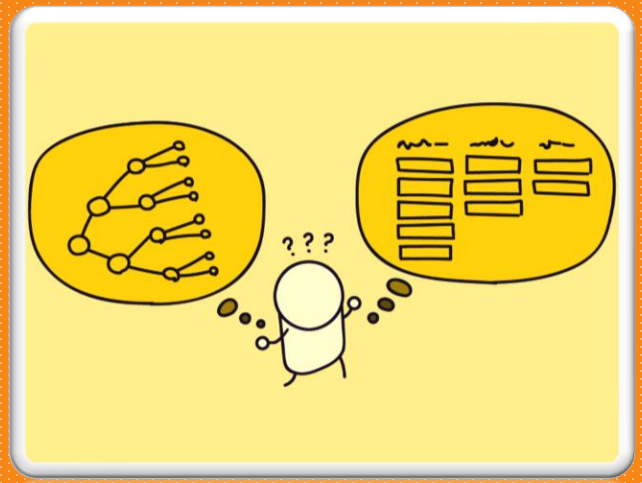


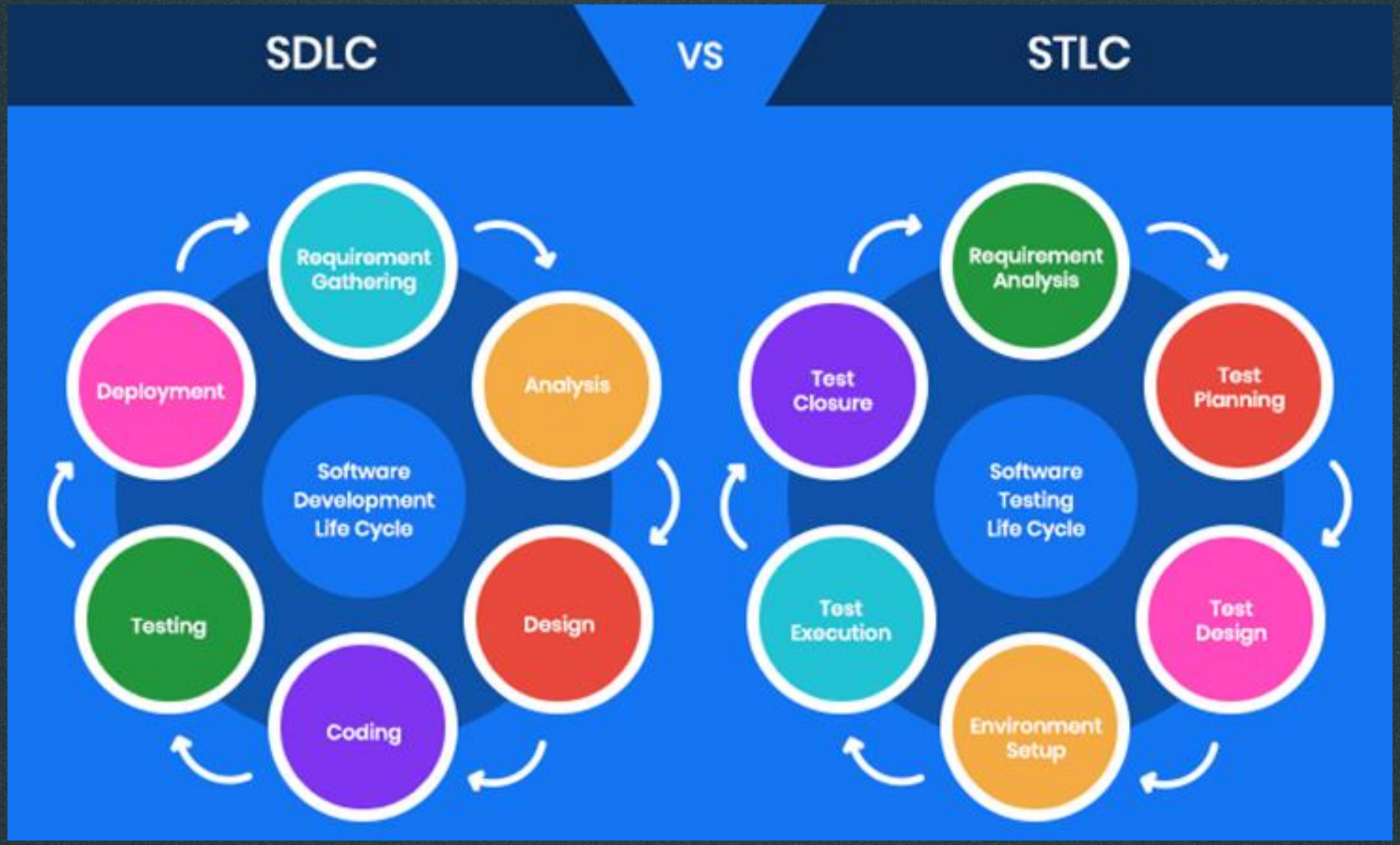
Manual Tester Interview Questions

Question 1

What is the difference between SDLC & STLC?

- **Software Development Life Cycle (SDLC)** is a process used by the software industry to design, develop and test high quality softwares. The **SDLC** aims to produce a high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

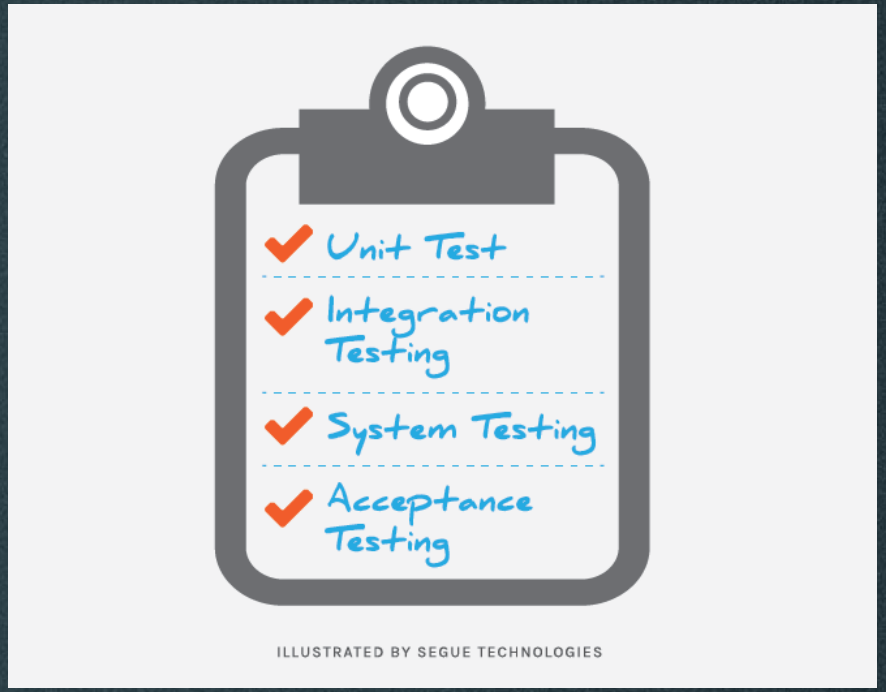
- **STLC** Software Testing Life Cycle is a sequence of different activities performed by the testing team to ensure the quality of the software or the product. **STLC** is an integral part of Software Development Life Cycle (SDLC). But, **STLC** deals only with the testing phases.


