**https://www.glassdoor.co.in/Interview/selenium-interview-questions-SRCH\_KO0,8\_IP2.htm**

**What is Software Testing?**

 A process of analyzing a software item to detect the differences between existing and required conditions (i.e., defects) and to evaluate the features of the software item

**What are Quality Assurance and Quality Control?**

**Quality Assurance:**

Quality Assurance involves in process-oriented activities.

It ensures the prevention of defects in the process used to make Software Application.

**Quality Control**

Quality Control involves in product-oriented activities.

It executes the program or code to identify the defects in the Software Application

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

***Verification*** is the process of checking that the software meets the specification.

**Validation** is the process of checking whether the specification captures the customer’s needs.

 Activities involved in this is Testing the software application.

|  |  |
| --- | --- |
| * It does ***not*** involve executing the code | * It always involves executing the code |
| * Verification uses methods like reviews, walkthroughs, inspections and desk- checking etc. | * It uses methods like [Black Box Testing](https://www.guru99.com/black-box-testing.html),[White Box Testing](https://www.guru99.com/white-box-testing.html) and non-functional testing |
|  |  |
| Are we building the system right? | Are we building the right system? |

**5. What is Static Testing?**

**Static** testing involves in without the execution of code.

Static Testing involves in reviewing the documents

**6. What is Dynamic Testing?**

Dynamic testing involves in the execution of code.

**What is Black Box Testing?**

 evaluate the functionality of the software without looking at the internal code structure.

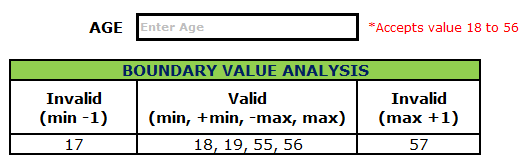
This can be applied to every level of software testing such as Unit, Integration, System and Acceptance Testing.

TECHNIQUES:

**Boundary Value Analysis**

Boundary value analysis (BVA) is based on testing the boundary values of valid and invalid partitions.

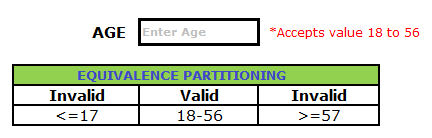
18 – 56



**Equivalence Class Partition**

 Hence selecting one input from each group to design the test cases.

Assume, we have to test a field which accepts Age 18 – 56

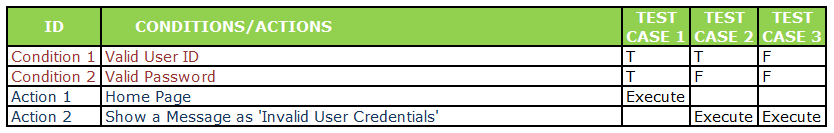


**91. What is Decision Table testing?** Cause-Effect Table

Decision table testing is a testing technique used to test system behavior for different input combinations.

From the case 2 and case 3, we could identify that if one of the condition failed then the system will display an error message as Invalid User Credentials.

So I am eliminating one of the test case from case 2 and case 3 and concluding with the below tabular column.



**What is State Transition?**

We have to design the table according to ,How the application is changing the state from one stage to next.

accordingly we can write testcase on it.  
example : U can take an eg of ATM machine,  
If u enter wrong password--Applicaton will ask u to re enter password  
Again if u enter wrong password --second time application will ask u to re enter password  
and third time if u enter wrong password --finally ur card will get blocked  
  
so there will be 3 diff phases of application..is nothing but state transition.

