

Name: Rasana Lalani  
 En.no.: 92100588005  
 Subject: Software Testing  
 Course: BSC IT-6A

## Assignment

Q-1 Define terms: Testing, Failure, Fault, Error, Test case, Testware, Incident, Test Oracle

Ans Testing: It is a process to evaluate product for any kind of defects

Failure: Failure means the inability of a system or component to perform a required function according to its specification

Fault: Fault is a condition that in actual causes a system to produce failure. It is also called as bug

Error: Whenever a developer makes a mistake in any phase of SDLC, errors are produced. It might be logical or syntax error, generally called as human mistakes

Test case: A test case refers to the actions required to verify a specific feature or functionality of software testing.

Testware: The documents created during testing activities are known as testware.

Incident: An incident is the symptoms associated with a failure that alerts the user about the occurrence of a failure.

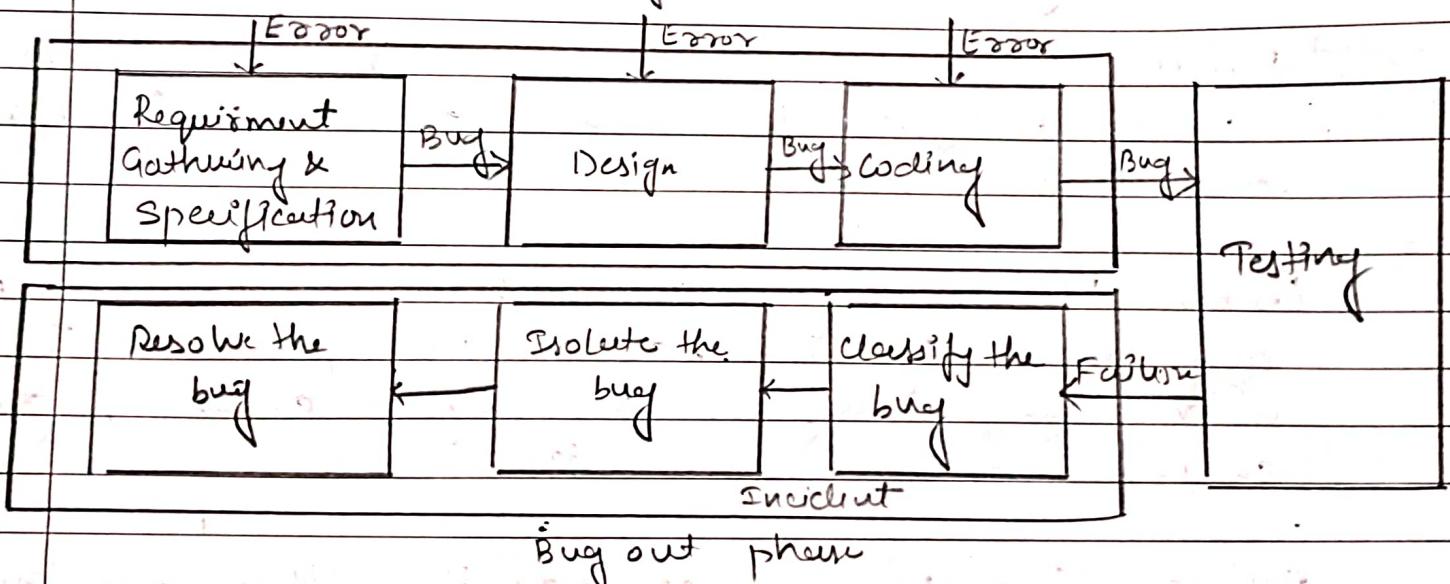
Test oracle: An oracle is the means to judge the success or failure of test. The simplest oracle is comparing actual results with expected results.

Q-2 What is Bug? Explain Life cycle of bug in details.

Ans It can be said that failure is manifestation of bugs.

One failure may be due to one or more bugs and one bug may cause one or more failures.

Bug in phase



It should be clear that any number of the development team can make an error in any phase of SDLC.

M	T	W	T	F	S
Page No.					
Date					FOUVA

If an error has been produced in the requirement specification phase and not detected in the same phase, then it results in a bug in the next phase i.e. design phase.

In the design phase, a bug has come from the previous stage, but an error can also be produced in this stage. Again, if the error in this phase is not deleted and it passes on to the next stage i.e. coding phase.

then it becomes a bug. In this way, errors and bugs appear and travel through various stages of SDLC.

In the testing phase, we analyze the incidents when the failure occurs.

On the basis of symptoms a bug can be classified into certain categories.

After this the bug can be isolated in that phase of SDLC and removed by finding PIs exact location.

