

Types of Software Testing

Principles of Software Testing

- All the tests should meet the customer's requirements.
- To make our software testing should be performed by a third party.
- Exhaustive testing is not possible. As we need the optimal amount of testing based on the risk assessment of the application.
- All the tests to be conducted should be planned before implementing it
- It follows the Pareto rule(80/20 rule) which states that 80% of errors come from 20% of program components.
- Start testing with small parts and extend it to large parts.
- Types of Testing

Fundamental categories of software testing:

1. **Manual Testing**
2. **Automation Testing**

1. Manual Testing

- Manual testing is a technique to test the software that is carried out using the functions and features of an application.
- In manual software testing, a tester carries out tests on the software by following a set of predefined test cases.
- In this testing, testers make test cases for the codes, test the software, and give the final report about that software.
- Manual testing is time-consuming because it is done by humans, and there is a chance of human errors.

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 require any high-level skill or a specific type of tool.
- **No coding is required:** No programming knowledge is 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.

2. Automation Testing

- Automated Testing is a technique where the Tester writes scripts on their own and uses suitable Software or Automation Tool to test the software.
- It is an Automation Process of a Manual Process.
- It allows for executing repetitive tasks without the intervention of a Manual Tester.

Advantages of Automation Testing:

- **Simplifies Test Case Execution:** 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 ensures that there is equal focus on all the areas of the testing, thus ensuring the best quality end product.
- **Increases amount of test coverage:** Using 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 automation testing, everything is automated from test case creation to execution thus there are no chances for human error due to neglect. This reduces the necessity for fixing glitches in the post-release phase.

Types of Manual Testing:

1. **White Box Testing**
2. **Black Box Testing**
3. **Gray Box Testing**
1. **White Box Testing**

- White box testing techniques analyze the internal structures the used data structures, internal design, code structure, and the working of the software rather than just the functionality as in black box testing.
- It is also called glass box testing clear box testing or structural testing. White Box Testing is also known as transparent testing or open box testing.
- The tester has access 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 Testing:** White box testing is 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.
- It 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.
- Test cases are easily reproducible.
- It is used to find the ambiguity and contradictions in the functional specifications.

3. Gray Box Testing

- Gray Box Testing is a software testing technique that is a combination of the Black Box Testing technique and the White Box Testing technique.
- In the Black Box Testing technique, the tester is unaware of 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.

Advantages of Gray Box Testing:

- Clarity of goals: Users and developers have clear goals while doing testing.
- Done from a user perspective: Gray box testing is mostly done from the user 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.
- Improved product quality: Overall quality of the product is improved.

Types of Black Box Testing

1. Functional Testing
2. Non-Functional Testing

1. Functional Testing

- Functional Testing is a type of Software Testing in which the system is tested against the functional requirements and specifications.
- Functional testing ensures that the requirements or specifications are properly satisfied by the application.
- 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.

Benefits of Functional Testing

- **Bug-free product:** Functional testing ensures the delivery of a bug-free and high-quality product.

- **Customer satisfaction:** It 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 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.

2. Non-Functional Testing

- **Non-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.
- Non-functional testing is a software 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.

Benefits of Non-functional Testing

- **Improved performance:** Non-functional testing checks the performance of the system and determines the performance bottlenecks that can affect the performance.
- **Less time-consuming:** Non-functional testing is overall less time-consuming than the other 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 Testing
2. Integration Testing
3. System Testing

1. Unit Testing

- Unit Testing is a method of testing individual units or components of a software application.

- It is typically done by developers and is used to ensure that the individual units of the software 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.

Advantages of Unit Testing:

- It helps to identify bugs early in the development process before they become more difficult and expensive to fix.
- It helps to ensure that changes to the code do not introduce new bugs.
- It makes the code more modular 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 include **JUnit**, **NUnit**, and **xUnit**.

- It's important to keep in mind that 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.

Example:

1. In a program we are checking if the loop, method, or function is working fine.
2. Misunderstood or incorrect, arithmetic precedence.
3. Incorrect initialization.

