

## **INTERVIEW QUESTIONS :**

### **\*#-----Section-1-----#\***

Que 1) **Difference between CRS and SRS document?**

CRS is a customer requirement specification that contains customer business requirements.

SRS is a Software Requirement Specifications which contain software related features.

Que 2) **What is a kickoff meeting?**

It's an initial meeting conducted by an organisation to develop a product.

Que 3) **Difference between debugging and testing?**

Debugging means identifying mistakes and fixing the bug, testing means identifying bugs.

Que 4) **What is retesting ?**

Verifying or checking bugs is really fixed or not.

Que 5) **what is fixing a bug?**

Modifying programs or removing wrong code and entering correct code.

Que 6) **How does the test lead decide the test cycle duration?**

- 1)Based on features develop in build
- 2)No test ers availability.
- 3)Complexity of features.

Que 7) **What is the difference between respin and new build?**

Respin contains only blocker bug fixing New build contains new + old features + fixed bugs

Que 8) **Why should we test new features first?**

Probability of identifying bugs will be more in new features compared to old.

Que 9) **Why should we test old features in every build?**

If developer creates any new build which contains changes(new features +bug fixing)

Might affect old features functionality.

Que 10) **Why do we need to maintain a separate environment for both developing and testing ?**

If we maintain common environment changes made by developers will show impact on testing.

Que11) **Difference between White Box Testing and Black Box Testing.**

<b>White Box Testing</b>	<b>Black Box Testing</b>
1.In WBT we will check each program	1.In BBT we will check each feature functionality.
2.It is performed by developers,Sometimes testers if they have coding knowledge.	2.Only Testing team
3.To perform WBT internal coding is required .	3.To perform BBT coding knowledge not require
4.As code is visible it is known as open box testing.	4.As code is not visible it is known as closed box testing.
5.To perform WBT developer use technique like statement coverage.	5.To perform BBT tester will use testing types like functional ,integration etc.

**Que:** what is functional testing?

Testing each and every component completely or thoroughly or rigorously by referring to requirement document.

**QUE:** Types of Functional Testing?

1) Over Testing:

Testing component with multiple inputs which are covering same requirement.

2) Under Testing:

Testing component with multiple inputs which are not covering requirement completely.

3) Optimized Testing:

Testing component with multiple inputs which covers requirement completely.

**QUE:** What are the types of relation?

Relations are classified into two types .

1) Mono-directional:-

Relation between features will be in any one direction.

A----->-----B

A----->-----B

E.g: Compose---->-----Inbox

## 2) Bidirectional:-

Relation between feature will be in both directions.

A -----<-->----- B

E.x: Inbox -----<-->-----Trash

QUE: **What is Integration Testing?**

Testing relation between two features or testing data flow between two features based on relation.

### Types Of Integration Testing:-

Integration testing is classified into two types based on how testing team will perform or check relations.

#### i) Non-incremental Integration Testing:-

Testing relation between features by grouping which are having a set of relation.

It is also known as BIG BANG integration testing.

### Limitations:-

- i) Its complex and time consuming.
- ii) Chances are their tester might miss one or two relations testing which might have bugs.
- iii) It can be perform affectively only if we are experienced.

## II) Incremental Integration Testing:-

Testing relation between two feature by incrementally adding one by one feature.

Incremental Integration Testing is classified into two types based on data flow between features.

### 1) Top Down Incremental Integration Testing:-

Testing data flow between parent to child or high level to low level module.

Chances are their developers may create high level module but not low level module to perform testing team might required low level module so developers will create dummy module or feature stubs.

### **Stubs:-**

It's dummy feature which will receive data from high level module.

### Limitations:-

- 1) Stub will not verify data is valid or invalid.
- 2) Stubs only receive data but not transfers.

## 2) Bottom Up Incremental Integration:-

Testing data flow from child to parent or low level feature to high level feature.

To perform bottom up , Top down is mandatory and relation between features should be bidirectional.

Chances are their developers may develop low level feature but not high level feature to perform testing on low level feature. So developer will create dummy feature Driver.

### **Drivers:-**

Its dummy feature which will receive, analyse and transfer the data.

Que 12) **What is sandwich incremental\_integration testing?**

-> It is a combination of top-down and bottom-up. We can use either stubs or drivers based on relation.

Que 13) **Difference between stubs and drivers.**

Stubs	Drivers
1) It's a dummy feature that will receive data	1) It is a dummy feature that will receive and transfer data.
2) Stubs are created when the low-level feature is not available	2) Drivers are created when the high-level feature is not available
3) Stubs are mainly used in top down approach.	3) Drivers are mainly used in Bottom-Up approach.
4) Stubs are created when the relation is mono-directional.	4) Drivers are created when relation is bidirectional.
5) Stubs are called functions.	5) Drivers are calling functions.
6) Stubs are Subprograms.	6) Drivers are the main Programs.

Que 14) **What is the defect?**

→ The deviation between the expected result and the actual result. OR Any feature functionality not working as expected.

Que 15) **Why will defects happen?**

→ 1) Because of a mistake in the program.

2) Because of logical mistakes.

3) Chances are their developer misunderstood the requirement.

4) Chances are their developer missed any feature adding.

5) Chances are their tester misunderstood the requirement.

Que 16) **Explain the following points :**

- 1) Project name: This section explains in which the project testing team found a bug.
- 2) Feature name: This section explains in which feature bugs happen.
- 3) Build no: This section explains by testing which build bug is identified.

