

@Real Time scenarios Questions asked for interview Manual & Automation testing March 2016
till date. 😊

By QA Professional group.

Prasad Hiwale 😊

“ whatever you want archive in life, Help other to achieve in that”

1) What are the key challenges of software testing that you faced in your career?

Following are some challenges of software testing that I faced in my career:

- i) Unstable Application Under Test.
- ii) Time constraints.
- iii) Understanding and Analysing the requirements.
- iv) Changing Requirements.
- v) Lack of Domain knowledge and business user perspective understanding.
- vi) Prioritizing Test cases
- vii) Lack of skilled team members.
- viii) Selecting Test Cases for Regression testing.
- ix) Lack of resources and training.
- x) Test Environment issues.

2) How you derived Test Cases?

That depends on project, sometimes we derived Test cases from requirements and sometimes from use cases.

3) How many Test cases did you write for last project?

Nearly 170 Test cases.

4) How much time is required to write a Test case?

That depends on complexity of the functionality.

5) How many defects did you detect in your last project?

I executed nearly 150 Test cases, in which some 22 defects were raised including 3 showstoppers.

6) Did you face any problems during defect reporting and tracking?

Yes, Developers rejected 2 or 3 valid defects.

7) Did you create RTM (Requirements Traceability Matrix) document in your project?

I only updated the RTM document.

8) Did you involve in Test Environment setup?

Yes, I involved in verifying Test Lab setup along with other team members.

9) Did you perform any Live testing?

Yes, In my current project we used live data for some test cases.

10) What is difference between Front End Testing and Back End testing?

Front End Testing is performed on the Graphical User Interface (GUI), whereas Back End Testing involves databases testing.

We conduct Database Testing using SQL Queries.

Database Testing is subset of Functional Testing.

11) What is the difference between System Testing and Functional Testing?

System Testing is a Test Level

Functional Testing is a Test Type that can be performed in all levels of Testing (Unit Testing, Integration Testing, system Testing and Acceptance Testing).

12) What is the difference between Performance Testing and Load Testing?

Load Testing, Stress Testing, Spike Testing and Endurance Testing all are subsets of performance Testing.

13) What are the Test types that you performed in your Software Testing career?

Functionality Testing

Security Testing

Usability Testing

Compatibility Testing

Installation Testing etc...

14) What are the Test deliverables?

What documentation we produce during testing all come under Test deliverables.

Test Plan

Test Scenarios

Test Cases

Opened and Closed Defect Reports

Test metrics reports

Test summary Report etc...

5) Did you involve in Test plan documentation?

Yes, I involved in Test plan documentation in the last project, identified Features to be Tested, Entry criteria, Exit criteria.

16) What is Exhaustive Testing?

Exhaustive Testing – testing with all possible inputs and pre-conditions and it is impractical, so we use Test design techniques to reduce the size of Input and Output domains.

17) What are the important phases in Formal Software test process or Software test life cycle?

i) Test Planning

ii) Test Design

iii) Test Execution

iv) Evaluating exit criteria and Test closure.

18) What are the important tasks in Test Planning phase?

Important tasks in Test planning phase are:

i) Understanding and Analyzing the Requirements

ii) Risk Analysis

iii) Test Strategy Implementation

iv) Test Estimations (Scope, Time, Resources, Budget etc...)

v) Team Formation

vi) Test Plan Documentation

vii) Configuration Management Planning

viii) Traceability Matrix documentation

ix) Define Test Environment Setup

19) What are important tasks in Test Design phase?

i) Understanding Requirements

ii) Generate Test Scenarios

iii) Test Case Documentation

iv) Test Data Collection

20) What are Test design techniques that you used?

I used Black box Test Design Techniques,

i) Equivalence Class Partitioning

- ii) Boundary Value Analysis
- iii) Decision Tables
- iv) State Transition Testing etc...

21) How you communicate with Developers to resolve issues?

That depends on Company and some times depends on Project, in my current project I am communicating Developers via our Test Lead.

22) What is Configuration Management? Did you any Configuration Management Tool in your Testing career?

Storing and Organizing all configurable items is called Configuration Management, It is not only for Testing Team, also for all Stake holders of the Project.

It is very important for Development than Testing Team, I used VSS Tool for Configuration Management in my last project.

23) You told Configurable items, What are Configurable items in Software Test Process?

What Software & Hardware we use and What Documents (Test Plan, Test Cases, defect Reports, Test Summary Report etc...) we produce during Testing all are come under Configurable items.

24) When we choose Informal Testing?

Whenever we don't have proper documentation (Requirements etc...) and sufficient Time then we choose Informal Testing. Using Experienced based Techniques (Ex: Error Guessing, Exploratory Testing etc...) we conduct Testing.

25) What are the important Test Types that can be applied for Web Applications?

Test Types that applied for Web Applications are,

- i) Functionality Testing
- ii) Security Testing
- iii) Compatibility Testing (OS Compatibility and Browser Compatibility)
- iv) Navigation Testing
- v) Database Testing
- vi) Reliability Testing
- vii) Usability Testing
- viii) Recovery Testing
- ix) Performance Testing Etc...

26) Do You have experience in Database Testing?

Yes, I conducted Database Testing Manually using SQL Commands,

I Tested the following Database operations during Database Testing,

- i) Data Manipulations (Add / Edit / Delete Records)
- ii) Data Integrity
- ii) Data Retrievals
- iii) Data Comparisons etc...

1. What is the difference between Quality Assurance (QA) and Quality Control (QC)?

Quality Assurance: Quality Assurance involves in process-oriented activities. It ensures the prevention of defects in the process used to make Software Application. So the defects don't arise when the Software Application is being developed.

Quality Control: Quality Control involves in product-oriented activities. It executes the program or code to identify the defects in the Software Application.

2. What is the difference between Preventative and Reactive approaches in testing?

Preventive approach: It is also known as Verification Process. This approach is to prevent defects. In this approach, tests are designed at early stages of SDLC i.e., before the software has been produced. Here in this approach testers try to prevent defects in the early stages. It comes under Quality Analysis (QA).

Reactive approach: It is also known as Validation Process. This approach is to identify defects. In this approach, tests are designed to execute after the software has been produced. Here we try to find defects. It comes under Quality Control (QC).

3. Why are you in QA?

I am in QA because I like this job.

Read more on [why did you choose Quality Assurance as a career](#)

4. List out the roles of Quality Assurance engineer?

A software quality assurance engineer usually involves in the following tasks.

- QA Team is responsible to monitor the entire development process.

- They are responsible to track the outcomes of each phase of SDLC and adjust them to meet the expectation.
- They are responsible to read and understand the requirement documents.
- Analyze test requirements, and design and execute tests.
- Develop test cases and prioritize testing activities.
- Record problems and issues in accordance with the project's problem and issue management plans.
- Work with the application team and/or client to resolve any issues that arise in the testing process.
- Carry out regression testing every time when changes are made to the code to fix defects.
- Have to interact with the clients to better understand the product requirements.
- Participate in walkthroughs of testing procedures.

5. Explain the process of QA testing?

In simple words, QA testing process is a step by step process which involves analyzing requirement documents, preparing test strategy, test plan and test cases, executing test cases when the build is ready. In the execution process QA's perform different types of testing to make sure the software reaches or exceeds the expectation.

[Read more..](#)

6. What is the role of documentation in QA?

Documentation plays a vital role in Quality Assurance. All the documents involved in SDLC such as Business Requirement Specifications, Designs, Inspection reports, Configurations, Code changes, Test Strategy, Test plans, Test cases, Bug reports, User manuals should be documented.

- Documentation helps us to achieve high quality software product.
- Documentation is necessary to make things more real
- We could use documentation as a reference material and reuse it when necessary
- We could save lot of organization's time, effort and money by maintaining proper documentation.
- Proper documentation makes easy for the client to review the software process.

7. What is quality audit?

Quality audit is the process of systematic and independent examination of a software product or process to assess compliance with specifications, standards, agreements and other relevant criteria.

8. Mention what are the test artifacts involved in QA?

The test artifacts involved in QA are Test Strategy, Test Plan, Test Scenarios, Test Cases, Test Summary Report, Bug Report etc.,

Read more and download [complete set of test artifacts](#) from here..

9. Have you written Test Strategy?

Usually, test strategy document will be prepared by Test Managers or Project Managers. If you are applying for a Project Manager position and you have experience in preparing Test Strategy document then you can say Yes else say I know what is a test strategy and its purpose but I never got a chance to write Test Strategy document.

10. What is a Test Strategy and what does it include?

Test Strategy is a high level document (static document) and usually developed by project manager. It is a document which captures the approach on how we go about testing the product and achieve the goals. It is normally derived from the Business Requirement Specification (BRS). Documents like Test Plan are prepared by keeping this document as base.

Read more on [detailed explanation of Test Strategy](#)..

11. Have you written Test Plan?

Usually, test plan document will be prepared by Test Leads or Test Managers. If you are applying for a Test lead position and you have experience in preparing Test Plan document then you can say Yes else say I know what is a test plan and its purpose but I never got a chance to write Test Strategy document.

