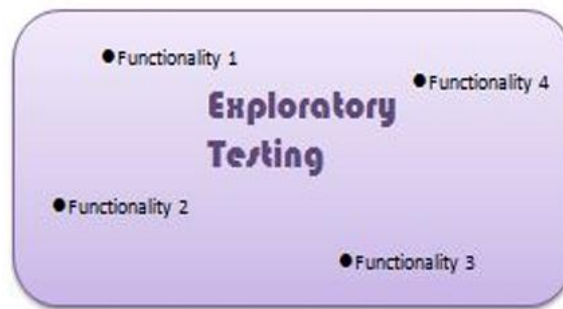


Q1. What is Exploratory Testing?

A1.



**Functionalities are checked
in a ad-hoc manner**

Exploratory testing is a concurrent process where test design, test execution and logging takes place simultaneously. Testing is often not recorded. Makes use of experience, heuristics and test patterns.

Q2. What is traceability matrix?

A2.

Requirement Traceability Matrix												
	Test Case ID	TC_1	TC_2	TC_3	TC_4	TC_5	TC_6	TC_7	TC_8	TC_9	TC_10	# Test Cases for respective Requirement
Req. ID												
Req_1		✖		✖			✖					3
Req_2			✖			✖						2
Req_3				✖								1
Req_4					✖		✖					2
Req_5						✖		✖				2
Req_6							✖					1
Req_7						✖		✖				2
Req_8									✖			1
Req_9										✖		1
Req_10											✖	1

© TestinoCup.blogspot.com

Mapping of the requirements with the test cases is known as traceability matrix. The document in which it is maintained is called RTM (Requirement Traceability Matrix).

Types of traceability Matrix are :

- Forward Traceability – Mapping of Requirements to Test cases
- Backward Traceability – Mapping of Test Cases to Requirements
- Bi-Directional Traceability - A Good Traceability matrix is the References from test cases to basis documentation and vice versa

Q3. What is Boundary value testing?

A3.

Boundary Value Testing

AGE * Accepts Value 21 to 65

Boundary Value Test Case		
Invalid Test Case (Min value - 1)	Valid Test Case (Min + Min, Max, -Max)	Invalid Test Case (Max value + 1)
20	21,22,65,64	66

Boundary value analysis is a methodology to analyse the boundary to get valid ranges. It is a method which refines equivalence partitioning.

Q4. What is Equivalence partitioning testing?

A4.

e. g emp_no : __58__ (1 to 100)

range (E.P)	Representative (input)	Result
1 - 20	15	Pass (in bound- valid)
21 - 40	22	Pass
41 - 60	58	Pass
61 - 80	77	Pass
81 - 100	91	Pass
101 - 120	105	Fail (out bound - invalid)

In this method, divide the range by equivalent partitions then select one representative value from each partition to test the whole partition as pass (in range) or fail (out of range). We are checking a range of numbers by this method.

Q5. What is Integration testing?

A5. Integration Testing is a level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing. Either Developers themselves or independent Testers perform Integration Testing.

There are 2 levels of Integration Testing

- Component Integration Testing (performed at s/f testing level-2)
- System Integration Testing (performed after s/f testing Level-3)

Q6. What is Alpha testing?

A6. Alpha Testing is a form of user acceptance testing, performed using both black box and white box testing techniques. It is performed and carried out at the developing organization's location with the involvement of developers. Alpha Testing is not open to the market and public. It is always performed within the organization. Alpha testing is done on "Project". It is performed in virtual environment.

Q7. What is beta testing?

A7. Beta Testing is a form of user acceptance testing, performed using only black box testing technique. It is always performed by the customers at their own site with the involvement of independent testing team. Beta testing is always open to the market and public. It is done on product. It is performed in real time environment.

Q8. What is component testing?

A8. Component is a level of the software testing process where individual units/components of a software/system are tested. It is also known as Unit Testing/ Module Testing/ Program Testing. It is performed by software developers in the form of sdebugging process.

Q9. What is functional system testing?

A9. Functional testing is a type of software testing in which the system is tested against the functional requirements and specifications. It focusses on testing the attributes of the system that directly affect the functionalities of the system. E. g. To verify checkout functionality on e-commerce site.

Q10. What is Non-Functional Testing?

A10. Non-functional testing is a type of software testing in which the system is tested against the non-functional requirements and specifications. It focusses on testing the attributes of the system that does not directly affect the functionalities of the system. It aims to performance, load, GUI, security, usability of a software/system. E. g. To verify performance of the social media app if no. of users suddenly exceeds the load limit.

Q11. What is GUI Testing?