The whole life cycle of a bug can be classified into two phases:

(i) Bug-in Phase

This phase is when the error and bug are introduced in the software.

Whenever we commit a mistake, it creates errors in specific location of the software. When this error goes unnoticed, it causes consequences to fail, leading to a bug in the software.

If you are not performing verification on earlier phases, then there is no chance of detecting these bugs.

## (ii) Bug -out Phase

If failure occurs while testing a software product, we come to the conclusion that it is affected by bugs.

However, there are situations where bugs are present, even though we don't observe any failure.

In this phase when we observe failure, the following activities are performed.

### o Bug classification

We observe the failure and classify the bug according to its nature.

A bug can be critical or catastrophic in nature or it may have no adverse effect on the output behaviour of the software.

this is necessary, because there may be many bugs to be resolved. But a tester may not have sufficient time.

- **Bug Isolation**

Bug isolation is the activity by which we locate the module in which the bug appears. Inclusion occurs in failure help in this activity.

We observe the symptoms and back-trace the design of the software and locate the module/file and the condition inside it, which has caused the bug. This is known as Bug Isolation.

- **Bug Resolution**

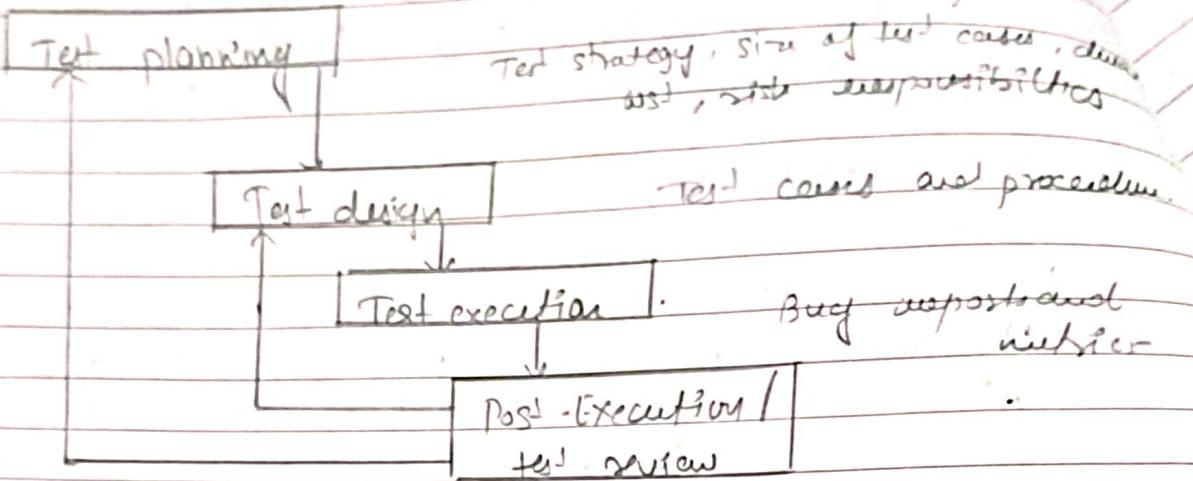
Once we have isolated the bug, we back-trace the design to pinpoint the location of the error.

In this way, a bug is resolved when we have found the exact location of its occurrence.

### Q-3 Explain STLC in detail with diagram

- **Test Planning**

The goal of test planning is to take into account the important issues of testing strategy, resources, schedules, responsibilities, risks and priorities as a roadmap.



Based on planning issues, analysis is done for various testing activities.

The major output of test planning is the test document.

Test plans are developed for each level of testing.

After analyzing the issues, the following activities are performed.

- Develop a test-case format
- Develop test case plans according to every phase of SDLC
- Identify test cases to be automated
- Prioritize the test cases according to their importance and criticality
- Test Design

One of the major activities in testing is the design

Page No.	
Date	PERIOD

of test cases.

The test design is an important phase after test planning. It includes the following critical activities:

- Determining the test objectives and their prioritization
- The test objectives reflect the fundamental elements that need to be tested to satisfy an objective
- The activity involves the broad categories of things to test.

Preparing list of items to be tested.

- The objectives thus obtained are now converted into lists of items that are to be tested under an objective.

Mapping items to test cases

- After mapping a list of items to be tested there is a need to identify the test cases. A matrix can be created for this purpose, identifying which test case will be conducted by which item.

Selection of test case design techniques

- While designing test cases there are two broad categories, namely black-box testing and white-box testing.

~~Creating test cases and test data.~~

- Setting up the test environment and supporting tools

Creating test procedure specification.