12. What is a Test Plan and what does it include?

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, or Use Case documents for all future activities

of the project. It is usually prepared by the Test Lead or Test Manager and the focus of the document is to describe what to test, what not to test, how to test when to test and who will do what test. Also, it includes the environment and tools needed, resource allocation, test technique to be followed, risks and contingencies plan. A test plan is a dynamic document and we should always keep it up-to-date. Test plan document guides us how the testing activity should go on. Success of the testing project completely depends on Test Plan.

Read more on [detailed explanation of Test Plan..](#)

13. What is a Test case template?

A test case template is a document comes under one of the test artifacts, which allows testers to develop the test cases for a particular test scenario in order to verify whether the features of an application are working as intended or not. 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. Most of the companies are using test case management tools such as Quality Center (HP QC), JIRA etc., and some of the companies still using excel sheets to write test cases.

14. What are the key components of a test case template

The key components of a test case template are Project name, Module name, Created by, Date of creation, reviewed by, date of review, executed by, Date of execution, test scenario, tase case id, test case description, Precondition, Test steps, Test data, expected result, post condition, actual result, status of the bug.

Check the below video on how to write effective test cases.

15. How do you decide when you have tested enough?

This is one of the most important questions in terms of ISTQB. Option will be tricky and you have to choose the right one.

As a project manager or project lead, sometimes you might face a situation to call off the testing to release the product early. In those cases, you have to decide whether the testers have tested the product enough or not.

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

- if we reach Testing deadlines or release deadlines
- By reaching the decided pass percentage of test cases
- if the risk in the project is under the acceptable limit

- if All the high priority bugs and blockers are fixed
- if we met the acceptance criteria

As per ISTQB, It depends on the risks for the system being tested.

16. What are the key components of a bug report?

Bug report is aka defect report, it conveys the detailed information (such as environment details, steps to reproduce etc.,) about the bug to the developers. It allows developers to replicate the bug easily. The key components of a bug report are Defect Id, title of the defect, Reporter Name, Defect Report Date, Reporter designation, Project name, Release Version, Environment details, Priority of the bug, Severity of the bug, Status of the bug, Defect Description, Steps to reproduce the bug, Expected result, Actual result, Attachments if any and Defect closed date.

Read more on [how to write a good report..](#)

17. Tell me some key points to consider while writing a bug report.

- i. Reproduce the bug 2-3 times.
- ii. Use some keywords related to your bug and search in the Defect Tracking Tool.
- iii. Check in similar modules.
- iv. Report the problem immediately.
- v. Write detailed steps to reproduce the bug.
- vi. Write a good defect summary. Watch your language in the process of writing the bug report, your words should not offend people. Never use capital letter whilst explaining the issue.
- vii. Advisable to Illustrate the issue by using proper screenshots.
- viii. Proofread your bug report twice or thrice before posting it.

18. What are the advantage and disadvantages of Automated Testing?

Advantages:

1. Automation testing is faster in execution
2. It is cheaper compared to manual testing in a long run
3. Automated testing is more reliable
4. Automated testing is more powerful and versatile
5. It is mostly used for regression testing
6. It does not require human intervention. Test scripts can be run unattended
7. It helps to increase the test coverage

Disadvantages:

1. It is recommended only for stable products
2. Automation testing is expensive initially
3. Most of the automation tools are expensive
4. It has some limitations such as handling captcha, fonts, color
5. Huge maintenance in case of repeated changes in the requirements

Not all the tools support all kinds of testing. Such as windows, web, mobility, performance/load testing

19. What is the difference between build and release?

Build: A build is a version of a software. Every build has a number for identification purpose. Build is a pre-release version of a Release. Build is given to testing team by developers to test the application locally. Build numbers are incremental.

Release: A release is the distribution of the final version of an application to the customer by software development team.

20. What is bug leakage and bug release?

Bug Leakage: A bug which is actually missed by the testing team while testing and the build was released to the Production. If now that bug (which was missed by the testing team) was found by the end user or customer then we call it as Bug Leakage.

Bug release: Releasing the software to the Production with some known bugs then we call it as Bug Release. These known bugs should be included in the release note. In other case, releasing the software to the testing team with some known bugs whose severity and priority is low. These bugs can be removed before releasing to production.

21. What is Bug triage?

Bug triage is a formal process to find which bugs are important by prioritizing them based on their severity, frequency, risk and other important parameters. Testers assign priority (high, medium, low) to each and every bug in a bug triage meeting and based on the priority those bugs will be fixed in an order. By doing this we could save a lot of organization's time.

22. Explain bug life cycle.

Bug life cycle is also known as Defect life cycle. In Software Development process, the bug has a life cycle. The bug should go through the life cycle to be closed. Bug life cycle varies depends upon the tools (QC, JIRA etc.,) used and the process followed in the organization. [Read more..](#)

23. What is MR and ER?

MR: MR stands for Modification Request. It is used to change the existing functionality in a software, it is usually requested by clients.

ER: ER stands for Enhancement report. It is used to add a new feature in a software. It is usually requested by clients.

24. Mention some of the types of software testing?

There are more than 100 types of software testing.

Must Read: [100+ Types of Testing](#)

25. What is CRUD testing?

CRUD (Create, Read, Update and Delete) is another term used for Black box testing. CRUD testing is another term for database testing.

Read more on [Black box testing](#) here..

- C – Create – Creating a new Transaction
- R – Read/Retrieve – Searching or viewing a transaction
- U – Update – Editing or modifying an existing transaction.
- D – Delete – Deleting a transaction from the database

Must Learn: [SQL Tutorial for Software Testers](#)

26. What is a Cookie testing?

A Cookie is also known as HTTP cookie, web cookie, internet cookie, browser cookie.

Read more on [Cookie testing..](#)

27. What is Cross browser testing?

Cross Browser Testing is a type of non-functional test which helps us ensure that our website or web application works as expected in various web browsers. We could do Cross Browser Testing on different browsers both manual and automated way. To do Cross Browser Testing manually, we (Software Testers) create tests for each browser and execute it manually on each browser. To do it in an automated way, we could create Selenium tests with multiple conditional statements that execute test cases based on specified browser type. Every browser displays a website in their own style. We usually cannot have all the browsers on one machine. Each browser is designed by a different vendor. So each browser has their own features to showcase their unique presence. While testing a website, we need to ensure that our website is appearing same across all the browsers. To do this we need to have all the browsers. Fortunately, there are some tools to perform cross-browser testing without testing individually in a manual way.

Read more on [Cross browser testing..](#)

28. What is the difference between Compatibility testing and Cross browser testing?

Compatibility testing: Testing an application on different hardware or software platform is Compatibility testing.

Example: Different devices such as iPhone, Samsung etc., Different operating system such as Windows, Linux etc.,

Cross browser testing: Testing a web application on different browsers is Cross browser testing. Cross browser testing is a subset of Compatibility testing.

Example: Google Chrome, IE 10, IE 11, Firefox 43 etc.,

29. What is Configuration management?

Configuration management is a process followed during the project life cycle to control and document each and every change.

30. What are the various tools you have used in testing process?

The tools which I have used during testing process are as follows.

Test Management Tools: JIRA, [TestLodge](#), Quality Center

Test Case Management Tools: [TestCaseLab](#)

Defect Tracking Tools: Bugzilla, MantisBT

Automation Tools: QTP/UFT, [Selenium](#), LoadRunner

GUI Tools: [Froglogic Squish](#)

Cross Browser Testing Tools: [CrossBrowserTesting](#), [BrowserStack](#)

Middle level Question:

1) How well do you handle ambiguity?

Test cases are not always straight-forward and QA Engineers will need to act on their personal judgement. They need to feel comfortable with ambiguity.

2) What is your mentality toward automating tests that are currently being done manually?

Automating test cases brings numerous benefits. Automation saves time and reduces human error. QA engineers should recognize the value in automation.

3) Explain how you distinguish a symptom vs. a cause when testing.

Often times within the QA process, test cases fail. But why are they failing? This can be tricky. A great QA engineer is able to provide exact reasons to the developer, rather than simply saying a test case “failed”.

4) Do you feel comfortable standing up to developers who disagree with your results?

In some situations, reporting failures can be a delicate process. Perhaps a developer spent a lot of time on some code which does not exactly meet specifications. QA engineers need to be able to stand up for what they know is right.

5) Are you willing to cut corners to save time?

The correct answer is NO. All test cases need to be run, and making assumptions frequently leads to issues down the road.

6) Can you explain the SDLC and Agile methodology?

QA engineers must understand their role and where it fits into the ecosystem.

7) Explain your attitude toward documentation. Do you believe more is better? Why or why not?

This is a trick question because more documentation is not always better. In fact, it can be detrimental. Documentation needs to be thorough, but still efficient as possible. If there is too much documentation, important details can be missed.

8) How do you go about learning a new product?

This is perhaps the biggest challenge of being a QA engineer. They need to be comfortable and patient with learning complex software. They should be prepared to ask a lot of questions.

