Soft Skill Interview questions

General Introductory / Behavioural Questions:

1. **Why did you choose software quality analysis as a profession/ career?**

   I have always looked at products from a user's perspective. If it is not designed to give the user a seamless experience, it is not a good quality product. User friendliness and ease of use of any product, including software applications, has always inspired me.

   Because of that personality trait, I enjoy looking for the loophole in any product. I love breaking things down analytically and thinking out of the box to see if there are any gaps that were not thought through. This love of building perfect products made me select quality analysis as my profession/ career.

2. Describe your typical day at work

My day at work typically starts with making a "to-do" list of thing i need to work on that day. A list might include but not be limited to , any of the following:

- Check my email (confirm is there any meeting that i need to attend)

- Attending Daily scrum meeting or any knowledge transfers meeting.
- Continue work on task, reading requirements to better understand which kinds of test cases i need to buil
- Writing automation scripts in scenarios where there is a need for automation
- Executing the testing and coordinating with developers to check the defects that i have found
- Meeting with the development team to communicate the daily defect log and plan around fixing and retesting defects.

3. Where do you see yourself 5 years from now ?

I would like to get some hands-on experience in conducting business within the IT industry. I would like to learn as much as possible to be more technical competitive person. Ideally, I'd like to become a real software developer in Testing field.

(Note: Some managers are already under stress. They are not comfortable to hire the person that is smarter than themselves. So you have to know how to make them comfortable by let them feel that you are technically good, but you will be Zero thread to their position.)

4. Are you a team player or a lone wolf?

I have been in both kinds of situations. While I feel teamwork is really necessary to boot the enthusiasm, and productivity, I am also comfortable working by myself. From my experience, things like brainstorming, going over user stories and scenarios and plugging gaps in user stories or acceptance criteria need a lot of teamwork. However, there are things like documentation, preparing presentations, data analysis which are best done alone in the start.

5. Do you like working in small teams or big teams?

I have had the chance to work in teams as big as more than 12 testers, and also in small teams of 4 or 5. While each situation has its pros and cons, I feel comfortable and work well either way.

For example, with a big team, you have more resources and a better division of labor. However, the coordination is challenging and chaotic

Similarly, small teams can be nimble and are suitable for agile style development. However, the scope has to be limited because of team size, and often the staff is stretched thin and mus work long hours.

6. Have you work under pressure?

I’m not strange to working under pressure. Good pressure- such as having a lot of assignments / tasks to work on or and upcoming deadline help me to stay motivated and productive. Of course, there are time when too much pressure can lead to stress; However, I can prioritize my job and meeting deadlines which prevents me from feeling stressed often during the work. For example, when developer couldn’t deploy the code on time, as a automation tester we shouldn’t do nothing and wait. In that situation , I read acceptance criteria/ requirement,analyze scenarios and create the test case or   write my pseudocode. So once it’s done my 70% of job is done also, therefore , I can meet my deadline or finish my job according to deadline.

This is how i handle my stress or avoiding unnecessary stress by simple prioritize , focus on and get the job done.

7. Why did you apply for this position? Or Why should we hire you ?

I feel that your vacancy ideally fits my work experience,skills and qualifications. Therefore, I will be able to make significant contribution to your company as well as fulfill my potential.

I would like to work for a company where i feel i can make a real difference. After did some research about your company I have discovered that it has a excellent reputation. I was also greatly impressed by your companies mission statement, values and culture. I feel strongly that your is a organization that I would like to work for and be associated with.

I believe your company will help me to develop my career in the direction that want it to go.

I think I would be a perfect candidate and compatible fit for this position.

8. Why are you looking for a change now ?

I have worked at my present employer for X number of years, and it has been a great journey. I still love my job and am very good at it. However, I have hit a ceiling in terms of development here, and I am looking for an opportunity to face new challenges, achieve certain key career aspirations and grow within the industry. When I read your job description, It immediately piqued my interest as it seems like a perfect match for my skills, and a great opportunity to add value to your organization.