2. Integration Testing

- Integration Testing is a method of testing how different units or components of a software application interact with each other.
- It is used to identify and resolve any issues that may arise when different units of the software are combined.
- Integration testing is typically done after unit testing and before functional testing and is used to verify that the different units of the software work together as intended.

Different Ways of Performing Integration Testing:

- Top-down integration testing: It starts with the highest-level modules and differentiates them from lower-level modules.
- Bottom-up integration testing: It starts with the lowest-level modules and integrates them with higher-level modules.

- **Big-Bang integration testing:** It combines all the modules and integrates them all at once.
- **Incremental integration testing:** It 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 are combined.
- It helps to ensure that the different units of the software work together as intended.
- It helps to improve the overall reliability 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, integration testing is only one aspect of software testing and it should be used in combination with other types of testing such as unit 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 testing in which a group of components is combined to produce output.

Integration testing is of four types: (i) Top-down (ii) Bottom-up (iii) Sandwich (iv) Big-Bang

Example:

1. **Black Box testing:** It is used for validation. In this, we ignore internal working mechanisms and focus on “what is the output?”
2. **White box testing:** It is used for verification. In this, we focus on internal mechanisms i.e. how the output is achieved.

3. System Testing

- System 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 testing is performed after the integration testing and before the acceptance testing.
- **System Testing** is a type of software testing that is performed on a completely integrated system to evaluate the compliance of the system with the corresponding requirements.
- In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are 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.

Types of Integration Testing

- 1. Incremental Testing**
- 2. Non-Incremental Testing**

1. Incremental Testing

- Like development, testing is also a phase of SDLC.
- Different tests 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 **Incremental Testing** which is performed after unit testing
- Several stubs and drivers are used to test the modules one after one which helps in discovering errors and defects in the specific modules.

Advantages 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**
- 2. Bottom-up Integration Testing**

1. Top-down Integration Testing

Top-down testing is a type of incremental integration testing approach in which testing is done by integrating or joining two or more modules by moving down from top to bottom through

the control flow of the architecture structure. In these, high-level 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 Integration Testing

1. There is no need to write drivers.
2. Interface errors are identified at an early stage and fault localization is also easier.
3. Low-level utilities that are not important are not tested well and high-level testers are tested well in an appropriate manner.
4. Representation of test cases is easier and simpler once Input-Output functions are added.

2. Bottom-up Integration Testing

Bottom-up Testing is a type of incremental integration testing approach in which testing is done by integrating or joining two or more modules by moving upward from bottom to top through the control flow of the architecture structure. In these, low-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 in high success rates with long-lasting results.

Advantages of Bottom-up Integration Testing

- 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

1. Performance Testing
2. Usability Testing
3. Compatibility Testing

1. Performance Testing

Performance Testing is a type of software testing that ensures software applications perform properly under their expected workload. It is a testing technique carried out to determine system performance in terms of sensitivity, reactivity, and stability under a particular workload.

Performance testing is a type of software testing that focuses on evaluating the performance and scalability of a system or application. The goal of performance testing is to identify bottlenecks, measure system performance under various loads and conditions, and ensure that the system can handle the expected number of users or transactions.

Advantages of Performance Testing

- Performance testing ensures the speed, load capability, accuracy, and other performances of the system.
- It identifies, monitors, and resolves 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:
- **Identifying bottlenecks:** Performance 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.

2. Usability Testing

You design a product (say a refrigerator) and when it becomes completely ready, you need a potential customer to test it to check it working. To understand whether the machine is ready to come on the market, potential customers test the machines. Likewise, the best example of usability testing is when the software also undergoes various testing processes which is performed by potential users before launching into the market. It is a part of the software development lifecycle (SDLC).

Advantages and Disadvantages of 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. During a Usability test, you can,

- Learn if participants will be able to complete the specific task completely.
- identify how long it will take to complete the specific task.
- Gives excellent features and functionalities to the product
- Improves user satisfaction and fulfills requirements based on user feedback
- The product becomes more efficient and effective

3. Compatibility Testing

Compatibility testing 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 becomes stable. This means simply this compatibility test aims to check the developed software 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.