9) How well do you work with others?

Many QA teams have team members who are from all corners of the world. QA engineers must be comfortable communicating with people of all backgrounds and all levels of language proficiency.

10) Do you see yourself as a perfectionist?

Great QA engineers are perfectionists. Their job is to ensure that all of the software they test meets or exceeds quality standards, and sloppy work is only going to lead to trouble further down the line.

11) How well do you work under pressure and with deadlines?

Testing usually comes at the end of the SDLC and testing can be viewed as a bottleneck, so being able to perform under tight deadlines and pressure is important.

12) What experience do you have with developing corner cases?

This question will help you to figure out if your tester is willing to dig for different test case scenarios that might not be as intuitive or clear up front

13) How do you keep up with current technology trends

Staying up to date on industry news and trends is important in order for you team to keep up with evolving technology and best practices changes. It also shows how much they like their job!

14) What motivates you?

There are several potential answers to this question. Company culture will play a large part in differentiating a good from bad answer. For instance, if teamwork is important within the product team, a candidate who is driven by internal competition might not be the best fit.

15) What are your ultimate career aspirations?

It is important to learn what your candidate hopes to be in the next year, or 3 years. If the candidate desires a career path you can't provide, you might be interviewing again sooner than you'd like.

16) How would a friend describe you in one sentence?

Sometimes this question pulls out unique personality qualities that you might not discover from traditional work focused questions.

17) What made you want to get into testing?

Figuring out how your job candidate achieved their current career status can uncover a lot around their ambitions and aspirations.

18) What do you think our company could do better?

This interview question helps you to uncover how much research the candidate did before meeting with you, and it demonstrates their ability to think on the spot.

19) Why do you want to work for us?

The best way to learn if your candidate is excited and passionate about this job opportunity.

20) What testing methods are you familiar with? Do you have a favorite?

It is important to hire a well rounded Test Engineer who is familiar with several different types of testing or find one who is willing to learn.

What are your favorite software tester interview questions to ask candidates?

1. In an application currently in production, one module of code is being modified. Is it necessary to re-test the whole application or is it enough to just test functionality associated with that module?

Vijay: Well, the answer is both. You will have to test the functionality of that module as well as the other modules. But you can differentiate on the stress to be given on the module to be tested.

I think this scenario will explain the answer to your question well.

If Module A is modified, Module B is depending on module A, and Module C is a general module independent of module A.

So, in this case, you will test the module A in depth to all test cases. Then your next stress will be on module B. Wait now what about module C? You will have to test this module as well but maybe with less stress because module C does not depend on module A but maybe depend on module B.

Again if you are a white box tester you probably know which modules will get affected and which modules should be tested. But as a black box tester, you will need to do regression testing as well.

2. What is the most challenging situation you had during testing?

3. What are you going to do if there is no Functional Spec or any documents related to the system and developer who wrote the code does not work in the company anymore, but you have a system and need to test?

1. How will you receive the project requirements?

A. The finalized SRS will be placed in a project repository; we will access it from there

2. What will you do with SRS?

A. SRS stands for software requirement specification. SRS is used to understand the project functionality from business and functional point of view.

3. What is FRS? How it different from SRS?

A.srs describes what client is expecting from the system. For example in case of Gmail SRS consists details like first page should be login, to access mail box user should be authenticated. FRS describes how above requirements will be developed .in FRS, the functionality in SRS will be written down in more technical terms. For example in case of Gmail FRS consists details like for login what fields should be present and what are valid inputs. This means FRS will have screen level details of the application.

Note: In many projects SRS itself will be designed at screen level details of the application.

4. Is the testing team involved in SRS preparation?

A. Business analyst prepare the SRS document by interacting with the client. However a senior testing team member can also be involved in requirements collections along with the development team and the business analyst team.

5. How does your requirements document look like?

A. It contains lots of use cases where each use case explains one or more functionalities

6. How will you understand the requirements?

A. If it is known domain by going through use cases i can understand the requirements. if i have some queries, i will discuss them with business analyst(BA) for clarifications. if it is new domain, first i will get domain training then i go through the use cases. If the project requirements are very confusing, then (BA) can also walk through each use case.

7. How do u understand functionality without screens?

A. We get wireframes in the usecases which helps a lot to understand the functionality

8. What is wireframe?

A. A diagram which stimulates the feel of the actual screen.

9. What is usecase?

A. Usecase explains the step by step procedure of how a particular functionality of s/w is used by the end user. usecase contains sections such as

- . usecase id
- . usecase name
- . decription
- . flow of events
- . alternative flow of events
- . pre,post conditions

10. Where you involved in writing the usecases?

A. I am aware of how usecases looks like and i can write if required. but i have never got an opportunity to write the usecases because these are prepared by requirements gathering team .Any how i have reviewed the usecases of certain functionalities and have given my inputs for betterment of the same.

11. What are the different sections present in SRS?

- A. overview
 - Scope
 - Features
 - User characteristics
 - Software requirements
 - Hardware requirements
 - Performance requirements
 - Use cases
 - Security and reliability requirements

12. How long do u spend on understanding SRS?

A. It depends on the familiarity of the domain and complexity of the project. if it is a familiar domain, we can understand around 25 pages of the documentation every day. For a new complex domain, we manage around 15 pages per day.

13. After understanding the SRS what do you do?

A. My lead asks for presentation of the functionalities i am assigned with if i am in a position to explain the functionalities clearly to the team, then i am considered as comfortable with functionalities.

14. Should you understand the whole project functionality or only the functionality assigned to you?

A. I should have a big picture of the whole project. In other words i should have an overview of the whole project and detailed screen and field level understanding of the assigned functionalities.

15. What are the different models generally followed in documenting requirements?

A. Two models are followed in documenting the requirements which are usecase model and paragraph model. in paragraph model business requirements are written like a paragraph which is old model. Now a days almost all companies follow the usecase model where the requirements are written by stating thier clear objectives and explained with the help of screen shots.

16. How big is your SRS?

A. You can answer anything like approx 250 pages. This question is asked just to cross check whether u have seen SRS or not

17. What will be the problem without SRS?

A. without srs we will not be able to understand the project features correctly. Hence we will not able to test the project in depth and deliver the best quality product.

18. What is SRS?

A. BRS is business requirement specification which is usually prepared before preparing an srs. This document gives a high-leval view of what is being required by the customer to meet business needs.

19. What is technical requirements specification?

A. This is also called as high-leval design, which consists of different modules present in the project.

20. What is user story?

A. user story is the method of documenting requirements in the agile model.

21. What is Review?

A. Review is a meeting in which a work product is verified by set if members (stake holders)

22. Explain the review process you follow in your organization?

A. The various phases of the review process followed in my organization are:

Planning:

- >selecting the personal for review
- >allocating roles
- >defining entry and exit criteria.

Kick-off:

- >Distributing documents
- >explaining the objectives
- >checking entry criteria, etc.

Individual preparation:

> in the phase, each of the participants will work before the review meeting and be ready with questions and comments.

Review Meeting:

- > Discussion among the review members by going through each line of the work product.
- > Logging comments
- > Making decision about the defect.

Rework:

- > Fixing defects found during the review, typically done by the author.

Follow-up:

- > checking the defects that have been addressed.
- > gathering metrics and checking the exit criteria.

23. What are the roles present in the review?**A. Manager:**

- > decides on execution of reviews.
- > allocates time in project schedules.
- > determines if the review objectives have met.

Moderator:

- > leads the review, including planning and running the meeting
- > follows-up after the meeting.

Author:

The author is the person who has created the item to be reviewed. the author may also be asked questions within the review.

Reviewer:

The reviewer are the attendees of the review who attempt to find errors in the item under review. they should come from different perspectives in order to provide a well balanced review of the item.

scribe:

The scribe or recorder is the person who is responsible for documenting issues raised during the process of the review meeting.

24. What is peer review?

A. Is a review of a software work product by colleagues?

25. What is the difference between static and dynamic testing?

A. Static testing means testing the project without executing the software and dynamic testing means testing the project by executing the software . i.e. by running the application and going through screens. To conduct dynamic testing you must use application screens and enter valid and invalid inputs and verify the application behavior. For static testing, we do not use any screens of application instead we use static techniques like review. During review, experts go through each line of the work products like requirement document, design document and identify mistakes in these documents. Any mistakes identified during this review are nothing but defects in the work product.

26. I want you to choose one among static and dynamic testing for your project. Which one will you choose and why?

A. static testing reduces the cost of fix and dynamic testing gives the complete confidence to release the product. According to me both are equally important and both of them contribute equally for the project success. so i prefer to have both. however if i have to choose one, i choose dynamic testing since i cannot let the project to be released until i see with my eyes that it is working.

27. out of formal and informal review, which one do you prefer?

A. In my view, both are important: informal review is fast and formal review is effective. we have to use both depending on the data we are reviewing . I prefer formal and informal review techniques as follows.