A11. Graphical User Interface (GUI) is a type of non-functional testing. In this type of testing, user interface of an application is tested based on look and feel. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars – tool bar, menu bar, dialog boxes and windows etc.

e. g

- Check all the GUI elements for size, position, width, length and acceptance of characters or numbers. For instance, you must be able to provide inputs to the input fields.
- Check font used in application is readable
- Check the alignment of the text is proper
- Check the colour of the font and warning messages is aesthetically pleasing
- Check that the images have good clarity
- Check that the images are properly aligned
- Font size, style, and colour for headline, description text, labels, infield data, and grid info should be as per standard as specified in SRS.

Q12. What is Adhoc testing?

A12. Adhoc testing is an informal testing type with an aim to break the system. Testers randomly test the application without any test cases or any business requirement document. Main aim of this testing is to find defects by random checking. Adhoc testing can be achieved with the testing

technique called Error Guessing. The Error guessing is a technique where the experienced and good testers are encouraged to think of situations in which the software may not be able to cope.

Types of Adhoc testing:

- Buddy testing – Two buddies mutually work on one module to identify defects. Mostly one buddy is from development team and other from testing team
- Pair testing - Two testers are assigned modules, share ideas and work on the same machines to find defects.
- Monkey testing - Randomly test the product or application without test cases with a goal to break the system.

Q13. What is load testing?

A13. Load testing is a type of performance testing to evaluate the performance of the system under normal and peak load conditions. This testing helps determine how the application behaves when multiple users access it simultaneously. This test validates the stability and response time of the system with designed no. of users.

e.g app will handle 1000 users at every 5 sec. You have to check 1000 or ≤ 1000 users with your app.

Q14. What is stress Testing?

A14. Stress testing is a type of performance testing to evaluate the performance of the system under extreme load conditions. This testing helps determine how the application behaves when multiple users access it simultaneously. This test validated the stability and response time of the system with more than designed no. of users.

e.g app will handle 1000 users at every 5 sec. You have to check 1000 or ≥ 1000 users with your app.

Q15. What is white box testing and list the types of white box testing?

A15. Testing based on an analysis of the internal structure of the component or system is called white box testing. Also known as "Glass box testing" or "Open box testing". White box testing is the detailed investigation of internal logic and structure of the code. It is done on source code.

Types of White box testing:

- Statement/segment coverage - covers only the true conditions.
- Decision/branch coverage - covers both the true and false conditions.
- Condition coverage - full condition coverage(all true/all false) does not guarantee full decision coverage.

Q16. What is black box testing? What are the different black box testing techniques?

A16. Testing based on an analysis of the system without any knowledge on the internal structure of the system is called black box testing. A tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon. It is done on executable code.

Types of Black box testing:

- Equivalent Partitioning - Divide the range by equivalent partitions, then select one representative value from each partition to test the whole partition

- Boundary Value Analysis - analyse the boundary to get the valid range.
- Decision Table - to test the various combinations of inputs.
- State Transition Technique - All the transactions are stored into finite machine and we are just evaluating each transition state by state.

Q17. Mention what bigbang testing is?

A17. Bigbang testing is a type of component Integration Testing where components or modules is integrated simultaneously, after which everything is tested as a whole.

Pros:

- Easy for smaller system.

Cons:

- Fault identification is difficult.
- there is a chance to miss out to test any module or component for integration.
- testing team will have less time for execution in the testing because of tested as a whole.
- all modules are tested at once, high risk for module isolation

Q18. What is the purpose of exit criteria?

A18. Exit criteria necessary in order to conclude testing activities. All the conditions in exit criteria mentioned in the test plan must be satisfied in order to stop testing activities.

