Module: 2

1. What is Exploratory Testing?

- Exploratory testing is a concurrent process where Test design, execution and logging happen simultaneously.

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 Analysis is based on testing the boundary values of valid and invalid partitions. The behavior at the edge of the equivalence partition is more likely to be incorrect than the behavior within the partition, so boundaries are an area where testing is likely to yield defects.
- 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 Method 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.
- In equivalence partitioning, 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 - Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems

6. What determines the level of risk?

- ‘A factor that could result in future negative consequences; usually expressed as impact and likelihood’

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.
- It is always performed by the developers at the software development site.

8. What is beta testing?

- Beta Testing is performed by real users of the software application in a real environment.
- Beta Testing (field testing) is performed and carried out by users or you can say people at their own locations and site using customer data.

9. What is component testing?

- A minimal software item that can be tested in isolation.
- It means “A unit is the smallest testable part of software.”
- Component Testing – The testing of individual software components.

10. What is functional system testing?

- Testing based on an analysis of the specification of the functionality of a component or system.
- Functional testing verifies that each function of the software application operates in conformance with the requirement specification.

11. What is Non-Functional Testing?

- Testing the attributes of a component or system that do not relate to functionality, e.g. reliability, efficiency, usability, interoperability, maintainability and portability.

12. What is GUI Testing?

- Graphical User Interface (GUI) testing is the process of testing the system’s GUI of the System under Test.
- 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.

13. What is Ad hoc testing?

- Adhoc testing is an informal testing type with an aim to break the system.
- In fact is does not create test cases altogether! This testing is primarily performed if the knowledge of testers in the system under test is very high.
- Main aim of this testing is to find defects by random checking. Adhoc testing can be achieved with the testing technique called Error Guessing.

14. What is load testing?

- Its a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system’s response time degrades or fails.
- This testing usually identifies –
- The maximum operating capacity of an application
- Determine whether current infrastructure is sufficient to run the application
- Sustainability of application with respect to peak user load
- Number of concurrent users that an application can support, and scalability to allow more users to access it.
- It is a type of non-functional testing. Load testing is commonly used for the Client/Server, Web based applications – both Intranet and Internet.

15. What is stress Testing?

- System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load.
- Stress Testing is done to make sure that the system would not crash under crunch situations. ->Stress testing is also known as endurance testing.

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

- Testing based on an analysis of the internal structure of the component or system. ---> Structure-based testing technique is also known as ‘white-box’ or ‘glass-box’ testing technique because here the testers require knowledge of how the software is implemented, how it works
- List the white box testing

<u>1 Statement coverage</u>

- The statement coverage is also known as line coverage or segment coverage.
- The statement coverage covers only the true conditions.

<u>2 Decision coverage</u>

- Decision coverage also known as branch coverage or all-edges coverage.
- It covers both the true and false conditions unlikely the statement coverage.

<u>3 Condition coverage</u>

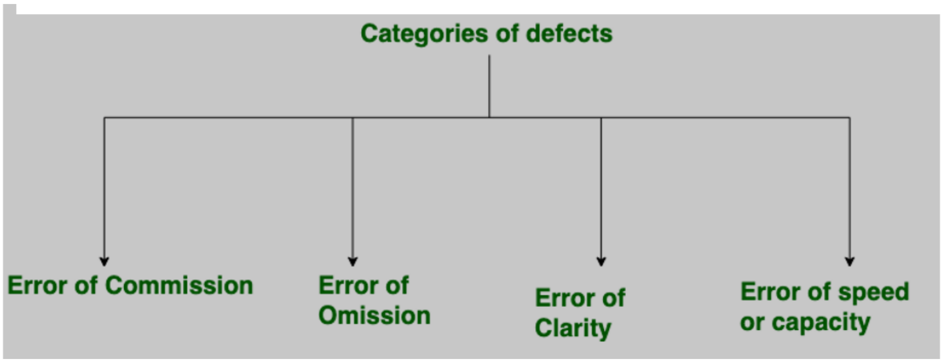
- A decision is an IF statement, a loop control statement (e.g. DOWHILE or REPEAT-UNTIL, JUMP, GO TO), or a CASE statement, where there are two or more outcomes from the statement.

