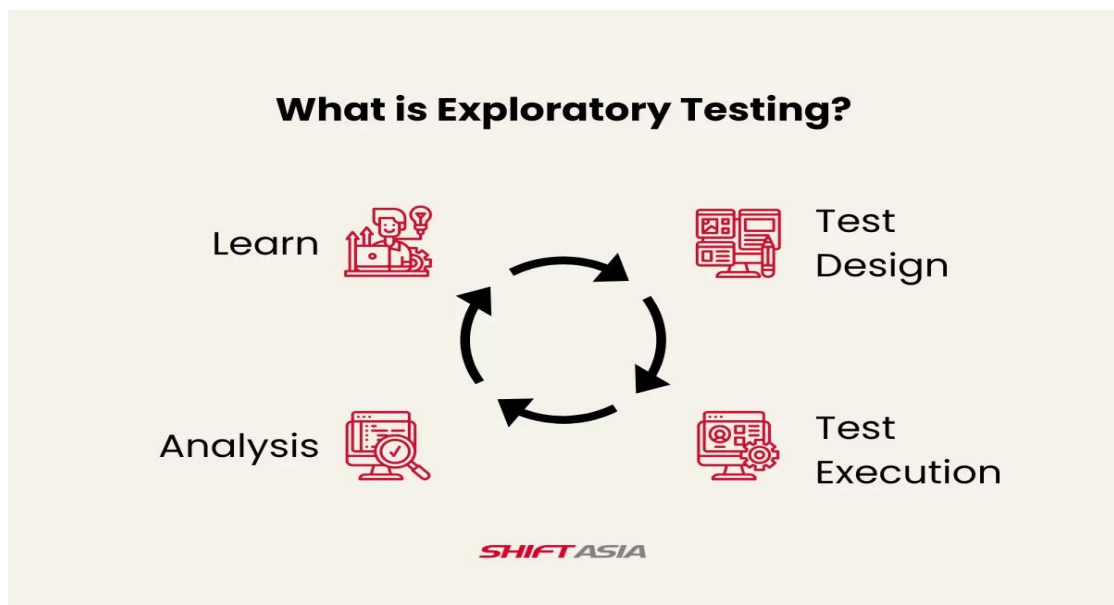


Software Testing Assignment

Module–2(Manual Testing)

1. What is Exploratory Testing?

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



2. What is Traceability matrix?

To protect against changes, you should be able to trace back from every system component to the original requirement that caused its presence. A software process should help you keeping the virtual table up-to-date

3. What is Boundary value testing?

Boundary value testing is a type of Black Box Testing. It checks for the input values near the boundary that have a higher chance of error. Every partition has its maximum and minimum values and these maximum and minimum values are the boundary values of a partition.

4. What is Equivalence partitioning testing?

Equivalence Partitioning testing is a type of Black Box Testing Method it 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. In this method equivalence classes are evaluated for given input conditions. Whenever any input is given, then type of input condition is checked, then for this input conditions, Equivalence class represents or describes set of valid or invalid states.

5. What is Integration Testing?

Integration testing is a software testing technique that focuses on verifying the interactions and data exchange between different components or modules of a software application. The goal of integration testing is to identify any problems or bugs that arise when different components are combined and interact with each other. Integration testing is typically performed after unit testing and before system testing. It helps to identify and resolve integration issues early in the development cycle, reducing the risk of more severe and costly problems later on.

6. What determines the level of risk?

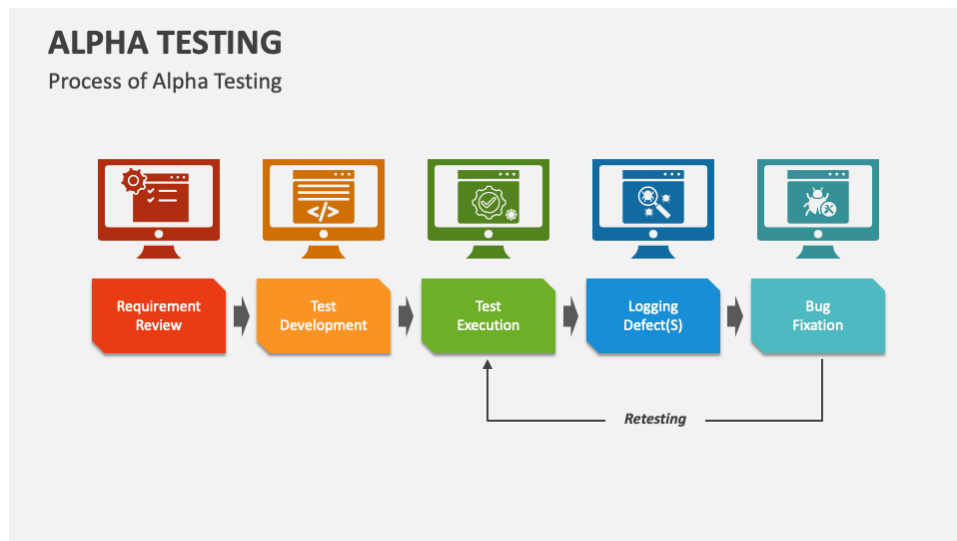
A Risk could be any future event with a negative consequence. You need to identify the risks associated with your project; its types are as follows