Que 17) **What is the severity and explain its types.**

Severity: This section explains how tester identified bugs will show the impact on customer business.

- a) Blocker: Further testing is possible.
- b) Critical: High loss in customer business.
- c) Major: No loss but the customer will face some difficulties.
- d) Minor: No loss no difficulties but application will not looks good.
- e) Feature: Not a bug, just a suggestion to make the application more effective.

Que 18) **What is priority and explain its types.**

Priority: This section explains how fast tester identified bug need to be fixed by developer.

\*Priority types\*

- a) Immediate/urgent: If bugs need to be fixed in the same build.
- b) High: Bug needs to be fixed in the next build.
- c) Medium: Bug needs to be fixed within 5 to 10 builds.
- d) Low: Bug needs to be fixed in any build even delivering product to customer.



#Note: Severity is set by Tester and Priority is set by Developers sometimes tester can also set.

Severity	Priority
Blocker	Immediate
Critical	High
Major	Medium
Minor	Low

Que 19) Explain High severity with a high priority with example.

The identified bug will show a savior impact on customer business and the number of users using the application in bug identified ways are more.

\*Bug: User unable to login app.

Severity: Critical/High

Priority: High

Que 20) Explain High severity with a low priority with example.

The identified bug will show saviour impact on customer business and number of users uses applications in bug identified ways are very rare/less.

\*Bug: user unable to login using help option.

Severity: High

Priority: Low

Que 21) Explain Low severity with a high priority with example.

The identified bug will not show any impact on customer business but every user can easily understand that mistake.

\*Bug: Logo contains mistakes.

Severity: Low

Priority: High

Que 22) Explain Low severity with a Low priority with example.

Identified bugs will not show any impact on customer business no user can understand mistakes are less.

\*Bug: In terms and condition 50 th line content mistake.

Severity: Low

Priority: Low

Que 23) Explain Reproducibility and its types.

This section explains whether the bug is consistent or inconsistent.

a) Consistent Bug: If the tester is able to reproduce the bug in all the attempts.

Status should be always.

b) Inconsistent Bug: If the tester is unable to reproduce the bug in multiple attempts.

# Status:

1) Alternate: Bug represent in alternate attempt.

2) Rarely: Bug reproduces twice or thrice in multiple attempts.

3) Only once: Bug reproduce only once in multiple attempts.

Que 24) Why should we send the inconsistent bugs to the developing team?

Chances are there bug which is inconsistent at testing side might become consistent at customer site. To have a proof document that the testing team identified the bug we need to send it to the developer.

Que 25) How will you fix inconsistent bugs?

By referring to the Log file.

Que 26) What is Log file?

Its document which will record all the actions that are performed on the application automatically.

Que 27) Explain the following points:

Platform:-

This section explains in which O.S and browser testing team found bugs.

# Bug summary:

Overview of the bug.

# Bug Description:

In details about the identified bug.

#Steps to reproduce:

Step by step procedure how tester identified bug.

#Expected Result:

How the system should respond.

#Actual Result:

How the system is responding.

# Attachment:

Bug related documents (log file, screenshots).

Que 28) **where we should prepare defect reports?**

- 1) Word documents.
- 2) Excel.
- 3) Spreadsheets.
- 4) Defect tracking tools (Mantis, Bugzilla)
- 5) Test management tools- Qc, JIRA.....

Que 29) **When should a tester send a bug?**

Immediately when the tester identifies the bug.

Que 30) Why can't the tester send all bug reports at the end of the day?

- 1) Chances are the tester might miss or forget one or two bug logging
- 2) Delay will happen in assigning bugs to the developer.

Que 31) What is a Duplicate bug?

Same bug identified by multiple testers.

Que 32) Why will duplicate bugs happen?

- 1) Become of common feature navigation.
- 2) Might be the relation between features.

Que 33) For whom the testing team should send a bug report?

→ Developing lead.

Que 34) Why should we not send bug reports to test leads or developers?

- 1) We don't know who develop that feature
- 2) If we send a bug report to test lead delay will happen in assigning bugs to the developer.

Que 35) Explain Defect Life Cycle

- 1) When the tester identifies a bug prepare a bug report with status new/open and send it to the development team.
- 2) Develop lead will review bug reports, identify developers who develop that feature and assign bug reports by changing status Assign.

3) Developers will review bug reports, reproduce a bug and modify source code, change bug report status to Fixed.

a. When a new build comes for testing the tester will recheck bug fixes or not if it is fixed change status to Closed, if bug is not fixed to Reopen and send it to develop lead.

4) This process continues until the bug is fixed.

Que 36) Why develop lead is give Duplicate Bug?

1) This status is given by develop lead

2) If the status is given developing lead not accepted tester bug report because similar bug report already assigned to developer.

Que 37) Why will duplicate bugs happen?

1) Because of common features navigation.

2) Chances are their features are having relations which are assigned to testers.

New/open>Duplicate>Closed

Que 38) Why do developers give Postponed/Deffert status?

This status is given by the developer. Developer accepted the bug but it will be fixed later or in the next build.

Que 39) Why did the developer change bug report status to postponed?

1) Chances are their developer is busy in adding new features.

2) Might be bug severity is major or minor.

- 3) Bug priority might be low or medium.
- 4) If the customer is expecting any changes in bug identified features.

New->Assigned->Postponed/Deferred->Fixed->Closed/Reopen

Que 40) Why developers give status Unable to reproduce.

This status will be given by the developer. If a developer is unable to reproduce a tested identified bug.

Que 41) Why developers will give unable to reproduce status?

