

Real Time – Interview Questions on Manual Testing

1.What is Software testing?

Software testing is nothing but to verify and validate a software product's functionality. Basically, it checks whether the software product matches client requirements and makes sure it is defect-free. It can be said that testing enhances the quality of the product by preventing bugs, reducing development costs, and reducing performance issues.

2.What are the different levels of testing?

There are mainly four testing levels and they are:

- ✓ Unit Testing
- ✓ Integration Testing
- ✓ System Testing
- ✓ Acceptance Testing

Basically, it starts with the Unit Testing phase and ends with Acceptance Testing.

3.Tell me difference between test case and test Scenario?

Test Case	Test Scenario
1.Test cases focus on “what to test” and “how to test”	1.Test scenarios just focus on “what to test”.
2.Test cases can be derived from test scenarios. They have many to one relationship with the test scenarios	2.Test Scenarios are derived from use cases.
3.Test case means detailed documenting the cases which help executing while testing.	3.Test Scenario means talking and thinking requirements in detail.

4.What is quality audit?

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

5.Have you written Test Strategy?

In our organization test strategy document will be prepared by Test Managers or Project Managers. I never got change but I know what is test Strategy doc.

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

Test Strategy is a high-level document (static document) and usually developed by project manager.

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

7.What will be the first approach if we get bugs in every build?

First, I need to log the bug with proper step to reproduce the bug even I need to attach a screen shot where I get the bug. According to test case and functionality of module we need to set the priority of test case.

8.How many Test cases did you write for last project?

In my current project nearly 170 test cases I have been written for payment/e-com/telecom billing application module.

9.How much time is required to write a Test case?

That depends on complexity of the functionality. First, we need to understand the user story points Then we need to create test date and test scenario based on that we are going to start developing test cases. In regular hrs.' of time I nearly write 45 to 50 test cases.

10. How many defects did you detect in your last project?

In my last project I executed nearly 170 Test cases, in which some 22 defects were raised including 3 showstoppers.

11.Did you face any problems during defect reporting and tracking?

Yes, recently Developers rejected 2 or 3 valid defects. But I always keep screen shot of my defect so it Would help to present my defect with team.

12. Tell me once difference between waterfall module and agile methodology?

The cost of fixing defect in waterfall model is more but cost of fixing defect in agile methodology is less. Waterfall model is liner sequential life cycle model where as agile is continuous iteration model.

13. Tell us end-to-end testing process? Or what is mean by end-to-end testing?

End to End testing is the process of testing a software system from start to finish. The tester tests the software just like an end-user prospective. In end-to-end testing, the software is tested along with all its dependencies and integrations, such as databases, networks, file systems, and other external services.

14.What is a test environment?

A test environment consists of a server/computer on which a tester runs their test case. It is different from a development environment and tester tries to represent the actual hardware on which the software will run once it is in production.

Whenever a new build of the software is released, the tester updates the test environment with the latest build and runs the regression test suite. Once it passes, the tester moves on to testing new functionality.

15.Explain black-box testing, white-box testing?

Black-box testing in software testing: In black-box testing, the system is tested only in terms of its external behavior .it does not consider how the software functions on the inside.

This is the only limitation of the black-box test. It is used in Acceptance Testing and System Testing.

White-box testing in software testing: A white-box test is a method of testing a program that take an responsibility of its internal workings code as part of its review. It is used in integration testing and unit testing.

16.What is defect , Bug and Error?

Defect: A Defect is a deviation from the Requirements. A Software Defect is a condition in a software product which does not meet a software requirement or end-user expectations.

Bug: A Bug is the result of a coding Error or Fault in the program which causes the program to behave in an unintended manner.

Error: Error is a human action that produces an incorrect result. It is deviation from actual and expected value. The mistakes made by programmer is known as an 'Error'.

17.When we choose Informal Testing?

Whenever we don't have proper documentation (Requirements etc...) and sufficient Time then we choose Informal Testing.

Using Experienced based Techniques (Ex: Error Guessing, Exploratory Testing etc...) we conduct Testing.

18. Do You have experience in Database Testing?

Yes, I conducted Database Testing Manually using SQL Commands,

I Tested the following Database operations during Database Testing,

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

19.Did you create RTM(Requirement Traceability Matrix) document in your project?

I only updated the RTM document because my team lead provides us RTM format and we just updated it.

20. What are the important phases in Formal Software test process or Software testing Life Cycle?

- ✓ Requirement Analysis.
- ✓ Test planning.
- ✓ Test case design.
- ✓ Environment set up.
- ✓ Test execution.

✓ Test closure.

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

SRS describes what client is expecting from the system. For example in case of Gmail SRS consists details like first page should be login, to access mail box user should be authenticated. FRS describes how above requirements will be developed and the functionality in FRS will be written down in more technical terms.

22.Is the testing team involved in SRS preparation?

