K.Sravanthi
22NU1A0560
CSE-A

4. Summarize the different phases involved in the software testing life cycle?

A The STLC life cycle typically has six main

1. Requirement Analysis:

→ In this first phase, the QA team studies the software requirements specification (SRS) to understand what needs to be tested & identify testable requirements.

→ Testers clarify doubts with stakeholders & identify missing, ambiguous or risky requirements from a testing point of view

→ Typical outputs are a list of testable requirements high level test scenarios and an initial RTM.

2. Test planning:

→ Based on analyzed requirements test lead defines the overall test, strategy, scope, objectives, types of testing and approach

→ effort, schedule, budget, tools and required resources are estimated

→ The main deliverable is the test plan document, which is reviewed and approved before moving ahead.

Testcase Development:-

→ In this phase, detailed test cases and test scripts are designed for each testable requirements including positive and negative scenarios.

Test data is prepared or identified to execute these test cases effectively in the chosen environments.

Test environment setup:

Required h/w, s/w, n/w configuration, databases and test is installed and configured to create the test environment

After this phase is done in parallel with test case development

5. Test Execution:

→ Testers run the prepared test cases in configured environments, Compare actual results with expected result.

→ Any failures are logged as defects/bugs with details such as steps to reduce severity & assign it to developers for fixing.

6. Test cycle closure:

→ The team evaluates whether exit criteria are met.

→ Testing activities and results are into Summary

5. Interpret the concepts of verification & validation in software testing?

A Verification and validation are two distinct yet complementary process in s/w testing that ensures software quality through different approches and focus area.

* Verification:

Verification is a static process that take place early in SDLC. It involves reviewing & evaluating documents such as requirements design specification and code without executing the software. The goal of verification is to confirm that the software is being developed according to the specified requirements and design standards. It answers the question "Are we building the product right? Verification activities include, inspection, walk throughs.

Validation:-

Validation is a dynamic process conducted after the s/w is develop involving the actual execution of s/w to ensure it is function. It answers

the question "are we building the right product"
It focuses on verifying that software meets user needs & requirements ensure usability and effectiveness validation on involves various testing techniques such as functional testing, System testing.

Verification	Validation
1. Occurs before implementation.	1. occurs after coding
2. Verification is static	2. validation is dynamic
3. Ensures product built is right	3. Ensures product meets Expectations.
4. catches design & requirement errors	4. functionality & Usability issues.
5. Are we building the Product right?	5. Are we building the right product?

6. Describe the role of activities and requirements in both high level and low level design stages?

A High level Design (HLD):

→ It outlines the major building blocks, components modules & their interactions to form an overall system architecture.

→ It decide on programming languages, framework. databases and infrastructure suitable for requirements

→ plans how modules interact and design major data flows.

→ Anticipate future growth & incorporate security principles.

* Requirements in HLD:
→ SRS which contains all functional & non-functional requirements
→ SRS focuses on what & how it will be implemented
→ The HLD produces a blueprint and allocate requirements to specific system components or modules.

* Low level Design (LLD):
- Specifies the internal logic algorithms and workflows for each module defined during HLD.
- Define detailed data structures, API endpoints, inputs/outputs and error handlings for each module.
- Document technical details, namings conventions & precise logic required for programming.

* Requirements in LLD:
→ The reviewed & approved HLD document is input
→ The required are detailed processing on how each module fulfills its assigned functionality
→ LLD produces program specifications, flowcharts, pseudocode, data definitions & unit test plans.

* Key differences & Relationships:

a) HLD answers what and where (system structure, responsibility, allocation) LLD answers how (detailed implementation and logic)

b) Successful translation of requirements across HLD & LLD phases ensures full coverage, maintainability, traceability through development.

Both stages are essential, HLD provides stability & vision while LLD delivers precision & guidance for building & testing each component.