**Manual, JIRA and Agile Testing**

**1.What is Different Type of Testing**

* **Unit testing: -**Unit testing validates the function of a unit, ensuring that the inputs (one to a few) result in the lone desired output.
* **Component testing:-**Also called module testing; component testing checks individual parts of an application. Similar to unit testing
* **Integration testing:-** Integration testing is often done in concert with unit testing. Through integration testing, QA professionals verify that individual modules of code work together properly as a group.
* **API testing:-** Application programming interfaces connect different applications or systems, and they are growing in popularity as consumers expect apps to interoperate. With API testing, testers validate that API connections and responses function as intended, including how they handle data and user permissions.
* **UI testing:-** With UI testing, QA professionals interact with the graphical interface of a software program.
* **System testing/END TO END:-** With system testing, QA professionals test the software in its entirety, as a complete product
* **Acceptance testing:-** The purpose of acceptance testing is purely to ensure that the end user can achieve the goals set in the business requirements. testing involves reviewing the feature-complete application flow and end-to-end experience.
* **Production testing:-** Once the product goes public, it is in a live production environment where any user can interact with it in any way

**2.Difference Between Verification and Validation**

**Verification in Software Testing** is a process of checking documents, design, code, and program in order to check if the software has been built according to the requirements or not. The main goal of verification process is to ensure quality of software application, design, architecture etc

**Validation in Software Engineering** is a dynamic mechanism of testing and validating if the software product actually meets the exact needs of the customer or not. The process helps to ensure that the software fulfils the desired use in an appropriate environment. The validation process involves activities like unit testing, integration testing, system testing and user acceptance testing.

**3.Difference Between QA and QC**

**Quality Assurance :-** Quality Assurance is known as QA and focuses on preventing defect. Quality Assurance ensures that the approaches, techniques, methods and processes are designed for the projects are implemented correctly.

Quality assurance activities monitor and verify that the processes used to manage and create the deliverables have been followed and are operative.

Quality Assurance is a proactive process and is Prevention in nature. It recognizes flaws in the process. Quality Assurance has to complete before Quality Control.

1. A QA aim is to prevent the defect.
2. QA is the technique of managing quality.

**Quality Control :-** Quality Control is known as QC and focuses on identifying a defect. QC ensures that the approaches, techniques, methods and processes are designed in the project are following correctly. QC activities monitor and verify that the project deliverables meet the defined quality standards.

1. A QC aim is to identify and improve the defects.
2. QC is a method to verify quality.

**4.Different stages of SDLC**

The Software Development Life Cycle (SDLC) refers to a methodology with clearly defined processes for creating high-quality software. in detail, the SDLC methodology focuses on the following phases of software development:

* Requirement analysis
* Planning
* Software design such as architectural design
* Software development
* Testing
* Deployment

5.Different meetings in AGILE

1.**Sprint planning meeting:-** The sprint planning session is when the Scrum team discusses what work they want to tackle in the next sprint and then prioritizes that work accordingly.

**2.Daily stand-up meeting :-** This Agile [stand-up meeting](https://www.wrike.com/blog/stand-up-meetings-best-practices/) happens every day of the sprint. It’s a quick check-in on what each team member is working on, how the process is going for them, and what stands in their way

3.**Sprint review meeting :-** During the sprint review, the development team presents the work that was done during the sprint (often with a demonstration), with the aim of collecting as much feedback as possible.

4.**Sprint Retrospective:-** the Scrum team focuses specifically on their work together — and not necessarily the product or output. What went well? What didn’t go well? What should they do differently in the next sprint? Each sprint retrospective should conclude with action items the team will implement to improve their collaboration.

