1: What is software testing?

The process consisting of all life cycle activities, both static and dynamic, concerned with planning, preparation, and evaluation of software products and related work products to determine that they satisfy specified requirements, to demonstrate that they are fit for purpose and to detect defects

2: What is Quality Assurance?

Quality assurance involves use of processes, standards and procedures to ensure quality. It is a bigger superset of testing and tries to prevent defects, whereas testing mainly finds defects

3: Why is testing needed/Why is testing important?

a: To find defects
b: To make sure the product adheres to requirements and has high quality
c: To increase customer satisfaction
d: To find security, performance and usability gaps

4: What is a defect?

A defect is the variation between expected and actual results. For example the requirement is that when user enters valid login and password they should be logged in to the website. However this does not happen, even after entering correct details the user is not logged in so this is a defect.

5: What is difference between error, bug, defect and failure?

A mistake made by the developer is an error
Defect is the difference between expected and actual result
Bug is synonym of defect, most commonly its used for defects found by end users
Failure is when the system fails to perform its required task
>> Error leads to defect/bug leads to failures

6: Why does any system have defects?

a: Incorrect/Incomplete/misunderstood requirements
c: Integration gaps between components
c: Programming Errors
d: Human mistakes

7: What are the principles of testing?

There are 7 principles of testing:

- Testing shows presence of defects
- Exhaustive testing is not possible
- Early testing
- Defect clustering
- Pesticide paradox
- Testing is context dependent
- Absence of errors fallacy

8: What is defect clustering?

Defect clustering means that a small module will contain most of the defects. This is based on the pareto principle which gives 80-20 rule, so 80% of the defects will be present in 20% of the modules (like complex modules, frequently changing modules etc)

9: What is pesticide paradox, and how can you avoid it?

Pesticide paradox is a testing principle that says if we keep using the same test cases again and again eventually it will not find any defects. This is like if a farmer keeps using same pesticide again and again every year the germs will become immune to it.

We can avoid pesticide paradox by frequently reviewing and maintaining our test cases.

10: What is defect cascading?

Defect cascading means that if a defect is unnoticed in the system it will invoke other defects later and result in overall high number of defects in the application.

11: What is exhaustive testing?

Exhaustive testing means verifying the functionality with all possible permutations and combinations, meaning all valid and invalid combinations, preconditions and setup. The principle of testing says that exhaustive testing is not possible, so we should use other methods for full coverage.

12: Why is it impossible to test a program thoroughly?

Any system has large number of test scenarios due to input variations, possible outputs and other combinations. And there is always limited amount of time and cost factor

13: What is early testing, and why is it helpful?

Early testing means we should start with the testing activities at the early stage of STLC itself. This will be helpful in identifying defects early on, which will reduce cost of fixing the defects

14: What are the different methods of testing?

There are three “methods” of testing:
a: Black box testing - Testing without any reference to internal structure of component/system
b: White box testing - Testing based on an analysis of internal structure of a component/system
c: Grey box testing – Mix of black & white where tester has limited knowledge of the internal structure

15: Can you talk about some white box and black box techniques?

White box techniques are based on the internal structure of the system, the code, so we check things like Statement coverage, Decisions coverage, Conditions coverage etc. So basically we are checking the flow of code by viewing the statements, and then if there are any branches like due to decision and coverage we check them all.

Black box techniques do not care about the internal structure of the system; we give inputs and check the output against it and compare it with expected requirements. Black box techniques are based on optimizing test case coverage by boundary value analysis, equivalence partitioning, decision table etc.

16: What are the different levels at which we can do testing?

There are four levels of testing:
a: Unit testing (done at the code module level)
b: Integration testing (done at combination of two or more modules)
c: System testing (done on the entire system)
d: User Acceptance testing (to check product meets end users requirements)

17: What is static and dynamic testing?

Static testing is testing a system without execution of code. Since we are not executing anything the method to use is review or inspection etc
Dynamic testing is testing a system by execution of code. This can be done either by thinking of the system as a non-visible component (black box) or by going into the internals (white box)

18: What are the methods of static testing?

a: Inspection – formal review of code, document, test case etc done in presence of a moderator
b: Walkthrough – formal review of product given to all. Meeting notes taken by scribe
c: Technical review – review of code to find technical gaps
d: Informal review – informal review where documents are sent to all and they provide informal feedback

19: How is UAT different from other stages of testing, especially system testing?

