Manual Testing

Interview Question and Answer

Freshers

Q1 What is Software Testing?

Software testing is a process of executing a program or application with the intent of finding a software bug.

Q1.1 What do you understand by software testing?

Software testing is a validation process that confirms that a system works as per the business requirements. It qualifies a system on various aspects such as usability, accuracy, completeness, efficiency, etc. ANSI/IEEE 1059 is the global standard that defines the basic principles of testing.

Q2 Why Software Testing is necessory?

Software Testing is necessary because we all make mistakes. Some of those mistakes are unimportant, but some of them are expensive or dangerous. We need to check everything and anything we produce because things can always go wrong, Humans make mistakes all the time.

- Software testing is really required to point out the **defects** and **errors** that were made during the **development phases**.
- It's essential since it makes sure that the customer finds the organization reliable and their satisfaction in the application is maintained.
- It is very important to ensure the Quality of the product. Quality product delivered to the customers helps in gaining their confidence.
- Testing is necessary in order to provide the facilities to the customers like the delivery of high quality product or software application which requires lower maintenance cost and hence results into more accurate, consistent and reliable results.
- Testing is required for an effective performance of software application or product.

- It’s important to ensure that the application should not result into any **failures** because it can be very expensive in the future or in the later stages of the development.
- It’s required to stay in business.

Q3 Software Testing objectives and purpose?

- Finding defects which may get created by the programmer while developing the software.
- Gaining confidence in and providing information about the level of quality.
- To prevent defects.
- To make sure that the end result meets the business and user requirements.
- To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
- To gain the confidence of the customers by providing them a quality product.

Q4 What are the two main categories of software testing?

Software testing is a huge domain but it can be broadly categorized into two areas such as :

- Manual Testing – This is the oldest type of software testing where the testers manually execute test cases without using any test automation tools. It means the software application is tested manually by QA testers.
- Automation Testing – This is the process of using the assistance of tools, scripts, and software to perform test cases by repeating pre-defined actions. Test Automation focuses on replacing manual human activity with systems or devices that enhance efficiency.

Q5 Why do we say testing is context dependent?

Testing is very much dependent upon context, as the test 'subject' will require a specific type of approach; for example, an application designed for use in the cruise industry will be quite different to one in the insurance industry.

Q6 What different types of manual testing are there?

Different types of manual testing are;

- Black Box Testing
- White Box Testing
- Unit Testing
- System Testing
- Integration Testing
- Acceptance Testing

Q7 What is Black Box Testing?

Black box testing involves testing a system with no prior knowledge of its internal workings. A tester provides an input, and observes the output generated by the system under test. This makes it possible to identify how the system responds to expected and unexpected user actions, its response time, usability issues and reliability issues.

Black box testing is a powerful testing technique because it exercises a system end-to-end. Just like end-users "don't care" how a system is coded or architected, and expect to receive an appropriate response to their requests, a tester can simulate user activity and see if the system delivers on its promises. Along the way, a black box test evaluates all relevant subsystems, including UI/UX, web server or application server, database, dependencies, and integrated systems.

Q8 What is White Box Testing?

White Box Testing is a testing technique in which software's internal structure, design, and coding are tested to verify input-output flow and improve design, usability, and security. In white box testing, code is visible to testers, so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing, and Glass box testing.

Q9 What is Unit Testing?

Unit testing involves the testing of each unit or an individual component of the software application. It is the first level of functional testing. The aim behind unit testing is to validate unit components with its performance.

A unit is a single testable part of a software system and tested during the development phase of the application software.

The purpose of unit testing is to test the correctness of isolated code. A unit component is an individual function or code of the application. White box testing approach used for unit testing and usually done by the developers.

Whenever the application is ready and given to the Test engineer, he/she will start checking every component of the module or module of the application independently or one by one, and this process is known as **Unit testing** or **components testing**.

Q11 What is Integration Testing?

Integration Testing is defined as a type of testing where software modules are integrated logically and tested as a group. A typical software project consists of multiple software modules, coded by different programmers. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated
Integration Testing focuses on checking data communication amongst these modules.

Q12 What is Acceptance Testing?

User acceptance testing (UAT), also called application testing or end-user testing, is a phase of software development in which the software is tested in the real world by its intended audience.

Q13 What are the Black Box Testing Techniques?

- **Boundary Value Analysis**
- **Equivalence partitioning**
- **State Transition Testing**
- **Decision Table Testing**
- **Graph-Based Testing**
- **Error Guessing Technique**

Q14 What is Functional testing?

