

SOFTWARE VERIFICATION, VALIDATION & TESTING

*What is Software Verification and Validation?
What is the difference between them?*

Verification

Verification is a process that determines the quality of the software. Verification includes all the activities related with producing high quality software such as- testing, inspection, design analysis, specification analysis, and so on. Verification is a relatively objective process.

Advantages of Verification : Verification helps to decrease the number of the defects that may encounter in the later stages of development.

It verifies that product at the starting phase of the development will help in understanding the product in a more comprehensive way.

It helps to reduce the chances of failures in the software application or product.

It helps in building the product as per the customer specifications and needs.

Validation

Validation is a process in which the requirements of the customer are actually met by the software functionality. Validation is done at the end of the development process and takes place after verifications are completed.

Advantages of Validation : During verification if some defects are missed, then during the validation process they can be caught as failures.

Validation helps in building the right product as per the customer's requirement which in turn will satisfy their business process needs.

If during verification some specification is misunderstood and development has already occurred then during the validation process the difference between the

actual result and expected result can be identified and corrective action taken.

Validation is done during testing like feature testing, integration testing, system testing, load testing, compatibility testing, stress testing, etc.

Difference between Verification and Validation :

Verification	Validation
To check development of product or not	To check whether the developed product is right
Verification is also known as static testing	Validation is also known as dynamic testing
Verification includes different methods like Inspections, Reviews, and Walkthroughs	Validation includes testing like functional testing, system testing, integration, and User acceptance testing.
It is a process of checking the work-products (not the final product) of a development cycle to decide whether the product meets the specified requirements.	It is a process of checking the software during or at the end of the development cycle to decide whether the software follow the specified business requirements.
Related with Quality assurance	Related with Quality control
There is execution of code	There is execution of code
Bugs can be detected at early development phase of the product.	Bugs cannot be detected in the verification process.
Verification testing is executed by the Quality assurance team to make sure that the product is developed according to customers' requirements.	Validation testing is executed by the testing team to test the application.
Verification is done before the validation testing.	After verification testing, validation testing is done.
In this type of testing, we can verify that the inputs follow the outputs or not.	In this type of testing, we can validate that the user accepts the product or not.

2. ***Explain the Evolutionary model in detail. Also state advantages and disadvantages of the same.***

Evolutionary model is a combination of Iterative and Incremental model of software development life cycle. If you want to deliver your system in a large scale, incremental process gives the output over time.

tools to the organization. With quality management standards, organizations can steadily assure that their software products achieve an acceptable level of quality.

Example : ISO 9000-3 and the Capability Maturity Model (CMM)

Project Process Standards : These focus on the methodologies for implementing the software development and maintenance projects. These standards include the following :

- (1) The steps to be taken
- (2) Design documentation requirements
- (3) Contents of design documents
- (4) Design reviews and review issues
- (5) Software testing to be performed
- (6) Testing topics

ISO 9001 Certification : ISO (the International Organization for Standardization) is a worldwide society of national standards bodies. ISO technical committees prepare the International Standards. ISO collaborates closely with the International Electro-technical Commission (IEC) on all matters of electro-technical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2. Draft of the International Standards adopted by the technical committees is circulated to the member bodies for voting.

Process Approach : This International Standard promotes the adoption of a process approach when developing, implementing, and improving the effectiveness of a quality management system, to enhance customer satisfaction by meeting the customer requirements. For an organization to function effectively, it has to determine and manage numerous linked activities. An activity or set of activities using resources, and managed in order to enable the transformation of inputs into outputs, can be considered as a process.

Often the output from one process directly forms the input to the next. The application of a system of processes within an organization, together with the identification

and interactions of these processes, and their management to produce the desired outcome, can be referred to as the “process approach”.

An advantage of the process approach is the ongoing control that it provides over the linkage between the individual processes within the system of processes, as well as over their combination and interaction.

When used within a quality management system, such an approach emphasizes the importance of the following :

- (1) Understanding and meeting the requirements
- (2) Need to consider the processes in terms of added value
- (3) Obtain the results of process performance and effectiveness
- (4) Continual improvement of processes based on objective measurement

□□

SO

V

What is Software
What is the difference

Verification

Verification of the software related with processes as testing, integration analysis, and so on process.

Advantages of verification decrease the number of defects in the later stages.

