Module -2

1. What is Error, Defect ,bug and failure?

_'A mistake in coding is called error, error found by tester is called defect, defect accepted by development team then called bug, bug build does not meet the requirement then it is failure.

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

White box testing can uncover structural problems, hidden errors and problems with specific components.

Types: Unit testing
Mutation testing
Integration testing
White box penetration testing
Static code analysis

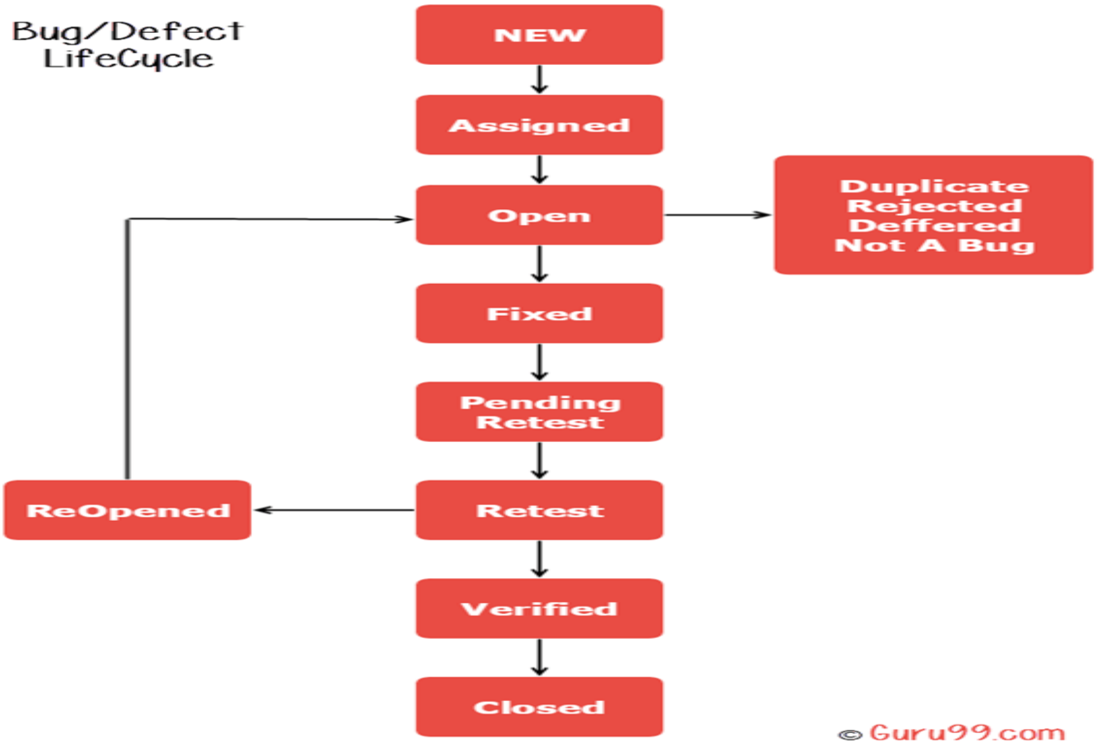
3. What is black box testing ? what are the different types of black box testing?

Black Box Testing is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths.

Technique: Equivalence Class Testing
Boundary Value Testing
Decision Table Testing

4. What is bug life cycle?

The duration or time span between the first time defects is found and the time that it is closed successfully, rejected, postponed or deferred is called as 'Defect Life Cycle'.



5. what is component testing?

Component testing is defined as a software testing type, in which the testing is performed on each individual component separately without integrating with other components.

6. What is functional system testing?

Functional Testing is a type of software testing that validates the software system against the functional requirements/specifications. Functional testing mainly involves black box testing and it is not concerned about the source code of the application.

1) Unit Testing. 2) Integration Testing. 3) Interface Testing
4) System Testing. 5) Regression Testing.6) Smoke Testing.
7) Sanity Testing.8) Acceptance Testing.

