Parul® University

Software Engineering(303105253)

Dr. Kapil Aggarwal, Associate Professor
Computer Science & Engineering

Parul®
University

Module-6

Software Testing and Quality Assurance: Concepts, Psychology of testing, Levels of testing, Testing Process-test plan, test case design, Execution, Black-Box testing ´Boundary value analysis ´Pair wise testing- state based testing, White-Box testing criteria and test case generation and tool support Quality Assurance : Quality Control, Assurance, Cost, Reviews, Software Quality Assurance, Approaches to SQA, Reliability, Quality Standards- ISO9000 And 9001

Parul® University

DIGITAL LEARNING CONTENT

Concepts

- Testing is the process of exercising a program with the specific intent of finding errors prior to delivery to the end user.
- To understand testing techniques that are geared to discover program faults
- To introduce guidelines for interface testing
- To understand specific approaches to object- oriented testing
- To understand the principles of CASE tool support for testing

Psychology of Testing

- All tests should be traceable to customer requirements.
- Tests should be planned long before testing begins.
- Testing should begin “in the small” and progress toward testing “in the large.”
- Exhaustive testing is not possible.
- To be most effective, testing should be conducted by an independent third party.

Software Testing Phases

- **Component testing**
  - Individual program components testing
  - Tests are derived from the developer's experience
- **Integration testing**
  - Testing of groups of components integrated to create a system or sub-system
  - Tests are based on a system specification

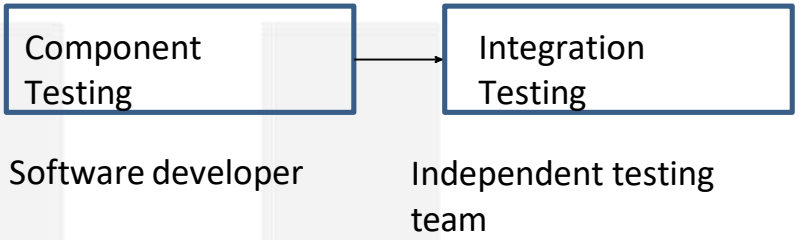


Figure 6.1 Software testing phase

Testing Process

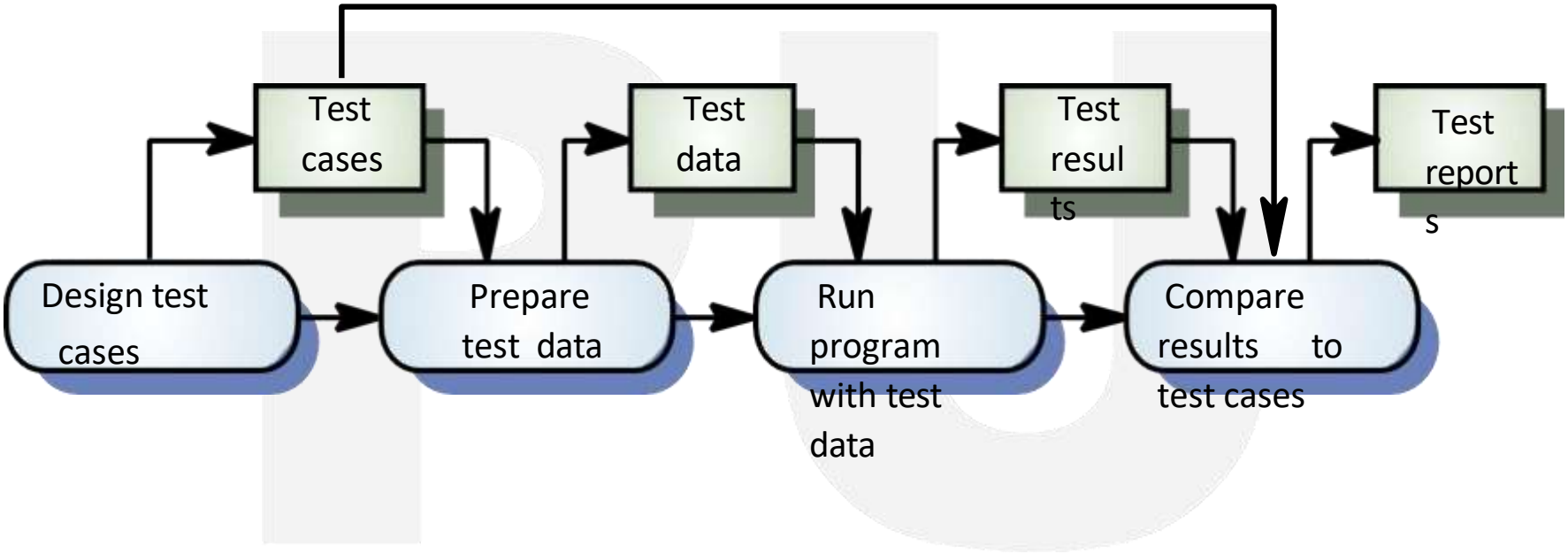


Figure 6.2 Software testing process

Test Plan

- Test planning, the most important activity to ensure that there is initially a list of tasks and milestones in a baseline plan to track the progress of the project.
- It also defines the size of the test effort.
- It is the main document often called as master test plan or a project test plan and usually developed during the early phase of the project.

