Effective Test Case Writing

Learning Objective

* Introduction to Test Ware
* Testware - Scenario
* Testware - Test Case Definition
* Test Case - Elements
* Test Case - Attributes
* Test Case – Development process
* Test Case - Format
* Test Case Format - Step-by-step actions
* Test Case Format – Table / Matrix
* Test Case Format - Script for record/playback
* Test Case - Anatomy
* Test Case – Points to Improve Testability
* Test Case - Components
* Test Case - Guidelines For Writing
* Test Case – Common Mistakes
* Test Case - Template
* Test Case - Review
* Test Case - Check List
* Test Case - Writing effective/successful cases

**Learning Objectives:**

* Understand the definition of Testware , Test Case, its elements, Attributes,
* Understand the Test Case – Development process, its format, Anatomy
* Understand how to improve Testability
* Understand the Test Case – Components, Guidelines For Writing, Common Mistakes, Template
* Understand the importance of Test Case Reviews, and the Check List
* Questions & Answers

**Introduction To Testware**

The collection of major work products (deliverables) of testing are called as Testware.

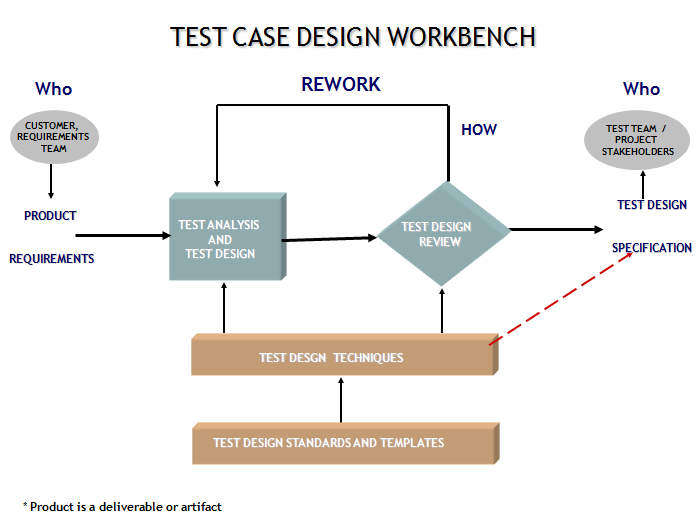
* Test Plans
* Test Architecture / Test Case Design
* Test Scenarios / Test cases
* Test Data
* Test Reports

Testware has its importance through out the lifecycle of the product.

Like the software, the testware needs to be saved and maintained

**Test Case Design Work Bench**

Test Case Design workbench describes the workflow and process involved in doing the test design activities.



**Testware - Test Scenario**

A test scenario summarizes the functionality to be tested without providing the steps to be followed to test the functionality mentioned in the Use case / Functional Specification

* Test scenario helps in keeping test focused.
* Test scenario’s are created at a very early stage in the development cycle.
* Creation of these scenarios requires considerable brain-storming and hence helps the tester in understanding the functionality and identifying the flows that need to be tested

E.g: The actor chooses to withdraw money from the savings bank account.

Inputs for Writing Test Scenario’s are :

* Business Requirement / Functional Specification Document / Use case
* Prototype
* High Level Design Document
* Test Plan Document

Test Scenarios are:

Based on realistic events that happen during the actual usage of the application

Hypothetical stories that are:

* Motivating - A stakeholder would push to fix a program that failed this test
* Involves a complex use of the program or environment or set of data
* Test Results are easy to evaluate
* Used to create a clear and simple flow of a complex problem or system
* Used to cover a specific functional area, business process, or use case
* Testable Business Scenarios, that can be translated to test cases during the Test Development phases

Why Test Scenarios?

Test scenarios perform the following functions:

* Bridge Requirement Analysis and Test Development
* Ease the process of understanding the product or application under test
* Aid in Requirement Elicitation
* Help in ensuring test coverage
* Help in identifying defects which might go undetected during execution of separate test cases because they tend to use the applications in the same sequence as it would be used in the real world

Test scenarios aids the following:

* Identifying Testable Requirements
* Identifying business scenarios
* Allowing ease of review by SMEs

Test Scenarios - Example

**Requirement:**

In an invoice approving system, if the maintenance engineer submits invoices with amount >= $500, it should be sent to manager for approval. All invoices with amount < $500 will get auto approved. Manager can also reject invoices if there are any mistakes, which can further be modified by the maintenance engineer and submitted again for approval.