- **Project risk:** Project risk refers to uncertainties and risks encountered in the execution of a specific project. Risk has to face during the development. For example, senior developer can take exit in between the project development.
- **Product Risk:** risk refers to all the uncertainties and risks that can cause a product development effort to be unsuccessful. Risk has to face after the development. For example, product launch on release date & next date same product available in the market.

7. What is Alpha Testing?

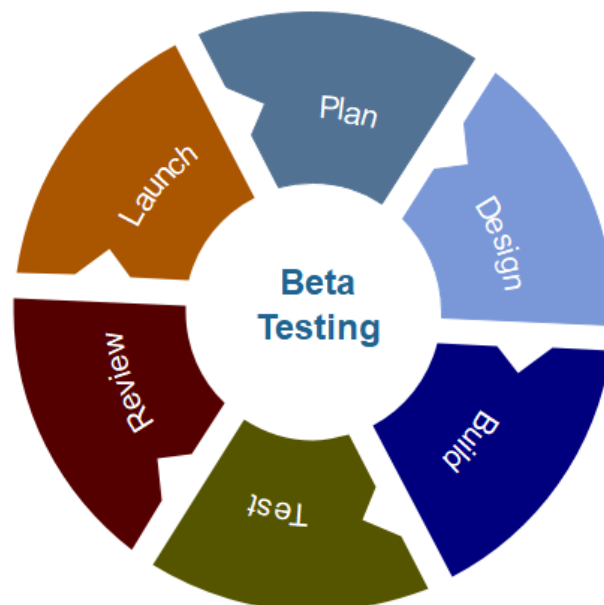
Alpha Testing is a type of software testing performed to identify bugs before releasing the product to real users or to the public. Alpha Testing is one of the user acceptance testings. This is referred to as alpha testing only because it is done early on, near the

end of the development of the software.



8. What is Beta Testing?

Beta Testing is performed by real users of the software application in a real environment. Beta testing is one of the types of User Acceptance Testing. A Beta version of the software, whose feedback is needed, is released to a limited number of end-users of the product to obtain feedback on the product quality. Beta testing helps in minimization of product failure risks and it provides increased quality of the product through customer validation. It is the last test before shipping a product to the customers.

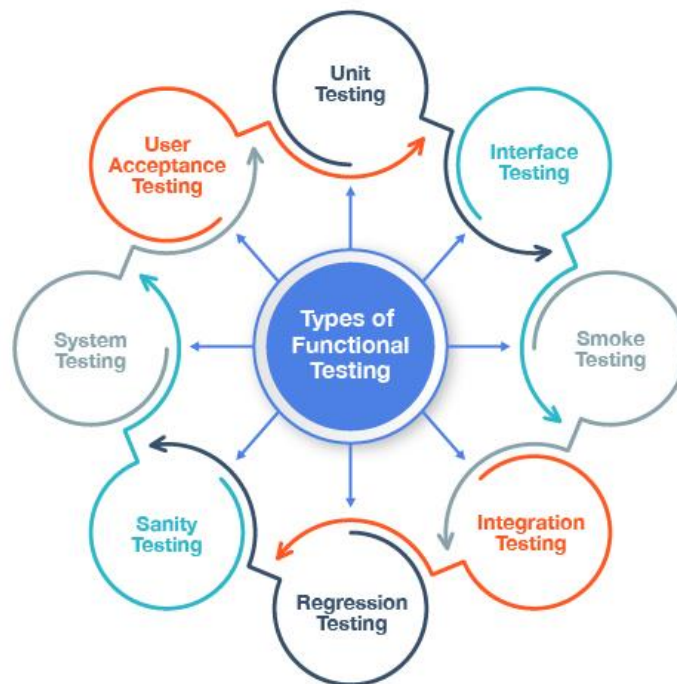


9. What is Component Testing?

Component Testing is a type of software testing in which usability of each individual component is tested. Along with the usability test, behavioural evaluation is also done for each individual component. To perform this type of testing, each component needs to be in independent state and also should be in controllable state. Each component of the software should be user comprehensible.