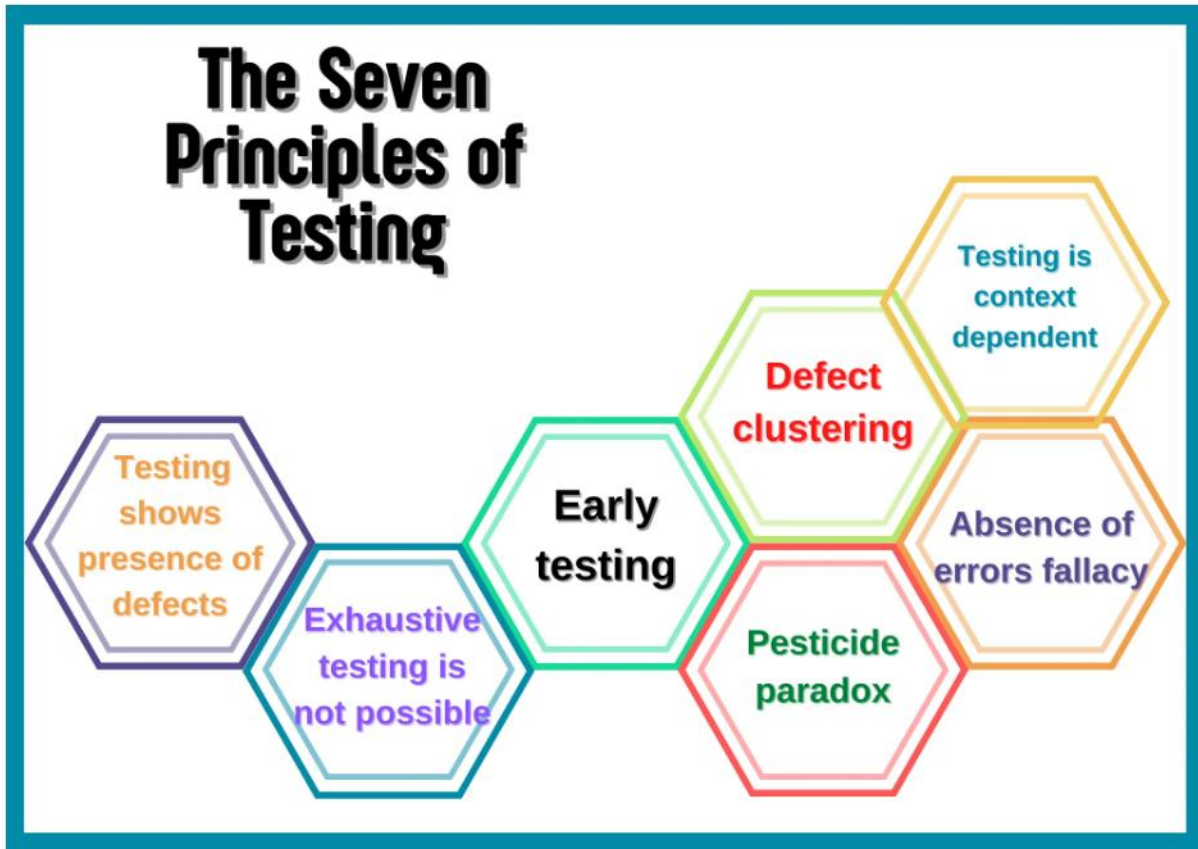
Q19. When should "Regression Testing" be performed?

A19. Regression testing should be carried out:

- when the system is stable and the system or the environment changes
- when testing bug-fix releases as part of the maintenance phase

Q20. What is 7 key principles? Explain in detail?

A20.



1. Testing shows presence of defects:
 - Testing can show that defects are present, but cannot prove that there are no defects.
 - Testing reduces the probability of undiscovered defects.
 - Testing cannot prove that there are no defects present
2. Exhaustive Testing is impossible:
 - Testing everything including all combinations of input and preconditions is not possible.
 - So instead of testing all possible combinations we can test subset of inputs to test an application.
3. Early Testing:
 - Testing activities should start as early as possible in the software development life cycle.
 - Software testing does not start once the code has been written.
 - If the defects are not found in the early stage of SDLC the defects can pass on to the later stages and add up the costs.
4. Defect Clustering:
 - In practice, defects tend to cluster within specific modules.
 - Identifying and focussing on such modules can lead to more efficient defect detection.
5. Pesticide Paradox:
 - Defects raised by testers are fixed by developers.

- Bug fixed software is now immune to old tests.
 - To overcome Pesticide paradox, test cases need to be regularly reviewed and revised. New test cases must be created based on new techniques in order to uncover new bugs.
6. Testing is context dependent:
- Testing is done differently in different contexts.
 - Different kinds of sites are tested differently.
 - E.g Healthcare app is tested differently from gaming app
7. Absence of errors is a fallacy:
- Even if defects are resolved, application can still be unusable if it does not satisfy user's needs and expectations.

Q21. Difference between QA v/s QC v/s Tester

A21.

Sr. No.	Quality Assurance	Quality Control	Tester
1	Activities which ensure the implementation of processes and procedures in context to verification of developed software.	Activities which ensure the verification of developed software with respect to documented requirements.	Activities which ensure the identification of bug/error/defect in the software.
2	Focuses on processes and procedures rather than actual testing.	Focuses on actual testing by executing the developed software	Focuses on actual testing.
3	Process oriented activities	Product oriented activities	Product oriented activities
4	Preventive process	Corrective process	Preventive process
5	Subset of STLC (Software Testing Life Cycle)	Subset of QA.	Subset of QC.

Q22. Difference between Smoke and Sanity

A22.