SDLC
VS
STLC
Requirement Gathering
Analysis
Design
Coding
Testing
Deployment
Software Development Life Cycle
Requirement Analysis
Test Planning
Test Design
Environment Setup
Test Execution
Test Closure
Software Testing Life Cycle

Question 2

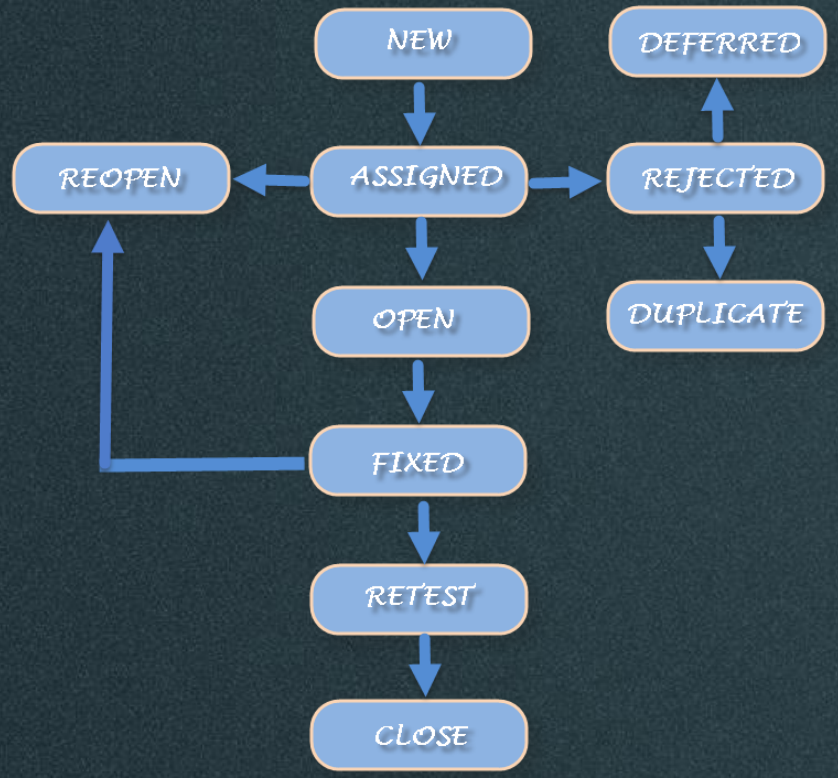
What are the different levels of software testing?

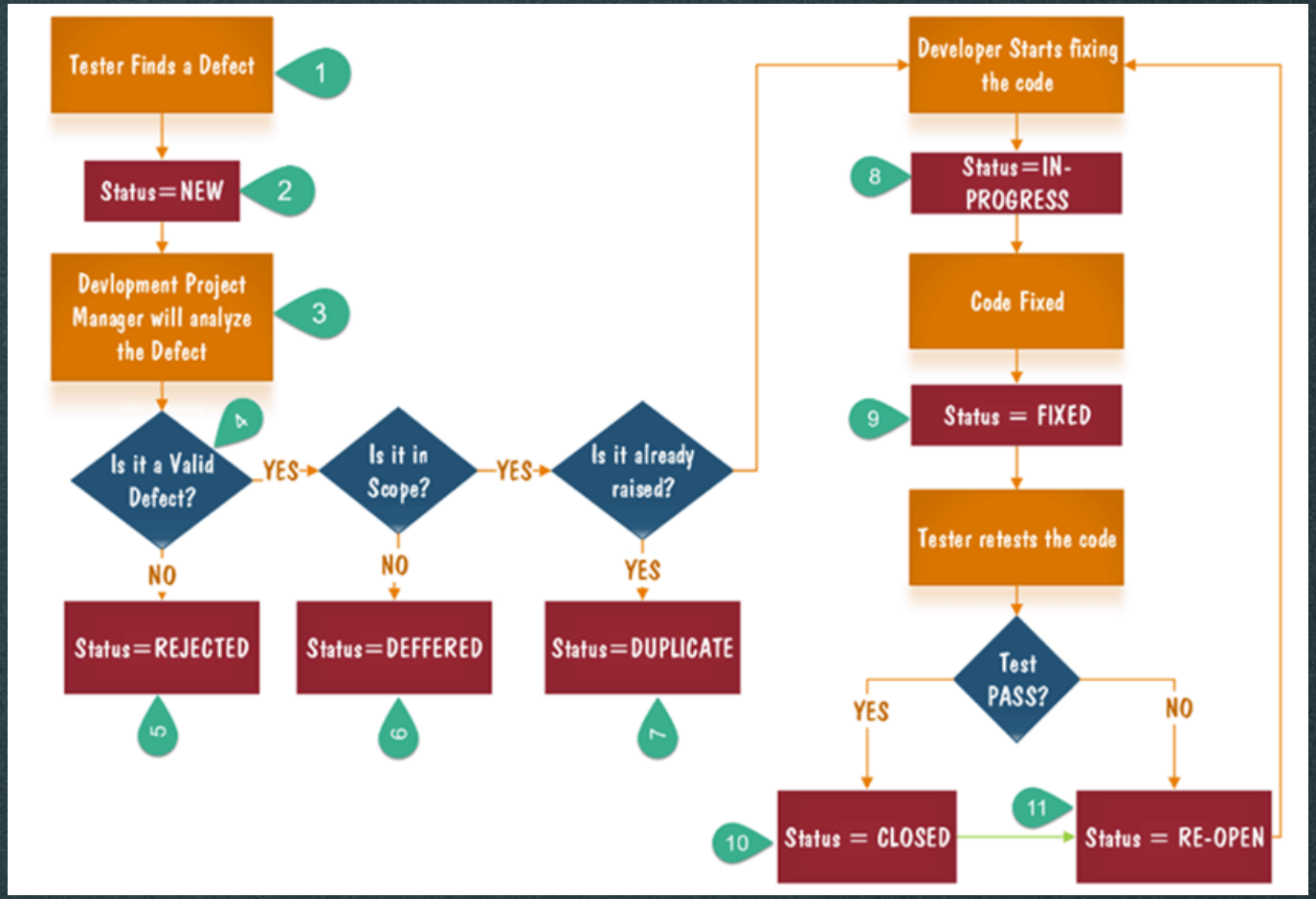
- **Unit Testing:** aims to verify each part of the software by isolating it and then perform tests to demonstrate that each individual component is correct in terms of fulfilling requirements and the desired functionality **[Done by developers]**
- **Integration Testing:** aims to test different parts of the system in combination in order to assess if they work correctly together. By testing the units in groups, any faults in the way they interact together can be identified [Done by developers]
- **System Testing:** all the components of the software are tested as a whole in order to ensure that the overall product meets the requirements specified [Done by Testers]
- **Acceptance Testing:** is the level in the software testing process where a product is given the green light or not. The aim of this type of testing is to evaluate whether the system complies with the end-user requirements and if it is ready for deployment [Done by users]



Question 3

Explain the steps that the defect goes through from discovery to resolution?





Question 4

What is the difference between a test case and a test scenario?

- **Test scenario** is defined as any functionality that can be tested. It is also called *Test Condition* or *Test Possibility*.
- Example: Test the login functionality

- **Test case** is a set of actions executed to verify a particular feature or functionality of your software application. A Test Case contains test steps, test data, precondition, postcondition developed for specific test scenario to verify any requirement.
  Example: Test login with a valid username and a valid password

- The test scenario can be tested with more than one test case

Question 5

What is the difference between functional and non-functional testing?

