What is functional testing, and what is its purpose?

Functional testing is primarily used to verify that a piece of software is providing the same output as required by the end-user or business4. It ensures that the system behaves according to the specified functional requirements and meets the business needs

The purpose of functional testing is to test each function of the software application and check whether each output is satisfying the functional requirements.

Explain the difference between functional testing and non-functional testing.

Functional testing is a type of software testing in which used to check whether the application satisfies the requirements properly or not.
Functional testing is mostly focused on the outcome of the application to the given input

Non functional testing is a type of software testing that is used for non functional requirements of the application i.e the behavior of the application, when there is more number of inputs coming to the application than expected then how the application reacts etc.

What are the key steps involved in functional testing?

1. Identify the test inputs
2. Compute the expected outcomes with the selected test inputs
3. Execute test cases
4. Compare the actual output and the expected output

What is Test cases?Explain About them

Test case is used for determining whether the actions performed on the application satisfies software requirements and functions correctly.

What is Test Scenarios?Explain About them

Test scenarios are a set of manual or automated test cases that help determine the positive and negative characteristics of a project. A single scenario can cover more than one test cases
Eg: Complete registration form

What is negative testing?Explain about them and give example

Negative testing is a form of software testing that examines how an application handles unexpected input data and test scenarios. In this testing we give application incorrect inputs to check how the application handles the situation
Eg: if a user tries to type number in text input field then the application should give error message that “Incorrect data entered, please enter correct data”

What is difference between Smoke and Sanity Testing?

Smoke testing is a process where the software is tested for the stability of the application. This verifies whether the important features of the application are working and is the application ready for testing

Sanity testing is used to validate the changes made to one or few specific parts of the application in order to make sure that the software is still functioning as expected after a small change or a bug fix.

Smoke testing checks the critical functionalities of the system, while sanity testing checks the new functionality like bug fixes.

What is difference between Smoke and Regression Testing?

Smoke testing is a process where the software is tested for the stability of the application. This verifies whether the important features of the application are working and is the application ready for testing

Regression Testing is the process of testing the modified parts of the code and the parts that might get affected due to the modifications to ensure that no new errors have been introduced in the software after the modifications have been made.

What is equivalence partitioning, and how is it useful in functional testing?

In equivalence portioning , it divides input domain into classes of data and with the help of these these classes of data , test cases can be derived.
Eg: Let us consider an example of any college admission process. There is a college that gives admissions to students based upon their percentage.
Consider percentage field that will accept percentage only between 50 to 90 %, more and even less than not be accepted, and application will redirect user to an error page. If percentage entered by user is less than 50 %or more than 90 %, that equivalence partitioning method will show an invalid percentage. If percentage entered is between 50 to 90 %, then equivalence partitioning method will show valid percentage.

What is the difference between positive testing and negative testing?

Positive testing is a software testing that is used to verify and validate an application to check whether it is working as expected.

What is black box testing, and what are the various techniques?

Black Box Testing is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure

Techniques of black box testing

- Equivalence Class Testing: It is used to minimize the number of possible test cases to an optimum level while maintains reasonable test coverage.
- Boundary Value Testing: Boundary value testing is focused on the values at boundaries. This technique determines whether a certain range of values are acceptable by the system or not. It is very useful in reducing the number of test cases. It is most suitable for the systems where an input is within certain ranges.
- Decision Table Testing: A decision table puts causes and their effects in a matrix. There is a unique combination in each column.

What is white box testing and its various techniques?

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.

Techniques of white box

- Statement Coverage
- Branch Coverage
- Condition Coverage
- Loop Testing

Eg: unit testing , integraton testing

What is difference between White Box and Black Box testing?

What is Explority testing?

Exploratory testing is a software testing technique where testers simultaneously learn, design, and execute tests without predefined scripts.

what is difference between regression testing and retesting

Regression Testing is the process of testing the modified parts of the code and the parts that might get affected due to the modifications to ensure that no new errors have been introduced in the software after the modifications have been made

Retesting is a process to check specific test cases that are found with bug/s in the final execution. Generally, testers find these bugs while testing the software application and assign it to the developers to fix it. Then the developers fix the bug/s and assign it back to the testers for verification. This continuous process is called Retesting.