- 1) Chances are their bug might be inconsistent.
- 2) Developers might use different platforms to reproduce the bug.
- 3) If the build is not installed properly in the testing environment.
- 4) Might be a bug report not clear to reproduce the bug.

a) Inconsistent:

Open->Assigned->Unable to reproduce->

b) Build installation problem:

Open->Assigned->Unable to reproduce->closed

c) Bug report not clear/platform:

Open->assign->unable to reproduce->fixed->closed/reopen

Que 42) Why developers can't be fixed?

This status is given by the developer. Developer accepted a tester identified bug but it will not be fixed.

Que 43) Why can't developers fix tester identified bugs?

- 1) Might be technology not supporting to fix the bug.
- 2) Chances are their cost of fixing is more when compared to loss given to loss given to that bug.

Que 44) Why developers are given status Rejected?

This status will be given by the developer when the bug is not accepted.

Que 45) Why did the developer not accept the tester identified bug?

- 1) Chances are their tester understands the requirement.
- 2) Chances are their developer misunderstood the requirement.
- 3) If the bug is inconsistent.
- 4) If a developer adds any new feature without updating the customer.
  - a) TE .Misunderstood-> New->assign->rejected->closed
  - b) DE .Misunderstood-> New->Assign->rejected->reopen->fixed->closed/reopen.
  - c) Inconsistent.
  - d) New feature.



Qu 46) Why do developers give status RFE (Request for enhancement)?

-->

This status will be given by developer if it is a bug from tester point of view but it can't be accepted from developing point of view because bug related content is not documented in requirement document.

All RFE's will be discussed with the customer if customer agrees then fix the bug otherwise reject.

(Customer approved) fixed->closed/reopen

New->assign->RFE->

( Customer not approved) Rejected ->closed

Que 47) What is smoke testing?

→ Testing basic and critical feature functionality before accepting the build.

To perform Smoke testing we use only valid data. So it is also known as Positive Testing.

Que 48) Explain Smoke testing?

→

- 1) Developers will develop a few features after WBT sends the build to the testing team.
- 2) Test lead will decide test cycle duration.
- 3) Testing lead will start testing each feature chances are their tester might identify blocker bug at the end of the test cycle.

- 4) To fix that blocker bug and to retest and respin the testing team might require sufficient time which results in delay in next delay testing.
- 5) To avoid this before testing build testing lead to verify whether build contains blocker bugs are not by performing smoke testing. If there is no blocker bug then accept the build and perform through testing.

Que 49) **Why should we perform Smoke testing?**

- 1) To identify blocker bugs at initial state.
- 2) To avoid delay in new build testing.
- 3) To avoid increasing project duration and cost.
- 4) To reduce tester efforts.
- 5) To insure the build is testable or not.
- 6) It acts as an entry criteria for the developing team to develop new builds.

Que 50) **What type of testing will perform in Smoke?**

Any type of testing which performed with valid data is considered as smoke testing.

Que 51) **When we should perform smoke testing?**

- 1) Developer will perform smoke testing after every build installation is testing the environment.
- 2) Tester will do smoke testing on every build before accepting.
- 3) Customers will do smoke testing after installing a product at a production site with sample/dummy data. i.e Dry Run Testing.

**\*#-----Section-2-----#\***

Que 52) **What is adhoc testing?**

→ Definition 1: Testing application randomly without referring to any requirement document.

Definition 2: Testing application by using creative scenarios.

Definition 3:- Testing an application without using any logic.

Que 53) **Why should we perform adhoc testing?**

- 1) Chances are their end user might use the application randomly and face difficulty to avoid that we should perform adhoc testing.
- 2) Application quality will increase.
- 3) To identify more bugs.
- 4) To improve tester test efficiency.
- 5) To improve test coverage.
- 6) To break product.

Que 54) **When we need to perform adhoc testing?**

- 1) While testing a build if the tester identifies creative scenarios stop normal testing and perform adhoc testing.
- 2) If we identify more creative scenarios, document and perform adhoc testing at the end of the test cycle.
- 3) If we identify more creative scenarios but time is not sufficient perform adhoc testing at the end of next test cycle.
- 4) In every build tester identified more creative scenarios but time is not sufficient contact test lead so that new tester will be assigned to perform adhoc testing.

Que 55) **Explain types of adhoc testing.**

1) Monkey testing:-

In this type every tester needs to identify creative scenarios on their assigned feature and performs adhoc testing.

2) Interchange testing:

In this approach every tester will identify creative scenarios, another tester assigned features and performs adhoc testing.

3) Pair Testing: In this approach experienced tester will identify creative scenarios and send it to tester by referring to those scenarios other tester will perform adhoc testing on their assigned feature.

4) Buddy testing:-

In this approach developers will explain internal design and coding structure to testers based on that knowledge creative scenarios will be identified and adhoc testing will be performed.

Que 56) **Difference between Defect, Bug, Error, Failure.**

- 1) Defect: If tester identifies any feature functionality not working as expected will be terms as defect.
- 2) Bug: If a developer accepts a tester identified defect it will be considered as Bug.
- 3) Error: Mistake in program is known as error.
- 4) Failure: If an application is unable to support customer business it will be considered as failure.

Que 57) **What is Reliability Testing?**

Verifying or checking application features functionality continuously for a period of time.

Que 58) **What is Migration Testing?**

Verifying or checking features functionality which are developed in old technology or supportable in new technology or not.

Que 59) **What is Recovery Testing?**

1) Chances are their application might get crashed at a customer site so we need to rebuild and check if the application works as expected or not.

