INTERNATIONAL STANDARD

# ISO/ IEC/IEEE 29119-2

Second edition
2021-10



# Software and systems engineering — Software testing —

## Part 2: Test processes

*Ingénierie du logiciel et des systèmes — Essais du logiciel —*

*Partie 2: Processus des essais*

Reference number
ISO/IEC/IEEE 29119-2:2021(E)



ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Institute of Electrical and Electronics Engineers, Inc
3 Park Avenue, New York
NY 10016-5997, USA

Email: stds.ipr@ieee.org
Website: www.ieee.org

Published in Switzerland

# Contents

Page

# Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO/IEC documents should be noted. This document was drafted in accordance with the rules given in the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see https://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

ISO/IEC/IEEE 29119-2 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*, in cooperation with the Systems and Software Engineering Standards Committee of the IEEE Computer Society, under the Partner Standards Development Organization cooperation agreement between ISO and IEEE.

This second edition cancels and replaces the first edition (ISO/IEC/IEEE 29119-2:2013), which has been technically revised.

The main changes compared to the previous edition are as follows:

— The definition of the test design and implementation process (8.2) has been updated. In the first edition, this process was based on the use of test conditions. Feedback on use of the standard highlighted a problem with users' understanding of 'test conditions' and their use for deriving test cases. This second edition has replaced the use of 'test conditions' with 'test models'. Annex E provides more detail on this change.

A list of all parts in the ISO/IEC/IEEE 29119 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

# Introduction

The purpose of this document is to define a generic process model for software testing that can be used by any organization when performing any form of software testing. It comprises test process descriptions that define the software testing processes at the organizational level, test management level and dynamic test levels. Supporting informative diagrams describing the processes are also provided. ISO/IEC/IEEE 29119 (all parts) supports dynamic testing, functional and non-functional testing, manual and automated testing, and scripted and unscripted testing. The processes defined in this document can be used in conjunction with any software development lifecycle model. Each process is defined using the generic process template that is provided in ISO/IEC TR 24774, and covers the purpose, outcomes, activities, tasks and information items of each test process.

Testing is a key approach to risk treatment in software development. This document follows a risk-based approach to testing. Risk-based testing is a best-practice approach to strategizing and managing testing, as it allows testing to be prioritized and focused on the most important features and quality attributes.

This document uses the traditional concept of organizations and projects, but some organizations, especially those using an agile approach, do not organize their development in terms of projects; instead, they run product development based on more long-lasting product teams. Users of this document can substitute the term 'product' for 'project' where appropriate.

The concepts that support ISO/IEC/IEEE 29119 (all parts) are defined in ISO/IEC/IEEE 29119-1. Templates and examples of test documentation that are produced during the testing process are defined in ISO/IEC/IEEE 29119-3. Software test design techniques that can be used during testing are defined in ISO/IEC/IEEE 29119-4.

ISO/IEC/IEEE 29119 (all parts) aims to provide those responsible for software testing with the information required to manage and perform software testing in any organization.

Users of ISO/IEC/IEEE 12207 perform several activities and tasks which are related to software testing. Annex B provides a mapping for such users between the clauses and subclauses of ISO/IEC/IEEE 12207 and the clauses and subclauses of this document.

Users of ISO/IEC 17025 perform several activities and tasks which are related to software testing. Annex C provides a mapping for such users between the clauses and subclauses of ISO/IEC 17025 and the clauses and subclauses of this document.

Users of BS 7925-2 perform several activities and tasks which are related to software component testing. Annex D provides a mapping for such users between the clauses and subclauses of BS 7925-2 and the clauses and subclauses of this document.

# Software and systems engineering — Software testing —

# Part 2: Test processes



## 1 Scope

This document specifies test processes that can be used to govern, manage and implement software testing for any organization, project or testing activity. It comprises generic test process descriptions that define the software testing processes. Supporting informative diagrams describing the processes are also provided.

This document is applicable to testing in all software development lifecycle models.

This document is intended for, but not limited to, testers, test managers, developers and project managers, particularly those responsible for governing, managing and implementing software testing.

## 2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC/IEEE 12207, *Systems and software engineering — Software life cycle processes*

## 3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO, IEC and IEEE maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

— IEEE Standards Dictionary Online: available at https://ieeexplore.ieee.org/xpls/dictionary.jsp

NOTE For additional terms and definitions in the field of systems and software engineering, see ISO/IEC/IEEE 24765, which is published periodically as a "snapshot" of the SEVOCAB (Systems and software Engineering Vocabulary) database and is publicly accessible at https://www.computer.org/sevocab.

**3.1**
**actual results**
set of behaviours or conditions of a *test item* (3.42), or set of conditions of associated data or the *test environment* (3.34), observed as a result of *test execution* (3.38)

EXAMPLE Outputs to screen, outputs to hardware, changes to data, reports and communication messages sent.

**3.2**
**completion criteria**
conditions under which the *testing* (3.61) activities are considered complete

**3.3**
**dynamic testing**
*testing* (3.61) in which a *test item* (3.42) is evaluated by executing it

**3.4**
**expected results**
observable predicted behaviour of the *test item* (3.42) under specified conditions based on its specification or another source

**3.5**
**exploratory testing**
type of unscripted experience-based *testing* (3.61) in which the tester spontaneously designs and executes tests based on the tester's existing relevant knowledge, prior exploration of the *test item* (3.42) (including the results of previous tests), and heuristic "rules of thumb" regarding common software behaviours and types of failure

**3.6**
**incident**
anomalous or unexpected event, set of events, condition, or situation at any time during the life cycle of a project, product, service, or system

**3.7**
**incident report**
documentation of the occurrence, nature, and status of an *incident* (3.6)

Note 1 to entry: Incident reports are also known as anomaly reports, bug reports, defect reports, error reports, issues, problem reports and trouble reports, amongst other terms.

**3.8**
**organizational test practices**
documentation that expresses the recommended approaches or methods for the *testing* (3.61) to be performed within an organization, providing detail on how the testing is to be performed

Note 1 to entry: The organizational test practices is aligned with the *organizational test policy* (3.52).

Note 2 to entry: An organization can have more than one organizational test practices document to cover markedly different contexts, such one for mobile apps and one for safety critical systems.

Note 3 to entry: The organizational test practices can incorporate the context of the test policy where no separate test policy is available.

**3.9**
**organizational test process**
*test process* (3.55) for developing and managing *organizational test specifications* (3.10)

**3.10**
**organizational test specification**
document that provides information about *testing* (3.61) for an organization, i.e. information that is not project-specific

EXAMPLE The most common examples of organizational test specifications are the *organizational test policy* (3.52) and the *organizational test practices* (3.8).

**3.11**
**performance testing**
type of *testing* (3.61) conducted to evaluate the degree to which a *test item* (3.42) accomplishes its designated functions within given constraints of time and other resources

**3.12**
**product risk**
risk that a product can be defective in some specific aspect of its function, quality, or structure

**3.13**
**project risk**
risk related to the management of a project

EXAMPLE Lack of staffing, strict deadlines, changing requirements.

**3.14**
**regression testing**
*testing* (3.61) performed following modifications to a *test item* (3.42) or to its operational environment, to identify whether failures in unmodified parts of the test item occur

Note 1 to entry: Regression testing differs from *retesting* (3.15) in that it does not test that the modification works correctly, but that other parts of the system have not been accidentally affected by the change.

Note 2 to entry: The adequacy of a set of regression *test cases* (3.23) depends on the item under *test* (3.21) and on the modifications to that item or its operational environment.

**3.15**
**retesting**
confirmation testing
*testing* (3.61) performed to check that modifications made to correct a fault have successfully removed the fault

Note 1 to entry: When retesting is performed it is often complemented by *regression testing* (3.14), to ensure that other unmodified parts of the *test item* (3.42) have not been accidentally adversely affected by the modifications.

**3.16**
**risk-based testing**
*testing* (3.61) in which the management, selection, prioritisation, and use of testing activities and resources are consciously based on corresponding types and levels of analysed risk

**3.17**
**risk treatment**
process to eliminate risk or reduce it to a tolerable level

Note 1 to entry: Risk treatment measures can include avoiding, optimizing, mitigating, transferring, or retaining risk.

[SOURCE: ISO/IEC/IEEE 15026-1:2019, 3.3.11, modified — Note 1 to entry has been added.]

**3.18**
**scripted testing**
*testing* (3.61) performed based on a documented *test script* (3.54)

Note 1 to entry: This term normally applies to manually executed testing, rather than the execution of an automated script.