**93. What is an entry criteria?**

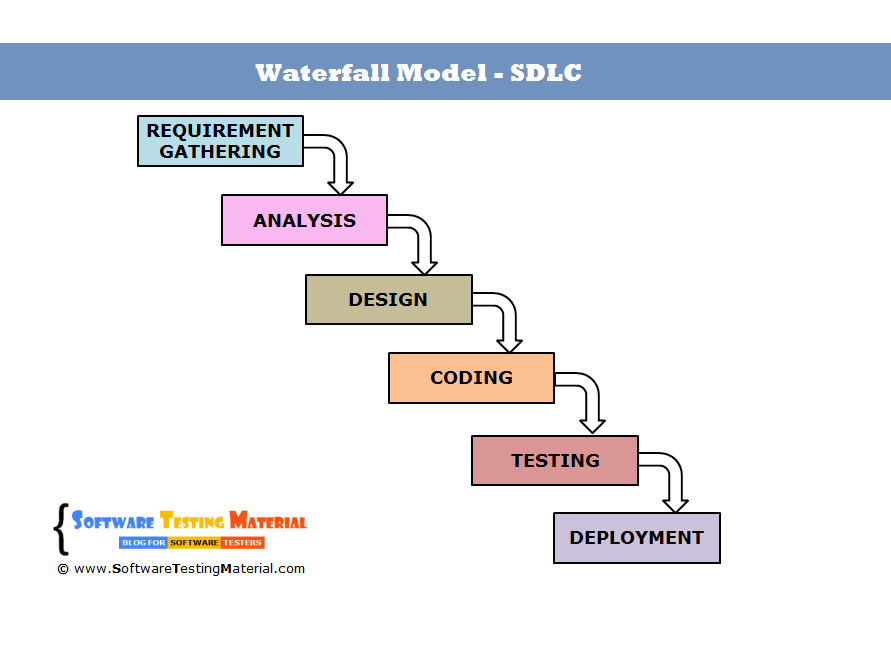
**94. What is an exit criteria?**

**95. What is SDLC?**

Software Development Life Cycle (SDLC) aims to produce a high-quality system that meets or exceeds customer expectations, works effectively and efficiently in the current and planned information technology infrastructure, and is inexpensive to maintain and cost-effective to enhance.

**96. What are the different available models of SDLC?**

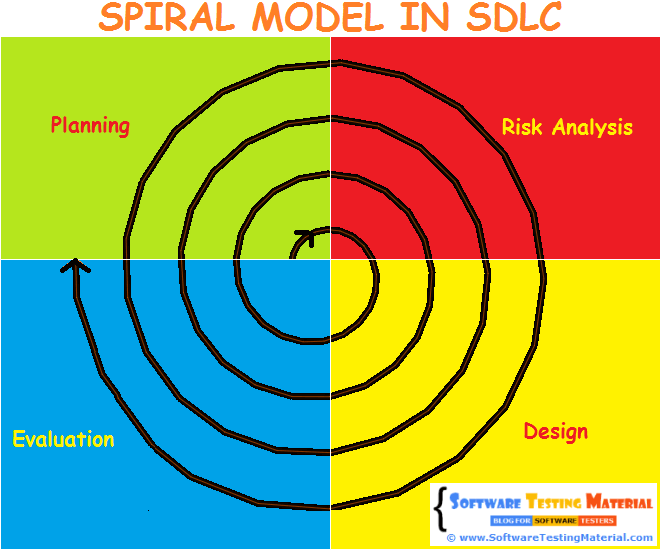
1. [Waterfall](https://www.softwaretestingmaterial.com/waterfall-model-in-sdlc/)
2. [Spiral](https://www.softwaretestingmaterial.com/spiral-model-in-sdlc/)
3. [V Model](https://www.softwaretestingmaterial.com/v-model-in-sdlc/)
4. Prototype
5. [Agile](https://www.softwaretestingmaterial.com/agile-scrum-methodology/)



Sequential Design Process

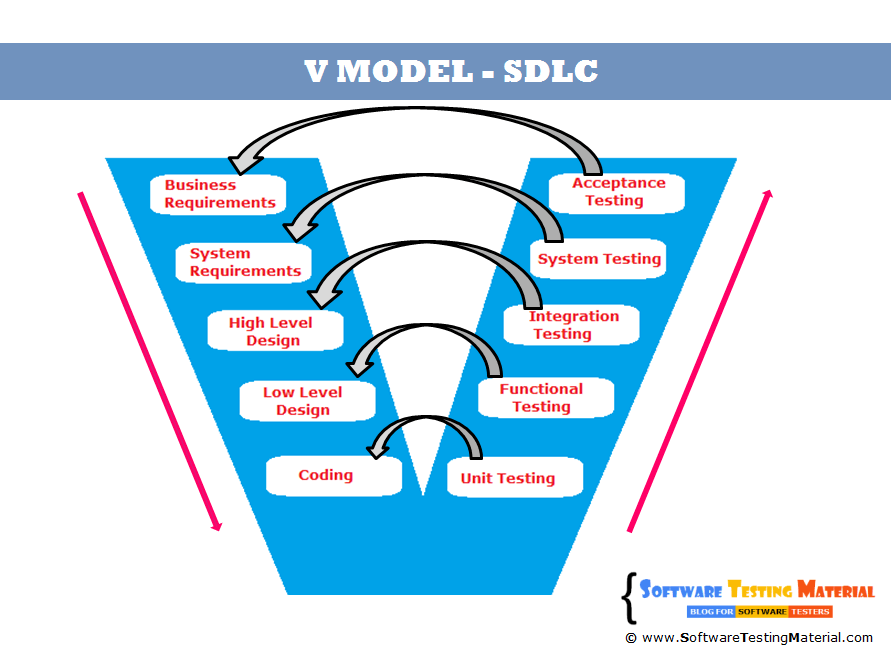
* Backtracking is not possible i.e., we cannot go back and change requirements once the design stage is reached.

