ST.1 Overall Testing Strategy

Testing requirements determined at the project planning stage (refer Software Project Management practice) serve as the major input towards establishing the overall testing strategy. In addition, the Software Life Cycle provides a reference point for determining dependencies in arriving at the overall test strategy detailed below.

Software testing involves the following common activities that are required to be carried out regardless of the kind of testing i.e. unit, integration, systems, user acceptance, being considered:

a) Test Plan - planning the general approach and the allocation of resources;

b) Test Design - detailing the general testing approach for the various kinds of tests to be conducted;

c) Test Case Specification - defining the inputs, predicted results and execution conditions for each case of testing;

d) Test Procedure - stating the sequence of actions to be carried out by test personnel;

e) Test Execution – reviewing the readiness of the test; conducting the test; signing off the completeness of the test;

f) Test Report - logging the results of executing a test procedure.

1 D _ _ A ST.1.2

During project planning, the Software Project Manager (SPM) determines the kinds of tests with associated tasks and deliverables that are required for the software project.

1 D _ _ A ST.1.3

The SPM determines where in the Software Life Cycle (SLC) planning activities will take place.

Table 1 below provides the most suitable points in the Software Life Cycle (SLC) to conduct test plans, test designs, test cases, test procedures, test execution and test reports.

Phase **Test Kind**	**Requir-ements (RQ)**	**Analys-is(AN)**	**Design (DG)**	**Coding (CO)**	**Testing (TS)**	**Accept-ance (AC)**	**Imple m-entatio n (IM)**
Unit			Plans Designs, Cases, Procedures	Execution & Reports.			
System -System -Integration		 Plans Plans	Designs, Cases, Procedures.		Execution & Reports		
User Acceptance	Plans		Designs, Cases, Procedures			Execution & Reports	

Table 1: Software Life Cycle/Testing Activities Relationship

1 D _ _ A ST.1.4

The SPM then determines the resources needed by the project for conducting software testing and when they will be required. These resources should include people, equipment and facilities.

1 D _ _ A ST.1.5

The SPM then incorporates into the project plan the details of software testing as determined above

a) Procedures for performing each type of activity within each kind of testing are provided below.
b) Refer to ST.A for a documentation template to prepare test specification.
c) Refer to ST.B for a sample test specification.

ST.2 Unit Testing (UT)

A unit of software is composed of one or more modules. Unit Testing refers to the process of testing modules that are defined at the design phase. These modules are assembled during unit testing to make the largest units defined at the design phase.

1 D _ _ A ST.2.1 UT Planning carried out in the design phase of the software life cycle (SLC)

The scope of the unit test is to verify the design and implementation of all components from the lowest level defined during design.

1 D _ _ A ST.2.1.1

The developers, mainly project managers or systems analysts, construct the plan to include scope, approach, resources and schedule of the intended unit tests.

1 D _ _ A ST.2.1.2

The developers review with and obtain the agreement from Software Project Manager (SPM) and/or operations staff before putting the UT plan into effect

1 D _ _ A ST.2.2 UT design carried out in the design phase of the software life cycle (SLC)

1 D _ _ A ST.2.2.1

The developers specify the details of the test approach for each software module defined in the design phase and

a) state the assembly sequence for constructing larger software units in accordance to the design

b) state the types of tests necessary for individual modules and units e.g. white-box, black-box

c) identify the associated test cases and procedures.

ST.H provides an example of how the assembly sequence could be determined.

ST.I provides a discussion of White-box and Black-box unit tests.

2 D _ _ A ST.2.2.2

The developer should specify unit testing tools that will be used to make the procedures more efficient and at the same time facilitate problem investigation.

2 D _ _ A ST.2.2.3

If a test of a software requirement is not possible, the software developer should use an alternative method of verification e.g. inspection, to qualify/quantify the acceptance.

1 D _ _ A ST.2.2.4

The developers review with and obtain the agreement from the Software Project Manager (SPM) and/or operations staff before finalizing the test design.

1 D _ _ A

ST.2.3 UT Case Specification carried out in the design phase of the software life cycle (SLC)

1 D _ _ A

ST.2.3.1

For each test case, the developers should specify the inputs, predicted results and execution conditions.

1 D _ _ A

ST.2.3.2

The developers review with and obtain the agreement from the Software Project Manager (SPM) and/or operations staff before finalizing the test case specifications.

2 D _ _ A

ST.2.4 UT Procedure Specification, carried out in the Design phase of the software life cycle (SLC)

2 D _ _ A

ST.2.4.1

For each test case, the developers should provide a step by step description of how to carry it out.