**3.19**
**security testing**
*test type* (3.60) conducted to evaluate the degree to which a *test item* (3.42), and associated data and information, are protected so that unauthorized persons or systems cannot use, read, or modify them, and authorized persons or systems are not denied access to them

**3.20**
**static testing**
*testing* (3.61) in which a *test item* (3.42) is examined against a set of quality or other criteria without the test item being executed

EXAMPLE Reviews, static analysis.

**3.21**
**test**
activity in which a system or component is executed under specified conditions, the results are observed or recorded, and an evaluation is made of some aspect of the system or component

**3.22**
**test basis**
information used as the basis for designing and implementing *test cases* (3.23)

Note 1 to entry: The test basis can take the form of documentation, such as a requirements specification, design specification, or module specification, but can also be an undocumented understanding of the required behaviour.

**3.23**
**test case**
set of preconditions, inputs and *expected results* (3.4), developed to drive the execution of a *test item* (3.42) to meet *test objectives* (3.49)

Note 1 to entry: A test case is the lowest level of test implementation documentation (i.e. test cases are not made up of test cases) for the *test level* (3.43) or *test type* (3.60) for which it is intended.

Note 2 to entry: Test case preconditions include the required state of the *test environment* (3.34), data (e.g. databases) used by the *test item* (3.42), and the test item itself.

Note 3 to entry: Inputs are the data information and actions, where applicable, used to drive *test execution* (3.38).

**3.24**
**test case specification**
documentation of a set of one or more *test cases* (3.23)

**3.25**
**test completion process**
*test management process* (3.45) for ensuring that useful test assets are made available for later use, *test environments* (3.34) are left in a satisfactory condition, and the results of *testing* (3.61) are recorded and communicated to relevant stakeholders

**3.26**
**test completion report**
test summary report
report that provides a summary of the *testing* (3.61) that was performed

**3.27**
**test condition**
testable aspect of a component or system, such as a function, transaction, feature, quality attribute, or structural element identified as a basis for *testing* (3.61)

Note 1 to entry: ISO/IEC/IEEE 29119 (all parts) does not use the concept of test conditions, but instead uses the concept of a *test model* (3.46) for test design. See Annex E for an explanation.

**3.28**
**test coverage**
degree, expressed as a percentage, to which specified *test coverage items* (3.29) have been exercised by a *test case* (3.23) or test cases

**3.29**
**test coverage item**
coverage item
measurable attribute of a *test item* (3.42) that is the focus of *testing* (3.61)

EXAMPLE Equivalence partitions, transitions between states, executable statements.

**3.30**
**test data**
data created or selected to satisfy the input requirements for executing one or more *test cases* (3.23)

Note 1 to entry: Test data can be stored within the *test item* (3.42) (e.g. in arrays or flat files), or can come from external sources, such as other systems, hardware devices, or human operators.

**3.31**
**test data readiness report**
document describing the status of each *test data* (3.30) requirement

**3.32**
**test design and implementation process**
*test process* (3.55) for deriving and specifying *test cases* (3.23) and *test procedures* (3.53)

**3.33**
**test design technique**
test technique
procedure used to create or select a *test model* (3.46), identify *test coverage items* (3.29) and derive corresponding *test cases* (3.23)

EXAMPLE Equivalence partitioning, boundary value analysis, decision table *testing* (3.61), branch testing.

Note 1 to entry: The test design technique is typically used to achieve a required level of *test coverage* (3.28).

Note 2 to entry: Some test practices, such as *exploratory testing* (3.5) or model-based testing, are sometimes referred to as "test techniques". Following the definition in ISO/IEC/IEEE 29119 (all parts), they are not test design techniques as they are not themselves providing a way to create test cases, but instead use test design techniques to achieve that.

**3.34**
**test environment**
environment containing facilities, hardware, software, firmware, procedures, needed to conduct a *test* (3.21)

Note 1 to entry: A test environment can contain multiple environments to accommodate specific *test levels* (3.43) or types (e.g. a unit test environment, a performance test environment).

Note 2 to entry: A test environment can comprise several interconnected systems or virtual environments.

**3.35**
**test environment readiness report**
document that describes the status of the *test environment* (3.34)

Note 1 to entry: This can list the status of each of the *test environment requirements* (3.36).

**3.36**
**test environment requirements**
description of the necessary properties of the *test environment* (3.34)

Note 1 to entry: All or parts of the test environment requirements can reference where the information can be found, e.g. in the *organizational test practices* (3.8) document, *test plan* (3.50), and *test specification* (3.57).

**3.37**
**test environment and data management process**
*test process* (3.55) for establishing and maintaining a required *test environment* (3.34) and corresponding *test data* (3.30)

**3.38**
**test execution**
process of running a *test* (3.21) on the *test item* (3.42), producing *actual results* (3.1)

**3.39**
**test execution log**
record of the execution of one or more *test procedures* (3.53)

**3.40**
**test execution process**
dynamic *test process* (3.55) for executing *test procedures* (3.53) created in the *test design and implementation process* (3.32) in the prepared *test environment* (3.34) and recording the results

**3.41**
**test incident reporting process**
dynamic *test process* (3.55) for reporting *incidents* (3.6) requiring further action that were identified during the *test execution process* (3.40) to the relevant stakeholders

**3.42**
**test item**
test object
work product to be tested

EXAMPLE Software component, system, requirements document, design specification, user guide.

**3.43**
**test level**
one of a sequence of test stages, each of which is typically associated with the achievement of particular objectives and used to treat particular risks

EXAMPLE The following are common test levels, listed sequentially: unit/component testing, integration testing, system testing, system integration testing, acceptance testing.

Note 1 to entry: It is not always necessary for a *test item* (3.42) to be tested at all test levels, but the sequence of test levels generally stays the same.

Note 2 to entry: Typical objectives can include consideration of basic functionality for unit/component testing, interaction between integrated components for integration testing, acceptability to end users for acceptance testing.

**3.44**
**test management**
planning, scheduling, estimating, monitoring, reporting, control and completion of test activities

**3.45**
**test management process**
process used to coordinate, monitor and control *testing* (3.61)

EXAMPLE *Test strategy and planning process* (3.51), *test monitoring and control process* (3.48), *test completion process* (3.25).

**3.46**
**test model**
representation of the *test item* (3.42), which allows the testing to be focused on particular characteristics or qualities

EXAMPLE Requirements statements, equivalence partitions, state transition diagram, use case description, decision table, input syntax description, source code, control flow graph, parameters and values, classification tree, natural language.

Note 1 to entry: The test model and the required *test coverage* (3.28) are used to identify *test coverage items* (3.29).

Note 2 to entry: A separate test model can be required for each different type of required test coverage included in the test *completion criteria* (3.2).

Note 3 to entry: A test model can include one or more *test conditions* (3.27).

Note 4 to entry: Test models are commonly used to support test design (e.g. they are used to support test design in ISO/IEC/IEEE 29119-4, and they are used in model-based testing). Other types of models exist to support other aspects of testing, such as *test environment* (3.34) models, test maturity models and test architecture models.

**3.47**
**test model specification**
document specifying the *test model* (3.46)

**3.48**
**test monitoring and control process**
*test management process* (3.45) for ensuring that *testing* (3.61) is performed in line with a *test plan* (3.50) and with *organizational test specifications* (3.10)

**3.49**
**test objective**
reason for performing *testing* (3.61)

EXAMPLE Checking for correct implementation, identification of defects, measuring quality.

**3.50**
**test plan**
detailed description of *test objectives* (3.49) to be achieved and the means and schedule for achieving them, organized to coordinate *testing* (3.61) activities for some *test item* (3.42) or set of test items

Note 1 to entry: A project can have more than one test plan, for example there can be a project test plan (also known as a master test plan) that encompasses all testing activities on the project; further detail of particular *test* (3.21) activities can be defined in separate test plans (e.g. a system test plan or a performance test plan).

Note 2 to entry: A test plan is typically a written document, although other formats can be possible as defined locally within an organization or project.

Note 3 to entry: Test plans can also be written for non-project activities, for example a maintenance test plan.

**3.51**
**test strategy and planning process**
*test management process* (3.45) used to design the *test strategy* (3.59), complete test planning and create and maintain *test plans* (3.50)

**3.52**
**test policy**
**organizational test policy**
executive-level document that describes the purpose, goals, principles and scope of *testing* (3.61) within an organization

Note 1 to entry: The test policy defines what testing is performed and what it is expected to achieve but does not detail how testing is to be performed.

