated output or not. It falls within the scope of black-box testing and the testers need not concernated output or not. It falls within the scope of black-box testing and the testers need not concernated output or not. functional testing refers to various aspects of the software such as performance, load, stress, scalability, etc., Main focus is to improve the user experience on how fast the system responds to a request.3. Manual testing: Manual testing is the process of testing the software manually to find the defects. A tester should have the perspective of an end-user and to ensure all the features are working as mentioned in the requirement document. In this process, testers execute the test cases and generate the reports manually without using any automation tools.4. Automated testing: Automated testing is the process of testing the software using an automation tool to find the defects. In this process, executing the test scripts and generating the results are performed automatically by automation tools. Some most popular tools to do automation testing are HP QTP/UFT, Selenium WebDriver, etc., Learn the Difference between Manual & Automated Testing here...5. Black box testing: Black Box Testing is a software testing method in which testers evaluate the functionality of the software under test without looking at the internal code structure. This can be applied to every level of software testing is a methodology of testing. that scans the program structure and then creates the test data based on the flow and logic of the program. This type of testing is mainly done by utilizing the internal programming logic to select the proper test data and to remove design errors. The glass box testing is called open box testing, logic-driven testing, path driven testing, or clear box testing. The techniques of glass box testing are Path Coverage, Branch Coverage, and Statement Coverage.7. White box testing: White Box Testing is also called as Glass Box, Clear Box, and Structural Testing. It is based on applications internal code structure. In white-box testing, an internal perspective of the system, as well as programming skills are used to design test cases. This testing usually was done at the unit level. Click here for more details.8. Specification-based testing: Specification-based testing is similar to behavior-driven testing of the internal logic or the flow of the program. Thus in short the testers are more concerned with the behavior of the application. Specification based testing includes both nonfunctional testing. A specification may be in the form of a prototype, a written document, a group of use cases. It is considered a reference for building test data. The techniques for specification-based testing are Decision Table, Equivalence Partitioning, Boundary Value Analysis, and State Transitioning.9. Structure-based testing requires the technical know-how of the program logic and flow. This type of testing ensures that there is maximum test coverage and test design is proper. The structure-based testing-the structure-based testing ensures that there is maximum test coverage and test design is proper. The structure-based testing-the structure-based testing ensures that there is maximum test coverage and test design is proper. The structure-based testing ensures that there is maximum test coverage and test design is proper. based testing is adopted in the initial phase of the project development to determine the amount of testing that is needed. The structure-based testing also assists in creating some additional test cases to increase the test coverage, different from preexisting test cases. So it achieves more depth in testing. The techniques for structure-based testing are Path testing, Condition Testing, Multiple Condition Testing, Decision Testing, and Statement Testing. 10. Gray box testing: Grey box is the combination of both White Box and Black Box Testing needs to have access to design documents. This helps to create better test cases in this process. 11. Unit testing: Unit Testing is also called Module Testing or Component Testing. It is done to check whether the individual unit or module of the source code is working properly. It is done by the developer's environment.12. Component testing: Refer Unit Testing13. Module testing: Refer Unit Testing14. Integration testing: Integration Testing is the process of testing the interface between the two software units. Integration testing is done by multiple approach, and Hybrid Integration Testing Complete Guide15. System testing: Testing the fully integrated application to evaluate the system's compliance with its specified requirements is called System Testing AKA End to End testing. This is done by the end-users along with the testers to validate the functionality of the application. After successful acceptance testing. Formal testing conducted to determine whether an application is developed as per the requirement. It allows the customer to accept or reject the application. Types of acceptance testing are Alpha, Beta & Gamma.17. Big bang Integration Testing: Combining all the modules once and verifying the functionality after completion of individual module testing. Top-down and bottom-up are carried out by using dummy modules known as Stubs and Drivers. These Stubs and Drivers are used to stand-in for missing components to simulate data communication between modules. 18. Top-down Integration Testing: Testing takes place from top to bottom. High-level modules known as Stubs and Drivers. are tested first and then low-level modules and finally integration testing. 19. Bottom-up Integration Testing: It is a reciprocate of the Top-Down Approach. Testing takes place from bottom to up. Lowest level modules are tested first and then high-level modules are tested first and then high-level modules to a low level to ensure the system is working as intended. Drivers are used as a temporary module for integration testing. 20. Hybrid Integration testing is the combination of both Top-down and bottom-up integration testing: Alpha testing is done by the client or outsourcing team with the presence of developers or testers. 22. Beta testing is done by a limited number of end-users before delivery. Usually, it is done in the client's place.23. Gamma Testing: Gamma testing is done when the software is ready for release with specified requirements. It is done at the client's place. It is done at the partitioning, inputs to the software or system are divided into groups that are expected to exhibit similar behavior, so they are likely to be proposed in the same way. Hence selecting one input from each group to design the test cases. Read more on Equivalence Partitioning Testing Technique...25. Boundary value analysis testing: Boundary value analysis (BVA) is based on testing the boundary values of valid and invalid partitions. The Behavior at the edge of each equivalence partition has its maximum and minimum values and these maximum and the maximum and minimum values are the boundary value for a valid partition is a valid boundary value. Similarly, a boundary value. Similarly, a boundary value. Similarly, a boundary value for a valid partition is a valid boundary value. Similarly, a boundary value for a valid boundary value for a valid boundary value. Similarly, a