

1) Basics of Software Testing

Page No.	1
Date	

1.1 Software Quality, Definition of Software Testing,

Role of Testing :- ~~part to 2 to 20 going on~~ ①

Software Quality :- Software quality measures how well software is designed (quality of its ~~part to 1~~ design), and how well software conforms to that design (quality of conformance).

~~part to 7 to - 9 part 10 to 21 part 24 to 27 part 28~~ ②

Key points :- ~~part to 10 to 21 part 24 to 27 part 28~~

① Software requirements are the foundations

from which quality is measured.

② Lack of conformance to requirement is lack

of quality or ~~part to 10 to 21 part 24 to 27 part 28~~ ②

③ Specified standard defines a set of development criteria that guide the manager in software engineering.

④ If criteria are not followed, lack of quality ~~part to 10 to 21 part 24 to 27 part 28~~ ③.1

⑤ A set of implicit requirements often goes unmentioned, for example ease of use, maintainability, etc.

⑥ If software conforms to its explicit requirement, but fails to meet implicit requirements, software quality is suspected.

~~part to 10 to 21 part 24 to 27 part 28~~ ④

Definition of Software Testing :-

Software Testing is a process, or a series of processes, designed to make sure what it was designed to do and that it does not do anything unintended.

Role of testing: ~~initial, ongoing, throughout~~

- ① The purpose of software testing can be quality assurance, verification and validation or reliability estimation.
- ② Correctness testing and reliability testing are two major areas of testing.
- ③ Software testing is a trade-off between budget, time and quality.
- ④ Software testing is to assess and evaluate the quality of work performed at each step of software development process.
- ⑤ The goal of testing is to ensure that the software performs as intended, and to improve software quality, reliability and maintainability.

1.2 Failure, Error, Fault, Defect, Bug terminology :-

Defect :- Commonly refers to several troubles with software products, with its external behaviour or internal features.

Failure :- The incapacity of a system to conduct its required functions within specified performance requirements.

Fault :- A false, wrong step, process or data definition in the software product.

Error :- A person act that generates an erroneous result. It refers to missing or wrong person action resulting in certain fault being injected into software.

- Software faults or defects are generally referred to as software bugs. (A)

~~Objectives of testing~~ (from p. no. 45 H)

1.3

Objectives of Testing

- Finding defects which may be get created by developer programmer while developing the software.
- Gaining confidence and providing information about level of quantity & nature of defects.
- To prevent defects in the program.
- To make sure that the result meets the business and user requirements.
- To ensure that it satisfies BRS (Business Requirements Specification) and SRS (Software Requirements Specification).
- Gain the confidence of the customer by providing them a quality product.

1.4 What is test case?

A test case is a documentation which specifies input values, expected output and the preconditions for executing the test.

Objectives behind writing and executing test cases are:

- ① Find defects in software products.
- ② Verify the software meets the end user requirement.
- ③ Improve software quality.
- ④ Minimize the maintenance and software support cost.

- ⑤ Avoid post development risks.
- ⑥ Compliance with processes at beginning.
- ⑦ Help management to make software delivery decisions.

~~1.5 When to Start and Stop testing of Software:~~

In order to know when to start and when to stop the testing work, it is very much important to define entry and exit criteria for the success of any project.

Entry Criteria: They are the minimum eligibility or the minimum set of conditions that should be met in order to start the testing process.

Typical entry criteria may include:

- ① All hardware platforms must have been successfully installed, configured and functioning properly.
- ② All necessary documentation, design and requirements information should be available that will allow testers to operate the system and judge the correct behaviour.
- ③ All standard software tools including the testing tools must have been successfully installed and functioning properly.
- ④ All personnel involved in the testing effort must be trained in the tools to be used in the testing process.

⑤ Proper test data is available.

⑥ The test requirements such as lab, hardware, software and system administration should be ready.

Exit Criteria :- They are the minimum eligibility or the set of conditions that should be met in order to close a particular project.

Typical exit criteria may include the following :-

① All test plan have been written.

② A certain level of requirements coverage has been achieved.

③ High priority bugs are left outstanding.

④ Although risks have been fully tested with only minor residual risk left outstanding.

⑤ Cost - When budget has been spent.

⑥ The schedule has been achieved.

1.6 Skills of Software Testers :-

① Communication skills : Testers are expected to be good listeners as well as good presenters. A good software tester must have strong verbal and written communication skills. They require

minimum good communication with developers before, during and after the project.

② Domain knowledge :- Testers should have general and detailed knowledge about the software like its architecture, application, whether they are not domain experts and this knowledge will help them to find such errors which ordinary users can face while using the application.

③ Desire to learn :- Testers should have the desire to learn.

④ Technical skill :- A good software tester should have strong technical skills. They must have proper knowledge of coding skills in order to understand the application, good communication with developers and write test automation. These skills can be obtained by practicing and proper training.

⑤ Analytical skills :- A good software tester should have the ability to check out how to reproduce the errors because only finding the errors is not sufficient. For better understanding and creating of good test reports, analytical skill will break the complex software system into smaller units.