Test Planning Activities

- To determine the scope and the risks that need to be tested and that are NOT to be tested.
- Documenting Test Strategy.
- Deciding Entry and Exit criteria.
- Evaluating the test estimate.
- Planning when and how to test and deciding how the test results will be evaluated, and defining test exit criterion.
- Ensuring that the test documentation generates repeatable test assets.

Test Case Design

- Test case design methods provide a mechanism that can help to ensure the completeness of tests
- Provide the highest likelihood for uncovering errors in software.

Any product or system can be tested on two ways:

1. Knowing the specified function that a product has been designed to perform; (Black Box)
2. knowing the internal workings of a product, tests can be conducted to ensure that "all gears mesh," that is, internal operations are performed according to specifications (White box testing)

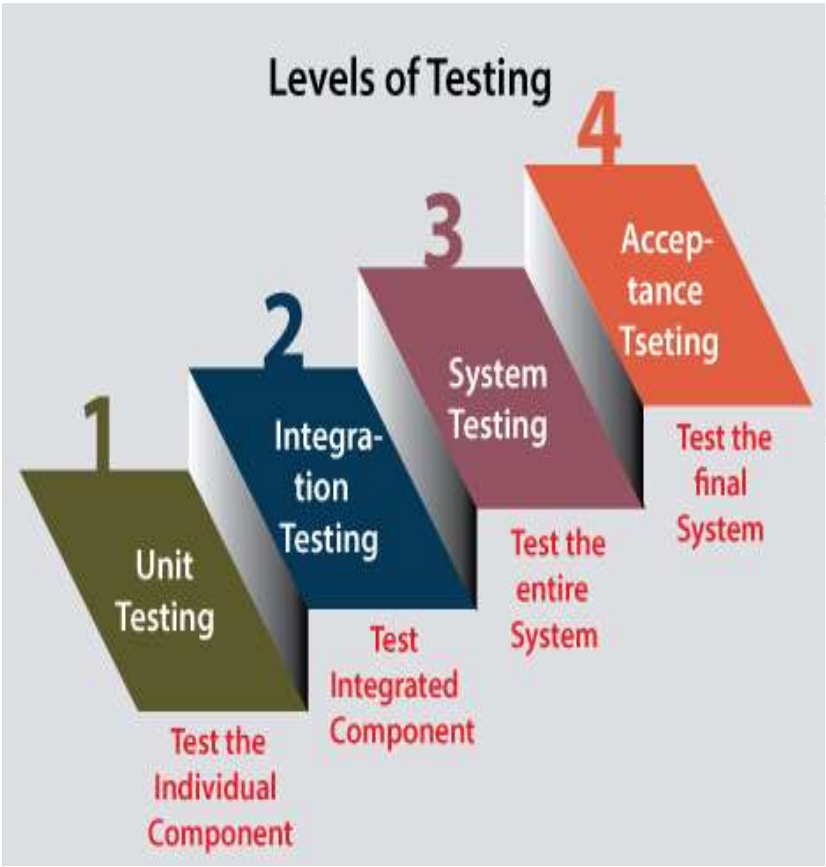
Test Execution

- Test execution is the process of executing the code and comparing the expected and actual results.

Following factors are to be considered for a test execution process:

- Based on a risk, select a subset of test suite to be executed for this cycle.
- Assign the test cases in each test suite to testers for execution.
- Execute tests, report bugs, and capture test status continuously.
- Resolve blocking issues as they arise.
- Report status, adjust assignments, and reconsider plans and priorities daily.
- Report test cycle findings and status.

Levels of Testing



1. **analyzing each unit or an individual component**

2. The primary purpose of executing the integration testing is to identify the defects at the interaction between integrated components or units.

3. which is used to test the software's functional and non-functional requirements.

4. Acceptance testing is formal testing based on user requirements and function processing. It determines whether the software is conforming specified requirements and user requirements or not

Levels of Testing

- There are different levels during the process of testing.
- Levels of testing include different methodologies that can be used while conducting software testing.