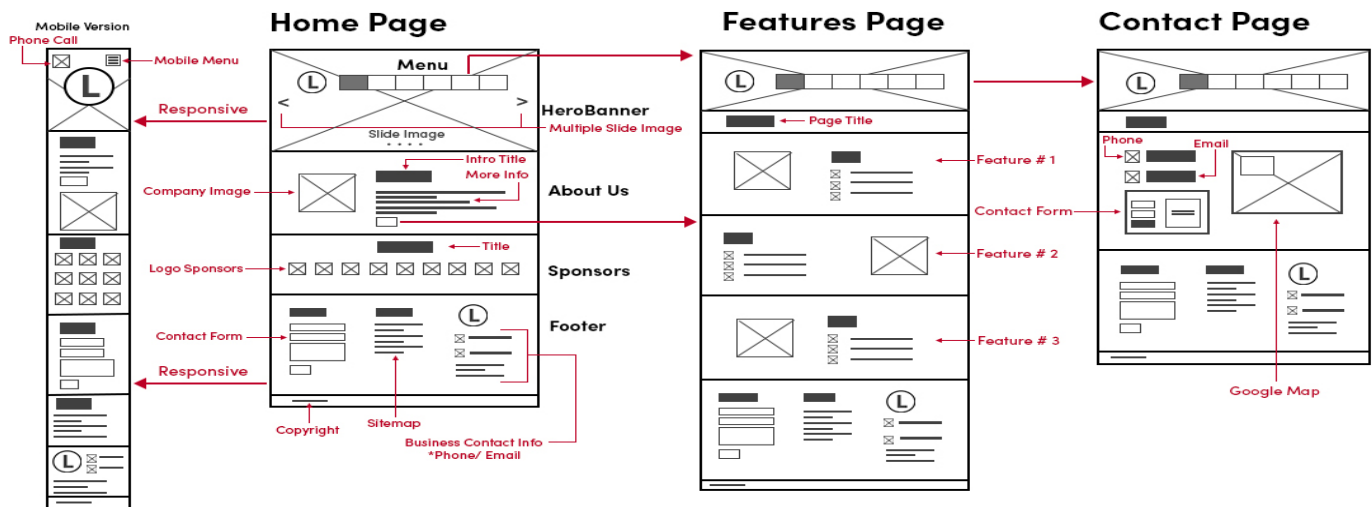
Business analyst prepare the SRS document by interacting with the client. However, a senior testing team member can also be involved in requirements collections along with the development team and the business analyst team.

23.How does your requirements document look like?

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

24.What is wireframe?

A diagram which stimulates the feel of the actual screen.



25.What is use case?

A use case in software testing is a detailed description of how a user interacts with a system to achieve a specific goal. It outlines the steps, inputs, and expected outcomes, helping testers validate the system's functionality from the user's perspective.

26.How long do you spend on understanding user story ?

The time spent on understanding a user story depends on its complexity. Typically, it ranges from 30 minutes to a few hours, including discussions with the team, asking clarifying questions, and analyzing acceptance criteria to ensure a clear understanding.

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

I should have an overview of the whole project and detailed screen and field level understanding of the assigned functionalities.

28.What is Review?

A **review** in software testing is a process where the project artifacts (like requirements, design, code, or test cases) are examined by team members to identify defects, inconsistencies, or improvements. It helps ensure quality and completeness before moving forward in the development or testing process.

29.What is peer review?

Is a review of a software work product by colleagues. One or two member of software testing Colleague perform peer review.

30. Tell us difference between verification and validation?

Verification	Validation
Verification is a static practice of verifying documents, design, code and program.	Validation is a dynamic mechanism of validating and testing the actual product.
Verification uses methods like inspections, reviews, walkthroughs, and Desk-checking etc.	Validation uses methods like black box (<i>functional</i>) testing, gray box testing, and white box (<i>structural</i>) testing etc.
Verification is to check whether the software conforms to specifications.	Validation is to check whether software meets the customer expectations and requirements.

31. What are the entry criteria for release?

System testing results must show that all requirements are completed and project is stable.

- ✓ Alpha and beta testing must be completed.
- ✓ All medium and above severity bugs must be fixed.
- ✓ The release package is available.
- ✓ The release CD label is ready.

32. What is the release process you follow?

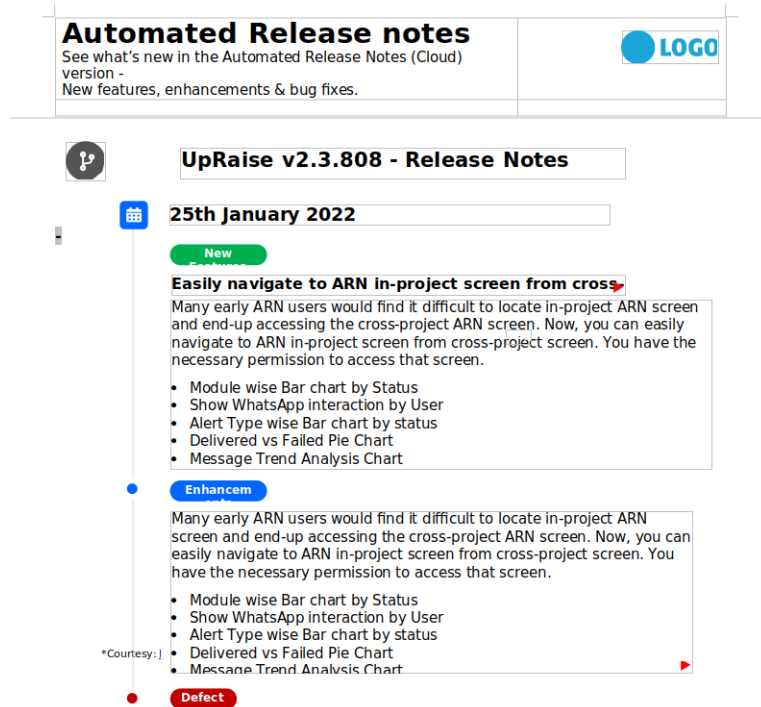
In our organization, the release process is co-ordinated by a release manager. After successful beta testing, the release manger sends an email to all stake holders (development manager, test manager, documentation manager) for their Approval for final release. The test manager further forwards the same mail to team members requesting their internal approval. Based on internal approval. The test manager can send approval to the release manager.