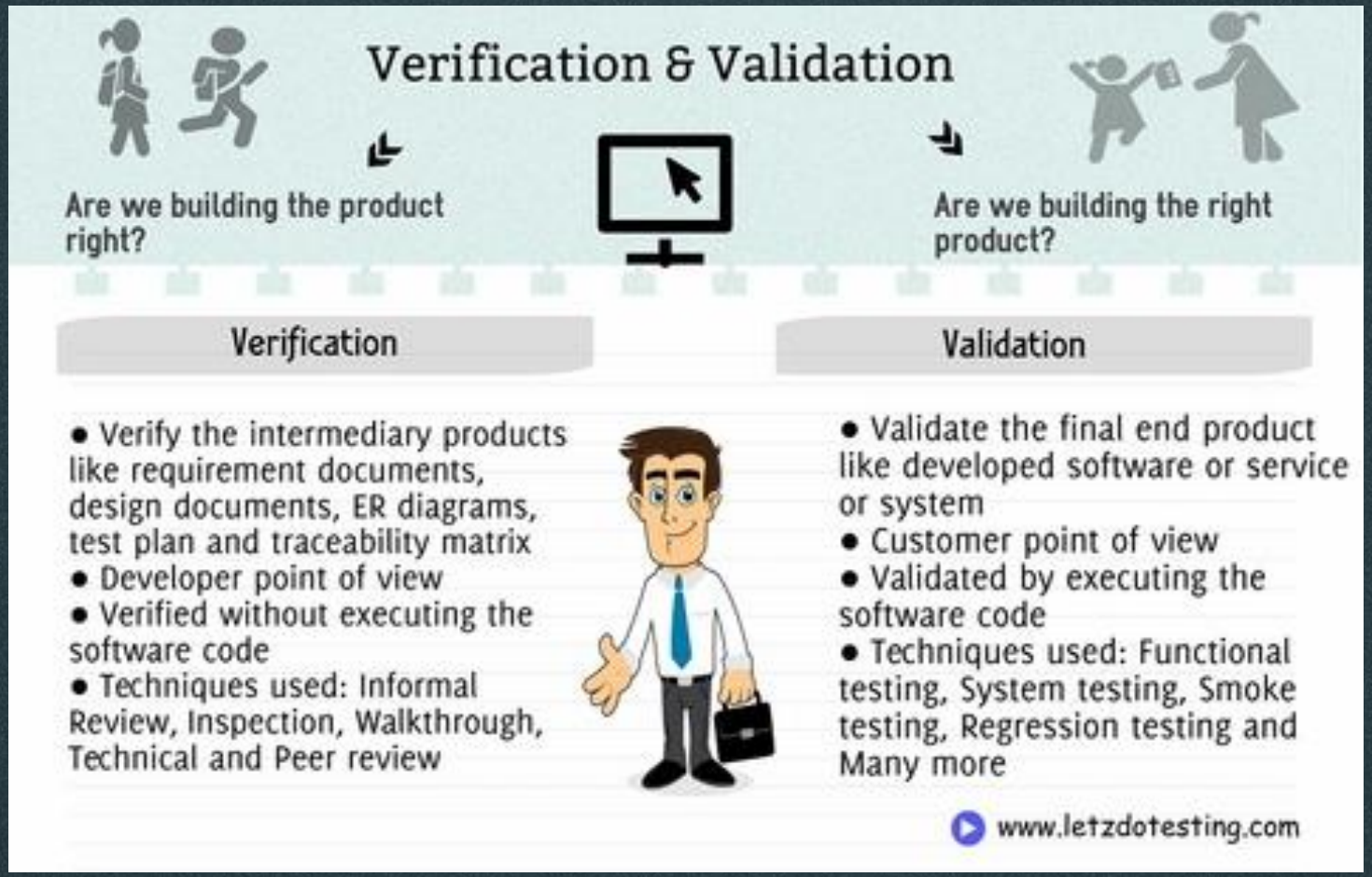
- **Functional testing** is a type of testing which verifies that each **function** of the software application operates in conformance with the requirement specification. It tests what the system does

- **Non-functional testing** is a type of testing to check non-functional aspects (performance, usability, reliability, etc.) of a software application. It tests how well the system performs.

- Example:
  - In functional testing, we test the login functionality, does it work like expected or not?
  - In non-functional testing we may test the performance of the system when 100 users login simultaneously.

Question 6

What is the difference between validation & verification?



Question 7

When should we start testing in our project?

- Software testing should start early in the Software Development Life Cycle. This helps to capture and eliminate defects in the early stages of SDLC i.e requirement gathering and design phases. An early start to testing helps to reduce the number of defects and ultimately the rework cost in the end.

- One of the seven principles of software testing is “Early testing saves time and money”.

Question 8

- If we don’t have clear written user requirements, how can we test the software?

1. Work with whatever little documentation you can get your hands on.
2. Use the older/current version of the application as a reference to test the future release of a software product.
3. Talk to the project team members
4. Use exploratory testing to test the application when it is ready

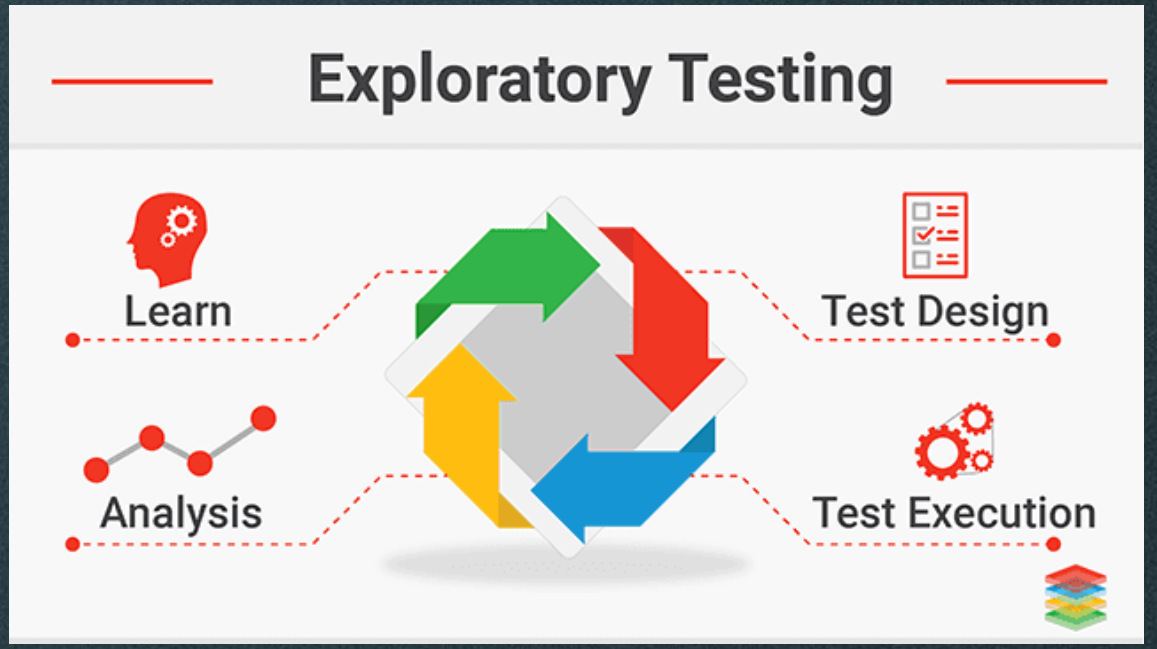
Question 9

- What is exploratory testing, why do we use it?

  - Exploratory testing is an approach to software testing that is concisely described as simultaneous learning, test design and test execution.

  - In exploratory testing, test cases are not created in advance but testers check system on the fly.

- Exploratory testing is used for two reasons:
  1. When we don't have time to design test cases
  2. When there are poor or no requirements

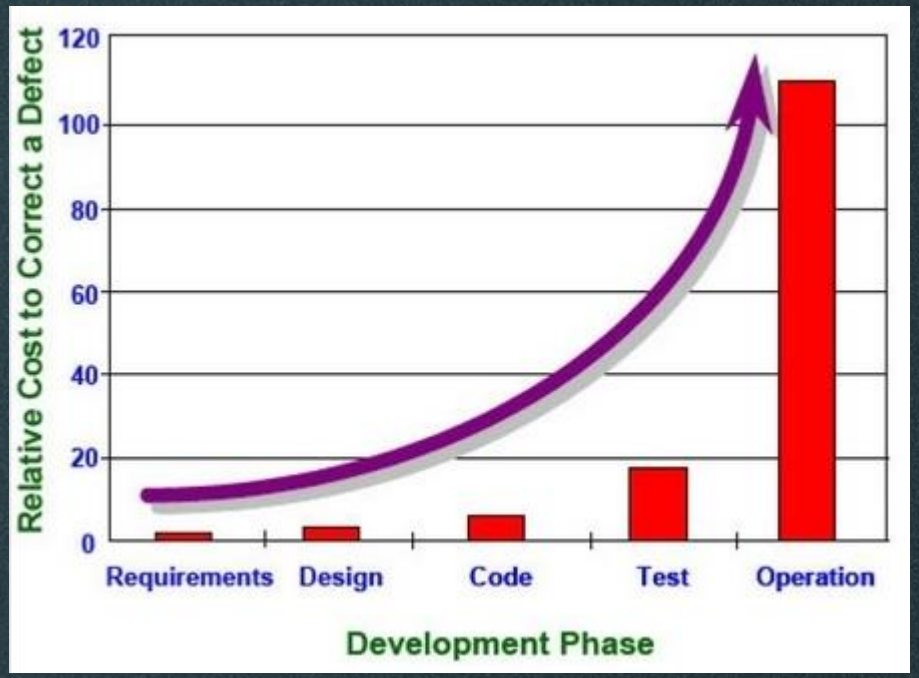


Question 10

A defect which could have been removed during the initial stage is removed in a later stage. How does this affect the cost?