**6.What is 7 Software testing Principle**

1. **Testing shows the presence of defects, not their absence** — It’s important to note that *not* finding defects does not make your software bug-free. Because no software is without its defects, testers iterate on their tests to find as many defects as possible.
2. **Exhaustive testing is impossible** — Testing all possible combinations of inputs and preconditions is not typically possible. Rather, teams and managers should prioritize testing based on risk analysis to the product and business.
3. **Early testing saves time and money** — Testers don’t need to wait until the software is deployed to test it. Bugs discovered earlier in a product’s lifecycle are significantly cheaper and easier to address than if it were a customer-discovered bug.
4. **Defects cluster together** — Most software issues follow the Pareto Principle–80% of the issues stem from the same 20% of its modules. While there may be outliers to this, it’s a helpful rule for focusing testing.
5. **Beware the Pesticide Paradox** — This borrows from the idea in agriculture that using the same pesticide over and over again will lead to a decline in its efficacy. In the software world, this means that the usual test cases will eventually stop finding new defects. Review and revise tests regularly.
6. **Testing is context-dependent** — A rinse-and-repeat testing model won’t work for all scenarios. For instance, a high-traffic ecommerce website must have different test cases than an inventory app used by warehouse staff.
7. **Absence of errors is a fallacy** — Software without known issues doesn’t equal error-free software. Finding and fixing *some* defects won’t guarantee the software’s overall success.
8. **Difference Between Smoke and Sanity and Regression**

**Smoke Testing:-** Smoke testing is done to assure that the acute functionalities of program is working fine. Smoke testing may be stable or unstable. Smoke testing is scripted. Smoke testing is done to measures the stability of the system/product by performing testing.

**Sanity Testing:-** Sanity testing is done to check the bugs have been fixed after the build. Sanity testing isn’t documented. Sanity testing is stable. Sanity testing is usually not scripted. Sanity testing is done to measures the rationality of the system/product by performing testing. Sanity testing is used in the case of only modified or defect functions of system/products.

**Regression Testing:-** Regression testing is performed to check the stability of all areas impacted by any functionality change or code change. It is executed based on the project and availability of resources, manpower and time. Regression Testing is often preferred to continue with automation.

**8.Difference Between Priority and Severity**

**Priority:-** Priority is the order in which the developer should resolve a defect whereas Priority is associated with scheduling. Priority indicates how soon the bug should be fixed whereas Priority value is subjective and can change over a period of time depending on the change in the project situation Priority status is based on customer requirements

**Severity:-** Severity is the degree of impact that a defect has on the operation of the product. Severity is associated with functionality or standards. Severity indicates the seriousness of the defect on the product functionality. Severity value is objective and less likely to change. Severity status is based on the technical aspect of the product.

**Example**

**High Priority, Low Severity bug :-** If the company name is misspelled in the home page of the website, then the priority is high and severity is low to fix it.

**High Severity, Low Priority :-** Web page not found when user clicks on a link (user&#39;s does not visit that page generally)

**Low Priority, Low Severity :-** Any cosmetic or spelling issues which is within a paragraph or in the report

**High Priority, High Severity :-** An error which occurs on the basic functionality of the application and will not allow the user to use the system (E.g. user is not able to login to the application)

**9.Difference between Bug leakage and BUG Release**

**Bug release** : when software or an application is handed over to the testing team knowing that the defect is present in a release. During this the priority and severity of bug is low, as bug can be removed before the final handover.  
**Bug leakage:** is something, when the bug is discovered by the end users or customer, and not detected by the testing team while testing the software.

**10.What is Black Box Test Design Technique and Boundary Value analysis and Equal Partitioning**

**Boundary Value Analysis (BVA):**Find the errors at boundaries of input domain (tests the behaviour of a program at the input boundaries) rather than finding those errors in the centre of input. So, the basic idea in boundary value testing is to select input variable values at their: minimum, just above the minimum, just below the minimum, a nominal value, just below the maximum, maximum and just above the maximum.

## **Equivalence Partitioning:** In this method, the input domain data is divided into different equivalence data classes – which are generally termed as ‘Valid’ and ‘Invalid’. The inputs to the software or system are divided into groups that are expected to exhibit similar behaviour.  Thus, it reduces the number of test cases to a finite list of testable test cases covering maximum possibilities.

EX:100-5000 Range Output :- 90 , 2000, 6600

**Decision Table based testing:** Decision table testing is a software testing technique used to test system behaviour for different input combinations. This is a systematic approach where the different input combinations and their corresponding system behaviour (Output) are captured in a tabular form.

**State Transition :** is a black box testing technique in which changes made in input conditions cause state changes or output changes in the Application under Test(AUT). State transition testing helps to analyse behaviour of an application for different input conditions. Testers can provide positive and negative input test values and record the system behaviour.

**Error Guessing :** It is an experience-based testing technique where the Test Analyst uses his/her experience to guess the problematic areas of the application.

