SDLC and Manual Testing IQ:

1. **Can you tell the difference between STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?**

SDLC is an acronym for Software Development Life Cycle which consists of the following phases:

- Requirements gathering/Analysis phase
- Design phase
- Coding/Development/Implementation phase
- Testing phase
- Production phase

But what exactly is SDLC?
SDLC is a process in which a software product is either created, enhanced or maintained
What actually happens in all of these phases? Lets analyze each phase:

- **Requirements gathering/Analysis Phase**

This is the phase when business stakeholders/client side members, Project Managers, and Business Analysts meet to discuss requirements for the project. Several meetings are hosted where the client states the expectations of the project and will also discuss topics regarding who will use the software product, how will users use the product, and specific information related to the development of the software product. This phase is arguably the most important phase of SDLC.

- **Design Phase**

This is when the software product is actually designed. This includes drawing plans, drawing an architectural style, and continuously hosting meetings with the client to get approvals. This is usually done by UI Designers, Graphic Designers, or sometimes even developers.

- **Coding/Development/Implementation Phase**

This is when the created design is actually developed.

- **Testing Phase**

This is when the testing is conducted of what was developed. This is where STLC comes in play. STLC is an acronym for Software Testing Life Cycle which consists of the

following phases: Requirement analysis, Test Planning, Test case Development, Setup testing environment, Test Execution, Test closure.

- **Production Phase/Deployment and Maintenance**

This is when the code developed has been deployed into production

2. Why do Software Applications have bugs?

There are various reasons why a system might have bugs, such as bad design, coding errors, miscommunication, continuous changes in requirements, rushing to meet deadlines and also the complexity of the application.

3. Can you tell the Difference between Verification and Validation?

Verification is when you are analyzing requirements and test cases to ensure we cover all scenarios and we are creating the right product and here there is no code involved. Simply reviewing documents to ensure everything is covered and NO coding is involved.

Validation is when we are executing our test cases and actually writing code to verify the system

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

Static Testing (Done In Verification Stage).

Static Testing is a White Box testing technique where the developers verify or test their code with the help of checklist to find errors in it, this type of testing is done without running the developed application or program. Code Reviews, Inspections, Walkthroughs are mostly done at this stage of testing.

Dynamic Testing (Done In Validation Stage).

Dynamic Testing is done by executing the actual application with valid inputs to check the expected output. Examples of Dynamic Testing methodologies are Integration Testing, System Testing, Regression and Acceptance Testing.

5. What Is the Difference between Black Box and White Box testing?

Black box testing is a testing approach which depends completely on the product requirements and specifications. Basically, front-end testing. The knowledge of internal paths, structures, or implementation of the software is not needed

White box testing is when you know the internal structure of the application meaning you know coding and can perform unit testing as well.

6. Can you tell different type of testing you are aware of?

- Unit testing
- Integration testing
- Regression testing
- Ad-Hoc
- Smoke testing
- Performance testing
- System testing
- UAT Testing

7. In which phase should testing begin?

Testing should start at the requirements phase.

8. What is Ad-Hoc Testing? Exploratory testing?

Both are the same thing, it is informal testing where you are testing without a test case and randomly testing a module conducting negative testing and positive testing but not in any order etc.

9. What is unit testing, and have you ever done unit testing?

No, I haven't done unit testing. Unit testing is done by the developers before they deploy their code from the Development environment to QA environment.

10. What is Integration testing?

Once every individual component is tested, we must make sure that when we integrated these components together they must work as expected. Example amazon.com has fresh and prime, but the entire website works together even though it has different components.

11. What is the difference between UAT and QA? What is Acceptance testing?

QA is one lower environment to UAT. QA can be a separate team that does testing and after testng PASS QA environment code is moved to UAT environment where again Testing is conducted to make sure no bugs have been missed. Also, UAT is mostly business side client side that conducts it. Acceptance testing is UAT testing another name for it.

12. What is system testing?

It is testing the entire system which includes but is not limited to the Front-end, Back-end, Database, Server, Hardware, and any related Software product/application.

13. What is Beta Testing?

Beta testing is conducted to gather feedback from the real-time usage of the product. It happens after the installation at the client-end or another good example is in real-world gaming industry they release a game to a certain audience before a mass release to get feedback.

14. What is 508 Compliance testing?

All government websites are supposed to be 508 compliant meaning disable personal should be able to use it. Features are specific font is used on these type of website, Text to speech capability and magnify capability etc. Just know what it is but you don't have to tell the interviewer that you have done so.