- The cost of defects identified during Software Testing, completely depends on the impact of the defects found. The earlier the defect is found, easier and less costly it is to fix these defects. For instance, if there is a defect found in the project requirement specifications and analysis, then it is relatively cheaper to fix it.

- Similarly, if the defects or failures are found in the design of the software, then the product design is corrected and then re-issued. However, if these defects somehow get missed by testers and if they are identified during the user acceptance phase, then it can be way too expensive to fix such type of errors.



Question 11

-What is change-related testing? And why do we use it?
-What is the difference between confirmation testing & regression testing?

- **Confirmation testing** or **re-testing** When a test fails because of the defect then that defect is reported and a new version of the software is expected that has had the defect fixed. In this case we need to execute the test again to confirm that whether the defect got actually fixed or not.

- **Regression testing** is defined as a type of software testing to confirm that a recent program or code change has not adversely affected existing features.

- Impact analysis is used to know how much regression testing will be required.

Question 12

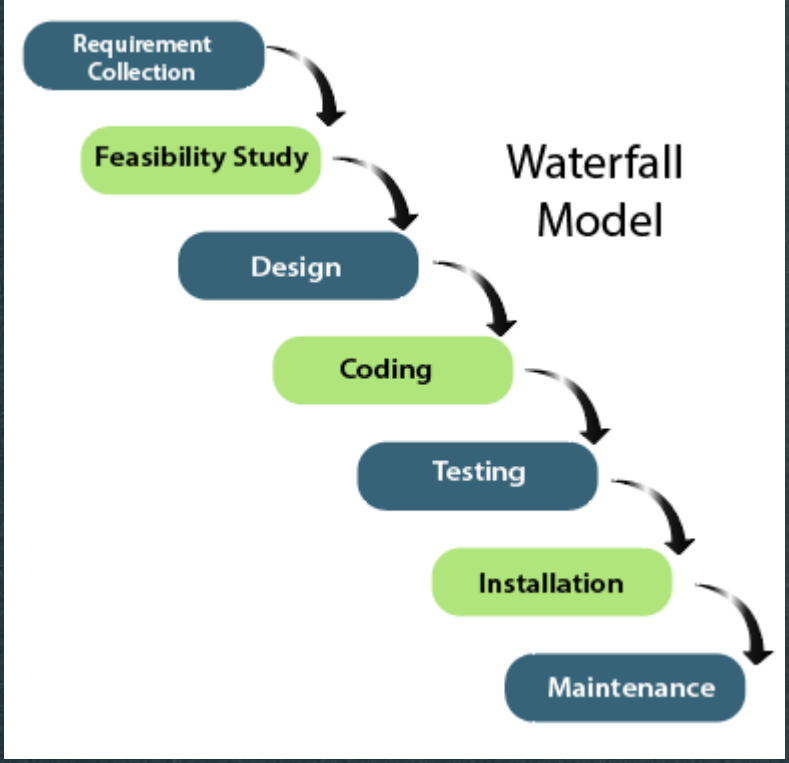
What is the difference between black-box, white-box, and grey-box testing?

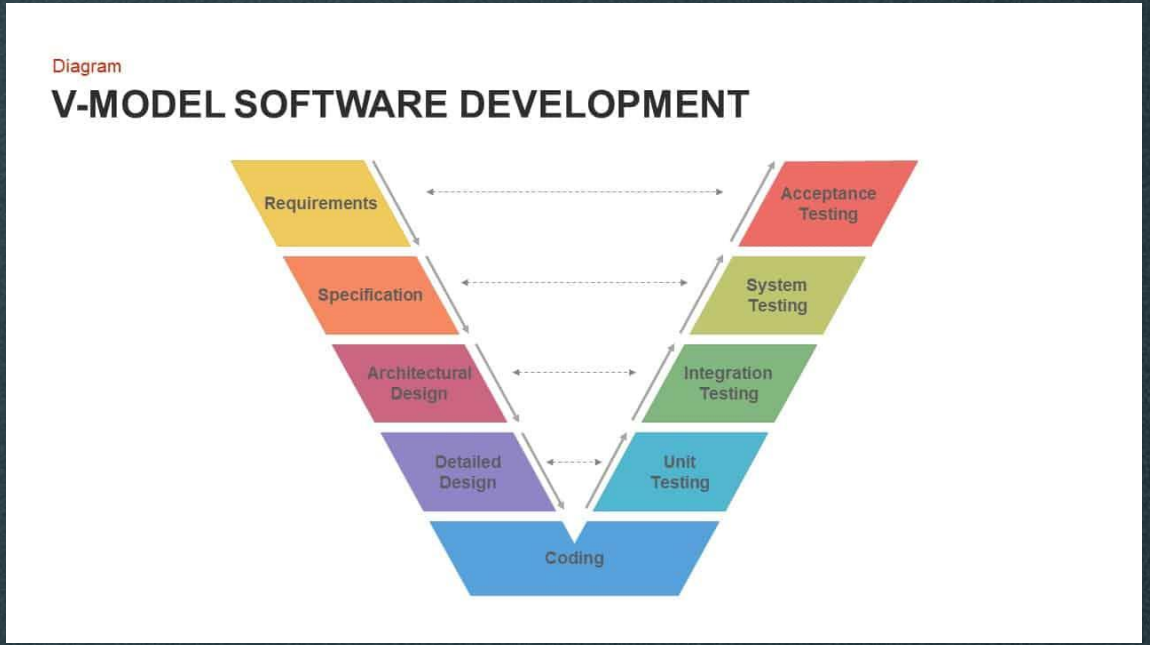
- **Black-box Testing** is a software testing method in which the internal structure/ design/ implementation of the item being tested is not known to the tester .
- **White-box Testing** is a software testing method in which the internal structure/ design/ implementation of the item being tested is known to the tester.
- **Grey-box Testing** is a software testing technique to test a software product or application with partial knowledge of internal structure of the application. The purpose of grey box testing is to search and identify the defects due to improper code structure or improper use of applications.

Question 13

Which test cases are written first, black-box or white box?

- **Black-box test cases are written first because their test basis are the user requirements and the SRS, while the test basis for white-box test cases are detailed design and components specification.**

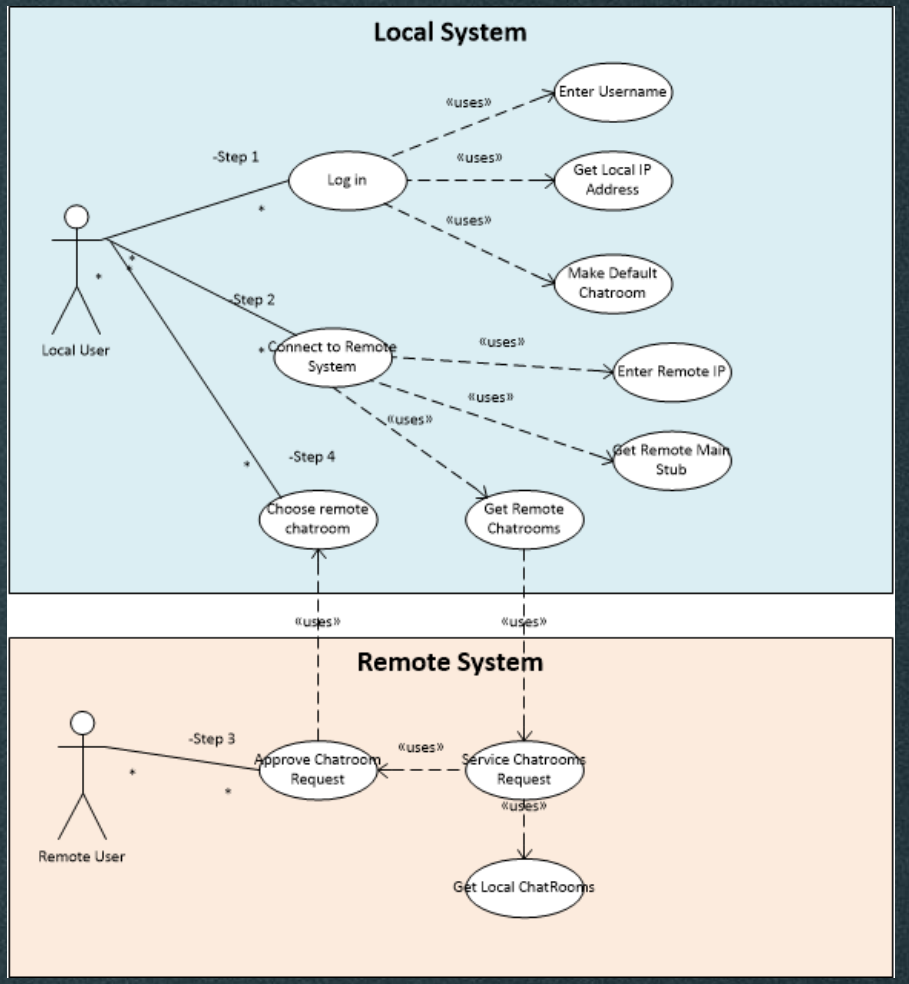




Question 14

What is use-case testing?