Sr. No.	Smoke Testing	Sanity Testing
1	After receiving software build, smoke testing will be performed to check the critical(mandatory) functionalities of the app.	After receiving a software build, with minor changes in code, or functionality, Sanity testing is performed to check that the bugs have been fixed and no further issues are introduced due to these changes.
2	to check the stability of the app.	to check the rationality of the app.
3	This testing is performed by the developers or testers.	This testing is usually performed by testers.

4	Smoke testing is usually documented or scripted.	Sanity testing is usually not documented.
5	Smoke testing is a subset of acceptance testing.	Sanity testing is a subset of regression testing.
6	Smoke testing is like General Health Check Up (e. g of smoke - technical)	Sanity Testing is like specialized health check-up (e. g sanity - technical)

Q23. Difference between Verification and Validation

A23.

Sr. No.	Verification	Validation
1	Verification is the process performed at development level.	Validation is the process performed at testing level.
2	Verification phases are: -Business requirement analysis -System Requirement -Technical Specification -Program Specification	Validation phases are: -User Acceptance Test -System Test -Integration Test -Unit Test
3	It is the process of evaluating product of development to check whether product meets the specification or not	It is the process of evaluating product of development to check whether product meets the customer requirements or not
4	Verification can be achieved by asking: Are you building the product right?	Validation can be achieved by asking: Are you building the right product?
5	Evaluation of verification can be achieved by planning, requirement specification, design specification, code specification and test cases	Evaluation of validation can be achieved as an actual product
6	Activities: Reviews, Walk-throughs, Inspections	Activities: Testing the developed software
7	Also called as Static testing	Also called as dynamic testing

Q24. Explain types of Performance testing.

A24. Following are the types of performance testing:

- Load Testing
- Stress Testing
- Scalability Testing
- Volume Testing (Flood Testing)

- Endurance Testing (Soak Testing)
- Spike Testing

Load Testing:

Load testing is a type of performance testing to evaluate the performance of the system under normal and peak load conditions. This testing helps determine how the application behaves when multiple users access it simultaneously. This test validates the stability and response time of the system with designed no. of users.

e.g app will handle 1000 users at every 5 sec. You have to check 1000 or ≤ 1000 users with your app.

Stress Testing:

Stress testing is a type of performance testing to evaluate the performance of the system under extreme load conditions. This testing helps determine how the application behaves when multiple users access it simultaneously. This test validated the stability and response time of the system with more than designed no. of users.

e.g app will handle 1000 users at every 5 sec. You have to check 1000 or ≥ 1000 users with your app.

Scalability Testing:

Stability testing is a type of performance testing to evaluate the performance of the system with increasing amount of load until system crashes. Scalability testing involves testing the system's ability to handle an increasing number of users or data without decrease in performance.

e.g 1500 users : 10 sec

2000 users : 20 sec

.....

1,00,000 users crashed...

Volume Testing (Flood Testing) Testing:

Volume testing is a type of performance testing to evaluate the performance of the system under varying data volumes. This type of testing is used to evaluate how well the application can handle large volumes of data.

e.g to check the capacity or volume of database.

Endurance Testing (Soak Testing):

Soak testing is a type of performance testing to evaluate the performance of the system under normal load for an extended period to determine if there are any issues that arise over time.

e.g to check how the system will run continuously.

Spike Testing:

Spike testing is a type of performance testing to evaluate the performance of the system when there is sudden and significant increase in traffic.

e.g to check extreme increment or decrement of load according to the response time.

Q25. What is Error, Defect, Bug and failure?

A25. Error: mistake in the code is called Error.

Defect: Error raised by tester is called Defect.

Bug: Defects accepted by the developer is called Bug.

Failure: When the software build does not meet customer requirements it is called failure.

Q26. Explain the difference between Functional testing and NonFunctional testing

A26.

Sr. No.	Functional Testing	Non-functional Testing
1	Functional testing is performed using the functional specification provided by the client and verifies the system against the functional requirements.	Non-functional testing checks the Performance, reliability, scalability, security and other non-functional aspects of the software system.
2	Functional testing is executed first	Non-functional testing should be performed after functional testing
3	Manual testing or automation tools can be used for functional testing	Using tools will be effective for this testing
4	Business requirements are the inputs to functional testing	Performance parameters like speed, scalability are inputs to non-functional testing
5	Functional testing describes what the product does	Nonfunctional testing describes how good the product works
6	Easy to do manual testing	Tough to do manual testing
7	Types of Functional testing are · Unit Testing, Smoke Testing, Sanity Testing, Integration Testing, White box testing, Black Box testing, User Acceptance testing, Regression Testing	Types of Nonfunctional testing are · Performance Testing, Load Testing, Volume Testing, Stress Testing, Security Testing, Installation Testing, Penetration Testing, Compatibility Testing, Migration Testing

Q27. What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?

A27.