boundary value for a valid boundary value for a valid boundary value for a valid boundary value. Similarly, a boundary value for a valid boundary value. Similarly, a boundary value for a valid boundary value for a valid boundary value for a valid boundary value. Similarly, a boundary value for a valid boundary value fo technique is appropriate for functionalities which has logical relationships between inputs (if-else logic). In the Decision table, we consider conditions and actions. We take conditions as inputs and actions as outputs. Read more on the Decision Table Testing Technique...27. Cause-effect graph testing: The cause-effect graph testing is a test case development methodology that begins with a collection of requirements and then identifies the optimal number of test cases needed to achieve maximum coverage with minimum time and cost.28. State transition testing; Using state transition testing, we pick test cases from an application where we need to test different system transitions. We can apply this when an application gives a different output for the same input, depending on what has happened in the earlier state. Read more on State Transition Test Design Technique...29. Exhaustive Testing all the functionalities using all valid and invalid inputs and preconditions is known as Exhaustive testing. 30. Early Testing: Use case testing is carried out with the help of a use case document. It is done to identify test scenarios to test end to end testing32. Scenario testing: Scenario testing is a software testing technique that is based on a scenario. It involves converting business requirements to test scenario should be motivating, credible, complex, and the outcome of which is easy to evaluate.33. Documentation testing: Documentation testing is done to validate the documented artifacts such as requirements, test plan, traceability matrix, test cases.34. Statement coverage testing is a white box testing technique which is to validate whether each and every statement in the code is executed at least once.35. Decision coverage testing/branch coverage testing: Decision coverage testing is a white box testing technique which is to validate every possible branch is executed at least once.36. Path testing: Path coverage testing is a white box testing technique which is to validate every possible branch is executed at least once.37. Mutation testing: Mutation testing is a type of white box testing which is to change (mutate) certain statements in the source code and verify if the tests are able to find the errors.38. Loop testing: Loop testing is a white box testing technique which is to validate a different kind of loops such as simple loops, nested loops, and unstructured loops.39. Performance testing: This type of testing determines or validates the speed, scalability, and/or stability, and/or stability, and/or stability characteristics of the system or application under test. Performance is concerned with achieving response times, throughput, and resource-utilization levels that meet the performance is concerned with achieving response times. Load testing: It is to verify that the system/application can handle the expected number of transactions.41. Stress testing: Running a system at to verify the behavior of the system/application can handle the expectations.42. Soak testing: Running a system at the system. high load for a prolonged period of time to identify the performance problems is called Soak Testing. An application is given a considerable load for a considerable load for a considerable duration to check its behavior and performance under such a condition.44. Stability testing: Stability testing is a testing methodology used to check the capacity of the application to do the required actions under a specific state or stress. It is a type of non-functional testing and is used to verify if the application can perform uninterruptedly for a specific duration of time with a significant number of users and stress. It also checks memory leaks or other issues that degrade the stability testing is a type of non-functional testing. It is to determine how the application under test scales with the increasing workload.46. Volume testing: It is to verify that the system/application can handle a large amount of data47. Robustness testing is a type of testing is the process of identifying the vulnerabilities or weaknesses in the application.49. Adhoc testing: Ad-hoc testing is quite opposite to the formal testing. It is an informal testing type. In Adhoc testing is primarily performed if the knowledge of testers in the application under test is very high. Testers randomly test the application without following any documents and testing type. In Adhoc testing is primarily performed if the knowledge of testers in the application under test is very high. test the application without any test cases or any business requirement. 50. Exploratory testing: To ensure that the defects which were found and posted in the earlier build were fixed or not in the current build. Say, Build 1.0 was released. The test team found some defects (Defect Id 1.0.1, 1.0.2) and posted. Build 1.1 was released, now testing the defects 1.0.1 and 1.0.2 in this build is retesting. S2. Regression testing: Repeated testing of an already tested program, after modification and 1.0.2 in this build is retesting. S2. Regression testing of an already tested program, after modification and 1.0.2 in this build is retesting. S2. Regression testing of an already tested program, after modification and 1.0.2 in this build is retesting. S2. Regression testing of an already tested program, after modification and 1.0.2 in this build is retesting. S2. Regression testing the defects (Defect Id 1.0.1, 1.0.2) and posted. Build 1.0 was released, now testing the defects (Defect Id 1.0.1, 1.0.2) and posted. Build 1.0 was released. The test team found some defects (Defect Id 1.0.1, 1.0.2) and posted. Build 1.0 was released. Build 1.0 wa to discover any defects introduced or uncovered as a result of the changes in the software being tested or in another related or unrelated software components.53. Smoke testing: Smoke testing: Smoke testing is done at the "build level". It helps not to waste the testing time to simply testing is done during the release phase to check for the main functionalities of the application without going deeper. It is also called as a subset of Regression testing. It is done at the "release level". At times due to release time constraints rigorous regression testing can't be done to the build, sanity testing does that part by checking main functionalities.55. Dynamic testing involves in reviewing the documents to identify the defects in the early stages of SDLC.57. Monkey testing: Perform abnormal action on the application.58. Gorilla testing: Gorilla testing: Gorilla testing: Berlow test robustness of the system.59. Usability testing: To verify whether the application is user-friendly or not and was comfortably used by an end-user or not. The main focus of this testing is to check whether the end-user or not. The main focus of this testing is to check whether the end-user or not. operate it.60. Accessibility testing: Accessibility testing: It aims to discover how easily people with disabilities (such as visual Impairment, Cognitive is working