ISTQB Syllabus Notes

Focus: Fundamentals of Testing & Testing Levels

◆ 1. Why Testing is Necessary

- **Defects can occur at any stage** → A requirement may be misunderstood, a developer may make a coding mistake, or deployment may introduce issues.
- **Impact of defects** → If defects remain in production, they can cause financial loss, reputational damage, or even safety issues.
- **Cost of fixing late defects** → Fixing a bug in production costs much more than fixing it during requirements or design.

◆ 2. What is testing?

- **Definition** → Software testing is a set of activities performed to find defects or bugs and evaluating the quality of software product.
- **Difference between Testing vs Debugging** → Testing focuses on finding the defects or errors in a software while debugging emphasizes on analyzing and fixing those bugs or defects.

◆ 3. Seven Testing Principles

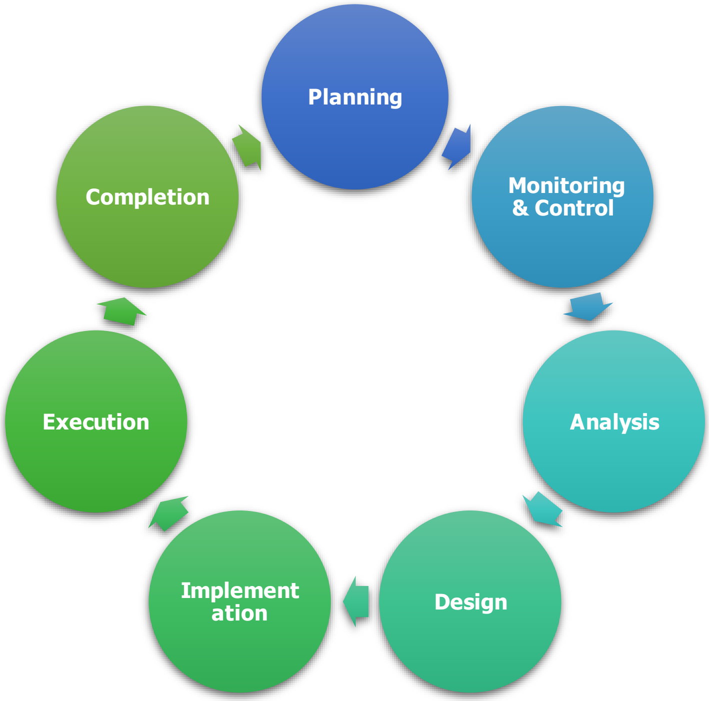
1. **Testing shows presence of defects, not their absence** → testing can show the defects that are present in the testing product but can't prove that there are no bugs in it.
2. **Exhaustive testing is impossible** → Rather than wasting time on testing each and every module of a software, we can use test techniques, risk based and prioritization testing etc to reduce time, cost and testing resources.
3. **Early testing saves time and money** → In order to save resources, like cost and time put into removing defects, we need to perform static and dynamic testing as early as possible in the SDLC.
4. **Defects cluster** together → It means that it's not probable that defects are spread out evenly across system. If we identify the big-prone areas, based on past experience, complexity or history, in our system then we can focus our testing there to find more defects faster, saving time and effort.
5. **Tests wear out** → Repeating same tests again and again doesn't find any new defects, testers should keep experimenting and design new and varied test cases to find new errors.

6. **Testing is context dependent** → Testing varies based on the context, there’s no universally applicable approach to test a software product.
7. **Absence-of-errors fallacy** → It is a misconception that a bug-free system will ensure success of the product. Validation should also be carried out in addition to verification. Thoroughly testing and fixing all the defects in a system doesn’t mean that it will fulfil user’s needs and expectations.

◆ 4. Fundamental Test Process

Steps:

1. **Test Planning** → It involves defining the test objectives and finding the best approach that helps us achieve those objectives within the given constraints.
2. **Test Monitoring & Control** → To monitor all the ongoing test activities and comparing actual progress against the objectives**.**
3. **Test Analysis** → It includes the requirements/design to assess the testability of a test object and then forming them into test cases. Test analysis answers the question “what to test?” in terms of measurable coverage criteria.
4. **Test Design** → Elaborates about how to make our test conditions executable. Involves writing test cases, deciding test data, setting up the environment and choosing tools to run tests later. Test design answers the question “how to test?”
5. **Test Implementation** → The stage where all the resources required for testing are prepared and organized. It involves collecting or creating test data, writing manual or automated test scripts, grouping test cases into test suites and setting priorities for execution.



6. **Test Execution** → This involves running manual or automated tests based on testing schedule, comparing actual results with expected results, logging results and analyzing anomalies.

7. **Test Completion** → In this phase, testing activities are formally closed as the project has reached a checkpoint and the testing phase is officially wrapped up. Any unresolved defects are documented and added to backlog for future fixes, any useful artifacts are archived instead of discarding them, the test environment is shut down properly, lessons are learned for process improvement and a test completion report is shared with stakeholders.

◆ 5. Testing Levels

Testing Level	Focus	Who Performs	Defects Found
Unit / Component	Individual modules/components	Developers	Coding & logic defects
Integration	Interaction between components	Dev + Testers	Interface/communication issues
System	Complete system behavior	Testers	Functional & non-functional defects
Acceptance	Validation against business needs	End-users / Clients	Usability, requirements gaps