9. What are your strengths and weaknesses as a QA?

Strengths:

- QAs have input into the entire software development process. Verifying software is just one aspect of the role- but most end users aren’t interested in what any applications can do for them. I have the ability to get a very clear understanding of the business requirements along with business logic much before starting the test strategy.
- Quick learning
- Communication skills
- Work under pressure or handle stress

Weaknesses

In the pass my greatest weakness was that i was very critical of my own work. I always thought in order to produce excellent and error-free work, have to go each and every detail, while this is beneficial to my job performance but it is possible to go to extremes. I have also found that I can easily waste time checking and rechecking the same stuff. For example, when i found defect , it is great for reproduce 2-3 times to confirm that is a bug or not. But what I did is I do recreate bug 6-7 times with using different test data or environment. But then i realized this is just waste of time. So now I'm always making a conscious effort to trust myself and my quality focus more on other task.

10**. Do you have any questions for me?**

Yes , I do have couple questions, Can you tell me about the team structure and who I will be working with? What are the challenge that your team facing now?

Basic QA questions

1.What is Software Development Life Cycle?

The systems (or software) development life cycle (SDLC) is a conceptual model used in project management that describes the stages involved in an information system development project from an initial feasibility study through maintenance of the completed application.

It includes the following different stages:

1.Requirement
2.Design phase
3.Coding
4.Testing
5.Release (Production)
6.Maintenance

2. What is Software Testing Life Cycle?

The testing of software has its own life cycle. It starts with study and analyzing the requirements.STLC refers testing process which has specific steps to be executed in a definite squesences to ensure that the quality goals have been met.

It included the following different stages:

1.Requirement Analysis
2.Test plan
3.Test development (writing a test case , review test case)

4.Test execution
5.Test closure.

3. What is Requirement document?

In other words, all the expected functionalities out of the application are documented in terms of **“Requirements”** and this document is called a **Requirement document.** It is also called an SRS document, which stands for **System Requirement Specification Document.**

4. What is a Module?

A “Module” is a software component that has a specific task. It can be a “link” which can go inside to its component detail.

5. What is Build?

When each of the different modules of software is prepared, they are put in a single folder by Configuration Management Team(CMT) and it is called the build. In other word, the developers put their code in the share location (folder) and all those code (modules) are combined together so that it is a complete application that works.

6. What is meant by the Build Deployment?

When the Build so prepared by the CMT is sent to different Test Environments, it is called the Build Deployment.

7. How do you test the application if the requirements are not available?

If requirement document is not available for an application, a test plan can be written based on assumptions made about the application. Assumptions that are made should be well documented in the test plan. (Bookish answer).

A-1 If the requirement is not available we have to do our best to gather as much information as possible from the end-users, client and similar applications from competitors. Based on our research we can still test the application. For example if we have to test an e-commerce application without requirement we can analyze amazon.com to gather basic requirements and perform our testing accordingly.

A-1.2 When we do not have any formal document (e.g. FSD,BRD.SRS) available for reference, we can get help from earlier versions of the application, defects description and comments, wireframes, etc. It is always a good idea to include some members on the team who have good domain knowledge. We can also talk directly with developers and business analysts, who help us understand application behavior. We can create reference documents for the testing team, which will help new team members to become productive quickly.

A-2.3 In my current company , while I was working on any production defect ticket which is not including any Acceptance Criteria (requirement).In that case, I go to developer desk discuss about root cause scenarios and take a note, analyze it , create a test case , execute them make sure the issue is fixed then complete my testing.

8. What is peer review ?

Peer review is process for finding any error or defect on various documents , it is conducted by team members . the purpose of peer review is find the defect as early as possible before it is deployed to next step.

9.Who approves test case?

The approver of test cases varies from one organization to the next. In some organizations, the team lead/QA lead may approve the test cases while another approves them by team members through peer review

10.How can you tell when enough test cases have been created to test a system or module?

That is a reason we need to have RTM(Requirement Traceability Matrix) we can tell how many requirement has been covered by test cases and how many still left from RTM . In other words, it is a document that maps and traces user requirement with test cases. The main purpose of Requirement Traceability Matrix is to see that all test cases are covered so that no functionality should miss while doing Software testing.

11. What is a test plan ? who writes test plans

A document describing the detailed approach to test the software and what the eventual workflow will be. It consists of features to be tested, features not to be tested, approach, entry criteria, exit criteria, test environment, training needs, resources, roles and responsibilities, risks and contingency plan.