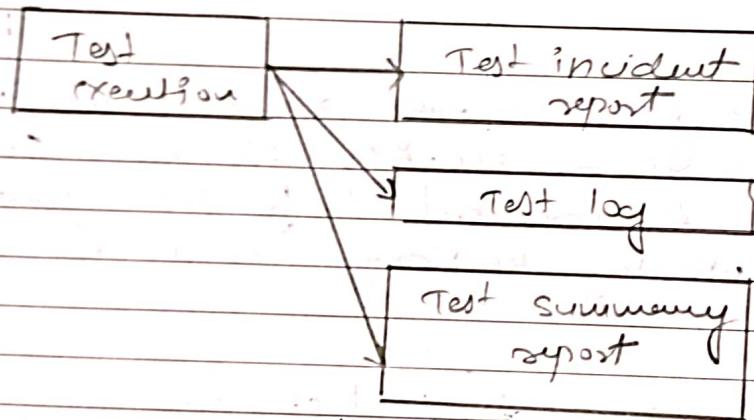
→ This is a description of how the test cases will be run. It is the form of required steps. This procedure is actually used by the tester at the time of execution of test cases.

#### • Test Execution.

In this phase all test cases are executed including verification and validation.

Verification test cases are started at the end of each ~~pre~~ phase of SDLC.

Validation test cases are started after the completion of review.



#### • Post Execution / Test Review

After successful test execution, bugs will be reported to the concerned developer.

This is the primary goal of all test activities done earlier.

As soon as the developer gets the bug report he performs the following activities

- Understanding the bug:

- Reproducing the bug

- Analyzing the nature and cause of the bug.

(Q-4) Differentiate between Blackbox Testing and Whitebox testing.

### White-Box testing

1. White box testing techniques are used for testing the module for initial stage testing.

2. The logical part of the software is checked in white Box testing.

3. Errors which have come from the design phase will also be reflected in the code.

4. Typographical errors are detected

### Black-Box testing

Black-Box testing is the second step for testing the software.

The functionality of the software is checked in Black-Box testing.

They are not focused as the software is running smoothly.

Typographical errors are not detected.

Q-5 Define terms : Stub & Driver , Unit Test , Integration Testing.

Ans Driver : The module where the required inputs for the module under test are simulated for the purpose of module or unit testing is known as driver module.

Stub :- Dummy module instead of actual module, which are not ready, are prepared for their subdomain module. This dummy module are called as stub.

Integration testing :-

In modular design of a software system where the system is composed of different modules, integration is the activity of combining the modules together when all the modules have been prepared.

Unit Test :-

Unit is the smallest building block of the software system, it is the first piece of system to be validated.

Unit must also be validated to ensure that every unit of software has been built in the right manner in conformance with other requirement.

(Q-6) Write comparison between top down and bottom up & integration testing

Issues	Top-Down Testing	Bottom-up Testing
Architecture	It discovers errors in high-level design, thus detects errors at an early stage.	High-level design is validated at a later stage.
System Demonstration	Since we integrate the demonstration module from top to bottom, the high-level feasibility of the design slowly expands however, if some as a working system, modules are already built as reusable components. Then the system can be demonstrated to the top management.	It may not be possible to show the feasibility of the design. However, if some modules are already built as reusable components, then it may be possible to procure some kind of demonstration.
Test	(Module) stubs are required for the stub or client modules.	(nodes - leaves) test drivers are required. Super-client module to test the lower-level nodes.

(Q-7) Differentiate between Alpha testing and Beta testing

### Parameters

### Alpha testing

### Beta testing

Input

Alpha testing involves both the white box and black box testing

Beta testing commonly uses black box testing

Performed by

Alpha testing is performed by testers who are usually internal employees of the organization

Beta testing is performed by clients who are not part of the organization

at

Alpha testing is performed at the developer's site

Beta testing is performed at the end-user of the product

Reliability and Security

Alpha testing

Reliability, security and robustness are checked during beta testing

~~Ensures~~ Alpha testing requirement requires a testing environment on a lab

Beta testing doesn't require a testing environment on lab

Ensures

Alpha testing ensures the quality of the product before proceeding to beta testing

Beta testing also focuses on the quality of the product but collects user input on the product and ensures that the product is user-friendly

for next time  
uses

Execution	Alpha testing may require a long execution cycle	Beta testing requires only a few weeks of execution
Issues	Developers can immediately address the critical issues, as fixes in alpha testing	Most of the issues or feedback collected from the Beta testing will be implemented in future version of the product
Test cycles	Multiple test cycles are organized in alpha testing	Only one or few test cycles are there in beta testing

## -8 Explain Master Test Plan in details

The master test plan provides the highest level description of verification and validation efforts and drives the testing at various levels

General project information is used to develop the master test plan.