2) Chances are their tester might immediately crash the software and check if rebuilt software works as expected or not.

Que 60) **What is a test Case?**

It's a document that contains all possible inputs that are used to test applications.

Que 61) **Why do we need to prepare test cases?**

- 1) To have consistency in testing.
- 2) To improve product quality.
- 3) To improve test coverage.
- 4) To avoid training of new testers.
- 5) To avoid delay in build testing.
- 6) It acts as a proof document to the customer that the testing team has tested the application.
- 7) Its base document for Automation testing.
- 8) To depend on the process rather than the person.

Que 62) **Difference between test case and test scenario.**

Test Scenario: Action that needs to be performed.

Test Case: Step by step procedure to execute action.

Que 63) What type of cases to be documented.

- 1) Functional Testing
- 2) Integration Testing
- 3) System Testing
- 4) Adhoc Testing

Que 64) Explain Test Scenario review process.

- 1) By referring to the requirement document every tester will prepare test scenario documents and send it to the test lead for approval.
- 2) Test lead will assign test scenario documents to the reviewer.
- 3) Reviewer by referring to the requirement document review test scenario identifies mistakes and prepares the review document and sends it to the author.
- 4) By referring to the review document author will fix all the mistakes and modify the test scenario document will be shared with the reviewer.
- 5) Reviewer will recheck if all mistakes are fixed or not and send it a letter for confirmation to test the lead.
- 6) Test lead will approve the test scenario document.

Que 65) What type of mistakes reviewers will identify?

- 1) Missing Scenario
- 2) Wrong Scenario
- 3) Modification Scenario

Que 66) How will the test lead identify the reviewer?

- 1) Chances are their features might be related
- 2) Based on domain knowledge
- 3) Based on testing experience

Que 67) **What are review ethics?**

- 1) Always review content but not the author.
- 2) Always identify mistakes but not a solution.
- 3) Even after review if any mistake is there both reviewer and author responsible.

Que 68) **What is Boundary Value Analysis?**

If any component accepting range A to B to test that component inputs are A-1, A, A+1 & B-1, B, B+1

E.g: Amount

Amount:

Req: It should accept within range 500 to 50000

Inputs: 499, 500, 501, 49999, 50000, 50001.

Que 69) **What is Pressmen Technique?**

If any component accepts a range test that component for one valid and two invalids.

a. Component accepts range

(One invalid I/P) --<A----valid input-----B<(One invalid I/P)

Eg: Account

Req: a) It should accept amount within range 500-50000

Inputs:-

1)30000

2)100

3)60000

Que 70) **What is Practice Technique?**

- a) If a component accepts range then test that component for multiple valids and two invalids.
- b) Multiple valid inputs identified by dividing range into small parts each part is known as equivalent class.

Ex.amount

- a) It should accept 500-50000

Inputs: 1)500    2)2123 3)12000 4)21000 5)25000 6) 29121  
7)38000 8)42000 9)50000

- c)If a component accepts a set of values test that component for multiple valid and two invalids.
- d)If a component accepts boolean condition then test multiple components for both true and false.

Que 71) **What is the error Guessing technique?**

→ This technique is used to predict inputs by referring to requirement. This tech is used if BVA and equivalence partitions are not possible to cover requirements.

Que 72) **What is the use case?**



→ Its document which explains how a system or software needs to respond for user actions.

Use cases are user oriented but not system oriented.

Que 73) **Who will use the use cases?**

→ 1) Developers might refer to use cases to implement coding.

2) Testing team might use cases to write test cases.

3) Business analysts or customers might refer to use cases to identify requirement specifications and to understand software needs.

4) UML(Unified Modeling Language).

Que 74) **How use cases will be prepared?**

→ By using UML(Unified Modeling Language).

Que 75) **What is the state transition technique?**

→ This technique is used to identify application behavior or state switching based on the previous history.

Que 76) **What is a Decision table?**

→ This technique is used to identify all possible ways to document test cases based on requirement and actions.

Que 77) **What is test data?**

→ Data that is created by the testing team to test the application.

Que 78) **What is the precondition?**

→ The conditions that need to be satisfied before executing the test case.

Que 79) **What is the platform?**

→ Where test cases need to be executed.

Que 80) **What is the test case repository?**

→ A common location where every tester needs to store all approved test cases.

Que 81) **Which will act as a test case repository?**

→ 1) Shared folder.

2) Test management tools like Qc, TestLink, JIRA, Test Rail.

3) Version control tools -

GITHUB, SunServer, Bit Bucket.

Que 82) **How will you review the test case?**

→ 1) I will check whether the author follows the standard test case template or not.

2) I will check if the test case header part is filled with relevant data or not.

3) I will check test case severity and priority relevant or not.

4) I will check if the test data is sufficient for the execution test case or not.

5) I will check all requirement preconditions documented or not.

6) I will check if the test case is standard with navigation steps or not.

7) I will check all test scenarios documented or not.

8) I will check if test scenarios are documented in organized order or not so that test case steps will be optimized.

9) I will check whether the actual result and status columns are empty or not.

Que 83) **What is the requirement traceability matrix?**

→ It's a document that explains every requirement given by the customer is covered at least once in the test case documented by the testing team.

Que 84) **When we should prepare RTM?**

→ Once all test cases are stored in the test case repository before executing test cases, the testing team needs to prepare RTM.

Que 85) **Why do we need to prepare RTM?**

→ 1) It acts as the proof document for the Customer.