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

- The technique of testing without having any knowledge of the interior workings of the application is Black Box testing.
- The tester is oblivious to the system architecture and does not have access to the source code.
- Typically, when performing a black box test, 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.
- Black box testing Techniques:
- Equivalence partitioning
- Boundary value analysis
- Decision tables
- State transition testing
- Use-case Testing
- Other Black Box Testing

  -Syntax or Pattern Testing

18. Mention what are the categories of defects?



19. Mention what bigbang testing is?

- Big Bang integration testing all components or modules is integrated simultaneously, after which everything is tested as a whole.
- Big Bang testing has the advantage that everything is finished before integration testing starts.

20. What is the purpose of exit criteria?

- Deadlines meet or budget depleted.
- Execution and updating of all test cases.
- Desired and sufficient coverage of the requirements and functionalities under the test.

- All the identified defects are corrected and closed.
- No high priority or severity or critical bug has been left out.

21. When should "Regression Testing" be performed?

- Testing of a previously tested program following modification to ensure that defects have not been introduced or uncovered in unchanged areas of the software, as a result of the changes made. It is performed when the software or its environment is changed.
- This testing is done to make sure that new code changes should not have side effects on the existing functionalities. It ensures that old code still works once the new code changes are done.

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

- 7key principle

1. Testing shows presence of Defects
2. Exhaustive Testing is Impossible!
3. Early Testing
4. Defect Clustering
5. The Pesticide Paradox
6. Testing is Context Dependent
7. Absence of Errors Fallacy

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 remaining in the software but, even if no defects are found, it is not a proof of correctness.
- We test to find Faults As we find more defects, the probability of undiscovered defects remaining in a system reduces.
- However Testing cannot prove that there are no defects present

2. Exhaustive Testing is Impossible!

- Testing everything including all combinations of inputs and preconditions is not possible.
- So, instead of doing the exhaustive testing we can use risks and priorities to focus testing efforts.
- For example: In an application in one screen there are 15 input fields, each having 5 possible values, then to test all the valid combinations you would need 30 517 578 125 (515) tests.
- This is very unlikely that the project timescales would allow for this number of tests.

- So, accessing and managing risk is one of the most important activities and reason for testing in any project.
- We have learned that we cannot test everything (i.e. all combinations of inputs and pre-conditions). That is we must Prioritise our testing effort using a Risk Based Approach.

3. Early Testing

- Testing activities should start as early as possible in the software or system development life cycle, and should be focused on defined objectives.
- Testing activities should start as early as possible in the development life cycle
- These activities should be focused on defined objectives – outlined in the Test Strategy Remember from our Definition of Testing, that Testing doesn't start once the code has been written!

4. Defect Clustering

- A small number of modules contain most of the defects discovered during pre-release testing, or are responsible for the most operational failures.
- Defects are not evenly spread in a system
- They are 'clustered'
- In other words, most defects found during testing are usually confined to a small number of modules
- Similarly, most operational failures of a system are usually confined to a small number of modules
- An important consideration in test prioritisation!

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.
- To overcome this "pesticide paradox", the test cases need to be regularly reviewed and revised, and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects.
- Testing identifies bugs, and programmers respond to fix them
- As bugs are eliminated by the programmers, the software improves
- As software improves the effectiveness of previous tests erodes