- A use case is a description of a particular use of the system by an actor (a user of the system).
- Each use case describes the interactions the actor has with the system in order to achieve a specific task (or, at least, produce something of value to the user).
- Actors are generally people but they may also be other systems.
- **Use case testing** is a technique that helps us identify test cases that exercise the whole system on a transaction by transaction basis from start to finish.



Question 15

What is the difference between equivalence partitioning & boundary-value analysis?

- Equivalence partitioning divides data into partitions (also known as equivalence classes) in such a way that all the members of a given partition are expected to be processed in the same way
- Boundary value analysis (BVA) is an extension of equivalence partitioning, but can only be used when the partition is ordered, consisting of numeric or sequential data. The minimum and maximum values (or first and last values) of a partition are its boundary values
- Behavior at the boundaries of equivalence partitions is more likely to be incorrect than behavior within the partitions.

Question 16

What is requirements traceability matrix?

- Requirements traceability is the ability to connect requirements to other artifacts — such as different types of software tests or bugs. It's used to track requirements — and prove that requirements have been fulfilled.
- Bidirectional traceability is the ability to trace forward (e.g., from requirement to test case) and backward (e.g., from test case to requirement).

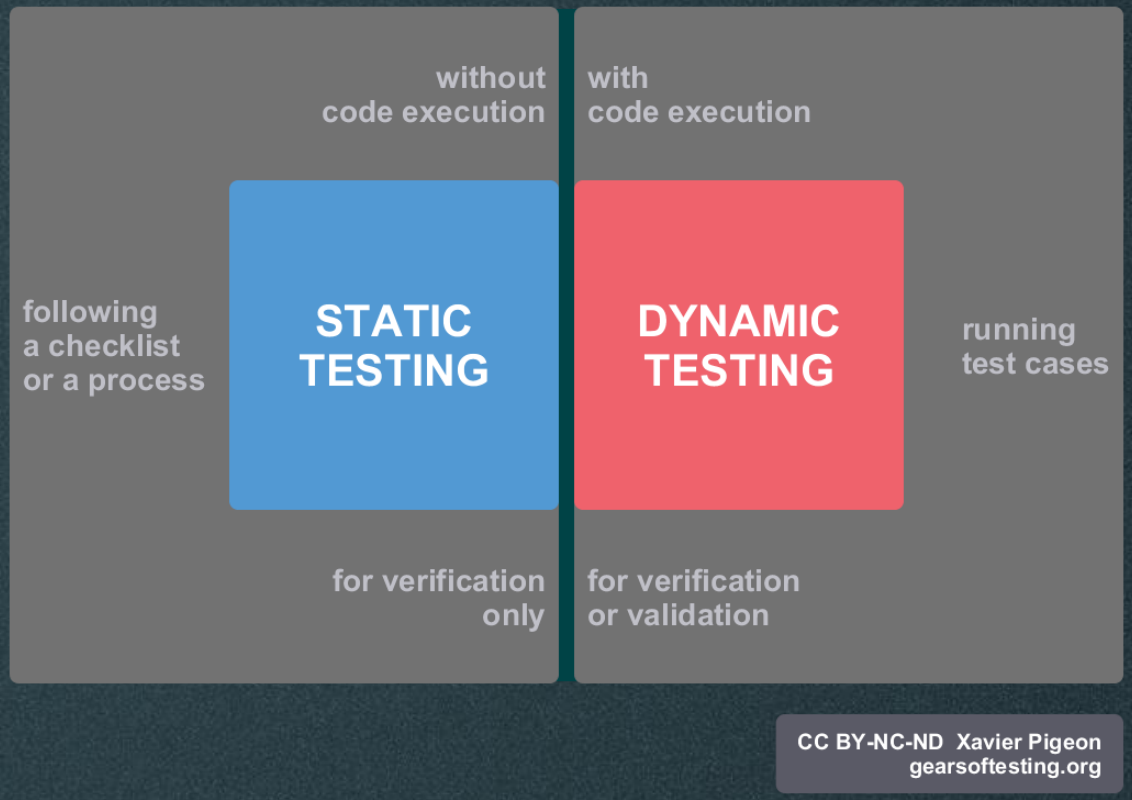
- **Requirement Traceability Matrix (RTM)** is a document that maps and traces user requirement with test cases. It captures all requirements proposed by the client and requirement traceability in a single document. The main purpose of Requirement Traceability Matrix is to validate that all requirements are checked via test cases such that no functionality is unchecked during Software testing.

Req No	Req Desc	Testcase ID	Status
123	Login to the application	TC01,TC02,TC03	TC01-Pass TC02-Pass
345	Ticket Creation	TC04,TC05,TC06, TC07,TC08,TC09 TC010	TC04-Pass TC05-Pass TC06-Pass TC06-Fail TC07-No Run
456	Search Ticket	TC011,TC012, TC013,TC014	TC011-Pass TC012-Fail TC013-Pass TC014-No Run

Question 17

What is the difference between static & dynamic testing?

- **Dynamic testing** involves the execution of the component or system being tested
- **Static testing** does not involve the execution of the component or system being tested. It relies on the manual examination of work products (i.e., reviews) or tool-driven evaluation of the code or other work products (i.e., static analysis)



Question 18

What is the test plan? And what is the information that should be covered in it?

- **Test plan**: A document describing the scope, approach, resources and schedule of intended test activities. It identifies amongst others test items, the features to be tested, the testing tasks, who will do each task, degree of tester independence, the test environment, the test design techniques and entry and exit criteria to be used, and the rationale for their choice, and any risks requiring contingency planning. It is a record of the test planning process.

- As the project and test planning progress, more information becomes available and more detail can be included in the test plan. Test planning is a continuous activity and is performed throughout the product's lifecycle.

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

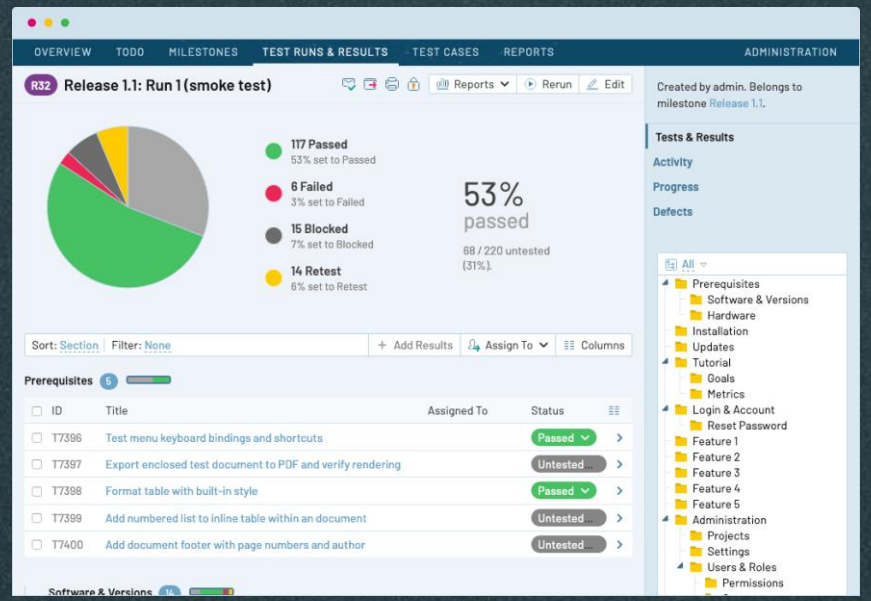
- Phase Test Plan: A test plan that typically addresses one test phase.