2) The testing team will get confidence whether all requirements are covered or not.

3) If the customer modifies any requirement, the test team easily identifies which test case needs to be updated.

4) Manual testers can easily identify test cases that need to be executed manually if automation testing is not possible.

Que 86) **Explain types of RTM.**

→ 1) Forward traceability matrix:-

Preparing RTM by mapping from the requirement to test cases.

2) Backward traceability matrix:-

Preparing RTM by mapping from test cases to requirement.

3) Bidirectional traceability matrix:-

It's a combination of forward and backward RTM.

Que 87) **When the testing team will prepare the test case?**

→ 1) While developers develop application testing teams will prepare test cases.

2) While testing builds if the tester identifies missing or creative scenarios then test cases need to be prepared.

3) If the customer modifies the requirement then the test case needs to be updated.

Que 88) **What is the retrospect meeting?**

→ In this meeting the test lead will interact with every team member and identifies all the drawbacks that are faced by the testing team.

All the difficulties will be documented in retrospect and store it in QMS(Quality Management Services). This document will be helpful to test leads for upcoming projects to prepare a test plan.

Que 89) **Difference between regression and retesting.**

→ Retesting definitions:-

- 1) Testing fixed bugs to ensure bugs are really fixed or not.
- 2) Testing changed features to check changes are effective or not.
- 3) Re Executing previous build failed test cases to insure test case pass or not.

Regression Testing definitions:-

- 1) Testing old features in a new build.
- 2) Testing unchanged features to insure because of modifications its affected or not.
- 3) Re Executing previous build pass test cases in new build.

Que 90) **Why should we perform regression testing?**

→ 1) Chances are their old feature functionality might get affected because of fixing bugs.

2) If a developer adds any new feature, old feature functionality might get affected.

Que 91) **What type of testing will perform in regression?**

→ Any type of testing performed on old features will be termed as regression.

Que 92) **What type of testing will you perform automation?**

→ Regression.

Que 93) **Do we need to test all old features in the new build?**

→ No, It depends upon fixed bugs and newly added feature relations with old features.

Que 94) **Explain Types of regression testing.**

→ There are basically three types of regression testing as follows:

**1) Unit regression testing:**

Testing unchanged feature functionality to ensure because of fixing a bug in one functionality other functionalities of the same feature are affected or not.

You need regression testing to be performed on independent features.

**2) Regional regression testing:-**

Testing related features to insure because of fixing bugs its related feature affected or not.

**3) Full regression testing:-**

Testing all old features in a build or testing complete application.

Que 95) **When we should perform full regression testing?**

→ 1) Before realising the product to the customer.

2) If any critical feature functionality is changed.

Que 96) **How will you identify related/ old features that need to perform on regression?**

→ 1) Based on domain knowledge.

2) By preparing an impact analysis matrix.

3) By conducting an impact analysis meeting in this meeting, the testing team will discuss changes made in the new build and identifies all the features that need to be performed with regression testing.

Que 97) **What is an impact analysis matrix?**

→ It's a document which explains how bugs fixed features will impact other features.

Que 98) **Difference between smoke and sanity testing.**

→ **Smoke testing:-**

1) It's build verification testing where all features functionality will be tested before accepting the build.

2) In smoke we perform only positive testing.

3) Smoke testing is scripted.

4) It's wide and shallow.

**Sanity testing:-**

1) It's build verification testing where critical features functionality will be tested completely before accepting the build.

2) In sanity we perform positive and negative testing.

3) Sanity testing is unscripted.

4) It's narrow and deep.

Que 99) **What is sanity testing?**

→ **Definition 1):** It's build verification testing where critical features functionality will be tested completely before accepting the build.

**Definition 2):** It's build verification testing where critical flows in old features are tested before accepting the build.

It's similar to regression testing.

**Definition 3):** It's similar to full regression testing where critical flows in old features will be tested completely before realising the product to the customer.

**Definition 4):** It's build verification testing where new features functionality will be tested before accepting the build.

\*#-----**Section-3**-----#\*

Que 100) **What is SDLC?**

→ It's a procedure followed by an IT organisation or project team to develop an application.

Que 101) **Why did we require SDLC?**

→ To develop software IT organisations will depend on cost, cost and time. If we focus on any two factors other factors will show severe impact on product development.

To optimize all these three factors we use SDLC.

Que 102) **Explain stages in SDLC.**

→ **Stage 1 (Requirement collection):**

- 1) IT organisations are classified into service base and product base.
- 2) In service based organisation customer will prepare CRS document and send it to business analyst and business analyst will convert it into SRS.

CRS: It explains customer business

SRS: It explains how software needs to be developed.

FRS: It explains how each feature should work.

- 3) In product based organisations product analysts will gather requirements by analysing market situations.

**Stage 2( Feasibility Study):**

In this stage organisation will estimate or analyse customer needs and decide whether it is possible to develop an application or not.

In this meeting organisation will estimate the number of resources required, cost, duration, etc.

**Stage 3( Design):**

In this stage architects and senior developers are involved. The main purpose of design is to make developers understand what kind of software needs to be developed.

Design is classified into :

- 1) High level design:-

It explains in detail each component .

- 2) Low level design:-



It explains in detail each component.

### 3) Detail level design:-

It explains each component's logical sequence/ flow.

### **Stage 4( Coding):**

In this stage the developer will develop few features and perform WBT and send the product to the testing team.

### **Stage 5(Testing):**

In this stage the tester will test product identifies bugs and send it to developers.