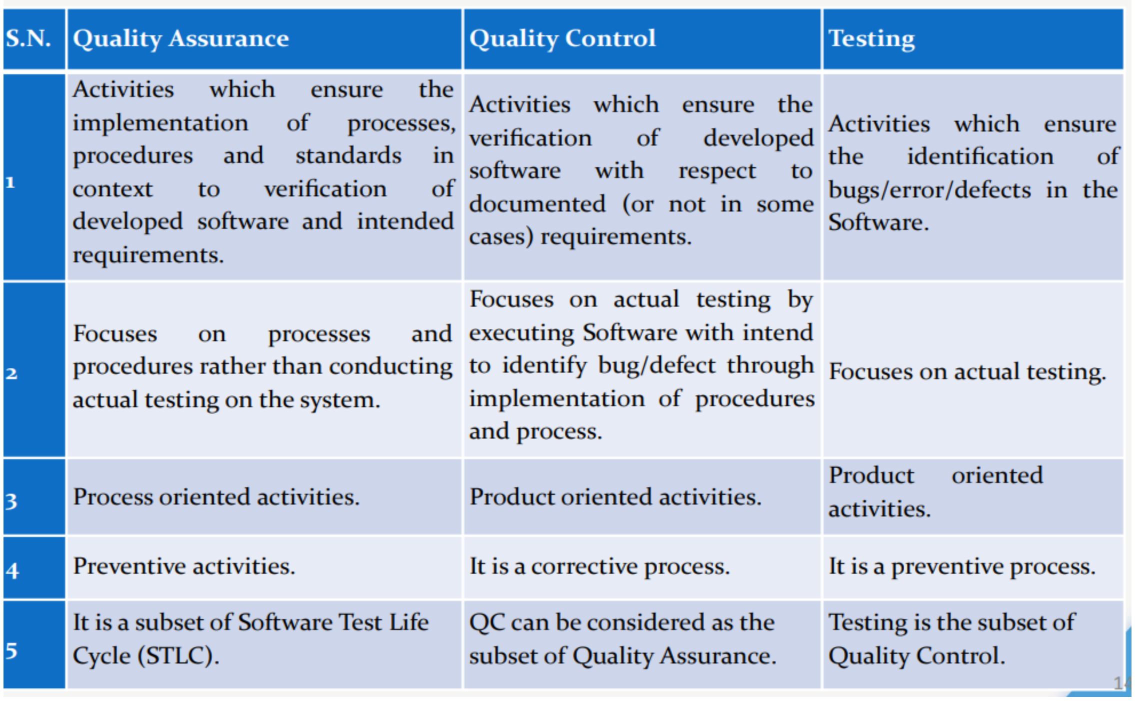
6. Testing is Context Dependent

- Testing is basically context dependent.
- Testing is done differently in different contexts
- Different kinds of sites are tested differently.
- For example Safety – critical software is tested differently from an e-commerce site.

7. Absence of Errors Fallacy

- If the system built is unusable and does not fulfill the user's needs and expectations then finding and fixing defects does not help.
- If we build a system and, in doing so, find and fix defects .... It doesn't make it a good system Even after defects have been resolved it may still be unusable and/or does not fulfil the users' needs and expectations