It verifies the correctness of development work more comprehensively.

It helps to identify the defects in the software application.

It helps in specifying the requirements.

Validation

Validation is the process of verifying the customer requirements against the functionality. It is performed during the development process after completion of each module.

Advantages of validation are defects are identified early, they can be caught and corrected.

Validation is the process of verifying the customer requirements against the functionality. It is performed during the development process after completion of each module.

SOFTWARE VERIFICATION, VALIDATION & TESTING

***What is Software Verification and Validation?
What is the difference between them?***

Verification

Verification is a process that determines the quality of the software. Verification includes all the activities related with producing high quality software such as- testing, inspection, design analysis, specification analysis, and so on. Verification is a relatively objective process.

Advantages of Verification : Verification helps to decrease the number of the defects that may encounter in the later stages of development.

It verifies that product at the starting phase of the development will help in understanding the product in a more comprehensive way.

It helps to reduce the chances of failures in the software application or product.

It helps in building the product as per the customer specifications and needs.

Validation

Validation is a process in which the requirements of the customer are actually met by the software functionality. Validation is done at the end of the development process and takes place after verifications are completed.

Advantages of Validation : During verification if some defects are missed, then during the validation process they can be caught as failures.

Validation helps in building the right product as per the customer's requirement which in turn will satisfy their business process needs.

If during verification some specification is misunderstood and development has already occurred then during the validation process the difference between the

actual result and expected result can be identified and corrective action taken.

Validation is done during testing like feature testing, integration testing, system testing, load testing, compatibility testing, stress testing, etc.

Difference between Verification and Validation :

Verification	Validation
To check development of product or not.	To check whether the developed product is right.
Verification is also known as static testing.	Validation is also known as dynamic testing.
Verification includes different methods like Inspections, Reviews, and Walkthroughs.	Validation includes testing like functional testing, system testing, integration, and User acceptance testing.
It is a process of checking the work-products (not the final product) of a development cycle to decide whether the product meets the specified requirements.	It is a process of checking the software during or at the end of the development cycle to decide whether the software follows the specified business requirements.
Related with Quality assurance	Related with Quality control
There is execution of code	There is execution of code
Bugs can be detected at early development phase of the product.	Bugs cannot be detected in the verification process.
Verification testing is executed by the Quality assurance team to make sure that the product is developed according to customers' requirements.	Validation testing is executed by the testing team to test the application.
Verification is done before the validation testing.	After verification testing, validation testing is done.
In this type of testing, we can verify that the inputs follow the outputs or not.	In this type of testing, we can validate that the user accepts the product or not.

2. Explain the Evolutionary model in detail. Also state advantages and disadvantages of the same.

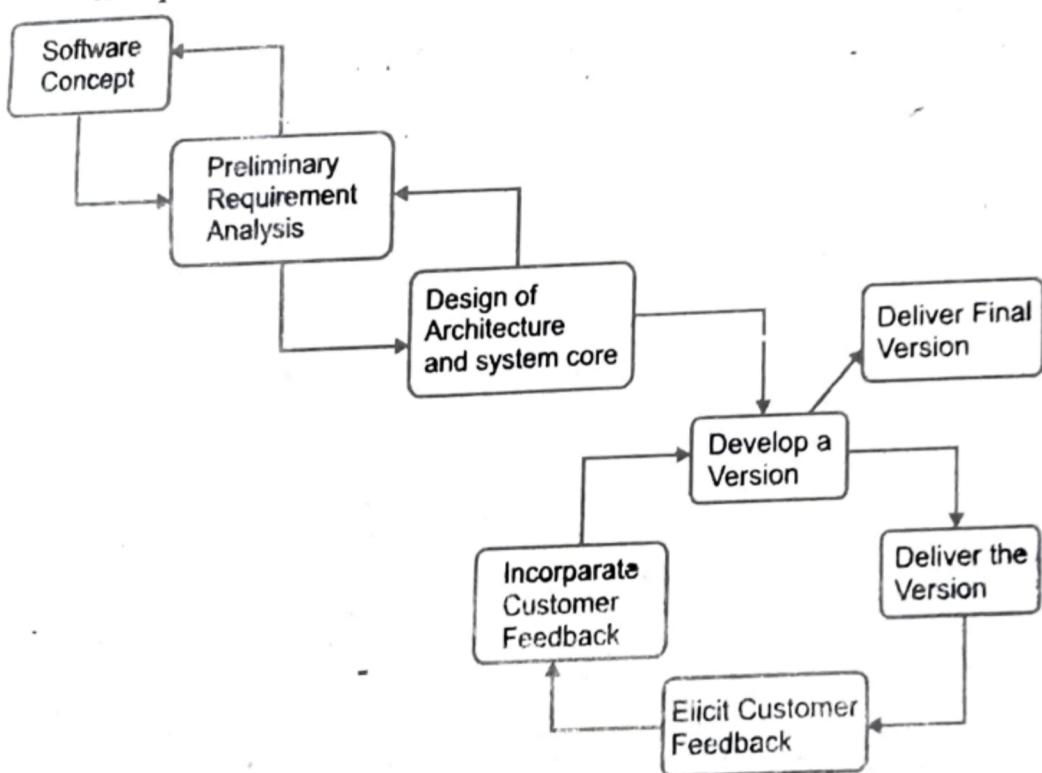
Evolutionary model is a combination of Iterative and Incremental model of software development life cycle. If you want to deliver your system in a large scale, incremental process gives the output over time.