What is Unit Testing?

Unit testing is a type of software testing that focuses on individual units or components of a software system. The purpose of unit testing is to validate that each unit of the software works as intended and meets the requirements.

What is Integration Testing?

Integration testing is a level of software testing where individual units are combined and tested to verify if they are working as they intend to when integrated

What is the difference between manual testing and automated testing in functional testing?

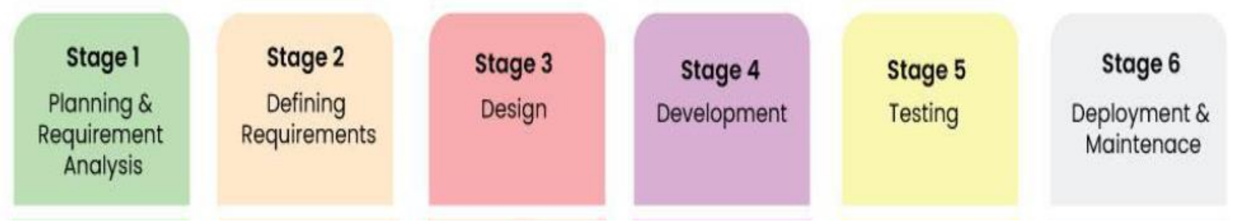
Functional testing is a manual process that involves human intervention to test the application's functionality against the functional requirements, while automated testing involves the use of automated tools and scripts to test the application's functionality.

What are the types of testing?

Types of Software Testing - GeeksforGeeks

What is SDLC?

Software Development Life Cycle (SDLC) - GeeksforGeeks



Planning and Requirement Analysis: in this stage, the basic project is designed with all the available information.

Defining Requirements: This is a sort of document that specifies all those things that need to be defined and created during the entire project cycle.

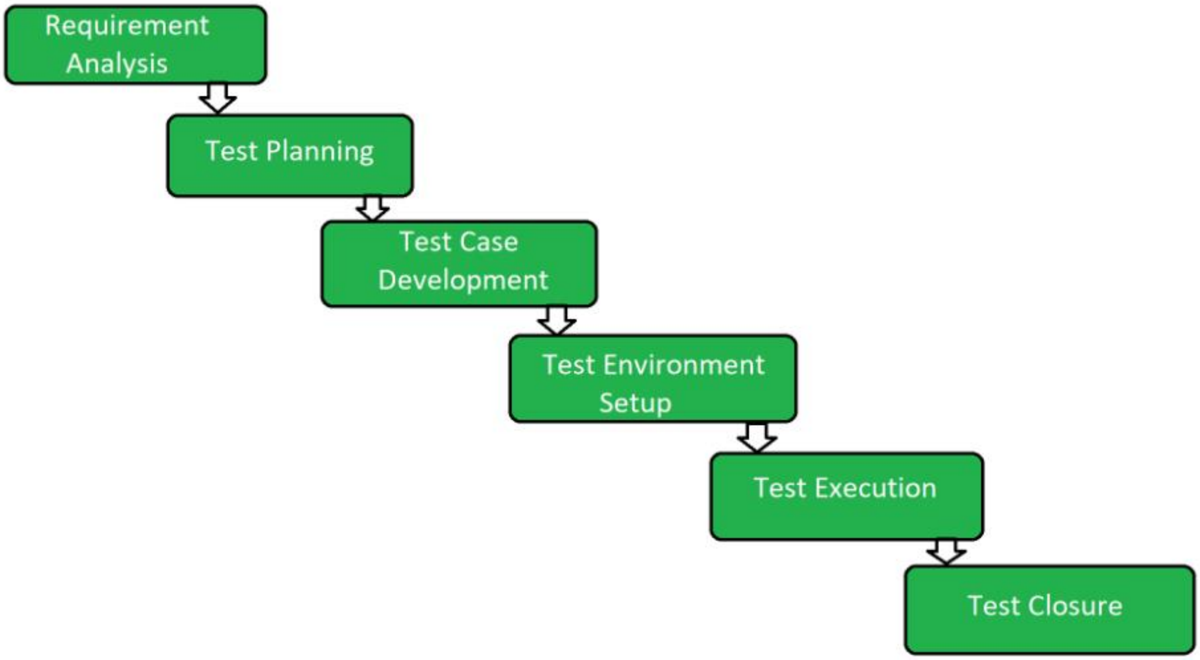
Designing Architecture
SRS is a reference for software designers to come up with the best architecture for the software.