**Scenarios:**

* User submits invoice, amt > $500 dollars, approver approves it
* User submits invoice, amt < $500 dollars, invoice gets auto approved
* User submits invoice, amt > $500, approver rejects it
* User submits invoice, amt > $500, approver rejects it, user submits after modifying to another value which is > $500, approver approves it
* User submits invoice, amt > $500, approver rejects it, user submits after modifying to another value which is < $500, invoice gets auto approved

In this example, auto approval may function correctly when the user submits the invoice for the first time with < $500, but may not function correctly when the invoice is rejected and then the user changes the amount to < $500.

**Types of Test Scenarios**

Test scenarios can be categorized based on the flow.

**Basic flow** - Scenarios aimed to ensure that the application attains it’s prime objective or function. This also covers the Scenarios aimed at showing how the application does not do what it is not supposed to do.

**Alternate flow** - Scenarios generated to achieve the goal by following an alternate workflow.

**Exception flow** - generated to test the application behavior under exception conditions.

Few Examples from a Leave Management System are as follows:

**A Basic flow:** Apply leave by mentioning the valid dates and reason in the Leave Management System.

**Another Basic flow:** Apply leave for the number of days that exceeds the available days (A check done on the basic flow with invalid data).

**Alternate:** During the apply leave process the user navigates to check the available leave balance and then returns to complete the apply leave process.

**Exception:** Apply for leave with a user who is serving notice period.

**Importance of Basic Scenario**

* Written to indicate likely threats which should be countered to ensure that systems have sufficient security, safety, and reliability.
* Basic scenarios can also include the scenarios which are aimed to show that the system does not do what it is not supposed to do
* The more the number of destructive test scenarios, the better the testing process
* A requirement coverage is perfect only when all kinds of Test Scenarios are written for the provided requirements

**Example1:**

Operation of lift functionality

**Basic Scenario with all possibilities:** Verification of the functionality of the lift upon pressing the Open button when the lift is moving.

**Example 2:**

Withdrawal of cash from ATM

**Basic Scenario with all the possibilities:** Withdrawal of an amount which exceeds the balance in the account.

Test Scenarios – Example

**Requirements for the leave management system:**

All employees should be able to apply for vacation leave, sick leave, compensatory off, and loss of pay. Vacation leave and sick leave can be availed only if there is sufficient leave balance accrued by the employee.

An employee is eligible to accrue one day of sick leave and two days of vacation leave per month. During the month of joining the employee is eligible for the following:

* Full month leave accrual if the employee has joined before 10th of the month,
* Half the leave accrual if the employee has joined between 10th and 20th of the month
* No leave accrual if joining date is beyond 20th of the month
* An employee should have worked for a minimum of six hours on a holiday, as per the working hours logged in the timesheet system, to avail a compensatory off.
* An employee cannot combine sick leave with any other type of leave except loss of pay. Vacation leave can be combined with compensatory off.
* A manager can approve or reject all types of leave.
* A user can cancel or modify the applied leave before approval by the manager.
* If an employee has availed loss of pay during a month, the salary of the employee for that month will be calculated based on the number of days the employee was on loss of pay.
* A user is not allowed to avail any kind of leave while serving the notice period.

**Following are the scenarios identified for the leave management system:**

SCN01: User applies for leave by providing valid dates and number of days

SCN02: User applies for leave with days which are more than the accrued leave

SCN03: User applies for leave and then cancels it before approval

SCN04: User applies for leave and then modifies it before approval

SCN05: User applies combining two types of leave which are valid

SCN06: User applies combining two types of leave which are not valid

SCN07: Manager approves leave applied by the employee

SCN08: Manager rejects leave applied by the employee

SCN09: User applies for loss of pay. Manager approves leave. Salary in payroll

system is calculated based on the number of days applied.