as expected in a different combination of environmental components.62. Configuration testing: Localization is a process of testing an application can work without any issues.63. Localization is a process of adapting globalization software for a specific region or language by adding local specific components.64. Globalization testing: Globa It is to determine what system supposed to do. It helps to check whether the application is justifying the requirements or not.67. Negative testing: Security testing is a process to determine whether the system protects data and maintains functionality as intended. Security Testing Complete Guide69. Penetration testing is also known as pen testing. It is a type of security testing is also known as pen testing. It is matched with the data stored in the database. It involves in checking the schema, tables, triggers etc., of the database.71. Bucket testing: Bucket testing: Refer Bucket testing: Refer Bucket testing...73. Split testing-Refer bucket testing...74. Reliability Testing: Perform testing on the application continuously for a long period of time in order to verify the stability of the application 75. Interface testing: Concurrency testing: Concurrency testing means accessing the application at the same time by multiple users to ensure the stability of the system. This is mainly used to identify if anything a massive amount of random data to the system in an attempt to make it crash to identify if anything breaks in the application.78. GUI Testing: Graphical User Interface Testing is to test the interface between the application and the end user. Mainly testers concern about the applearance of the elements such as fonts and colors conforms to design specifications.79. API testing: API testing is to test the interface. API testing is a type of software testing that involves testing APIs using some tools like SOAPUI, PostMan.80. Agile testing: Agile testing is a type of testing is conducted throughout the lifecycle of the continuously evolving project instead of being confined to a particular phase.81. End to end testing: The end to end testing is a testing methodology to check if the flow of the software from beginning till the end is as per the expected result. It helps to determine the dependencies in the system and ensures there is no data loss or corruption while the interaction between multiple components. While doing end to end testing, the key features like interaction among the database, other systems, network, and so on are tested and verified if they are happening as per expected results.82. Recovery testing: Recovery testing: Recovery testing is performed in order to determine how quickly the system can recover after the system can be used. non-functional testing.83. Risk-based testing: Identify the modules or functionalities which are most likely cause failures and then testing those functionalities.84. Installation.85. Formal Testing: It is a process where the testers test the application by having pre-planned procedures and proper documentation.86. Pilot testing: Eackend testing is a testing technique for the database and server-side validation. It is often known as database testing. It is done to test if the entered data in the front end is stored and reflected in the database. The backend testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testing is used to prevent data truncation and loss. The database testi structural [which deals with the testing of the database structure like tables, views, and so on]. Thus the backend testing cross Browser testing: Cross Browser testing is a type of non-functional test which helps us to ensure that our website or web application works as a browser compatibility testing are the font look and feel in browsers, header and footer, styles of the page, formats in date, image positioning, HTML and CSS validation, zoom in and zoom out and alignment of the elements on the page and so on.90. Forward compatibility testing is to validate the application under test is working as intended in the later versions of the software's current version.91. Backward compatibility testing: Backward compatibility testing: ...93. Compliance testing: Compliance testing is non-functional testing which is done to validate whether the software meets a defined set of standards.94. Conformance testing: Conformance testing is a testing technique to check that a product meets a defined set of standards.94. the software is compliant. The conformance testing features include the below points: It is carried out by external organizations that are certified in their domain. It checks the robust testing processes. 95. UI testing; In UI testing; testers aim to test both GUI and Command Line Interfaces (CLIs)Also, refer to GUI Testing...96. Destructive testing is a testing technique that aims to validate the robustness of the application by testing is a testing technique that examines the requirements of an application for pre-conditions, initial states, and configuration for the proper functioning of the application.98. Crowdsourced testing: ETL (Extract, Transform, and Load) testing involves invalidating the data movement from source to destination and verifying the data count in both source and destination and verifying data extraction, transformation, and also verifying the table relations.100. Data warehouse testing: Fault injection testing: Fault injection testing: Fault injection testing: Fault injection testing is a testing technique in which fault is intentionally introduced in the code in order to improve the test coverage.102. Failover testing: Failover testing is a testing technique that validates a system's ability to be able to allocate extra resource during the server failure and transferring of the processing part to back-up systems103. All pair testing: All pair testing approach is to test the application with all possible combination of the values of input parameters.104. Pairwise Testing: The pairwise testing is a testing method to test an application with permutation and combinations of specifications. By using the normal exhaustive testing approach, it may become impossible to test the complete product. But by following the permutation and combination of inputs, the testing of the product is achievable. For example, let us consider a product to be tested with 15 inputs and there are 15 possible configurations for each input. So there are a total 15^15 inputs to be tested. In such a scenario, complete testing is not possible and we have to choose combinations of inputs. Here I am going to conclude different types of software testing types. If you like this post, please share it with your friends. Here I have hand-picked a few posts which will help you to learn more interview related stuff: If you have any more questions, feel free to ask via comments. If you find this post, please share it with your friends on Social Networking. Software Testing is an important part of the Software Development Life Cycle, which includes many more Types of Software Testing is an important process that is used for the enhancement of the Software Quality and Reliability of the application. It is important to understand the key principles will be helpful for the tester to identify the Software issue earlier and verify the Software meets to the