Formal Review:

- > Reviewing test case document created
- > Reviewing test plan document created
- > reviewing test scripts developed

Informal Review:

- > reviewing tests used for retesting
- > Reviewing minor changes in test case, test plan or test scripts.

28. How do you decide the review outcome?

A. The review outcome is decided by the moderator. i can share my views with him. for example in test cases review, the outcome decision it as follows

Review observation	Review outcome
--------------------	----------------

A. Most of the critical test cases are missed

B. Documentation standards are poor

Major changes are suggestedAccept after correction with another round of review

Minor changes are suggested Accept after correction without another round of review

No changes suggestedAccepts as it is

29. Explain what do you document during the review process?

A. we document page and line number of defect, origin of defect, severity of defect. We also document other information like work product ID, reviewers, etc

30. How much information you can review in one day?

A. per hour we review around 20 pages if it is documentation and 200 lines if it is code.

31. How do you say review was successful?

A. if every reviewer prepares well before the review and provides good comments for improvement of the work product, we can say that the review was successful.

32. What is code review?

A. code review is the process of reviewing the code written. code reviews are conducted for the code developed by the developer and also for the automation scripts developed by the automation engineer.

33. What is desk check?

A. This is an informal review where a colleagues comes to the desk/computer of the author and quickly goes through the work product along with the author and also shares comments while going through it.

34. What are the entry criteria for release?

- A. > system testing results must show that all requirements are completed and project is stable.
> Alpha and beta testing must be completed.
> All medium and above severity bugs must be fixed.
>The release package is available.
> The release CD label is ready.

35. What is the release process you follow?

A. In our organization, the release process is coordinated by a person called release manager. After successful beta testing, the release manger sends an email to all stake holders (development manager, test manager, documentation manager) for their

Approval for final release. The test manager further forwards the same mail to team members requesting their internal approval. Based on internal approval. The test manager can send approval to the release manager.

36. What is your involvement in the release process?

A. As a testing team member, i go through the defect tracking tool and check whether all the defects are fixed. In case any defects are not fixed i communicate the same to my test lead and test manager, sharing my opinion regarding each bug whether it must be fixed before release or it can be fixed after release. The test manager takes the final decision on whether to fix or not after discussion with the development manager.

I am further involved in preparing release notes, where i document known issues in my module along with the issues resolved from the previous release.

Exit criteria for release are:

- > All stake holders have approved for release.
- > The new package is deployed in production and users are happy about the release.
- > The code has been base lined in the configuration management.

37. What is a code Freeze?

A. code freeze means the code has been locked from further modifications from developers. After the code freeze the code should be changed by any developer. if at all any changes are required it should be only for very critical bugs after taking permission

from the top management of the project. code freezes are often employed in the final stages of development.

38. What are the entry and exit criteria for test execution?

Entry criteria:

-
- > coding should be completed

- > test cases should be ready and base lined
- > RTM should be updated
- > test data should be read and base lined
- > test environment/set up should be ready.
- > s/w tools should be ready and approved.

Exit criteria:

- > All test cases must be executed and passed
- > All defects identified must be fixed, retested and closed
- > Test execution summary report must be prepared

39. Explain different test execution strategies?

- A. There are 3 test execution strategies.
They are pass1, pass2, pass 3

Pass1 Test execution strategy:

In this execution model one execution cycle will be there. In this one execution cycle itself testers log defects and retest that defect. This is useful in stable, small with

2nd pass:

Development team releases new build claiming all the defects are fixed. Testing team retest all defects with adhoc regression. if new defects are found development team release new build and the life cycle is repeated until no new defects.

3rd pass:

Testing team runs full regression suite and this phase completes only when full regression is completed. This is good model for large, complex and critical projects. In case of getting large no of defects in pass -2 strategy, one may have to move to pass-3 strategy.

40. How do you know you have a build ready for testing?

- A. Frequency of build creation would vary from project to project .however below is the guideline to answer this question. In our project automatic build creation deployment happens on every x day. we receive a confirmation mail on every day morning about successful deployment along with URL for testing .please refer our build process FAQ'S for exactly how build deployment and release process.

41. How many test cases can you execute per day?

- A. it depends on the size and complexity of the test cases. Approximately i execute around 50 test cases per day which comes to 40 pages approx.

42. How do you run the test cases?

- A. I will perform each step in the test case on the application and compare the application behavior with expected result of the step. if it is same as expected result then step is passed else step is failed. if the password field shows * or some other special character while entering the password, it is called password masking, NOT password encryption. Encryption means converting the user entered characters into different characters before sending over the network. This can be checked with the help of

network sniffers.

example s/w for this is WIRESHARK. These s/ was capture every data packet travelling over the network including the IP address of source and destination computers. By analyzing these packets we can identify whether the password string is encrypted or not.

43. How do you check broken links?

A. Many tools are available for this. we use tools like menu.

44. What is test log?

A. It is a report of what tests have been executed and thier status like pass/ fail. it is also known as Test execution report.

45. Did you observe any application logs during the test execution?

A. yes. We do observe logs of the application server to check whether the server has thrown any runtime errors.

46. Do you run all regression tests for every bug fixed?

A. No, I didn't run regression test cases for every bug fixed. I run regression tests once for every build.

47. Do you run all regression tests every time ?

A. Depends. if we are sure that the fix might not affect other modules, we run regression tests specific to the module of the bugs fixed, else we run for the entire project.

48. In the modules you have worked on, are there any issues identified after release?

A. projects if i am supposed to write stubs or drivers in the current project i am confident that i can handle it.

49. When you fill the data in the application form, how do you ensure that the data is stored in the correct tables and columns?

A. we can write an SQL query to retrieve data from the data base and compare the query result with the data we have filled in the application forms.

50. What is test case?

A. Test case is a set of inputs, conditions and expected outcomes which a tester will determine whether an application is working correctly or not.

51. What fields a test case will have?

A. The following are the fields that a test case will usually have.....
test case id, description, precondition, step name, expected results, actual results and status.

52. Where do you write test case?

A. Depending on the project we can write test cases in an excel or in QC.

53. How do you know for which functionalities you should write test case?

A. My lead writes top level requirements in QC and assigns to each team member. we divide test requirements further into sub requirements. Then we identify test conditions for each sub requirement and create test cases. test cases are reviewed after that. Reviewed and approved test cases will go to ready state.

54. What is test scenario?

A. Test scenario is nothing but a functional scenario for which testing is to be conducted. It is also called as a test condition.

55. What is the difference between test scenario and test case?

A. Test scenario is a high level description of business requirements, which is latter decomposed in to a set of test cases. These test cases will be reviewed and approved by peers. we follow formal review process for approving test cases written for each functionality.

56. How do you know your test cases are completed?

A. We follow two step approaches to ensure that test cases are completed.
a. reviews-- it ensures that quality of the test cases is good.
b. Requirement traceability matrix ---it ensures that all requirements have been covered through test cases.

57. How do you find whether a test case is a good test case or bad test case?

A. A good test case is one which finds the bug or one which has a high probability of finding the bug. A good test case should be documented clearly, so that it can be executed by anyone without any difficulties and confusion.

58. What is the percentage of positive and negative test cases that you write?

A. Approx 30% positive and 70% negative

59. Do you update the test cases after receiving build based on the application screen?

A. During execution, if we feel any test case requires an update, we will do it with the approval of the team lead. but this work is very limited.

60. Explain one scenario where you were not able to write test cases for a given requirement?

A. Effort for a new domain by putting extra effort for through understanding of the domain.

61. What is the difference between a positive and negative test case?

A. A positive test caes checks whether the system does what it is suppose to do. I.e. to check that we got the desired result with a valid set of inputs.

ex: the user should login in to the system with a valid user name and password.

Negative test case: A negative test case checks whether the system will do what it is not supposed to do .i.e to check the system generates the correct error or warning messages with an invalid set of inputs.

ex: if the user entered the wrong user name or password, then the user should not login in to the system and appropriate error message should be shown.

62. What are the documents required for test analysis?

- A. 1. SRS/FRS
2. Use case
3. Architecture document

63. What is an entry criterion for test closure?

- A. Decision to stop testing

64. Who takes this decision?

- A. The Test Manager

65. What parameters do the test manager considers to take the decision to stop testing ?

- A. The important parameters a test manager looks into are
- > Whether all requirements have been developed or not.
 - > Whether all requirements have been covered through testing.
 - > Whether all requirements have been handled through fixed or differed status.

66. What are the exit criteria for test closure?

- A. > checking whether planned deliverables have been delivered.
> Finalizing and archiving test ware.
> Hand over of test ware for maintenance.
> analyzing lessons learned for improvement of test maturity.
> Testing sign off.

67. What is testware?

A. Test ware is Artifacts produced during the testing process. Test ware include test cases, test plan, automation scripts, test data, test environment set-up and clear up procedures and any additional software or utilities used in testing.

68. What is lessons learnt document?

- A. > no of test cases/scenarios blocked
> No of defects verified and their respective status.
> Weekly status reporting:
> Test case summary
> Issues found
> Issues resolved
> Critical issues which are still open and which requires immediate attention from the client side
> The report should also contain high plan for the next week.