Note 2 to entry: The test policy can provide a framework for establishing, reviewing and continually improving the organization's testing.

**3.53**
**test procedure**
sequence of *test cases* (3.23) in execution order, with any associated actions required to set up preconditions and perform wrap-up activities post execution

**3.54**
**test procedure specification**
**test script**
document specifying one or more *test procedures* (3.53)

**3.55**
**test process**
set of *testing* (3.61) activities performed to achieve a *test objective* (3.49)

Note 1 to entry: The test process for a particular project can consist of multiple *test levels* (3.43) and *test types* (3.60).

**3.56**
**test result**
indication of whether a specific *test case* (3.23) has passed or failed, i.e. if the *actual results* (3.1) corresponds to the *expected results* (3.4) or if deviations were observed

**3.57**
**test specification**
complete documentation of the test design, *test cases* (3.23) and *test procedures* (3.53) for a specific *test item* (3.42)

Note 1 to entry: A test specification can be detailed in one document, in a set of documents, or in other ways, for example in a mixture of documents and database entries.

**3.58**
**test status report**
report that provides information about the status of the *testing* (3.61) that is being performed in a specified reporting period

**3.59**
**test strategy**
part of the *test plan* (3.50) that describes the approach to *testing* (3.61) for a specific project, *test level* (3.43) or *test type* (3.60)

Note 1 to entry: The test strategy usually describes some or all of the following: the test levels and test types to be implemented; the *retesting* (3.15) and *regression testing* (3.14) to be employed; the *test design techniques* (3.33) and corresponding test *completion criteria* (3.2) to be used; *test data* (3.30); *test environment* (3.34) and testing tool requirements; and expectations for test deliverables.

**3.60**
**test type**
*testing* (3.61) that is focused on specific quality characteristics

EXAMPLE *Security testing* (3.19), functional testing, usability testing, and *performance testing* (3.11).

Note 1 to entry: A test type can be performed at a single *test level* (3.43) or across several test levels (e.g. performance testing performed at a unit test level and at a system test level).

**3.61**
**testing**
set of activities conducted to facilitate discovery and/or evaluation of properties of one or more *test items* (3.42)

Note 1 to entry: Testing activities include planning, preparation, execution, reporting, and management activities, insofar as they are directed towards testing.



# 4 Conformance

## 4.1 Intended usage

### 4.1.1 General

The requirements in this document are contained in Clauses 4, 6, 7 and 8. This document provides requirements for a number of test processes suitable for use during the life cycle of a software system

or product. It is recognized that particular projects or organizations may not need to use all of the processes provided by this document. Therefore, implementation of this document typically involves selecting and declaring a set of processes suitable to the organization or project. There are two ways that an implementation can be claimed to conform to the provisions of this document – full conformance and tailored conformance.

The organization shall assert whether it is claiming full or tailored conformance to this document:

There are two criteria for claiming full conformance. Achieving either criterion suffices for conformance, although the chosen criterion (or criteria) shall be stated in the claim. Claiming "full conformance to tasks" asserts that all of the requirements of the activities and tasks of the declared set of processes are achieved. Alternatively, claiming "full conformance to outcomes" asserts that all of the required outcomes of the declared set of processes are achieved. Full conformance to outcomes permits greater freedom in the implementation of conforming processes and can be useful for implementing processes to be used in the context of an innovative life cycle model.

NOTE 1 Options for conformance are provided for needed flexibility in the application of this document. Each process has a set of objectives (phrased as "outcomes") and a set of activities and tasks that represent one way to achieve the objectives.

NOTE 2 Users who implement the activities and tasks of the declared set of processes can assert full conformance to tasks of the selected processes. Some users, however, can have innovative process variants that achieve the objectives (i.e., the outcomes) of the declared set of processes without implementing all of the activities and tasks. These users can assert full conformance to the outcomes of the declared set of processes. The two criteria — conformance to task and conformance to outcome — are necessarily not equivalent since specific performance of activities and tasks can require, in some cases, a higher level of capability than just the achievement of outcomes.

NOTE 3 When this document is used to help develop an agreement between an acquirer and a supplier, clauses of this document can be selected for incorporation in the agreement with or without modification. In this case, it is more appropriate for the acquirer and supplier to claim compliance with the agreement than conformance with this document.

NOTE 4 An organization (for example, national, industrial association, company) imposing this document, as a condition of trade, can specify and make public the minimum set of required processes, outcomes, activities, and tasks, which constitute suppliers' compliance with the conditions of trade.

NOTE 5 Requirements of this document are marked by the use of the verb "shall". Recommendations are marked by the use of the verb "should". Permissions are marked by the use of the verb "may". However, despite the verb that is used, the requirements for conformance are selected as described previously.

Organizations and projects that perform both scripted and exploratory testing can find that meeting all the requirements of some processes (e.g. Test Design and Implementation) is difficult when performing exploratory testing. In such cases, if the scripted testing meets all the requirements of the process, the organization or project can claim full conformance for the scripted testing (i.e. testers are allowed to execute additional informal tests that do not meet the full set of requirements defined in this document). Otherwise, if the organization's testing does not meet the requirements of specific clauses, then they can claim tailored conformance as per 4.1.3.

### 4.1.2 Full conformance

#### 4.1.2.1 Full conformance to outcomes

A claim of full conformance declares the set of processes for which conformance is claimed. Full conformance to outcomes is achieved by demonstrating that all of the outcomes of the declared set of processes have been achieved. In this situation, the provisions for activities and tasks of the declared set of processes are guidance rather than requirements, regardless of the verb form that is used in the provision. One intended use of this document is to facilitate process assessment and improvement. For this purpose, the objectives of each process are written in the form of 'outcomes' compatible with the provisions of ISO/IEC 33002. ISO/IEC 33002 provides for the assessment of the processes of this document, providing a basis for improvement. Users intending process assessment and improvement

may use the process outcomes written in this document as the “process reference model” required by ISO/IEC 33002.

#### 4.1.2.2 Full conformance to tasks

A claim of full conformance declares the set of processes for which conformance is claimed. Full conformance to tasks is achieved by demonstrating that all of the requirements of the activities and tasks of the declared set of processes have been achieved. In this situation, the provisions for the outcomes of the declared set of processes are guidance rather than requirements, regardless of the verb form that is used in the provision.

NOTE A claim of full conformance to tasks can be appropriate in contractual situations where an acquirer or a regulator requires detailed understanding of the suppliers’ processes.

### 4.1.3 Tailored conformance

When this document is used as a basis for establishing a set of processes that do not qualify for full conformance, the subset of processes, for which tailored conformance is claimed, is recorded. Tailored conformance is achieved by demonstrating that all of the requirements (i.e. "shall" statements) for the recorded subset of processes have been satisfied.

Where tailoring occurs, justification shall be provided (either directly or by reference), whenever a process defined in Clauses 6 to 8 is not followed. All tailoring decisions shall be recorded with their rationale, including the consideration of any applicable risks. Tailoring decisions shall be agreed by the relevant stakeholders.

EXAMPLE Where organizations follow information item management processes in standards such as ISO 15489-1 or ISO 9001 or use similar internal organizational processes, they can decide to use those processes in place of the information item management tasks defined in this document.

## 5 Multi-layer test process model

This document groups the testing activities that may be performed during the life cycle of a software system into three process groups, as shown in Figure 1. Each of the processes within those groups is described in terms of its purpose and desired outcomes and activities and tasks which need to be performed are listed.

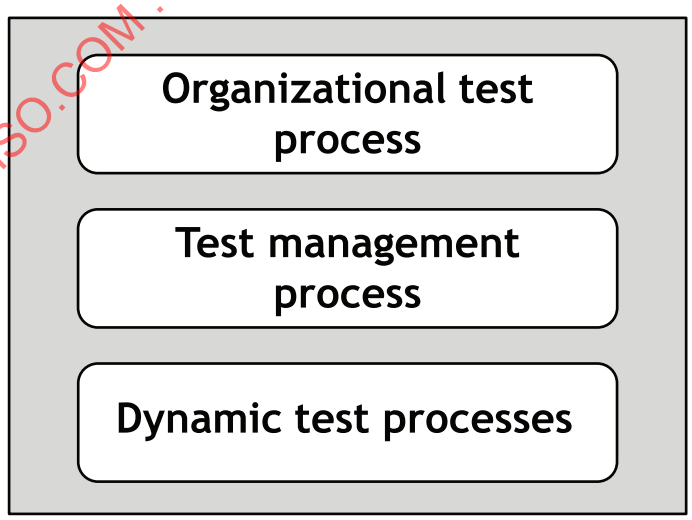



**Figure 1 — The multi-layer testing processes**

The goal of each layer is as follows:

a) Organizational test process (Clause 6)

   1) Defining a process for the creation and maintenance of organizational test specifications, such as organizational test policies, practices, processes, procedures and other assets.

b) Test management processes (Clause 7)

   1) Defining processes that cover the management of testing for a whole project or any test level (e.g. system testing) or test type (e.g. performance testing) within a project (e.g. project test management, system test management, performance test management).