**11.What is Alpha Testing and Beta Testing**

**Alpha :** Alpha testing is performed by testers who are usually internal employees of the organization. Alpha testing ensures the quality of the product before forwarding to beta testing. Alpha testing may require long execution cycle. Developers can immediately address the critical issues or fixes in alpha testing.

**Beta :** Beta testing is performed by clients who are not part of the organization. Beta testing is performed at end-user of the product. Reliability, security and robustness are checked during beta testing. Most of the issues or feedback collected from beta testing will be implemented in future versions of the product.

**12.Whit box testing strategy**

Testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing

Open box testing, Transparent box testing, Code-based testing and Glass box testing.

* Internal security holes
* Broken or poorly structured paths in the coding processes
* The flow of specific inputs through the code
* Expected output
* The functionality of conditional loops

**13.** **What is Test Bed**

The test execution environment configured for testing. Test bed consists of specific hardware, software, Operating system, network configuration, the product under test, other system software and application software.

14. **What is Review in ISTQB or Work Product Review Process**

**Planning:** Defining the scope, which includes the purpose of the review, what documents or parts of documents to review, and the quality characteristics to be evaluated. Estimating effort and timeframe.

**Initiate review:** Distributing the work product (physically or by electronic means) and other material, such as issue log forms, checklists, and related work products. Explaining the scope, objectives, process, roles, and work products to the participants. Communicating identified potential defects (e.g., in a review meeting).

**Fixing and reporting:** Creating defect reports for those findings that require changes. Fixing defects found in the work product reviewed. Communicating defects to the appropriate person or team. Recording updated status of defects, potentially including the agreement of the comment originator. Gathering metrics. Checking that exit criteria are met.

**15. What is usability Testing**

Usability testing is a method used to evaluate how easy a website is to use. The tests take place with real users to measure how ‘usable’ or ‘intuitive’ a website is and how easy it is for users to reach their goals.

**16. Explain Test Plan**

Test Plan is A document describing the scope, approach, resources, and schedule of intended test activities.

**1.Analyse the product:**

Who will use the product?

What is the main purpose of this product?

How does the product work?

What are the software and hardware specifications?

**2.Design the Test Strategy**

Project objectives and how to achieve them.

The amount of effort and cost required for testing.

Scope of Testing

Type of Testing

Risks and Issues

Test Logistics

**3.Define the Test Objectives**

This phase defines the goals and expected results of test execution.

A list of all software features – functionality, GUI, performance standards- that must be tested.

The ideal result or benchmark for every aspect of the software that needs testing. This is the benchmark to which all actual results will be compared.

**4. Establish Test Criteria**

Suspension Criteria

Exit Criteria

**5.Planning Resource Allocation**

**6.Planning Setup of Test Environment**

**7.Determining Test Schedule and Estimation**

Employee availability, number of working days, project deadlines, daily resource availability.

Risks associated with the project which has been evaluated in an earlier stage.

**8.Establish Test Deliverables**

Test Deliverables refer to a list of documents, tools, and other equipment that must be created, provided, and maintained to support testing activities in a project.

A different set of deliverables is required before, during, and after testing.

**17.What is requirement matrix**

Is a document that maps and traces user requirement with test cases. It captures all requirements proposed by the client and requirement traceability in a single document, delivered at the conclusion of the Software development life cycle. The main purpose of Requirement Traceability Matrix is to validate that all requirements are checked via test cases such that no functionality is unchecked during Software testing.

**18. Difference between Bug, Defect, Error, Fault & Failure**

**Error**:- The Problem in code leads to errors, which means that a mistake can occur due to the developer's coding error as the developer misunderstood the requirement or the requirement was not defined correctly. The developers use the term error.

**Defect:-** The Defect is the difference between the actual outcomes and expected outputs. The Testers identify the defect. And it was also solved by the developer in the development phase or stage.

**Bug**:-It is an informal name specified to the defect. The Test Engineers submit the Defect and accepted by dev team is called as Bug

**Failure:-** If the software has lots of defects, it leads to failure or causes failure. The failure finds by the Test engineer

**Fault:-** The Fault is a state that causes the software to fail to accomplish its essential function. Due to wrong input or condition

**19.What is Testing Pyramid?**

Unit tests

Integration tests

End-to-End tests

**20.What is Ad hoc Testing**

**Ad hoc Testing:** is an informal or unstructured software testing type that aims to break the testing process in order to find possible defects or errors at an early possible stage.