69. What is the status can give to a test case?

A. Status are pass, fail, blocked, no run.

70. What is web server log?

A. Every time a web page is requested, the webserver automatically logs the following information.

- > The IP address of the visitor
- > Date and time of the request
- > The url of the requested file
- > The url, the visitor came from immediately before
- > The visitors web browser type and os

How Do you handled Non reproducible bug issues, while automating the test cases.

What are the key challenges you faced for Automating the things?

Did handled multiples frames if yes then how?

Q #1) What do you understand by web application?

Answer: Web application is a means to communicate and exchange information with the customers. Unlike any desktop applications which are executed by an operating system, a web application runs on a web server and is accessed by a web browser which acts as a client.

The best Example of a web application is 'Gmail'. In Gmail, the interaction done by an individual user is completely independent of the others. You can send and receive information through emails and also through attachments.

You can maintain documents in a drive, maintain spreadsheets in Google docs and includes much more such features which make a user realize that they have an environment which is customized to their specific identity.

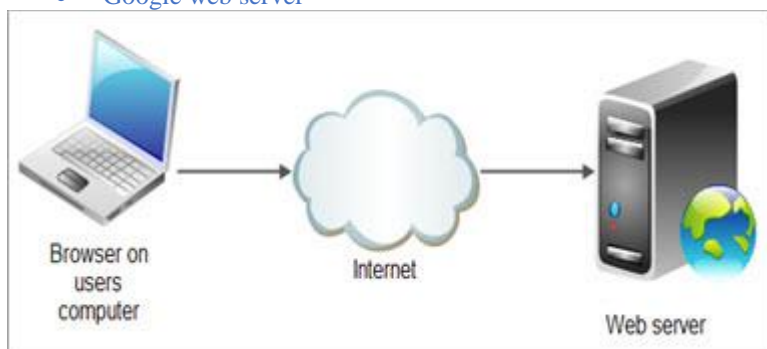
Q #2) Define a Web server.

Answer: Web server follows the client/server model where the program uses HTTP (Hypertext Transfer Protocol). In response to the request of a HTTP client, web server handles client and server side validation and delivers the web content in the form of web pages to the users.

The browsers we use, such as Safari, Chrome, Internet Explorer, Firefox etc., read the files stored on the web servers and bring the information to us in the form of images and texts with the means of internet. Any computer which hosts websites must have web servers.

Some of the leading web servers are:

- Apache
- Microsoft's Internet Information Server (IIS)
- Java web server
- Google web server



Q #3) Enlist some important test scenarios for testing a website.

Answer: There are many parameters that should be considered while deciding the important test scenarios for testing any website. Also, the type of website to be tested and its requirement specification plays an important role here.

Enlisted below are few important test scenarios that are applicable for testing any type of website:

- Test the GUI (Graphical User Interface) of the website for checking the consistency of the design elements and page layout.
- All page links and hyperlinks are checked for their redirection to the desired page.
- In case of presence of any forms or fields on the website, testing scenarios consist of testing with valid data, invalid data, testing with existing records as well as testing with empty records.
- Functionality testing as per the requirement specification is done.
- Performance of a website is tested under heavy loads to determine the web server response time and database query time.
- Compatibility testing is done to test the behavior of an application on a different browser and OS (operating system) combinations.
- Usability testing and Database testing is also performed as a part of test scenarios.

Q #4) What are the different configurations which have to be considered while testing a website?

Answer: Different configuration includes different browsers as well as an operating system on which a website is being tested. Browser plugins, text size, video resolution, color depth, browser setting option also come under consideration when we talk about configurations.

Different combinations of browsers and operating system are used to test the compatibility of the website. Usually, the latest and the last latest versions are included. Well, these versions are usually specified in the requirement documents.

Few important Browsers include:

- Internet Explorer
- Firefox
- Chrome
- Safari
- Opera

Few important Operating systems include:

- Windows
- UNIX
- LINUX
- MAC

Q #5) Is Web application testing different from Desktop Application testing? Explain how.

Answer: Yes, there is a lot of difference between web application and desktop application.

This can be explained with the help of the below-mentioned table:

	Web Application	Desktop Application
Definition	Web applications are the one which can run on any client machine having the internet connection without any installation of the execution file.	Desktop applications are one which are separately installed and executed on the personal computer.
Performance	User actions, feedback, statistics can be easily monitored as well as data updating in one place is reflected everywhere in web application.	User actions cannot be monitored as well as changes in data can be only reflected at the machine.
Connectivity	Web application can be accessed on any PC having internet connection using web browser where performance of the application depends on internet speed.	Desktop application can be only accessed on specific PC where application is installed.
Security Risks	Web application is more prone to security threats as applications can be accessed by anyone on the internet.	Desktop application is less prone to security threats where user can keep a check on security issues at the system level.

Web Application	Desktop Application
User data User data is saved and accessed remotely in case of web applications.	Data is stored, saved and accessed from the same machine on which application is installed.

Q #6) What is Intranet Application?

Answer: Intranet application is a kind of private application which is deployed and run on local LAN server and can only be accessed by the people within the organization. It uses local network to share information.

Example: Organization usually has an application which stores information about your attendance, holidays, upcoming celebrations within the organization or some important event or information that needs to be circulated within the organization.

Q #7) Explain the difference between Authorization and Authentication in Web testing.

Answer: The difference between Authorization and Authentication is explained in below table:

Authentication	Authorization
1 Authentication is the process with which the system identifies who the user is?	Authorization is the process with which system identifies what user is authorized to do?
2 Authentication determines the identity of the user.	Authorization decides the privileges given to the user i.e. whether the user can access or manipulate features of certain program.
3 There are different types of authentications, like password based, device based, etc.	There are two types of authorizations, like read only and read write both.
4 For example: Within an organization, each and every employee can login into an intranet application.	For example: Only account manager or person in accounts department can access account section.

Q #8) What are the types of Web testing security problems?

Answer: Few web security problems include:

- Denial of Service (DOS) attack
- Buffer overflow
- Directly passing internal URL through browser address
- Viewing other stats

Q #9) Define HTTP.

Answer: HTTP stands for Hypertext Transfer Protocol. HTTP is the data transfer protocol which defines how messages are formatted and transferred over World Wide Web. HTTP also determines the response of the actions performed by web servers and browsers.

For Example, when an URL is entered on the web browser, the HTTP command is sent to the web server which in turn fetches the requested web browser.

Q #10) Define HTTPS.

Answer: HTTPS stands for Hypertext Transfer Protocol Secure. This is basically HTTP over SSL (Secure Socket Layer) for security purposes. There is always chances of eavesdrop on data being transferred between a user and the web server when the website uses HTTP protocol.

Therefore, websites use secure way i.e. SSL encryption of data sent back and forth using HTTPS protocol.

Almost all the websites that require user log in uses HTTPS protocol. Few **Examples** are banking websites, e-commerce websites, etc.

Q #11) What are the common problems faced in Web testing?

Answer: Some of the common problems faced in web testing are enlisted below:

- Server Problem, which includes server down and server under maintenance problems.
- Database connection problem.
- Hardware and browser compatibility problems.
- Security related problems.
- Performance and load related problems.

- GUI (graphical user interface) related problems.

Q #12) What is Cookie testing?

Answer: Cookie is said to be a personalized user's identity or information which is required to communicate between different web pages as well as track user's navigation through the website pages. Whenever we access any website on any web browser, their respective cookie is written on the hard disk.

Cookies are used to track user sessions, displays ads, remember user's choice while accessing any website, remember and retrieve user's shopping cart, track the unique number of visitors, etc.

Suppose an e-commerce site is accessible in many countries like US, Canada, Australia and their testing is done in India. In that case, while testing the e-commerce site for different countries in India, at first respective countries cookies is set so that actual data like time zone etc., are accessed of that particular country.

Q #13) Define Client-side validation.

Answer: Client-side validation is the one which is basically done at the browser level where user's input is validated at the browser itself with no involvement of the server.

Let's understand it with the help of an Example.

Suppose a user is entering an incorrect email format while filling a form. The browser will then and there prompt an error message to correct it before moving on to next field. Thus every field is corrected before submitting the form.

The client-side validation is usually done by script language such as JavaScript, VBScript, HTML 5 attributes.

The two types of Client-side validation are:

- Field level validation
- Form level validation

Q #14) What do you understand by Server-side validation?

Answer: Server-side validation occurs where the validation and processing of user requests require the response from the server. To understand it more clearly, user's input is being sent to the server and validation is done using server-side scripting languages such as PHP, Asp.NET, etc.

After the validation process, feedback is sent back to the client in the form of dynamically generated web page.

When compared to the Client Side validation process, Server side validation process is more secure because here application is protected against malicious attacks and users can easily bypass client-side scripting language.

Q #15) Differentiate between Static and Dynamic website.