SCN10: User is allowed to apply for compensatory off based on the hours worked from the timesheet system.

SCN11: Leave accrual for a user who has joined during the second month of a year

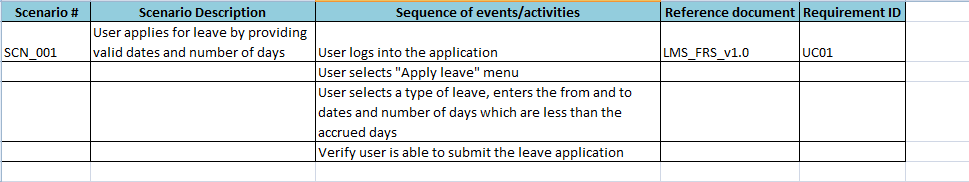
SCN12: User serving the notice period applies for leave

**Contents of Test Scenarios**

Following are the components of Test Scenarios:

* Scenario #
* Scenario Description
* Sequence of events/activities
* Reference document
* Requirement ID/Use case name

*Fig: Sample Test Scenario Template*



**Requirement Traceability Matrix**

A traceability matrix is a document, usually in the form of a table. It correlates any two base lined documents that require a many to many relationship to determine the completeness of the relationship. It is often used with high-level requirements and detailed requirements of the software product to the matching parts of high-level design, detailed design, test plan, test scenario, and test cases.

The tracking of the link between the requirement and the test scenario is frequently done using a traceability matrix.

To ease the creation of traceability matrices, relationships are added to the source documents for both backward traceability and forward traceability. In other words, when an item is changed in one base lined document, it is easy to see what needs to be changed in the other.

Requirement Traceability Matrix

* A traceability matrix is a document, usually in the form of a table. It correlates any two base lined documents that require a many to many relationship to determine the completeness of the relationship. It is often used with high-level requirements and detailed requirements of the software product to the matching parts of high-level design, detailed design, test plan, test scenario, and test cases.
* The tracking of the link between the requirement and the test scenario is frequently done using a traceability matrix.
* To ease the creation of traceability matrices, relationships are added to the source documents for both backward traceability and forward traceability. In other words, when an item is changed in one base lined document, it is easy to see what needs to be changed in the other.

**Requirement Traceability Matrix**

Sample Traceability Matrix:

* Forward Traceability Matrix:

|  |  |  |
| --- | --- | --- |
| **Requirement #** | **Requirement Description** | **Test Scenario #** |
| a | All employees should be able to apply for vacation leave, sick leave, compensatory off, and loss of pay. | SCN01 |
| b | An employee is eligible to accrue one day of sick leave and two days of vacation leave per month. | SCN01, SCN02, SCN11 |

**Backward Traceability Matrix:**

|  |  |  |
| --- | --- | --- |
| **Test Scenario #** | **Test Scenario Description** | **Requirement #** |
| SCN01 | User applies for leave by providing valid dates and number of days | a, b |
| SCN02 | User applies for leave with days which are more than the accrued leave | b |
| SCN11 | Leave accrual for a user who has joined during the second month of a year | b |

Requirement Traceability Matrix

Following are the benefits of a Requirement Traceability Matrix:

* Used to ensure design coverage. Makes sure every requirement is covered by at least one test scenario and test case (after test case development)
* Useful to track defects back to requirements, which would be useful in identification of regression test scenarios and test cases.
* Useful for impact analysis. If there is a change in a requirement, it is easy to identify the test scenarios/cases which have to be changed in line with the requirement change.
* Backward traceability matrix ensures that just enough test scenarios/cases are written and that they are not redundant

**System Testing and System Integration Test Scenarios**

Test scenarios should be identified for both system and system integration testing.

Definition of system and system integration testing (recap):

**System testing:** This is the testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing. It does not require any knowledge of the inner design of the code or logic.

**System Integration Testing (SIT):** This is the testing performed when two systems, generally presumed stable themselves, are integrated with one another. The goal of systems integration testing is to ensure that the data is crossing the boundary.