UAT testing is focused more on making sure the product meets the end users requirements, whereas system testing checks the functionality of entire system to find defects.

20: What is risk based testing (RBT), and when you should do it?

Risk based testing is based on the principle of prioritizing the features, on their chances of failure, their criticality to user, and the impact to customer if a failure happens.
RBT is done in cases where there is not enough time in the testing schedule and we have to prioritize things. So instead of just picking out test scenarios randomly, we base them on risk

21: Can you explain functional and non-functional testing with some examples?

Functional testing relates to verifying the functional aspects of the system, example if the user can login to the system or place an order
Non-functional testing relates to the non-functional aspect like how fast this login is, or how secure the website is with user credit card.

22: What is exploratory testing? Why should we do it when we have already done so much other testing?
Exploratory testing is done without any predefined test scripts. The domain experts or testers will do this testing to try out random stuff that can be user behaviour or to try to break the system.

It can be the only option when there are no/incomplete requirement documents. In fact one round of exploratory testing should anyway be done after formal testing is completed to cover unscripted scenarios. This testing helps improve tester's creativity and knowledge of system.

23: What is Alpha and Beta testing?
Alpha testing is the acceptance testing done at developer site. So the end users are called to the developer location for validation.

Beta testing is done at the end-user's site. So the end users try out the application from their own location and report any issues.

24: What are the different types of software testing?

There are several types of testing like – unit testing, integration testing, system testing, acceptance testing, smoke testing, sanity testing, regression testing, performance testing, load testing, usability testing, security testing etc .

25: What is verification and validation?

Verification is evaluating the product at initial phase, to make sure it satisfies specified requirements (are we building the product right?)
Validation is evaluating the end product, to make sure it meets the customer requirements (are we building the right product?)

26: What is the difference between testing and debugging?

Testing is the process of evaluating a system to find defects, and debugging is the process of checking the code to find the root cause of defect. Testing is done by the tester and debugging is done by developer.

27: Can you give some examples of non-functional testing? Why is it important?

Most common examples of non-functional testing are:
a: Performance testing – verifying the response time of system to user's actions, like how much time it takes to login, or how much time it takes the home page of website to load
b: Security testing – checking security aspects like credit card data, vulnerabilities etc., like is the credit card number safely hidden in order placement, password is not visible in URL etc.
c: Compliance testing – if the system confirms to industry standards, like the PCI standard for cards
d: Accessibility testing – checking if system confirms to accessibility rules, like 508 testing for USA

28: What is severity and priority?

Severity is the assessment of a defect in terms of the functional aspect, so how much the defect is impacting functionality. This is given by the testers
Priority is the assessment of a defect in terms of urgency to fix it, and it is given by the business team.

For example,
a: High Severity, High Priority - cannot login to the system, it is high severity because such a critical functionality is impacted, and it is also high priority because it will lead to loss of business.
b: High Severity, Low Priority - If the same login button is broken but on a page deep within the site that is rarely used, then it is high severity for tester but business will give low priority because the page is rarely used. And also there is login option at top of every page.
c: Low Severity, High Priority – If the company logo is missing, or the company CEO name is wrongly spelled, or on credit card page "icici bank" is spelled as "ici bank" then tester will give low severity because it does not impact functionality. But business will say high priority because it can lead to loss of business, or customers will treat side as fraud due to incorrect card name.
d: Low Severity, Low Priority – cosmetic issue, minor spelling mistakes anywhere on terms & conditions page etc.

(Note: always try to answer this question with example, even if the interviewer does not ask you)

29: What is difference between re-testing and regression testing?

When we log a defect and it gets fixed and comes back to us for verification, we re-verify it and that is called as re-testing. Now for the same defect, or any other changes to the application, we also have to verify that it did not break any other functionality that is called regression. So retesting is verifying the defect, regression is verifying everything else.
Example, facebook had a defect in login with mobile number, so checking that defect fix is re-testing and checking other login like with email, username, alphanumeric email still work is regression testing.