   2) The test management processes are:

      i) test strategy and planning process (7.2);

      ii) test monitoring and control process (7.3);

      iii) test completion process (7.4).

c) Dynamic test processes (Clause 8)

   1) Defining generic processes for performing dynamic testing. Dynamic testing may be performed at a particular test level (e.g. unit, integration, system, and acceptance) or for a particular test type (e.g. performance testing, security testing, and functional testing) within a project.

   2) The dynamic test processes are:

      i) test design and implementation process (8.2);

      ii) test environment and data management process (8.3);

      iii) test execution process (8.4); and

      iv) test incident reporting process (8.5).

NOTE In IEEE 1012, the dynamic test process is referred to as "the Test Process".

The layers of the test process model comprise varying numbers of test process, as shown in Figure 2.

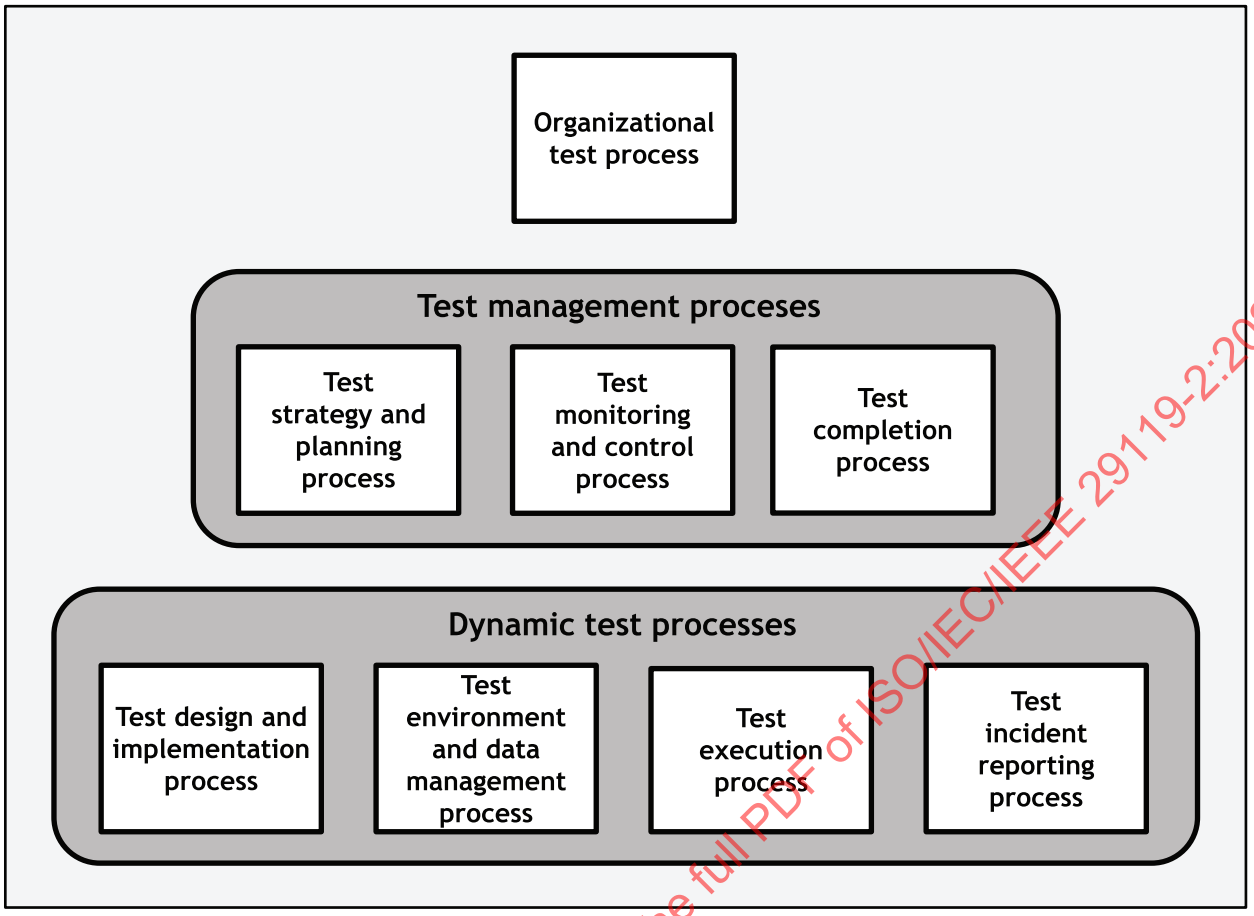



**Figure 2 — The multi-layer model showing all test processes**



# 6 Organizational test process

## 6.1 General

The organizational test process is used to develop and manage organizational test specifications. These specifications typically apply to testing across the whole organization (i.e. they are not project-based). The organizational test policy and organizational test practices document are examples of organizational test specifications. The organizational test process is generic and can be used to develop and manage other non-project specific test documents, such as a programme test strategy that applies to a number of related projects.

The organizational test policy is an executive-level document that describes the purpose, goals, and overall scope of testing within the organization. It also establishes organizational testing practices and provides a framework for establishing, reviewing and continually improving the organization's test policy, test practices and approach to project test management.

The organizational test practices document is a detailed, technical document that defines how testing is carried out within the organization. It is a generic document that provides guidelines for a number of projects in the organization and is not project specific.

In Figure 3 the organizational test process is shown in a typical situation where it has been applied to create and maintain both an organization's test policy and test strategy. As Figure 3 illustrates, the two instances of the organization-level processes communicate with each other. The organizational test practices document needs to align with the organizational test policy and feedback from this activity is provided back to the test policy for possible process improvement. Similarly, the test management processes being used on each of the projects within the organization need to align with

the organizational test practices document (and test policy) and feedback from the management of these projects is used to improve the organizational test process which formulates and maintains the organizational test specifications.