Question 19

What is the difference between test progress report and test summary report?

- The purpose of test reporting is to summarize and communicate test activity information, both during and at the end of a test activity (e.g., a test level). The test report prepared during a test activity may be referred to as a test progress report, while a test report prepared at the end of a test activity may be referred to as a test summary report.

- The test report content changes based on the context of the project and the audience of the report



Question 20

Which mistakes do testers tend to do?

1. Failure to communicate
2. Being afraid of asking questions
3. Begin testing before understanding the scope and requirements
4. Writing poor defect reports
5. Missing some requirements while writing test cases
6. Not having any type of planning
7. False positive & False negative

Question 21

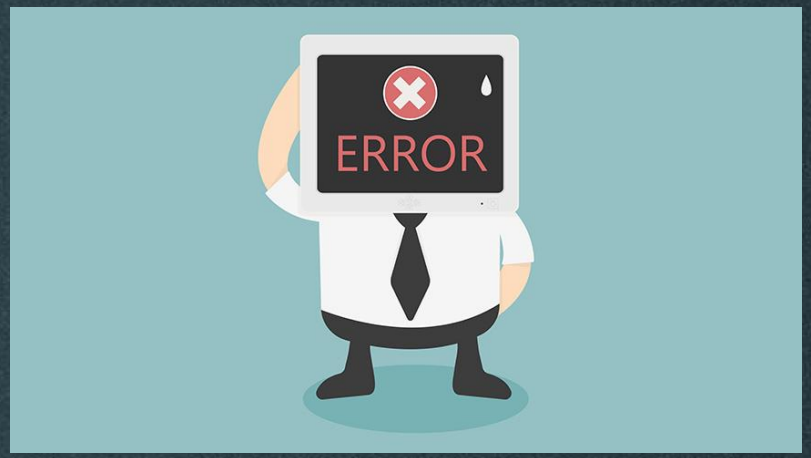
If you reported a defect to a developer and he rejected it, what shall you do?

- Communicate with him [show an example]
- Return to the work products (SRS-Product Backlog)
- Ask the product owner
- Check the test environment [Repeat the steps on different environments]
- Escalate the issue
- Accept that it is not a defect

Question 22

What is the difference between a mistake, defect, and failure?

- A person can make an error (mistake), which can lead to the introduction of a defect (fault or bug) in the software code or in some other related work product.
- An error that leads to the introduction of a defect in one work product can trigger an error that leads to the introduction of a defect in a related work product. For example, a requirements elicitation error can lead to a requirements defect, which then results in a programming error that leads to a defect in the code.
- If a defect in the code is executed, this may cause a failure, but not necessarily in all circumstances. For example, some defects require very specific inputs or preconditions to trigger a failure, which may occur rarely or never.



Question 23

What are the most important components of a defect report?

1. Title
2. Steps to reproduce
3. Expected result
4. Actual result
5. Priority
6. Screenshot or video

Question 24

What is risk-based testing?

Risk Based Testing (RBT) is a software testing type which is based on the probability of risk. It involves assessing the risk based on software complexity, criticality of business, frequency of use, possible areas with defect, etc.
Risk based testing prioritizes testing of features and functions of the software application which are more impactful and likely to have defects.

Risk-based testing steps:
1-Identify the risks
2-Analyze the risks
3-Prioritize the risks
4-Mitigate risks

Question 25

What is the difference between alpha testing & beta testing?

- Alpha and beta testing are typically used by developers of commercial off-the-shelf (COTS) software who want to get feedback from potential or existing users, customers, and/or operators before the software product is put on the market.
- Alpha testing is performed at the developing organization's site, not by the development team, but by potential or existing customers, and/or operators or an independent test team.
- Beta testing is performed by potential or existing customers, and/or operators at their own locations. Beta testing may come after alpha testing, or may occur without any preceding alpha testing having occurred.

Question 26

What are the benefits of test independence?

Benefits of test independence include:

- Independent testers are likely to recognize different kinds of failures compared to developers because of their different backgrounds, technical perspectives, and biases
- An independent tester can verify, challenge, or disprove assumptions made by stakeholders during specification and implementation of the system
- Independent testers of a vendor can report in an upright and objective manner about the system under test without (political) pressure of the company that hired them

Question 27

What are the potential drawbacks of test independence?

Potential drawbacks of test independence include:

- Isolation from the development team, may lead to a lack of collaboration, delays in providing feedback to the development team, or an adversarial relationship with the development team
- Developers may lose a sense of responsibility for quality
- Independent testers may be seen as a bottleneck
- Independent testers may lack some important information (e.g., about the test object)

Question 28

What is the difference between test techniques and testing tools?

- The purpose of a test technique, is to help in identifying test conditions, test cases, and test data.

- Examples:
  - Black-box Techniques (Equivalence Partitioning)
  - White-box Techniques (Statement Coverage)
  - Experience based techniques (Error guessing)

- Tools from a software testing context can be defined as a product that supports one or more test activities right from planning, requirements, creating a build, test execution, defect logging and test analysis.

- Examples:
  - Test Management Tools (Google Sheets-Trello-Jira)
  - Test Automation Tools (Selenium Webdriver – Cypress – Robot Framework)
  - Performance Testing Tools (Jmeter – HP Loadrunner)
  - API Testing Tools (Postman – Soap UI – Rest Assured)

Question 29

What is random/monkey testing? When do we use it?

Random testing is often known as monkey testing. In such type of testing data is generated randomly often using a tool or automated mechanism. With this randomly generated input, the system is tested, and results are analyzed accordingly. These testing are less reliable; hence it is normally used by the beginners and to see whether the system will hold up under adverse effects.

Question 30

What is the difference between negative and positive testing?

A negative test is when you put in an invalid input and receives errors. While positive testing is when you put in a valid input and expect some action to be completed in accordance with the specification.

Question 31

What is Decision table testing? When should we use it?

Decision table testing is used for testing systems for which the specification takes the form of rules or cause-effect combinations. In a decision table, the inputs are listed in a column, with the outputs in the same column but below the inputs. The remainder of the table explores combinations of inputs to define the outputs produced.

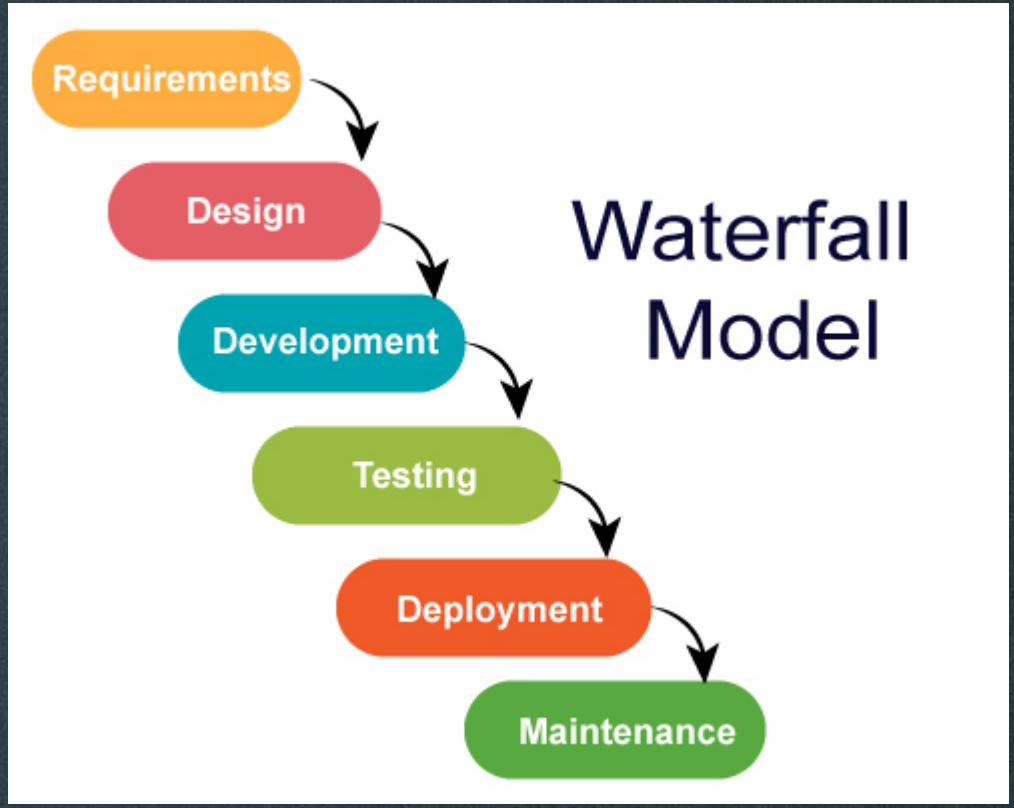
(Decision Table)

	Rules							
Conditions	1	2	3	4	5	6	7	8
Preferred Customer	Y	Y	Y	Y	N	N	N	N
Order $1,000 or more	Y	Y	N	N	Y	Y	N	N
Used our charge card	Y	N	Y	N	Y	N	Y	N
Actions								
5% discount	X	X						
Additional 5% discount	X							
$25 bonus coupon			X	X				
$5 bonus coupon					X	X	X	X

Question 32

What is the waterfall model? How is testing performed in it?