Answer: Difference between static and dynamic websites are as follows:

Static Website	Dynamic website
Static websites are the one which gives out information only and there is no sort of interaction between the user and the website.	Dynamic websites are the one where user interaction is possible between the website and user along with imparting information.
Static websites are cheapest to develop and host.	Dynamic websites are more expensive to develop as well as their hosting cost is also more.
Static websites are easily loaded on client browser because of its fixed content and no database connectivity.	Dynamic websites usually take the time to load on client browser because contents to display are dynamically created and retrieved using database queries.
Static websites can be created from HTML, CSS and does not require any server application language.	Dynamic websites require server application language like ASP.NET, JSP, PHP to run the application on the server and display the output on the webpage.

Static Website	Dynamic website
----------------	-----------------

Change in the content of the page of any static website; require being uploaded on server many times.

Dynamic website provides facilities to change the page content using server application.

Q #16) What do you understand by Client-Server testing?

Answer: Client-server application is the one where the application itself is loaded or installed on a server whereas application EXE file is loaded on all client machines. This environment is usually used in Intranet networks.

Following tests are performed on a Client-server application:

- GUI testing on both client and server systems.
- Client-server interaction.
- The functionality of an application.
- Load and performance testing.
- Compatibility testing.

All the test cases and test scenarios used in client-server application testing is derived from the tester's experience and requirement specifications.

Q #17) Enlist HTTP response codes that are returned by the server.

Answer: HTTP response codes are enlisted below:

- 2xx – This means 'Success'
- 3xx- This means 'Redirection'
- 4xx- This means 'Application error'
- 5xx- This means 'Server error'

Q #18) What is the role of Usability testing in Web testing?

Answer: In web testing, Usability testing plays an important role. It is well known that usability testing is the means to determine the ease with which an end user can easily access the application with or without having any programming language knowledge.

In terms of web testing, usability testing comprises of the following:

- To check whether the website is user-friendly?
- Is the end user able to easily navigate within the application?
- Presence of any issues or ambiguity which can hinder the user experience.
- Check how quickly the user is able to complete the task within the application.

Q #19) What are the available environments on Web?

Answer: The different types of the environment on Web are:

- Intranet (Local Network)
- Internet (Wide Area Network)
- Extranet (Private network over the internet)

Q #20) What are the test case formats in case of a Static website and Dynamic website?

Answer: The following test case formats will be used in case of Static websites:

- Front-end test cases
- Navigation test cases

The following test case formats will be used in case of Dynamic websites:

- Front-end test cases
- Back-end test cases
- Navigation test cases
- Field validation test cases
- Security test cases, etc.

Q #21) Enlist some sub-classes of HTTP response objects?

Answer: Write, Flush, tell, etc are few HTTP response objects.

The sub-classes of HTTP response are:

- HttpResponseRedirect
- HttpResponseRedirectPermanentRedirect
- HttpResponseRedirectBadRequest

- `HttpResponseNotFound`

Q #22) Enlist some Web Testing Tools.

Answer: Few Web testing tools are enlisted below:

- eggplant functional
- Selenium
- SOA test
- JMeter
- iMacros, etc.

Q #23) Give some examples of web applications that are used in our day to day life.

Answer: Few Examples include:

- Web portals like eBay, Amazon, Flipkart etc.
- Banking applications like ICICI, Yes Bank, HDFC, Kotak Mahindra etc.
- Email service providers like Gmail, Yahoo, Hotmail etc.
- Social Networks like Facebook, Twitter, LinkedIn etc.
- Discussion and Information forums like www.Softwaretestinghelp.com

Q #24) What is a Proxy server?

Answer: Proxy server is a server which acts as an intermediary or is the one that lies between the client and the main server.

The communication between the main server and client-server is done through a proxy server as the client request of any connection, file, resources from the main server is sent through a proxy server and again the response from the main server or local cached memory to client-server is done through the proxy server.

Some of the most common proxy servers based on their purpose and functionality are listed below:

- Transparent proxy
- Web proxy
- Anonymous proxy
- Distorting proxy
- High anonymity proxy

The proxy server is basically used for the following purposes:

- To improve the performance of web response.
- In case of presence of a document in a cache memory, the response is directly sent to the client.
- Proxy server filters web page content in the form of web proxies.
- A proxy server is also used to block offensive web content to be accessed by the user especially in an organization, school, and college.
- Web proxies prevent the attack of computer viruses and malware.

Q #25) What is Database server?

Answer: A Database server can be defined as a server that refers to the back-end system of a database application that provides database services such as accessing and retrieving data from the database.

Database server uses client/server architecture where the data can be accessed either through the database server by a “front end” which runs and displays data on user’s machine or “back end” which runs on the database server itself.

A database server is like a data warehouse and also holds on Database Management System (DBMS).

Few More Basic Software Testing Interview Questions

Q. What is Dynamic Testing?

Ans. It is the testing done by executing the code or program with various input values and later on the output is verified.

Q. What is GUI Testing?

Ans. GUI or Graphical user interface testing is the process of testing the software user interface against the provided requirements/mockups/HTML designs etc.,

Q. What is Formal Testing?

Ans. Software verification carried out by following a test plan, testing procedures and proper documentation with an approval from the customer is termed as Formal Testing.

Q. What is Risk Based Testing?

Ans. Identifying the critical functionality in the system and then deciding the orders in which these functionalities are to be tested and applying testing after that is termed as Risk-based Testing.

Q. What is Early Testing?

Ans. Conducting testing as soon as possible in the development lifecycle to find defects at early stages of **STLC**. Early testing is helpful to reduce the cost of fixing defects at the later stages of **STLC**.

Q. What is Exhaustive Testing?

Ans. Testing functionality with all valid, invalid inputs and pre-conditions is called Exhaustive testing.

Q. What is Defect Clustering?

Ans. Any small module or functionality may contain a number of defects – concentrating more on testing these functionalities is known as Defect Clustering.

Q. What is Pesticide Paradox?

Ans. If prepared test cases are not finding defects, add/revise test cases to find more defects, this is known as Pesticide Paradox.

Q. What is Static Testing?

Ans. Manual verification of the code without executing the program is called as Static Testing. In this process, the issues are identified in the code by checking code, requirement and design documents.

Q. What is Positive Testing?

Ans. It is the Testing which is conducted on the application to determine if the system works properly. Basically known as “test to pass” approach.

Q. What is Negative Testing?

Ans. Testing Software with a negative approach to check if the system is not “showing error when not supposed to” and “not showing error when supposed to” is termed as Negative Testing.

Q. What is an End-to-End Testing?

Ans. Testing the overall functionality of the system including the data integration among all the modules is called End-to-End Testing.

Q. What is Exploratory Testing?

Ans. Exploring the application, understanding its functionalities, adding (or) modifying the existing test cases for better testing is called Exploratory testing.

Q. What is Monkey Testing?

Ans. Testing conducted on an application without any plan and carried out randomly with the tests to find any system crash with an intention of finding tricky defects is called Monkey Testing.

Q. What is Non-Functional Testing?

Ans. Validating various non-functional aspects of the system such as user interfaces, user-friendliness, security, compatibility, Load, Stress, and Performance etc., is called Non-Functional testing.

Q. What is Usability Testing?

Ans. Checking how easily the end users are able to understand and operate the application is called Usability Testing.

Q. What is Security Testing?

Ans. Validating whether all security conditions are properly implemented in the software (or) not is called Security testing.

Q. What is Performance Testing?

Ans. The process of measuring various efficiency characteristics of a system such as response time, load stress transactions per minutes, transaction mix etc., is termed as Performance Testing.

Q. What is Load Testing?

Ans. Analyzing both the functional and performance behavior of an application under various conditions is called Load Testing.

Q. What is Stress Testing?

Ans. Checking the application behavior under stress conditions
(or)

Reducing the system resources and keeping the load as constant and checking how the application is behaving is called Stress Testing.

Q. What is Process?

Ans. A process is a set of practices performed to achieve a given purpose; it may include tools, methods, materials or people.

Q. What is Software Configuration Management?

Ans. The process of identifying, Organizing and controlling changes to the Software development and maintenance.

(or)

It is a methodology to control and manage a software development project.

Q. What is a **Testing Process** / LifeCycle?

Ans. It includes the below factors:

Writing a Test Plan

Test Scenarios

Test Cases

Executing the Test Cases

Test Results

Defect Reporting

Defect Tracking

Defect Closing

Test Release

Q. What is full form of **CMMI**?

Ans. Capability Maturity Model Integration

Q. What is a **Code Walk Through**?

Ans. An informal analysis of the program source code to find the defects and verify the coding techniques is termed so.

Q. What is **Unit Level Testing**?

Ans. Testing of single programs, modules or unit of code is termed as Unit Level Testing.

Q. What is **Integration Level Testing**?

Ans. Testing of related programs, Modules (or) Unit of code.

(or)

Partitions of the system which is ready for testing with other partitions of the system is termed so.

Q. What is **System Level Testing**?

Ans. Testing of the entire computer system across all the modules is termed so. This kind of testing can include Functional as well as Structural Testing.

Q. What is **Alpha Testing**?

Ans. Testing of a whole computer system before rolling out to the UAT is termed so.