### **Stage 6(Installation):**

In this stage senior developer or realise engineer or installation engineer all involved. We need to explain the installation process to customers and how to use software.

### **Stage 7(Maintenance):**

Project manager will arrange a maintenance team to maintain the product for a certain period.

This maintenance team will resolve customer problems and add new features if customer requirements change.

Que 103) **What is the cost of fixing bugs?**

→ It will be decided based on the number of teams involved in fixing bug.

- 1) Expensive:- To fix bugs the whole project team is involved.
- 2) Moderate:- To fix bugs more than two teams are involved .
- 3) Low:- To fix bugs only two teams are involved.

Que 104) : What is a waterfall model, and explain advantages and disadvantages.

- 1) Waterfall model is a traditional and old model followed by organisations to develop applications. In this model application will be developed in design top-down approach.
- 2) To develop the stage, the previous stage needs to be completed 100% and it should be freezed.
- 3) We can't make any changes to the previous stage.

Advantages:-

- 1) Simple model to develop application.
- 2) Project cost will be less.

Disadvantages:-

- 1) Developers are involved in testing applications.
- 2) Backtracking is not possible.
- 3) Cost of fixing bugs is expensive.
- 4) Requirement and designing stages are not tested.
- 5) Product cost might get affected.

Que 105) When should we use the waterfall model?

- 1) If Customer freeze requirement.
- 2) For a short term project.

Que 106) What is backtracking?

→ Changing requirements frequently by customer as per requirement changes.

Que 107) Why developer should not perform testing?

- 1) Chances are their developers may do only positive testing.
- 2) Because of over confidence few developers may not perform effective testing.
- 3) Few developers might utilize it to develop applications.

Que 108) **Explain Spiral model and advantages and disadvantages also.**

- 1) In this model application will be developed in stages.
- 2) The customer will prepare a set A requirement and send it to a business analyst.
- 3) BA will prepare an SRS document for set A and send it to the designing team.
- 4) Architect and senior developer will create HLD and LLD for set A requirement and send it to the developing team.
- 5) Developer will develop the product in build wise and send it to the testing team.
- 6) Once all features in set A are developed, release the product to the customer.
- 7) Customers will start using this software and store a new set of requirements for the project team.

**Advantages:-**

- 1) Backtracking is possible.
- 2) Complex applications can be developed in stages.
- 3) We can expect high quality products.

**Disadvantages:-**

- 1) Testing phase is involved only after coding.
- 2) Initial stages like requirement and design are not tested.
- 3) Cost of fixing bugs is expensive.

Que 109) **When should we use the Spiral model?**

- 1) If a customer is expecting a product within a short period.
- 2) If customers want to give requirements in stages.
- 3) For complex and long term projects.

Que 110) **Explain Proto-type model and advantages and disadvantages also.**

- 1) In this model the customer will prepare requirement documents and send it to the project team.
- 2) Designing team will convert the requirement document into prototype images/ Wireframes.
- 3) These wireframes will be shared with the customer if the customer wants to make any changes at the initial stage can be done.
- 4) Once wireframes are approved, actual application development will start.

**Advantages:**

- 1) At the initial stage only we can set expectations for customers about software.
- 2) Good communication between customer and project team.
- 3) Product can be developed as per customer expectations.
- 4) Customers can make requirement changes at the initial stage.

**Disadvantages:**

- 1) Actual application development will be delayed.
- 2) Initial investment is high.
- 3) Project duration will increase.
- 4) Cost of fixing bugs is moderate.
- 5) Requirement collection and design stages are not tested.

Que 111) When should we use a prototype model?

- 1) If the customer has expectations product.
- 2) If the developer knew to domain.
- 3) If the customer is not clear about business requirements.

Que 112) Explain V-V model and advantages and disadvantages also.

- 1) Customer will prepare the CRS and send it to the testing team.
- 2) Testing team will review CRS, identify mistakes, prepare review documents and send it to the customer.
- 3) Customers will modify the CRS document and share the modified document with BA and the Testing team.
- 4) Ba will convert CRS to SRS and the testing team will prepare acceptance testing test plan ,test cases.
- 5) BA will share the SRS document to the testing team.
- 6) Testing team will review SRS and send mistakes to BA.
- 7) BA will modify SRS and share it with the architect and testing team.
- 8) Architect will prepare the HLD and the testing team will prepare the test plan, test cases.
- 9) Architect will share the HLD with the testing team which will be reviewed and mistakes will be sent to the architect.
- 10) Architect will modify the HLD and share the document with the senior developer, testing team.
- 11) Senior developers will convert HLD to LLD and the testing team will prepare integration testing test plans and test cases.

12) Senior developers will share LLD to the testing team which will be reviewed and mistakes will be shared to senior developers.

13) Modified LLD will be shared with the development team and testing team.

14) Developer will develop the application and the testing team will prepare a functional testing test plan and test case.

15) After white box testing the developer will send the build to the testing team.

16) By referring to test cases the testing team will perform BBT like Functional, Integration.

17) Once features are completely developed after acceptance realise product to the customer.

#### **Advantages:**

- 1) Cost of fixing bugs is low.
- 2) Backtracking is possible.
- 3) Overall project cost is less.
- 4) Testing team is involved in every stage.

#### **Disadvantages:-**

- 1) Too much documentation.
- 2) Initial investment is high.

Que 113) **When should we use the V and V model?**

→ 1) For long term projects.

2) If the customer expects a high quality product.

Que 114) **Explain Agile model and advantages and disadvantages also.**

→ 1) Agile is the combination of waterfall model and spiral model.