The main levels of software testing are –

- Functional Testing
- Non-functional Testing

Functional Testing

- This is a type of black-box testing that is based on the specifications of the software that is to be tested.
- Functional testing of a software is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

There are five steps that are involved while testing an application for functionality.

1. The determination of the functionality that the intended application is meant to perform.
2. The creation of test data based on the specifications of the application.
3. The output based on the test data and the specifications of the application.
4. The writing of test scenarios and the execution of test cases.
5. The comparison of actual and expected results based on the executed test cases.

Unit Testing

- This type of testing is performed by developers before the setup is handed over to the testing team to formally execute the test cases.
- Unit testing is performed by the respective developers on the individual units of source code assigned areas.
- The developers use test data that is different from the test data of the quality assurance team.
- The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

Integration Testing

- Integration testing is defined as the testing of combined parts of an application to determine if they function correctly.
- Integration testing can be done in two ways: Bottom-up integration testing and Top-down integration testing.

1. **Bottom-up integration**

This testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds.

2. **Top-down integration**

In this testing, the highest-level modules are tested first and progressively, lower-level modules are tested thereafter.

System Testing

- System testing tests the system as a whole.
- This type of testing is performed by a specialized testing team.

System testing is important because of the following reasons –

- System testing is the first step in the Software Development Life Cycle, where the application is tested as a whole.
- The application is tested thoroughly to verify that it meets the functional and technical specifications.
- The application is tested in an environment that is very close to the production environment where the application will be deployed.
- System testing enables us to test, verify, and validate both the business requirements as well as the application architecture.

Non-Functional Testing

- Non-functional testing involves testing a software from the requirements which are nonfunctional in nature but important such as performance, security, user interface, etc.

  Some of the important and commonly used non-functional testing types are discussed below.

  - Performance Testing
  - Load Testing
  - Stress Testing
  - Usability Testing
  - Security Testing
  - Portability Testing

Parul®
University

DIGITAL LEARNING CONTENT

Types of Software Testing

1. Manual Testing

2. Automation Testing

Manual Testing

Testing software only by human intervention is known as manual testing. Verification of existing conditions with requirements is done. Manual testing may include detailed step-by-step test cases for testing periods.

Automation Testing

Testing software by using automation tools is known as automation testing. There are various tools accessible that can assist you in automatically testing your software, which will be discussed later in this article.

Parul®
University

DIGITAL LEARNING CONTENT

Approaches to Software Testing

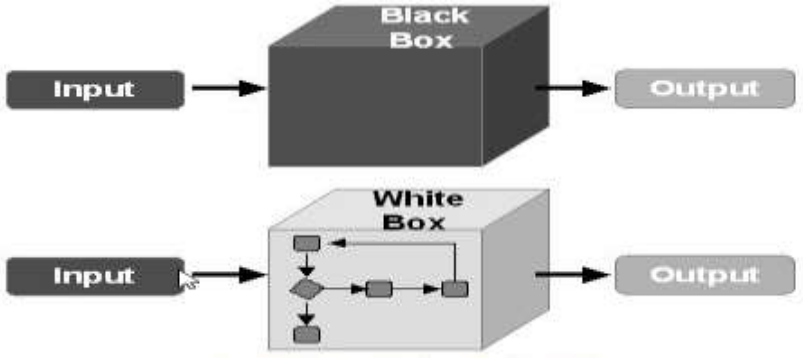
- White Box Testing
- Black Box Testing
- Grey Box Testing

White Box Testing

- White box testing is also known as clear box testing/glass box testing/structural testing/open box testing.
- White box testing analyses the inner functioning of the system.

Some advantages of White Box Testing are:

- It can be performed at the initial stages.
- It allows us to find hidden defec
- It helps in code optimization.