⑥ Planning :- For better judgement of testing report, good planning is very necessary.

⑦ Integrity :- Testers find the errors in applications but they must withdraw assurance that developers will always absolutely fix them. The testing report should have to show the priorities of the suggestions.

⑧ Curiosity :- As every domain has its own speciality, so testers must have the curiosity to understand the domain under testing. They should have an eagerness of understanding the complexity and expectations.

⑨ Think from user's perspective :- Customers may not have all the technical skills as the tester have; if you fail to keep this in mind, you may miss important bugs.

⑩ Be a good judge of your product :- You have to be a good judge of your product. Ask yourself questions whether the software is meeting all the requirements it should have.

With respect to

1.7 Quality Assurance, Quality Control, Verification and Validation, V model, waterfall

□ **Quality Assurance :-** It is any systematic process of checking to see whether a product or service being developed is meeting specified requirements. A quality assurance system is said to increase customer confidence and company's credibility, to improve work processes and efficiency and to enable a company to better compete with others.

□ **Quality Control :-** It is a procedure or set of procedures intended to ensure that a manufactured product or performed service adheres to a defined set of quality criteria and meets the requirements of the client or customer. It is also defined as a procedure or set of procedures intended to ensure that a product or service under development meets specified requirements.

Quality Assurance

- 1) It is to develop good process.
- 2) QA is for process
- 3) QA is used to follows process and avoiding bugs
- 4) It is separate process
- 5) It gives confidence to you
- 6) It is preventing defects

Quality Control

- It is to develop good team
- QC is for product
- QC is used to follows testing for product
- quality and avoiding bugs
- QC is part of SDLC
- It gives expected results
- It is correcting defects

~~Verification and Validation~~

There are two aspects of V&V tasks:

- ① Conforms to requirement (Producer's view of quality)
- ② Fit for use (Consumer's view of quality)

Verification

It is a process of evaluating the intermediary work products of a software development life cycle (SDLC) to check if we are in right track of creating the product.

Validation: It is the process of evaluating the final product to check whether the software meets business needs.

Verification vs Validation

Verification	Validation
1) Evaluates the intermediary products to check whether it meets the specific requirements of the particular phase.	Evaluates the final product to check whether it meets the business needs.
2) Checks whether the product is built as per the specified requirement and design specification.	It determines whether the software is fit for use and satisfy the business need.
3) "Are we building the product right?"	"Are we building the right product?"
4) This is done without executing the software.	It is done with executing the software.

Verification vs Validation

- 5) Involves all the static testing techniques. Includes all the dynamic testing techniques
- 6) Ex:- reviews, inspections & walkthroughs (functional walkthroughs)
- Ex:- all types of testing like smoke, regression, functional, systems & UAT finalization

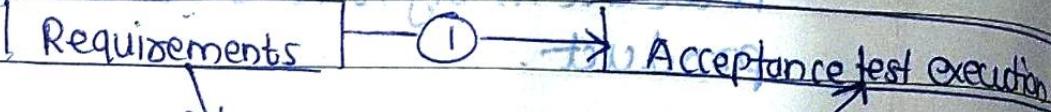
~~soft boundaries to working part~~

~~soft boundaries to working part~~

~~soft boundaries to working part~~

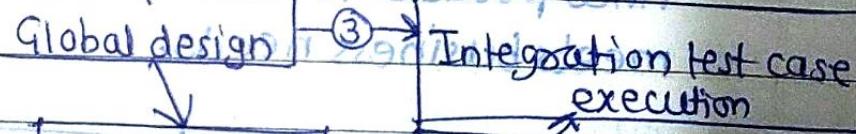
~~V Model :-~~

~~validation~~ ~~verification~~



~~soft boundaries to working part~~

~~soft boundaries to working part~~



~~soft boundaries to working part~~

~~soft boundaries to working part~~

① = Preparation of acceptance test cases

② = System test cases

③ = Integration test cases

④ = Unit test cases

~~soft boundaries to working part~~

SDLC = Software Development Life Cycle

Page No.	11
Date	

Before starting the actual testing, testing team has to work on various activities like preparation of test strategy, test planning, creation of test cases and test scripts which works parallel with the development activity which helps to get the tests/deliverable on time.

In V-Model SDLC, based on same information (Requirement specification document) the development and testing activity is started.

The V-Model is also called as Verification and Validation model. The testing activity is performed along with each phase of SDLC phases. In the first half of the model, validation testing activity is integrated and in the next half verification testing activity comes into picture. Typical V Model shows Software Development activities on left side of model and on the right hand side of the model actual testing phases can be performed.

In this procedure, "Do procedure" would be followed by developer team and "Check procedure" would be followed by the testing team to meet mentioned requirements.

* Unit testing :- Preparation of unit test cases

* Integration testing :- Preparation of integration test cases

* System testing :- Preparation of system test cases.

* Acceptance testing :- Preparation of acceptance test cases.