The following topics must be addressed before planning:

- Project identification
- Main goals
- Summary of verification and validation efforts
- Responsibilities conveyed with the plan
- Software to be verified and validated
- Identification of changes to organization Standards

#### \* Master Schedule

- The master schedule summarizes various V&V tasks and their relationships to the overall project.
- Describes the project life cycle and project milestones including completion dates.
- Summarizes the schedule of V&V tasks and how verification and validation results provide feedback to the development process to support overall project management function.

#### \* Resource Summary

- This activity summarizes the resources needed to perform V&V tasks, including staffing, facility, tools, finance and special procedural requirements such as security, access rights, and documentation control. In this activity:
  - Use graphs and tables to present resource utilization.

Q	F	M	T	S	O
Prepared					
Date					VIVEKA

- Include equipment and laboratory return required.
- Summarize the purpose and cost of hardware and software tools to be employed.

#### \* Responsibilities

- Identify the organization responsible for performing V & V tasks.
- There are two type of responsibilities - general responsibilities assigned to different organization and specific responsibilities for the V & V tasks to be performed assigned to individuals.

#### \* Tools, Techniques, and Methodology

- Identify the special software tools, techniques and methodologies to be employed by the V & V team.
- The purpose of each should be defined and the plans for the acquisition, training, support and qualification of each should be described.

#### Q-9 Explain CMM Model in Detail.

Ans - Capability Maturity Model is a framework meant for software development pro-

## CMM Structure:

The Structure of CMM consists of five maturity levels. These levels consist of key process areas (KPAs), which are organized by common features.

Maturity level	contains	key process areas	organized by	common features	contains	key practices
----------------	----------	-------------------	--------------	-----------------	----------	---------------

- The maturity level protocols the process
- Causality key practices describe the causality to be done
- common feature address implementations. In this way, by working on the activities described in key practices under some common features, we achieve a goal under a KPA.
- Each KPA identifies a cluster of related activities that, when performed collectively achieve a set of goals considered appropriate for enhancing the process respectively

Maturity level	Indicate	Process capability
Key process areas	Achieve	Goals
Common features	Address	Implementation / Institutionalization
Key practices	Describe	Infrastructure / Activities

## CMM Structure:

The Structure of CMM consists of five maturity levels. These levels consist of key process areas (KPAs), which are organized by common features.

Maturity level	contains	key Process areas	organized by	common feature	contains	key practices
----------------	----------	-------------------	--------------	----------------	----------	---------------

- The maturity level indicates the process capability key practices describe the capacity to be attained.
- common feature address implementations. In this way, by working on the activities described in key practices under some common features, we achieve a goal under a KPA.
- Each KPA identifies a cluster of related activities that, when performed collectively, achieve a set of goals considered important for enhancing the process capability.

Maturity level	Indicate	Process capability
key process areas	Achieve	Goals
Common features	Address	Implementation / Institutionalization
key practices	Describe	Infrastructure / Activities

- Maturity levels - A brief description of the five maturity levels is given below.
- Initial - At this level, the process is chaotic or ad-hoc, where there is no control on development processes.
- Repeatable - Earlier project success can be used here and the lessons learned from past projects are incorporated in the current project.
- Defined - The management scope is widened to the organization.
- Managed - A quantitative understanding of the processes is established in order to monitor and control them.
- Optimizing - The process is continually improved through incremental and innovative technological changes or improvements.

#### Q-10 Explain Software quality Matrix in detail.

- Software quality metrics are a subset of software metrics that focus on the quality aspects of the product, process, and project.
- Software quality metrics can be grouped into the following three categories in accordance with the software lifecycle.

## ① Product Quality Metrics:

In product quality, the following metrics are considered:

- Mean-time of failure (MTTF) - metric is an estimate of the average or mean time until a product's first failure occurs.
- Defect density metric: It measures the defects relative to the software size.
- Customer problem metric - This metric measures the problems which customers face while using the product.
- Customer satisfaction metric - customer satisfaction is usually measured through various methods of customer surveys via five point scale.

## ② In Process Quality Metrics:

- Defect density during testing: Higher defect rates found during testing is an indicator that the software has experienced higher error injection during its development process.
- Defect arrival pattern during testing -

The pattern of defect arrivals on the line between consecutive failure give more information.

### ③ metrics for software maintenance.

In this phase, the goal is fix the defects as soon as possible and with excellent quality.

- Fix backlog and backlog management -  
Fix backlog metric is the count of reported problems that remain open at the end of a month or a week
- Fix response time and fix responsiveness -  
The goal is to get high fix responsiveness to meet customer expectations and have highly satisfied customers
- Fix quality - It is the metric to measure the number of fixes that are turn out to be defective.