2) An agile model application will be developed by using scrum methodology.

**Scrum:** Its framework or procedure followed by scrum team to develop application sprintwise.

- a) Customer or stack holder will prepare a requirement document or user stories or epics and send it to the business analyser/ product owner.
- b) BA will prepare SRS documents.
- c) Stand up meeting will be conducted between stack holder and product owner.
- d) In this meeting stack holder will prioritize user stories.
- e) Product owners prepare documents as per stack holder priorities order. I.e. product backlog.
- f) Product backlog will be shared with the scrum team.
- g) Scrum team will split product backlog into sprint backlogs.
- h) Each sprint document will contain at least of tasks or user stories that need to be completed in the sprint.
- i) To develop a sprint scrum team will conduct a sprint analysis meeting.
- j) In this meeting the scrum team will conduct decides sprint duration by dividing each user story into tickets or units.
- k) Designing, coding and testing phase will execute parallely to develop the sprint.
- l) Every day scrum master will conduct a daily standup meeting or scrum meeting.
- m) In this meeting every team member should explain what they did yesterday, what the difficulties face and what they will do today.
- n) At the end of the sprint scrum master will conduct a sprint execution meeting.
- o) In this meeting the scrum team will give a live demo on sprint to customers.
- p) If the customer is satisfied, the product should be shifted.

- q) Before developing the next sprint scrum master will conduct a sprint retrospect meeting.
- r) Every team member should explain any difficulties faced in the previous sprint.
- s) All the difficulties will be documented and identify solutions that in the next sprint development scrum team should not face difficulties.

#### **Advantages:-**

- 1) Backtracking is possible.
- 2) Cost of fixing bugs is low.
- 3) Good communication between customer and organisation.
- 4) Customers can easily track application progress.
- 5) Customers can use the product within a short period.

#### **Disadvantages:-**

- 1) Too much documentation.
- 2) Initial investment is high.
- 3) Application development will be delayed.

Que 115) **What is scrum?**

→ Its framework or procedure followed by the scrum team to develop application sprintwise.

Que 116) **What is sprint?**

→ It's the time duration taken by the scrum team to develop a few user stories.

Sprint duration will be 1 week to 4 week.

Que 117) **When should we use Agile?**

→ 1) For a long term project.

2) If the customer is not clear about requirements.



3) If a customer is expecting a product within a short period.

Que 118) **What is a Burn-Down chart?**

→ It's a document which explains no of units of work scrum teams need to complete per day.

Que 119) **What is a Hybrid model?**

→ It's combination of two models-

I.e. Spiral + prototype, v&v + Prototype, etc.

Que 120) **What is system testing?**

→ It's End to End testing performed when the testing environment is similar to production environment.

Que 121) **What is End to End testing?**

→ Navigating through all features and checking end feature functionality working or not.

Que 122) **Why should we perform system testing?**

→ 1) To avoid loss in customer business.

2) To check whether developing software supports customer business or not.

Que 123) **When we should perform system testing?**

→ 1) After functional and integration testing.

2) When the testing environment is similar to the production environment.

3) When the minimum number of modules or features is functional is stable.

Que 124) **What is the environment? Explain its types.**

→ It's setup which contains server, hardware, networking and database configuration.

Types of environment:

**1) Development environment: -**

Its setup contains a server used by the development team to develop applications.

**2) Testing Environment:-**

Its setup contains a server where the testing team tests the application.

**3) UAT Environment:-**

Its setup contains a server where customers will perform acceptance testing.

**4) Production Environment:-**

Its setup which contain server with huge configuration where end users will use the application.

Que 125) **Why testing environment should be similar to the production environment?**

→ Chances are their features which are working in a testing environment may not working production environment because of production changes.

To avoid this we need to maintain a staging environment.

Que 126) **What is the staging Environment?**

→ It's setup which contains configuration similar to production environment.

It's provided by a customer.

Que 126) **What is bug release?**

→ Bug identified tester by tester not fixed by developer and known by customer.

Que 127) **What is release?**

→ Time taken by the project team to gather requirements, develop and test for multiple builds, deliver products to customers.

For every project will have multiple releases in every release product will be tested for multiple builds.

Que 128) **When should we release products to customers?**

→ 1) When all features are completely developed.

2) No critical/ blocker and major bugs.

3) Few minor bugs are acceptable.

4) System testing/ Customer business flows should work fine.

Que 129) **What is acceptance testing?**

→ Testing performed by the customer to check if the application supports customer business or not.

Que 130) **When will the customer perform acceptance testing?**

→ 1) Once a product is delivered to the customer before moving it to production.

Que 131) **Why will the customer perform acceptance testing?**

→ 1) Under business pressure the project team might realise a product with bugs.

2) To check if the application supports customer business or not.

3) To avoid loss in customer business.

4) Chances are their customer might miss one or two requirements documentation. While sharing requirements documents at the initial stage by performing acceptance testing, customers will get a chance to add requirements.

Que 132) **How will the customer perform acceptance testing?**

→ 1) To perform acceptance customer will take help of-

- a) End user
- b) By hiring its own testing team.
- c) With the help of a software site testing team.

Que 133) **Acceptance testing definitions/Approaches:**

**Approach1:**

It is end to end testing performed by end user by taking real time business situations and checks application supports customer business or not.

**Approach 2:**

It is end to end testing performed by client site testers by referring to user documents and checks application supports customers business or not.