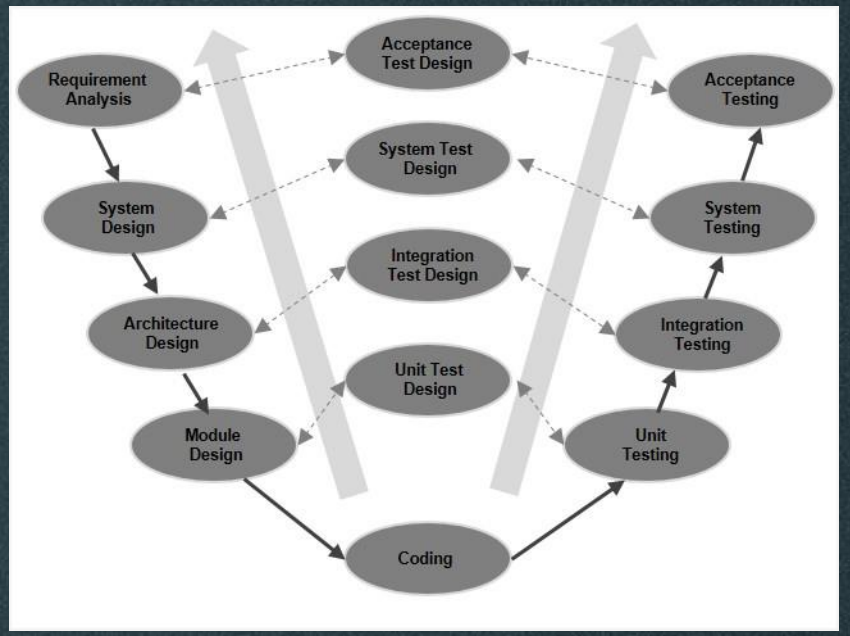
- The Waterfall model is the earliest SDLC approach that was used for software development.
- The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.



Question 33

What is the v-model? How is testing performed in it?

- The V-model is an SDLC model where execution of processes happens in a sequential manner in a V-shape. It is also known as **Verification and Validation model**.
- The V-Model is an extension of the waterfall model and is based on the association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle, there is a directly associated testing phase. This is a highly-disciplined model and the next phase starts only after completion of the previous phase.



Question 34

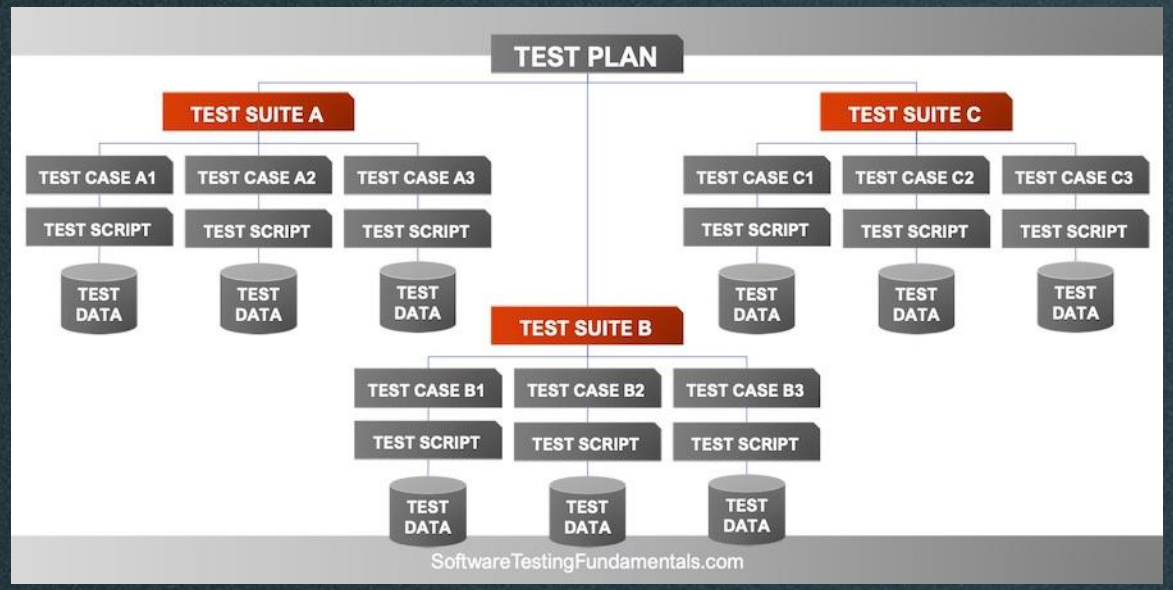
What are the best practices for writing test cases?

- Write test cases with end-users perspective
- Write test steps in a simple way that anyone can follow them easily
- Make the test cases reusable
- Set the priority
- Provide a test case description, test data, expected result, precondition, postcondition.
- Write invalid test cases along with valid test cases
- Follow proper naming conventions
- Review the test cases regularly and update them if necessary.

Question 35

What is the test suite?

- Test suite is a container that has a set of tests which helps testers in executing and reporting the test execution status.
- A Test case can be added to multiple test suites.
- In some tools “like testlink”, test suites are created before creating test cases



Question 36

What is the test environment?

A testing environment is a setup of software and hardware for the testing teams to execute test cases. In other words, it supports test execution with hardware, software and network configured.

Example:
A typical Environmental Configuration for a web-based application is given below:

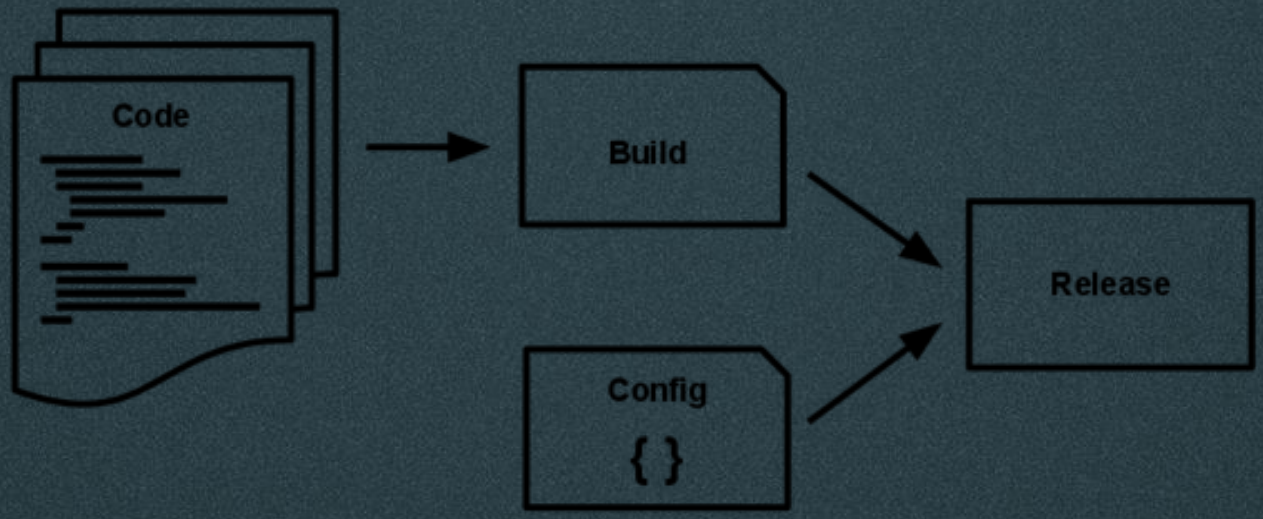
Web server – ITS/Apache
Database – MS SQL
OS – Windows / Linux
Browser – IE/Firefox
Java version – Version 6

Question 37

What is the difference between build and release?