What are the tasks involved in Test Planning?

1. Understand and analyze the requirements
2. Risk analysis
3. Test Strategy Implementation
4. Test Estimations
5. Team formation
6. Test Plan documentation
7. Configuration Management planning
8. Traceability Matrix
9. Define Test Environment set up

Test Lead prepares Test Plan

12. Who creates the test strategy? What are the main contents that you would include in it?

The test manager creates the test strategy. It is a company-level document the template for which is typically recommended by the PMO or an equivalent shared services group. It is then customized for the project by the test manager based on the unique considerations for the particular application. The typical things I would include in a test strategy document are :

- Types of testing
- Steps that we need to complete before testing
- The testing approach, including details like number of users, creation of test cases, execution approach, ownership,etc.
- QA timelines
- Testing process with actors and daily steps.
- Documents to be prepared with respective formats.

13.Does test strategy and test plan define the same purpose?

Yes, the end purpose of test strategy and test plan is same i.e. to works as a guide or manual to carry out the software testing process, but still they both differs

14. What is test case and who creates test cases?

It is documentation which describes step by step how to perform testing. it includes: test case ID,test case name ,pre-condition,step description,test data,expected result and actual result and pass/fail. Testers will create test cases.

15. What is the differences between test scenario, test case and test script?

Test scenario is a description of user's multiple actions that might face when using the applications. It is about what to test. It

is high level of test cases.

Test script in software testing is a set of instructions that will be performed on the system under test to test that the system functions as expected.This terminology mostly used for automation testing

Test case is documentation which specifies input values,expected output and the preconditions for executing the test.This terminology mostly used for manual testing. It is about how to test . It's a layout of the low-level details on how to test the scenario

16.What is the validation and verification ?

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.

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.

17. **What is the difference between verification and validation approach of software testing?**

Verification is done throughout the development phase on the software under development whereas validation is performed over final product produced after the development process with respect to specified requirement and specification.

18. **What Is Entry And Exit Criteria In Software Testing?**

Entry criteria – It is a process that should run when a system begins. It includes the following artifacts.

SRS (Software Requirement Specification)
FRS (Functional Requirement Specification)
Test-Case
Test-plan
Test strategy

Exit Criteria – It signals when the testing should complete and when should the product be ready to release. It includes the following artifacts.

Test Summary Report

Metrics

Defect Analysis report

19. In a long term project, many times the requirements change. Do the test plans also change along with the requirements?

Most times, yes. If requirements change, the design documents and specifications(for that particular module which implements the requirements) will also change. Hence, the test plan and test cases only for that portion would also need to be updated. This is because “Resource Allocation” is one section of the test plan. We would need to write new test cases, review, and execute the test cases. Hence, resource allocation would have to be done accordingly. As a result, the test plan would change.

20. What are some common problems you have observed in projects that might make testing challenging? Also provide ways in which you have overcome these challenges.

Some common problems in software development process that might make testing challenging are:

Inaccurate or incomplete requirements:

There are many reasons why requirements can be inaccurate or incomplete. This directly affects the development and subsequently the testing. Sometimes the requirements might not be at a level of detail to tie back to granular test case. In such situations, the test team has to take the best guess on what the outcome of a test case should be, and this can make it ambiguous and problematic when the product goes to UAT. To resolve this, a tester should also behave like BA. Whenever you find requirement where a clear test case cannot be defined, or enough detail is not available, you should speak up and request the BA to get the missing information. This will consume time, but will make sure that the product you deliver is what the users really want.

Unrealistic schedule:

Unrealistic schedule and a short time allocated for testing. There are ways to handle this. One way is to ensure that the test cases are clearly prioritized so that we can test out the high and medium cases first. Alternatively, we can also do less documentation and more ad hoc testing to cover the key features of the system in the time that has been provided.

21**. What would you do if the defect you reported could not be reproduced by the development team, even after repeatedly adding more details to the defect?**

I assume the screenshots of the defect are already provided in this situation. The only alternative would be to create a video and upload it with the defect.