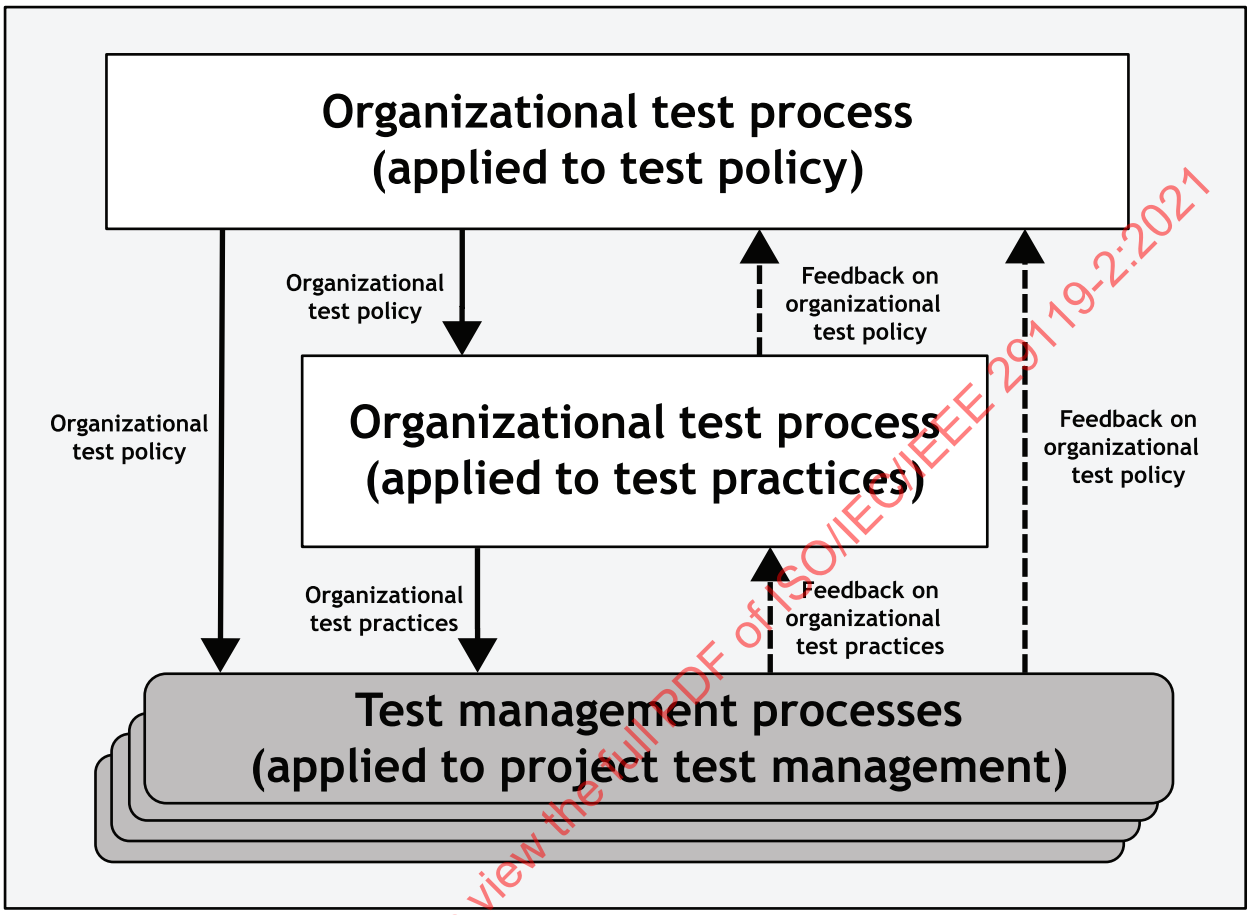



**Figure 3 — Example organizational test process implementation**

## 6.2 Organizational test process

### 6.2.1 Overview

The organizational test process comprises activities for the creation, review and maintenance of organizational test specifications. It also covers the monitoring of organizational compliance with them (see Figure 4).

Typical inputs to this process include:

— views of primary stakeholders;

— knowledge of the current test practices within the organization;

— organization's mission statement;

— IT policy;

— IT project management policy;

— quality policy;

— organizational test policy;

— organizational test practices;

— feedback on test policy;

— feedback on test practices;

— typical test plans from the organization; and

— industry and/or government standards.

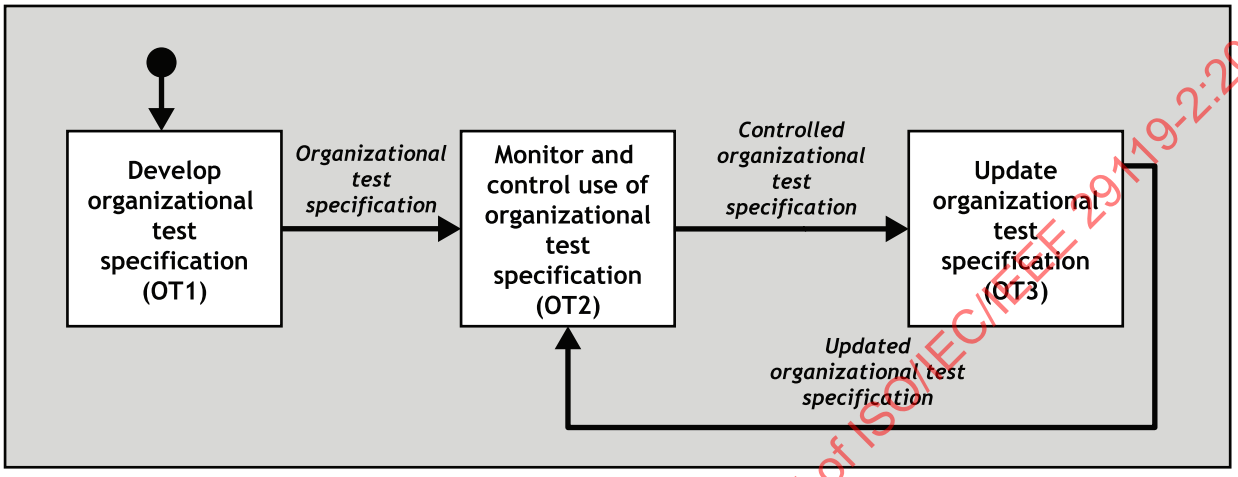



**Figure 4 — Organizational test process**

### 6.2.2 Purpose

The purpose of the organizational test process is to develop, monitor conformance and maintain organizational test specifications, such as the organizational test policy and organizational test practices document.

### 6.2.3 Outcomes

As a result of the successful implementation of the organizational test process:

a) The requirements for organizational test specifications are identified.

b) The organizational test specifications are developed.

c) The organizational test specifications are agreed to by stakeholders.

d) The organizational test specifications are made accessible.

e) Conformance to the organizational test specifications is monitored.

f) Updates to organizational test specifications are agreed to by stakeholders.

g) Updates to the organizational test specifications are made.

### 6.2.4 Activities and tasks

#### 6.2.4.1 General

The person responsible for organizational test specifications will implement the activities and tasks specified in 6.2.4.2 to 6.2.4.4 in accordance with applicable organization policies and procedures with respect to the organizational test process.

#### 6.2.4.2 Develop organizational test specification (OT1)

This activity consists of the following tasks:

a) Requirements for the organizational test specifications shall be identified from the current testing practices within the organization, from stakeholders and/or shall be developed by other means.

NOTE This can be achieved by analysing relevant source documents, through workshops, interviews or other suitable means.

b) The organizational test specification requirements shall be used to create the organizational test specification.

c) Approval on the content of the organizational test specification shall be obtained from the stakeholders.

d) The availability of the organization test specification shall be communicated to the stakeholders in the organization.

#### 6.2.4.3 Monitor and control use of organizational test specification (OT2)

This activity consists of the following tasks:

a) Usage of the organizational test specification shall be monitored to determine whether it is being used effectively within the organization.

b) Appropriate actions shall be taken to encourage alignment of stakeholders to the organizational test specification.

#### 6.2.4.4 Update organizational test specification (OT3)

This activity consists of the following tasks:

a) Feedback on use of the organizational test specification should be reviewed.

b) The effectiveness of the use and management of the organizational test specification should be considered and any feedback and changes to improve its effectiveness should be determined and approved.

NOTE This can be achieved by reviewing feedback, through workshops, interviews and other suitable means.

c) Where changes to the organizational test specification have been identified and approved, these changes shall be implemented.

d) All changes to the organizational test specification shall be communicated throughout the organization including to all stakeholders.

### 6.2.5 Information items

As a result of carrying out this process, the following information item shall be produced: organizational test specification.

EXAMPLE Organizational test policy, organizational test practices document.

# 7 Test management processes

## 7.1 General

There are three test management processes:

a) test strategy and planning;

b) test monitoring and control; and

c) test completion.

These generic test management processes may be applied at the project level (project test management), for test management at different test levels (e.g. system test management, acceptance test management) and for managing various test types (e.g. performance test management, usability test management).

When applied at the project test management level, these test management processes are used to manage the testing for the whole project, based on a project test plan. For many projects, each of the individual test levels and types will also require the test management processes to be applied to its management separately; these will typically be based on separate test plans, such as the system test plan, reliability test plan and acceptance test plan.

Figure 5 illustrates the relationships between the three test management processes, and how they interact with the organizational test process, other applications of the test management processes (e.g. if in a large project a high-level test management process is applied on lower level test management processes specific for test levels or test types) and the dynamic test processes.