expectations. Testing Shows the Presence of DefectsExhaustive TestingDefect ClusteringPesticide ParadoxTesting principle. Different Types of Software TestingHere are the Types of Software Testing mainly categorized into the two domain, which are below. Types of Software Testing 1. Manual Testing will be check the defect manually with trying one by one function is working as expected. Advantages of Manual Testing: Fast and accurate visual feedback: It detects almost every bug in the software application and is used to test the dynamically changing GUI designs like layout, text, etc.Less expensive: It is less expensive as it does not required while using the black box testing method. It is easy to learn for the new testers. Efficient for unplanned changes: Manual testing is suitable in case of unplanned changes to the application, as it can be adopted easily. Automation Testing is a technique where the Tester writes scripts independently and uses suitable Software or Automation Tools to test the software. It is an Automation Process of a Manual Process. It allows for executing repetitive tasks without the use of a Manual Tester. Advantages of Automation testing can be left virtually unattended and thus it allows monitoring of the results at the end of the process. Thus, simplifying the overall test execution and increasing the efficiency of the application. Improves Reliability of Tests: Automation testing, more test cases can be created and executed for the application under test. Thus, resulting in higher test coverage and the detection of more bugs. This allows for the testing of more complex applications and more features can be tested. Minimizing Human Interaction: In automated from test case creation to execution thus there are no changes for human error due to neglect. This reduces the necessity for fixing glitches in the post-release phase. Manual Vs. Automated testing. The test cases are executed by the software testing. The test cases are executed by the human tester. In automated testing. The test cases are executed by the software tools.Processing TimeManual testing is time-consuming.Automation testing is faster than manual testing.Exploratory testing is not possible in manual testing. automation testing, Framework requirementManual testing doesn't use frameworks. Automation testing uses frameworks like Data Drive, Keyword, etc. Types of Manual Testing 1. White Box Testing uses frameworks like Data Drive, Keyword, etc. Types of Manual Testing technique that involves testing technique testing testing testing technique testing technique testing techniqu to the source code and uses this knowledge to design test cases that can verify the correctness of the software at the code level. Advantages of White box testing: Thorough as the entire code and structures are tested. Code Optimization: It results in the optimization of code removing errors and helps in removing extra lines of code. Early Detection of Defects: It can start at an earlier stage as it doesn't require any interface as in the case of black box testing. Integration with SDLC: White box testing can be easily started in the Software Development Life Cycle. Detection of Complex Defects: Testers can identify defects that cannot be detected through other testing techniques. 2. Black Box Testing Black-box testing is a type of software testing in which the tester is not concerned with the internal knowledge or implementation details of the software but rather focuses on validating the functionality based on the provided specifications or requirements. Advantages of Black Box Testing: The tester does not need to have more functional knowledge or programming skills to implement the Black Box Testing. It is efficient for implementing the tests in the larger system. Tests are executed from the user's or client's point of view. specifications. 3. Gray Box Testing Gray Box Testing technique and the Black Box Testing technique and tester. The internal structure is partially known in Gray Box Testing. This includes access to internal data structures and algorithms to design the test cases. Advantages of Gray Box Testing: Clarity of goals: Users and developers have clear goals while doing testing. This includes access to internal data structures and algorithms to design the test cases. perspective. High programming skills not required: Testers are not required to have high programming skills for this testing. Non-intrusive: Gray box testing is non-intrusive: Gray box testing is non-intrusive: Gray box testing is non-intrusive. the system is tested against the functional requirements and specifications. Functional testing ensures that the requirements or specifications. Functional testing ensures that the requirements are properly satisfied by the application. structure assumptions. The article focuses on discussing function testing. Advantages of Functional Testing: Bug-free product: Functional testing ensures that all requirements are met and ensures that the customer is satisfied. Testing focused on specifications

Functional testing is focused on specifications as per customer usage. Proper working of application: This ensures the quality of the product: Functional testing ensures the quality of the product and improves the quality of the product and improves the quality of the product. the product. 2. Non-Functional Testing Non-Functional Testing is a type of Software Testing is a software Testing is a software to verify the non-functional testing is a software testing is a software testing that is performed to verify the non-functional testing is a software testing that is performed to verify the non-functional testing is a software testing testin testing technique that checks the non-functional attributes of the system. Non-functional testing is defined as a type of software testing to check non-functional aspects of a software application. It is designed to test the readiness of a system as per nonfunctional parameters which are never addressed by functional testing. Non-functional testing is as important as functional testing. Advantages of Non-functional Testing: Improved performance of the system and determines the performance. Less time-consuming: Non-functional testing process. Improves user experience: Non-functional testing like Usability testing checks how easily usable and user-friendly the software is for the users. Thus, focus on improving the overall user experience for the application. More secure product: As non-functional testing specifically includes security testing that checks the security bottlenecks of the application and how secure is the application against attacks from internal and external sources. Types of Functional Testing 1. Unit tests are working as intended. Unit tests are usually automated and are designed to test specific parts of the code, such as a particular function or method. Unit testing is done at the lowest level of the software development process, where individual units of code are tested in isolation. Note: Unit Testing basically Included in both White Box Testing and Black Box Testing. Advantages of Unit Testing: Some of the advantages of Unit Testing are listed below. It helps to identify bugs early in the development process before they become more difficult and easier to understand and maintain. It helps to improve the overall quality and reliability of the software. Note: Some popular frameworks and tools that are used for unit testing is only one aspect of software testing and it should be used in combination with other types of testing such as integration testing, functional testing, and acceptance testing to ensure that the software meets the needs of its users. It focuses on the smallest unit of software design. In this, we test an individual unit or group of interrelated units. It is often done by the programmer by using sample input and observing its corresponding outputs. or function is working fine. Misunderstood or incorrect, arithmetic precedence. Incorrect initialization. 