33. What is your role in the release process?

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

34. What is software release note?

A **software release note** is a document shared with stakeholders detailing the new features, enhancements, bug fixes, and known issues in a software version being released.



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

A build is ready for testing when it passes **smoke testing**, meets the entry criteria, and is stable enough for further functional or detailed testing.

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

The number of test cases executed per day depends on their complexity, the system's stability, and the type of testing. On average, **10-50 test cases** can be executed in a day for manual testing, while automation may cover more.

37. How do you check broken links?

To check for **broken links**, you can:

1. **Manually:** Click each link and verify it navigates to the correct page without errors.
2. **Automation Tools:** Use tools like Selenium, Broken Link Checker, or Screaming Frog to detect links returning errors like 404.
3. **Developer Tools:** Inspect network responses in browser dev tools for HTTP errors

38. What is test log?

A **test log** is a detailed record of test execution activities, documenting information such as test cases executed, results (pass/fail), defects found, environment details, and timestamps. It helps track testing progress and analyse outcomes.

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

Yes, observing **application logs** during test execution helps identify errors, exceptions, or unexpected behaviour in the application. Logs provide insights into back-end processes, aiding in debugging and verifying system performance.

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

No, I did not run regression test cases for every bug fixed. I run regression tests once for every build.

41. How do you consider that review was successful?

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

42. What is code review?

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

43.What is a code Freeze?

Code Freeze is a stage in the development process where no further code changes or additions are allowed, except for critical bug fixes. This ensures stability and prepares the build for testing or release.

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

Entry criteria:

- ✓ Coding should be completed.
- ✓ Test cases should be ready and base lined.
- ✓ RTM should be updated.
- ✓ Test data should be read and base lined.
- ✓ Test environment/set up should be ready.
- ✓ S/w tools should be ready and approved.

Exit criteria:

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

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

To ensure data is stored in the correct tables and columns:

1. **Verify through Database Queries:** Check the database using SQL queries to confirm the data is saved in the correct table and column.

2. **Use Logs or Debug Tools:** Monitor application logs to trace data flow and storage.
3. **Backend Validation:** Use tools like Postman or API testing to verify database operations.
4. **UI-DB Mapping:** Ensure field-to-column mapping is as per requirements or design document

46. What is test case?

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

47. Where do you write test case?

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

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

We follow two step approaches to ensure that test cases are completed.

1. Reviews-- it ensures that quality of the test cases is good.
2. Requirement traceability matrix ---it ensures that all requirements have been covered through test cases.

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

A **good test case**:

- Covers all the requirements and acceptance criteria.
- Has clear, concise steps and expected outcomes.
- Is reusable, traceable, and easy to understand.
- Identifies defects effectively with minimal overlap.

A **bad test case**:

- Is ambiguous or lacks sufficient detail.
- Fails to test the intended functionality.
- Misses edge cases or negative scenarios.
- Is redundant or not maintainable.

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

The percentage of **positive and negative test cases** depends on the application's requirements and risk assessment.

- **Positive test cases:** ~60-70%, focusing on validating expected functionality with valid inputs.
- **Negative test cases:** ~30-40%, covering edge cases, invalid inputs, and error-handling scenarios to ensure robustness.

This balance may vary based on the application's criticality and complexity.

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

Yes, **test cases** should be updated after receiving a new build if there are changes in the application's UI or functionality. This ensures that the test cases remain aligned with the latest version of the application and cover any new or modified features.

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

A **positive test case** verifies that the application behaves as expected when provided with valid inputs or scenarios. It checks if the system handles typical use cases correctly.

A **negative test case** ensures that the application correctly handles invalid inputs or error scenarios, confirming the system's robustness and error-handling capabilities.

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

A **test manager** considers several parameters before deciding to stop testing, including:

1. **Test completion criteria** (e.g., all critical test cases executed).
2. **Test coverage** (sufficient test cases for all functionalities).
3. **Defect severity and number** (no critical defects open).
4. **Time and resource constraints** (testing time has been exhausted).
5. **Risk analysis** (no major risks identified).
6. **Test progress** (test execution is on track with expected results).

54. What is the difference between Front End Testing and Back End testing?

Front-End Testing focuses on validating the user interface (UI) and user experience (UX). It ensures the application's design, functionality, and responsiveness are correct, and it interacts with users as expected.

Back-End Testing focuses on the server-side of the application, ensuring that databases, servers, and APIs function correctly. It verifies data processing, storage, and retrieval operations, typically through database queries or API testing.

55. What is the difference between System Testing and Functional Testing?

System Testing is the process of testing the entire application to ensure that all components work together as expected. It evaluates both functional and non-functional aspects like performance, security, and compatibility.

Functional Testing focuses specifically on verifying that the application's features and functionalities work according to the specified requirements. It tests individual features or modules without considering non-functional aspects.

56. What is Exhaustive Testing?

Exhaustive Testing is testing all possible inputs and scenarios, which is often impractical for large systems due to time and resource constraints.

57. What are the Test design techniques that you used?

Some common **test design techniques** include:

1. **Equivalence Partitioning** – Dividing input data into valid and invalid partitions.
2. **Boundary Value Analysis** – Testing the boundaries of input ranges.
3. **Decision Table Testing** – Using decision tables for combinations of inputs.
4. **State Transition Testing** – Testing different states and transitions in the system.
5. **Exploratory Testing** – Exploring the application to find defects without predefined test cases.
6. **Use Case Testing** – Verifying the system based on use case scenarios.