## ****Spiral Model****



It is a combination of both Prototype development process and Linear development process ([waterfall model](https://www.softwaretestingmaterial.com/waterfall-model-in-SDLC)) Each spiral can be termed as a loop and each loop is a separate development process in a spiral model. The four activities (Planning, Risk analysis, engineering and evaluation) form the intermediary phases of a spiral model and is repeated again for each loop.

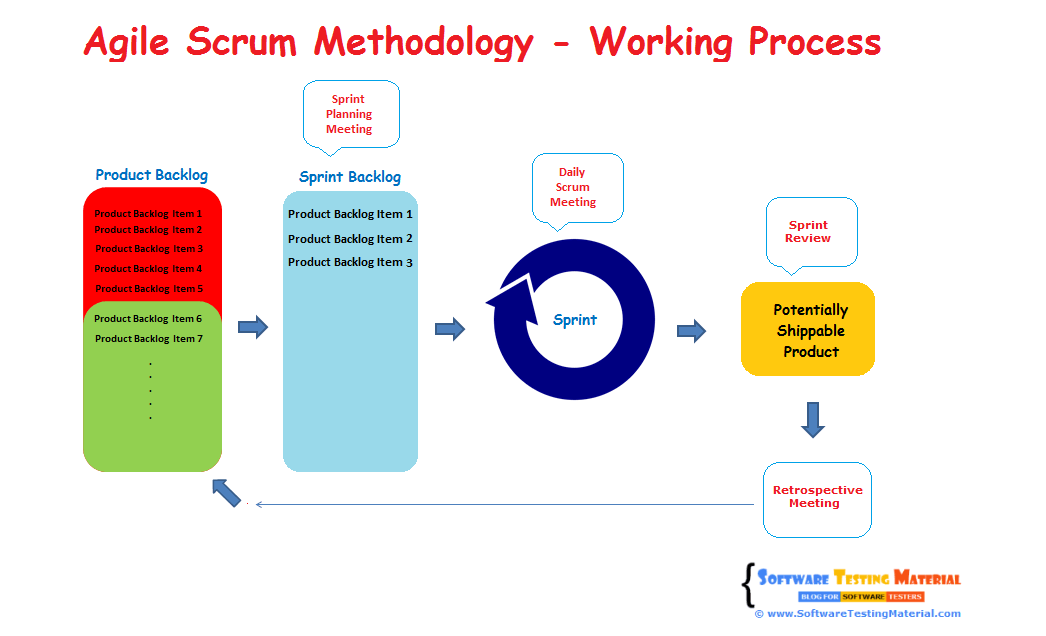
* Risk analysis is important phase so requires expert people.
* Is not beneficial for smaller projects.
* Spiral may go infinitely.
* Documentation is more as it has intermediate phases.
* It is costly for smaller projects.



V-model is also known as **Verification and Validation (V&V) model**.

1) Agile mandates customer interaction on a regular basis, V doesn't  
2) Task(requirements) are never measured to their weight in V, yes in Agile, hence we see Task Break down in smaller chunks  
3) In Agile we can change the direction on will, meaning back logs can be postponed or preponed, V doesn't have a look back or ahead and change direction concept.

**Agile Scrum Methodology**



Agile Scrum Methodology is a combination of both Incremental and Iterative model for managing product development.

Each Sprint has a specified time line (2 weeks to 1 month).

Sprint Grooming is done where our product owners explain about their new stories(Product Backlog Items).

Then,poker will happen where we do some estimation for the entire sprint.

In sprint planning meeting,my scrum master allocate the task to us for the particular sprint.

**Daily Scrum Meeting:** (Daily Stand-up) will occur where we inform what I did yesterday, what I am going to do today and what obstacles are impeding in my progress to scrum team.

Test analysis->

**Sprint Review Meeting** is done where I provide demonstration about what I did for that sprint.if he gave additional recuirement we will add those things in next sprint.

Retrospective meeting is done where we discuss what went well and what can be improved.

**What is Grey Box Testing?**

Grey box is the combination of both White Box and Black Box Testing.

**37. What are the Structure-based (white-box) testing techniques?**

They are commonly called 'glass-box'

evaluate the functionality of the software with looking at the internal code structure.

**10. What is Positive and Negative Testing?**

**Positive Testing:** It is to determine what system supposed to do. It helps to check whether the application is justifying the requirements or not.

**Negative Testing:** It is to determine what system not supposed to do. It helps to find the defects from the software.

Test plan->test analysis->implementation->execution->report

**11. What is Test Strategy?**

**It is normally derived from the Business Requirement Specification (BRS).** Test Strategy is a high-level document (static document) and usually developed by project manager which captures the approach on how we go about testing the product and achieve the goals.

It is high level static document which is **derived from the Business Requirement Specification (BRS). It is developed by project manager which captures how we will test the product to meet the client’s need**

**12. What is Test Plan and contents available in a Test Plan?**

Test plan document is a document which contains the plan for all the testing activities to be done to deliver a quality product.