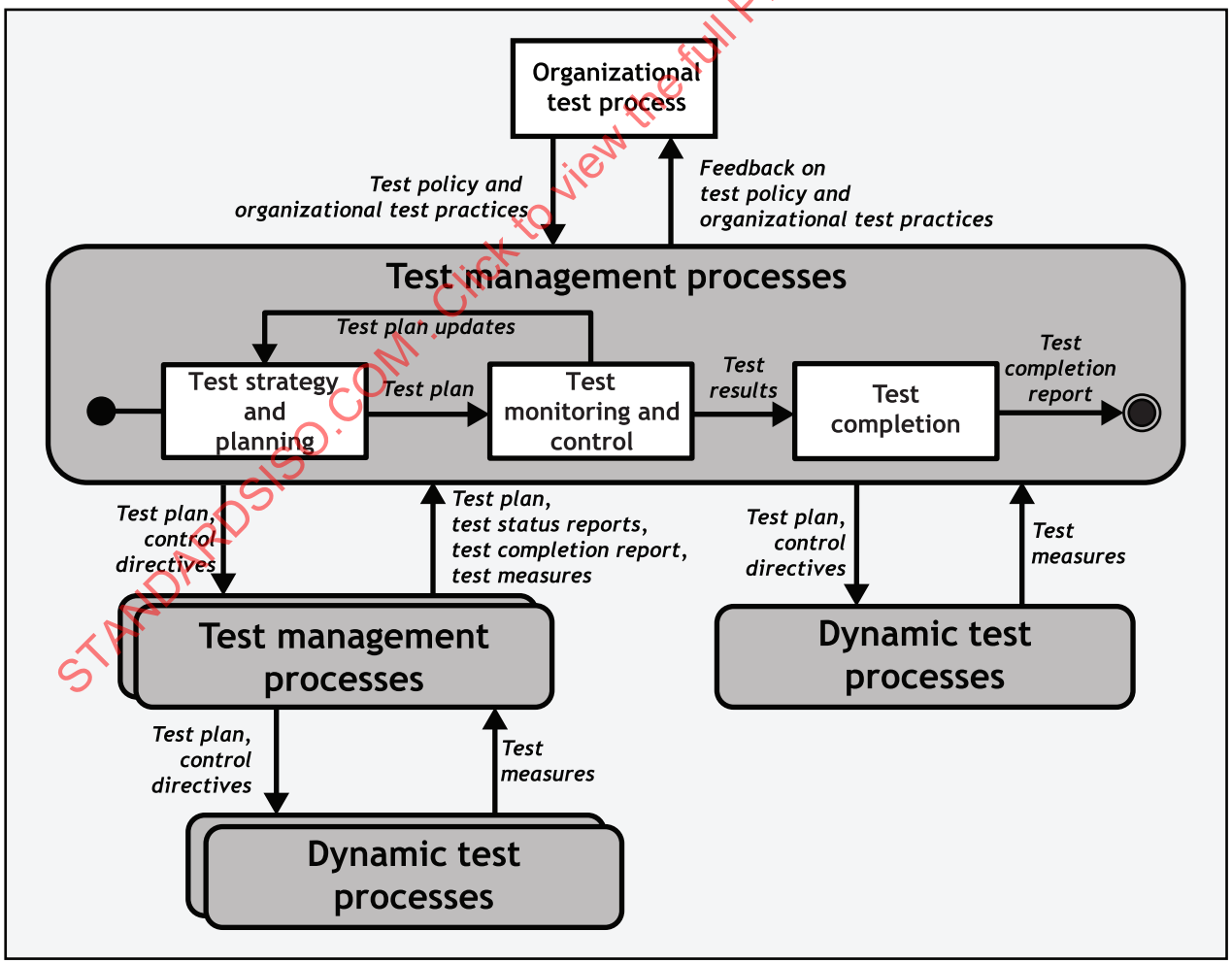



**Figure 5 — Example test management process relationships**

Test management processes need to align with the outputs of the organizational test process, such as the organizational test policy and organizational test practices document. Based on the practical implementation of these outputs, the test management processes may produce feedback on the organizational test process.

## 7.2 Test strategy and planning process

### 7.2.1 Overview

The test strategy and planning process is used to specify the purpose, scope, risks, typical test activities, and required resources for testing. Depending on where in the project this process is implemented, the results of designing the test strategy and test planning may be recorded in a project test plan or a test plan for a specific level, such as a system test plan, or a test plan for a specific type of testing, such as a performance test plan.

Typical inputs to this process include:

— organizational test policy;

— organizational test practices;

— regulatory standards;

— project test plan (if planning testing for a specific phase or type within a project);

— incident reports;

— project management plan;

— applicable product documentation (e.g. system requirements, test item specification);

— software development plan;

— project and product risks; and

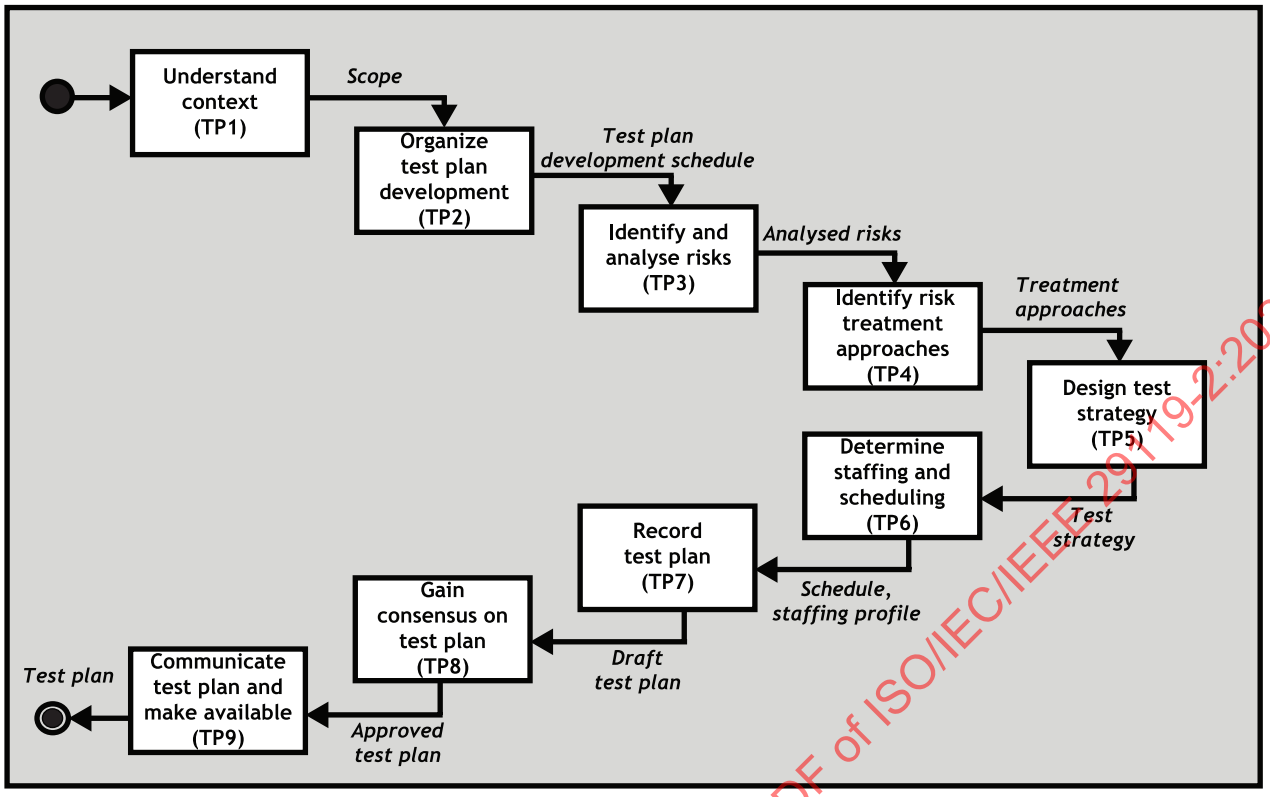
— test plan updates.

To create a test plan, the activities shown in Figure 6 shall be performed. As content for the test plan becomes available through performing the defined activities, a draft test plan will gradually be elaborated until the complete test plan is recorded. Due to the iterative nature of the process, a number of the activities shown in Figure 6 may need to be revisited before the complete test plan can be made available. Most typically the activities TP3, TP4, TP5 and TP6 will need to be performed iteratively in order to achieve an acceptable test plan.

ISO/IEC/IEEE 29119 (all parts) recommends embedding the test strategy within the test plan, although conformance with this document does not require this. By designing the test strategy and then considering test planning elements, such as available resources, it allows the optimal test strategy to be balanced against the available budget, schedule and resources for testing. The test strategy is based on the perceived risks, which are explicitly managed through the activities TP4 and TP5.

NOTE 1 ISO/IEC/IEEE 16085 provides a more detailed set of risk management activities and tasks. This risk management process is aligned with ISO 31000.

NOTE 2 The previous edition of this document used the term risk mitigation for addressing risks. In this edition, the term risk treatment has been used to include risk mitigation and other means of addressing risks, and to align with the terminology in ISO/IEC/IEEE 16085 and ISO 31000.




NOTE The process is shown as purely sequential, but in practice it can be carried out iteratively, with some activities being revisited.

