**DAY 2: Introduction to Software Testing**

**What is Testing: -**

**When we say testing, we think of it as:**

* A process of verifying and validating a software, website or application.
* Ensuring product is bug-free\*\*
* Ensuring the product meets business requirement.
* Ensuring the product meets technical requirement.
* Ensuring the product efficiently handles all use cases.

**Official Definition (Source ISTQB):**

Testing is the process consisting of all lifecycle activities, both static and dynamic, concerned with planning, preparation and evaluation of software products and related work products to determine that they satisfy specified requirements, to demonstrate that they are fit for purpose and to detect defects.

**Objectives of Testing: -**

* Bug detection.
* Bug prevention.
* Assure quality of software/Product.
* Increase customer satisfaction.
* Security, usability, performance.
* Reduce cost.
* Improvement in process.

**Principles of Testing: -**

There are 7 principle of testing:

* Testing shows presence of defects.
* Exhaustive testing is not possible.
* Early testing.
* Defect clustering.
* Pesticides paradox.
* Testing is context dependent.
* Absence of errors fallacy.

**Common terms in testing: -**

**Error:** A mistake, misconception, or misunderstanding made by humans that leads to discrepancy.

**Defect:** The variation between the actual results and expected results.

**Bug/Ticket/Incident:** Synonyms of “Defect”, used interchangeably in companies. For example, some companies call it a “Bug” if it is detected by real users.

**Failure:** Anything that causes the software to fail to perform its required function.

**Debugging:** The process of finding, analyzing and removing the cause of failure in software.

**Verification:** The process of evaluating work-products (not final product) to determine whether they meet the specified requirements. “*Are we building the product right?*”

* Checking documents, codes, designs
* Non-execution testing (static testing).
* Happens before validation.

**Validation:** The process of evaluating software during or at end of development process to determine whether it satisfied business requirements. *“Are we building the right product*?”.

* Testing actual product.
* Execution testing (dynamic testing).

**Black Box Testing:** Testing without any reference to internal structure of component/system.

**White Box Testing:** Testing based on an analysis of internal structure of component/system.

**Severity:** Indicates the degree of impact the defect has on functionality of component or system (Testers evaluation). Functional importance.

**Priority:** Indicates how quickly the bug should be fixed. (Business evaluation).

* **High Severity & High Priority:** Key features failed and no workaround e.g. Login button not working
* **High Severity & Low priority:** Key features failed but there is no impact on customers’ business. E.g. Calculation fault in yearly report which end user won’t use regularly.
* **Low Severity & High priority:** Basic features failed butthere is huge impact on customer’s business. E.g. Misspelled company Logo.
* **Low Severity & Low priority:** Cosmetic issues. E.g. Font family mismatch in a report.

**Risk:** Probability of an event, hazard, accident and its impact (when happens).

**Risk based Testing:** Testing oriented and prioritized on the basis of importance of a feature or component, and its likely chance of failure.

**Static Testing:** Testing of a component or system without execution of that software.

**Dynamic Testing:** Testing that involves execution of the software of a component or system.

**Agile Testing:** Testing practices for a project using agile methodologies, such as emphasizing the test-first design (TDD).

**Continuous Testing:** Testing practice that involves the process of testing early on, testing more often, testing everywhere, and automating.

**STLC: Software Testing Life Cycle**

Testing process with specific steps to be executed in a pre-defined sequence to ensure that quality goals are met.

* Consists of 6 phases.
* Pre-defined sequence.
* All phases are necessary to achieve desired quality.
* Subset of software development life cycle (SDLC).
* Used by testing team.

**Phases of STLC: -**

Requirement Analysis

↓

Test Planning

↓

Test design(Development)

↓

Environment Setup

↓

Test Execution & reporting

↓

Test closure

**Software Testing Models: -**

1. **Waterfall Model: -**

Most basic, highly used but now replacing with agile.

1. **V-Model: -**

Superior than waterfall, but rigid model.

1. **Rapid Application Development (RAD) Model: -**

Incremental, module based, costly & complex.

In this components are development in parallel & finally integrated.

1. **Spiral Model: -**

For complex system, focus on risk analysis

For large & complex projects only.

1. **Agile Model: -**

Most common used now-a-days

Ensure continuous development in line with changing business requirements.