(Note: always try to answer this question with example, even if the interviewer does not ask you)

30: What is smoke testing and sanity testing?

Smoke testing is checking the application build for health and stability, so each time the developer gives a new build we will first run a smoke test to quickly make sure the application is stable.
Sanity testing is verifying the main, critical functionalities of the application. This is done to check things quickly without full regression testing, or to do a daily check of production environment.

Example, when testing our website the developer gave a new build so I will run smoke test to quickly check if build is good. And when we deploy changes to production, or before we do full regression testing, we will do a sanity test to verify all critical functionalities. We can also do this testing everyday at 9AM before users start using the application to check the system.
So the coverage is smoke testing >> sanity testing >> regression testing

31: What is A/B testing (or split testing, or bucket testing)?

This testing launches two version of the change in parallel and then compares user feedback to decide which one is better. Example, facebook expanded the like option to include 5 more emojis but this feature was launched to customers in some countries also whereas others were using 1 option only. After checking user feedback, the feature was rolled over to all customers. This is A/B testing.

The aspect of A/B testing impacting testers is that we have to validate both variations are working fine.

32: What is SDLC?

Software development life cycle (SDLC) is the standard product development workflow, for development of applications. Common SDLC models are Agile, Waterfall, V-model, RAD etc

33: What is STLC?

Software testing life cycle (STLC) is the testing process which specific steps to be done in a pre-defined sequence to achieve quality goals

(Note: *Remember the key aspects - STLC is a process not a single activity; it has specific steps which have to be done in proper order not randomly; and lastly it helps to achieve quality results*)

34: What are the stages of STLC?

STLC stages are:

- Requirement Analysis
- Test Planning
- Test Design
- Test Environment setup
- Test Execution
- Test Closure

(Note: *make sure to tell in the same order as above*)

36: What is the requirement analysis phase of STLC?

This is the first phase of STLC and the purpose is to review the requirements, and determine the scope of testing. If there is any gap in the requirements we will get them addressed in this phase.

37: What is Requirement Traceability Matrix (RTM)? How do you do it in your organization?

RTM is a document created to map the requirements and test cases, to make sure there is full coverage.

The simple way to do it is put the requirements and test case ids in an excel sheet and make sure there is coverage. We can also add other details like test case creator, reviewer, date etc.

38: What is a test plan?

The test plan is a document containing the scope, approach, resourcing, schedule etc of testing. If provides each and every detail of testing along with setup details, roles and responsibilities, risks etc. The test plan is created after requirement analysis phase of STLC and serves as the guide for all next stages.

39: What are some of the details present in test plan?

The test plan should contain:

- Brief project details

- Scope - what will be tested
- What will not be tested
- Testing types planned (including non-functional, automation etc)
- Schedule
- Roles & responsibilities
- Risks

40: What is test strategy?

Test strategy is contained within the test plan and provides details on which all testing types would be done on the application.

41: What are some of the most common risks?

a: Not enough time or resources for testing
b: Frequently changing requirements
c: Testing environment not setup or test data not available

42: What is a test case?

A test case is a series of pre-defined steps to validate functionality of system. Here we input the data and verify the actual result matches expected result.

43: Is test case, scenario and script same thing?

Although all three are used for validation, there is some difference.

Test case is a series of steps executed for validation. Test scenario is a broader term and talks of the functionality to be tested, so example login is a scenario and we will create 10 test cases to verify it.

Test script is written in programming language to verify the functionality.

44: How can you write good test cases?

Some factors are:
a: The test case should be based on requirements and not assumptions
b: It should provide full coverage with testable steps
c: It should be written clearly
d: It should be written with user behaviour in mind
e: We should get the test case reviewed to make sure it satisfies all above criteria

45: How can you ensure full coverage in your test cases?

Some factors are:
a: Read the requirement documents thoroughly
b: Use techniques like boundary value analysis, equivalence partitioning etc
c: Cross check by requirement traceability matrix (RTM)
d: Get the test cases reviewed in case someone has missed something

46: What is equivalence partitioning?