2 D _ _ A

ST.2.4.2

Wherever possible, test tools should be used to minimize the effort required to test the software.

2 D _ _ A

ST.2.4.3

The developers review with and obtain the agreement from the Software Project Manager (SPM) and/or operations staff before finalizing the test procedure specifications.

1 D _ _ A

ST.2.5 UT Execution and Reporting carried out in the coding phase of the software life cycle (SLC)

1 D _ _ A

ST.2.5.1

Software developers should conduct a readiness test review to ensure that test resources are available in accordance to test plan design and specifications before commencement.

1 D _ _ A

ST.2.5.2

Software developers conduct the test in accordance with the test specifications and records the results in:

a) Unit test result forms recording the date and the outcome of the test cases executed by the procedure and/or
b) Execution log-file

1 D _ _ A ST.2.5.3

Testers should reference any Software Problem Reports raised during the test when recording the results. Appendix 6.4 provides a form template for reporting software problems.

1 D _ _ A ST.2.5.4

Proper software change control procedures should be exercised for software being modified to resolve any problems encountered.

1 D _ _ A ST.2.5.5

If there are any software problems, testers should work closely with developers to plan the next round of tests and what test cases and procedures should be repeated after corrections have been made.

2 D _ _ A ST.2.5.6

When all software problems have been resolved, testers should signoff to indicate acceptance of the software from Unit Test.

ST.3 System Testing (ST)

The process of system testing is best described under two main areas:

- Integration and integration testing;
- System testing.

For large software projects, integration testing is required whereas for a small software project, eg, 3 to 4 programs, it may only be necessary to perform System Testing after Unit Testing.

Integration and Integration Testing (IT)

A software system is composed of one or more subsystems, which are composed of one or more units that are composed of one or more modules. Integration testing refers to the process of testing unit groups, each of which is a major component of the software design. During Integration Testing, the groups of units are integrated to build the system.

1 D _ _ A ST.3.1 IT planning carried out in the analysis phase of the software life cycle (SLC)

1 D _ _ A ST.3.1.1

The developers, mainly project managers or systems analysts, construct the plan to include scope, approach, resources and schedule of the intended integration tests.

1 D _ _ A ST.3.1.2

The developers review with and obtain the agreement from users and/or operations staff before putting the IT plan into effect.

1 D _ _ A ST.3.2 IT Design carried out in the design phase of the software life cycle (SLC)

1 D _ _ A ST.3.2.1

The developers specify the details of the test approach for each software unit group defined during software design;

2 D _ _ A ST.3.2.2

The developers state the integration sequence for constructing the system (ST.F provides an example of how the integration sequence could be determined);

2 D _ _ A ST.3.2.3

The developers state the types of tests necessary for individual unit groups e.g. white-box, black-box (ST.G provides a discussion of White-box and Black-box integration tests);

1 D _ _ A ST.3.2.4

The developers identify the associated test cases and procedures.

2 D _ _ A ST.3.2.5

The developer should specify integration testing tools that will be used to make the procedures more efficient to execute and/or for problem investigation.

2 D _ _ A ST.3.2.6

If a test of a software requirement is not possible, the software developer should use an alternative method of verification e.g. inspection, to qualify/quantify the acceptance.

1 D _ _ A ST.3.2.7

The developers review with and obtain the agreement from users and/or operations staff before finalizing the test design.

1 D _ _ A ST.3.3 IT case specification carried out in the design phase of the software life cycle (SLC)

1 D _ _ A ST.3.3.1

For each test case, the developers should specify the inputs, predicted results and execution conditions.

1 D _ _ A ST.3.3.2

The developers review with and obtain the agreement from users and/or operations staff before finalizing the test case specifications.

2 D _ _ A ST.3.4 IT procedure specification carried out in the design phase of the softwar life cycle (SLC)

2 D _ _ A ST.3.4.1

For each test case, the developers should provide a step by step description of how to carry it out.

2 D _ _ A ST.3.4.2

Wherever possible, test tools should be used to minimize the effort required to test the software.

2 D _ _ A ST.3.4.3

The developers review with and obtain the agreement from users and/or operations staff before finalizing the test procedure specifications.

1 D _ _ A ST.3.5 IT execution and reporting carried out in the testing phase of the software life cycle (SLC)

1 D _ _ A ST.3.5.1

Software developers should conduct a readiness test review to ensure that test resources are available in accordance with test plan, design and specifications before commencement.

1 D _ _ A ST.3.5.2

Software developers conduct the test in accordance with the test specifications and records the results in:

a) Integration test result forms recording the date and the outcome of the test cases executed by the procedure and/or

b) Execution log-file.