FUNCTIONAL TESTING is a type of software testing that validates the software system against the functional requirements/specifications. The purpose of Functional tests is to test each function of the software application, by providing appropriate input, verifying the output against the Functional requirements.

Q15 What is Non-Functional testing?

Non-Functional Testing is defined as a type of Software testing to check non-functional aspects (performance, usability, reliability, etc) of a software application. It is designed to test the readiness of a system as per nonfunctional parameters which are never addressed by functional testing.
An excellent example of a non-functional test would be to check how many people can simultaneously login into a software.

Q16 What is regression testing?

Regression testing is **a software testing practice that ensures an application still functions as expected after any code changes, updates, or improvements**. Regression testing is responsible for the overall stability and functionality of the existing features.

Q17 What is Smoke and Sanity Testing?

:- **Smoke Testing** is performed to ascertain that the critical functionalities of the program are working fine.

:- **Sanity Testing** is done to check the new functionality/bugs have been fixed.

Q18 What is UI Testing?

UI testing is a testing type that helps testers ensure that all the fields, labels, buttons, and other items on the screen function as desired. It involves checking screens with controls, like toolbars, colors, fonts, sizes, icons, and others, and how these respond to the user input.

Q19 What is Responsive Testing?

Responsive testing involves how a website or web application looks and behaves on different devices, screen sizes, and resolutions. The goal of responsive testing is to ensure that the website or web application can be used effectively on various devices, including desktops, laptops, tablets, and smartphones.

Q20 What is Cross Browser Testing?

Cross Browser testing is a type of non-functional testing that lets you check whether your website works as intended when accessed through: Different Browser-OS combinations i.e., on popular browsers like Firefox, Chrome, Edge, Safari—on any of the popular operating systems like Windows, macOS, iOS and Android.

Q21 What’s the difference between verification and validation in testing?

:- Verification is a process of determining if the software is designed and developed as per the specified requirements.

:- **Validation** is the process of checking if the software (end product) has met the client's true needs and expectations

Q22 What is the difference between a bug, a defect and an error?

Error :- We can say that a mistake made by a programmer during coding is called an error.

Defect :- an error found during the testing in the development phase is called a defect.(Deviation from requirement)

Bug:- an error found during the testing phase is called a bug.

Q23 What is the test case and test scenario?

- **The test case** is an in-detailed document that includes **all possible inputs** such as **positive and negative**, and **the navigation steps.** These are implemented during the testing process to check whether the software application is performing the task for which it was developed or not.

- **Test scenario** It is a detailed document of test cases that cover **end to end functionality** of a software application in liner statements. The test scenario is a high-level classification of testable requirements. Before performing the test scenario, the test engineer needs to consider the test cases for each scenario.

Q24 What are the phases involved in the Software Testing Life Cycle?