Q. What is **User Acceptance Testing (UAT)**?

Ans. Testing of a computer system by the client to verify if it adhered to the provided requirements.

Q. What is a **Test Plan**?

Ans. It is a document describing the scope, approach, resources, and schedule of testing activities. It identifies test items, features to be tested, testing tasks, who will do each task, and any risks requiring contingency planning.

Q. What is a **Test Scenario**?

Ans. Identifying all the possible areas to be tested (or) what is to be tested is termed so.

Q. What is **ECP (Equivalence Class Partition)**?

Ans. It is a method for deriving test cases.

Q. What is a **Defect**?

Ans. Any flaw or imperfection in a software work product is termed as a Defect.

(or)

When the expected result does not match with the application actual result, it is termed so.

Q. What is **Severity**?

Ans. It defines the importance of the defect from the functional point of view i.e. how critical is a defect with respect to the application.

Q. What is **Priority**?

Ans. It indicates the importance or urgency of fixing a defect

Q. What is **Re-Testing**?

Ans. Re-testing the application means verifying whether the defects have been fixed or not.

Q. What is **Regression Testing**?

Ans. Verifying an existing functional and non-functional area after making changes to the part of a software or addition of new features is termed so.

Q. What is **Recovery Testing**?

Ans. Checking whether the system is able to handle some unexpected or unpredictable situations is called Recovery Testing.

Q. What is **Globalization Testing**?

Ans. It is the process of verifying whether a software can be run independently of its geographical and cultural environment. Checking if the application is having features of setting and changing language, date, format, and currency if it is designed for global users.

Q. What is Localization Testing?

Ans. Verifying globalized application for a particular locality of users, under cultural and geographical conditions is termed so.

Q. What is Installation Testing?

Ans. Checking whether we are able to install a software successfully (or) not, as per the guidelines given in the installation document is called Installation Testing.

Q. What is Un-Installation Testing?

Ans. Checking whether we are able to uninstall the software from the system successfully (or) not is called Un-Installation Testing

Q. What is Compatibility Testing?

Ans. Checking whether the application is compatible with different software and hardware environment or not is called Compatibility Testing.

Q. What is a Test Strategy?

Ans. It is a part of a test plan describing how testing is carried out for the project and what testing types need to be performed on the application.

Q. What is a Test Case?

Ans. A Test case is a set of pre-conditional steps to be followed with input data and expected behavior to validate the functionality of a system.

Q. What is Business Validation Test Case?

Ans. A test case which is prepared to check the business condition or a business requirement is called Business Validation test case.

Q. What is a Good Test Case?

Ans. A Test case that has the high priority of catching defects is called a **Good Test Case**.

Q. What is Use Case Testing?

Ans. Validating a software to confirm whether it is developed as per the use cases or not is called Use Case testing.

Q. What is a Defect Age?

Ans. The time gap between the date of detection & the date of closure of a defect is termed so.

Q. What is Showstopper Defect?

Ans. A defect which is not permitting to continue further with testing is called **Showstopper Defect**.

Q. What is a Test Closure?

Ans. It is the last phase of the STLC, where the management prepares various test summary reports that explain the complete statistics of the project based on the testing carried out.

Q. What is Bucket Testing?

Ans. Bucket testing is also known as A/B testing. It is mostly used to study the impact of various product designs in the website metrics. Two simultaneous versions are run on a single or a set of web pages to measure the difference in click rates, interface, and traffic.

Q. What is meant by Entry Criteria and Exit Criteria in Software Testing?

Ans. **Entry Criteria** is the process that must be present when a system begins, like,
SRS – Software
FRS

Use Case

Test Case

Test Plan

Exit criteria ensure whether the testing is completed and the application is ready for release, like,
Test Summary Report,
Metrics

Defect Analysis Report.

Q. What is Concurrency Testing?

Ans. This is a multiple user testing to access the application at the same time to verify the effect on code, module or DB and it is mainly used to identify the locking and deadlocking situations in the code.

Q. What is Web Application Testing?

Ans. Web application testing is done on a website to check – load, performance, security, Functionality, Interface, Compatibility and other usability-related issues.

Q. What is Unit Testing?

Ans. Unit testing is done to check whether the individual modules of the source code are working properly or not.

Q. What is Interface Testing?

Ans. Interface testing is done to check whether the individual modules are communicating properly as per the specifications or not. Interface testing is mostly used to test the user interface of GUI applications.

Q. What is Gamma Testing?

Ans. Gamma testing is done when the software is ready for release with the specified requirements, this testing is done directly by skipping all the in-house testing activities.

Q. What is Test Harness?

Ans. Test Harness is configuring a set of tools and test data to test an application under various conditions, which involves monitoring the output with the expected output for correctness.

The benefits of Testing Harness are: Productivity increase due to process automation and increase in the product quality

Q. What is Scalability Testing?

Ans. It is used to check whether the functionality and performance of a system are capable to meet the volume and size changes as per the requirements.

Scalability testing is done using load test by changing various software, hardware configurations, and testing environment.

Q. What is Fuzz Testing?

Ans. Fuzz testing is a black box testing technique which uses a random bad data to attack a program to check if anything breaks in the application.

Q. What is Difference between QA, QC, and Testing?

Ans. QA?

It is process oriented and its Aim is to prevent the defects in an application.

QC?

QC is product oriented and it is a Set of activities used to evaluate a developed work product

Testing?

Executing and verifying an application with the intention of finding defects.

Q. What is Data Driven Testing?

Ans. It is an Automation testing process in which an application is tested with multiple sets of data with different preconditions as an input to the script.

1. What is Software Testing?

According to ANSI/IEEE 1059 standard – 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. [Click here for more details.](#)

2. 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. So the defects don't arise when the Software Application is being developed.

Quality Control: Quality Control involves in product-oriented activities. It executes the program or code to identify the defects in the Software Application.

Must Read: [Software QA Interview Questions](#)

3. What is Verification in software testing?

Verification is the process, to ensure that whether we are building the product right i.e., to verify the requirements which we have and to verify whether we are developing the product accordingly or not. Activities involved here are Inspections, Reviews, Walk-throughs. [Click here for more details.](#)

4. What is Validation in software testing?

Validation is the process, whether we are building the right product i.e., to validate the product which we have developed is right or not. Activities involved in this is Testing the software application. [Click here for more details.](#)

5. What is Static Testing?

Static Testing involves in reviewing the documents to identify the defects in the early stages of SDLC.

6. What is Dynamic Testing?

Dynamic testing involves in the execution of code. It validates the output with the expected outcome.

7. What is White Box Testing?

White Box Testing is also called as Glass Box, Clear Box, and Structural Testing. It is based on applications internal code structure. In white-box testing, an internal perspective of the system, as well as programming skills, are used to design test cases. This testing usually was done at the unit level. [Click here for more details.](#)

8. What is Black Box Testing?

Black Box Testing is a [software testing](#) method in which testers evaluate the functionality of the software under test 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. [Click here for more details.](#)

9. What is Grey Box Testing?

Grey box is the combination of both White Box and Black Box Testing. The tester who works on this type of testing needs to have access to design documents. This helps to create better test cases in this process.

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.

11. What is Test Strategy?

Test Strategy is a high-level document (static document) and usually developed by project manager. It is a document which captures the approach on how we go about testing the product and achieve the goals. It is normally derived from the Business Requirement Specification (BRS). Documents like Test Plan are prepared by keeping this document as a base. [Click here for more details.](#)

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, or Use Case documents for all future activities of the project. 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

[Click here for more details.](#)

13. What is Test Suite?

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

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.

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. [Click here for more details.](#)

16. What is Test Bed?

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

17. What is Test Environment?

Test Environment is the combination of hardware and software on which Test Team performs testing.

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

18. What is Test Data?

Test data is the data that is used by the testers to run the test cases. Whilst running the test cases, testers need to enter some input data. To do so, testers prepare test data. It can be prepared manually and also by using tools.

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.

19. What is Test Harness?

A test harness is the collection of software and test data configured to test a program unit by running it under varying conditions which involves monitoring the output with expected output.

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)

[Click here for more details.](#)

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

[Click here for more details.](#)

24. What is System Testing?

Testing the fully integrated application to evaluate the system's compliance with its specified requirements is called System Testing AKA End to End testing. Verifying the completed system to ensure that the application works as intended or not.

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

Manual Testing Interview Questions – 26-50:

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.

28. What is End-To-End Testing?

Refer System Testing.

29. What is Functional Testing?

In simple words, what the system actually does is functional testing. To verify that each function of the software application behaves as specified in the requirement document. 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 and the testers need not concern about the source code of the application.

30. What is Non-Functional Testing?

In simple words, how well the system performs is non-functionality testing. Non-functional testing refers to various aspects of the software such as performance, load, stress, scalability, security, compatibility etc., Main focus is to improve the user experience on how fast the system responds to a request.