It is better for software products to set their features which are redefined during development because of user feedback and other factors. In the Evolutionary development model, there is division of the development cycle into smaller, incremental waterfall models in which users are able to get access to the product at the end of each cycle.

During the evolutionary model, feedback is provided by users. Feedback is provided by the users on the product for the planning stage of the next cycle and the development team responds, often by changing the product, plan or process.

Therefore, the software product evolves with time.

All the models have the disadvantage that the duration of time from start of the project to the delivery time of a solution is very high. Evolutionary model solves this problem in a different approach.



(Figure)

According to the Evolutionary model, the total work is broken down into smaller chunks and priority is assigned to them and then delivering those chunks to the customer one by one.

If the number of chunks are huge, then larger the number of deliveries made to the customer. Here is the

main advantage of the Evolutionary model that it increases customer's confidence. Because it constantly gets quantifiable goods or services from the beginning of the project to verify and validate his requirements. The model allows for changing requirements as well as all work in broken down into maintainable work chunks.

Application of Evolutionary Model :

- (1) It is used in large projects where you can easily find modules for incremental implementation. Evolutionary model is commonly used when the customer wants to start using the core features instead of waiting for the full software.
- (2) Evolutionary model is also used in object oriented software development because the system can be easily portioned into units in terms of objects.

Advantages :

- (1) In evolutionary model, a user gets a chance to experiment partially developed system.
- (2) It reduces the error because the core modules get tested thoroughly.

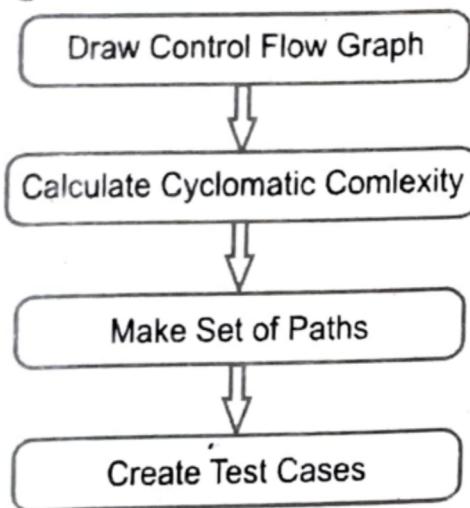
Disadvantages : Sometimes it is hard to divide the problem into several versions that would be acceptable to the customer which can be incrementally implemented and delivered.

3. Explain Path testing in detail. What are the different techniques used in the Path testing?

Path Testing is a method that is used to design the test cases. In path testing method, the control flow graph of a program is designed to find a set of linearly independent paths of execution. In this method Cyclomatic Complexity is used to determine the number of linearly independent paths and then test cases are generated for each path.

It gives complete branch coverage but achieves that without covering all possible paths of the control flow graph. McCabe's Cyclomatic Complexity is used in path testing. It is a structural testing method that uses the source code of a program to find every possible executable path.