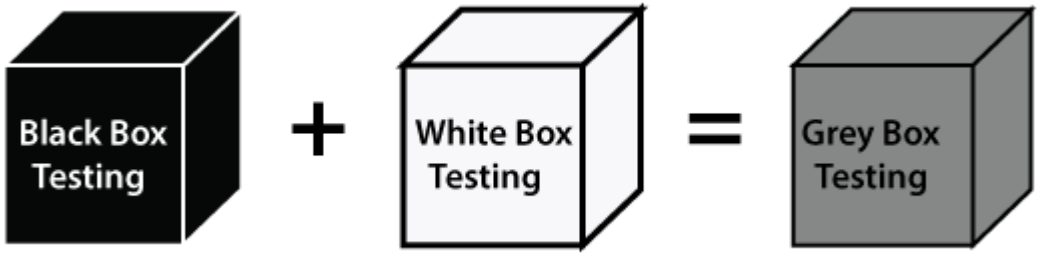


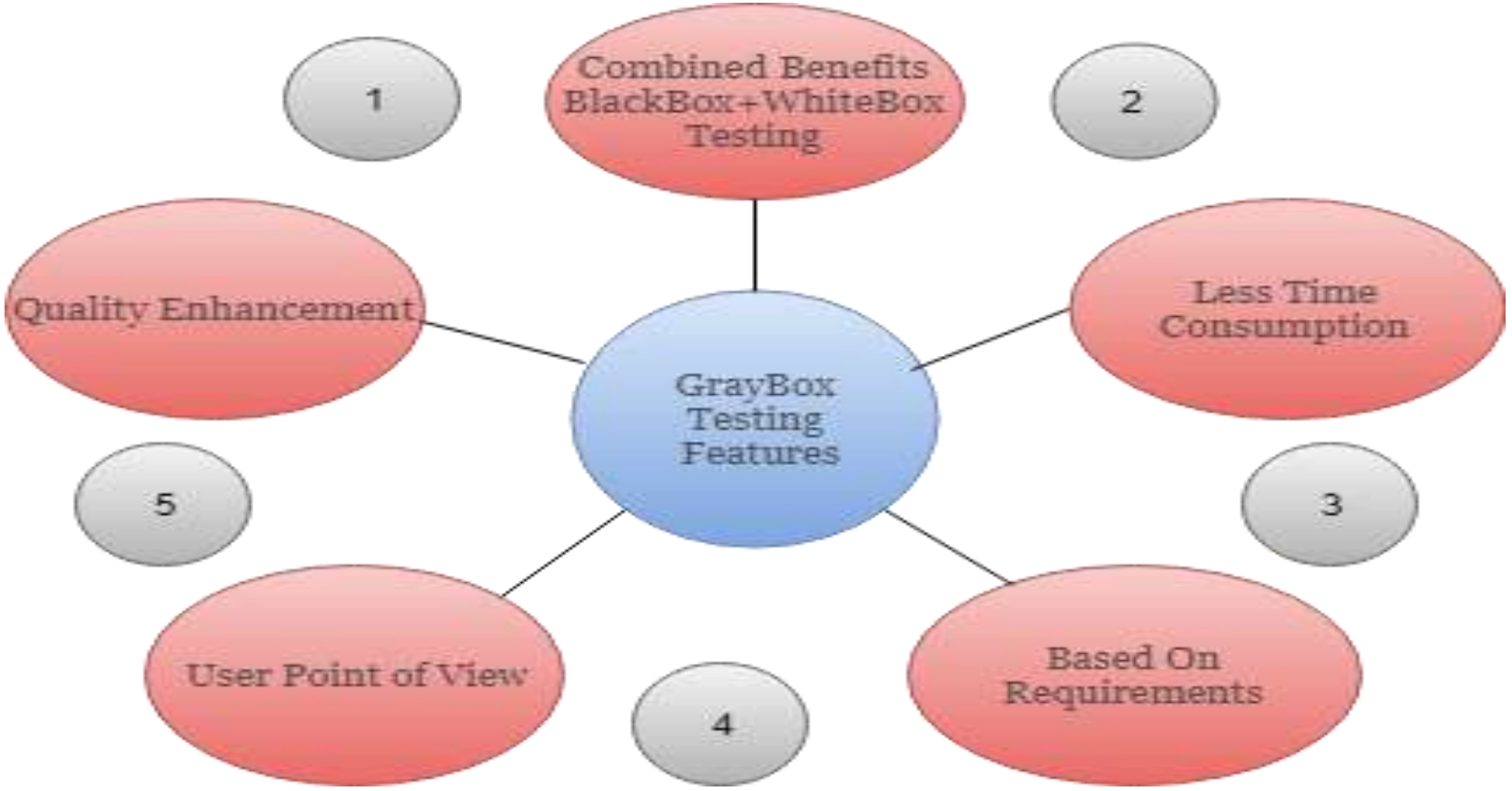
Black Box Testing

Another type of manual testing is **black-box testing**. In this testing, the test engineer will analyze the software against requirements, identify the defects or bug, and sends it back to the development team.

Grey Box Testing

- Another part of **manual testing** is **Grey box testing**. It is a **collaboration of black box and white box testing**.
- Since, the grey box testing includes access to internal coding for designing test cases. Grey box testing is performed by a person who knows coding as well as testing.




GrayBox Testing Features
1
Combined Benefits BlackBox+WhiteBox Testing
2
Less Time Consumption
3
Based On Requirements
4
User Point of View
5
Quality Enhancement

TEST PLAN

- A test plan is a detailed document which describes software testing areas and activities. It outlines the test strategy, objectives, test schedule, required resources (human resources, software, and hardware), test estimation and test deliverables.