There are 4 Types of Performance Testing

- 1. Load Testing**
- 2. Stress Testing**
- 3. Scalability Testing**
- 4. Stability Testing**

1. Load Testing

Load testing determines the behaviour of the application when multiple users use it at the same time. It is the response of the system measured under varying load conditions.

1. The load testing is carried out for normal and extreme load conditions.
2. Load testing is a type of performance testing that simulates a real-world load on a system or application to see how it performs under stress.
3. The goal of load testing is to identify bottlenecks and determine the maximum number of users or transactions the system can handle.
4. 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 of Load Testing:

Load testing has several advantages that make it an important aspect of software testing:

1. **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.
2. **Improved scalability:** By identifying the system's maximum capacity, load testing helps ensure that the system can handle an increasing number 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.
3. **Improved reliability:** Load testing helps identify any potential issues that may occur under heavy load conditions, such as increased error rates or slow response times. This helps ensure that the system is reliable and stable when it is deployed to production.

2. Stress Testing

In Stress Testing, we give unfavourable conditions to the system and check how it perform in those conditions.

Example:

1. Test cases that require maximum memory or other resources are executed.
2. Test cases that may cause thrashing in a virtual operating system.
3. Test cases that may cause excessive disk requirement Performance Testing.

It is designed to test the run-time 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

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 up or 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. Scalability Testing is defined as the ability of a network, system, application, product or a process to perform the function correctly when changes are made in the size or volume of the system to meet a growing need. It ensures that a software product can manage the scheduled increase in user traffic, data volume, transaction counts frequency, and many other things. It tests the system, processes, or database's ability to meet a growing need.

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.

4. Stability Testing

Stability Testing is a type of Software Testing to checks the quality and behavior of the software under different environmental parameters. It is defined as the ability of the product to continue to function over time without failure.

It is a Non-functional Testing technique that focuses on stressing the software component to the maximum. Stability testing is done to check the efficiency of a developed product beyond normal operational capacity which is known as break point. It has higher significance in error handling, software reliability, robustness, and scalability of a product under heavy load rather than checking the system behaviour under normal circumstances.

Stability testing assesses stability problems. This testing is mainly intended to check whether the application will crash at any point in time or not.

Advantages of Stability Testing

1. It gives the limit of the data that a system can handle practically.
2. It provides confidence on the performance of the system.
3. It determines the stability and robustness of the system under load.
4. Stability testing leads to a better end-user experience.

Other Types of Testing:

1. Smoke Testing
2. Sanity Testing
3. Regression Testing
4. Acceptance Testing
5. User Acceptance Testing
6. Exploratory Testing
7. Adhoc Testing
8. Security Testing
9. Globalization Testing
10. Regression Testing
11. Alpha Testing
12. Beta Testing
13. Object-Oriented Testing

1. Smoke Testing

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

1. Smoke testing is easy to perform.
2. It helps in identifying defects in the early stages.
3. It improves the quality of the system.
4. Smoke testing reduces the risk of failure.
5. Smoke testing makes progress easier to access.

2. Sanity Testing

It is a **subset** of regression testing. Sanity testing is performed to ensure that the code changes that are made are working properly. Sanity 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 is generally performed on a build where the production deployment is required immediately like a critical bug fix.

Advantages of Sanity Testing

- Sanity 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, the project is rejected which is helpful in saving time for execution of regression tests.
- This testing technique is not so expensive when compared to another type of testing.
- It helps to identify the dependent missing objects.

3. Regression Testing

The process of testing the modified parts of the code and the parts that might get affected due to the modifications ensures that no new errors have been introduced in the software after the modifications have been made. Regression means the return of something and in the software field, it refers to the return of a bug.

Advantages of Regression Testing

- It ensures that no new bugs have been introduced after adding new functionalities to the system.
- As most of the test cases used in Regression Testing are selected from the existing test suite, and we already know their expected outputs. Hence, it can be easily automated by the automated tools.
- It helps to maintain the quality of the source code.

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

1. This testing helps the project team to know the further requirements of the users directly as it involves the users for testing.
2. Automated test execution.
3. It brings confidence and satisfaction to the clients as they are directly involved in the testing process.
4. It is easier for the user to describe their requirement.