10. What is Functional Testing?

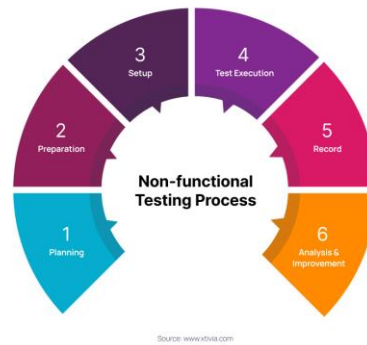
Functional Testing is a type of Software Testing in which the system is tested against the functional requirements and specifications. Functional testing ensures that the application properly satisfies the requirements or specifications.



11. What is Non-Functional Testing?

Non-functional testing is a software testing technique that checks the non-functional attributes of the system. It is designed to test the readiness of a system as per nonfunctional parameters which are never addressed by functional testing.

Process of Non-Functional Testing. Main Steps



12.What is GUI Testing?

Graphical User Interface Testing (GUI) Testing is the process for ensuring proper functionality of the graphical user interface (GUI) for a specific application. GUI testing generally evaluates a design of elements such as layout, colors and also fonts, font sizes, labels, text boxes, text formatting, captions, buttons, lists, icons, links, and content.

13.What is Ad hoc Testing?

Ad hoc testing is a type of software testing that is performed informally and randomly after the formal testing is completed to find any loophole in the system.



14.What is Load Testing?

Load testing determines the behaviour of the application when multiple users use it at the same time. It is the response of the system measured under varying load condition

15.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

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

White box testing is also known as structural testing or code-based testing, and it is used to test the software's internal logic, flow, and structure. The tester creates test cases to examine the code paths and logic flows to ensure they meet the specified requirements. There are 3 types as bellow,

- Statement Coverage
- Decision Coverage
- Condition Coverage.

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

Black-box testing is a type of software testing in which the tester is not concerned with the internal knowledge or implementation details of the software but rather focuses on validating the functionality based on the provided specifications or requirements.

There are number of Black Box techniques are as bellow,

- Equivalence partitioning
- Boundary value analysis
- Decision tables
- State transition testing

18.Mention what are the categories of defects?

Categories of defects are:

- Error of Commission

- Error of Omission
- Error of Clarity
- Error of Speed or Capacity

19.Mention what Big bang testing is?

Big bang integration testing is a testing approach where all components or modules are integrated and tested as a single unit. This is done after all modules have been completed and before any system-level testing is performed.

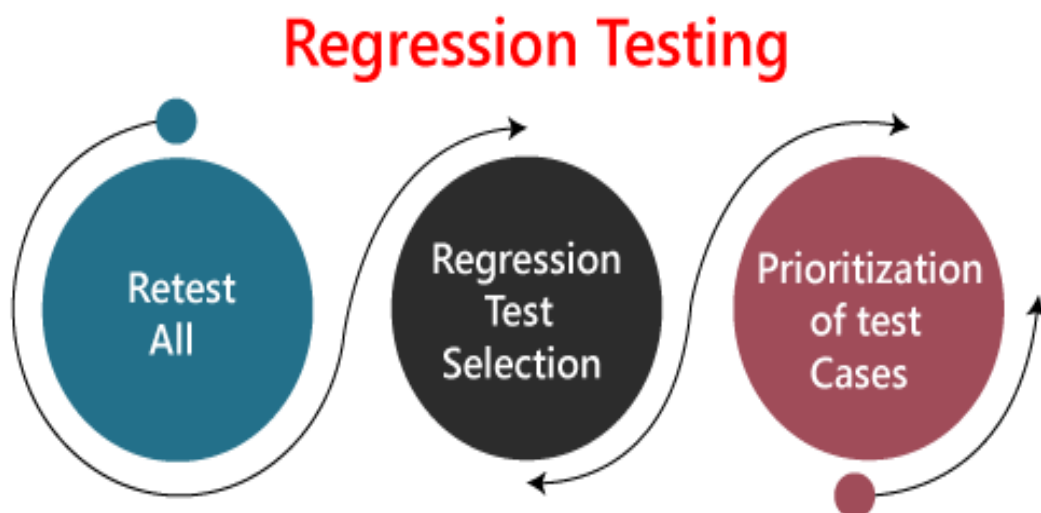
20.What is the purpose of Exit criteria?

Software testing teams will use exit criteria to determine if a test plan or project can exit to the next stage or be considered complete.

- All the high priority bugs are fixed.
- The rate at which bugs are found is too small.
- The testing budget is exhausted.
- The project duration is completed.
- The risk in the project is under acceptable limit.