Parul®
University

Seven Steps below to create a Test Plan as per IEEE 829

- Analyze the product
- Design the Test Strategy
- Define the Test Objectives
- Define Test Criteria
- Resource Planning
- Plan Test Environment
- Schedule & Estimation
- Determine Test Deliverables

Analyze the product

Before writing an effective test plan, you first need to analyze the product.
Who will use the website?
What is it used for?
How will it work?
What are software/ hardware the product uses?

Design the Test Strategy

A number of different factors need to be considered when developing a test strategy, including the type of testing, resources required, and schedule for testing.

Step 2.1 Define scope of Testing

Step 2.2 Identify Testing Type

Step 2.3 Document Risks & Issues

Step 2.4 Create Test Logistics

How do you determine scope your project?

To determine scope, you must –

Precise customer requirement

Project Budget

Product Specification

Skills & talent of your test team

Create Test Logistics

1. Who will test?

To select the right member for specified task, you have to consider if his skill is qualified for the task or not, also estimate the project budget. Selecting wrong member for the task may cause the project to **fail** or **delay**.

2. When will the test occur?

Define the Test Objectives

List all the software features (functionality, performance, GUI…) which may need to test.

Define the **target** or the **goal** of the test based on above features

Define Test Criteria

Suspension Criteria

Test Criteria is a standard or rule on which a test procedure or test judgment can be based. There're 2 types of test criteria as following. **suspended** until the criteria are **resolved**.

- Test Plan Example: If your team members report that there are **40%** of test cases failed, you should **suspend** testing until the development team fixes all the failed cases.

Exit Criteria

It specifies the criteria that denote a **successful** completion of a test phase. The exit criteria are the targeted results of the test and are necessary before proceeding to the next phase of development. Example: **95%** of all critical test cases must pass.

Resource Planning

- Resource plan is a **detailed summary** of all types of resources required to complete project task.
- **Human Resource**
- **System Resource**

No.	Member	Tasks
1.	Test Manager	**Manage** the whole project Define project **directions** Acquire appropriate resources
2.	Tester	Identifying and describing appropriate test techniques/tools/automation architecture Verify and assess the Test Approach **Execute** the tests, **Log** results, **Report** the defects. Tester could be in-sourced or out-sourced members, base on the project budget For the task which required **low** skill, I recommend you choose **outsourced** members to **save** project cost.
3.	Developer in Test	**Implement** the test cases, test program, test suite etc.
4.	Test Administrator	Builds up and ensures Test Environment and assets are **managed** and **maintained** **Support**Tester to use the test environment for test execution
5.	SQA members	Take in charge of quality assurance Check to confirm whether the testing process is meeting specified requirements

System Resource

For testing, a web application, you should plan the resources as following tables:

No.	Resources	Descriptions
1.	Server	Install the web application under test This includes a separate web server, database server, and application server if applicable
2.	Test tool	The testing tool is to automate the testing, simulate the user operation, generate the test results There are tons of test tools you can use for this project such as Selenium, QTP... etc.
3.	Network	You need a Network include LAN and Internet to simulate the real business and user environment
4.	Computer	The PC which users often use to connect the web server

Plan Test Environment

- What is the maximum user connection which this website can handle at the same time?
- What are hardware/software requirements to install this website?
- Does the user's computer need any particular setting to browse the website?

➢ Schedule & Estimation

Employee and project deadline: The working days, the project deadline, resource availability are the factors which affected to the schedule

Project estimation: Base on the estimation, the Test Manager knows how long it takes to complete the project. So he can make the appropriate project schedule

Project Risk : Understanding the risk helps Test Manager add enough extra time to the project schedule to deal with the risks

- Determine Test Deliverables

TEST CASE DESIGN

Project Name:	**Google Email**
Module Name:	**Login**
Reference Document:	**If any**
Created by:	**Rajkumar**
Date of creation:	**DD-MMM-YY**
Date of review:	**DD-MMM-YY**