However, there are times when the bug has resulted from a totally random situation which cannot be recreated. In such a situation, I will note it separately and continue testing to see if it happens again. If not, it can be included as a low priority, low probability defect.

22.How will you resolve a conflict between developer and tester where tester wants the defect to be fixed, while developer does not consider it to be a defect?

The testers and developers look at the requirement from a different perspective. So my approach is to look at the requirement from the customer's perspective and make the decision.

I can give an example from a recent project. This project involves a lot of online forms that a user has to fill. These forms have hundreds of fields, and a user has to select one or more values from hundreds of choices from drop-down list. However, I realized while testing that a drop -down is not user friendly at all if you are going to select multiple values from a set of hundreds of values. A multi-select box is a much better option. The developer did not think it was a bug because they built it exactly as per the requirement specification. However, it would not have worked for an end user. When I explained this to them, they saw value in it, and it was finally recorded as a defect.

23. Please give an example each of high priority-low severity, low priority-high severity, and low priority-low severity defect.

Low priority-low severity:

Any misspelling, font size or color.

Low priority-high severity:

Company logo color not as expected. Or written by lower case.

High priority-high severity:

Application login functionality is not work.

24. Distinguish between priority and severity.

Priority defines the business need to fix or remove identified defect whereas severity is used to describe the impact of a defect on the functioning of a system

25. Which parameter tells the importance of fixing the defect from a customer's perspective?

Defect priority is the parameter user to assess the importance of the defect fix from the user/customer's perspective. The higher the priority, the more important it is to fix the defect.

26. Explain the difference between bug severity and bug priority?

Severity : Bug/Defect severity can be defined as the impact of the bug on the application. It can be Critical, Major or Minor. In simple words, how much effect will be there on the system because of a particular defect

Priority : . Main focus on 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.

Development team takes up the high priority defects first rather than of high severity. Generally, severity is assigned by Tester / Test Lead & priority is assigned by Developer/Team Lead/Project Lead.

27. What should be the approach when there is very little time given for testing, and your have to complete the testing within that short time frame?

Check with the BA and developers to decide on the high priority test cases. Execute high priority test cases first.

Ad hoc testing always yields good results in less time. The key here is to assign different areas of the application to different team members to perform ad hoc testing.

28. What is bug triage?

Bug triaging is the determination of how important the bug is with respect to other bugs identified in the system. The steps involved would be :

Analyze the bug
Step appropriate bug priority
Assign bug to proper bug owner
Adjust bug severity if needed
Move defects to deferred state if needed.

29. What is a latent defect?

A latent defect is an existing defect which has not caused failure because a particular set of conditions never occurred simultaneously

30. **What is the responsibility of the tester when a bug was not caught during testing but was found by the client during UAT?**

When a client reports any bug to the development team, it's a major drawback of the system which leaves the client with a negative impression of the delivered system. In such a case, the tester should reproduce the client-reported bug in his/her system. If it is not reproduced in the local systems, but only produced in the client's system, it should be labeled as an inconsistent bug. We can mark the bug as inconsistent and temporarily close the bug.

31. What are the functional testing types?

Unite testing
Smoke testing
Sanity testing
Integration testing
System testing
Regression testing
UAT (user acceptance testing)

32.What are different Test Levels?

There are 4 test levels a.

- Unit/Component
- Integration testing c
- System Testing d
- Acceptance Testing

33.What is the difference between Re-testing and Regression Testing?

Re-Testing is done to verify defect fixes and Regression testing is done to check if the defect fixes have disturbed existing functionality that was working fine before making the changes.

34.Why non-functional testing is equally important to functional testing?

Functional testing tests the system’s functionalities and features as specified prior to software development process. It only validates the intended functioning of the software against the specified requirements and specification but the performance of the system to function in the unexpected circumstances and conditions in real world environment at the users end and to meet customer satisfaction is done through non-functional testing technique. Thus, non-functional testing looks after the non-functional traits of the software.

35.Which is a better testing methodology: black-box testing or white-box testing?

Both black-box and white-box testing approach have their own advantages and disadvantages. Black-box testing approach enables testers to externally test the system on the basis of specified requirement and specification and does not provide the scope of testing the internal structure of the system, whereas white-box testing methodology verify and validates the software quality through testing of its internal structure and working.