Path Testing Process :



(Figure)

- (1) **Control Flow Graph** : Draw the corresponding control flow graph of the program in which all the executable paths are to be discovered.
- (2) **Cyclomatic Complexity** : After the generation of the control flow graph, calculate the cyclomatic complexity of the program using the following formula.
- (3) **McCabe's Cyclomatic Complexity** = $E - N + 2P$
Where,
 E = Number of edges in control flow graph
 N = Number of vertices in control flow graph
 P = Program factor
- (4) **Make Set** : Make a set of all the path according to the control flow graph and calculated cyclomatic complexity. The cardinality of set is equal to the calculated cyclomatic complexity.
- (5) **Create Test Cases** : Create test case for each path of the set obtained in above step.

Path Testing Techniques :

- (1) **Control Flow Graph** : The program is converted into control flow graph by representing the code into nodes and edges.
- (2) **Decision to Decision Path** : The control flow graph can be broken into various Decision to Decision paths and then collapsed into individual nodes.
- (3) **Independent Paths** : Independent path is a path through a Decision to Decision path graph which

cannot be reproduced from other paths by other methods.

Advantages of Path Testing :

- (1) Path testing method reduces the redundant tests.
- (2) Path testing focuses on the logic of the programs.
- (3) Path testing is used in test case design.

4. What is Formal Verification-Proof of Correctness?

It is always mandatory to have formal proof of the correctness of verification process. Proof of correctness specifies that how users can interact with the software and how the software should behave when it is used correctly. If the software behaves incorrectly, it might take considerable amount of time to achieve the task or sometimes it is impossible to achieve it.

Below are some of the important rules for effective programming which are consequences of the program correctness theory.

- (1) Problem definition should be complete.
- (2) First develop the algorithm and then the program logic.
- (3) Reuse the proved models as much as possible.
- (4) Prove the correctness of algorithms during the design phase.
- (5) Developers should pay attention to the clarity and simplicity of your program.
- (6) Verifying each part of a program as soon as it is developed.

A proof of correctness is a mathematical proof that a computer program. When this program is executed, it gives the correct. Before proving a program correct, the theorem must be proved. There are two types of statements :

- (1) **Hypothesis** : The hypothesis of such a correctness theorem is typically a condition that the relevant program variables must satisfy immediately "before" the program is executed. This condition is called the "precondition".
- (2) **Thesis** : The thesis of the correctness theorem is typically a condition that the relevant program variables must satisfy immediately "after" execution

of the program. This latter condition is called the ‘post-condition’.

Thus the correctness theorem can be stated as:
“If the condition, “ V ”, is true before execution of the program, “ S ”, then the condition, “ P ”, will be true after execution of “ S ”.

Where “ V ” is pre-condition and “ P ” is post-condition.

Notation : Such a correctness theorem is usually written as $\{V\} S \{P\}$, where V , S and P have been explained above.

By “program variable” we broadly include input and output data, e.g., data entered via a keyboard, displayed on a screen or printed on paper. Any externally observable aspect of the program’s execution may be covered by the precondition and post-condition.

5. *What is Software Testing? What is the need of testing? Explain with suitable examples.*

Software Testing

Software Testing is a method used to check whether the developed software product meets the expected requirements and to ensure that software product is Defect free. It involves the execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

Here are the top reasons why the testing of the software is really important :

- (1) **Helps in Saving Money** : The testing of software has a wide array of benefits. The cost-effectiveness of the project happens to be one of the top reasons why companies go for software testing Services. The testing of software comprises of a bunch of projects. In case you find any bug in the early phases, fixing them costs a reduced amount of money.
- (2) **Security** : It is another crucial point why software testing should not be taken into consideration. It is considered to be the most vulnerable and sensitive part. There are a bunch of situations in which the information and details of the users are stolen and

they are used for the benefits. It is considered to be the reason why people look for the well tested and reliable products.

- (3) **Quality of the Product :** For ensuring that the specific product comes to life, it should work in accordance with the following. Following the needs of the product is a prerequisite as it is helpful in getting the prerequisite results. Products should be serving the user in one way or the other. It is a must that it is going to bring the value, as per the promise.
- (4) **Satisfaction of the Customer :** The primary objective of the owner of the products is offering the best satisfaction of the customers. The reason why it is necessary to opt for software testing is due to the fact that it offers the prerequisite and perfect user experience.
- (5) **Enhancing the Development Process :** With the aid of Quality Assurance, you can find a wide array of scenarios and errors, for the reproduction of the error.
- (6) **Easy While Adding New Features :** The more interconnected and older the code, the more difficult it is to change. Tests counteract this calcification tendency by allowing developers to confidently add new features.
- (7) **Determining the Performance of the Software :** If you find software or application that has low or reduced performance, you will find that it brings your reputation down in the market. Users are not going to trust any people. There are chances that the reputation of your organization is going to suffer.