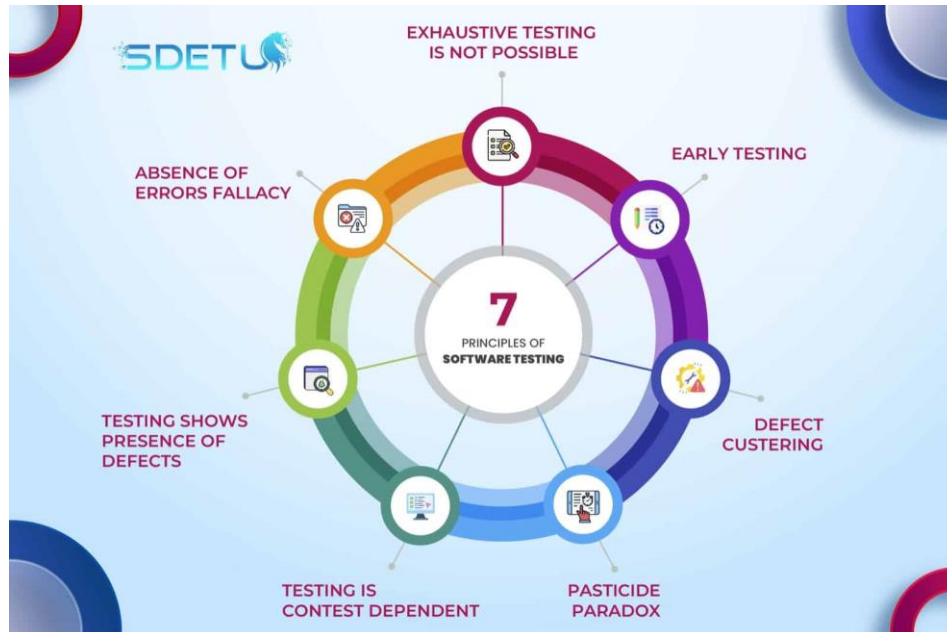
21.When should "Regression Testing" be performed?

Regression testing is performed before each new instance of the product is deployed, guaranteeing that the program works perfectly in each setting.



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

The seven principles of testing help you set optimum quality standards and give you and your customers confidence that the software is ready for action.7 Key Principles are as follows



1. Testing shows the presence of defects:

Testing can show that defects are present, but cannot prove that there are no defects. Even multiple tests 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 impossible:

Testing everything including all combinations of inputs and preconditions is not possible. Each and every test case cannot test by the test engineer its too complex, too expensive and takes too long to complete

3. Early Testing:

software testing will start at the initial phase of development, defect reported in the early phases of SDLC will be very less expensive and better performance of Software. Software testing will start from Requirement Gathering phase to continue till the Deployment of the Software.

4. Defect clustering:

Small number of modules contain most of the defects. approximately 80% of the problems are found in 20% of the modules.

5. Pesticide Paradox:

If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new defects. So, it is necessary to review the test cases and add or update test cases for finding new bugs.

6. Testing is Context-Dependent:

Different types of software need to perform different types of testing. For example, the testing of the e-commerce site is different from the testing of the Android application.

7. Absence of Errors Fallacy:

If a built software is 99% bug-free but does not satisfy the user requirement then it is unusable. It's not about 99% bug free software but to fulfil all the customer requirements.

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

QA	QC	TESTER
<ul style="list-style-type: none">QA includes activities that ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements.	<ul style="list-style-type: none">Activities which ensure the verification of developed software with respect to documented (or not in some cases) requirements.	<ul style="list-style-type: none">The primary objective of software testing is to identify defects or bugs in the software.

<ul style="list-style-type: none"> • Focuses on processes and procedures rather than conducting actual testing on the system. 	<ul style="list-style-type: none"> • Focuses on actual testing by executing Software with intend to identify bug/defect through implementation of procedures and process. 	<ul style="list-style-type: none"> • Testing is typically carried out after the software has been developed.
<ul style="list-style-type: none"> • Process oriented activities. 	<ul style="list-style-type: none"> • Product oriented activities. 	<ul style="list-style-type: none"> • Testers are responsible for conducting testing activities. They execute test cases, report defects, and ensure that the software functions as expected.
<ul style="list-style-type: none"> • Preventive activities. 	<ul style="list-style-type: none"> • It is a corrective process. 	<ul style="list-style-type: none"> • The primary deliverables of testing are test plans, test cases, test reports, and defect logs. These artifacts provide information about the software's quality and reliability.
<ul style="list-style-type: none"> • It is a subset of Software Test Life Cycle (STLC). 	<ul style="list-style-type: none"> • QC can be considered as the subset of Quality Assurance 	<ul style="list-style-type: none"> • Testing is more focused on short-term goals, such as identifying and fixing defects in the current software release.