1 D _ _ A ST.3.5.3

Testers should reference any Software Problem Reports raised during the test when recording the results. Appendix 6.4 provides a form template for reporting software problems.

1 D _ _ A ST.3.5.4

As far as possible, testers should stick to the test plan in completing a full cycle of tests without being interrupted or distracted by other software developers who could be taking corrective action on problems uncovered during the test.

1 D _ _ A ST.3.5.5

Proper software change control procedures should be exercised for software being modified to resolve any problems encountered.

1 D _ _ A ST.3.5.6

If there are any software problems, testers should work closely with developers to plan the next round of tests and what test cases and procedures should be repeated after corrections have been made.

2 D _ _ A ST.3.5.7

When all software problems have been resolved, testers should signoff to indicate acceptance of the software from Integration Test (IT).

Systems Testing (ST)

1 D _ _ A ST.3.6 ST planning carried out in the analysis phase of the software life cycle (SLC)

1 D _ _ A ST.3.6.1

The developers, mainly project managers or systems analysts, construct the plan to include scope, approach, resources and schedule of the intended systems tests.

1 D _ _ A ST.3.6.2

The developers review with and obtain the agreement from users and/or operations staff before putting the ST plan into effect.

1 D _ _ A

ST.3.7 ST design carried out in the design phase of the software life cycle (SLC)

1 D _ _ A

ST.3.7.1

The developers specify the details of the test approach for each software requirement specified during analysis and

a) state the necessary types of tests required e.g. functional test, stress test etc;

b) identify the associated test cases and procedures.

2 D _ _ A

ST.3.7.2

The developers should specify a distinct system test approach to address each type of software requirement e.g.

a) Function Tests;
b) Performance Tests
c) Interface Tests;
d) Operations Tests;
e) Resource Tests;
f) Security Tests;
g) Portability Tests;
h) Reliability Tests;
i) Safety Tests;
j) Regression Tests and
k) Stress Tests

Appendix ST.E provides a discussion of the above system test approaches.

2 D _ _ A

ST.3.7.3

The developer should specify system testing tools that will be used to make the procedures more efficient to execute and/or for problem investigation

2 D _ _ A

ST.3.7.4

If a test of a software requirement is not possible, the software developer should use an alternative method of verification e.g. inspection, to qualify/quantify the acceptance.

1 D _ _ A

ST.3.7.5

The developers review with and obtain the agreement from users and/or operations staff before finalizing the test design.

1 D _ _ A ST.3.8 ST case specification carried out in the design phase of the software life cycle (SLC)

1 D _ _ A ST.3.8.1

For each test case, the developers should specify the inputs, predicted results and execution conditions.

1 D _ _ A ST.3.8.2

The developers review with and obtain the agreement from users and/or operations staff before finalizing the test case specifications.

2 D _ _ A ST.3.9 ST procedure specification carried out in the design phase of the software life cycle (SLC)

2 D _ _ A ST.3.9.1

For each test case, the developers should provide a step by step description of how to carry it out.

2 D _ _ A ST.3.9.2

Wherever possible, test tools should be used to minimize the effort required to test the software.

2 D _ _ A ST.3.9.3

The developers review with and obtain the agreement from users and/or operations staff before finalizing the test procedure specifications.

1 D _ _ A ST.3.10 Test execution and reporting carried out in the testing phase of the software life cycle (SLC)

1 D _ _ A ST.3.10.1

Software developers should conduct a readiness test review to ensure that test resources are available in accordance with test plan, design and specifications before commencement.

1 D _ _ A ST.3.10.2

Software developers conduct the test in accordance with the test specifications and records the results in:

a) System test result forms recording the date and the outcome of the test cases executed by the procedure and/or

b) Execution log-file.

1 D _ _ A ST.3.10.3

Testers should reference any Software Problem Reports raised during the test when recording the results. Appendix 6.4 provides a form template for reporting software problems.

1 D _ _ A ST.3.10.4

As far as possible, testers should stick to the test plan in completing a full cycle of tests without being interrupted or distracted by other software developers who could be taking corrective action on problems uncovered during the test.

1 D _ _ A ST.3.10.5

Proper software change control procedures should be exercised for software being modified to resolve any problems encountered.

1 D _ _ A ST.3.10.6

If there are any software problems, testers should work closely with developers to plan the next round of tests and what test cases and procedures should be repeated after corrections have been made.

2 D _ _ A ST.3.10.7

When all software problems have been resolved, testers should signoff to indicate acceptance that the software has completed System Test (ST).