6. What are the different types of Software Testing?

Testing is the process of executing a program with the aim of finding errors. To make our software perform well it should be error-free. If testing is done successfully it will remove all the errors from the software.

Principles of Testing :

- (1) All the test should meet the customer requirements
- (2) To make our software testing should be performed by a third party

- (3) Exhaustive testing is not possible. As we need the optimal amount of testing based on the risk assessment of the application.
- (4) All the test to be conducted should be planned before implementing it
- (5) It follows the Pareto rule (80/20 rule) which states that 80% of errors come from 20% of program components.
- (6) Start testing with small parts and extend it to large parts.

Types of Testing :

- (1) **Unit Testing** : 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.
- (2) **Integration Testing** : 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.
- (3) **Regression Testing** : 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.
- (4) **Smoke Testing** : This test is done to make sure that software under testing is ready or stable for further testing. It is called a smoke test as the testing an initial pass is done to check if it did not catch the fire or smoke in the initial switch on.
- (5) **Alpha Testing** : This is a type of validation testing. It is a type of acceptance testing which is done before the product is released to customers. It is typically done by QA people.
- (6) **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
- (7) **System Testing** : This software is tested such that it works fine for the different operating systems. It is

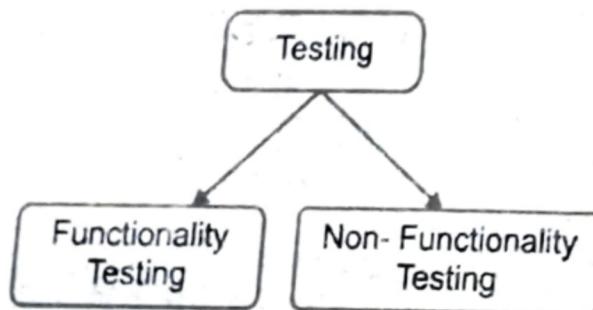
- covered under the black box testing technique. In this, we just focus on the required input and output without focusing on internal working.
- (8) **Stress Testing** : In this, we give unfavorable conditions to the system and check how they perform in those conditions.
 - (9) **Performance Testing** : 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.
 - (10) **Object-Oriented Testing** : This 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.

7. What is functional testing?

Software testing is a technique to check whether the actual result matches the expected result and to ensure that the software has not any defect or bug.

Software testing ensures that the application has not any defect or the requirement is missing to the actual need. Either manual or automation testing can do software testing.

There are two types of testing :



(Figure)

Functional Testing : It is a type of software testing which is used to verify the functionality of the software application, whether the function is working according to the requirement specification. In functional testing, each function tested by giving the value, determining the

output, and verifying the actual output with the expected value.

Functional testing performed as black-box testing which is presented to confirm that the functionality of an application or system behaves as we are expecting. It is done to verify the functionality of the application. Functional testing also called as black-box testing, because it focuses on application specification rather than actual code. Tester has to test only the program rather than the system.

The purpose of the functional testing is to check the primary entry function, necessarily usable function, the flow of screen GUI. Functional testing displays the error message so that the user can easily navigate throughout the application.

Process of Functional Testing : Testers follow the following steps in the functional testing :

- (1) Tester does verification of the requirement specification in the software application.
- (2) After analysis, the requirement specification tester will make a plan.
- (3) After planning the tests, the tester will design the test case.
- (4) After designing the test, case tester will make a document of the traceability matrix.
- (5) The tester will execute the test case design.
- (6) Analysis of the coverage to examine the covered testing area of the application.
- (7) Defect management should do to manage defect resolving.

Objectives of Functional Testing : The main objective of functional testing is checking the functionality of the software system. It concentrates on:

- (1) **Basic Usability :** Functional Testing involves the usability testing of the system. It checks whether a user can navigate freely without any difficulty through screens.
- (2) **Accessibility :** Functional testing test the accessibility of the function.
- (3) **Mainline Function :** It focuses on testing the main feature.