Developing Product
At this stage, the fundamental development of the product starts. For this, developers use a specific programming code as per the design in the DDS. Hence, it is important for the coders to follow the protocols set by the association.

Testing and Integration
After the development of the product, testing of the software is necessary to ensure its smooth execution.

Deployment and Maintenance of Products
After detailed testing, the conclusive product is released in phases as per the organization's strategy. Then it is tested in a real industrial environment. It is important to ensure its smooth performance. If it performs well, the organization sends out the product as a whole. After retrieving beneficial feedback, the company releases it as it is or with auxiliary improvements to make it further helpful for the customers. However, this alone is not enough

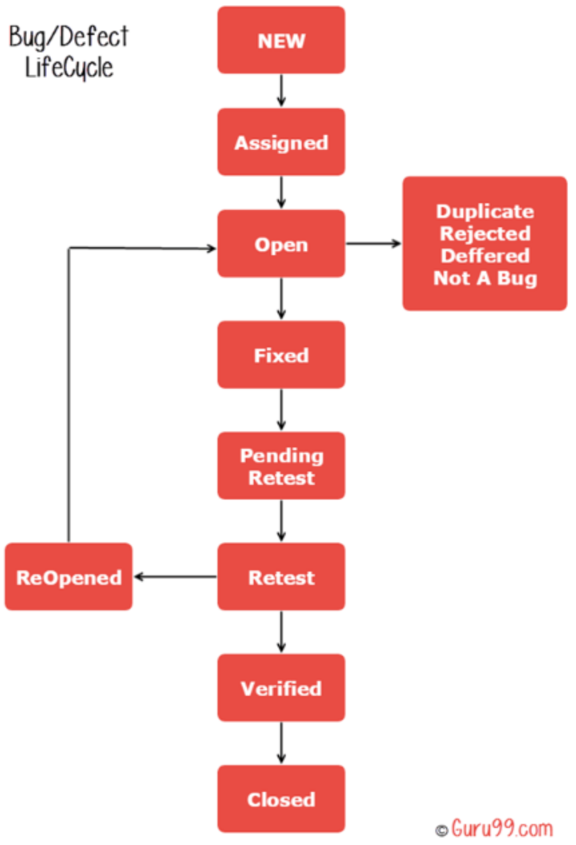
What is STLC?



What is Defect/Bug Life cycle?

Bug Life Cycle in Software Development - GeeksforGeeks

The purpose of Defect life cycle is to easily coordinate and communicate current status of defect which changes to various assignees and make the defect fixing process systematic and efficient.



How do you perform boundary value analysis in functional testing?

Boundary Value Analysis is based on testing the boundary values of valid and invalid partitions. The behavior at the edge of the equivalence partition is more likely to be incorrect than the behavior within the partition, so boundaries are an area where testing is likely to yield defects.

What is the importance of requirement traceability in functional testing?

Requirement traceability in functional testing is like a roadmap that helps us track the journey of a requirement from its origin to its final implementation. Here's why it's important:

1. **Ensures Coverage**: It helps to ensure that all requirements are covered by tests. This means we don't miss testing any feature.
2. **Identifies Impact**: If a requirement changes, traceability helps identify what tests and other requirements might be impacted by this change.
3. **Improves Communication**: It provides a clear picture of the project status and progress to all stakeholders, improving communication and understanding.
4. **Reduces Risks**: By ensuring every requirement is tested and any change is properly managed, it reduces the risk of issues slipping into the final product.

In simple words, requirement traceability is like a detective's notebook in a mystery novel. It keeps track of all clues (requirements), how they connect to each other, and ensures no clue is left uninvestigated (untested). It also helps to understand the impact if a clue changes or a new clue is added. It’s a crucial tool for delivering a high-quality product.