24.Difference between Smoke and Sanity

Smoke	Sanity
<ul style="list-style-type: none"> • Smoke Testing is performed to ascertain that the critical 	<ul style="list-style-type: none"> • Sanity Testing is done to check the new functionality / bugs have

functionalities of the program is working fine	been fixed
<ul style="list-style-type: none"> The objective of this testing is to verify "stability" of the system in order to with more rigorous testing 	<ul style="list-style-type: none"> The objective of the testing is to verify the "rationality" of the system in order proceed to proceed with more rigorous testing
<ul style="list-style-type: none"> This testing is performed by the developers or testers 	<ul style="list-style-type: none"> Sanity testing is usually performed by testers
<ul style="list-style-type: none"> Smoke testing is usually documented or scripted. 	<ul style="list-style-type: none"> Sanity testing is usually not documented and is unscripted
<ul style="list-style-type: none"> Smoke testing is subset of Acceptance testing. 	<ul style="list-style-type: none"> Sanity Testing is subset of Regression testing.

25.Difference between Verification and Validation

Verification	Validation
<ul style="list-style-type: none"> Verification is a process which is performed at development level. 	<ul style="list-style-type: none"> Validation is a process which is performed at testing level.
<ul style="list-style-type: none"> Verification phases are: Business Requirement Analysis System Design/ System Requirement Architectural Design (Technical Specification) 	<ul style="list-style-type: none"> Validation Phases are: Unit Testing Integration Testing System Testing Acceptance Testing

Module Design (Program Specification)	
<ul style="list-style-type: none"> It is the process of evaluating product of development to check whether the specified requirements meet or not. 	<ul style="list-style-type: none"> It is the process of evaluating the product of development to check whether it satisfied business requirements or not.
<ul style="list-style-type: none"> Verification can be achieved by asking “Are you building a product, right?” 	<ul style="list-style-type: none"> Validation can be achieved by asking “Are you building a right product?”
<ul style="list-style-type: none"> Verification activities are Reviews and Inspections. 	<ul style="list-style-type: none"> Validation activity is Testing.

26.Explain types of Performance Testing

- 1. Load Testing:** Load testing simulates a real-world load on the system to see how it performs under stress. It helps identify bottlenecks and determine the maximum number of users or transactions the system can handle.
- 2. Stress Testing:** Stress testing is a type of load testing that tests the system’s ability to handle a high load above normal usage level.
- 3. Spike Testing:** Spike testing is a type of load testing that tests the system’s ability to handle sudden spikes in traffic. It helps identify any issues that may occur when the system is suddenly hit with a high number of requests.
- 4. Soak Testing:** Soak testing is a type of load testing that tests the system’s ability to handle a sustained load over a prolonged period.
- 5. Endurance Testing:** Endurance testing is similar to soak testing, but it focuses on the long-term behaviour of the system under a constant load. It is performed to ensure the software can handle the expected load over a long period.
- 6. Volume Testing:** In Volume testing, a large number of data is saved in a database and the overall software system’s behaviour is observed.

- 7. Scalability Testing:** In Scalability testing, the software application's effectiveness is determined by scaling up to support an increase in user load. It helps in planning capacity additions to your software system.

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

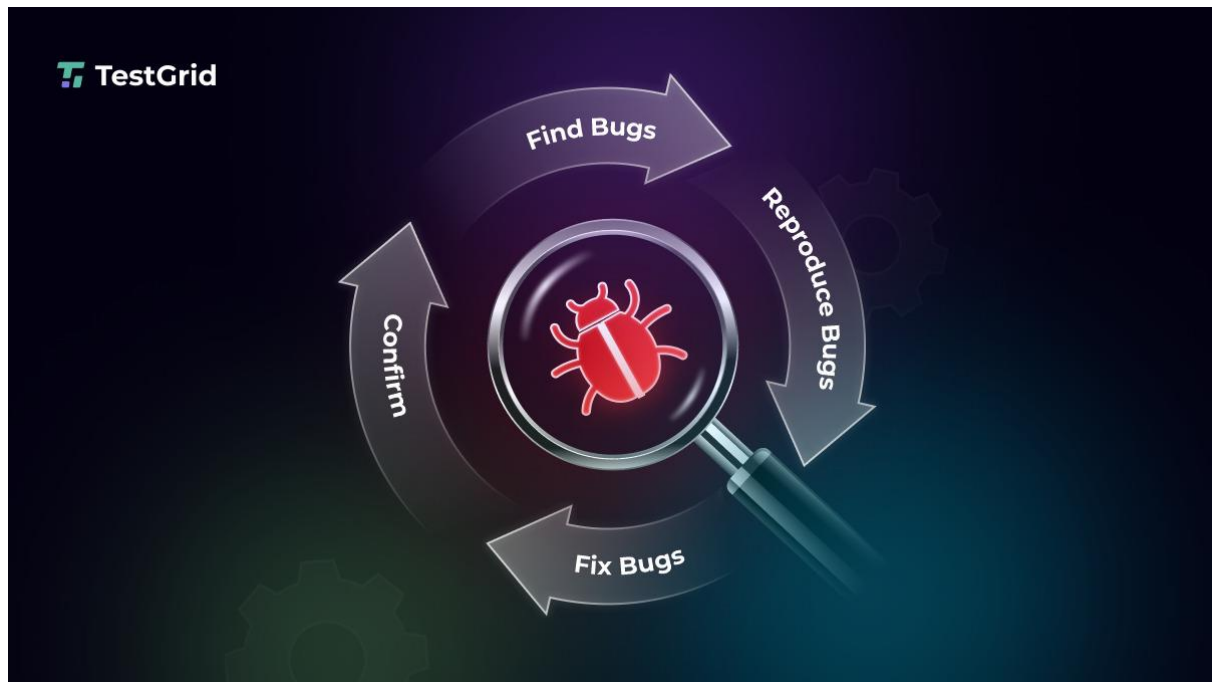
- **Error:** Defect accepted by the developer/development team.
- **Bug:** Defect accepted by the developer/development team.
- **Defect:** Identify error during the testing of Software
- **Failure:** When Product does not meet the customers, requirement called Failure

