



# Testing Types

innovative • entrepreneurial • global

[www.utm.my](http://www.utm.my)



## Test Types – The Targets of Testing

- **Target For Testing** - A group of test activities aimed at verifying the software system (or a part of a system) based on a specific reason.
- **Test type** - *A group of test activities aimed at testing a component or system regarding one or more interrelated quality attributes. A test type is focused on a specific test objective, e.g.*
  - reliability test
  - usability test
  - Structure or architecture of the system/software
  - regression test*and may take place on one or more test levels or test phases.*

innovative • entrepreneurial • global

[www.utm.my](http://www.utm.my)



## Testing Type: A. Functional Testing

### According to ISTQB Glossary:

Testing conducted to evaluate the compliance of a component or system with functional requirements.

### Functional requirements (is per IEEE):

'A requirement that specifies a function that a system or system component must perform'.

A Requirement may exist as a text document and/or a model

Functional testing is also can be derived from requirements-based testing.



## Functional Testing

### Requirements based testing

- According to ISTQB glossary, is an approach to testing in which test cases are designed based on test objectives and test conditions derived from requirements, e.g., tests that exercise specific functions or probe non-functional attributes such as reliability or usability.
- Testing against requirements and specifications
- Test procedures and cases derived from:
  - detailed user requirements
  - system requirements functional specification
  - User documentation/instructions
  - High level System design

**Functional Testing - Example**

**Given a detail user requirements as follows:**

R01: A waiter shall be able to log into a tablet using their assigned username and password.

**Below is example, a design of test condition that can be derived:**

ID	Test Condition
TC01	Check/Validate waiter login using valid data.
TC02	Check/Validate waiter login using invalid data.

innovative • entrepreneurial • global      www.utm.my

**Functional Testing - Example**

**Given a requirements as follows:**

R01: A waiter shall be able to log into a tablet using their assigned username and password.

**The test procedure and test case can be derived as follows:**

ID	Test Condition	Test Procedure	Test Data	Expected Result	Actual Result	Pass?
TC01	Check waiter login with valid data.	<ol style="list-style-type: none"> <li>Open Restaurant Order Management System using browser.</li> <li>Enter User ID</li> <li>Enter User password</li> <li>Click submit</li> </ol>	User ID: roha Password: roha99	User successfully login	As expected	Pass
TC02	Check waiter login with invalid data.	<ol style="list-style-type: none"> <li>Open Restaurant Order Management System using browser.</li> <li>Enter User ID</li> <li>Enter User password</li> <li>Click submit</li> </ol>	User ID: roha Password: roha19	User unsuccessfully login	As expected	Pass

innovative • entrepreneurial • global      www.utm.my

**Functional System Testing**

**A. Requirements-based functional testing - techniques**

- Starts by using the most appropriate black-box testing techniques. Eg: How many valid and invalid will be enough to test?
- May support this with white-box techniques (e.g. menu structures, web page navigation)
- Risk based approach important (management, selection, prioritization, and use of testing activities and resources are based on corresponding risk types and risk levels).

ID	Test Scenario	Test Procedure	Test Data	Expected Result	Actual Result	Pass?
TC01	Check waiter login with valid data.	<ol style="list-style-type: none"> <li>Open Restaurant Order Management System using browser.</li> <li>Enter User ID</li> <li>Enter User password</li> <li>Click submit</li> </ol>	User ID: roha Password: roha99	User successfully login	As expected	Pass

innovative • entrepreneurial • global [www.utm.my](http://www.utm.my)

**Functional System Testing**

**B. Business Process based testing**

- Test procedures and cases derived from:
  - expected user profiles
  - Business scenarios
  - use cases
- Testing should reflect the business environment and processes in which the system will operate.
- Therefore, test cases should be based on real business processes.

- Based on this Use Case diagram, what functions to be tested?
- What test scenario to be tested?
- How many test cases to be produced?

innovative • entrepreneurial • global [www.utm.my](http://www.utm.my)

# Functional System Testing

## Requirements-based functional testing - Functionality

<b>Accuracy</b>	Provision of right or agreed results or effects
<b>Compliance</b>	Adhere to applicable standards, conventions, regulations or laws
<b>Auditability</b>	Ability to provide adequate and accurate audit data
<b>Suitability</b>	Presence and appropriateness of functions for specified tasks
<b>Interoperability</b>	The capability of the software product to interact with one or more specified components or systems.
<b>Security</b>	The functions (e.g. a firewall) relating to detection of threats, such as viruses, from malicious outsiders

# Functional System Testing

## Testing Pearl of Wisdom



- “**System testing is the most misunderstood and most difficult testing process**”

Myers - 2004



## Non-functional System Testing

- ▶ Definition
- ▶ Non-functional requirements
- ▶ Non-functional test types

innovative • entrepreneurial • global

[www.utm.my](http://www.utm.my)



## Non-functional Testing

**According to ISTQB Glossary:**  
 Testing conducted to evaluate the compliance of a component or system with non-functional requirements.

**Non-Functional requirements (is per IEEE):**  
 'A requirement that describes how the component or system will do what it is intended to do.'

Non-functional testing is also can be derived from requirements-based testing.

innovative • entrepreneurial • global

[www.utm.my](http://www.utm.my)



# Non-functional System Testing

## Definition

- *testing of those requirements that do not relate to functionality*



innovative • entrepreneurial • global

[www.utm.my](http://www.utm.my)



# Non-functional System Testing

## Non-functional requirements

- Emphasis on non-functional requirements:
  - Performance
  - Load
  - Data volumes
  - Storage
  - Recovery
  - Usability
  - Stress

innovative • entrepreneurial • global

[www.utm.my](http://www.utm.my)

**Non-Functional Testing - Example**

Given a non-functional requirements as follows:

R101: The server shall be capable of supporting no less than 200 concurrent connections from any combination of surface computers, tablets and displays.