**Figure 6 — Test strategy and planning process**

During the course of testing, the test plan may need to be modified in response to the results of implementing the plan and new information becoming available. Depending on the scale and nature of the changes, a variety of activities in Figure 6 will need to be revisited to maintain the test plan.

For example, if, after the test plan is initially made available, it is realised that new risks threaten either the project or the deliverable product, or that the threat from existing risks has changed, then the process should be re-entered at the identify and analyse risks activity (TP3).

Alternatively, if it is believed to be necessary to change the test strategy for reasons other than risk (e.g. a different test environment is to be used) then the process should be re-entered at the design test strategy activity (TP5).

Or, if it is believed to be necessary to change either the staffing or scheduling of the testing for reasons other than risk (e.g. the availability of test items from development has changed) then the process should be re-entered at the determine staffing and scheduling activity (TP6).

### 7.2.2 Purpose

The purpose of the test strategy and planning process is to develop, agree, record and communicate to relevant stakeholders the scope and approach that will be taken to testing, enabling early identification of resources, environments and other requirements of testing.

### 7.2.3 Outcomes

As a result of the successful implementation of the test planning process:

a) The scope of the testing is analysed and understood.

b) The stakeholders who will participate in designing the test strategy and the test planning are identified and informed.

c) Risks that can be treated by testing are identified, analysed and classified with an agreed level of risk exposure.

d) Test strategy, test environment, test tool and test data needs are identified.

EXAMPLE 1 Tools, special equipment, test environment, office space.

e) Staffing and training needs are identified.

f) Each activity is scheduled.

g) Estimates are calculated and evidence to justify the estimates is recorded.

EXAMPLE 2 Cost, staff, and timeline estimates.

h) The test plan is agreed to and distributed to all stakeholders.

### 7.2.4 Activities and tasks

#### 7.2.4.1 General

The persons responsible for designing the test strategy and test planning will implement the activities and tasks specified in 7.2.4.2 to 7.2.4.10 in accordance with applicable organization policies and procedures with respect to the test strategy and planning process.

#### 7.2.4.2 Understand context (TP1)

This activity consists of the following tasks:

a) Understanding of the context and the scope of the software testing shall be obtained to support the preparation of the test plan.

NOTE 1 The scope of the software testing includes the identification of test items.

NOTE 2 The following documentation can be used:

— organizational test specifications, such as the organizational test policy and the organizational test practices;

— project management plan, for information that will affect the testing, such as the allocated testing budget and resources;

— higher level test plans (e.g. project test plan if managing a lower level of testing, such as system testing) for requirements and constraints on this level of testing, such as testing estimates, staffing and expected deliverables and their timing;

— applicable regulatory standards for information on regulations that may impact the testing;

— applicable product documentation, such as system requirement specifications, quality objectives described by system quality characteristics, and test item specifications, for information that relates to possible testing requirements for this level or type of testing;

— quality characteristics are defined in ISO/IEC 25010;

— software development plan, for information that may impact on testing timelines or cycles such as expected development deliverables and their timing;

— project risk register for information on identified project and product risks;

— verification and validation plan.

b) An understanding of the context and the software testing requirements should be obtained by identifying and interacting with the relevant stakeholders.

c) The test basis shall be identified and recorded.

d) Communication planning should be initiated, and lines of communication recorded.

NOTE 3 The activity, understand context, will be on-going throughout the lifetime of the project. The tasks in this activity can, in principle, be carried out in any order.

#### 7.2.4.3 Organize test plan development (TP2)

This activity consists of the following tasks:

a) Based on the testing requirements identified in the understand context activity (TP1), those activities that need to be performed to complete test design and test planning, shall be identified and scheduled.

b) The stakeholders required to participate in these activities should be identified.

c) Approval of the activities, schedule and participants shall be obtained from the relevant stakeholders.

EXAMPLE 1 The project manager and/or project test manager.

NOTE This can require repeating tasks a) and b).

d) Stakeholder involvement should be organized.

EXAMPLE 2 Request project manager to schedule a meeting for review of the test strategy.

#### 7.2.4.4 Identify and analyse risks (TP3)

This activity consists of the following tasks:

a) Any risks that have been previously identified shall be reviewed to identify those that relate to and/or can be treated by software testing.

EXAMPLE 1 Risks held in the project risk register.

b) Additional risks that relate to and/or can be treated by software testing shall be identified. Any identified risks that are not related to software testing should be communicated to the relevant stakeholders.

NOTE This can be achieved by reviewing product specifications and other appropriate documentation, through workshops, interviews or by other suitable means.

c) The risks shall be classified using an appropriate classification scheme that, at a minimum, discriminates between project and product risks.

d) Each risk shall be assigned a level of exposure (such as by considering its impact and likelihood).

e) Approval shall be obtained for the results of this risk assessment from the stakeholders.

f) The results of this risk assessment shall be recorded.

EXAMPLE 2 In the test plan, in the project risk register.

#### 7.2.4.5 Identify risk treatment approaches (TP4)

This activity consists of the following tasks:

a) Appropriate means of treating the risks shall be identified, based on the risk type, classification and level of risk exposure.

NOTE Appropriate means can include test levels, test types, test design techniques, test completion criteria, etc. Practitioners can consider the concept of software criticality covered in ISO/IEC/IEEE 15026-1 or IEEE 1012. Where constraints (such as time and cost) on testing are known, the treatments for risks with low risk exposure levels that are not expected to be treatable within these constraints will be identified as being out of scope for that reason.

b) The identified means of treating risks shall be recorded.

EXAMPLE In the test plan, in the project risk register.

#### 7.2.4.6 Design test strategy (TP5)

This activity consists of the following tasks:

a) A test strategy shall be designed that considers the test basis, risks, and organizational, project and product constraints.

NOTE 1 This takes into consideration the results of the risk assessment to prioritise the test activities and determine the resources needed to perform actions (e.g. time, costs, skills, tool support and environment needs) while meeting organizational, project and product constraints, such as:

— regulatory standards;

— the requirements of the organizational test policy and organizational test practices;

— higher level test plans and strategies on the project;

— contractual requirements;

— availability of appropriately skilled testers;

— availability of tools and environments.

Where it is not possible to design a test strategy that implements all the requirements of the organizational test practices and the recommendations for treating all identified risks while still meeting the project and product constraints, then a judgement is made to arrive at a test strategy that best meets these conflicting requirements. How this compromise is achieved will vary dependent on the project and on the organization. It can require the constraints to be relaxed and the identify risk treatment approaches activity (TP4) and this task to be repeated until an acceptable test strategy is achieved. Where it is decided to deviate from the organizational test practices, this should be recorded in the test strategy.

NOTE 2 The format of a test strategy is defined in ISO/IEC/IEEE 29119-3. It includes decisions on test levels, test types, aspects to be tested, test design techniques, test completion criteria, and suspension and resumption criteria.

NOTE 3 A test strategy would typically address static testing (e.g. reviews, inspections, static analysis) as well as dynamic testing.

b) Activities required to implement the test strategy shall be identified.

c) Metrics to be used for test monitoring and control (see activities TMC1 to TMC4) shall be identified.

d) Test data requirements shall be identified.

EXAMPLE Factors to consider when identifying test data requirements include regulations on data confidentiality (it can require data masking or encryption), volume of data required and data clean-up upon completion.

e) Test environment requirements and test tool requirements shall be identified.

f) Test deliverables shall be identified, and their degree of formality and frequency of communication should be recorded.

g) An initial estimate of the required resources to perform the complete set of activities required to implement the test strategy shall be produced.

NOTE 4 The initial test estimate that is produced in this step is finalised in the record test plan activity (TP7).

h) The test strategy shall be recorded.

NOTE 5 The test strategy will normally be a section of the test plan, but in some cases it can be recorded as a separate document.

i) Approval on the test strategy shall be obtained from the stakeholders.

NOTE 6 This can require repeating earlier tasks in this activity.

#### 7.2.4.7 Determine staffing and scheduling (TP6)

This activity consists of the following tasks:

a) The roles and skills of staff required to carry out the testing described in the test strategy should be identified.

NOTE 1 This can require identification of staff recruitment and/or training needs.

b) Each required test activity in the test strategy shall be scheduled based on the estimates, dependencies and staff availability.

c) Approval on staffing and scheduling shall be obtained from the relevant stakeholders.

NOTE 2 This can require repeating tasks a) and b), and if the test strategy needs revision, then the design test strategy activity (TP5) will need to be revisited.