2. Integration testing how different units of the software application interact with each other. It is used to identify and resolve any issues that may arise when different units of the software area. combined. Integration testing is typically done after unit testing and before functional testing. Different ways of Integration Testing are discussed below. Top-down integration testing: It starts with the highest-level modules and differentiates them from lower-level modules. Big-Bang integration testing: It starts with the lowest-level modules and integrates them all at once. Incremental integrates the modules in small groups, testing each group as it is added. Advantages of Integrating Testing: It helps to identify and resolve issues that may arise when different units of the software work together as intended. It helps to improve the overall reliability and stability and stability of the software. It's important to keep in mind that Integration testing is essential for complex systems where different components are integrated. As with unit testing, functional testing, functional testing, and acceptance testing to ensure that the software meets the needs of its users. The objective is to take unit-tested components and build a program structure that has been dictated by design. Integration testing is of four types: (i) Bottom-up (iii) Sandwich (iv) Big-Bang Example: Black Box testing: It is used for validation. In this, we ignore internal working mechanisms and focus on "what is the output?" White box testing: It is used for verification. In this, we focus on internal mechanisms i.e. how the output?" White box testing is a type of software testing that evaluates the overall functionality and performance of a complete and fully integrated software solution. It tests if the system meets the specified requirements and if it is suitable for delivery to the end-users. This type of software testing is a type of software testing and before the acceptance testing. compliance of the system with the corresponding requirements. In system testing, integration testing passed components are taken as input. The goal of integrated. Advantages of System Testing: The testers do not require more knowledge of programming to carry out this testing. It will test the entire product or software so that we will easily detect the errors or defects that cannot be identified during the unit testing and integration testing. The testing environment is similar to that of the real-time production or business environment. It checks the entire functionality of the system with different test scripts and also it covers the technical and business requirements of clients. After this testing, the product will almost cover all the possible bugs or errors and hence the development team will confidently go ahead with acceptance testing. 4. End-to-end testing is the type of software testing used to test entire software from starting to the end along with its integration with external interfaces. The main purpose of end-to-end testing is to identify system dependencies and to make sure that the data integrity and communication with other systems, interfaces and databases to exercise complete production.5. Acceptance TestingAcceptance Testing is formal testing according to user needs, requirements, and business processes conducted to determine whether a system satisfies the acceptance criteria or not and to enable the users, customers, or other authorized entities to determine whether to accept the system or not. Advantages of Acceptance Criteria or not and to enable the users, customers, or other authorized entities to determine whether to accept the system or not. users directly as it involves the users for testing. Automated test execution. It brings confidence and satisfaction to the clients as they are directly involved in the testing process. It is easier for the user to describe their requirement. It covers only the Black-Box testing process and hence the entire functionality of the product will be tested. Types of Integration Testing Here are the Types of Integration testing is also a phase of SDLC (Software Development tist are performed at different stages of the development cycle. Incremental Testing is one of the testing approaches that is commonly used in the software field during the testing phase of integration testing which is performed after unit testing. Several stubs and drivers are used to test the modules of Incremental Testing: Each module has its specific significance. Each one gets a role to play during the testing as they are incremented individually. Defects are detected in smaller modules rather than denoting errors and then editing and re-correcting large files. It's more flexible and cost-efficient as per requirements and scopes. The customer gets the chance to respond to each building. There are 2 Types of Incremental Testing 1. Top-down Integration Testing Top-down Integration Testing is a type of incremental integration testing approach in which testing is done by integration testing approach in which testing is done by integration testing approach in which testing is done by integration testing is a type of incremental integration testing is done by integrating or joining two or more modules are tested first, and then low-level modules are tested. Then, finally, integration is done to ensure that the system is working properly. Stubs and drivers are used to carry out this project. This technique is used to increase or stimulate the behavior of Modules that are not integrated into a lower level. Advantages Top Down Integrated into a lower level. identified at an early stage and fault localization is also easier. Low-level utilities that are not tested well and high-level testers are tested well and high-level testers are tested well in an appropriate manner. Representation of test cases is easier and simpler once Input-Output functions are added. approach in which testing is done by integrating or joining two or more modules are tested first, and then high-level modules are tested first, and then high-level modules are tested. This type of testing or approach is also known as inductive reasoning and is used as a synthesis synonym in many cases. Bottom-up testing is user-friendly testing and results in an increase in overall software development. This testing results. It is easy and simple to create and develop test conditions. It is also easy to observe test results. It is not necessary to know about the details of the structural design. Low-level utilities are also tested well and are also compatible with the object-oriented structure. Types of Non-Functional Testing Here are the Types of Non-Functional Testing 1. Performance Testing is a type of software testing that ensures software testing that ensures are also tested workload It is a testing technique carried out to determine system performance in terms of sensitivity, reactivity, and stability under a particular workload. Advantages of Performance testing: Performance testing ensures the issues if anything occurs. It ensures the great optimization of the software and also allows many users to use it at the same time. It ensures the client as well as the end-customer's satisfaction. Performance testing has several advantages that make it an important aspect of software testing. system such as slow database gueries, insufficient memory, or network congestion. This helps developers optimize the system and ensure that it can handle the expected number of users or transactions. 