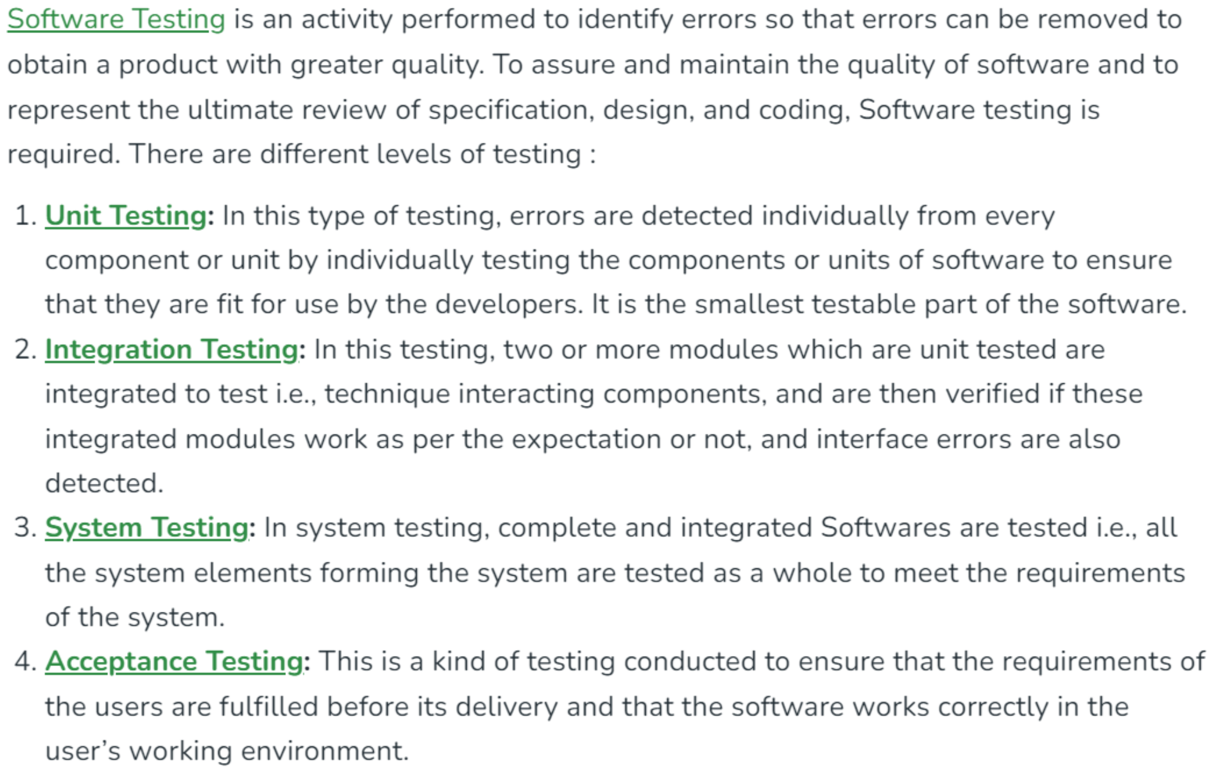
Requirement Traceability is mapping of requirements to test cases. There are two main traceability: Forward and Backward. It is important to know whether all the requirements mentioned in the Requirement Document have a corresponding test case or not. At the same time, it is important to know which requirement a particular test case has been written for. These are important if there are any changes in the requirement, we should know which test cases need to be re-written or modified

Explain the concept of test coverage in functional testing.

Test coverage is a measure of how much of the code or functionality of a system is being tested by a particular set of tests. For example, if a set of tests only covers 50% of the code, then it has 50% coverage.

What is regression testing, and why is it important in functional testing?

What are the different levels of testing in functional testing?