58. How you communicate with Developers to resolve issues?

I communicate with developers by providing clear steps to reproduce issues, collaborating to identify the root cause, using issue tracking tools like JIRA, prioritizing defects, and verifying fixes through testing.

59. What are the important Test Types that can be applied for Web Applications?

Important **test types** for web applications include:

1. **Functional Testing** – Verifying the functionality of the application.
2. **Usability Testing** – Ensuring a user-friendly experience.
3. **Performance Testing** – Checking load and stress handling.
4. **Security Testing** – Identifying vulnerabilities and ensuring data protection.
5. **Compatibility Testing** – Verifying compatibility across different browsers, devices, and operating systems.
6. **Regression Testing** – Ensuring new changes don't break existing functionality.
7. **Cross-browser Testing** – Ensuring the application works on all supported browsers.

60. Explain the process of QA testing?

QA testing involves analyzing requirements, planning tests, designing test cases, setting up the test environment, executing tests, reporting defects, performing regression testing, and closing the process with reports and evaluations.

61. What is functional testing?

Functional Testing is the process of verifying that the application's features work as per the specified requirements. It focuses on testing the functionality of the software without considering the internal code structure, checking if each function operates correctly.

62.If a product is in the production stage and one of its modules gets updated, then is it necessary to retest?

Yes, it is necessary to **retest** if a module is updated in a product that is already in the production stage. This ensures that the changes do not introduce new defects and that the updated module integrates well with the rest of the system. This process is typically referred to as **regression testing**.

63. What is Compatibility Testing?

Compatibility Testing ensures that a software application works as expected across different environments, including various browsers, operating systems, devices, and hardware configurations. It checks for consistent performance and appearance, ensuring compatibility with all target platforms.

64. What is Security Testing?

Security Testing is the process of evaluating an application to identify vulnerabilities, threats, and risks. It ensures that the system is protected from unauthorized access, data breaches, and attacks, and verifies that data privacy and security measures are in place.

65. What do you mean by regression and confirmation testing?

Regression Testing involves re-running previously executed test cases to ensure that new changes or updates to the software have not introduced any new defects or broken existing functionality.

Confirmation Testing (also called **Re-testing**) refers to running tests on a specific defect after it has been fixed to confirm that the issue has been resolved and no new problems have been introduced.

66. Purpose of Bug Life Cycle?

The **Bug Life Cycle** tracks the stages a defect goes through from identification to resolution. Its purpose is to:

1. **Track Progress** – Monitor the status of bugs from discovery to closure.
2. **Ensure Transparency** – Provide clear visibility to developers, testers, and stakeholders.
3. **Prioritize Fixes** – Help prioritize bugs based on severity and impact.
4. **Improve Quality** – Ensure issues are resolved and verified, leading to higher-quality software.

67. What is Verification?

- ✓ Verification is the process of evaluating the artifacts as well as the process of software development in order to ensure that the product being developed will comply with the standards.
- ✓ “Are we building the product, right?” process oriented.

71. What is a test bed?

A **test bed** is a controlled environment that includes the necessary hardware, software, network configurations, and tools required for testing. It ensures that the application is tested in a stable and consistent setup, replicating real-world conditions for accurate results.

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

Bug Leakage occurs when a defect is not detected during the testing phase and is found by the end user in the production environment.

Bug Release refers to a defect that is intentionally released to the user because it is deemed to have low severity or impact, or it may be scheduled for fixing in a future update.

75. What is Defect Cascading?

Defect Cascading occurs when a single defect causes a chain of subsequent defects or failures in other parts of the system. This happens because one issue in the software affects other interconnected components, leading to multiple defects being discovered in the process.

76.What is the difference between sanity and smoke testing?

Smoke Testing: checks if the basic functionality works after a new build, ensuring stability.

Sanity Testing: verifies specific functionality or bug fixes to ensure they work as expected.

77.What are the different types of smoke testing?

The different types of smoke testing include:

1. **Build Verification Smoke Testing** – Verifying the stability of the build by checking critical functionalities.
2. **User Interface Smoke Testing** – Ensuring that the basic UI elements (buttons, links, etc.) work.
3. **Feature Smoke Testing** – Testing specific features or modules of the application for correctness.
4. **Regression Smoke Testing** – Verifying that core functionalities remain unaffected after new changes or bug fixes.

78. What is Reviews, Walkthrough And Inspection In Software Testing?

Review: Informal examination of documents or code to find issues.

Walkthrough: Author explains the document or code to the team for feedback.

Inspection: Formal and structured process for identifying defects in work products.

79.What means cross browser testing?

Cross-browser testing involves testing a web application across multiple browsers (e.g., Chrome, Firefox, Safari, Edge) to ensure that it behaves consistently and appears correctly on all of them. It verifies that the application functions as expected regardless of the browser used by the end user.

80.What is performance testing?

Performance Testing is the process of evaluating how well a software application performs under various conditions. It measures factors like speed, responsiveness, stability, and scalability to ensure the application can handle the expected load and usage without performance degradation

81.What are some attributes of a test case? Or Test case formate?

Some key **attributes of a test case** include:

1. **Test Case ID** – Unique identifier for the test case.
2. **Test Case Description** – A brief description of the test objective.
3. **Preconditions** – Any conditions that must be met before the test starts.
4. **Test Steps** – Detailed steps for performing the test.