36.If black-box and white-box, then why gray box testing?

Gray box testing is a third type of testing and a hybrid form of black-box and white-box testing approach, which provides the scope of externally testing the system using test plans and test cases derived from the knowledge and understanding of internal structure of the system.

37. Difference between static and dynamic testing of software.

The primary difference between static and dynamic testing approach is that the former does not involves the execution of code to test the system whereas latter approach requires the code execution to verify and validate the system quality.

38. List out various methodologies or techniques used under static testing.

1. Inspection
2. Walkthroughs
3. Technical reviews
4. Informal reviews
5. Peer reviews

39.Smoke and Sanity testing are used to test software builds. Are they similar??

Although, both smoke and sanity testing is used to test software builds but smoke testing is used to test the initial build which are unstable whereas sanity tests are executed on relatively stable builds which had undergone multiple time through regression testing.

40.Why exploratory testing is preferred and used in the agile methodology?

As agile methodology requires the speedy execution of the processes through small iterative cycles, thereby calls for the quick, and exploratory testing which does not depends on the documentation work and is carried out by tester through gradual understanding of the software, suits best for the agile environment.

41.What is positive and negative testing?

Positive testing is the activity to test the intended and correct functioning of the system on being fed with valid and appropriate input data whereas negative testing evaluates the system’s behavior and response in the presence of invalid input data.

42.How system testing is different to acceptance testing?

System testing is done with the perspective to test the system against the specified requirements and specification whereas acceptance testing ensures the readiness of the system to meet the needs and expectations of a user.

43. What is the importance of database testing?

Database is an inherited component of a software application as it works as a backend system of the application and stores different types of data and information from multiple sources. Thus, it is crucial to test the database to ensure integrity, validity, accuracy and security of the stored data.

44.What are the tasks involved in Test Design phase?

i. Creating Test scenarios
ii. Test case documentation
iii. Test data collection

45.What is Requirement Traceability Matrix?

Document showing the relationship between Requirements and Test Cases.

46.Who prepares and updates Traceability Matrix?

Test Lead or Team Lead creates Traceability Matrix and Testers update Traceability Matrix throughout the STLC.

47. **What is Software Quality Assurance (SQA)?**

    Software quality assurance is an umbrella term, consisting of various planned process and activities to monitor and control the standard of whole software development process so as to ensure quality attribute in the final software product.

48. **What is Software Quality Control (SQC)?**

    With the purpose similar to software quality assurance, software quality control focuses on the software instead to its development process to achieve and maintain the quality aspect in the software product.

49. **What is Software Testing?**

    Software testing may be seen as a sub-category of software quality control, which is used to remove defects and flaws present in the software, and subsequently improves and enhances the product quality.

50. **Whether, software quality assurance (sqa), software quality control (sqc) and software testing are similar terms?**

    No, but the end purpose of all is same i.e. ensuring and maintaining the software quality.

51.**Then, what’s the difference between QA, QC and Testing?**

SQA is a broader term encompassing both SQC and testing in it and ensures software development process quality and standard and subsequently in the final product also, whereas testing which is used to identify and detect software defects is a subset of SQC.

52.Why entry criteria and exit criteria is specified and defined?

Entry and exit criteria is defined and specified to initiate and terminate a particular testing process or activity respectively, when certain conditions, factors and requirements is/are being met or fulfilled.

53. **What is a software testing artifact?**

Software testing artifact or testing artifact are the documents or tangible products generated throughout the testing process for the purpose of testing or correspondence amongst the team and with the client.

54.Why there is a bug/defect in software?

A bug or a defect in software occurs due to various reasons and conditions such as misunderstanding or requirements, time restriction, lack of experience, faulty third party tools, dynamic or last time changes, etc.

55.How to categorize bugs or defects found in the software?

A bug or a defect may be categorized on the priority and severity basis, where priority defines the need to correct or remove defect, from business perspective, whereas severity states the need to resolve or eliminate defect from software requirement and quality perspective.

56. **What is bug life cycle?**

Bug or Defect life cycle describes the whole journey or the life of a defect through various stages or phases, right from when it is identified and till its closure.