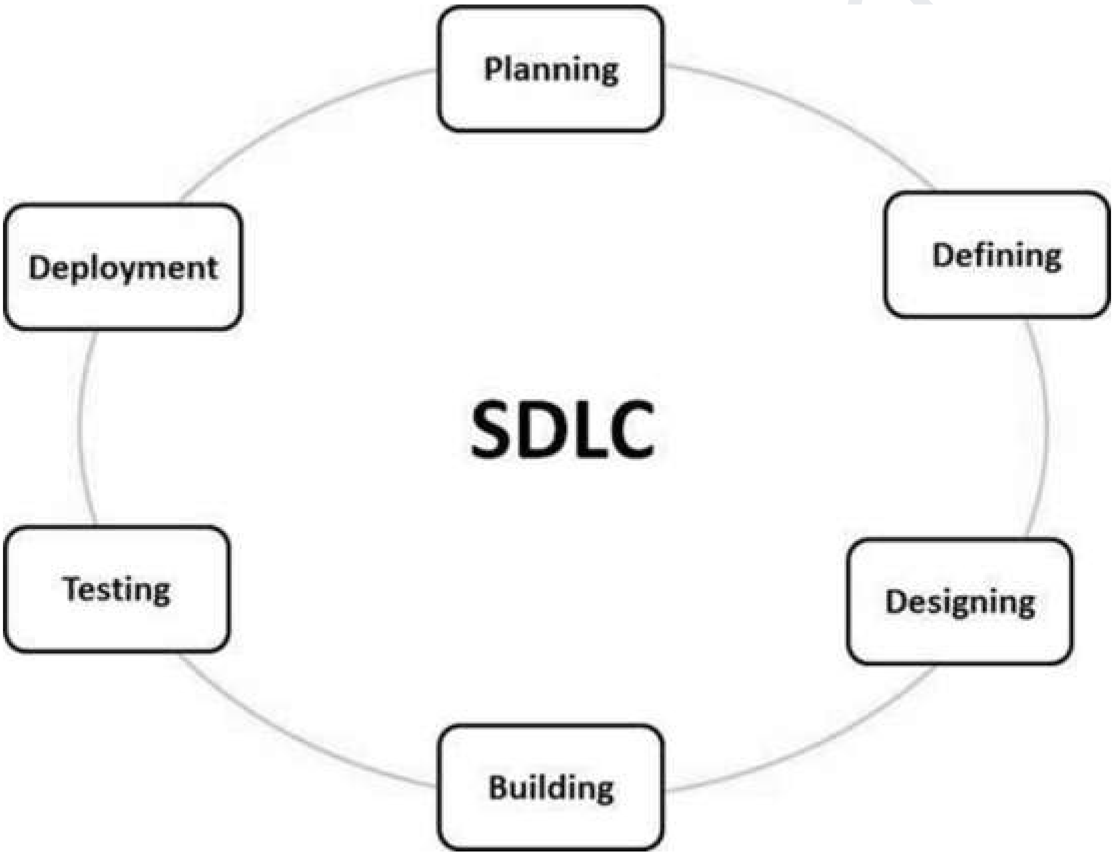
The different phases involved in the software testing life cycle are:

Phases	Explanation
Requirement Analysis	QA team understands the requirement in terms of what we will testing & figure out the testable requirements.
Test Planning	In this phase, the test strategy is defined. Objective & the scope of the project is determined.
Test Case Development	Here, detailed test cases are defined and developed. The testing team also prepares the test data for testing.
Test Environment Setup	It is a setup of software and hardware for the testing teams to execute test cases.
Test Execution	It is the process of executing the code and comparing the expected and actual results.
Test Cycle Closure	It involves calling out the testing team member meeting & evaluating cycle completion criteria based on test coverage, quality, cost, time, critical business objectives, and software.

Q25 What is SDLC?

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

The following figure is a graphical representation of the various stages of a typical SDLC.



A typical Software Development Life Cycle consists of the following stages −

Stage 1: Planning and Requirement Analysis

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

Stage 2: Defining Requirements

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an SRS (Software Requirement Specification) document which consists of all the product requirements to be designed and developed during the project life cycle.

Stage 3: Designing the Product Architecture

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification.

This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product.

A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

Stage 4: Building or Developing the Product

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle.

Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high level programming languages such as C, C++, Pascal, Java and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

Stage 5: Testing the Product
This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

Stage 6: Deployment in the Market and Maintenance
Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).
Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

Q26 Bug Life Cycle?

- **New:** When a new defect is logged and posted for the first time. It is assigned a status as NEW.