7. What is non-functional system testing?

Non-Functional Testing is defined as a type of Software testing to check non-functional aspects (performance, usability, reliability, etc) of a software application.

Types:

1.usibility testing 2.compatibility testing 3.performance testing 4.security testing.

8. What is GUI testing?

graphical user interface testing is the process of testing a product's graphical user interface (GUI) to ensure it meets its specifications. This is normally done through the use of a variety of test cases.

9. What is adhoc testing?

Adhoc 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.

10. What is load testing?

Load testing is a type of **performance testing** that simulates a real-world load on any software, application, or website. Without it, your application could fail miserably in real-world conditions.

11. What is stress testing?

Stress Testing is a software testing technique that determines the robustness of software by testing beyond the limits of normal operation. Stress testing is particularly important for critical software but is used for all types of software.

12. What is alpha testing?

- It id always performed by the developers at the software development site.
- It is not open in market and public.
- It is always performed virtual environment.

13. What is beta testing?

- It is always performed by customers.
- It is not performed independent testing team.
- It is always open to the market and public.

14. What is Exploratory Testing?

Exploratory testing is type of software testing in which the tester is free to select any possible methodology to test the software. It is an unscripted approach for software testing. In exploratory testing, software developers use their personal learning, knowledge, skills, and abilities to test the software developed by themselves.

15. What is Traceability Matrix (TM)?

A Traceability Matrix is a document that co-relates any two-baseline documents that require a many-to-many relationship to check the completeness of the relationship. It is used to track the requirements and to check the current project requirements are met.

16. What is Boundary value testing?

The boundary values of such a testing mechanism are identified by the values present at the extreme boundaries, i.e. minimum and maximum value. This is used mainly to analyze the testing at the partition boundaries and also to detect anomalies that may occur during testing cases.

17. What is Equivalence partitioning testing?

Equivalence partitioning is also known as equivalence class partitioning (ECP). It is a software testing technique or black box testing that divides input domain into classes of data, and with the help of these classes of data, test cases can be derived. An ideal test case identifies class of error that might require many arbitrary test cases to be executed before general error is observed.

18. What is Integration testing?

Integration testing is the second level of the software testing process comes after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.

Types:

Gray box testing, component testing, system testing

19. What determines the level of risk?

Risk can be defined as the probability of an event, hazard, accident, threat or situation occurring and its undesirable consequences. It is a factor that could result in negative consequences and usually expressed as the product of impact and likelihood.

In software terminology, the risk is broadly divided into two main categories:

Project Risks: • Supplier issues • Organizational factors • Technical issues

20. Mention what are the categories of defects?

Data Quality/Database Defects: Deals with improper handling of data in the database.

Examples:

• Values not deleted/inserted into the database properly

• Improper/wrong/null values inserted in place of the actual values Critical

Functionality Defects: The occurrence of these bugs hampers the crucial functionality of the application.

Examples: -

•Exceptions Functionality Defects: These defects affect the functionality of the application.

• All JavaScript errors

• Buttons like Save, Delete, Cancel not performing their intended functions

• A missing functionality (or) a feature not functioning the way it is intended to Continuous execution of loops

Security Defects: Application security defects generally involve improper handling of data sent from the user to the application. These defects are the most severe and given highest priority for a fix.

Examples:

- Authentication: Accepting an invalid username/password

- Authorization: Accessibility to pages though permission not given

21. Mention what bigbang testing is?

Big Bang Integration Testing is an integration testing strategy wherein all units are linked at once, resulting in a complete system. When this type of testing strategy is adopted, it is difficult to isolate any errors found, because attention is not paid to verifying the interfaces across individual units.

22. What is the purpose of exit criteria?

Exit criterion is used to determine whether a given test activity has been completed or NOT. Exit criteria can be defined for all of the test activities right from planning, specification and execution. Exit criterion should be part of test plan and decided in the planning stage.