Test Plan document is derived from the Product Description, SRS It is usually prepared by the Test Lead or Test Manager.

1. Test plan identifier
2. References
3. Introduction
4. Test items (functions)
5. Software risk issues
6. Features to be tested
7. Features not to be tested
8. Approach
9. Items pass/fail criteria
10. Suspension criteria and resolution requirements
11. Test deliverables
12. Remaining test tasks
13. Environmental needs
14. Staff and training needs
15. Responsibility
16. Schedule
17. Plan risks and contingencies
18. Approvals
19. Glossaries

**13. What is Test Suite?**

Test Suite is a collection of test cases. The test cases which are intended to test an application.

**15. What is Test Case?**

Test cases are the set of positive and negative executable steps of a test scenario which has a set of pre-conditions, test data, expected result, post-conditions and actual results.

**14. What is Test Scenario?**

Test Scenario gives the idea of what we have to test. Test Scenario is like a high-level test case.

**16. What is Test Bed?**

An execution environment configured for testing. Test bed consists of hardware, software, network configuration, an application under test, other related software.

* test bed---->required test cases which are ready for execution.

**17. What is Test Environment?**

Test environment--->required hardware and software components for the test execution...

Example:

* Application Type: Web Application
* OS: Windows
* Web Server: IIS
* Web Page Design: Dot Net
* Client Side Validation: JavaScript
* Server Side Scripting: ASP Dot Net
* Database: MS SQL Server
* Browser: IE/FireFox/Chrome

Test harness : A programe or test tool used to execute a test. Also known as test driver.

test harness:it indicates"readyness to start test execution"

* test harness=test environment+test bed

**18. What is Test Data?**

Test data is the data that is used by the testers to run the test cases.

For example, To test a basic login functionality having a user id, password fields. We need to enter some data in the user id and password fields. So we need to collect some test data.

**20. What is Test Closure?**

Test Closure is the note prepared before test team formally completes the testing process. This note contains the total no. of test cases, total no. of test cases executed, total no. of defects found, total no. of defects fixed, total no. of bugs not fixed, total no of bugs rejected etc.

**21. List out Test Deliverables?**

1. Test Strategy
2. Test Plan
3. Effort Estimation Report
4. Test Scenarios
5. Test Cases/Scripts
6. Test Data
7. Requirement Traceability Matrix (RTM)
8. Defect Report/Bug Report
9. Test Execution Report
10. Graphs and Metrics
11. Test summary report
12. Test incident report
13. Test closure report
14. Release Note
15. Installation/configuration guide
16. User guide
17. Test status report
18. Weekly status report (Project manager to client)

**22. What is Unit Testing?**

Unit Testing is also called as Module Testing or Component Testing.

It is done to check whether the individual unit or module of the source code is working properly.

It is done by the developers in developer’s environment.

**23. What is Integration Testing?**

Integration Testing is the process of testing the interface between the two software units.

 Integration testing is done by three ways. Big Bang Approach, Top Down Approach, Bottom-Up Approach

**25. What is Big Bang Approach?**

Combining all the modules once and verifying the functionality after completion of individual module testing.

Top down and bottom up are carried out by using dummy modules known as Stubs and Drivers.

Stubs and Drivers are used to stand-in for missing components to simulate data communication between modules.

**26. What is Top-Down Approach?**

Testing takes place from top to bottom.

High-level modules are tested first and then low-level modules and

finally integrating the low-level modules to a high level to ensure the system is working as intended.

Stubs are used as a temporary module if a module is not ready for integration testing.

**27. What is Bottom-Up Approach?**

It is a reciprocate of the Top-Down Approach. Testing takes place from bottom to up.

Lowest level modules are tested first and then high-level modules and

finally integrating the high-level modules to a low level to ensure the system is working as intended. 

Drivers are used as a temporary module for integration testing.

**24. What is System Testing?** End to End testing

Testing the fully integrated application to ensure that the application works as intended or not.

**29. What is Functional Testing?**

In simple words, what the system actually does is functional testing.

 Testing all the functionalities by providing appropriate input to verify whether the actual output is matching the expected output or not.

It falls within the scope of black box testing.

**30. What is Non-Functional Testing?**

In simple words, how well the system performs is non-functionality testing.

Main focus is to improve the user experience on how fast the system responds to a request.

 performance, load, stress, scalability, security, compatibility etc.,

**31. What is Acceptance Testing?**

It is also known as pre-production testing. It is perfomed to determine whether or not the software system has met the requirement specifications.

Types of acceptance testing are Alpha, Beta & Gamma.

**32. What is Alpha Testing?**

Alpha testing is done by the in-house developers (who developed the software) and testers.

**33. What is Beta Testing?**

Beta testing is done by a limited number of end users before delivery. Usually, it is done in the client place.