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

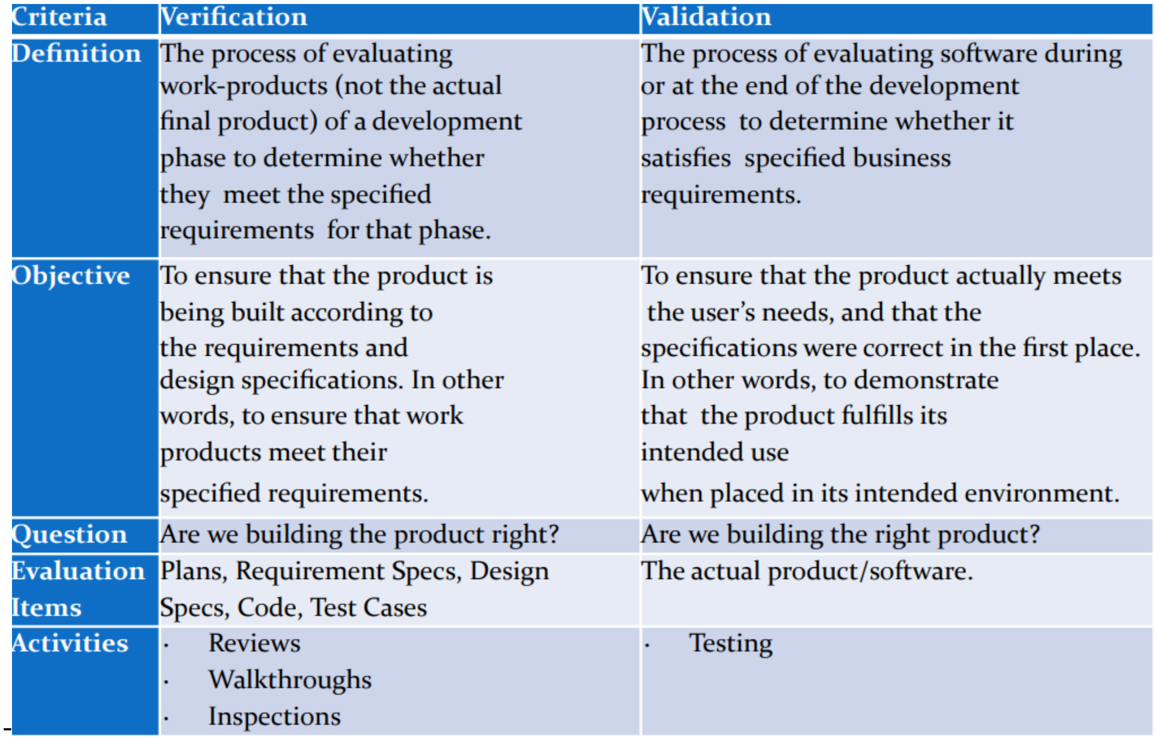


S.N.	Quality Assurance	Quality Control	Testing
1	Activities which ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements.	Activities which ensure the verification of developed software with respect to documented (or not in some cases) requirements.	Activities which ensure the identification of bugs/error/defects in the Software.
2	Focuses on processes and procedures rather than conducting actual testing on the system.	Focuses on actual testing by executing Software with intend to identify bug/defect through implementation of procedures and process.	Focuses on actual testing.
3	Process oriented activities.	Product oriented activities.	Product oriented activities.
4	Preventive activities.	It is a corrective process.	It is a preventive process.
5	It is a subset of Software Test Life Cycle (STLC).	QC can be considered as the subset of Quality Assurance.	Testing is the subset of Quality Control.

24. □ Difference between Smoke and Sanity?

SMOKE TESTING	SANITY TESTING
Smoke Testing is performed to ascertain is working fine	Sanity Testing is done to check the that the critical functionalities of the new functionality / bugs have been fixed program
The objective of this testing is to verify "stability" of the system in order to with more rigorous testing	The objective of the testing is to verify the "rationality" of the system in order proceed to proceed with more rigorous testing
This testing is performed by the developers or testers	Sanity testing is usually performed by testers
Smoke testing is usually documented and or scripted	Sanity testing is usually not documented or unscripted
Smoke testing is a subset of Regression testing	Sanity testing is a subset of Acceptance testing
Smoke testing is like General Health Check up	Sanity Testing is like specialized health check up

25. □ Difference between verification and Validation



Criteria	Verification	Validation
Definition	The process of evaluating work-products (not the actual final product) of a development phase to determine whether they meet the specified requirements for that phase.	The process of evaluating software during or at the end of the development process to determine whether it satisfies specified business requirements.
Objective	To ensure that the product is being built according to the requirements and design specifications. In other words, to ensure that work products meet their specified requirements.	To ensure that the product actually meets the user's needs, and that the specifications were correct in the first place. In other words, to demonstrate that the product fulfills its intended use when placed in its intended environment.
Question	Are we building the product right?	Are we building the right product?
Evaluation Items	Plans, Requirement Specs, Design Specs, Code, Test Cases	The actual product/software.
Activities	· Reviews · Walkthroughs · Inspections	· Testing

26 Explain types of Performance testing.

- Load testing
- Stress testing
- Endurance testing
- Spike testing
- Volume testing
- Scalability testing

1.Load testing

- Its a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system's response time degrades or fails.
- Load testing is a kind of performance testing which determines a system's performance under real-life load conditions. This testing helps determine how the application behaves when multiple users access it simultaneously.

2.Stress testing

- System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load.

- Stress testing is used to test the stability & reliability of the system. This test mainly determines the system on its robustness and error handling under extremely heavy load conditions.

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

ERROR:

- a human action that produces an incorrect result
- A discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition. This can be a misunderstanding of the internal state of the software, an oversight in terms of memory management, confusion about the proper way to calculate a value, etc.