15. What is Boundary Testing?

Boundary testing is to make sure software accepts valid data inside the valid Boundary and rejects invalid data outside the valid Boundary. A simple example would be testing the input field limitation is from 0 to 100 where the lower boundary is 0 and the higher boundary is 100. We make sure we can't go lower than 0 and higher than 100. We only test the edges and not the middle.

16. What is Negative Testing and what is Positive Testing?

Negative testing is testing the application/system with a negative scenario using incorrect/invalid data and verifies if the application responds as expected.
Positive testing is testing the application/system with correct and valid data and verifies if the application responds as expected.

17. What is non-functional testing? Have you done any performance testing?

Performance testing, Security testing.
Simply say NO you have not done so and that performance testing was conducted by a separate Team.

18. What is Functional testing?

Simply, testing any functionality is functional testing. Manual testers do functional testing. As an Automation Tester, you may be required to perform Manual testing based on project needs.

19. What is a testing hierarchy?

- Unit testing: Developers test their code to make sure the functionality they are working on created is working.
- Component Testing: Component is a standalone functionality that can work by itself.
- Integration Testing All modules are combined. When integrated do they still work together.
- System Testing: Hardware, Software, Database, servers everything.
- Acceptance Testing: (User Acceptance Testing). It's testing by UAT team mostly business/client-side team or hired UAT team that validates what QA team tested before moving to production.

20. What is the difference between regression testing and retesting?

Regression testing is performed if a new functionality is added or bug fix occurs and we want to make sure other parts of the application are still functioning correctly.
While retesting is performed when a defect gets fixed and you have to retest the functionality to see it is really fixed or not.

21. What is the difference between system testing and integration testing?

System testing is when the entire system is checked such as hardware, software, servers, databases etc., whereas for integration testing, the integration between the individual modules is tested.

22. What is a hotfix?

A hotfix is an emergency issue that needs to be fixed right away and does not need to go through a whole sprint cycle. It's an exception where it can be developed, tested and deployed in a few hours or a day since it's an extreme priority and client wants the critical issue fixed ASAP and deployed.

23. When is a test considered to be successful?

Tests that cover more functionalities and discover more errors in your software product are considered more successful. The whole purpose of a testing process is to discover as many bugs as possible and as early as possible.

24. If there isn't enough time for testing what would you do?

If we are short on time I would prioritize my test cases and tackle ones that cover most functionalities and I will also check with my team members if they can help out since I have been on agile projects we work together in such matters.

When we prioritize here is what we look for:

- Which functionality is most visible to the user?
- Which functionality has the largest safety impact?
- Which aspects of the application are most important to the customer?
- Which aspects of the application can be tested early in the development cycle?
- Which parts of the code are most complex and thus most subject to errors?
- Which parts of the application were developed in rush or panic mode?
- Which parts of the requirements and design are unclear or poorly thought out?
- What do the developers think are the highest-risk aspects of the application?
- What kinds of problems would cause the most customer service complaints?
- What kinds of tests could easily cover multiple functionalities?

25. What would you do if you end up with unclear requirements or requirements are not available?

In this scenario, your knowledge of application plays a big role. I would start to do some random testing to understand what is being asked and try to discover to get an idea, so I can start writing scripts. Then I will also check out our meeting notes, emails, and discussions. On top of that, look for wireframes which might give me an idea but overall my common sense and experience plays a part here and also I would think from a customer's perspective. Also if BA is available to clarify with them as well.

26. When do you know if you have enough test cases for your project or a specific module?

If all requirements have at least one test case, then we know we have covered everything also in this case a Requirements Traceability Matrix (RTM) is very useful since it maps out Requirements to Test Cases.

27. Tell us some key challenges you face in Testing industry?

- Requirements changing.
- Application not stable.
- Rush testing to meet deadline
- Unclear requirements
- Domain knowledge and business user perspective understanding.
- Lack of Regression testing or not enough.
- Lack of skilled testers.
- Lack of resources, tools and training

28. Can you do 100% manual testing and find all bugs?

No, it is impossible because there are a number of possibilities and scenarios. If we cover all functionalities and test them then we can reduce the risk of bugs and have a satisfied client.

29. What is Requirement Traceability Matrix?

RTM is mapping requirements to test cases and test cases to defects. If there is a defect found, we can trace back to tell which specific requirement failed. RTM also tells us if we are missing test cases in case requirement is there and no test case is available.