[E.12]

- (4) **Error Condition :** Functional testing is used to check the error condition. It checks whether the error message displayed.

8. What are the different types of functional testing?

The main objective of functional testing is to test the functionality of the component. Functional testing is divided into multiple parts.

Here are the following types of functional testing.

- (1) **Unit Testing :** Unit testing is a type of software testing, where the individual unit or component of the software tested. Unit testing, examine the different part of the application, by unit testing functional testing also done, because unit testing ensures each module is working correctly.
- (2) **Smoke Testing :** Functional testing by smoke testing. Smoke testing includes only the basic (feature) functionality of the system. Smoke testing is known as "Build Verification Testing."
- (3) **Sanity Testing :** Sanity testing involves the entire high-level business scenario is working correctly. Sanity testing is done to check the functionality/bugs fixed.
- (4) **Regression Testing :** This type of testing concentrate to make sure that the code changes should not side effect the existing functionality of the system. Regression testing specifies when bug arises in the system after fixing the bug, regression testing concentrate on that all parts are working or not. Regression testing focuses on is there any impact on the system.
- (5) **Integration Testing :** Integration testing combined individual units and tested as a group. The purpose of this testing is to expose the faults in the interaction between the integrated units
- (6) **White Box Testing :** White box testing is code-based testing, structural testing, extensive testing, and glass box testing, transparent box testing. The white box testing needs the analysis of the internal structure of the component or system.

- (7) **Black Box Testing :** It is also known as behavioral testing. In this testing, the internal structure/ design/ implementation not known to the tester. This type of testing is functional testing.
- (8) **User Acceptance Testing :** It is a type of testing performed by the client to certify the system according to requirement. The final phase of testing is user acceptance testing before releasing the software to the market or production environment.
- (9) **Retesting :** Retesting is a type of testing performed to check the test cases that were unsuccessful in the final execution are successfully pass after the defects fixed. Usually, tester assigns the bug when they find it while testing the product or its component. The bug allocated to a developer, and he fixes it. After fixing, the bug is assigned to a tester for its verification. This testing is known as retesting.
- (10) **Database Testing :** Database testing is a type of testing which checks the schema, tables, triggers, etc. of the database under test. Database testing may involve creating complex queries to load/stress test the database and check its responsiveness. It checks the data integrity and consistency.
- (11) **Ad-hoc Testing :** Ad-hoc testing is an informal testing type whose aim is to break the system. This type of software testing is unplanned activity. It does not follow any test design to create the test cases.
- (12) **Recovery Testing :** Recovery testing is used to define how well an application can recover from crashes, hardware failure, and other problems. The purpose of recovery testing is to verify the system's ability to recover from testing points of failure.
- (13) **Static Testing :** Static testing is a software testing technique by which we can check the defects in software without actually executing it. Static testing is done to avoid errors in the early stage of the development as it is easier to find failure in the early stages.
- (14) **Component Testing :** Component testing is also a type of software testing in which testing is performed on each component separately without integrating with other parts.

(15) **Grey Box Testing** : Grey Box Testing defined as a combination of both white box and black-box testing. Grey Box testing is a testing technique which performed with limited information about the internal functionality of the system.

9. *What is Structural Testing?*

Structural testing is also known as glass box testing or white box testing. It is an approach where the tests are derived from the knowledge of the software's structure or internal implementation.

The other names of structural testing include clear box testing, open box testing, logic driven testing or path driven testing.

Structural Testing Techniques :

(1) **Statement Coverage** : This technique is aimed at exercising all programming statements with minimal tests.

$$\text{Statement Testing} = (\text{Number of Statements Exercised} / \text{Total Number of Statements}) \times 100\%$$

(2) **Branch Coverage** : This technique is running a series of tests to ensure that all branches are tested at least once.

$$\text{Branch Testing} = (\text{Number of decisions outcomes tested} / \text{Total Number of decision Outcomes}) \times 100\%$$

(3) **Path Coverage** : This technique corresponds to testing all possible paths which means that each statement and branch is covered.

$$\text{Path Coverage} = (\text{Number paths exercised} / \text{Total Number of paths in the program}) \times 100\%$$

Advantages of Structural Testing :

- (1) Forces test developer to reason carefully about implementation.
- (2) Reveals errors in "hidden" code
- (3) Spots the Dead Code or other issues with respect to best programming practices.