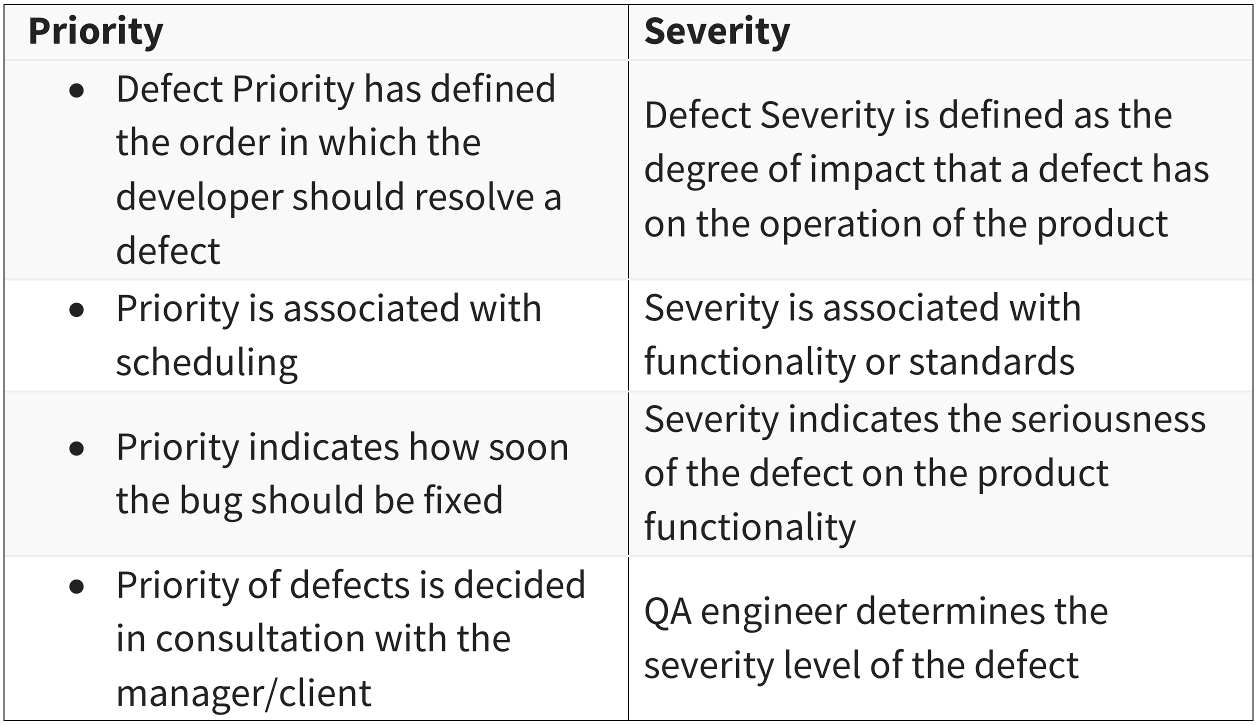
DEFECT:

- A flaw in a component or system that can cause the component or system to fail to perform its required function
- Commonly refers to several troubles with the software products, with its external behavior or with its internal features.

BUG

- A fault in a program which causes the program to perform in an unintended or unanticipated manner. See: anomaly, defect, error, exception, and fault. Bug is terminology of Tester.