5. 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** is a testing methodology where clients/end users participate in product testing to validate the product against their requirements.
- It is done at the client's site on the developer's site.
- For industries such as medicine or aerospace, contractual and regulatory compliance testing, and operational acceptance tests are also performed as part of user acceptance tests.
- UAT is context-dependent and UAT plans are prepared based on requirements and are not required to perform all kinds of user acceptance tests and are even coordinated and contributed by the testing team.

6. Exploratory Testing

Exploratory 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 testing. In exploratory testing, software developers use their learning, knowledge, skills, and abilities to test the software developed by themselves. Exploratory testing checks the functionality and operations of the software as well as identifies the functional and technical faults in it. Exploratory testing aims to optimize and improve the software in every possible way.

Advantages of Exploratory Testing

- **Less preparation required:** It takes no preparation as it is an unscripted testing technique.
- **Finds critical defects:** Exploratory testing involves an investigation process that helps to find critical defects very quickly.
- **Improves productivity:** In 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 that is performed informally and randomly after the formal 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.

Advantages of Adhoc testing

- The errors that cannot be identified with written test cases can be identified by Adhoc 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 SDLC.

8. Security Testing

Security Testing is a type of 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 reputation of the organization.

Advantages of Security Testing

1. **Identifying vulnerabilities:** Security testing helps identify vulnerabilities in the system that could be exploited by attackers, such as weak passwords, unpatched software, and misconfigured systems.
2. **Improving system security:** Security testing helps improve the overall security of the system by identifying and fixing vulnerabilities and potential threats.
3. **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

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.

Benefits of Globalization Testing

- **Helps to create scalable products:** It makes the software product more flexible and scalable.
- **Save time:** It saves overall time and effort for software testing.
- **Reduce time for localization testing:** Globalization testing helps to reduce the time and cost of localization testing.

10. Regression Testing

Regression Testing is a method of testing that is used to ensure that changes made to the software do not introduce new bugs or cause existing functionality to break. It is typically done after changes have been made to the code, such as bug fixes or new features, and is used to verify that the software still works as intended.

Regression testing can be performed in different ways, such as:

- **Retesting:** This involves testing the entire application or specific functionality that was affected by the changes.

- **Re-execution:** This involves running a previously executed test suite to ensure that the changes did not break any existing functionality.
- **Comparison:** This involves comparing the current version of the software with a previous version to ensure that the changes did not break any existing functionality.

Advantages of Regression Testing

- It helps to ensure that changes made to the software do not introduce new bugs or cause existing functionality to break.
- It helps to ensure that the software continues to work as intended after changes have been made.
- It helps to improve the overall reliability and stability of the software.
- It's important to keep in mind that regression testing is an ongoing process that should be done throughout the software development.
- lifecycle to ensure that the software continues to work as intended. It should be automated as much as possible to save time and resources. Additionally, it's important to have a well-defined regression test suite that covers

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.

11. 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 organisation.

12. Beta Testing

The beta test is conducted at one or more customer sites by the end-user of the software. This version is released for a limited number of users for testing in a real-time environment.

Example:

When software testing is performed for the limited number of people.

13. Object-Oriented Testing

Object-Oriented testing is a combination of various testing techniques that help to verify and validate object-oriented software. This testing is done in the following manner:

- Testing of Requirements,

- Design and Analysis of Testing,
- Testing of Code,
- Integration testing,
- System testing,
- User Testing.

Advantages of Software Testing

1. Improved software quality and reliability.
2. Early identification and fixing of defects.
3. Improved customer satisfaction.
4. Increased stakeholder confidence.
5. Reduced maintenance costs.
6. Customer Satisfaction
7. Cost Effective
8. Quality Product
9. Low Failure
10. Bug-Free Application
11. Security
12. Speed Up the Development Process
13. Early Defect Detection
14. Reliable Product

Disadvantages of Software Testing

1. Time-Consuming and adds to the project cost.
2. This can slow down the development process.
3. Not all defects can be found.
4. Can be difficult to fully test complex systems.
5. Potential for human error during the testing process.