STM
www.SoftwareTestingMaterial.com

TEST CASE ID	TEST SCENARIO	TEST CASE	PRE-CONDITION	TEST STEPS	TEST DATA	EXPECTED RESULT	POST CONDITION	ACTUAL RESULT	STATUS (PASS/ FAIL)
TC_LOGIN_001	Verify the login of Gmail	Enter valid User Name and valid Password	1. Need a valid Gmail Account to do login	1. Enter User Name	<Valid User Name>	Successful login	Gmail inbox is shown		
				2. Enter Password	<Valid Password>				
				3. Click "Login" button					
TC_LOGIN_001	Verify the login of Gmail	Enter valid User Name and invalid Password	1. Need a valid Gmail Account to do login	1. Enter User Name	<Valid User Name>	A message "The email and password you entered don't match" is shown			
				2. Enter Password	<Invalid Password>				
				3. Click "Login" button					
TC_LOGIN_001	Verify the login of Gmail	Enter invalid User Name and valid Password	1. Need a valid Gmail Account to do login	1. Enter User Name	<Invalid User Name>	A message "The email and password you entered don't match" is shown			
				2. Enter Password	<Valid Password>				
				3. Click "Login" button					
TC_LOGIN_001	Verify the login of Gmail	Enter invalid User Name and invalid Password	1. Need a valid Gmail Account to do login	1. Enter User Name	<Invalid User Name>	A message "The email and password you entered don't match" is shown			
				2. Enter Password	<Invalid Password>				
				3. Click "Login" button					

Parul®
University

Test Case Template

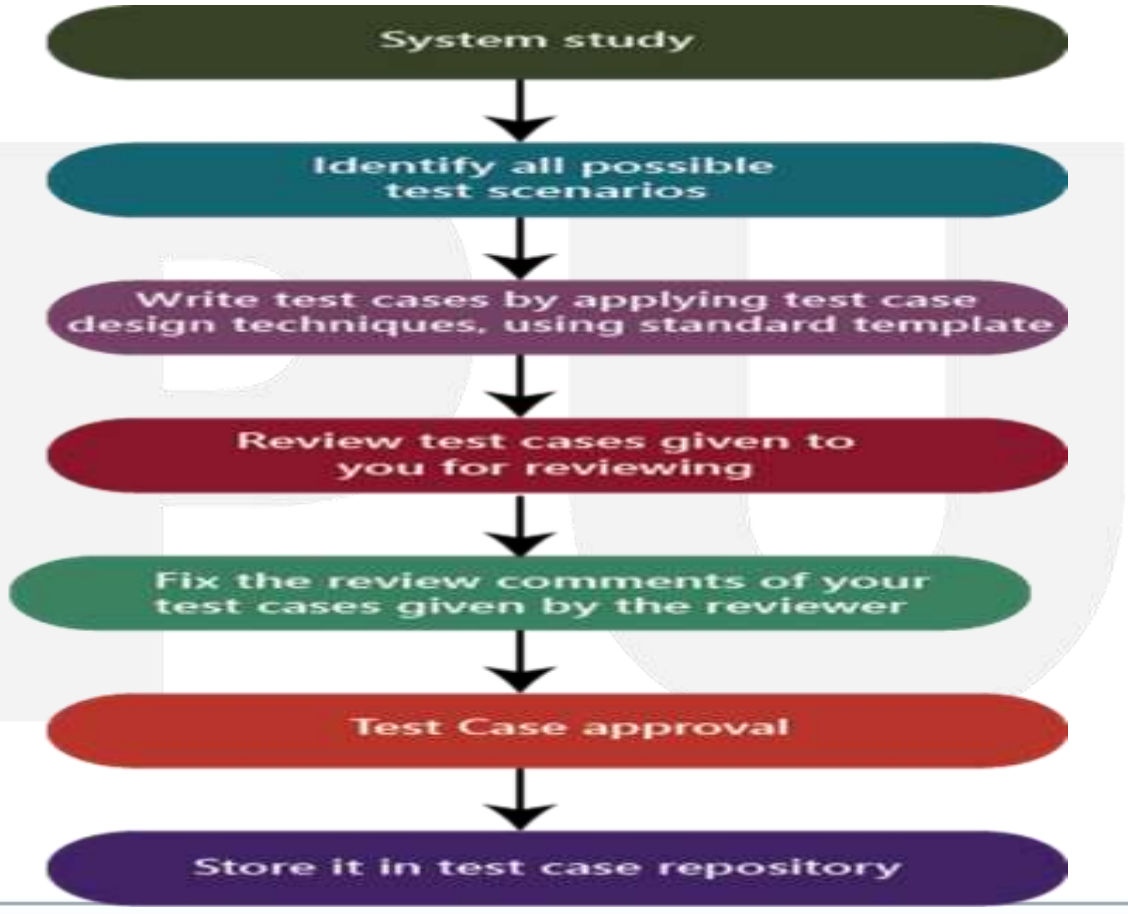
Functional Test case tamplate						
Test case name	beta-1.0-4CIO-amount transfer					
Test case type	Functional test case					
Requirement no	6					
Module	amount transfer					
Status	XXX					
Severity	Critical					
Release	Beta					
Version	1					
Pre-condition	sender login two account number Balance-->exist					
Test data	Username:xyz, Password:1234 1231, 4321 3000-9000					
Summary	to check the functionality of amount transfer					

if we are saying (5000-9000) balance, but if want to test for 9001, so obiously it will give the error message (unsufficent message)