Equivalence Partitioning is a method to get maximum possible coverage with minimum number of test cases. We divide the input set into partitions and take one input from each partition.

47: What is boundary value analysis?

Boundary value analysis is a method to get maximum possible coverage with minimum number of test cases. We verify the boundary conditions of any input, testing the below and above edges of data.

48: Can you give an example of using EP and BVA on date of birth data?

Lets suppose the field accepts year from 1900-2020 so we will verify:
EP – 1890, 1950, 2025
BVA – 1899, 1900, 2020, 2021

49: Can you give an example of using EP and BVA on any quantity box?

Lets suppose the field accepts quantity from 5-100 so we will verify:
EP – 2, 50, 110
BVA – 4, 5, 100, 101

(Note: likewise you can prepare other examples also. If interviewer just asks you to give example of EP and BVA, first mention the scenario, that is, let's think of a quantity box on ecommerce website that accepts quantity 5-100 so we the EP and BVA values will be……..)

50: What is decision table testing, and can you give any example?

In decision table testing, we list the different combinations of inputs in a column and outputs below it. This way we are able to design scenarios around all possible input combinations.
Let's say we have the login box on website, the decision table will be:

Login	Y (Valid)	Y (Valid)	N (Invalid)	N (Invalid)
Password	Y (Valid)	N (Invalid)	Y (Valid)	N (Invalid)
Output →	Login successful	Cannot login, Error message	Cannot login, Error message	Cannot login, Error message

51: What is use case testing, and can you give any example?

This testing involves designing use cases. The use case is an interaction between application and users (actors). So this testing is closer to user behaviour.
Example, for an ecommerce website, we can think of order placement use case in which the user logs in to the system, searches for any item like laptop, adds to cart, checkout and place order.

52: Write some test cases for login flow?

a: Verify login with valid username and password
b: Verify login with email id and password, using enter key
c: Error message scenarios like valid username but invalid password etc
d: After some number of incorrect attempts, account should lock *(Security testing)*
e: Error message if both fields left blank
f: Verify password should not be visible, it should show in asterisks *(Security testing)*
g: After password change, old password should not work
h: After entering details and click on cancel, it should not login and clear out data
i: User should be able to login fast *(performance testing)*
j: If cookies cleared or user is idle for long time, it should ask login again, and the password data should not be pre-filled

(Note: when giving example you can also mention the category, like this is functional test, this is security test, this is performance test etc.)

53: Write some test cases for Amazon search box?

a: Verify valid search criteria – alphabets, numeric, alphanumeric etc – by click and enter key both
b: Predictive search functionality, like type “lap” it should start showing about laptops
c: Auto-correction, like type “pancil” it should show results for “pencil”
d: Recent search history should be retained, and click on it should take to that search results
e: If no result found, appropriate page should show
f: Search history can be cleared
g: Search box should show at top of page
h: Search box should show correctly on mobile site, and other scenarios should work
i: Search response time should be good *(performance testing)*
j: If I type “my order” in search box, does it take to order history page?

(Note: there can be several other scenarios also, there is no fixed answer. Try to mention the common, user-based scenarios first, and try to keep a mix of functional, non-functional, security scenarios)

54: Write some test cases for pen?

a: Verify the pen is writing things correctly, we can write fast, slow, straight, curve etc.

b: Verify the pen colour is correct, if it says blue pen it is blue
c: Verify the pen material, size etc per specification
d: Verify the company name on pen is written correctly
e: Verify the pen cap can be put or removed correctly
f: Verify the pen does not leak in any position (negative testing)
g: Verify the pen does not break on dropping from short height
h: Verify we can open the pen and put another refill

(Note: there is no fixed, correct answer to this question, the interviewer is only trying to check your analytical skills. He/she can also ask you for test cases of pencil, telephone, lift, chair etc. Think for a minute, about what the product does, and then answer. Try to keep a mix of positive, negative, capacity, usability scenarios etc)

55: What is TDD?

TDD or test driven development is a software development methodology in which development of application is driven by test cases. So we first create test cases and based on that the developer writes their code. This helps to make sure all scenarios are covered in code.