Sr. No.	SDLC	STLC
1	Focused on software development	Focused on software testing
2	Helps to develop good quality software.	Helps to make software defects free.
3	SDLC phases are completed before the STLC phases.	STLC phases are performed after SDLC phases.
4	Coders create a well-organized development plan.	QA team defines the test plan.
5	Developers create the actual software.	Tester designs test cases, set up the environment & work out the RTM.

Q28. What is the difference between test scenarios, test cases, and test script?

A28. Test Scenario - A Scenario is any functionality that can be tested. Test Scenario is 'What to be tested'. Test Scenario can be functional or non-functional to test. Test scenario is nothing but test procedure. The scenarios are derived from use cases.

Test case - Test cases are the input & output provided to the developed software. Test cases is 'How to be tested'.

TC Document includes:

- Test scenario id
- Test case id
- Test case summary
- Pre-conditions
- Steps to perform
- Test data
- Expected output
- Actual Output
- Status (post condition)

Test Suite - Collection of Test cases.

Q29. Explain what Test Plan is? What is the information that should be covered.

A29. A document describing the scope, approach (model), resources and schedule of intended test activities. Determining the scope and risks, and identifying the objectives of testing.

According to test policy (rules for the organization):

- Functional Test Plan

- System Integration Test Plan
- UAT Test Plan

Following are the information points that should be covered in test plan :

- Scope of testing
- Test Objectives
- Levels and types of testing to be done
- Timeline for testing
- Manual Testers required
- Automation Testers required
- Tools required
- Project risks
- Project constraints
- Testing Environment
- Testing Budget

Q30. Develop test scenarios for the following components

A30.

Sr No	Component	Link
1	Whatsapp Chat messages	https://github.com/deepsha094/MT_Assignment_2/blob/main/Test_Scenario_Template.xlsx
2	Pen	
3	Pen Stand	
4	Door	
5	ATM	
6	Microwave oven	
7	Coffee vending machine	
8	Chair	
9	Wrist Watch	
10	Lift (Elevator)	
11	Whatsapp Payment	

Q29. Develop HLR and Test Cases for following Topics

A30.

Sr No	Topic	Link
1	Instagram First Page	https://github.com/deepsha094/MT_Assignment_2/blob/main/Instagram_HLR.xlsx
2	Facebook First Page	https://github.com/deepsha094/MT_Assignment_2/blob/main/Facebook_HLR.xlsx
3	WhatsApp Web	https://github.com/deepsha094/MT_Assignment_2/blob/main/WhatsappWeb_HLR.xlsx

Q31. Develop Test Cases for following Topics

A31.

Sr No.	Topic	Link
1	WhatsApp Group Chat	https://github.com/deepsha094/MT_Assignment_2/blob/main/Whatsapp%20Group%20chat_TestCases.xlsx
2	Compose Mail Functionality	https://github.com/deepsha094/MT_Assignment_2/blob/main/Compose%20Mail_TestCases.xlsx
3	Online shopping to buy product (Flipkart)	https://github.com/deepsha094/MT_Assignment_2/blob/main/Flipkart_TestCases.xlsx

Q32. Mention what are the categories of defects?

A32. Defects can be categorized into different types basing on the core issues they address. Some defects address security or database issues while others may refer to functionality or UI issues.

- Functionality Defects:
Defects directly related to functionalities. Not working features properly. e.g Calculator has no '=' button for the calculation.
- Performance Defects:
Software doesn't meet the expected performance requirements. e.g Website's loading time to open.
- User Interface Defects:
Difficult to operate for the users. Not user friendly. e.g Login page has no cancel button, Alignment problem.
- Comfortable Defects:
Software does not work correctly on different hardware and software configuration.

e.g Application not running on Android or Windows platform.

Application interface shows differently in different browsers.

- Security Defects:

Software doesn't protect the user's data from malicious attack. e.g Password entered in visible form. Authentication: Accepting an invalid username/password. Authorization: Accessibility to pages though permission not given.

- Documentation Defects:

Document is incorrect or inaccurate to use the features of the app. e.g TC had a wrong entry.