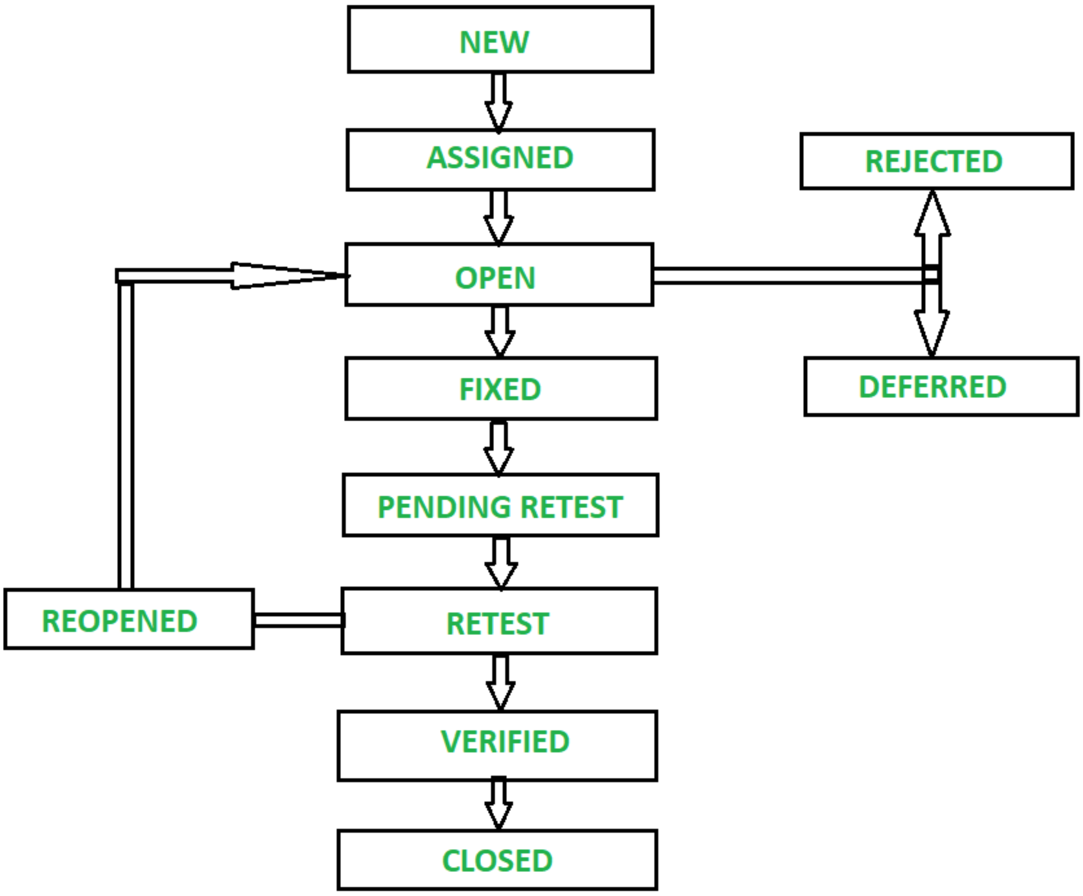
- **Assigned:** Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to the developer team.

- **Open:** The developer starts analyzing and works on the defect fix.

- **Fixed:** When a developer makes a necessary code change and verifies the change, he or she can make bug status as “Fixed.”

- **Pending retest:** Once the defect is fixed the developer gives a particular code for retesting the code to the tester. Since the software testing remains pending from the testers end, the status assigned is "pending retest."

- **Retest:** Tester does the retesting of the code at this stage to check whether the defect is fixed by the developer or not and changes the status to "Re-test."



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

- **Reopen:** If the bug persists even after the developer has fixed the bug, the tester changes the status to "reopened". Once again the bug goes through the life cycle.

- **Closed:** If the bug is no longer exists then tester assigns the status "Closed."

- **Duplicate:** If the defect is repeated twice or the defect corresponds to the same concept of the bug, the status is changed to "duplicate."
- **Rejected:** If the developer feels the defect is not a genuine defect then it changes the defect to "rejected."

- **Deferred:** If the present bug is not of a prime priority and if it is expected to get fixed in the next release, then status "Deferred" is assigned to such bugs.

- **Not a bug:** If it does not affect the functionality of the application then the status assigned to a bug is "Not a bug".

Q27 What is the difference between Severity and Priority?

Severity is basically a parameter that denotes the total impact of a given defect on any software.

Priority is basically a parameter that decides the order in which we should fix the defects.

Q28 What are the 7 principles of testing?

The seven principles of testing

- **Testing shows the presence of defects, not their absence.**
- **Exhaustive testing is impossible.**
- **Early testing saves time and money.**
- **Defects cluster together.**
- **Beware of the pesticide paradox.**
- **Testing is context dependent.**
- **Absence-of-errors is a fallacy.**

Q29 Test Cases of Pen?

Prerequisite for performing all these test cases is, a Pen must have a refill and the refill must have ink.

1. The grip of the pen: Verify if you are able to hold the pen comfortably.
2. Writing: Verify if you are able to write smoothly.
3. Verify that the pen is not making any sound while writing.
4. Verify the ink flow. It should not overflow nor get a break either.
5. Verify the quality of the material used for the pen.
6. Verify if the company or pen name is visible clearly.
7. Verify if the pen color or text written on the pen is not getting removed easily.
8. Verify, whether the width of the line drawn by the pen is as per the expectations or not.
9. Verify the ink color, it should be consistent from the start till the end.
10. Verify if a pen can write on a variety of papers like smooth, rough, thick, thin, glossy etc.
11. Verify for the waterproof ink. [Not for gel and ink pens].
12. Verify if the ink will not get dried easily by keeping the pen open for some time. [Not for ink pen]
13. Verify if any other refill fits in the pen or not.
14. Verify that the pen doesn’t have sharp edges or corners.
15. Verify if the ink and external assembly of the pen is made of non-toxic material.