Examples of Exit Criteria:

- Verify if all tests planned have been run.

- Verify if the level of requirement coverage has been met.

- Verify if there are NO Critical or high severity defects that are left outstanding.

- Verify if all high risk areas are completely tested.
- Verify if software development activities are completed within the projected cost.
- Verify if software development activities are completed within the projected timeline.

23. When should "Regression Testing" be performed?

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.
- It should be considered complete when agreed completion criteria for regression testing have been met.
- Regression test suites evolve over time and given that they are run frequently are ideal candidates for automation.

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

Software testing is the process of executing a program with the aim of finding the error. To make our software perform well it should be error-free. If testing is done successfully it will remove all the errors from the software.

There are seven principles in software testing:

1. Testing shows the presence of defects
2. Exhaustive testing is not possible
3. Early testing
4. Defects clustering
5. Pesticide paradox
6. Testing is context-dependent
7. Absence of errors fallacy

1. Testing shows the presence of defects: The goal of software testing is to make the software fail. Software testing reduces the presence of defects. Software testing talks about the presence of defects and doesn't talk about the absence of defects. Software testing can ensure that defects are present but it cannot prove that software is defect-free. Even multiple testing can never ensure that software is 100% bug-free. Testing can reduce the number of defects but not remove all defects**.**

2. Exhaustive testing is not possible: It is process of testing the functionality of the software in all possible inputs (valid or invalid) and pre-condition is known as exhaustive testing. Exhaustive testing is impossible means the software can never test at every test case. It can test only some test cases and assume that the software is correct and it will produce the correct output in every test case. If the software will test every test case then it will take more cost, effort, etc. which is impractical.

3. Early testing: To find the defect in the software, early test activity shall be started. The defect detected in the early phase of SDLC will be very less expensive. For better performance of software, software testing will start at the initial phase i.e. testing will perform at the requirement analysis phase.

4. Defects clustering: In a project, a small number of modules can contain most of the defects. Pareto principle to software testing state that 80% of software defect comes from 20% of modules**.**

5. Pesticide paradox: Repeating the same test cases, again and again, will not find new bugs. So it is necessary to review the test cases and add or update test cases to find new bugs. 6. Testing is context-dependent: The testing approach depends on the context of the software developed. Different types of software need to perform different types of testing. For example, The testing of the ecommerce site is different from the testing of the android application.

7. Absence of errors fallacy: If a built software is 99% bug-free but it does not follow the user requirement then it is unusable. It is not only necessary that software is 99% bug-free but it is also mandatory to fulfil all the customer requirements.

25. What is difference smoke and sanity?

Sr no	Smoke testing	Sanity testing
1	Smoke Testing is performed to ascertain that the critical functionalities of the program is working fine.	Sanity Testing is done to check the new functionality / bugs have been fixed.

2	The objective of this testing is to verify "stability" of the system in order with more rigorous testing.	The objective of the testing is to verify the the "rationality" of the system in order proceed the "rationality" of the system in order proceed.
3	This testing is performed by the developers.	Sanity testing is usually performed by testers.
4	Smoke testing is usually documented or scripted .	Sanity testing is usually not documented and unscripted.
5	Smoke testing is a subset of Regression testing	Sanity testing is a subset of Acceptance testing
6	Smoke testing exercises the entire system from end to end.	Sanity testing exercises only the particular component of the entire system.
7	Smoke testing is like General Health Check up.	Sanity Testing is like specialized health Check up.

26. Difference between QA v/s QC v/s Tester?

sr.no	QA	QC	Testing
1	It focuses on providing assurance that the quality requested will be achieved.	It focuses on fulfilling the quality requested.	It is responsible for evaluating individual software.
2	It is the technique of managing quality.	It is the technique to verify quality.	It simply evaluates functionality of software application.