The test procedure and test case can be derived as follows:

ID	Test Condition	Test Procedure	Test Data	Expected Result	Actual Result	Pass?
TC101	Test server capability with 199 users concurrently.	<ol style="list-style-type: none"> <li>Run simulator accessing RMOS system from tablet with 100 users.</li> <li>Run simulator accessing RMOS system from surface with 50 users.</li> <li>Run simulator accessing RMOS system from desktop computer with 49 users.</li> </ol>	User ID: roha Password: Roha99	System does not crash.	As expected	Pass

**Test Types – The Targets of Testing**

**Definitions**

- **Target For Testing** - A group of test activities aimed at verifying the software system (or a part of a system) based on a specific reason.
- **Test type** - *A group of test activities aimed at testing a component or system regarding one or more interrelated quality attributes. A test type is focused on a specific test objective, e.g.*
  - reliability test
  - usability test
  - Structure or architecture of the system/software
  - regression test*and may take place on one or more test levels or test phases.*


innovative • entrepreneurial • global      [www.utm.my](http://www.utm.my)



## Testing Type: C. Confirmation Testing (Re-Testing)

### **According to ISTQB Glossary:**

Dynamic testing conducted after fixing defects with the objective to confirm that failures caused by those defects do not occur anymore.

- Whenever a fault is detected and fixed then the software should be re-tested to show that the original fault has been fixed. This is known as Re-Testing.
- It is important that the test case is repeatable.
- In order to support this the test identifier should be included on the fault report.
- It is important that the environment and test data used are as close as possible as those used during the original test.



## Testing Type: D. Regression Testing

### **According to ISTQB Glossary:**

Testing of a previously tested component or system following modification to ensure that defects have not been introduced or have been uncovered in unchanged areas of the software, as a result of the changes made.

- If the test is re-run and passes you cannot necessarily say the fault has been resolved because you also need to ensure that the modifications have not caused unintended side-effects elsewhere and that the modified system still meets its requirements.
- Regression testing should be carried out :-
  - when the system is stable and the system or the environment changes
  - when testing bug-fix releases as part of the maintenance phase
- It should be applied at all Test Levels.
- It should be considered complete when agreed completion criteria for regression testing have been met.
- Regression test suites evolve over time and given that they are run frequently are ideal candidates for automation.

## Regression Testing

- Selecting suitable tests involves :-
  - knowledge of the bug fixes and how they affect the system
  - understanding the areas that have frequent faults
  - understanding which areas of the system have undergone the most recent changes
  - understanding the areas of the system which are most critical to the user
  - understanding the core features of the system which must function correctly.
- The effectiveness of a regression test suite can diminish over time for a number of reasons :-
  - tests are added for short term goals but not removed
  - tests become redundant due to functionality changes
  - test suite is not updated when major functionality changes are implemented
  - execution time becomes prohibitively high
  - maintenance of the test suite becomes prohibitively high.

## Regression Testing

- Reduction in effectiveness can be countered by :-
  - maintaining cross references between system features and their corresponding tests
  - monitoring the addition of tests to the suite
  - Periodic review and removal of redundant tests
  - review of the test suite when major enhancements are made to the system
  - evaluation of the effectiveness of the test suite using metrics.

**Regression Testing**



**Testing Pearl of Wisdom**

*"The probability of making an incorrect change is more than 50 %.  
Much of this is due to overconfidence and ineffective or nonexistent software change testing.  
We change just a couple of statements and believe we have not affected anything adversely.  
We execute one case that tests the path that was changed and tell ourselves that the change has been tested.  
**IS IT, THEN, ANY WONDER THAT WE EXPERIENCE SO MANY PROBLEMS?!"***

Hetzl 1998

innovative • entrepreneurial • global      [www.utm.my](http://www.utm.my)

**Maintenance Testing**

- ▶ What is Maintenance testing?
- ▶ Objectives of Maintenance testing
- ▶ Problems of Maintenance testing
- ▶ Concerns of Maintenance testing
- ▶ How can we test changes?

innovative • entrepreneurial • global      [www.utm.my](http://www.utm.my)

## What is Maintenance Testing?

- **Maintenance testing:** *Testing the changes to an operational system or the impact of a changed environment to an operational system*
- testing changes to a Live System
- Triggered by, for example,
  - **Modification**
    - software upgrades
    - Operating system changes
    - system tuning
    - emergency fixes
  - **software Retirement** (may necessitate data archiving tests)
  - **Migration**
    - System migration (including operational tests of new environment plus changed software)
    - database migration



## Objectives of Maintenance Testing

- Develop tests to detect problems prior to placing the change into production
- Correct problems identified in the live environment
- Test the completeness of needed training material
- Involve users in the testing of software changes



## Problems of Maintenance Testing

- All that is available is the source code (usually with poor internal documentation and no record of testing) – poor or missing specifications
- Program structure, global data structures, system interfaces and performance and/or design constraints are difficult to determine and frequently misinterpreted
- Baseline test plans and/or regression test packs often not updated



## Concerns of Maintenance Testing

- will the testing process be planned?
- will testing results be recorded?
- will new faults be introduced into the system?
- will system problems be detected during testing?
- how much regression testing is feasible?
- will training be considered?



## How can we test changes?

- Maintenance testing involves testing what has been changed (i.e. **Re-Testing**)
- It also, importantly, utilises Impact Analysis as a method for determining what **Regression testing** is required for the whole system
- Traceability of Testware to source documents essential for effective impact analysis (we cover this more in a later topic)
- Scope of Maintenance tests based on Risk assessment – including size of change and size of system
- Maintenance testing may involve one or more test levels and one or more test types