28.Difference between Priority and Severity

Priority	Severity
<ul style="list-style-type: none">• Priority is a parameter to decide the order in which defects should be fixed.	<ul style="list-style-type: none">• Severity is a parameter to denote the impact of a particular defect on the software
<ul style="list-style-type: none">• Priority means how fast the defect has to be fixed.	<ul style="list-style-type: none">• Severity means how severe the defect is affecting the functionality.
<ul style="list-style-type: none">• Priority is related to scheduling to resolve the problem.	<ul style="list-style-type: none">• Severity is related to the quality standard.
<ul style="list-style-type: none">• The product manager decides the priorities of defects.	<ul style="list-style-type: none">• The testing engineer decides the severity level of the defect.
<ul style="list-style-type: none">• It is associated with scheduling.	<ul style="list-style-type: none">• It is associated with functionality or standards.
<ul style="list-style-type: none">• It indicates how soon the bug should be fixed.	<ul style="list-style-type: none">• It indicates the seriousness of the bug in the product functionality.

29.What is Bug Life Cycle?

A defect/bug life cycle is the sequence of steps a bug or defect goes through from its identification to its resolution in software development.



30.Explain the difference between Functional testing and Non-functional testing

Functional testing	Non-functional testing
<ul style="list-style-type: none">• It is performed using the functional specification provided by the client and verify the system against the functional requirements.	<ul style="list-style-type: none">• Testing checks the performance, reliability, scalability, and other non-functional aspects of the software system.
<ul style="list-style-type: none">• Functional testing is executed first	<ul style="list-style-type: none">• Non-functional testing should be performed after functional testing
<ul style="list-style-type: none">• Manual testing or automation tools can be used for functional testing	<ul style="list-style-type: none">• Using tools will be effective for this testing
<ul style="list-style-type: none">• Functional testing describes	<ul style="list-style-type: none">• Nonfunctional testing describes how good the product works

what the product does	
<ul style="list-style-type: none"> • Easy to do manual testing 	<ul style="list-style-type: none"> • Tough to do manual testing
<ul style="list-style-type: none"> • Types of Functional Testing are Unit Testing Smoke Testing Sanity Testing Integration Testing White box testing Black Box testing User Acceptance testing Regression Testing 	<ul style="list-style-type: none"> • Types of Non-Functional Testing are Performance Testing Load Testing Volume Testing Stress Testing Security Testing Installation Testing Penetration Testing Compatibility Testing Migration Testing

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

SDLC	STLC
<ul style="list-style-type: none"> • Focused on software development 	<ul style="list-style-type: none"> • Focused on software testing
<ul style="list-style-type: none"> • Helps to develop good quality software. 	<ul style="list-style-type: none"> • Helps to make software defects free.
<ul style="list-style-type: none"> • SDLC phases are completed before the STLC phases. 	<ul style="list-style-type: none"> • STLC phases are performed after SDLC phases.
<ul style="list-style-type: none"> • Coders create a well-organized development plan. 	<ul style="list-style-type: none"> • QA team defines the test plan.
<ul style="list-style-type: none"> • Developers create the actual software. 	<ul style="list-style-type: none"> • Tester designs test cases, set up the environment & work out the RTM.

