MANUAL TESTING:-

>Quality Center:

-This is a tool

-Quality Center defines, manages and tracks functional, performance, and security test scripts in one place.

-It assigns business risk to requirements, and calculates where to apply testing resources.

-QA teams can emulate business processes, run tests unattended with QTP and execute manual and automated tests.

>WaterFall: The next phase starts after the earlier phase has fully completed.

>Agile-in the form of Sprints…each sprint spans upto around 2 weeks.

Scrum Master, Scrum Meeting / Standup meeting

The no. of requirements that can fit in a 2 weeks span.

The input for the Testing team is they have go through the User stories created by BA’s.

>Requirement Traceability Matrix(RTM):

-This is a table format which will have the mapping between the requirements and Testcases.

-RTM is been created by your Testing engineer/Test lead to have a track and mapping between all the requirements and the testcases – status.

R1 TC1 link to the testcase doc -> XLS sheet.

\*Test Strategy: Test Approach – defines how the testing should be carried out!

\*Test plan: It is a detailed doc which describes the test strategy.

-Schedule, estimates, Deliverables and test data etc, includes in Test plan.

Test Scenario: It is groups of testcases.

Given a requirement/usecase there can be many possibilities -> Test scenario.

When the test scenario is broken up - > Test case.

Test case-> emphasizes one particular flow/ has the steps to execute one single flow. This is written in a xls file.

Test Script: same as test case but created programmatically.

STLC:

Software Testing Life cycle:

Requirement is confirmed -> Test planning -> RTM -> Test Strategy -> Test plan -> Test Scenarios -> Test cases / Test scripts -> Test Execution -> Go / no Go (defect fixing by developers)

>Bug and a defect:

-Bug is a mistake in the program -> unit / dev-Integration phases.

-Defect: this is a deviation in the requirement. The program is not behaving as per the requirement.

QA -> Quality Analyst-> part of the testing team.

Scope : The scope of a test defines what areas of a customer's product are supposed to get tested, what functionalities to focus on, what bug types the customer is interested in, and what areas or features should not be tested by any means.

Types of testing:

Unit testing

Dev/Integration testing

UAT – User Acceptance Testing

Pre – Production testing

Production testing

Smoke testing: If at all, all the basic functionalities are working fine.

Sanity testing: whenever a minor change is made, we check the entire application if its still working fine.

Functional testing: To see if all the functionalities working fine or not.

Performance testing: How fast the page is rendered. Not more then 3 mins (as specified by client).

Regression testing: Any new code has not impacted the existing functionality.

BlackBox testing: You don’t care on the internal code, you test the functionalities of the applications.

WhiteBox testing/open box testing/Code based testing/Glass box testing: The tester has the visibility on the flow of the data, structure of the code etc is available to the tester.

STLC phases:

Acceptance Criteria:

Expected behavior of the system as per the requirement.

Epic -> piece of user story

Entry and Exit criterias:

-In case of software testing, entry criteria defines the conditions to be satisfied in order for the testing to begin.

-Exit criteria define the conditions that have to be satisfied in order to stop the testing.

-Both of these will be defined in the test plan

>Test Management : Test Management is a process of managing the testing activities in order to ensure high quality and high-end testing of the software application. The method consists of organizing, controlling, ensuring traceability and visibility of the testing process in order to deliver the high quality software application.

-It ensures that the software testing process runs as expected.

>Risk Analysis:Risk analysis in software testing is an approach to software testing where software risk is analyzed and measured.

>QA/QE:

-QA: Quality Analyst – one who ensures/maintains the quality of a product by executing on CodeScience's quality procedures.

-QE: Quality Engineer – one who automates quality procedures to minimize manual testing efforts.

-He puts in the stringent rules to what tools to be used and policing on the quality regulations so that the quality is assured.

>Defects life cycle:

-defect life cycle is a process in which defect goes through different stages in its entire life.

-This lifecycle starts as soon as a bug is reported by the tester and ends when a tester ensures that the issue is fixed and won't occur again.

-Using a tool u need to manage the testcases and defects.

Defects life cycle:

New -> assign-> in progress-> Test-> re-test(to QA)->Closed / re-opened

>Test pyramid:

-Some model or a structure that should be followed to have a quality product which is made faster and more efficient.

-Testing Pyramid is a framework that can help both developers and QAs create high-quality software. It reduces the time required for developers to identify if a change they introduced breaks the code.

-It can also be helpful in building a more reliable test suite.

>V & V : Verification and Validation testing

-This means that for every single phase in the development cycle, there is a directly associated testing phase.

User stories.

FRD: Functional Requirement Document

BRD: Business Requirement Document

HLD: High level Design Document

LLD: Low Level Design Document

>Requirement testing:

-It is based on the requirements provided by the client.

-All the test cases, test scenarios, test data are inclined from requirement.

>What are the types of requirements in software testing?

-A software requirement can be of 3 types:

\*Functional requirements.

\*Non-functional requirements.

Types of requirement testing:

\*Explicit Requirements: The Things You Wrote Down.

-Explicit requirements are most commonly found in documents communicated by stakeholders to the development team.

\*Implicit Requirements: The Things Your Customers Will Expect.

-These are all the things that users are going to expect that were not captured explicitly.

\*Latent Requirements: Things That Will Delight Your Customers

-Latent requirements represent behaviour that users do not expect based on their previous experiences but which will make them like the software more.

>Importance of static testing:

-Static testing should be completed before the implementation of project.

-Detecting the maximum number of errors in the requirement phase will save the time and cost of organisation