28. Difference between Priority and Severity



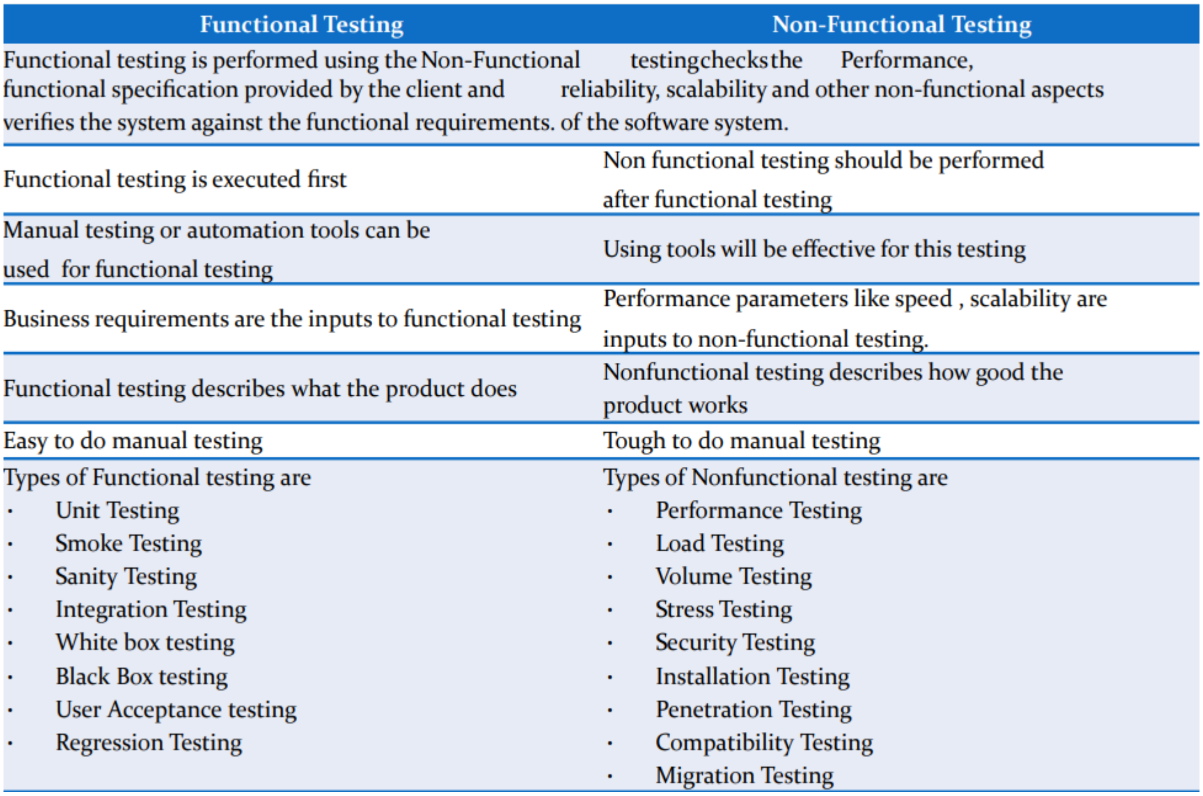
Priority	Severity
• Defect Priority has defined the order in which the developer should resolve a defect	Defect Severity is defined as the degree of impact that a defect has on the operation of the product
• Priority is associated with scheduling	Severity is associated with functionality or standards
• Priority indicates how soon the bug should be fixed	Severity indicates the seriousness of the defect on the product functionality
• Priority of defects is decided in consultation with the manager/client	QA engineer determines the severity level of the defect

• Priority is driven by business value	Severity is driven by functionality

29. What is Bug Life Cycle?

-The bug life cycle in testing refers to a cycle of defects in which it goes through different states throughout its life.
-The life cycle begins with a new defect discovered by a tester while testing the application.
-It continues until the tester discovers a specific solution and closes the bug, so it does not reoccur

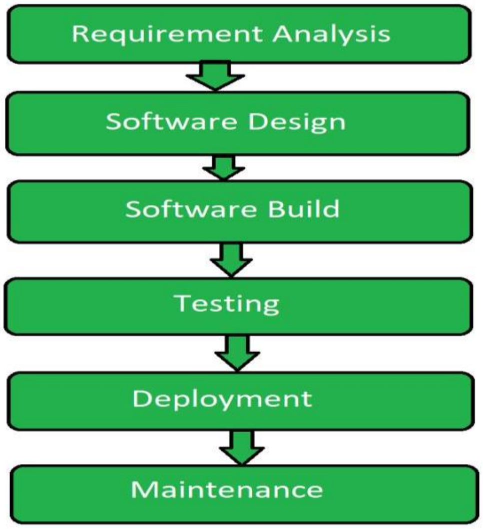
30. Explain the difference between Functional testing and NonFunctional testing



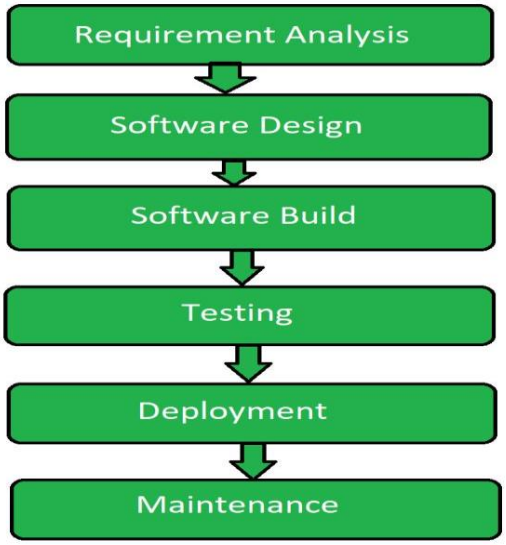
Functional Testing	Non-Functional Testing
Functional testing is performed using the functional specification provided by the client and verifies the system against the functional requirements.	Non-Functional testingchecksthe Performance, reliability, scalability and other non-functional aspects of the software system.
Functional testing is executed first	Non functional testing should be performed after functional testing
Manual testing or automation tools can be used for functional testing	Using tools will be effective for this testing
Business requirements are the inputs to functional testing	Performance parameters like speed , scalability are inputs to non-functional testing.
Functional testing describes what the product does	Nonfunctional testing describes how good the product works
Easy to do manual testing	Tough to do manual testing
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