ST.4 User Acceptance Testing (UAT)

1 D _ _ A ST.4.1 UAT Planning, carried out in the Requirements phase of the software life cycle (SLC).

1 D _ _ A ST.4.1.1

The initiators of the software project e.g. user management, provide the principles by which user acceptance testing will be conducted and the basis for acceptance of the system in production.

1 D _ _ A ST.4.1.2

The developers, mainly project managers or systems analysts, construct the plan to include scope, approach, resources and schedule addressing general issues in the following:

a) where will the acceptance tests be carried out?
b) who will attend?
c) who will carry them out?
d) are tests needed for all user requirements?
e) must any special test software be used?
f) what special documentation is required by users who will perform the test?
g) how long is the acceptance test program expected to last?

1 D _ _ A ST.4.1.3

The developers review with and obtain the agreement from users before putting the UAT plan into effect.

1 D _ _ A ST.4.2 UAT design carried out in the design phase of the software life cycle (SLC)

1 D _ _ A ST.4.2.1

The developers classify specific requirements in the User Requirements Document (URD) into:

a) Capability Requirements i.e. what the user can do with the software;

b) Constraint requirements which place restrictions on how the software can be built and operated and may predefine external interfaces or specify attributes such as adaptability, availability, portability and security.

1	D _ _	A	ST.4.2.2

The developers specify the details of the test approach for a user requirement or a combination of user requirements and

a) state the necessary types of tests required;

b) identify the associated test cases and procedures.

Note: Systems test cases that verify functional, performance and operational requirements may be reused to validate capability requirements. Systems test cases that verify compliance with requirements for interfaces, resources, security, portability, reliability, maintainability and safety may be reused to validate constraint requirements.

1	D _ _	A	ST.4.2.3

The developers review with and obtain the agreement from users before finalizing the test design.

1	D _ _	A	ST.4.3 UAT case specification carried out in the design phase of the software life cycle (SLC)

1	D _ _	A	ST.4.3.1

For each test case, the developers should specify the inputs, predicted results and execution conditions

1	D _ _	A	ST.4.3.2

The developers review with and obtain the agreement from users before finalizing the test case specifications.

2	D _ _	A	ST.4.4 UAT procedure specification carried out in the design phase of the software life cycle (SLC)

2	_ _ A		ST.4.4.1

For each test case, the developers should provide a step by step description of how to carry it out.

2	D _ _	A	ST.4.4.2

Wherever possible, test tools should be used to minimize the effort required to validate the software.

2 D _ _ A ST.4.4.3

The developers review and obtain the agreement from users before finalizing the test procedure specifications.

1 D _ _ A ST.4.5 UAT execution and reporting carried out in the acceptance phase of the software life cycle (SLC).

1 D _ _ A ST.4.5.1

Users should conduct a readiness test review to ensure that test resources are available in accordance with test plan, design and specifications before commencement.

1 D _ _ A ST.4.5.2

Users conduct the test in accordance with the test specifications and record the results in:

a) UAT result forms recording the date and the outcome of the test cases executed by following the procedure and/or

b) Execution log-file

1 D _ _ A ST.4.5.3

Users should reference any Software Problem Reports raised during the test when recording the results. ST.D provides a form template for reporting software problems.

1 D _ _ A ST.4.5.4

As far as possible, users should stick to the test plan in completing a full cycle of tests without being interrupted or distracted by developers who could be taking corrective action on problems uncovered during the test.

1 D _ _ A ST.4.5.5

Proper software change control procedures should be exercised for software being modified to resolve any problems encountered.

1 D _ _ A ST.4.5.6

If there are any software problems, users should work closely with developers to plan the next round of tests and what test cases and procedures should be repeated after corrections have been made.

1 D _ _ A ST.4.5.7

When all software problems have been resolved, the users or the user management representative should signoff to indicate acceptance of the software on the Software Transfer Document (STD). Appendix 6.3 provides a STD documentation template.

ST.5 Test Completion Activities

1 D _ _ A ST.5.1 Hand over software testing documentation

1 D _ _ A ST.5.1.1

When the system and software have been successfully tested to be free from any defects, the SPM shall seek agreement and approval from users (normally represented by user management and/or initiators), senior software managers and the software operations group (normally represented by software operations management) to implement the software into the operating environment.

1 D _ _ A ST.5.1.2

The SPM hands over the testing documentation (testing specifications and results) to the operations support and/or software support group to file for future reference.

2 D _ _ A ST.5.1.3

The SPM then executes plans to reallocate resources, equipment and facilities to other areas or projects of the software organization.

2 D _ _ A ST.5.1.4

The SPM obtains the signature(s) of the users (normally represented by user management and/or initiators) and the software operations group (normally represented by software operations management) to indicate successful completion of the software testing, and to commence software implementation activities.