- Database Defects:

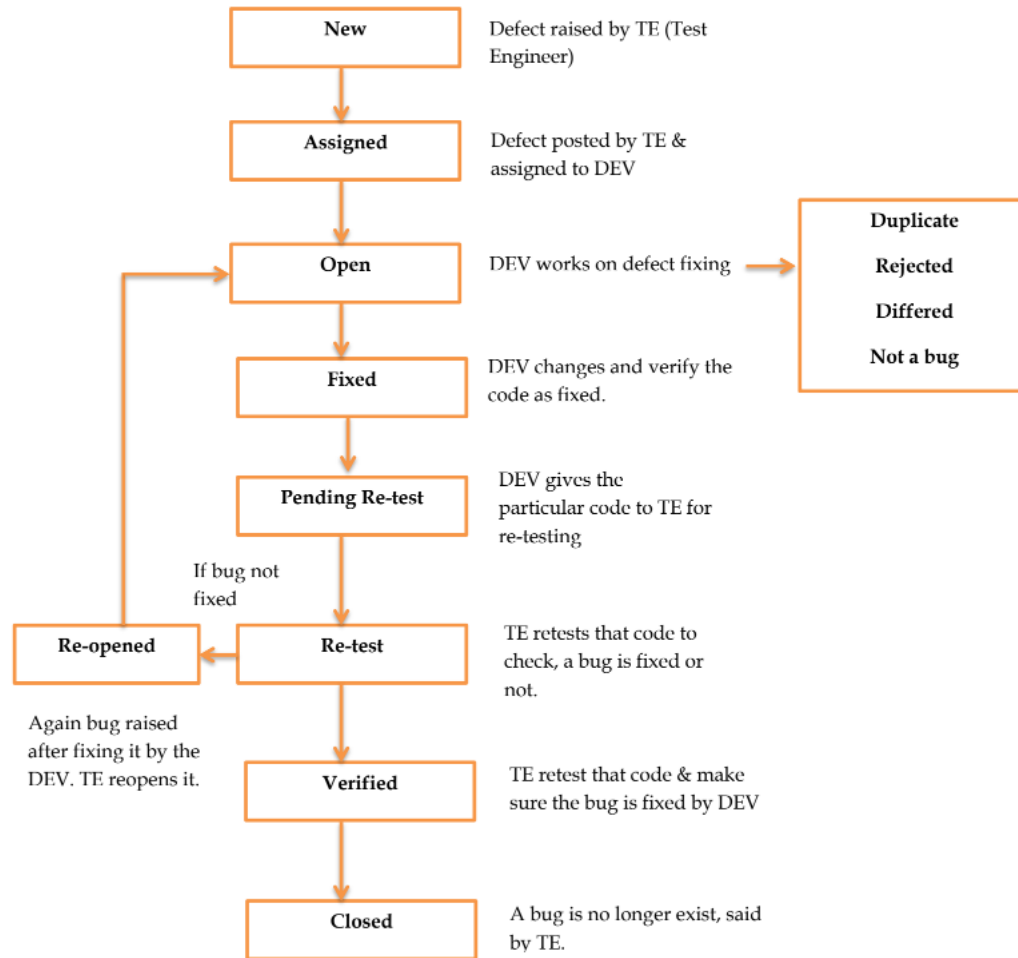
Q33. Difference between Priority and Severity

A33.

Sr No	Priority	Severity
1	Priority is a parameter to decide the order in which defects should be fixed.	Severity is a parameter to denote the impact of a particular defect on the software.
2	Priority means how fast the defect has to be fixed.	Severity means how severe the defect is affecting the functionality.
3	Priority is related to scheduling to resolve the problem.	Severity is related to the quality standard.
	Priority is divided into 3 categories: <ul style="list-style-type: none"> • Low • Medium • High 	Severity is divided into 4 categories: <ul style="list-style-type: none"> • Critical • Major • Medium • Low
4	The product manager decides the priorities of defects.	The testing engineer decides the severity level of the defect.
5	Its value is subjective.	Its value is objective.
6	Its value changes from time to time.	Its value doesn't change from time to time.

Q34. What is Bug Life Cycle?

A34. The duration or timespan between the first time defect found and the time when defect closed, rejected, differed or postponed is called defect life cycle.



Duplicate: Defect repeating twice.

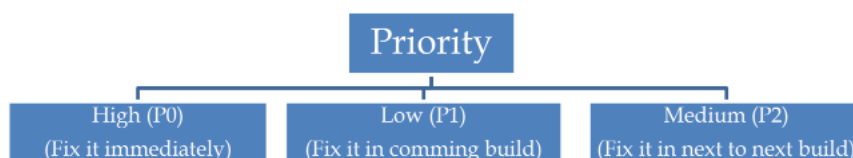
Rejected: DEV feels defect is not genuine.

Differed: Present bug not having prime priority. Defect can be fixed in next release.

Not a bug: Defect does not affect the functionality of an app.

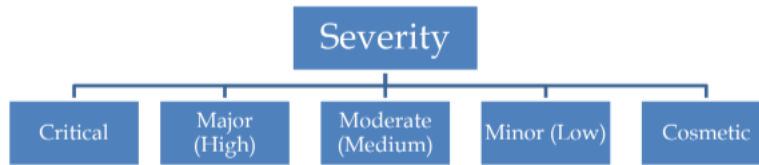
Q35. What is priority?