5. **Test Data** – The input data required for the test.
6. **Expected Result** – The expected outcome or behavior.
7. **Actual Result** – The actual outcome observed during testing.
8. **Pass/Fail Status** – Whether the test passed or failed.
9. **Priority** – The importance of the test case.
10. **Test Environment** – The environment where the test is executed (e.g., OS, browser).
11. **Comments** – Additional notes or observations during testing.

82.What are some defect reporting attributes? Or Defect report format?

Some common **defect reporting attributes** or **defect report format** include:

1. **Defect ID** – A unique identifier for the defect.
2. **Title/Name** – A brief description of the defect.
3. **Description** – Detailed explanation of the defect, including steps to reproduce.
4. **Severity** – The impact level of the defect (e.g., Critical, Major, Minor).
5. **Priority** – The urgency for fixing the defect (e.g., High, Medium, Low).
6. **Test Environment** – The environment where the defect was found (e.g., OS, browser version).
7. **Status** – The current status of the defect (e.g., Open, In Progress, Fixed, Closed).
8. **Assigned To** – The person or team responsible for fixing the defect.
9. **Reported By** – The person who discovered the defect.
10. **Steps to Reproduce** – Detailed steps that can recreate the defect.
11. **Expected Result** – The correct behavior or outcome.
12. **Actual Result** – The observed behavior or outcome.
13. **Attachments** – Screenshots, logs, or videos related to the defect.
14. **Date Reported** – The date the defect was logged.

83.What is defect density?

Defect Density is a metric used to measure the number of defects (bugs) found in a software product relative to its size, usually calculated per thousand lines of code (KLOC). It helps assess the quality of the software and the effectiveness of testing efforts.

Formula:

$$\text{Defect Density} = \frac{\text{Number of Defects}}{\text{Size of the Software (in KLOC)}}$$

84.What is defect priority?

A defect priority is an urgency of fixing the defect. Normally the defect priority is set on a scale of P0 to P3 with the P0 defect having the most urgency to fix.

85.What is defect severity?

Defect severity is the severity of the defect impacting the functionality. Based on the organization, we can have different levels of defect severity ranging from minor to critical or show stopper.

89.What is a blocker?

A blocker is a bug of high priority and high severity. It prevents or blocks testing of some other major portion of the application as well.

90.What is penetration testing?

Penetration Testing (also known as **Pen Testing**) is a simulated cyberattack on a system, application, or network to identify vulnerabilities that could be exploited by attackers. It helps assess the security of the system by attempting to breach its defences, and then providing insights for improving its security posture.

91.What is localization testing?

Localization Testing is the process of verifying that a software application is adapted correctly for a specific region or language. It ensures that the product is culturally appropriate and meets local standards, such as language translation, date/time formats, currency, and cultural nuances.

92.What is risk analysis?

Risk Analysis is the process of identifying and evaluating potential risks in a project, assessing their impact, and planning strategies to mitigate them.

93.What is the difference between Release and Build?

Build refers to a version of the software that is created by compiling the source code and is ready for testing. It is typically an internal version used by developers and testers.

Release refers to a version of the software that is considered stable, tested, and is made available to users or customers. It is the final version that is distributed for production use.

94.What do you mean by Defect Triage?

Defect Triage is the process of reviewing, prioritizing, and assigning defects (bugs) based on their severity, impact, and urgency.

The goal is to ensure that the most critical issues are addressed first, and the appropriate resources are allocated for fixing them. It typically involves collaboration between developers, testers, and project managers.

95.What is Agile?

Agile is a software development methodology that emphasizes flexibility, collaboration, and iterative progress. It focuses on delivering small, incremental improvements in short cycles (sprints), allowing teams to adapt to changes quickly and deliver working software more frequently.

96.What is Scrum?

Scrum is an Agile framework for managing and completing complex projects. It divides the project into small, manageable units called **sprints**, typically lasting 2-4 weeks.

Scrum focuses on collaboration, flexibility, and continuous improvement, with key roles such as Product Owner, Scrum Master, and Development Team. It includes regular meetings like **Daily Standups**, **Sprint Planning**, and **Sprint Reviews** to ensure progress and address challenges.

97.Define the roles in Scrum?