2. Usability Testing is a type of testing, that is done from an end user's perspective to determine if the system is easily usable. Usability testing is generally the practice of testing how easy a design is to use on a group of representative users. In Usability testing: Usability testing is preferred to evaluate a product or service by testing it with the proper users. In Usability testing, the development and design teams will use to identify issues before coding and the result will be earlier issues will be solved. User-Centric Design: By involving actual users in the testing process, you ensure that your product or website is designed with their needs and preferences in mind. Identifying User Pain Points: Usability testing helps uncover areas where users struggle or encounter difficulties while interacting with your product. This insight allows you to address these pain points and improve the overall user experience. Optimizing User Interface: Through usability testing, you can evaluate the effectiveness of your user interface (UI) design, including layout, navigation, and interactive elements. This enables you to refine and optimize the UI for better usability. Enhancing User Satisfaction: By addressing usability issues and making improvements based on user feedback, you can enhance user satisfaction: By addressing usability resting is software testing that comes under the non functional testing category, and it is performed on an application to check its compatibility (running capability) on different platforms/environments. This testing is done only when the application functionality on various software, hardware platforms, networks 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. Advantages of Compatibility Testing: It ensures complete customer satisfaction. It provides service across multiple platforms. Identifying bugs during the development process. Types of Performance Testing Load Testing Load Testing Load Testing load conditions. The load testing is carried out for normal and extreme load conditions. Load testing is a type of performance testing is to identify bottlenecks and determine the maximum number of users or transactions the system can handle. It is an important aspect of software testing as it helps ensure that the system can handle the expected usage levels and identify any potential issues before the system is deployed to production. Advantages that make it an important aspect of software testing: Identifying bottlenecks: Load testing helps identify bottlenecks in the system such as slow database queries, insufficient memory, or network congestion. This helps developers optimize the system and ensure that it can handle the expected number of users or transactions. Improved scalability: By identifying the system and ensure that it can handle the expected number of users or transactions. of users or transactions over time. This is particularly important for web-based systems and applications that are expected to handle a high volume of traffic. Improved reliability: Load testing helps identify any potential issues that the system is reliable and stable when it is deployed to production. 2. Stress Testing is defined as types of software testing that verifies the system's robustness and error handling under the burden of some load conditions. It tests beyond the normal operating point and analyses how the system works under extreme conditions. Example: Test cases that may cause thrashing in a virtual operating system. Test cases that may cause excessive disk requirement Performance of software within the context of an integrated system. It is used to test the speed and effectiveness of the program. It is also called load testing. In it, we check, what is the performance of the system in the given load. Example: Checking several processor cycles. 3. Scalability Testing is a type of non-functional testing in which the performance of a software application, system, network or process is tested in terms of its capability to scale down the number of user request load or other such performance attributes. It can be carried out at a hardware, software or database level. Advantages of Scalability Testing: It provides more accessibility to the product. It detects issues with web page loading and other performance issues. It finds and fixes the issues earlier in the product which saves a lot of time. It ensures the end-user experience under the specific load. It provides customer satisfaction. It helps in effective tool utilization tracking. behavior of the software in different environmental parameters. It is defined as the ability testing is mainly intended to check whether the application will crash at any point in time or not. Advantages of Stability Testing: It gives the limit of the data that a system can handle practically. It provides confidence on the performance of the system. It determines the stability and robustness of the system under load. Stability testing leads to a better end-user experience. Other Types of Testing 1. Smoke Testing 1. Smoke Testing is done to make sure that the software under testing is ready or stable for further testing It is called a smoke test as the testing of an initial pass is done to check if it did not catch fire or smoke in the initial switch-on. Example: If the project has 2 modules so before going to the module make sure that module 1 works properly. Advantages of Smoke testing is easy to perform. It helps in identifying defects in the early stages. It improves the quality of the system. Smoke testing reduces the risk of failure. Smoke testing is a stoppage to check whether testing for the build can proceed or not. The focus of the team during the sanity testing process is to validate the functionality of the application and not detailed testing. Sanity Testing: Sanity Testing: Sanity Testing: Sanity Testing is generally performed on a build where the production deployment is required immediately like a critical bug fix. Advantages of Sanity Testing: testing helps to quickly identify defects in the core functionality. It can be carried out in less time as no documentation is required for sanity testing. If the defects are found during sanity testing. If the defects are found during sanity testing technique is not so expensive when compared to another type of testing. It helps to identify the dependent missing objects. 