TEST STRATEGY AND TEST PLANNING

Test strategy is a high level plan consisting of principles that guide the overall software testing process. It is done at a organisation level.

It provides a structured approach to the entire QA team, guiding them towards achieving testing objectives in the most efficient way.

Q) Why Test Strategy?

1. To make sure that all purposes are covered entirely and understood by all stakeholders
2. To support quality assurance with respect to planning of resources, language, test, roles and responsibilities etc.

Test Strategy document

It is a well described document in software testing which clearly defines the exact software testing approach and testing objectives of the software application.

Components of Test Strategy

- Scope & Overview
- Testing methodology

- Testing environment
- Testing tools
- Release
- Risk analysis
- Review and approvals

Test Plan

Test Plan derives future activities of a test Engineer/future scope of project.

It is prepared by Test Lead and approved by Test Manager.

We have 15 attributes in Test Plan. Lets discuss:

- **Objective**: It speaks about the aim of writing the test plan. We can mention which model is used and what process is used.

- **Scope**: Here we mention features which need to be tested and features which need not be tested.

- **Test methodology**: Based on type of application, we decide what type of testing needs to be done.
  - ❖ Web based application
  - ❖ Client Server application

- ❖ Stand-alone application
- ❖ **Test approach**: It explains how we test the application. Several approaches are:
- ❖ Writing Test cases
- ❖ Writing Test scenarios
- ❖ Writing Test cases and Test scenarios
- ❖ Writing the flowchart

- **Assumption**: While writing Test plan, team will assume few aspects:
  - ❖ Assumption from resource point of view
  - ❖ Assumption from technology point of view
  - ❖ Assumption from development team point of view
  - ❖ Assumption from Knowledge Transfer point of view
  - ❖ Assumption from supporting documents point of view

- **Risk**: When assumption fails, we will be in Risk, to overcome it, we have a Backup plan.

- **Backup plan/Mitigation Plan**: If assumption fails, will lead to 100% risk, by having a Backup plan we can reduce risk to 20%. Example: Let's say 3 TE's are working on a project, due to some reasons, 1 TE quits job suddenly. Now the team is at risk as we are supposed to finish the project in a given deadline. To overcome this risk, we will initially assign a secondary TE, so that they can act as a Backup and ensure software is tested in a given timeline before it is released to the customer.

- **Roles & Responsibilities of TE**:

- Understand customer requirement
- Write Test scenarios & Test cases
- Conduct Brainstorming meeting for better test case coverage & get test cases reviewed
- Always do optimised testing
- Conduct Smoke, Functional, Integration, System, ad hoc testing etc
- Perform Test case execution
- Log the defect to the defect tracking tool
- Give Knowledge Transfer to other team members
- **Scheduling**: In this section, we mention the start and end date for each and every testing activity.

- **Defect Tracking**: As soon as TE finds any defect, he will login to the Defect tracking tool and log the defect, generate a defect report, and a unique id will be created.

TE will communicate this defect to the developer and the developer will fix the defect and give it back to TE. TE will retest to check if the defect is really fixed or not. If the defect is fixed, TE will close the defect. This process is called Defect Tracking.

- **Test environment**: It is an environment configured for testing, where in TE will test the application by executing the test cases. Test environment consists of hardware, software, database, OS, CPU etc details.

- **Entry & Exit criteria**: These are the set of conditions that should be met in order to start and end the project. It is also the key attribute of Test plan.

Entry criteria:

- Coding should
- WBT should be done
- Software should be installed
- Smoke Testing should be done
- Functional Test scenarios & Test cases should be ready
- Resource should be present

Exit Criteria:

- Based on number of Test cases executed, test case pass% should be 85%, test case fail % should be 15%

Integration Testing:

Entry criteria:

- It should match exit criteria of Functional Testing
- Functional Test scenarios & Test cases should be ready
- Resource should be present

Exit Criteria:

- Based on number of Test cases executed, test case pass % should be 95%

System Testing:

Entry criteria:

- It should match exit criteria of Integration Testing
- Integration Test scenarios & Test cases should be ready
- Resource should be present

Exit Criteria:

- Based on number of Test cases executed, test case pass% should be 99%
- **Test Automation**: Here we mention which automation tool we use on a project, which features to be tested and which need not be tested, which automation framework we use.

- **Deliverables**: These are the outcomes of the Testing team. This section contains what we are going to give to customers by the end of the project. The document includes – Test plan document, Test scenario & Test case document, Traceability matrix document, Test execution report document, Defect report document, Release notes, Graphs & metrics

Q) What do you mean by Release notes?

A) These are the set of documents given to the customer along with software which is signed off by the Test manager. Release notes consist of – List of Pending and open defects, List of defects fixed in current release found in previous release, platform on which software is tested, platform on which software is not tested, list of features added, deleted, modified, procedure to install the software, version of software.

- **Templates**: Here, we mention all the empty templates which will be used in future by TE's.

Templates include – Test case template, Test case review template, Traceability matrix template, Test case execution template, Defect report template, Test plan template.