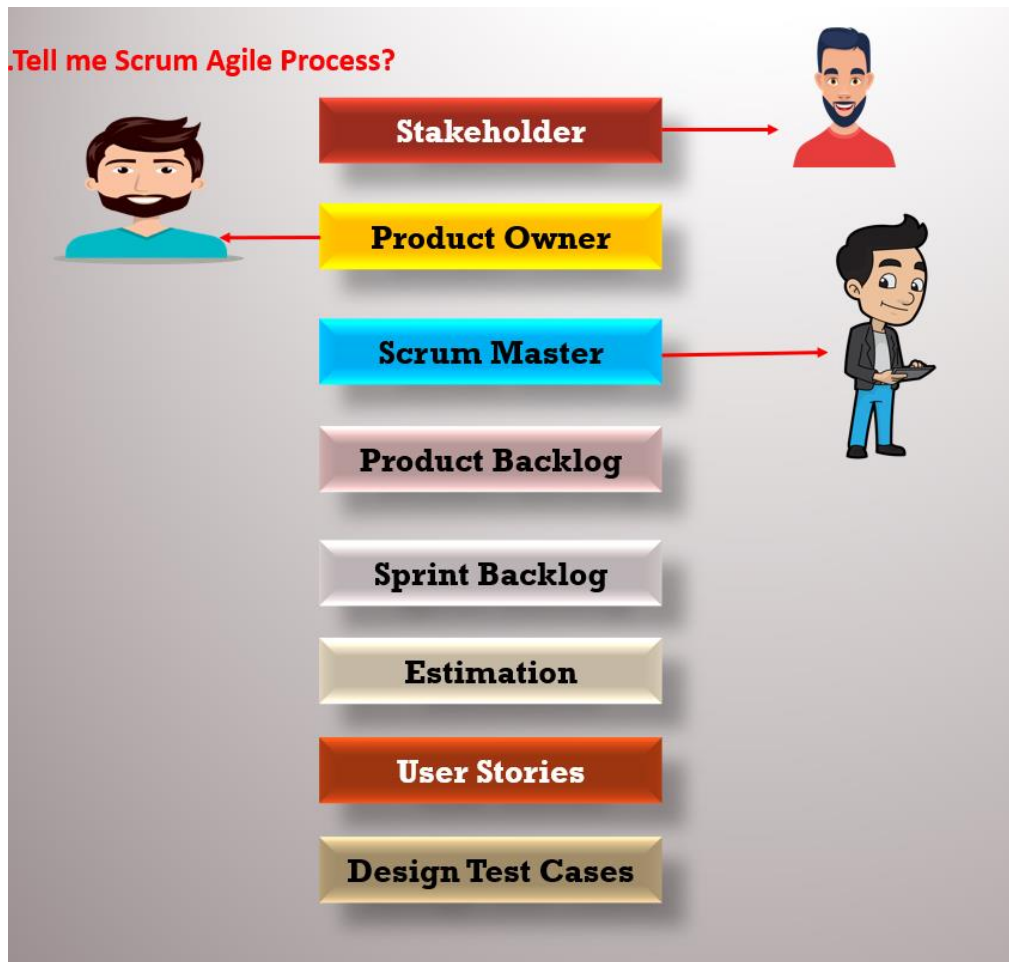
In Scrum:

1. **Product Owner** – Prioritizes and defines the product backlog.
2. **Scrum Master** – Facilitates Scrum processes and removes obstacles.
3. **Development Team** – Delivers the product increment during each sprint.

In Scrum:

1. **Product Owner** – Prioritizes and defines the product backlog.
2. **Scrum Master** – Facilitates Scrum processes and removes obstacles.
3. **Development Team** – Delivers the product increment during each sprint.

98. Scrum Agile Process:



99.What are advantages and disadvantages of Agile Process.

Advantages:

- ✓ Adapt well with changing requirements.
- ✓ Face-to-face conversation with team members and customers.
- ✓ Focuses on technical excellence and good design.
- ✓ Fast and continuous development.
- ✓ Enables collaboration and interaction between client and project team.
- ✓ Ensure and promote customer satisfaction.
- ✓ Faster feedback from customers or end-users.
- ✓ Quick identification and elimination of errors found in the code.

Disadvantages

- ✓ Lack of formal documentation and designing.
- ✓ Difficult to estimate resource requirement and effort.
- ✓ Not good for small development projects.
- ✓ Costly as compared to other development methodologies.
- ✓ Requires more time and energy from everyone.
- ✓ Risk of ever-lasting project.
- ✓ Difficult to scale large projects.

100. What good qualities an Agile Tester should have?

- ✓ Positive attitude and solution-oriented.
- ✓ Focused towards goal.
- ✓ Excellent communication skills.
- ✓ Understand and fulfill customer requirements.
- ✓ Basic knowledge about the Agile process and its principles.
- ✓ Critical and creative thinking.
- ✓ Share ideas effectively.
- ✓ Plan and prioritize work on the basis of requirements.
- ✓ Cope up with change.

101. What are type of agile methodology?

- ✓ Lean Software Development.
- ✓ Kanban.
- ✓ Scrum.
- ✓ Extreme Programming (XP).
- ✓ Dynamic Systems Development Method (DSDM).
- ✓ Feature Driven Development (FDD).

102. What do you mean by refactoring?

In Scrum Agile, refactoring is the process of improving the internal structure of code without changing its external behaviour. It enhances code readability, maintainability, and reduces technical debt, often done as part of sprints to ensure high-quality, clean code.

103.What are the responsibilities of a Product Owner in your organization?

The **Product Owner** defines the product vision, prioritizes the backlog, communicates with stakeholders, clarifies requirements, ensures quality, and maximizes product value. They ensure the product meets customer needs and business goals.

104. Who is a Scrum Master? And what does he/she do?

A Scrum Master is someone who promotes and supports the usage of Scrum within the team.

- He/She understands the theory, practices, rules and, values of Scrum.
- He/She ensures that the team follows the values, principles and, practices of Scrum.
- They remove any distractions and impediments that hamper the progress of the project.
- The Scrum Master ensures that the team delivers value during the sprint.

105. What is 'Scrum of Scrums'?

A Scrum of Scrums (SoS) is a coordination meeting for multiple Scrum teams working on the same project. Representatives from each team discuss progress, plans, and blockers to align efforts and resolve cross-team dependencies. It is like a larger-scale daily Scrum, promoting collaboration and transparency across teams.

106. What is a product backlog?

A product backlog is a prioritized list of work items (features, bug fixes, etc.) needed to develop and improve a product. Managed by the Product Owner, it guides the development team by focusing on the most important tasks first.

107. What is a sprint backlog?

A **Sprint Backlog** is a list of tasks or work items selected from the **Product Backlog** that the Scrum team commits to completing during a single **Sprint**. It includes user stories, features, and technical tasks, and is created during the **Sprint Planning** meeting.