3	It is involved during the development phase.	It is not included during the development phase	Software tester generally test whether or not code runs as we expected it to run
4	It is a managerial tool.	It is a corrective tool.	Its aim is to find bugs and errors in software application if present.
5	The aim of quality assurance is to prevent defects.	The aim of quality control is to identify and improve the defects.	They not only find bugs, but also find its root cause so that it can be resolved permanently.
6	The aim of quality assurance is to prevent defects.	The aim of quality control is to identify and improve the defects.	They not only find bugs, but also find its root cause so that it can be resolved permanently.
7	It is responsible for the entire software development life cycle	It is responsible for the software testing life cycle.	They mainly focus on behaviour of end user while testing software application

27. Difference between verification and validation?

sr. no	validation	Verification
1	Validation is the process of checking whether the specification captures the customer's requirements	while verification is the process of checking that the software meets specifications
2	It includes testing and validating the actual product	It includes checking documents, design, codes and programs.

3	Validation is the dynamic testing	Verification is the static testing
4	It includes the execution of the code.	It does *not* include the execution of the code
5	Methods used in validation are Black Box Testing, White Box Testing and non-functional testing	Methods used in verification are reviews, walkthroughs, inspections and desk-checking
6	It checks whether the software meets the requirements and expectations of a customer or not	It checks whether the software conforms to specifications or not.
7	It can find the bugs in the early stage of the development.	It can only find the bugs that could not be found by the verification process.
8	The goal of verification is application and software architecture and specification.	The goal of validation is an actual product.
9	Quality assurance team does verification.	Validation is executed on software code with the help of testing team.
10	It comes before validation.	It comes after verification.
11	It consists of checking of documents/files and is performed by human.	It consists of execution of program and is performed by computer.
12	Verification refers to the set of activities that ensure software correctly implements the specific function.	Validation refers to the set of activities that ensure that the software that has been built is traceable to customer requirements.
13	After a valid and complete specification the verification starts.	Validation begins as soon as project starts.
14	Verification is for prevention of errors.	Validation is for detection of errors.
15	Verification is also termed as white box testing or static testing as work product goes through reviews.	Validation can be termed as black box testing or dynamic testing as work product is executed.
16	Verification finds about 50 to 60% of the defects.	Validation finds about 20 to 30% of the defects.

28.Difference between priority and severity?

sr.no	Priority	Severity

1	Priority is a term that defines how fast we need to fix a defect.	Severity is basically a parameter that denotes the total impact of a given defect on any software .
2	Defined by the impact on business.	Defined by the impact of a specific problem on any application’s functionality
3	Category decided by developers or product owners	Category decided by testers.
4	Deals with the timeframe or order to fix the defects.	Deals with the technical aspects of the application.
5	The priority value is subjective and may change after comparing with other defects	The value does not change with time, it’s fixed

29 . Explain types of Performance testing?

Types:

1. Capacity Testing
2. load testing
3. volume
4. stress testing
5. soak testing

1 .capacity testing: Tests how many users the system can handle before performance dips below acceptable levels. By testing a software’s capacity it helps developers anticipate issues in terms of scalability and future user-base growth.

2. **load testing**: Confirms that the system can handle the required number of users and still operate at a high level of performance. This ensures that there is no day to day issues in performance.

3. volume testing: Checks that the software can handle and process a large amount of data at once without breaking, slowing down, or losing any information.

4. stress testing: Intentionally tries to break the software by simulating a number of users that greatly exceeds expectations. The launch day of a new iPhone and the sudden spike in user traffic on the Apple website is a good example of a stress test in the real world.

5. soak testing: Simulates high traffic for an extended period of time. Checks the software's ability to tolerate extended periods of high traffic.