56: What is BDD?

BDD or behaviour driven development is a software development methodology in which test cases are written in plain, clear English keeping in mind the end-user's perspective. So it's a modification from TDD in which test cases can be technical, but BDD keeps things non-technical by focusing on end user.

57: What are some of the fields/attributes in a test case?

A test case should contain some of the following fields:

a: Test case id, the unique identifier of test case
b: Name
c: Description
d: Pre-condition
e: Test step
f: Expected results
g: Creation date
h: Created by
i: Any screenshot attached (this will help in UI scenarios)

58: How can you make sure that your test cases always find defects?

This can be done by several steps. First, the test cases should be designed as per the requirements, so all requirements are covered. Also we can use boundary value analysis, equivalence partitioning etc to ensure full coverage. Once test cases are created they should be reviewed and frequently maintained also to make sure they are up-to-date on changes.

59: What are some common test management tools? If you were to choose a test management tool for your company, what factors you will consider?

Some of the common test management tools are ALM, Qtest, TFS etc.
The choice of test management tool depends on features, cost, usability, training needs, integration with other tools like automation, defect management tool etc.

60: What is test data setup? Why do we need test data? How can we get test data?

Test data is set of input values to execute test cases. We need test data so we can verify both positive and negative scenarios. There are several ways to get test data like we can create it on test environment manually or by automation, we can copy production data to test environment but hiding privacy data like card number, password etc. We can also use database query languages to fetch test data.

61: What is the Test Execution phase in STLC?

The test execution phase is when team starts running the test cases to validate functionality. Based on the outcome the tests may be pass, fail and we will log defects. So this is the action phase of STLC, where all run time things happen.

62: What will be different status when you will execute test cases?

The statuses are:
a: Pass – when test case actual results meet expected results
b: Fail – when test case actual results does not meet expected results. We should log defects
c: Incomplete – when there is any query such as with requirement
d: Blocked – when testing cannot be done due to some defect, or test data, or environment down
e: Deferred – when test case cannot be done in current release but it will be ready in future
f: Unexecuted – test case has not been run yet

63: When you are running a test case and things do not work per expectation what will you do?

We will fail the test case and log a defect. We will assign the defect accordingly to correct person as per company rules. If there is more than 1 test case affected we will block the remaining test cases. However we will not update to fail or block just blindly, we will still try to execute other steps if we can, so we find defects early.

64: What are the different details given when logging defects?

The most common fields are:
a: Defect description
b: Environment or version found on
c: Severity
d: Steps to replicate with any test data used
e: Any screenshot or video to explain the issue
f: Assignee
g: Status

65: How can you make sure to log correct defects?

a: We should provide clear steps, including any test data needed
b: Provide any screenshot or video that helps to identify the issue
c: Assign severity and priority correctly
d: Do not log duplicate defects or put multiple issues in 1 defect

66: There is a defect that developer says they cannot reproduce (non-reproducible), what will you do?

We should first check if the defect steps and test data is correct. If possible we should provide any videos or screenshot that helps to reproduce. If still the problem persists, we should reach out to the developer via email/chat/in person and work with them to demo the defect.

67: What are the different severity/priority types for defects?

The most common values are:
a: Blocker – user is completely blocked on functionality and there is no work-around, example cannot place order
b: Critical – severely blocking user, but there is a work-around available that user can use, example cannot place order with COD but credit card is working
c: Major – affecting progress of user but impact less than critical, example product image not coming
d: Minor – minor problem and very easy workaround present, example wrong alignment of quantity box
e: Trivial – almost negligible or no impact on user, example spelling mistake on terms & condition page

68: What is the defect life cycle?

A defect goes through several statuses after it is reported by a tester. This makes up the defect life cycle. The stages are:

a: New – when the defect is logged
b: Assigned – when the defect is assigned to business or development team
c: Rejected – if business or development team think defect is not valid or deferred
d: Open – when the defect has been accepted valid and assigned to developer
e: In Progress – when developer starts working on defect
f: Ready to test – the developer has fixed the defect and given to tester for re-testing
g: Closed – tester verifies the defect and its working fine
h: Reopened – tester verifies the defect and it still exists

69: What are some of the defect management tools?