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

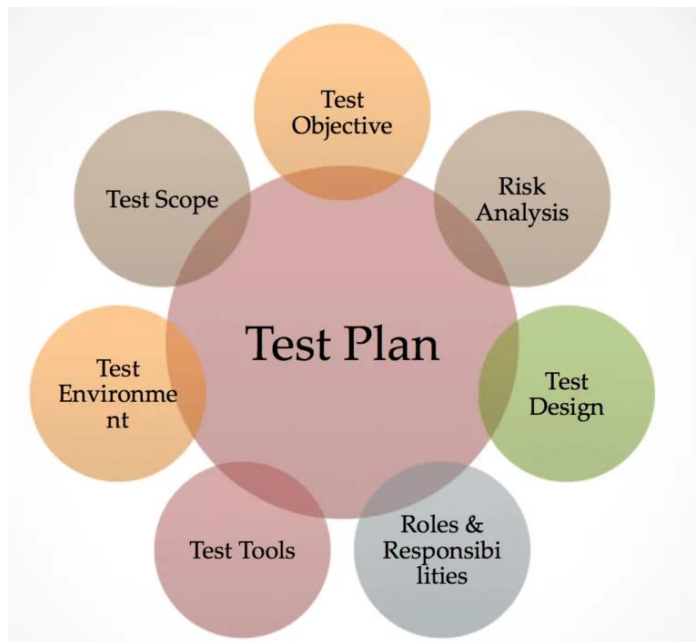
--	--	--

Scenarios	Test Case	Test Script
<ul style="list-style-type: none"> Is any functionality that can be tested 	<ul style="list-style-type: none"> Is a set of actions executed to verify particular features or functionality 	<ul style="list-style-type: none"> It is a set of instruction that how to execute core business function.
<ul style="list-style-type: none"> It is derived from Business requirement and Software Requirement Specifications (SRS) 	<ul style="list-style-type: none"> It is mostly derived from Test Scenarios 	<ul style="list-style-type: none"> It is mostly derived from Test cases
<ul style="list-style-type: none"> It is more focus on what to test 	<ul style="list-style-type: none"> It is more focus on what and how to test 	<ul style="list-style-type: none"> It is focus on expected results
<ul style="list-style-type: none"> It takes less time and fewer resources to create 	<ul style="list-style-type: none"> Required more resources and time 	<ul style="list-style-type: none"> Require less time to testing but more resources for script creating and updating
<ul style="list-style-type: none"> Allow quickly accessing the testing scope 	<ul style="list-style-type: none"> Allow Detecting Errors and defects 	<ul style="list-style-type: none"> Allows carrying out and automatic execution of test cases

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

A test plan is a document that consists of all future testing-related activities. It is prepared at the project level and in general, it defines work products to be tested, how they will be tested, and test type distribution among the testers.

- The test plan serves as the blueprint that changes according to the progressions in the project and stays current at all times.
- It serves as a base for conducting testing activities and coordinating activities among a QA team.
- It is shared with Business Analysts, Project Managers, and anyone associated with the project.



34.What is Priority?

Priority is defined as a parameter that decides the order in which a defect should be fixed. Defects having a higher priority should be fixed first.

35.What is severity?

Severity is defined as the extent to which a particular defect can create an impact on the software. Severity is a parameter to denote the implication and the impact of the defect on the functionality of the software.

36.What are the Bug Categories

- There is number of bug categories are as bellow,
- Security
- Database
- Functionality (General/Critical)
- UI

37.Advantages of Bugzilla

Bugzilla is an open-source bug-tracking system. it is used as a bug-reporting tool in the market. Advantages of Bugzilla

Are as follows,

- **Deadlines:** To fix the bugs, deadlines can be established.
- **Types:** It reports in a variety of formats and types.
- **Request System:** You can use the ‘request system’ provided by Bugzilla to ask other users to evaluate codes, provide information and other things.
- **Flexible:** Bugzilla is quite flexible, so you can modify it to fit your unique process and requirements.
- **Bug tracking tool:** Bugzilla is extremely good at monitoring and handling bugs and issues.