30. Diffrence between functional and non-functional testing?

Parameters	Functional	Non-functional testing
Execution	It is performed before non-functional testing.	It is performed after the functional testing.
Focus area	It is based on customer's requirements.	It focusses on customer's expectation.
Requirement	It is easy to define functional requirements.	It is difficult to define the requirements for non-functional testing.
Usage	Helps to validate the behavior of the application.	Helps to validate the performance of the application.
Objective	Carried out to validate software actions.	It is done to validate the performance of the software.
Requirements	Functional testing is carried out using the functional specification.	This kind of testing is carried out by performance specifications
Manual testing	Functional testing is easy to execute by manual testing.	It's very hard to perform non-functional testing manually.
Functionality	It describes what the product does.	It describes how the product works.

Example Test Case	Check login functionality.	The dashboard should load in 2 seconds.
Testing Types	Examples of Functional Testing Types	Examples of Non-functional Testing Types
	Unit testing	Performance Testing
	Smoke testing	Volume Testing
	User Acceptance	Scalability
	Integration Testing	Usability Testing
	Regression testing	Load Testing
	Localization	Stress Testing
	Globalization	Compliance Testing
	Interoperability	Portability Testing
		Disaster Recover Testing

31. Difference Between SDLC and STLC?

Parameter	**SDLC**	**STLC**
Origin	Development Life Cycle	Testing Life Cycle
Objective	The main object of SDLC life cycle is to complete successful development of the software including testing and other phases.	The only objective of the STLC phase is testing.
Requirement Gathering	In SDLC the business analyst	In STLC, the QA team analyze requirement documents like functional and non-functional documents and create System Test Plan
	gathers the requirements and	
	create Development Plan	

High & Low-Level Design	In SDLC, the development team creates the high and low-level design plans	In STLC, the test analyst creates the Integration Test Plan
Coding	The real code is developed, and actual work takes place as per the design documents.	The testing team prepares the test environment and executes them
Maintenance	SDLC phase also includes post-deployment supports and updates.	Testers, execute regression suits, usually automation scripts to check maintenance code deployed.

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

sr.no	Test scenario	Test cases	Test script
1	The test scenario is just a document that is detailed and provides details about the assessment method, testing process, precondition, and anticipated output.	Test cases is a step by step procedure to test any functionality of the software application/product.	Test script is set of instruction or a short program to test any functionality of software application/product.
2	• The test scenarios are the ones based on the use situation and give one-line information one what to check.	Test cases is a manual approach of software testing.	Test script is an automatic approach of software testing.
3	Test scenarios are one-liner statement, however, it is linked to a few test instances	It is a set up that is used by the tester to test any specific function of the software product	It is a program developed by the tester, intended to test any specific function of the software product

4	These are highlevel actions.	Point by point test case configuration encourages tester to test viably.	Automatic testing approach is beneficial for constant execution
5	Writing the test scenario’s primary objective is an address end to get rid of functionality of a software program.	Test cases are written by manually	Test scripting is done by scripting format
6	It will take less time as compared to test cases.	Test case is developed in form of templates.	Test script is developed in form of scripting.
7	Test scenario are really easy to maintain due to their highlevel design	Test case is used in manual testing environment	Test script is used in automatic testing environment.
8	The test scenarios tend to be work on the essential to “things to be tested”.	Test cases are classified as delegated, positive, reusable, negative and UI test cases.	• Test script are characterized as manual test script and automatic test scripts
9	Requires fewer resources and less time.	Requires more resources and time.	Requires less time for testing scripts.

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

- A test plan is a detailed document that describes the test strategy, objectives, schedule, estimation, deliverable, and resources required to perform testing for a software product.

- As per ISTQB definition: “test plan is a document describing the scope, approach, resources, and schedule of intended test activities.”

- The test plan serves as a blueprint to conduct software testing activities as a defined process, which is minutely monitored and controlled by the test manager.
- A test plan will include the following.