57. Difference between error, defect and failure.

In the software engineering, error defines the mistake done by the programmers. Defect reflects the introduction of bugs at production site and results into deviation in results from its expected output due to programming mistakes. Failure shows the system's inability to execute functionalities due to presence of defect. i.e. defect explored by the user.

58. How testing is different with respect to debugging?

Testing is done with the purpose of identifying and locating the defects by the testing team whereas debugging is done by the developers to fix or correct the defects

59.What are the different approaches to perform software testing?

Generally, there are two approaches to perform software testing viz. Manual testing and Automation. Manual testing involves the execution of test cases on the software manually by the tester whereas automation process involves the usage of automation framework and tools to automate the task of test scripts execution.

60. What is the advantage of automation over manual testing approach and vice-versa?

In comparison to manual approach of testing, automation reduces the efforts and time required in executing the large amount of test scripts, repetitively and continuously for a longer period of time with accuracy and precision.

However, 100% automation testing is impossible.Manual testing is still necessary. The advantage of manual testing is, Manual testing can be done on all kinds of applications, application must be tested manually before it is automated, it is preferred in the projects where the requirements change frequently and the products where the GUI changes constantly,it allows tester to perform adhoc testing , exploratory testing.

61.Difference between ad-hoc testing and exploratory testing?

Both ad-hoc testing and exploratory testing are the informal ways of testing the system without having proper planning & strategy. However, in ad-hoc testing, a tester is well-versed with the software and its features and thereby carries out the testing whereas in exploratory, he/she gets to learn and explore more about the software during the course of testing and thus tests the system gradually along with software understanding and learning throughout the testing process.

62. When, what and why to automate?

Automation is preferred when the execution of tests needs to be carried out repetitively for a longer period of time and within the specified deadlines. Further, an analysis of ROI on automation is desired to analyse the cost-benefit model of the automation. Preferably functional, regression and functional tests may be automated. Further, tests which requires accuracy and precision, and is time-consuming may be considered for automation, including data driven tests also.

63. What are the challenges faced in automation?

Some of the common challenges faced in the automation are

1. Initial cost is very high along with the maintenance costs. Thus, requires proper analysis to assess ROI on automation.
2. Increased complexities.
3. Limited time.
4. Demands skilled tester, having appropriate knowledge of programming.
5. Automation training cost and time.
6. Selection of right and appropriate tools and frameworks.
7. Less flexible.
8. Keeping test plans and cases updated and maintained

64. Which is better approach to perform regression testing: manual or automation?

Automation would provide better advantage in comparison to manual for performing regression testing.

65.Which part of the regression test should be automated?

Or

66.When do you choose automated testing over manual testing?

If the test cases are high priority test cases
if the functionality is critical functionality
if the test cases are part of smoke test
If the test case are too long and too difficult to execute on manually
The regression test cases based on the priority .
we should automated test cases as much as possible.

67.what type of testing only done by manual or not automated

1. Dynamically changing functionality
2. Test cases that require more Human interaction
3. We can not capture content of an Image
4. Look and Feel of Web Application.

5. one time testing.

68. Whether a software application can be 100% tested?

No, as one of the principles of software testing states that exhaustive testing is not possible.

69. When to start and stop testing?

Basically, on the availability of software build, testing process starts. However, testing may be started early with the development process, as soon as the requirements are gathered and available. Moreover, testing depends upon the requirement of the software development model like in waterfall model, testing is done in the testing phase, whereas in agile testing is carried out in multiple and short iteration cycle. Testing is an infinite process as it is impossible to make a software 100% bug free. But still, there are certain conditions specified to stop testing such as:

- Deadlines

- Complete execution of the test suites and scripts.
- Meeting the specified exit criteria for a test.
- High priority and severity bugs are identified and resolved.
- Complete testing of the functionalities and features.

70. Brief out different forms of risks involved in software testing.

Different types of risks involved in software testing are budget risk, technical risk, operational risk, scheduled risk and marketing risk

71. What constitutes a test case?

A test case consists of several components. Some of them are test suite id, test case id, description, pre-condition, test procedure, test data, expected results, test environment.