Software Testing is an activity performed to identify errors so that errors can be removed to obtain a product with greater quality. To assure and maintain the quality of software and to represent the ultimate review of specification, design, and coding, Software testing is required. There are different levels of testing :

1. **Unit Testing:** In this type of testing, errors are detected individually from every component or unit by individually testing the components or units of software to ensure that they are fit for use by the developers. It is the smallest testable part of the software.
2. **Integration Testing:** In this testing, two or more modules which are unit tested are integrated to test i.e., technique interacting components, and are then verified if these integrated modules work as per the expectation or not, and interface errors are also detected.
3. **System Testing:** In system testing, complete and integrated Softwares are tested i.e., all the system elements forming the system are tested as a whole to meet the requirements of the system.
4. **Acceptance Testing:** This is a kind of testing conducted to ensure that the requirements of the users are fulfilled before its delivery and that the software works correctly in the user's working environment.

Explain the concept of smoke testing in functional testing.

Explain the concept of accessibility testing in functional testing.

This type of testing involves evaluating the functionality of a software deliverable to ensure widespread use by all. It includes testing the product, service, or environment with assistive technologies, such as screen readers, keyboard-only navigation, and through actual people with disabilities.

What is performance testing in functional testing?

Performance testing and functional testing are two different types of software testing. Performance testing validates that application software can handle real time scenarios and address issues, if any, to deliver a robust and efficient product to the end user

What is the purpose of security testing in functional testing?

Security testing of any system is focused on finding all possible loopholes and weaknesses of the system that might result in the loss of information or repute of the organization.

Explain the concept of user acceptance testing (UAT) in functional testing.

What is the difference between verification and validation in functional testing?

User Acceptance Testing is a critical phase in software development where technically proficient developers and functional testers validate the software against functional specifications. They interpret these specifications based on their understanding, developing and testing the software accordingly.

Explain the concept of defect life cycle in functional testing.

What is the purpose of a test plan in functional testing?

A test plan is a document that consists of all future testing related activities

What is the difference between a test scenario and a test case in functional testing?

How to report a bug?

A well-written bug report is essential in software testing to facilitate effective communication between testers and developers, leading to improved program quality and user satisfaction.

What is Priority?Types of Priority

Priority is an order in which we decide which bug should be fixed .

It can be **urgent, high, medium, and low**.

High: it is a major impact on the customer application, and it has to be fixed first.

Medium: In this, the problem should be fixed before the release of the current version in development.

Low: The flow should be fixed if there is time, but it can be deferred with the next release.

What is Severity?Types of Severity

The impact of the bug on the application is known as severity.

It can be a **blocker, critical, major, and minor** for the bug.

Blocker: if the severity of a bug is a blocker, which means we cannot proceed to the next module, and unnecessarily test engineer sits ideal.

Critical: if it is critical, that means the main functionality is not working, and the test engineer cannot continue testing.

Major: if it is major, which means that the supporting components and modules are not working fine, but test engineer can continue the testing.

Minor: if the severity of a bug is major, which means that all the U.I problems are not working fine, but testing can be processed without interruption.

In the context of software testing, the terms "bug", "defect", and "error" are often used interchangeably, but they have slightly different meanings:

- **Bug**: A bug is a term used when the software or application is not working as per the requirement. When there's a logical error in the code, it causes the program to break, which results in a bug[1,2,4].

- **Defect**: A defect refers to a situation when the application is not working as per the requirement and the actual and expected result of the application or software are not in sync with each other[1,2,4]. It can arise when a developer makes major or minor mistakes during the development phase[1].

- **Error**: An error occurs when the Development team or the developer fails to understand a requirement definition and hence that misunderstanding gets translated into buggy code[1,2,4]. Errors are generated due to wrong logic, syntax, or loop that can impact the end-user experience[1].

It's important to note that these terms are often used based on the phases of the Software Development Life Cycle (SDLC) and may vary across different teams or organizations[2]. Understanding these terms can help in better communication and collaboration within a software development team.

- **Verification**: This is like asking, "Are we building the product right?"[1,2]. It's about checking whether the software is being developed correctly and according to the specified requirements[1,2,3,4]. It involves activities like reviewing documents, designs, codes, and programs[1]. It's also known as static testing because it does not involve executing the code[1,2].
- **Validation**: This is like asking, "Are we building the right product?"[1,2]. It's about checking whether the final product meets the user's needs and expectations[1,2,3,4]. It involves testing the actual product, which includes executing the code[1,2]. This is why it's also known as dynamic testing