**34. What is Gamma Testing?**

Gamma testing is done when the software is ready for release with specified requirements. It is done at the client place. It is done directly by skipping all the in-house testing activities.

**5. What is Smoke Testing?**

Smoke Testing is done to make sure if the build we received from the development team is testable or not.

It is also called as “Day 0” check. It is done at the “build level”.

**36. What is Sanity Testing?**

Sanity Testing is done during the release phase to check for the main functionalities of the application without going deeper.

It is also called as a subset of Regression testing. It is done at the “release level”

However, a sanity test is generally without test scripts or test cases.

|  |  |
| --- | --- |
| **Regression Testing** | **Re-testing** |
| * Regression Testing is carried out to confirm whether a recent program or code change has not adversely affected existing features | * Re-testing is carried out to confirm the test cases that failed in the final execution are passing after the defects are fixed |

|  |  |
| --- | --- |
| * You can do automation for regression testing, [Manual Testing](https://www.guru99.com/manual-testing.html) could be expensive and time consuming | * You cannot automate the test cases for Retesting |
| * Regression testing is known as a generic testing | * Re-testing is a planned testing |

Usually, we do regression testing in the following cases:

1. New functionalities are added to the application
2. Change Requirement (In organizations, we call it as CR)
3. Defect Fixing
4. Performance Issue Fix
5. Environment change (E.g., Updating the DB from MySQL to Oracle)

**39. What is GUI Testing?**

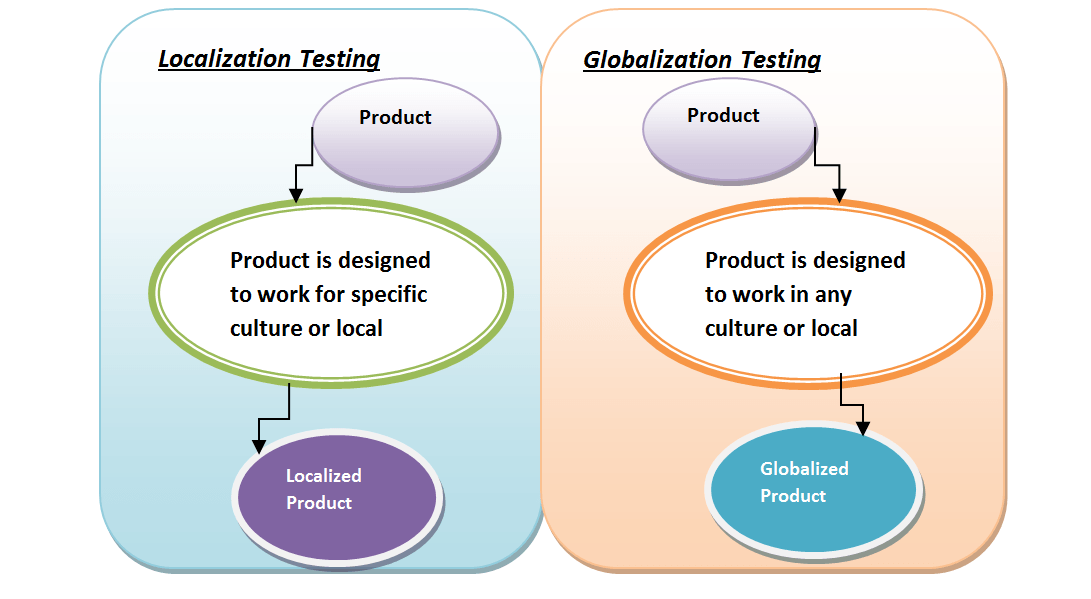
Graphical User Interface Testing is to test the interface between the application and the end user.

**40. What is Recovery Testing?**

Recovery testing is performed in order to determine how quickly the system can recover after the system crash or hardware failure.

It comes under the type of non-functional testing.

**41. What is Globalization Testing? and Internationalization Testing**Globalization is a process of designing a software application so that it can be adapted to various languages and regions without any changes.



**44. What is Installation Testing?**It is to check whether the application is successfully installed and it is working as expected after installation.

**47. What is Compatibility Testing?**to check whether the application is working as expected in a different combination of environmental components.

**48. What is Exploratory Testing?**Usually, this process will be carried out by domain experts. They perform testing just by exploring the functionalities of the application without having the knowledge of the requirements.

**49. What is Monkey Testing?**

Perform abnormal action on the application deliberately in order to verify the stability of the application.

**50. What is Usability Testing?**

 to check whether the end user can understand and operate the application easily or not.