**Approach 3:**

It is end to end testing performed by software site tester at client site by referring to user documents and checks application supports customer business or not.

**Approach 4:**

It is end to end testing performed by software site tester at organisation level by referring to user document and checks application supports customer business or not.

Que 134) What type of testing performed in acceptance?

-> End to end testing.

Que 135) What is the difference between system end to end and acceptance end to end?

-> In system end to end software site testers by referring to customer requirement checks application supports customer business or not.

In acceptance end to end by taking real time business situation application will be tested.

Que 136) What is alpha testing?

-> alpha testing is performed at organisation level where developers and testers are involved.

Que 137) What is beta testing?

-> Beta testing is performed at client site where end users are involved.

Que 138) What is defect masking?

-> Bug in a feature functionality will restrict tester to identify bug in other functionality of that feature.

Until that bug is fixed other bugs can't be identify by tester.

Que 139) What is latent bug?

-> Bug identified by tester, not fix by developer, not known to customer.

Que 140) What is test bed?

-> It's combination of test data and test environment.

Que 141) What is patch?

-> It's piece of software which contains modified program or new program.

Que 142) **When we need to create patch?**

-> 1) When we identify blocker bug.

2) For new feature.

3) To update application security.

Que 143) **What is defect clustering?**

-> Defect clustering means distribution of defect will not be uniformed across the feature but it will be limited or centalised in a specific feature.

Advantages of defect clustering:

1) Tester can improve test efficiency.

2) Testing team can identify more no of bugs.

Que 144) **What is hotspot?**

-> While testing as a tester we need to observe which feature creates more no. of bugs that are known as hotspots.

Que 145) **What is pesticide paradox?**

-> Pesticide paradox means executing same test case or same inputs for multiple times will drop defect numbers or count.

Que 146) **What is bug leakage?**

-> Bug identified by end user or customer while using application.

Que 147) **What is hot fix?**

-> Time taken by project team to fix end user / customer identified bug.

Hot fix duration will be 1 hr- 1 day based on bug severity.

Que 148) **When we should regression testing?**

-> In release1 regression will starts.

- From build 2 onwards from next release regression will start from build.

Que 149) **What is compatibility testing?**

-> Testing applications in multiple operating systems using browser and versions.

Que 150) **Why do we need to perform compatibility testing ?**

-> 1) To avoid loss in customer business.

2) If the application is not working in multiple browsers then no of users might reduce.

3) Chances are their negative impact or bad reputation will spread to customer business.

Que 151) **When we need to perform compatibility testing?**

-> 1) Once the staging environment is set up.

2) Once the application is completely stable in the base platform.

3) Before releasing products to customers.

Que 152) **What type of testing will be performed while doing compatibility?**

-> We can perform any type of testing but it consumes more time so mainly we perform only system testing.

Que 153) **What are types of compatibility testing?**

-> 1) Forward compatibility testing:-

Testing application in latent version.

2) Backward compatibility testing:-

Testing application in older version.

Que 154) **How to perform compatibility testing?**

-> Approach 1:

In this approach test lead will assign complete application to testing team. Each tester need to perform compatibility testing an assign platform.

Approach 2:

In this approach multiple platforms will be provided to each tester and compatibility testing will be performed on assigned features.

Approach 3:

BY using a compatibility testing tool like browser stack vm more software is used.

Que 155) **Do we need to perform compatibility testing on every application?**

-> No, compatibility testing will be decided based on software uses by the customer. If a product is used within organisation compatibility testing is not required.

If software is used outside organisation the compatibility is mandatory.

Que 156) **What is defect triage?**

-> It's a meeting conducted by the project team to prioritize bugs.

Que 157) **What is globalisation testing?**

-> Developing applications in multiple languages is known as globalise application.

Testing application for multiple languages is known as globalisation testing.

Que 158) **Why do we need to perform globalisation testing?**



-> 1) Chances are there if application content is not displayed properly users might face difficulty.

2) To verify the correct content displayed at the right place or not.

3) To verify selected language content displayed or not.

Que 159) **When we need to perform globalisation testing?**

->1) When the staging environment is provided.

2) When application is stable in base language and before releasing product to production.

Que 160) **What are the types of globalisation technique?**

-> 1) **I18N**:(Internationalisation)

In this type we will check as per language selection any feature functionality changed or not.

2) **L10N**:( Localisation)

Application content might change as per selected country standards.

Que 161) **What is performance testing?**

-> **Definition 1**:)Testing speed or response time for application features.

**Response Time**: Total time taken by system to send request to server+ Time taken by server to analyse request+ Time taken by server to response to system.

**Definition2**): Testing stability of an application by applying huge load.

Que 162) **Why do we need to perform performance testing?**

-> 1) To verify application sustenance load changes or not.

2) To check application responsiveness and scalability.

- 3) To identify the crash point.
- 4) To identify bottlenecks.
- 5) To avoid loss in customer business.

Que 163) **When we perform performance testing?**

- > 1) Once application is completely stable.
- 2) When staging environment is provided.

Que 164) **What are the types of performance testing?**

->1) **Load Testing**:-

Testing stability of an application by applying load less than maximum load.

2) **Stress Testing**:-

Testing stability of an application by applying load greater than maximum load.

3) **Soak/ Endurance testing**:-

Testing stability of an application by applying load for a timing period.

4) **Spike Testing**:-

Testing stability of an application by sudden increasing and decreasing the load.

5) **Volume Testing**:-

Testing stability of an application by transferring the data.

***\*#\*#\*#\* Manual Testing End\*#\*#\*#\****