Disadvantages of Structural Box Testing :

- (1) Expensive as one has to spend both time and money to perform white box testing.
- (2) Every possibility that few lines of code is missed accidentally.

- (3) In-depth knowledge about the programming language is necessary to perform white box testing.

10. Explain Error oriented/ Error handling Testing?

Error handling testing is a type of software testing that is performed to check whether the system is capable of or able to handle the errors that may happen in future.

This type of testing is basically performed with the help of both developers and the testers. Error handling testing not only focuses on the determination of error but also focuses on the exception handling.

Objective of Error Handling Testing : The objective of error handling testing is :

- (1) To check the system ability to handle errors.
- (2) To check the system highest soak point.
- (3) To make sure errors can be handles properly by the system in the future.
- (4) To make system capable of exception handling also.

Steps Involved in the Error Handling Testing : Following are the steps involved in the error handling testing :

- (1) **Test Environment Set up :** Test environment is set according to the software testing technique so that the testing process can run smoothly. This step includes planning for the testing. System which is going to be tested is made sure have less significant data as there might be crash problem in the system during testing.
- (2) **Test Case Generation :** In this software testing test case generation is nothing but making different test cases which may cause error. Suppose software operates on fractions then setting the denominator of the fractions as zero. Test case generation is associated with the developing team as without knowing the internal code, test cases can't be designed.
- (3) **Test Case Execution :** After the test case generation, real testing process begins. This is the most prominent part of the testing process. It includes the running the program over the test case generated.

- (4) **Result and Analysis :** After the execution of the test case, its result is analyzed. It includes the checking of the inconsistency in the expected output for the generated test case. There might be a chance of the program going into an infinite loop which may lead up to software failure.
- (5) **Re-test :** If the testing is failed then after the analysis once more all the above steps are performed to test the system. It also includes the testing of the system under new test cases generated recently.

Advantages of Error Handling Testing :

- (1) It helps in construction of an error handling powered software.
- (2) It makes the software ready for all circumstances.
- (3) It develops the exception handling technique in the software.
- (4) It helps in maintenance of the software.

Disadvantages of the Error Handling Testing :

- (1) It is costly as both the developing and testing team is involved.
- (2) It takes lot of time to perform the testing operations.

11. *What is Static and Dynamic Testing? What is the difference between them?*

Static Testing

Static testing is testing, which checks the application without executing the code. It is a verification process. Some of the essential activities are done under static testing such as business requirement review, design review, code walkthroughs, and the test documentation review.

Static testing is performed in the white box testing phase, where the programmer checks every line of the code before handing over to the Test Engineer.

Static testing can be done manually or with the help of tools to improve the quality of the application by finding the error at the early stage of development; that why it is also called the verification process.

The documents review, high and low-level design review, code walkthrough take place in the verification process.

Dynamic Testing

Dynamic testing is testing, which is done when the code is executed at the run time environment. It is a validation process where functional testing [unit, integration, and system testing] and non-functional testing [user acceptance testing] are performed.

We will perform the dynamic testing to check whether the application or software is working fine during and after the installation of the application without any error.

Static Testing	Dynamic Testing
To check the code or the application without executing the code.	To check the code/application by executing the code.
Static testing includes activities like code Review, Walkthrough, etc	Dynamic testing includes activities like functional and non-functional testing such as UT (usability testing), IT (integration testing), ST (System testing) & UAT (user acceptance testing).
Static testing is a Verification Process.	Dynamic testing is a Validation Process.
Static testing is used to prevent defects.	Dynamic testing is used to find and fix the defects.
Static testing is a more cost-effective process.	Dynamic testing is a less cost-effective process.
This type of testing can be performed before the compilation of code.	Dynamic testing can be done only after the executable are prepared.
Under static testing, we can perform the statement coverage testing and structural testing.	Equivalence Partitioning and Boundary Value Analysis technique are performed under dynamic testing.
It involves the checklist and process which has been followed by the test engineer.	This type of testing required the test case for the execution of the code.

12. What are the different testing tools?

Software testing tools are required for the betterment of the application or software. That's why we have so many tools available in the market where some are open-source and paid tools.

The significant difference between open-source and the paid tool is that the open-source tools have limited