A35. If you are raising any bug for any application, how soon you want the developer to fix that bug is called priority. Priority is considered as customer's point of view. But priority can be set by the QA tester. Later on it can be changed by project manager. The importance given to the bug to fix it that is priority. The high priority indicates that the bug to fix it first.



Q36. What is severity?

A36. The impact of Defect /bug on the customer business workflow is known as Severity. If that impact is more then, there is high severity. If that impact is less then, there is low severity.



Critical:

- The main functionality is not working.
- For E.g Login by username & password, it will show a blank page then the next step is critical to deal with.
- For E.g If you have sent Rs. 10000 to account "X". Then you login to check balance and it shows only Rs. 5000 as balance.

Major (High):

- One of the functionality expected from the software which is not happening. For example, If you booked Ola cab, and you got the message for cab booking confirmation. Your cab came but you still not receiving any OTP. Then how the next step of your ride will be possible?
- For example, You are sending any mail but you didn't get any pop up message for mail sent. You need to go to "Sent" to check that mail.

Moderate (Medium):

- Result is not coming consistency. For example, Run the application 5 times, then 3 times its working fine but 2 times it shows some errors. So result is inconsistent and error is also inconsistent.

Minor (Low):

- Defect is minor here. Usability of functionality not affected much but must be fixed. For example, Run the same application with both the browsers Firefox & Chrome. Look & feel can be different but not much affected to its usability.

Cosmetic:

- Related to GUI issue like spell mistake, alignment problem.
- For e.g Username & password textfield is not formatted with proper alignment.

Q37. Advantage of Bugzilla.

A37: Following are the advantages of Bugzilla:

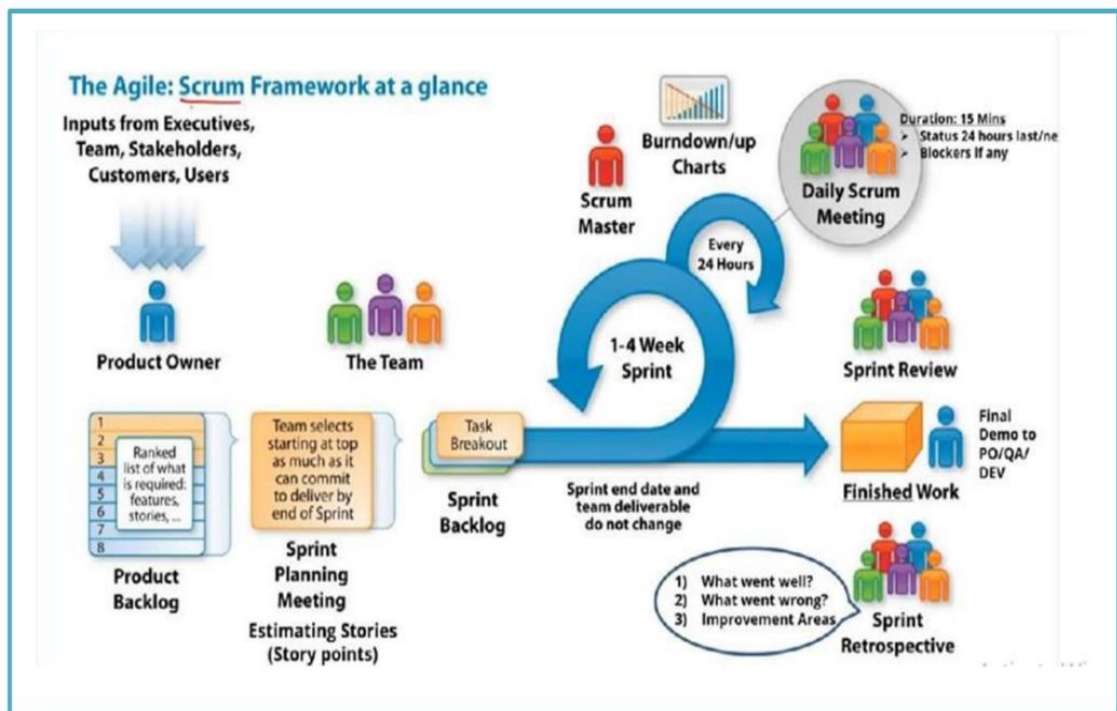
- You don't have to shell out a dime, as it's free and open-source, making it super accessible for all project sizes.
- This tool is seriously adaptable, letting you tweak things to fit specific project needs.
- A robust community keeps the improvements and support rolling in.

- With its hefty feature set, productivity and teamwork levels up significantly.
- Bugzilla's analytics tools are spot-on for providing essential project insights.
- Its smooth integration capabilities streamline overarching project processes.
- Deployed across numerous industries, its reliability and utility are assured.