**51. What is Security Testing?**

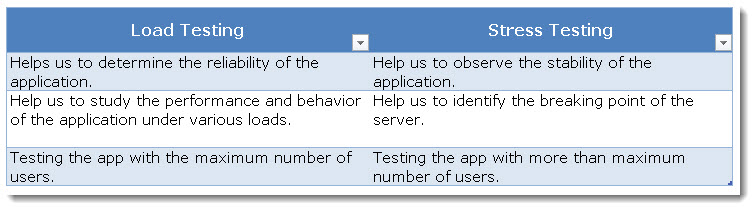
Security testing is a process to determine whether the system protects data and maintains functionality as intended.

**52. What is Soak Testing?**

Running a system at high load for a prolonged period of time to identify the performance problems is called Soak Testing.

**53. What is Performance Testing?**

This type of testing determines or validates the speed, scalability, and/or stability characteristics of the system or application under test. Performance is concerned with achieving response times, and resource-utilization levels that meet the performance objectives for the project or product.



**55. What is Volume Testing?**

It is toverify that the system/application can handle a large amount of data

<https://www.careerride.com/database-testing-interview-questions-part-2.aspx>

**scalability testing :**

**It is a type of non-functional testing.**

**Scalability testing lets you determine how your application scales with increasing workload.**

**58. What is Concurrency Testing?**

Concurrency testing means accessing the application at the same time by multiple users to ensure the stability of the system. This is mainly used to identify deadlock issues.

**59. What is Fuzz Testing?**

Fuzz testing is used to identify coding errors and security loopholes in an application.

**60. What is Adhoc Testing?** Testers randomly test the application without any test cases or any business requirement document.

knowledge of testers in the application under test is very high

**61. What is Interface Testing?**

Interface testing is performed to evaluate whether two intended modules pass data and communicate correctly to one another.

**62. What is Reliability Testing?**Perform testing on the application continuously for long period of time in order to verify the stability of the application

**63. What is Bucket Testing? A/B Testing Split Testing**

Bucket testing is a method to compare two versions of an application against each other to determine which one performs better.

**66. What are the principles of Software Testing?**

1. Testing shows presence of defects
2. Exhaustive testing is impossible  
   (Testing all the functionalities using all valid and invalid inputs and preconditions)
3. Early testing

(Defects detected in early phases of SDLC are less expensive to fix)

1. Defect clustering(a small module or functionality contains most of the bugs)
2. Pesticide Paradox(the process of repeating the same test cases, again and again, eventually, the same test cases will no longer find new bugs.

it is necessary to review the test cases regularly and add or update them to find more defects.)

REVIEW TOOL -RADAR

1. Testing is context depending
2. Absence of error fallacy

**74. What is a Defect?**

The variation between the actual results and expected results is known as a defect. If a developer finds an issue and corrects it by himself in the development phase then it’s called a defect.

**75. What is a Bug?**

If testers find any mismatch in the application/system in testing phase then they call it as Bug

**76. What is an Error?**

We can’t compile or run a program due to a coding mistake in a program.

**77. What is a Failure?**

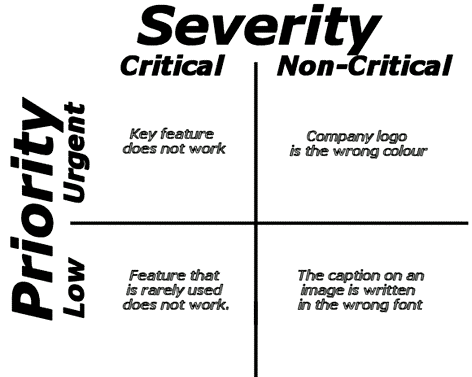
After release, if an end user finds an issue then that particular issue is called as a failure.

**78. What is Bug Severity?**  
Severity of a defect is related to how severe a bug is

**79. What is Bug Priority?**

Defect priority can be defined as how soon the defect should be fixed.

|  |  |
| --- | --- |
| * Priority is categorized into three types   + Low   + Medium   + High | * Severity are categorized into five types   + Critical   + Major   + Moderate   + Minor   + Cosmetic |



**What is an API?**

API is an acronym for **A**pplication **P**rogramming **I**nterface.

It enables communication and data exchange between two separate software systems.

**81. What is the difference between a Standalone application, Client-Server application and Web application?**

**Standalone application:**

Standalone applications follow one-tier architecture. Presentation, Business, and Database layer are in one system for a single user.