3. Regression testing features of a software application. It ensures that everything still works fine after updates or bug fixes. Regression testing can be performed in different ways, such as: Retesting - Checking the entire application or specific features that were affected by the changes. Re-execution - Running previously tested cases to make sure everything still functions properly. Comparison - Comparing the latest version of the software with an older version to ensure no features are broken. Advantages of Regression Testing: Prevents New Bugs - Ensures that software updates, bug fixes, or new features do not break existing functionality. Keeps Software Reliable - Confirms that the software continues to work as expected after any changes. Improves Stability - Regular regression testing helps maintain the overall stability and performance of the software. Every time a new module is added leads to changes in the program. This type of testing makes sure that the whole component works properly even after adding components to the complete program. Example: In school records, suppose we have module staff, students, and finance combining these modules and checking if the integration of these modules works fine in regression testing. 4. Acceptance Testing Acceptance Testing is done by the customers to check whether the delivered products perform the desired tasks or not, as stated in the requirements. We use Object-Oriented Testing for discussing test plans and for executing the projects. Advantages of Acceptance Testing: This testing helps the project team to know the further requirements of the users directly as it involves the users for testing process. It is easier for the user to describe their requirement. It covers only the Black-Box testing process and hence the entire functionality of the product will be tested. 5. User Acceptance Testing User Acceptance Testing (UAT) serves the purpose of ensuring that the software meets the business requirements and is ready for deployment by validating its functionality in a real-world environment. It allows end-users to test the software to ensure it meets their needs and operates as expected helping to identify and fix any issues before the final release. Advantages of User Acceptance testing: Ensures the software Meets User Needs: UAT helps make sure the software does what the end-users expect and addresses their business needs, which increases the incre out UAT, it can uncover real-world problems that developers might not have thought of. This ensures the software works in the way users need it to.Boosts User Satisfaction: By getting feedback from users early on, UAT helps shape the product into something that meets their expectations. The result is a happier and more satisfied user base. Reduces Post-Release Surprises: By catching issues before the software is released, UAT helps minimize the risk of bugs or broken features that could appear once it's live.Builds Confidence in the Product: When users test and approve the software, it builds confidence in the product. Testing is a type of software testing in which the tester is free to select any possible methodology to test the software. It is an unscripted approach to software developers use their learning, knowledge, skills, and abilities to test the software developers use their learning. Less preparation required: It takes no preparation as it is an unscripted testing technique. Finds critical defects: Exploratory testing, testers use their knowledge, skills, and experience to test the software. It helps to expand the imagination of the testers by executing more test cases, thus enhancing the overall quality of the software. 7. Adhoc Testing Adhoc testing is a type of software testing is a type of software testing is completed to find any loophole in the system. For this reason, it is also known as Random or Monkey testing. Adhoc testing is not performed in a structured way so it is not based on any methodological approach. That's why Adhoc testing is a type of Unstructured Software Testing. It can be performed within a very limited time. Helps to create unique test cases. This test helps to build a strong product that is less prone to future problems. This testing can be performed at any time during Software Testing that uncovers vulnerabilities in the system and determines that the data and resources of the system are protected from possible intruders. It ensures that the software system and application are free from any threats or risks that can cause a loss. Security testing of any system is focused on finding all possible loopholes and weaknesses of the system that might result in the loss of information or repute of the organization Advantages of Security Testing: Identifying vulnerabilities: Security testing helps identify vulnerabilities in the system by identifying and fixing vulnerabilities and potential threats. Ensuring compliance: Security testing helps ensure that the system meets relevant security standards and regulations, such as HIPAA, PCI DSS, and SOC2. 9. Globalization Testing is a type of software testing that is performed to ensure the system or software application can function independently of the geographical and cultural environment. It ensures that the application can be used all over the world and accepts all language texts. Nowadays with the increase in various technologies, every software product is designed in such a way that it is a globalized software product. scalable products: It makes the software product more flexible and scalable. Save time: It saves overall time and effort for software testing. 10. Alpha Testing Alpha testing is a type of validation testing. It is a type of acceptance testing that is done before the product is released to customers. It is typically done by QA people. Example: When software testing is performed internally within the organization. Advantages of Alpha testing:Early Bug Detection : Identifies and addresses bugs early in the development process, reducing the risk of major issues later. Improved Quality Enhances the overall quality and stability of the software before it reaches real users. Cost-Effective : Fixing issues during alpha testing is generally cheaper than addressing them after release. Usability Insights : Provides valuable feedback on the user experience, allowing for improvements in usability and interface design. Requirement Validation Ensures the software meets business and user requirements, aligning it more closely with intended goals.11. Beta Testing is performed for the limited number of people. Advantages of Beta Testing: It reduces product failure risk via customer validation. Beta Testing allows a company to test post-launch infrastructure. It helps in improving product failure risk via customer set and increases customer satisfaction.12. Object-Oriented Testing Object-Oriented Testing is a combination of various testing techniques that help to verify and validate object-oriented software. This testing of Code, Integration testing, System testing, User Testing. 