Ad hoc testing is done randomly and it is usually an unplanned activity which does not follow any documentation and test design techniques to create test cases.

Ad hoc Testing does not follow any structured way of testing and it is randomly done on any part of application. Main aim of this testing is to find defects by random checking.

**21.What is Defect Masking**

Defect that hides other defects in the system. This defect is of the kind which is not detected at a given point of time. Meaning that if there is an existing defect which is not caused (found) to reproduce another defect, the other defect is masked with the previous defect.

**22. Define Latent Defect.**

Latent defect, as the name suggests, is a type of defect or bug which has been in the software system for a long time but is discovered now. A latent defect is an existing defect that can be found effectively with inspections. It usually remains hidden or dormant and is a low-priority defect.

### 23.Explain use-case testing.

### Use case testing is basically defined as a technique that helps developers and testers to identify test cases that exercise the whole system on each transaction basis right from start to finish. It is a part of black-box testing that is used widely in developing tests or systems for acceptable levels.

### 24. What is the importance of Localization Testing?

### Localization testing is a type of testing that is performed to ensure whether the software product offers full functionality and usability in a particular locale or not. It is simply used to verify the accuracy and suitability of the content. It is not only about linguistics, but also about traditions, common herd behaviour, and other similar factors. It generally deals with the functionality and GUI of the application.

### 25.What is Test Harness and Test Closure?

**Test Harness**: Test harness, also known as the automated test framework, is a collection of software and test data required to unit test software modules during development. It is mostly used by the developers and helps in the automation and execution of unit test cases. It generally includes two main parts as given below:

* Test execution engine
* Test script repository

**Test Closure**: Test closure is basically a document that provides the summary of all the tests that are performed during SDLC. It gives full detailed analysis reports of the bugs that are discovered and removed. It is usually performed prior to the end of the testing process.

26.What is Defect Life Cycle

* **New:** When a new defect is logged and posted for the first time. It is assigned a status as NEW.
* **Assigned:** Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to the developer team
* **Open**: The developer starts analysing and works on the defect fix
* **Fixed**: When a developer makes a necessary code change and verifies the change, he or she can make bug status as “Fixed.”
* **Pending retest**: Once the defect is fixed the developer gives a particular code for retesting the code to the tester. Since the software testing remains pending from the testers end, the status assigned is “pending retest.”
* **Retest**: Tester does the retesting of the code at this stage to check whether the defect is fixed by the developer or not and changes the status to “Re-test.”
* **Deferred**: *The bug, changed to deferred state means the bug is expected to be fixed in next releases.*
* **Rejected**: If the developer feels that the bug is not genuine, developer rejects the bug. Then the state of the bug is changed to “**rejected**”.
* **Duplicate** : If the bug is repeated twice or the two bugs mention the same concept of the bug, then the recent/latest bug status is changed to “**duplicate**“.

**27.Data required to raise defect in Jira**

* **Defect\_ID** – Unique identification number for the defect.
* **Defect Description** – Detailed description of the Defect including information about the module in which Defect was found.
* **Version** – Version of the application in which defect was found.
* **Steps** – Detailed steps along with screenshots with which the developer can reproduce the defects.
* **Date Raised** – Date when the defect is raised
* **Reference**– where in you Provide reference to the documents like . requirements, design, architecture or maybe even screenshots of the error to help understand the defect
* **Detected By** – Name/ID of the tester who raised the defect
* **Status** – Status of the defect , more on this later
* **Fixed by** – Name/ID of the developer who fixed it
* **Date Closed** – Date when the defect is closed
* **Severity** which describes the impact of the defect on the application
* **Priority** which is related to defect fixing urgency. Severity Priority could be High/Medium/Low based on the impact urgency at which the defect should be fixed respectively

**27.How to Ensure Test Coverage**

#### 1. Test Coverage vs. Code Coverage

2.Create a checklist for all of the testing activities.

3.Write down the risks inherent to the application.

**28. How Do You Test the Login Feature of a Web Application?**

1. Sign in with valid login, close browser and reopen and see whether you are still logged in or not.
2. Session management is important – how do we keep track of logged in users, is it via cookies or web sessions?
3. Sign in, then logout and then go back to the login page to see if you are truly logged out.
4. Login, then go back to the same page, do you see the login screen again?
5. Sign in from one browser, then open another browser to see if you need to sign in again?
6. Login, change password, and then logout, then see if you can login again with the old password.