**Client-Server Application:**

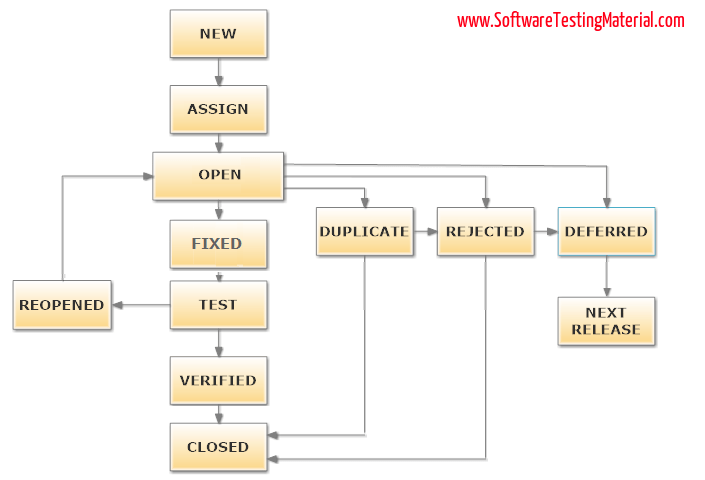
Client-server applications follow two-tier architecture. Presentation and Business layer are in a client system and Database layer on another server. It works majorly in Intranet.

**Web Application:**

Web server applications follow three-tier or n-tier architecture. The presentation layer is in a client system, a Business layer is in an application server and Database layer is in a Database server. It works both in Intranet and Internet.

**82. What is Bug Life Cycle? Defect life cycle**

**DEFECT TRACKING TOOL JIRA**



When a tester finds a new defect. In this state, the status of the defect posted by tester is “New”

Once the defect is assigned then the status of the bug changes to “Assigned”

**Open:** The development team starts analyzing and works on the defect fix

**Fixed:** When a developer makes the necessary code change and verifies the change, then the status of the bug will be changed as “Fixed” and the bug is passed to the testing team.

**Test:** If the status is “Test”, it means the defect is fixed and ready to do test whether it is fixed or not.

**Verified:** The tester re-tests the bug after it got fixed by the developer. If there is no bug detected in the software, then the bug is fixed and the status assigned is “verified.”

**Closed:** After verified the fix, if the bug is no longer exits then the status of bug will be assigned as “Closed.”

**Reopen:** If the defect remains same after the retest, then the tester posts the defect using defect retesting document and changes the status to “Reopen”. Again the bug goes through the life cycle to be fixed.

**Duplicate:** If the defect is repeated twice

**Deferred:** If it is expected to get fixed in the next release

**Rejected:**If the system is working according to specifications and bug is just due to some misinterpretation

**83. What is Bug Leakage?**

missed by the testing team) was found by the end user or customer then we call it as Bug Leakage.

**84. What is Bug Release?**

Releasing the software to the Production with the known bugs then we call it as Bug Release.

**85. What is Defect Age?**

Defect age can be defined as the time interval between date of defect detection and date of defect closure.

Defect Age = Date of defect closure – Date of defect detection

**86. What is Error Seeding?**

Error seeding is a process of adding known errors intendedly in a program to identify the rate of error detection

**87. What is Showstopper Defect?**

A showstopper defect is a defect which won’t allow a user to move further in the application. It’s almost like a crash.

Assume that login button is not working. Even though you have a valid username and valid password, you could not move further because the login button is not functioning.

**What is STLC?**

STLC (Software Testing Life Cycle) identifies what test activities to carry out and when to accomplish those test activities.

* [1.1 Test](https://www.softwaretestingmaterial.com/stlc-software-testing-life-cycle/" \l "Requirement_Analysis) plan(done by our project lead)
* [1.2 Test analysis and design:](https://www.softwaretestingmaterial.com/stlc-software-testing-life-cycle/" \l "Test_Planning) I gone through all the requirements from srs for my module.If I found any ambiguous I asked my lead.after that I do design of test cases.once I complete that design I send that one to my lead for review.Once review completed I do
* [1.4 Test Implementation (Environment Setup):](https://www.softwaretestingmaterial.com/stlc-software-testing-life-cycle/" \l "Test_Environment_Setup)where I made environmental setup for my software and also I write test script for my automation purpose.

Then I do smoke test for my module inorder to check whether give build is working fine or not and also to find any showstopper present or not.

* [1.5 Test Execution:](https://www.softwaretestingmaterial.com/stlc-software-testing-life-cycle/" \l "Test_Execution)After that I execute the cases and if any bugs I found then I added that in the JIRA tool
* [1.6 Defect tracking:](https://www.softwaretestingmaterial.com/stlc-software-testing-life-cycle/" \l "Test_Closure)

**98. What is RTM?** Traceability Matrix **Cross Reference Matrix**

is used to trace the requirements to the test cases that are needed to verify whether the requirements are fulfilled.

**Forward Traceability**: Mapping requirements to test cases is called Forward Traceability Matrix.

**Backward or Reverse Traceability:**Mapping [test cases](https://www.softwaretestingmaterial.com/test-case-template-with-explanation/) to requirements is called Backward Traceability Matrix.

**Bi-directional traceability (Forward + Backward):**Mapping requirements to [test cases](https://www.softwaretestingmaterial.com/test-case-template-with-explanation/)(forward traceability) and [test cases](https://www.softwaretestingmaterial.com/test-case-template-with-explanation/) to requirements (backward traceability) is called Bi-directional Traceability Matrix.