30. What is Test Plan?

The Test Plan document is usually prepared by the Test Lead or QA Manager and the focus of the document is to describe what to test, how to test, when to test and who will do what test.

Components of the Test Plan:

- Introduction
- Test items
- Features to be tested
- Features not to be tested
- Test techniques
- Testing tasks
- Features pass or fail criteria
- Test environment (Entry criteria, Exit criteria)
- Test deliverables
- Staff and training needs
- Responsibilities
- Schedule

31. What is Test Strategy?

A Test Strategy document is a high level document and normally developed by project manager. This document defines "Software Testing Approach" to achieve testing objectives. It sets the standards for testing processes and activities.

Components of the Test Strategy:

- Scope and Objectives
- Business issues
- Roles and responsibilities
- Communication and status reporting
- Industry standards to follow
- Test automation and tools
- Testing measurements and metrics
- Risks and mitigation
- Defect reporting and tracking

32. What is a test case and what does it include?

A test case is a document which has a set of actions that needs to be executed to verify a feature or functionality of your software application. It includes

- Test Scenario/Objective
- Pre-Condition
- Test Case ID
- Test Data
- Test Description
- Expected Result
- Actual Result
- Pass or Fail
- Comments

33. Who is responsible for writing test cases and test plans?

We testers write test cases for various scenarios, while test lead or even QA Manager is responsible for making a test plan.

34. What is test coverage?

What are we testing and How much are we testing!
Test coverage means how many test cases that we have and what functional area those test cases are covering. Are our test cases cover the entire application or no?

35. How do you define the bug life cycle?

Bug life cycle basically comprises of numerous statuses of an error during its life cycle. A few examples are open, deferred, solved, reopened, fixed, retest, Reject, solved and closed.

36. How do you deal with a bug which is not consistently reproducible?

The best approach is to take a screenshot of it if you are using Snagit you can even capture video of it and note it down. Discuss the bug with the team so everyone can keep an eye on it.

37. What would you do if a developer rejects your bug and refuse to fix it?

Go back and retest it several times that it is a bug. Then check it against the requirements', wireframes. If the developer still refuses to fix it then set up a meeting between BA and developers. If in rare situation developer still does not want to fix, then QA manager or Project manager get them involved last resort.

38. What do you do when you find a bug?

First you will go back and retest it to make sure it is a bug, then you take a screenshot and create/log a bug in Jira with as many details as possible such as short description, environment found in, exact steps to reproduce, severity, priority etc. so the developer can try to recreate it.

39. Bug priority vs Bug severity

Priority represents the importance of fixing a bug, and reflects a business decision as to how soon that bug should be fixed: all priority 1 bugs should be fixed before priority 2 bugs, etc.
Severity represents "how bad a bug is." For example, a bug that causes the program to crash would be considered high severity, while a small spelling error might be low severity.

EXAMPLES:

- Low priority bugs are small errors that don't stop shipping or cause real trouble for any user.
- High priority bugs are problems that our customers will see, can corrupt data, or crash a system.
- High Priority & High Severity: An error which occurs on the basic functionality of the application and will not allow the user to use the system.
- High Priority & Low Severity: The spelling mistakes that happens on the cover page or heading or title of an application.
- High Severity & Low Priority: An error which occurs on the functionality of the application (for which there is no workaround) and will not allow the user to use the system but it's on part of the application rarely used.
- Low Priority and Low Severity: Any cosmetic or spelling issues which is within a paragraph or in the report (Not on the cover page, heading, title).

40. What is the difference between bug release and bug leakage?

Bug release is when a version of the software is released with a set of known bugs. These bugs are usually of low severity/priority. It is done when a software company can afford the existence of a bug in the released software rather than the time/cost for fixing it in that version.

Bug leakage is something when the bug is discovered by the end users or customers and missed by the testing team to detect while testing the software.

41. Let's say you have a bug that comes in the day before the sprint is ending, do you let it go into production or move it to the next sprint?

If the bug is of a very low priority and severity we will log it and fix it at a later time since it doesn't impact the system much and release it into production, but if it's a show stopper we will never release it into production since it can cause other issues.

42. How many bugs do you find a day?

We don't test every day since we have meetings and time must be allocated to create and automate test cases. When we do actual testing anywhere from 2 to 3 bugs a day at the very beginning and as we continue testing the bug rate falls to 0 to 1.