A build is an executable file provided by the developers to the testing team for testing the application. It undergoes various iterations of fixing and testing until the application works as expected. Once the application becomes stable and ready for the end-users, it's released in the market.

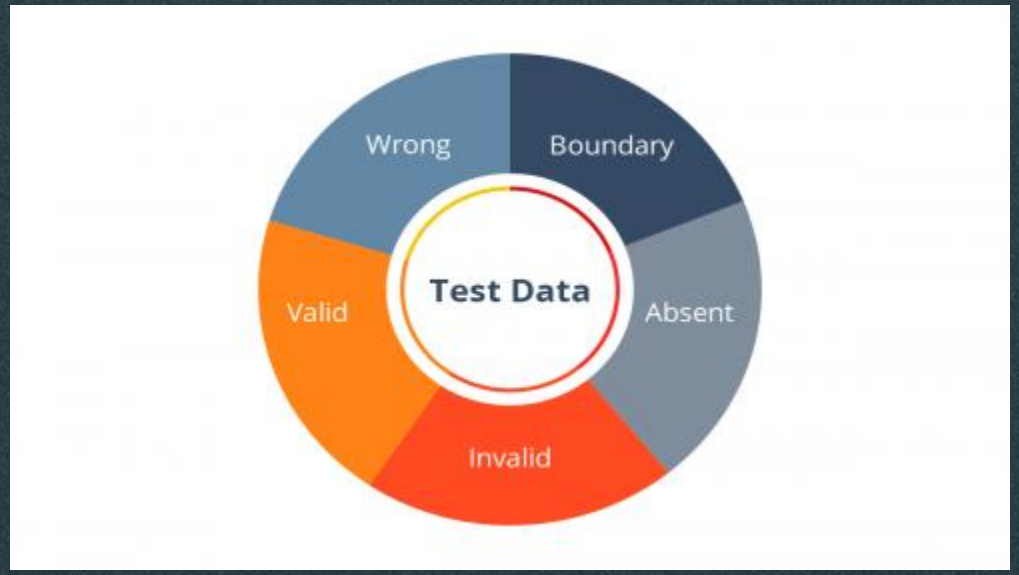
Whereas, a release is an installable software provided to the end-users after it gets certified by the testing team. During the release of any software to the client, release notes are attached to it that includes a number of defects still open, covered user stories, change-requirements, and version of the release.



Question 38

What is the test data?

Test data is the data that is used by the testers to run the test cases. Whilst running the test cases, testers need to enter some input data. To do so, testers prepare test data. It can be prepared manually and also by using tools.
For example, To test a basic login functionality having a user id, password fields. We need to enter some data in the user id and password fields. So we need to collect some test data.



Question 39

What is the difference between quality control and quality assurance?

- Quality assurance is typically focused on adherence to proper processes, in order to provide confidence that the appropriate levels of quality will be achieved.
- When processes are carried out properly, the work products created by those processes are generally of higher quality, which contributes to defect prevention.
- In addition, the use of root cause analysis to detect and remove the causes of defects, along with the proper application of the findings of retrospective meetings to improve processes, are important for effective quality assurance.

- Quality control involves various activities, including test activities, that support the achievement of appropriate levels of quality.

Question 40

What is a show stopper? How should we handle it?

- A critical bug is a show stopper which means a large piece of functionality or major system component is completely broken and there is no workaround to move further.
- For example, Due to a bug in one module, we cannot test the other modules because that blocker bug has blocked other modules. Bugs which affects the customers business are considered as critical.

- **Example:**
- 1. “Sign In” button is not working on Gmail App and Gmail users are blocked to login to their accounts.
  2. An error message pops up when a customer clicks on transfer money button in a Banking website.

Question 41

-What are the integration approaches?
-What is the difference between big-bang integration, top-down integration, and bottom-up integration?

Big Bang Approach: Combining all the modules once and verifying the functionality after completion of individual module testing.

Top-down and bottom up are carried out by using dummy modules known as Stubs and Drivers. These Stubs and Drivers are used to stand-in for missing components to simulate data communication between modules.

Top-Down Approach: Testing takes place from top to bottom. High-level modules are tested first and then low-level modules and finally integrating the low-level modules to a high level to ensure the system is working as intended. Stubs are used as a temporary module if a module is not ready for integration testing.

Bottom-Up Approach: It is a reciprocate of the Top-Down Approach. Testing takes place from bottom to up. Lowest level modules are tested first and then high-level modules and finally integrating the high-level modules to a low level to ensure the system is working as intended. Drivers are used as a temporary module for integration testing.

Question 42

What is the difference between smoke testing & sanity testing?

Smoke Testing is done to make sure if the build we received from the development team is testable or not. It is also called as “Day 0” check. It is done at the “build level”. It helps not to waste the testing time to simply testing the whole application when the key features don’t work or the key bugs have not been fixed yet.

Sanity Testing is done during the release phase to check for the main functionalities of the application without going deeper. It is also called as a subset of Regression testing. It is done at the “release level”. At times due to release time constraints rigorous regression testing can’t be done to the build, sanity testing does that part by checking main functionalities.

Question 43

What is ad-hoc testing?

Ad-hoc testing is quite opposite to the formal testing. It is an informal testing type. In Ad-hoc testing, testers randomly test the application without following any documents and test design techniques. This testing is primarily performed if the knowledge of testers in the application under test is very high. Testers randomly test the application without any test cases or any business requirement document.

Question 44

What is the difference between globalization & localization testing?

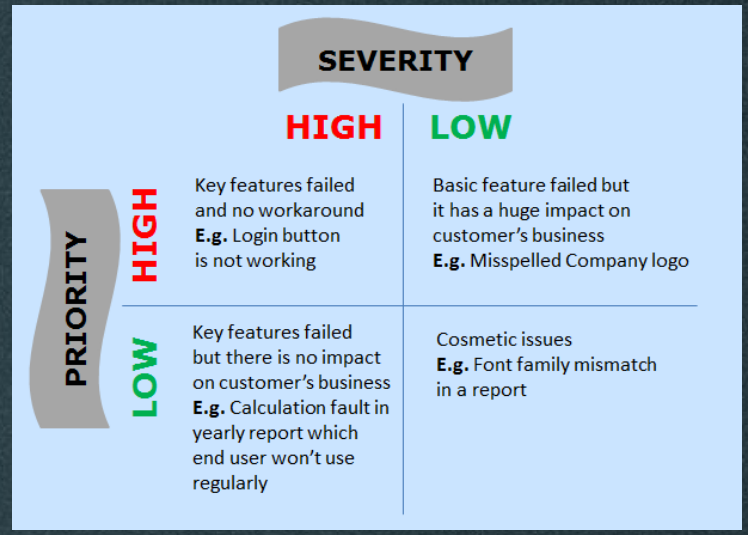
Localization testing(I10n) is the software testing process for checking the localized version of a product for that particular culture or locale settings. The areas affected by localization testing are UI and content.

Globalization/Internationalization Testing(i18n) is a software testing method used to ensure that the software application can function in any culture or locale (language, territory or code page) by testing the software functionalities using each type of international input possible. The purpose of Globalization testing is to ensure that software can be used internationally or worldwide. It is also called Internationalization Testing.

Question 45

What is the difference between defect priority & severity, give examples?

- Defect priority is the order in which a defect should be fixed. Higher the priority the sooner the defect should be resolved.
- Defect severity is a degree of impact a bug or a Defect has on the software application under test. A higher effect of bug/defect on system functionality will lead to a higher severity level.



- Order of priority:
  1. High priority – High severity
  2. High priority – Low severity
  3. Low priority – High severity
  4. Low priority – Low severity

- **High Priority & High Severity:** Submit button is not working on a login page and customers are unable to login to the application
- **Low Priority & High Severity:** Crash in some functionality which is going to deliver after couple of releases
- **High Priority & Low Severity:** Spelling mistake of a company name on the homepage
- **Low Priority & Low Severity:** FAQ page takes a long time to load


See you in
next section