Common defect management tools are – JIRA, Bugzilla, Mantis etc

70: What is defect density?

Defect density is a way to measure the quality of an application. It is calculated by dividing number of defects with size of application (lines of code or number of requirements etc)

71: What is the Age of Defect?

Age of defect is the time gap between the day it was logged, and the day it got fixed. It is used as a measurement of test effectiveness.

72: What are the effects of finding a defect late (OR) Why is it costly to fix defects later in the STLC cycle?

We need to find defects as early as possible because it is easier, cheaper and quicker to fix them. Example, if we find defects in requirement analysis phase we only need to update the requirement documents, but if same is found in testing phase, fixing it will need re-analyze of requirement documents, coding, testing etc. If found on production, it will need all above and also it can cause loss of business, brand damage and customer complaint. So it’s best to test as early as possible and fix defects as early as possible.

73: What is latent and masked defect?

A latent defect is any defect that has existed in system but never caused a failure because its conditions were not met.
A masked defect is an existing defect that has not surfaced yet because another defect has prevented that part of code to be executed.

74: When should we stop testing?

Testing can be stopped when:
a: All planned test cases are executed with certain percentage passed
b: No blocker or critical defects are open
c: Low defect detection rate
d: Deadline reached

75: What is performance testing?

This type of testing measures various performance metrics of the system such as response time, transactions per minute etc.

76: What are load, stress and volume testing?

All these are types of performance testing, but they are different.
Load testing – checking application’s performance at expected load, like 10,000 users
Stress testing – checking application’s performance at more than expected load, like 11-12,000 users
Volume testing – checking application’s performance with large amount of data.

77: What is the purpose of entry and exit criteria?

In STLC we cannot start or end a stage whenever we want, that is why entry and exit criteria are needed. The purpose of entry criteria is to determine when a stage can be started, and exit criteria is to determine when it can be called completed.
Example for test design phase the entry criteria is freezing of requirements, and exit criteria is end of test case creation, review, test data setup etc.

78: What does a test report contain?

A test report should, at minimum, contain:
a: Details of test execution – run%, pass%, fail% etc
b: Details of defects – open defects, by priority/assignee etc
c: Any risks or show-stoppers

79: What are the attributes or features of a good test report? Who should it be sent to?

A test report should be:
a: Complete – provide complete information
b: Accurate – do not hide facts, send as is
c: Readable with most relevant information, not too much details

d: Sent periodically

The report should be sent to all stakeholders of project such as business managers, development managers, system admins etc

80: What are test closure activities?

Test closure activities are done at end of testing cycle, like checking completion of artefacts, archiving all documents, retrospective of entire testing cycle etc. It is important to preserve these details for future reference in case same project comes up again or any similar project comes.

81: What is the meaning of release phase?

After all testing is completed, the new changes or bug fixes are pushed to production server, so it can be used the end-users. That is called as product release.

82: Is release done by the testing team (OR) what is the work of testers when application is released to production?

All teams - business, development, system admin, testers – participate in the product release phase. Tester's job is to verify the new changes or bug fixes on production, and also to do a sanity testing of entire application.

83: What is difference between build and release?

A build is the developed application that is given by development team to testing team for validation. A release is the official launch of changes to production servers so end users can use it.

84: What is automation testing?

Automation testing is the use of tools and scripts to verify the system. Instead of writing test cases manually, we are using coded scripts. And instead of executing manually we are doing the execution via a tool.

85: What are the advantages of automation testing?

a: Faster testing, faster release cycles
b: Create once, execute multiple times
c: Cost saving
d: Increase in coverage and accuracy
e: Freeing manual resources for other work

86: What are the advantages of manual testing over automation testing?

a: Low initial setup cost
b: Adaptable to frequent changes
c: Real user experience

(Note, you must mention here that it's not like one approach is better and other is not. Testing should be a mix of both manual and automation strategies)

87: Should we automate everything?

It's best to automate scenarios that are redundant, too much time consuming and not frequently changing. Regression test cases are best examples of these, so we first pick our regression test cases for automation

88: What all things you should consider before selecting automation tools?