Negative Test Cases/Scenarios

1. Put the pen in water and then try to write. Verify if you are able to write with this pen. The pen can get wet because of the water spill on the table or during the rainy season. It can be due to any reason.
2. Drop the pen from some height (Table height) in the upside-down position. Verify if you are able to write with this pen. By mistake, the pen can any time fall on the ground. So testing this possibility to know its impact, will help us in knowing the quality of the pen.

For both the above test cases, the frequency that these scenarios will happen may not be very high but it is not even very low. So by knowing its impact, we will be able to know more about the quality of the pen.

Performance Test Cases/Scenarios

1. Verify how fast you can write with this pen.
2. Verify if the pen will perform the same even though you use it continuously for hours.
3. Verify ‘How much can be written in one refill/ink sac?’
4. Verify if the tip or nib of the pen is not destroyed after continuous writing for hours.

Out of Scope Test Cases

1. Verifying the pen working in different gravity.
2. Verifying the pen working at different temperatures (Especially at 0 degrees).

Q30 Test Cases of Chair?

- Verify that the chair is stable enough to take an average human load.
- Check the material used in making the chair-wood, plastic etc.
- Check if the chair's leg are level to the floor.
- Check the usability of the chair as an office chair, normal household chair.
- Check if there is back support in the chair.
- Check if there is support for hands in the chair.
- Verify the paint's type and color.
- Verify if the chair's material is brittle or not.
- Check if cushion is provided with chair or not.
- Check the condition when washed with water or the effect of water on the chair.
- Verify that the dimension of chair is as per the specifications.
- Verify that the weight of the chair is as per the specifications.
- Check the height of the chair's seat from floor.

Q31 Test Cases of Login Functionality?

- Verify that as soon as the login page opens, by default the cursor should remain on the username textbox.
- Verify that the user is able to navigate or access the different controls by pressing the 'Tab' key on the keyboard.
- Check if the password is in masked form when typed in the password field.
- Check if the password can be copy-pasted or not.
- Verify that the user is able to login by entering valid credentials and clicking on the 'Login' button.
- Verify that the user is able to login by entering valid credentials and pressing Enter key.

- Check that the user is not able to login with an invalid username and password.
- Verify that the validation message gets displayed in case the user leaves the username or password field blank.
- Check that the validation message is displayed in case the user exceeds the character limit of the user name and password fields.
- Verify that the reset button functionality is on the login page. Clicking on it should clear the textbox's content.
- Verify if there is a checkbox with the label "remember password" on the login page.
- Verify that closing the browser should not log out an authenticated user. Launching the application should lead the user to the login state only.

Security Test Cases for Login Page

- Verify that there is a limit on the total number of unsuccessful login attempts. So that a user cannot use a brute-force mechanism to try all possible combinations of username-password.
- Verify that in case of incorrect credentials, a message like "incorrect username or password" should get displayed. Instead of an exact message pointing to the incorrect field. This is because a message like "incorrect password" will help a hacker in knowing that the username is correct. In this way, he will just need to try a different combination on the password field only.
- Verify the login session timeout duration. So, once logged in a user cannot be authenticated for a lifetime.
- Verify that once logged in, clicking the back button doesn't log out the user.

- Verify if SQL Injection attacks work on the login page. The application should not be vulnerable to SQL injection attacks.

- Verify that the XSS vulnerability should not work on the login page.

1 To 3 Years Of Experience

Q1 What is Quality Assurance and what are the different activities involved in Quality assurance?

Quality assurance is a process-driven approach that checks if the process of developing the product is correct and conforming to all the standards. It is considered a preventive measure. This is because it identifies the weakness in the process to build software. It involves activities like document review, test case review, walk-throughs, inspection.

Q2 What is Quality Control and what are the different types of testing involved in QC?

Quality control is a product-driven approach that checks that the developed product conforms to all the specified requirements. It is considered a corrective measure as it tests the built product to find the defects. It involves different types of testing like functional testing, performance testing, usability testing, etc.

Q3 What are the different types of testing?

Ans. Testing can broadly be defined into two types-

- **Functional testing** – Functional testing involves validating the functional specifications of the system.
- **Non Functional testing** – Non-functional testing is a type of testing that involves testing of non-functional requirements of the system such as performance, scalability, security, endurance, portability, etc.

the way the testing is done, it can be categorized as-