108. What is Spike and Zero Sprint in Agile?

Spike: A short time spent researching or figuring out how to solve a tricky problem.

Zero Sprint: A setup Sprint used to prepare everything before starting real development work.

109. How can storyboard be defined in Agile?

In Agile, a storyboard is a visual tool used to organize and track the progress of tasks or user stories. It displays work items in different stages, such as **"To Do," "In Progress,"** and **"Done."** It helps teams see the flow of work, identify bottlenecks, and manage tasks efficiently.

110. What is difference between Epic, User Stories, and Tasks?

- **Epic:** A large, high-level work item that is too big to complete in a single Sprint. It represents a major feature or objective and is later broken down into smaller user stories.
- **User Story:** A smaller, more detailed piece of work derived from an epic. It describes a specific functionality or feature from the user's perspective, typically following the format: "As a [user], I want to [do something] so that I can [achieve a goal]."
- **Task:** The smallest unit of work. Tasks break down user stories into specific actions needed to complete them, such as coding, testing, or design.

In short, an **Epic** is a big feature, a **User Story** is a smaller piece of that feature, and a **Task** is the individual work required to complete a user story.

111. What is Incremental and Iterative Development?

- **Incremental Development:** Building the product in small, usable pieces, adding new features over time.
- **Iterative Development:** Repeating development cycles to improve and refine the product based on feedback.

112.What do you mean by the term "increment"?

- ✓ The increment is simply the sum or total of all the product backlog items that were completed during a sprint and the value of increments of all previous sprints.
- ✓ It is the total work completed within the current and previous sprints.

117.What are standard or common metrics for Agile?

- ✓ Velocity.
- ✓ Cumulative Flow Diagram.
- ✓ Defect Removal Awareness.
- ✓ Work Category Allocation.
- ✓ Sprint Burndown Metric.
- ✓ Defect Resolution Time.
- ✓ Time Coverage or Code Coverage.
- ✓ Business Value Delivered.

118. What are the Agile Ceremonies?

The Agile Ceremonies are key meetings or events in the Agile process that help teams collaborate and stay on track. They include:

1. **Sprint Planning:** Team plans what to work on in the upcoming Sprint.
2. **Daily Standup (Daily Scrum):** A brief daily meeting to discuss progress, obstacles, and plans.
3. **Sprint Review:** A meeting at the end of the Sprint to demo the completed work.
4. **Sprint Retrospective:** A reflection meeting where the team discusses what went well and what could be improved.

These ceremonies help ensure effective communication and continuous improvement.

119 What is Scrum meeting?

A Scrum Meeting, also called a Daily Standup, is a short, time-boxed meeting (usually 15 minutes) held daily in Scrum to ensure team alignment.

Key Points:

1. **Purpose:** Discuss progress, identify obstacles, and plan the day's work.
2. **Participants:** Scrum team (Developers, Scrum Master, and optionally Product Owner).
3. **Structure:**
 - What did you do yesterday?
 - What will you do today?
 - Are there any blockers?
4. **Benefits:** Improves communication, identifies issues early, and ensures everyone is aligned.

120 What is Sprint meeting?

Sprint Planning happens at the start of a sprint. For a two-week sprint, it usually lasts 1-2 hours but can go up to 8 hours for a one-month sprint, depending on the product backlog's complexity.

During this meeting, the Scrum team selects high-priority tasks from the **Product Backlog** (a list of all possible tasks) and moves them to the **Sprint Backlog** (the to-do list for the sprint). These tasks, often called **User Stories**, represent what the team will focus on during the sprint.

Sprint Planning ensures the team commits to completing the most important work in the upcoming sprint.

121 What is Sprint Retrospective Meeting?

The Sprint Retrospective is a meeting held at the end of a Sprint where the Scrum team reflects on the sprint that just finished.

Key Points:

- Purpose: Discuss what went well, what didn't, and how to improve.
- Focus: Identify strengths, challenges, and process improvements.

- Outcome: Actionable steps for the team to improve in the next Sprint.

The goal is to continuously improve how the team works together and delivers results.

