



Kalinga University

Faculty of Computer Science & IT

Course- BCAAIML

Subject- Software Engineering and Testing

Course Code – BCAAIML503

Sem- 5th

Unit IV

Types of Software Testing

Software Testing always **aligns** with the **Customer's Requirement**, which they want. 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 principle of software testing, which guides you throughout the process of Software Development.

These principles will be helpful for the tester to identify the software issue earlier and verify the Software meets to the expectations.

1. Testing Shows the Presence of Defects
2. Exhaustive Testing is Not Possible
3. Early Testing
4. Defect Clustering
5. Pesticide Paradox
6. Testing is Context-Dependent
7. Absence of Errors Fallacy

Different Types of Software Testing

Here are the **Types of Software Testing** mainly categorized into the two domain, which are below.

1. Manual Testing

Manual testing is a technique to test the software that is carried out using the functions and features of an application. Which means 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 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 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:

- **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 changes for human error due to neglect. This reduces the necessity for fixing glitches in the post-release phase.

Manual vs. Automated testing

Here is the table of comparing [**Manual Testing and Automated Testing:**](#)

Parameters	Manual Testing	Automation Testing
Definition	In manual testing, the test cases are executed by the human tester.	In automated testing, the test cases are executed by the software tools.
Processing Time	Manual testing is time-consuming.	Automation testing is faster than manual testing.
Resources requirement	Manual testing takes up human resources.	Automation testing takes up automation tools and trained employees.
Exploratory testing	Exploratory testing is possible in manual testing.	Exploratory testing is not possible in automation testing.
Framework requirement	Manual testing doesn't use frameworks.	Automation testing uses frameworks like Data Drive, Keyword, etc.

Types of Manual Testing

1. White Box Testing

White Box Testing is a software testing technique that involves testing the internal structure and workings of a software application. 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 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.
- 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.

1. 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.
2. The internal structure is partially known in Gray Box Testing.
3. This includes access to internal data structures and algorithms to design the test cases.

Advantages of Gray Box Testing:

1. **Clarity of goals:** Users and developers have clear goals while doing testing.
2. **Done from a user perspective:** Gray box testing is mostly done from the user perspective.
3. **High programming skills not required:** Testers are not required to have high programming skills for this testing.

4. **Non-intrusive:** Gray box testing is non-intrusive.
5. **Improved product quality:** Overall quality of the product is improved.

Types of Black Box 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. The article focuses on discussing function testing.

Advantages 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.

Advantages 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

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:

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 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.

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:

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.
- 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.

4. End-to-end Testing

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 Testing

Acceptance 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 Testing:

- This testing helps the project team to know the further requirements from the 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:

1. Incremental Testing

In **Incremental Testing** Like development, testing is also a phase of **SDLC (Software Development Life Cycle)**. 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 integration 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

Top-down Integration 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 Integration 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

Here are the Types of Non-Functional 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.

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

Usability Testing in software 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. Several tests are performed on a product before deploying it.

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.

- **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 and loyalty, leading to increased engagement and retention.

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.

Types of Performance Testing

Here are the Types of Performance testing:

1. Load Testing

Load Testing determines the behavior 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

Stress Testing is defined as types of **software testing** that verifies the stability and reliability of the system. This test particularly determines 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:

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.

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 check the quality and behavior of the software in different environmental parameters. It is defined as the ability of the product to continue to function over time without failure.

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.