Q38. What are the different Methodologies in Agile Development Model?

A38. Different methodologies in Agile Development are : Scrum, Kanban, XP

1. Scrum:
 - Scrum is a framework through which we build software product by following Agile principles.
 - SCRUM is an agile development method which concentrates particularly on how to manage tasks within a team based development environment.
 - Scrum includes group of people called a scrum team. Normally contains 5 to 9 members.
 - Scrum team can involve the people like product owner, scrum master, DEV team, QA team etc.



Scrum Team

- **Product Owner:** Responsible for defining the product backlog and ensuring it meets business and customer needs.
- **Scrum Master:** Facilitates the Scrum process, removes obstacles, and ensures the team follows Scrum principles.
- **Development Team:** A cross-functional group responsible for delivering the product increment

Artifacts

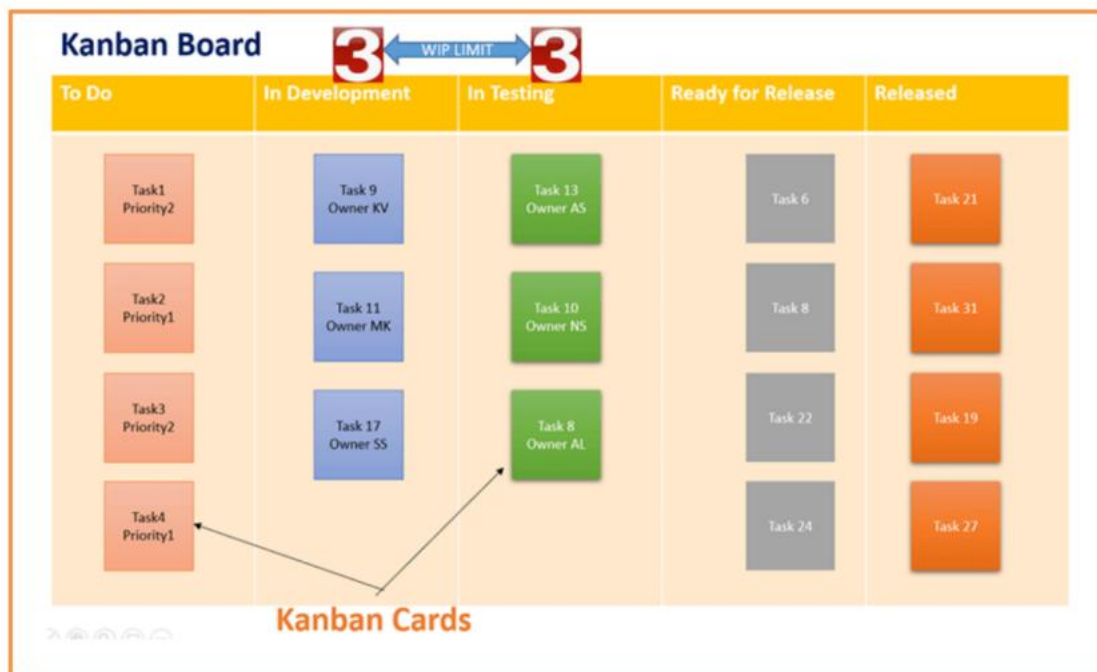
- **Product Backlog:** A prioritized list of features, fixes, and enhancements for the product.
- **Sprint Backlog:** A list of tasks the team commits to completing during the current sprint.
- **Increment:** The finished work at the end of a sprint that could be released to users

Events

- **Sprint:** A time-boxed period (usually 1-4 weeks) where the team works on specific tasks from the sprint backlog.
- **Sprint Planning:** A meeting to decide what to work on during the sprint.
- **Daily Scrum:** A short daily meeting to check progress and plan the day's work.
- **Sprint Review:** A meeting at the end of the sprint to showcase the completed work and get feedback.
- **Sprint Retrospective:** A meeting to discuss what went well and what could be improved for the next sprint

Q39. Explain the difference between Authorization and Authentication in Web testing. What are the common problems faced in Web testing?

A39.



- Kanban is a very popular framework for development in the agile software development methodology.
- It provides a transparent way of visualizing the tasks and work capacity of a team.
- It mainly uses physical and digital boards to allow the team members to visualize the current state of the project they are working on.

- Kanban originated in Toyota in the 1940s.
- Kanban's meaning in Japanese is "billboards."
- The Kanban board has columns and story cards.
- The columns are nothing, but workflow states and cards are nothing but a demonstration of the actual task a team member is performing.
- Kanban should be used when you want to visualize your work, and you want to see the progress of your tasks visually.