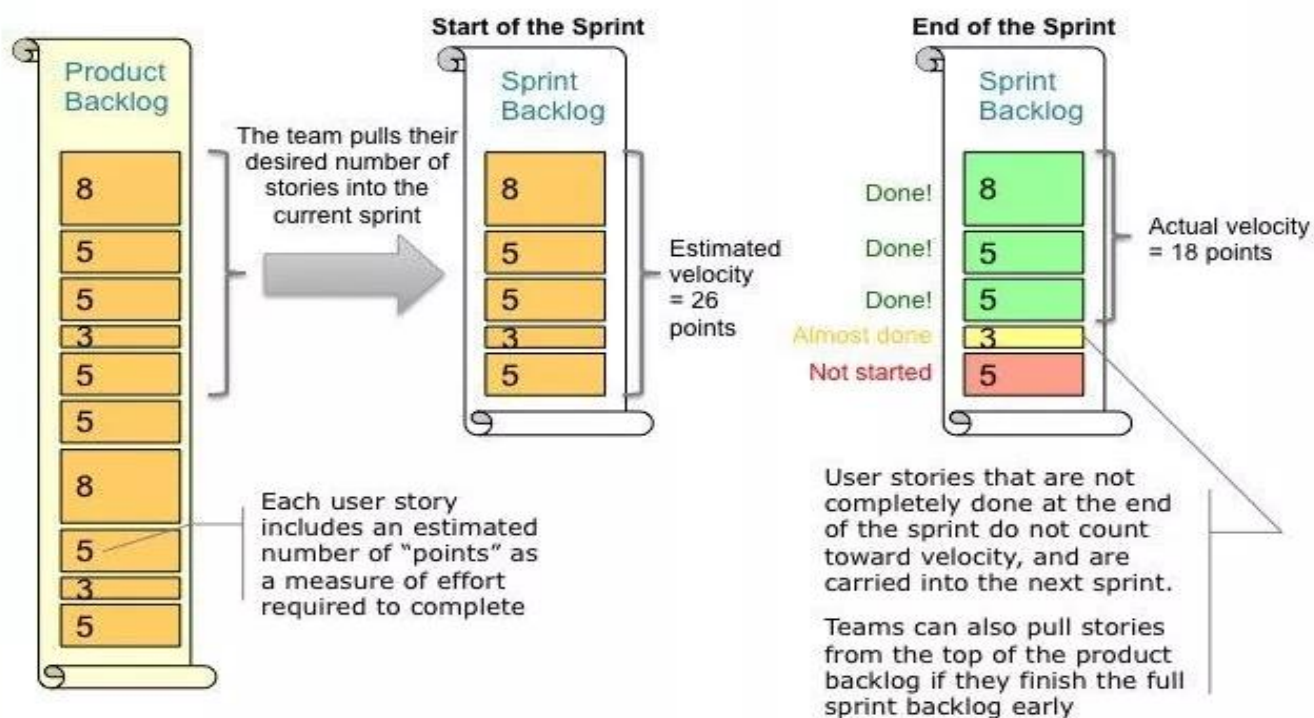
122.What is Velocity of Sprint ?

Velocity in Scrum refers to the amount of work a team can complete during a Sprint, usually measured in story points or work hours.

Key Points:

- **Measurement:** Velocity is typically calculated at the end of a Sprint by adding up the story points for all completed user stories.
- **Purpose:** It helps teams estimate how much work they can handle in future Sprints and track their progress over time.
- **Consistency:** Velocity is a useful metric to determine the team's capacity but can vary based on team experience and complexity of tasks.

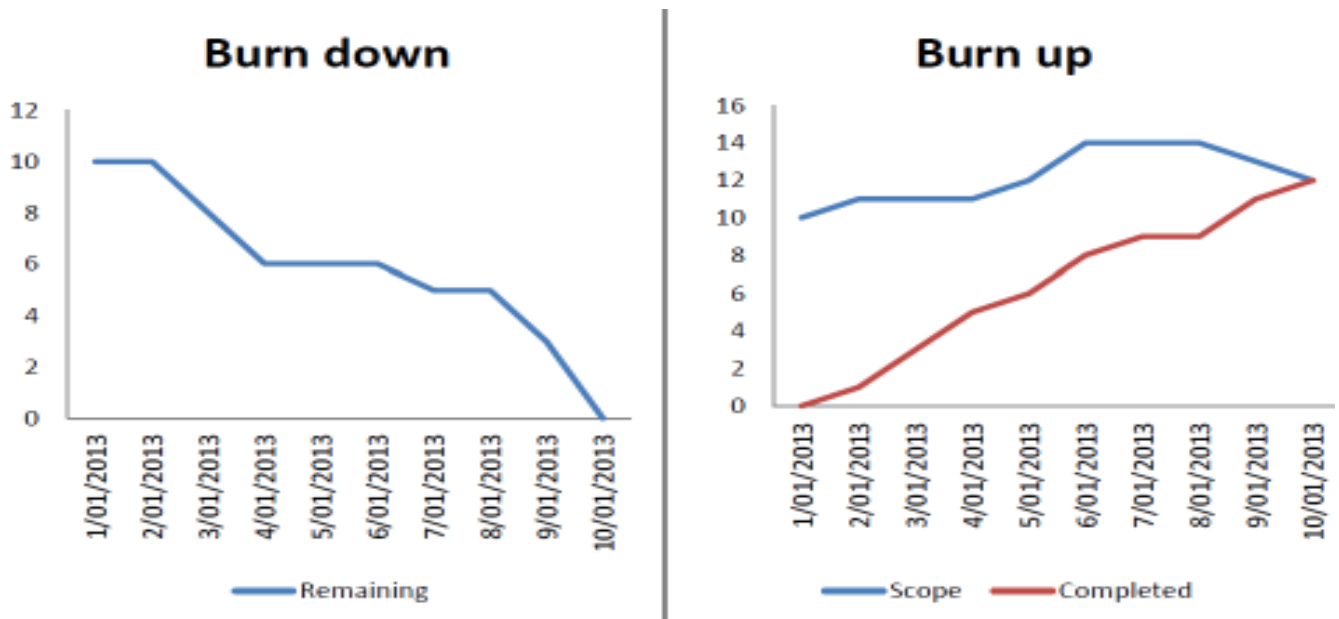
In short, Velocity measures the work done in a Sprint and helps with planning and forecasting future work.



123.What is a Burnup and Burndown Chart?

Burndown Chart: Tracks remaining work over time, showing progress towards completing a Sprint or project.

Burnup Chart: Tracks completed work and total scope, showing both progress and scope changes.



124. How is Estimation Done in a Scrum Project?

Estimation of user stories is based on the difficulty, effort, and complexity of the project. The process involves the Product Owner, Scrum Master, and senior team members (like Team Leads and Project Managers) to ensure accurate estimates.

125. What does DoD mean?

DoD stands for Definition of Done. It is a set of criteria that must be met for a user story or task to be considered complete. It ensures quality and consistency in the team's work.