31. What is Acceptance Testing?

It is also known as pre-production testing. This is done by the end users along with the testers to validate the functionality of the application. After successful acceptance testing. Formal testing conducted to determine whether an application is developed as per the requirement. It allows the customer to accept or reject the application. 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. Sometimes alpha testing is done by the client or outsourcing team with the presence of developers or 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.

35. 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". It helps not to waste the testing time to simply testing the whole application when the key features don't work or the key bugs have not been fixed yet.

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". At times due to release time constraints rigorous regression testing can't be done to the build, sanity testing does that part by checking main functionalities.

37. What is Retesting?

To ensure that the defects which were found and posted in the earlier build were fixed or not in the current build. Say, Build 1.0 was released. Test team found some defects (Defect Id 1.0.1, 1.0.2) and posted. Build 1.1 was released, now testing the defects 1.0.1 and 1.0.2 in this build is retesting.

38. What is Regression Testing?

Repeated testing of an already tested program, after modification, to discover any defects introduced or uncovered as a result of the changes in the software being tested or in another related or unrelated software components.

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?

Globalization is a process of designing a software application so that it can be adapted to various languages and regions without any changes.

42. What is Internationalization Testing (I18N Testing)?

Refer Globalization Testing.

43. What is Localization Testing (L10N Testing)?

Localization is a process of adapting globalization software for a specific region or language by adding local specific components.

44. What is Installation Testing?

It is to check whether the application is successfully installed and it is working as expected after installation.

45. What is Formal Testing?

It is a process where the testers test the application by having pre-planned procedures and proper documentation.

46. What is Risk Based Testing?

Identify the modules or functionalities which are most likely cause failures and then testing those functionalities.

47. What is Compatibility Testing?

It is to deploy and 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 verify whether the application is user-friendly or not and was comfortably used by an end user or not. The main focus in this testing is to check whether the end user can understand and operate the application easily or not. An application should be self-exploratory and must not require training to operate it.

Manual Testing Interview Questions – 51-75:

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, throughput, and resource-utilization levels that meet the performance objectives for the project or product.

54. What is Load Testing?

It is to verify that the system/application can handle the expected number of transactions and to verify the system/application behavior under both normal and peak load conditions.

55. What is Volume Testing?

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

56. What is Stress Testing?

It is to verify the behavior of the system once the load increases more than its design expectations.

57. What is Scalability Testing?

Scalability testing is a type of non-functional testing. It is to determine how the application under test 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. By inputting massive amount of

random data to the system in an attempt to make it crash to identify if anything breaks in the application.

60. What is Adhoc Testing?

Ad-hoc testing is quite opposite to the formal testing. It is an informal testing type. In Adhoc testing, testers randomly test the application without following any documents and test design techniques. This testing is primarily performed if the knowledge of testers in the application under test is very high. Testers randomly test the application without any test cases or any business requirement document.

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?

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

64. What is A/B Testing?

Refer Bucket Testing.

65. What is Split Testing?

Refer Bucket Testing.

66. What are the principles of Software Testing?

1. Testing shows presence of defects
2. Exhaustive testing is impossible

3. Early testing
4. Defect clustering
5. Pesticide Paradox
6. Testing is context depending
7. Absence of error fallacy

[Click here for more details.](#)

67. What is Exhaustive Testing?

Testing all the functionalities using all valid and invalid inputs and preconditions is known as Exhaustive testing.

68. What is Early Testing?

Defects detected in early phases of SDLC are less expensive to fix. So conducting early testing reduces the cost of fixing defects.

69. What is Defect clustering?

Defect clustering in software testing means that a small module or functionality contains most of the bugs or it has the most operational failures.

70. What is Pesticide Paradox?

Pesticide Paradox in software testing is the process of repeating the same test cases, again and again, eventually, the same test cases will no longer find new bugs. So to overcome this Pesticide Paradox, it is necessary to review the test cases regularly and add or update them to find more defects.

71. What is Walk Through?

A walkthrough is an informal meeting conducted to learn, gain understanding, and find defects. The author leads the meeting and clarifies the queries raised by the peers in the meeting.

72. What is Inspection?

Inspection is a formal meeting lead by a trained moderator, certainly not by the author. The document under inspection is prepared and checked thoroughly by the reviewers before the meeting. In the inspection meeting, the defects found are logged and shared with the author for appropriate actions. Post inspection, a formal follow-up process is used to ensure a timely and corrective action.

73. Who are all involved in an inspection meeting?

Author, Moderator, Reviewer(s), Scribe/Recorder and Manager.

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. [Click here for more details.](#)

75. What is a Bug?

If testers find any mismatch in the application/system in testing phase then they call it as Bug. [Click here for more details.](#)

Software Testing Interview Questions – 76-100:

76. What is an Error?

We can't compile or run a program due to a coding mistake in a program. If a developer unable to successfully compile or run a program then they call it as an error. [Click here for more details.](#)

77. What is a Failure?

Once the product is deployed and customers find any issues then they call the product as a failure product. After release, if an end user finds an issue then that particular issue is called as a failure. [Click here for more details.](#)

78. What is Bug Severity?

Bug/Defect severity can be defined as the impact of the bug on customer's business. It can be Critical, Major or Minor. In simple words, how much effect will be there on the system because of a particular defect. [Click here for more details.](#)

79. What is Bug Priority?

Defect priority can be defined as how soon the defect should be fixed. It gives the order in which a defect should be resolved. Developers decide which defect they should take up next based on the priority. It can be High, Medium or Low. Most of the times the priority status is set based on the customer requirement. [Click here for more details.](#)

80. Tell some examples of Bug Severity and Bug Priority?

High Priority & High Severity: Submit button is not working on a login page and customers are unable to login to the application

Low Priority & High Severity: Crash in some functionality which is going to deliver after couple of releases

High Priority & Low Severity: Spelling mistake of a company name on the homepage

Low Priority & Low Severity: FAQ page takes a long time to load

[Click here for more details.](#)

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?

Bug life cycle is also known as **Defect life cycle**. In Software Development process, the bug has a life cycle. The bug should go through the life cycle to be closed. Bug life cycle varies depends upon the tools (QC, JIRA etc.,) used and the process followed in the organization. [Click here for more details.](#)

83. What is Bug Leakage?

A bug which is actually missed by the testing team while testing and the build was released to the Production. If now that bug (which was 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. These known bugs should be included in the release note.

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

Assume, a tester found a bug and reported it on 1 Jan 2016 and it was successfully fixed on 5 Jan 2016. So the defect age is 5 days.

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. It helps in the process of estimating the tester skills of finding bugs and also to know the ability of the application (how well the application is working when it has errors.)

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.

88. What is HotFix?

A bug which needs to handle as a high priority bug and fix it immediately.

89. What is Boundary Value Analysis?

Boundary value analysis (BVA) is based on testing the boundary values of valid and invalid partitions. The Behavior at the edge of each 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. Every partition has its maximum and minimum values and these maximum and minimum values are the boundary values of a partition. A boundary value for a valid partition is a valid boundary value. Similarly, a boundary value for an invalid partition is an invalid boundary value. [Click here for more details.](#)

90. What is Equivalence Class Partition?

Equivalence Partitioning is also known as Equivalence Class Partitioning. In equivalence partitioning, inputs to the software or system are divided into groups that are expected to exhibit similar behavior, so they are likely to be proposed in the same way. Hence selecting one input from each group to design the test cases. [Click here for more details.](#)

91. What is Decision Table testing?

Decision Table is aka Cause-Effect Table. This test technique is appropriate for functionalities which has logical relationships between inputs (if-else logic). In Decision table technique, we deal with combinations of inputs. To identify the test cases with decision table, we consider conditions and actions. We take conditions as inputs and actions as outputs. [Click here for more details.](#)

92. What is State Transition?

Using state transition testing, we pick test cases from an application where we need to test different system transitions. We can apply this when an application gives a different output for the same input, depending on what has happened in the earlier state. [Click here for more details.](#)

93. What is an entry criteria?

The prerequisites that must be achieved before commencing the testing process. [Click here for more details.](#)

94. What is an exit criteria?

The conditions that must be met before testing should be concluded. [Click here for more details.](#)

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.

[Click here for more details.](#)

96. What are the different available models of SDLC?

1. [Waterfall](#)
2. [Spiral](#)
3. [V Model](#)
4. Prototype
5. [Agile](#)

97. What is STLC?

STLC (Software Testing Life Cycle) identifies what test activities to carry out and when to accomplish those test activities. Even though testing differs between Organizations, there is a testing life cycle. [Click here for more details.](#)

98. What is RTM?

Requirements Traceability Matrix (RTM) is used to trace the requirements to the tests that are needed to verify whether the requirements are fulfilled. Requirement Traceability Matrix AKA Traceability Matrix or Cross Reference Matrix. [Click here for more details.](#)

99. What is Test Metrics?

Software test metrics is to monitor and control process and product. It helps to drive the project towards our planned goals without deviation. Metrics answer different questions. It's important to decide what questions you want answers to. [Click here for more details.](#)

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

Thanks, Good Luck, 😊

Regards,
Prasad Hiwale