### ****Advantage of Requirements Traceability Matrix (RTM):****

1. 100% test coverage
2. It allows to identify the missing functionality easily
3. It allows to identify the test cases which needs to be updated in case of change in requirement
4. It is easy to track the overall test execution status

<https://www.softwaretestingmaterial.com/test-metrics/>

**100. When to stop testing? (Or) How do you decide when you have tested enough?**

There are many factors involved in the real-time projects to decide when to stop testing.

1. Testing deadlines or release deadlines
2. By reaching the decided pass percentage of test cases
3. The risk in the project is under acceptable limit
4. All the high priority bugs, blockers are fixed
5. When acceptance criteria is met

**3. What is the KEY difference between preventative and reactive approaches to testing?**

Preventative tests are designed early; reactive tests are designed after the software has been produced.

**4. What is the purpose of exit criteria?**

The purpose of exit criteria is to define when a test level is completed.

**8. Which of the following defines the expected results of a test? Test case specification or test design specification.**

Test case specification defines the expected results of a test.

**15. What is the difference between Testing Techniques and Testing Tools?**

Testing technique: – Is a process for ensuring that some aspects of the application system or unit functions properly there may be few techniques but many tools.

Testing Tools: – Is a vehicle for performing a test process. The tool is a resource to the tester, but itself is insufficient to conduct testing

**23. What are the different Methodologies in Agile Development Model?**

1. Extreme Programming (XP)
2. Scrum
3. Lean Software Development
4. Feature-Driven Development
5. Agile Unified Process
6. Crystal
7. Dynamic Systems Development Model (DSDM)

**28. Consider the following techniques. Which are static and which are dynamic techniques?**

1. Equivalence Partitioning.
2. Use Case Testing.
3. Data Flow Analysis.
4. Exploratory Testing.
5. Decision Testing.
6. Inspections.

Data Flow Analysis and Inspections are static; Equivalence Partitioning, Use Case Testing, Exploratory Testing and Decision Testing are dynamic.

Incident,TD

**Test Coverage Matrix**

The test coverage is documented as simple matrix called as test coverage matrix. In this test coverage matrix the test implementation and operations are mentioned in column and the test cases are mentioned in row

**67. What is test coverage?**

Test coverage measures in some specific way the amount of testing performed by a set of tests (derived in some other way, e.g. using specification-based techniques). Wherever we can count things and can tell whether or not each of those things has been tested by some test, then we can measure coverage.

**74. What is DRE?**

To measure test effectiveness a powerful metric is used to measure test effectiveness known as DRE (Defect Removal Efficiency) From this metric we would know how many bugs we have found from the set of test cases. Formula for calculating DRE is

DRE=Number of bugs while testing  / number of bugs while testing + number of bugs found by user

**95. What is Fault Masking?**

Error condition hiding another error conditio

 Different types of white box testing are

1. Statement Coverage
2. Decision Coverage

Example : If the "Negative Value" cause a firing of unhandled system exception,  the developer will prevent the negative values inpu. This will resolve the issue and hide the defect of unhandled exception firing.

**Data Driven Testing (DDT):**In data driven testing process, application is tested with multiple test data. Application is tested with different set of values.

**116. What are the two parameters which can be useful to know the quality of test execution?**

To know the quality of test execution we can use two parameters

* Defect reject ratio
* Defect leakage ratio

Latency-???

**122. What are the categories of debugging?**

Categories for debugging

a)      Brute force debugging

b)      Backtracking

c)       Cause elimination

d)      Program slicing

e)      Fault tree analysis

**136. What is difference between Test matrix and Traceability matrix?**

**Test Matrix**:  Test matrix is used to capture actual quality, effort, the plan, resources and time required to capture all phases of software testing

**Traceability Matrix**:Mapping between test cases and customer requirements is known as Traceability Matrix

**149. Mention what is the difference between a "defect" and a "failure" in software testing?**

In simple terms when a defect reaches the end customer it is called a failure while the defect is identified internally and resolved then it is referred as defect.

**152. Explain what is the difference between latent and masked defects?**

* **Latent defect:** A latent defect is an existing defect that has not caused a failure because the sets of conditions were never met(session time out)
* **Masked defect:** It is an existing defect that has not caused a failure because another defect has prevented that part of the code from being executed

**154. Mention what are the different types of test coverage techniques?**

Different types of test coverage techniques include

* **Statement Coverage:** It verifies that each line of source code has been executed and tested
* **Decision Coverage:** It ensures that every decision in the source code is executed and tested
* **Path Coverage:** It ensures that every possible route through a given part of code is executed and tested

**155. Mention what is the meaning of breadth testing?**

Breadth testing is a test suite that exercises the full functionality of a product but does not test features in detail