13. Recovery TestingRecovery TestingRecovery Testing is a type of software testing that checks how well an application can recover from crashes, failures, or other unexpected issues. It involves intentionally causing problems in the software to see if it can quickly and effectively return to normal operation. Advantages of Recovery TestingRisk elimination is possible as the potential flaws are detected and removed from the system. Improved performance as faults are removed, and the system becomes more reliable for users. Identifies Weaknesses: Helps uncover potential weaknesses or vulnerabilities in the system that could lead to failures. Enhances User Experience: Ensures a smooth user experience by minimizing downtime and data loss during unexpected events. Improves System Stability: Compatibility TestingCompatibility Testing is software testing is software testing is category, and it is performed on an application to check its compatibility (running capability) on different platforms/environments. This testing is done only when the application becomes stable. Advantages of Compatibility Testing:Works Everywhere: Compatibility testing verify your app runs smoothly on all platforms, devices, and browsers, helping you reach a wider audience. Better User Experience: By checking how your app behaves in different environments, you can spot and fix issues that could frustrate users, giving them a more seamless experience. Catches Hidden Problems: This type of testing helps you find issues that might pop up due to different hardware, software, or network setups, which could go unnoticed if you only test on one system. Consistency across all devices and browsers, so users have the same experience no matter how they access it. Avoids Problems After Launch: By finding and fixing compatibility issues early, you prevent headaches and customer complaints after your app is released, making the whole process smoother. 15. Volume TestingVolume Testing Volume Testing application with a certain amount of data. The amount used in volume testing could be a database size or it could also be the size of an interface file that is the subject of volume testing. Advantages of Volume testing is also helpful in a rapid start for scalability plans. Volume testing also helps in early identification of bottlenecks. Volume testing ensures that the system is capable of real world usage 16. Installation Testing Installation testing checking full or partial upgrades and other features install/uninstall processes are included. Advantages of Installation Testing: The first biggest advantage is that it verifies the designs of apps and software on a basic level of test performance. It's a very quick and handy method to check the version of the softwareThe greater output results of installation testing help the developer to improve the app or software quality: Installation issues and errors, improving the software's overall quality. 17. Localization TestingLocalization Testing is a Type of Software Testing that is performed to verify the quality of a product for a specific culture or locale. Localization testing cost. Localization testing cost. Localization testing reduces the overall support cost. It helps in reducing the time for testing.Localization testing has more flexibility and scalability.18. A/B TestingA/B Testing or split testing, in a nutshell, is a means to compare two iterations of an email, website, or other marketing asset and assess the performance differences between them.Advantages of A/B Testing:Enhanced Content: For instance, while testing marketing content, users must be shown a list of potential upgrades. Reduces Costs: Companies can save money by using A/B testing to find procedures that produce better results. One marketing effort will always be superior to the other; Low Risks: You can lower risks by using A/B tests. You can run an A/B test to observe how a new update or component on your product affects your system and how users respond to it if you're unsure of how it will perform. 19. Mutation TestingMutation Testing was a type of Software tests and also evaluate the quality of already existing software tests. Mutation testing is related to modification a program in small ways.Advantages of Mutation Testing: It brings a good level of error detection in the program. It discovers ambiguities in the source code. 20. Graphical User Interface TestingGraphical User Interface Testing is the process for ensuring proper functionality of the graphical user interface (GUI) for a specific application. GUI testing generally evaluates a design of elements such as layout, colors and also fonts, font sizes, labels, text boxes, text formatting, captions, buttons, lists, icons, links, and content. Advantages of Graphical User Interface Testing: It provides a customizable test report. It is run tests in parallel or distributed on a Selenium Grid with built-in Selenium Grid with built-in Selenium Webdriver. It allows you to test the functionality from a user's perspective. Sometimes the internal functions of the system work correctly but the user interface doesn't then GUI testing is good to have in addition to the other types. How to Automate Your Tests? To automate your tests, you'll need to research or ask in your programming language. For example, you can use PHPUnit for PHP, Mocha for JavaScript, and RSpec for Ruby. There are many frameworks available, so you might need to research or ask other developers to find the best one for your needs. Once your tests can run from your terminal, you can automate them using a continuous integration (CI) server like Bamboo or a cloud service like Bitbucket Pipelines. These tools monitor your code repository and run your tests automatically whenever new changes are pushed. Advantages of Software Testing Below are the benefits of software testing: Customer Satisfaction: Software testing makes sure that your application works exactly as it should, meeting the needs of your customers. This leads to higher satisfaction and trust in your product. Cost-Effective: By catching issues early, software testing helps you save money on future fixes and maintenance, making the whole development process more efficient. Quality Product: Testing ensures that your product is of high quality by finding bugs, checking for compatibility, and ensuring it meets user expectations. Low Failure: Testing helps identify potential weak points in your application, making it more stable and reliable, which reduces the chance of failures, especially under heavy use. Bug-Free Application: The main goal of software testing is to find and fix bugs. While it's hard to achieve a 100% bug-free application: The main goal of software testing is to find and fix bugs. from threats and that sensitive data is protected, especially in critical sectors like banking. Easy Recovery: Testing helps ensure that if your application fails, it can quickly recover and return to normal functionality, minimizing downtime. Speed Up the Development Process: When testing is done alongside development, issues are caught early, allowing for quicker fixes and faster delivery of the final product. Early Defect Detection: By starting testing early in development, issues are found and fixed sooner, saving time and preventing bigger problems later in the process. Reliable Product: giving users a smooth and dependable experience. Disadvantages of Software Testing Time-Consuming and adds to the project cost. This can be found. Can be difficult to fully test complex systems. Potential for human error during the testing process.