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

-SDLC(software development life cycle)



-STLC(Software Testing Life Cycle)



SDLC	STLC

SDLC is mainly related to software development.	STLC is mainly related to software testing.
Besides development other phases like testing is also included.	It focuses only on testing the software.
SDLC involves total six phases or steps.	STLC involves only five phases or steps.
In SDLC, more number of members (developers) are required for the whole process.	In STLC, less number of members (testers) are needed.
In SDLC, development team makes the plans and designs based on the requirements.	In STLC, testing team(Test Lead or Test Architect) makes the plans and designs.
Goal of SDLC is to complete successful development of software.	Goal of STLC is to complete successful testing of software.
It helps in developing good quality software.	It helps in making the software defects free.
SDLC phases are completed before the STLC phases.	STLC phases are performed after SDLC phases.
Post deployment support , enhancement , and update are to be included if necessary.	Regression tests are run by QA team to check deployed maintenance code and maintains test cases and automated scripts.

Creation of reusable software systems is the end result of SDLC.	A tested software system is the end result of STLC

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

- A Scenario is any functionality that can be tested.
- It is also called Test Condition, or Test Possibility.
- Test Scenario is ‘What to be tested’ Test scenario is nothing but test procedure.
- The scenarios are derived from use cases.
- Test Scenario represents a series of actions that are associated together.
- Scenario is thread of operations

Test Cases

- Test cases involve the set of steps, conditions and inputs which can be used while performing the testing tasks.
- Test Case is ‘How to be tested’ Test case consist of set of input values, execution precondition, expected Results and executed post-condition developed to cover certain test Condition.
- Test cases are derived (or written) from test scenario.
- Test Case represents a single (low level) action by the user.
- Test cases are set of input and output given to the System.

Test Script

- A set of sequential instruction that detail how to execute a core business function
- One script is written to explain how to simulate each business scenario
- Written to a level of detail for which someone else (other than the script writer) would be able to easily execute
- Identifies the test condition that is being satisfied for each step, if applicable
- Identified the input/test data that should be entered for each transaction Identifies the expected results for each step, if applicable Should demonstrate how the system can support the HCA warehouse business processes

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

Test plan

- A document describing the scope, approach, resources and schedule of intended test activities Determining the scope and risks, and identifying the objectives of testing.
- Defining the overall approach of testing (the test strategy), including the definition of the test levels and entry and exit criteria.

34. What is priority?

- One can define Priority as a parameter for deciding the order in which one can fix the defect. In this, the defect with a higher priority first needs to get fixed. Priority basically defines the order in which one would resolve any given defect.

35. What is severity?

- Severity is a term that denotes how severely a defect can affect the functionality of the software. Priority is a term that defines how fast we need to fix a defect.

36. Bug categories are…

Security, Database, Functionality (Critical/General), UI

- Functional Bugs.
- Logical Bugs.
- Workflow Bugs.
- Unit Level Bugs.
- System-Level Integration Bugs.
- Out of Bound Bugs.
- Security Bugs.

37. Advantage of Bugzila

- It improves the quality of the product.
- It enhances the communication between the developing team and the testing team.
- It has the capability to adapt to multiple situations.

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

- Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.
- Agile Methods break the product into small incremental builds.

- These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing.
- At the end of the iteration a working product is displayed to the customer and important stakeholders.

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

40. When to used Usablity Testing?

- When should you conduct a usability test?
- Conducting usability tests before any design decisions are made helps us identify the most important user pain points.

41. □ What is the procedure for GUI Testing?

1. Check Screen Validations.

2. Verify All Navigations.
3. Check usability Conditions.
4. Verify Data Integrity.
5. Verify the object states.
6. Verify the date Field and Numeric Field Formats.