Steps no	Description	Inputs	Expected result	Actual results	Status	Comments
1	Open "**Browser**" and enter the "**Url**"	https://QA/Main/#/	"**Login page**" must be display	As Expected	pass	XXX
2	Enter the following values for "**Username**"and "**Password**" and click on the "**OK**" button	xyz, 1234	"**Home page**"must be displayed	As Expected	pass	XXX
3	Click on the "**Amount Transfer**"	Null			pass	XXX
4	Enter the following for **From Account number(FAN):**					
	valid	1234	**Accpect**	As Expected	pass	
	invalid	1124	Error message invalid account		fail	
	blank	——	Error message FAN value cannot be blank		fail	
	——	——	——			
	——	test maximum coverage				
5	Enter the following values for "**TO account number**"(TAN)					
	valid	4321	**Accpect**	As expected	pass	XXX
	invalid	6655	Error message invalid account		fail	
	Blank	——	Error message TAN value cannot be blank			
	——	test maximum coverage	——			
6	enter the value for "**Amount**"					
	valid	5000, 5001,9000, 84	**Accpect**	As expected	pass	XXX
	invalid	4999, 9001	error message amount shoulb be between (5000-9000)		fail	
7	Enter the value for "**FAN, TAN, Amount**"click on the "**Transfer**" button					
	FAN	1234				
	TAN	4321	"**Confirmation Message**" amount transfer successfully must be displayed	As expected	pass	XXX
	Amount	6000				
8	Enter the value for "**FAN, TAN, Amount**"click on the "**Cancel**" button					
	FAN	1234				
	TAN	4321	**All field must be cleared**	As expected	pass	XXX
	Amount	6000				

Author	Sem	
Date	4/1/2020	
Reviewd by	jessica	
Approved by	ryan	

Process to Test Cases



Boundary value analysis

- Boundary value analysis is one of the widely used case design technique for black box testing. It is used to test boundary values because the input values near the boundary have higher chances of error.
- Assume that, age is a variable of any function, and its minimum value is 18 and the maximum value is 30, both 18 and 30 will be considered as boundary values.

Example:
accepts a number between 18 to 30, where 18 is the minimum and 30 is the maximum value of valid partition.

12 14 15 16 17 **18** 20 22 24 25 26 28 **30** 31 32 34 36 38 40

---|---|---

Invalid Partition | Valid Partition | Invalid Partition

Invalid test cases	Valid test cases	Invalid test cases
11, 13, 14, 15, 16, 17	18, 19, 24, 27, 28, 30	31, 32, 36, 37, 38, 39

Pair wise testing

- Pairwise Testing also known as All-pairs testing is a testing approach taken for testing the software using combinatorial method.
- It's a method to test all the possible discrete combinations of the parameters involved.

Example

An application with simple list box with 10 elements (Let's say 0,1,2,3,4,5,6,7,8,9) along with a checkbox, radio button, Text Box and OK Button.

List Box - 0,1,2,3,4,5,6,7,8,9

Check Box - Checked or Unchecked

Radio Button - ON or OFF

Text Box - Anv Value between 1 and 100

```
List Box = 10
Check Box = 2
Radio Button = 2
Text Box = 100

Total Number of Test Cases using Cartesian Method : 10*2*2*100 = 4000
Total Number of Test Cases including Negative Cases will be > 4000
```

- We can consider the list box values as 0 and others as 0 is neither positive nor negative.
- Radio button and check box values cannot be reduced, so each one of them will have 2 combinations (ON or OFF).
- The Text box values can be reduced into three inputs (Valid Integer, Invalid Integer, Alpha-Special Character).

software testing technique, 2*2*2*3 = 24 (including negative cases).

Text Box	List Box	Check Box	Radio Button
Valid Int	0	check	ON
Valid Int	others	uncheck	OFF
Invalid Int	0	check	ON
Invalid Int	others	uncheck	OFF
AlphaSpecialCharacter	0	check	ON
AlphaSpecialCharacter	others	uncheck	OFF

Result of Pair-Wise Testing:

Exhaustive Combination results in > 4000 Test Cases.

Conventional Software Testing technique results in 24 Test Cases.

Pair Wise Software Testing technique results in just 6 Test Cases.

STATE BASED TECHNIQUE

- State-based testing, also known as state transition testing, is a software testing technique that verifies how a system responds to inputs and events. I
- different forms of the same situation, and according to the meaning, the state transition method does the same. Eg. ATM
- in the login function we use email and password, it gives a specific number of attempts to access the application, after crossing the maximum number of attempts it gets locked with an error message.