72. **Why and how to prioritize test cases?**

Due to the abundance of test cases for the execution within the given testing deadline arises the need to prioritize test cases. Test prioritization involves the reduction in the number of test cases, and selecting & prioritizing only those which are based on some specific criteria.

73. What are the roles and responsibilities of a tester or a QA engineer?

A QA engineer has multiple roles and is bounded to several responsibilities such as defining quality parameters, describing test strategy, executing test, leading the team, reporting the defects or test results.

74. Whether test coverage and code coverage are similar terms?

No, code coverage amounts the percentage of code covered during software execution whereas test coverage concerns with the test cases to cover specific functionality and requirement.

75.List out different types of documentation/documents used in the software testing.

a. Test plan
b. Test scenario
c. Test cases
d. Traceability Matrix
e. Test Log and Report

76.If an application is in production, and one module of code is modified, is it necessary to retest just that

module or should all of the other modules be tested as well?

It is a good idea to perform regression testing and to check all of the
other modules as well. At the least, system testing should be
performed.

77. **What is a negative test case?**

Negative test cases are created based on the idea of testing in a destructive manner. For example, testing what will happen if inappropriate inputs are entered into the application

78. Explain random testing.

Random testing involves checking how the application handles input data that is generated at random. Data types are typically ignored and a random sequence of letter, numbers, and other characters are inputted into the data field

79.Define smoke testing.

A1-Smoke testing is a form of software testing that is not exhaustive and checks only the most crucial components of the software but does not check in more detail. (bookish way)
A2- In my current project we run smoke test to make sure if the

application is stable enough to perform other major testing activities.

- Smoke testing is used to test all areas of the application without going into too deep.
- A smoke test always use an automated test or a written set of tests. It is always scripted.
- Smoke testing is designed to include every part of the application in a not thorough or detailed way.
- Smoke testing always ensures whether the most crucial functions of a program are working, but not bothering with finer details.

80. What steps are involved in sanity testing?

Sanity testing is very similar to smoke testing. It is the initial testing of a component or application that is done to make sure that it is functioning at the most basic level and it is stable enough to continue more detailed testing.

- Sanity testing is a narrow test that focuses on one or a few areas of functionality, but not thoroughly or in-depth.
- A sanity test is usually unscripted.
- Sanity testing is used to ensure that after a minor change a small part of the application is still working.
- Sanity testing is a cursory testing, which is performed to prove that the application is functioning according to the specifications. This level of testing is a subset of regression testing.

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

For system testing, the entire system as a whole is checked, whereas for integration testing, the interaction between the individual modules are tested.

Agile

1.What is Agile?

Agile software development refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams

2.What kind of Agile methodology did you use in your previous project?

Scrum is a subset of Agile. It is a lightweight process framework for agile development, and the most widely-used one.

- A “process framework” is a particular set of practices that must be followed in order for a process to be consistent with the framework. (For example, the Scrum process framework requires the use of development cycles called Sprints, the XP framework requires pair programming, and so forth.)
- “Lightweight” means that the overhead of the process is kept as small as possible, to maximize the amount of productive time available for getting useful work done.

3.Can you describe me the scrum?

Scrum is very commonly used Agile framework for software development. There are several roles in Srum:

Product owner: product owner is the one who created wish list to the project which is called product backlog. Product owner usually prioritizes the product backlog item and comes up with sprint backlog.

Scrum master: he or she is responsible to move the team to the right direction. Coaching the team to be more agile and more productive.If we have any blocker, he or she is go to person to remove our blocker.

Team: normally scrum development team have 4-9 team members including testers and developers.

Events:

Sprint planning meeting: Sprint planning is a collaborative effort involving a ScrumMaster, who facilitates the meeting, a Product Owner, who clarifies the details of the product backlog items and their respective acceptance criteria, and the Entire Agile Team, who define the work and effort necessary to meet their sprint commitment.

Daily standup : everyday we will have daily stand up meeting to answer following questions:

- What did you do yesterday?
- What are you going to do today?
- Is there any blockers?

  Sprint Review Meeting. In Scrum, each sprint is required to deliver a potentially shippable product increment. ... So at the end of each sprint, a sprint review meeting is held. During this meeting, the Scrum team shows what they accomplished during the sprint. Typically this takes the form of a demo of the new features.