The selection of automation tool depends on:
a: application to be automated (web, windows, both)
b: testing requirements
c: cost and benefits
d: available of skilled resources
e: scalability and reusability
f: reporting options

89: What is Automation framework?

Automation framework is the structuring and other guidelines, to make automation efficient, effective and consistent. It allows easy maintenance and promotes reusability.

90: What are some components of Automation framework?

Automation framework has components for:
a: Configuration (config file)
b: Common libraries and utilities
c: Test Scripts
d: Object repository
e: Test data repository

91: What is Agile testing?

Agile testing is the practice followed by testing team in Agile methodology. Agile is fast changing, incremental development model so testing activities are done in parallel to development.

92: What are the qualities needed to be a good Agile tester?

Agile involves frequent changes and faster delivery cycles. So a tester should be able to understand requirements quickly and prioritize their tasks as per sprint capacity. It’s also important to communicate effectively with team and raise any risks.

Agile is not just a methodology but it’s a mind-set, so to be effective in Agile the tester should be agile in his/her thought process also.

93: How is agile different from waterfall?

a: In agile, the product development is done in small cycles called sprints whereas in waterfall the product development is done as a whole application at once.
b: In agile, testing activity is done in parallel to development whereas in waterfall testing is a separate stage that is done after development stage is completed
c: Agile is more flexible to changes and any new requirement can be accommodated quickly, whereas in waterfall any change will travel back to all earlier stages and then implemented.

94: How do you deal with frequently changing requirements in Agile?

As tester, we will need to work very closely with product owner or business analyst to clarify the requirements, and then update our test cases. We also need to be flexible because requirements can change frequently and there might not be sufficient time. We should also raise any risks to the team in advance.

(Note: there is no fixed answer to this question; the interviewer is just checking your flexibility and openness to changes)

95: Who are the main actors in Agile?

a: Product owner – the business owner, creates requirements. Manages the backlog and prioritizes it
b: Scrum master – ensures team is able to work properly as per agile and not blocked. Maintains synergy in whole team
c: Scrum team – developers, testers etc in the team. They are involved in the development and testing work.

96: What are Agile ceremonies?

a: Sprint planning – this is done before start of each sprint where team discusses backlog and pulls in work to be done in sprint, known as sprint backlog
b: Stand-up – short meeting each day to discuss progress and any blockers
c: Backlog grooming – this is done couple days before start of next sprint to understand the requirements to be taken in next sprint and resolve any queries early on
d: Sprint demo – the product developed in a sprint is demoed to senior management, who can suggest any changes to be done (which will be taken in next sprint)
e: Retrospective – done after each sprint, to discuss what went well in the sprint, and what can be improved.

97: What is Product Backlog & Sprint Backlog?

a: Product backlog – requirements created by the product owner for the entire product
b: Sprint backlog – a subset of product backlog that is taken by the team to be done in a sprint

98: What is burn-down and velocity in Agile?

Burn-down charts are used to track of progress of sprint, it shows how much we are closing requirements in the sprint and how much work is pending. Velocity is calculated by adding all the points of stories in a sprint. It shows how much work team can do in a sprint.

99: Give some example of story points in Agile?

We can use Fibonacci series for pointing – 1,3,5,8,13,21 etc
1 – very small change, like a sentence change, or button name change
3 – small change but bigger than 1, like adding a field to a form, or adding validation error messages to an entire form
5 – Integrate form to backend database
8 – Change the shopping cart layout on amazon (this can be multiple small point stories)
13 – Big change, redesign entire checkout flow of amazon (this should be broken into small point stories)

100: What is jira tool? Why is it used?

JIRA is a project management and defect management tool. It can be used for project tracking, product implementation, defect logging and tracking, time tracking etc.

101: Explain the JIRA workflow.

The jira workflow is like the issue life cycle. It contains status and what is the flow from one status to another.

102: What are the different issue types in jira?

a: Story – a new requirement
b: Epic – a big product requirement from which story is created
c: Task – new requirement but smaller effort than story
d: Bug – defects logged by testers
e: Sub-task – individual (developer, tester etc) work in a story or task