STATE	LOGIN	VALIDATION	REDIRECTED
S1	First Attempt	Invalid	S2
S2	Second Attempt	Invalid	S3
S3	Third Attempt	Invalid	S5
S4	Home Page		
S5	Error Page		

Parul®
University

DIGITAL LEARNING CONTENT

Quality Assurance : Quality Control, Assurance, Cost, Reviews, Software Quality Assurance, Approaches to SQA, Reliability, Quality Standards- ISO9000 And 9001

Parul® University

Quality Assurance : Quality Control

- Software Quality Assurance is a process that works parallel to Software Development. It focuses on improving the process of development of software so that problems can be prevented before they become major issues.
- SQA process Specific quality assurance and quality control tasks (including technical reviews and a multitiered testing strategy) Effective software engineering practice (methods and tools) .

SOFTWARE QUALITY FACTORS(McCall's Factor Model
)

Product operation factors − Correctness, Reliability, Efficiency, Integrity, Usability.

Product revision factors − Maintainability, Flexibility, Testability.

Product transition factors − Portability, Reusability, Interoperability.

Correctness

- ✓ The completeness of the output information, which can be affected by incomplete data.
- ✓ The up-to-dateness of the information defined as the time between the event and the response by the software system.
- ✓ The availability of the information.

Reliability

Reliability requirements deal with service failure. They determine the maximum allowed failure rate of the software system

Efficiency

It deals with the hardware resources needed to perform the different functions of the software system. It includes processing capabilities (given in MHz), its storage capacity (given in MB or GB) and the data communication capability (given in MBPS or GBPS).

Integrity

This factor deals with the software system security, that is, to prevent access to unauthorized persons.

Usability

Usability requirements deal with the staff resources needed to train a new employee and to operate the software system.

Product Revision Quality Factors

- Maintainability
- Flexibility
- Testability

Product Transition Software Quality Factor

- Portability
- Reusability
- Interoperability

Approaches to SQA

Preventive Approach:
Focusing on preventing defects by implementing robust processes like thorough requirement gathering, design reviews, and code standards enforcement.

Detective Approach:
Utilizing testing methodologies to identify defects at different stages of development through unit testing, integration testing, system testing, and acceptance testing.

Corrective Approach:
Addressing identified defects through bug fixes and updates, including regression testing to ensure new changes don't introduce new issues.

Specific SQA techniques and practices:

Static Analysis:
Examining code without executing it to identify potential issues like syntax errors, code smells, and security vulnerabilities.

Code Reviews:
Peer review process where developers examine each other's code to identify quality issues and best practices compliance.

Walkthroughs:
A structured review process where developers present their work to a group for feedback on design, functionality, and potential issues.

Requirement Reviews:
Assessing the completeness, clarity, and consistency of software requirements to ensure they are testable and aligned with user needs.

Functional Testing:
Verifying that the software performs its intended functions according to specifications.

Performance Testing:
Evaluating the system's responsiveness under various load conditions to identify performance bottlenecks.

Quality Standards- ISO9000 And 9001

- The **International standards organization (ISO)** is a standard which serves as a for contract between independent parties. It specifies guidelines for development of **quality system**.

ISO 9000

- Provides a framework for quality management systems
- Defines the vocabulary and principles of quality management
- Encourages the production of goods and services that meet global quality standards
- Includes guidelines for quality manuals, procedures, records, and data management

ISO 9001

- Outlines the requirements for establishing and maintaining a QMS
- Defines how to provide products and services that meet customer expectations and regulatory requirements
- Requires organizations to conduct internal audits
- Can help organizations improve processes, reduce errors, and increase efficiency

Parul® University

DIGITAL LEARNING CONTENT

How to get ISO 9000 Certification?

Application: Once an organization decided to go for ISO certification, it applies to the registrar for registration.

Pre-Assessment: During this stage, the registrar makes a rough assessment of the organization.

Document review and Adequacy of Audit: During this stage, the registrar reviews the document submitted by the organization and suggest an improvement.

Compliance Audit: During this stage, the registrar checks whether the organization has compiled the suggestion made by it during the review or not.

Registration: The Registrar awards the ISO certification after the successful completion of all the phases.

Continued Inspection: The registrar continued to monitor the organization time by time.

Features of ISO 9001 Requirements :

Document control –

All documents concerned with the development of a software product should be properly managed and controlled.

Planning –

Proper plans should be prepared and monitored.

Review –

For effectiveness and correctness all important documents across all phases should be independently checked and reviewed .

Testing –

The product should be tested against specification.

Organizational Aspects –

Various organizational aspects should be addressed e.g., management reporting of the quality team.

DIGITAL LEARNING CONTENT

Parul® University

www.paruluniversity.ac.in