38. Difference between Priority and Severity

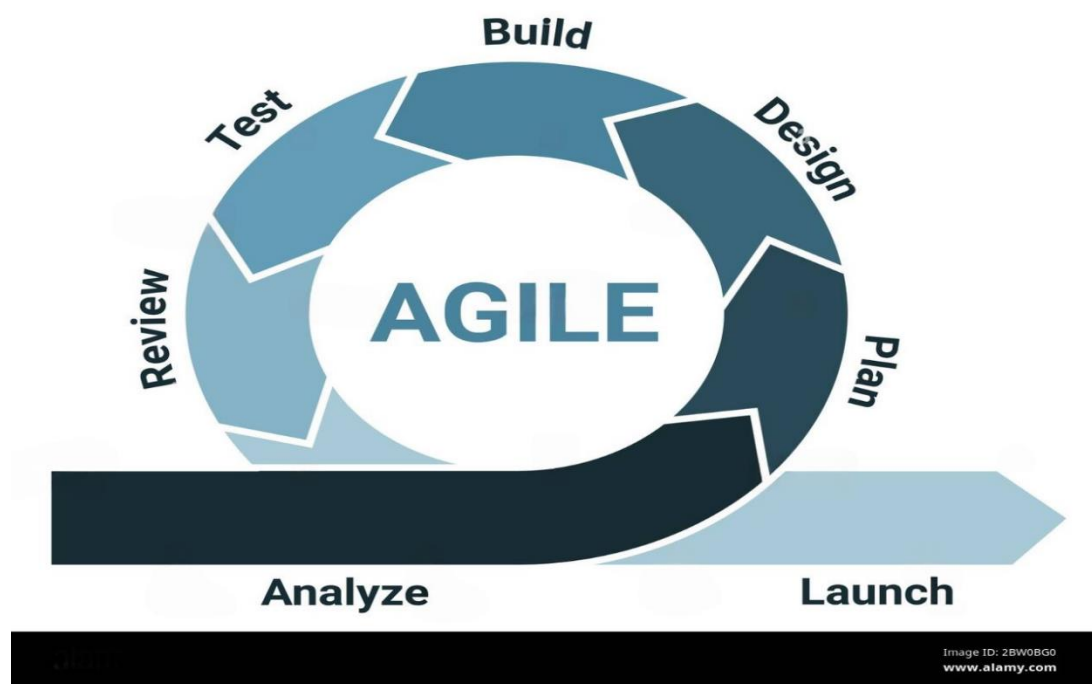
Priority	Severity
<ul style="list-style-type: none">• Priority is a parameter to decide the order in which defects should be fixed.	<ul style="list-style-type: none">• Severity is a parameter to denote the impact of a particular defect on the software
<ul style="list-style-type: none">• Priority means how fast the defect has to be fixed.	<ul style="list-style-type: none">• Severity means how severe the defect is affecting the functionality.
<ul style="list-style-type: none">• Priority is related to scheduling to resolve the problem.	<ul style="list-style-type: none">• Severity is related to the quality standard.
<ul style="list-style-type: none">• The product manager decides the priorities of defects.	<ul style="list-style-type: none">• The testing engineer decides the severity level of the defect.
<ul style="list-style-type: none">• It is associated with scheduling.	<ul style="list-style-type: none">• It is associated with functionality or standards.
<ul style="list-style-type: none">• It indicates how soon the bug should be fixed.	<ul style="list-style-type: none">• It indicates the seriousness of the bug in the product functionality.

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

The goal of every Agile methodology is to embrace and adapt to change while delivering working software as efficiently as possible. However, each method varies in

the way it defines the steps of software development. The most widely used Agile methods include the following:

- Scrum
- Lean software development
- Extreme programming
- Crystal
- Kanban
- Dynamic systems development method
- Feature-driven development



40.Difference between Authorization and Authentication in web testing, what are the common problem in web testing?

Authentication	Authorization
<ul style="list-style-type: none">• In the authentication process, the identity of users is checked for providing the access to the	<ul style="list-style-type: none">• While in authorization process, the person's or user's authorities are checked for accessing the

system.	resources.
<ul style="list-style-type: none"> • In the authentication process, users or persons are verified. 	<ul style="list-style-type: none"> • While in this process, users or persons are validated.
<ul style="list-style-type: none"> • It is done before the authorization process. 	<ul style="list-style-type: none"> • While this process is done after the authentication process.
<ul style="list-style-type: none"> • It needs usually the user's login details 	<ul style="list-style-type: none"> • While it needs the user's privilege or security levels
<ul style="list-style-type: none"> • Authentication determines whether the person is user or not. 	<ul style="list-style-type: none"> • While it determines What permission does the user have?

The quality assurance of Web applications is always a tough task because there are so many factors which can affect the quality of these applications. The main critical challenges in the Web based applications are listed based on their criticality rank are:

1. Integration
2. Interoperability
3. Security
4. Performance
5. Usability

41. When to use Usability Testing?

Usability Testing During a Redesign Getting a user to run over a design before you put into production will let you know if an idea has the mileage to really be valuable or whether you've misinterpreted what users really wanted from the product

42. What is the procedure of GUI Testing?

Graphical User Interface Testing (GUI) Testing is the process for ensuring proper functionality of the graphical user interface (GUI) for a specific application. GUI testing generally evaluates a design of elements such as layout, colors and also fonts, font sizes, labels, text boxes, text formatting, captions, buttons, lists, icons, links, and content. GUI testing processes may be either manual or automatic and are often performed by third-party companies, rather than developers or end users.

- MANUAL BASED TESTING
- RECORD AND REPLAY
- MODEL BASED TESTING