1. Introduction to the test plan document

2. Assumptions when testing application

3. List of test cases included in testing the application

4. List of features to be tested

5. What sort of approach to use when testing the software

6. List of deliverables that need to be tested

7. They resources allocated for testing the application

8. Any risks involved during the testing process

9. A schedule of tasks and milestones as testing is stared

- Master test plan: A test plan that typically addresses multiple test levels.

- Phase test plan: A test plan that typically addresses one test phase.

- Specific test plan: In this type of test plan, it is designed for specific types of testing especially non – functional testing.

34. What is priority?

Priority is Relative and Business-Focused. Priority defines the order in which we should resolve a defect. Should we fix it now, or can it wait? This priority status is set by the tester to the developer mentioning the time frame to fix the defect. If high priority is mentioned then the developer has to fix it at the earliest. The priority status is set based on the customer requirements.

- For example: 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.

35. What is severity?

Severity is absolute and Customer-Focused. It is the extent to which the defect can affect the software. In other words it defines the impact that a given defect has on the system.

- **For example**: If an application or web page crashes when a remote link is clicked, in this case clicking the remote link by an user is rare but the impact of application crashing is severe. So the severity is high but priority is low.

36. Bug categories are…

Bugs by severity and priority

To determine bug severity, test engineers consider how strongly it impacts the software functionality, performance, usability, etc. and how frequently it occurs. According to this classification, bugs can be critical, high-, medium-, and low-severity .

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

are various methodologies present in agile testing and those are listed below: Scrum

eXtreme programming

Below listed methodologies are used less frequently

• Dynamic system development method (DSDM): This is an iterative and incremental approach that emphasizes on the continuous user involvement.

• Total driven development (TDD): This is a technique which has short iterations where new test cases covering the desired improvement or new functionality are written first.

• Feature driven development: This is an iterative and incremental software development process and this can aim depends on the features.

• XBreed: Agile enterprise previously known as Xbreed. It is agile way of managing, architecting and monitoring the enterprise

• Crystal: Crystal is an adaptive techniques mainly used for software development methodologies.

38. Advantages of Bugzilla….

Bugzilla is open source issue/bug tracking system.

- Bugzilla is defect tracking tools.

-This open bug tracker enables users stay connected with their clients, employees to communicate about problems effectively throughout data management tools.

-Advanced search capabilities

-E-mail notifications

-Modify/file bug by e-mail

-Time tracking, strong security, customization, localization.

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

commonproblems faced in Web testing:

- **general testing problems** that are not specific to any type of testing, but apply to all different types of testing.
- **test type-specific problems** that are specific to a single type of testing such as unit testing, integration testing, and system testing.

Authentication	Authorization
In the authentication process, the identity of users are checked for providing the access to the system.	While in authorization process, a the person's or user's authorities are checked for accessing the resources.
In the authentication process, users or persons are verified.	While in this process, users or persons are validated.
It is done before the authorization process.	While this process is done after the authentication process.
It needs usually the user's login details.	While it needs the user's privilege or security levels.
Authentication determines whether the person is user or not.	While it determines What permission does the user have?
Generally, transmit information through an ID Token.	Generally, transmit information through an Access Token.
The OpenID Connect (OIDC) protocol is an authentication protocol that is generally in charge of user authentication process.	The OAuth 2.0 protocol governs the overall system of user authorization process.
The user authentication is visible at user end.	The user authorization is not visible at the user end.

The user authentication is identified with username, password, face recognition, retina scan, fingerprints, etc.	The user authorization is carried out through the access rights to resources by using roles that have been pre-defined.
Example: Employees in a company are required to authenticate through the network before accessing their company email.	**Example:** After an employee successfully authenticates, the system determines what information the employees are allowed to access.

40. When to used Usablity Testing?

-If possible, usability testing can and should be conducted on the current iteration of a product before beginning any new design work, after you've begun the strategy work around a brand new site or app.

-This will quickly identify areas for opportunity, and reduce the amount of assumptions your design team will make with regard to what the user wants.

Types:

Exploratory testing, accessibility testing, UI testing.