Sprint retrospective meeting : The purpose of the Sprint Retrospective is to improve the practices, teamwork and environment for the next Sprint based on how the previous Sprint went.In the meeting discuss about following topics:

- What went well?
- What did not going well?
- What needs to be improve?

Product backlog- The entire application that we want to develop

Sprint backlog- the list of items that we are going to develop in a specific sprint

4.How do you describe a scrum team?

For me the team is a group of people who are sharing the same goal, moving to the same direction, who trust each other and who will effectively communicate and collaborate with each other to build great product.

5.What is your challenge in scrum ?

Since scrum emphasizes cross functional team (it means developer should be able to test and testers should be able to develop). It is hard to be part of development team as a traditional QA tester. Because generally QAs don’t know how to write code. That is why I have to keep myself very competitive person. Whenever i have time I am learning more coding .

6.How long is your sprint?
In my current project it is 4 weeks / 2 weeks. How long your sprint here? (you can ask back , remember you should not act like an ATM. They generally forget people Only answering questions. Try to make conversation.)

7.What is user story?
In software development and product management, a user story is an informal, natural language description of one or more features of a software system. A user story is a tool used in Agile software development to capture a description of a software feature from an end-user perspective. A user story describes the type of user, what they want and why. A user story helps to create a simplified description of a requirement.

8.What is an epic?
An Epic Can be defined as a big chunk of work that has one common objective. It could be a feature, customer request or business requirement. ... These details are defined in User Stories. An epic usually takes more than one sprint to complete.

9.What is Acceptance Criteria?

Acceptance criteria define what must be done to complete an user story. They specify the boundaries of the story and are used to confirm when it is working as intended . For example, for user story < As a customer I want to be able to search a product so I can buy them.> We can have following acceptance criteria for above user story:

1. Customer should be able to search by product name.
2. Customer should be able to search by product ID
3. User should be able to sort the search result by (hot selling, price,rating,trending)

If all of the conditions are met, then we know the story is successfully developed. We also write test cases based on acceptance criteria.

10.What is parking lot?

In Agile it means this:

In the meeting when you have a problem that is not really relevant to other prople we should not keep discussing that item in the meeting because we are wasting other people’s time.<Let’s make it **parking lot** item> means whoever is interested in that issue can talk after current meeting.

11.What is rat hole?

Since there is a lot of communication going on in agile team, team has to discuss a lot of stuffs. But sometimes the discussion will last too long for one topic and it is not really productive. We will say it is < **rat hole**> it means we should not keep taking about that issue too long and move forward.

12.What are the advantages and disadvantages of Agile Model?

Advantages of Agile Model: 1. Success rate of the project very high compared to any other models. 2. Can adopt changes in requirements at any point of time. 3. Working software is delivered frequently. 4. It emphasizes on responding to change rather than extensive planning and documentation. 5. It is recommended for Product Development.

Disadvantages of Agile Model: 1. Expensive Model as more number of resources are required. 2. Complex in Managing. 3. There is lack of emphasis on necessary designing and documentation. 4. The project can easily get taken off track if there is any communication gap.

13. What is Scrum Model?

Scrum is an iterative and incremental agile software development methodology for managing software development. In this model, System is divided into small parts known as Sprints. The duration of each sprint can range from one week to three weeks. At the end of the sprint, team members and stakeholders meet to assess the progress of the project and identify further plan of action. This assessment helps in taking stalk of the current state and rework the line of work and complete the project on time and not just speculate or predict the further outcome.

14.Why is Agile Model so popular than other SDLC models?

Agile Model is popular because of its flexibility in adopting changes in requirements and at the same time delivering software in shortest possible time. For example, after the sprint review / demo, if client does not like something we can take their feedback and improve the product. **Requirement change is ok.**

Since it is iterative development process, the development team can developed piece of functionality, get feedback and improve next iteration. So the product will be continuously improve.

Waste is eliminated in agile with the help of scrum master. For example, if I am blocked I don’t have to wait and waste my time. Since team members communicates with each other efficiently we can be more productive by preventing duplicated effort.