#### 7.2.4.8 Record test plan (TP7)

This activity consists of the following tasks:

a) Final estimates for the testing shall be calculated based on the test strategy designed in the design test strategy activity (TP5) and the staffing and scheduling agreed in the determine staffing and scheduling activity (TP6).

NOTE Where these disagree with previous initial estimates it can be necessary to revisit the determine staffing and scheduling (TP6) and/or the design test strategy (TP5) activities.

b) The test strategy identified in the design test strategy activity (TP5), the staffing profile and schedule agreed in the determine staffing and scheduling activity (TP6), and the final estimates calculated in the previous task shall be incorporated in the test plan.

#### 7.2.4.9 Gain consensus on test plan (TP8)

This activity consists of the following tasks:

a) The views of the stakeholders on the test plan shall be gathered.

NOTE 1 This can be achieved through workshops, interviews or by other suitable means.

b) Conflicts between the test plan and stakeholders' views shall be resolved.

c) The test plan shall be updated to take into account feedback from stakeholders.

NOTE 2 This can require repeating earlier activities in the test strategy and planning process.

d) Obtain approval of the test plan from the relevant stakeholders.

EXAMPLE Relevant stakeholders can include the project manager, project owner or some other designated authority.

NOTE 3 This can require repeating tasks a) to c).

#### 7.2.4.10 Communicate test plan and make available (TP9)

This activity consists of the following tasks:

a) The test plan shall be made available.

b) The availability of the test plan shall be communicated to the stakeholders.

### 7.2.5 Information items

As a result of carrying out this process, the following information item shall be produced: test plan.

## 7.3 Test monitoring and control process

### 7.3.1 Overview

The test monitoring and control process as shown in Figure 7 scrutinizes whether testing progresses in accordance with the test plan and the organizational test specifications, such as the organizational test policy and the organizational test practices. If there are significant departures from planned progress, planned activities, or other aspects of the test plan, activities will be initiated to correct or compensate for the resultant variances.

Typical inputs to this process include:

— organizational test policy;

— organizational test practices;

— regulatory standards;

— project test plan (if planning testing for a specific phase or type within a project);

— incident reports;

— project management plan;

— applicable product documentation (e.g. system requirements, test item specification);

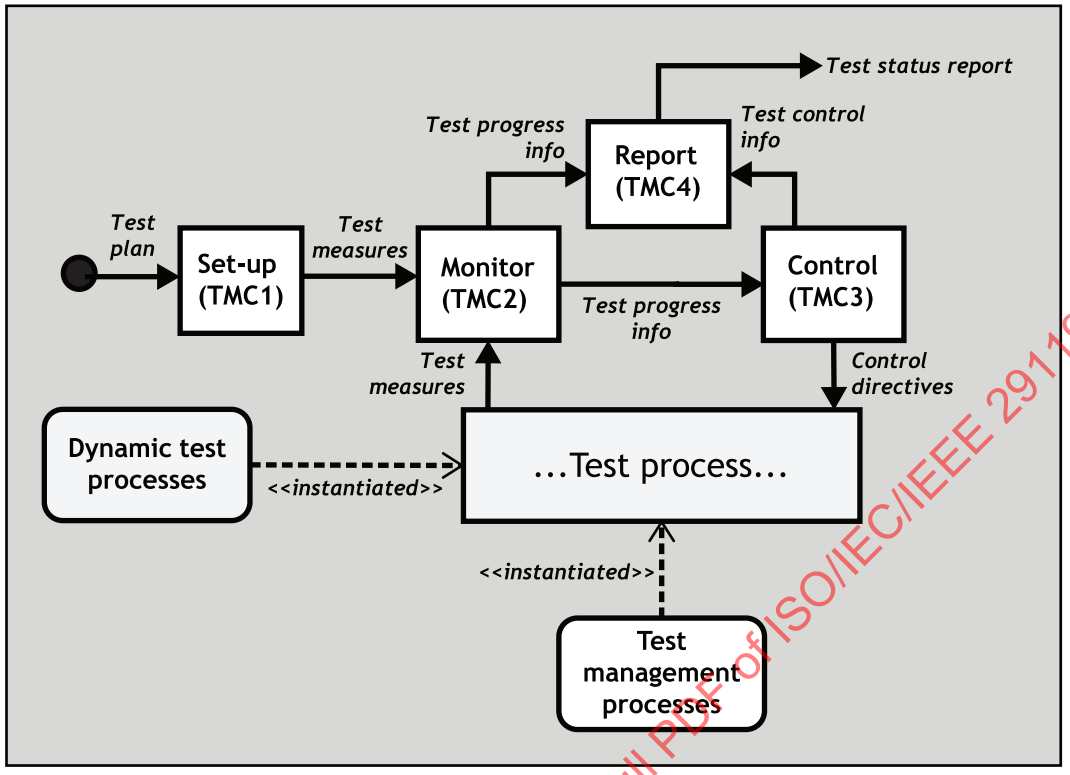
— software development plan;

— project and product risks; and

— test plan updates.

This process can be applied to the management of a whole project (normally made up of a number of test levels and test types) or to the management of the testing of a single test level (e.g. system testing) or test type (e.g. performance testing). In the latter case it is applied as part of the monitoring and control of dynamic testing described by the dynamic test processes. When applied as part of the monitoring

and control of the testing for a complete project then it will interact directly with the test management processes being used to manage the individual test levels and test types of the project.




NOTE The process is shown as purely sequential, but in practice it can be carried out iteratively, with some activities being revisited.

**Figure 7 — Test monitoring and control process**

### 7.3.2 Purpose

The purpose of the test monitoring and control process is to determine whether testing progresses in accordance with the test plan and with organizational test specifications (e.g. the organizational test policy and the organizational test practices). It also initiates control actions as necessary and identifies necessary updates to the test plan (e.g. revise completion criteria or identify new actions to compensate for deviations from the test plan).

The process is also used to determine whether testing progresses in accordance with higher level test plans, such as the project test plan, and to manage the testing performed at specific test levels (e.g. system testing) or for specific test types (e.g. performance testing).

### 7.3.3 Outcomes

As a result of the successful implementation of the test monitoring and control process:

a) The means of collecting suitable measures to monitor test progress and changing risk are set up.

b) Progress against the test plan is monitored.

c) New and changed test-related risks are identified, analysed and necessary action(s) invoked.

d) Necessary control actions are identified.

e) Necessary control actions are communicated to the relevant stakeholders.

f) The decision to stop testing is approved.

g) Test progress and changes to the risks are reported to stakeholders.

### 7.3.4 Activities and tasks

#### 7.3.4.1 General

The persons responsible for test monitoring and control will implement the activities and tasks specified in 7.3.4.2 to 7.3.4.5 in accordance with applicable organization policies and procedures with respect to the test monitoring and control process.

#### 7.3.4.2 Set-up (TMC1)

This activity consists of the following tasks:

a) Suitable measures for monitoring progress against the test plan should be identified if these measures are not already defined in the test plan or the organizational test practices document.

b) Suitable means of identifying new and changing risks should be identified if these are not already defined in the test plan or the organizational test practices document.

c) Monitoring activities, such as test status reporting and test metrics collection, shall be put in place to collect the measures identified in tasks a) and b), and in the test plan and the organizational test practices document.

#### 7.3.4.3 Monitor (TMC2)

This activity consists of the following tasks:

a) The test measures shall be collected and recorded.

b) Progress against the test plan shall be monitored using the collected test measures.

EXAMPLE 1 By examining test status reports, analysing test measures and meeting with stakeholders.

c) Divergence from planned testing activities shall be identified and any factors blocking progress recorded.

d) New risks shall be identified and analysed to identify those that require treatment by testing and those that need to be communicated to other stakeholders.

e) Changes to known risks shall be monitored to identify those that require treatment by testing and those that need to be communicated to other stakeholders.

EXAMPLE 2 Communicate risks that require testing as treatment to the project manager.

NOTE Tasks a) to e) are repeated on a regular basis, until it is determined that the testing specified in the test plan can be terminated or is complete, which would typically be done by checking whether the completion criteria have been achieved.

#### 7.3.4.4 Control (TMC3)

This activity consists of the following tasks:

a) Those actions necessary to implement the test plan shall be performed.

EXAMPLE 1 Assigning responsibility for testing activities to testers.

b) Those actions necessary to implement control directives received from higher level management processes shall be performed.

EXAMPLE 2 Actions from the project test manager if a specific level of testing is being managed.

c) Those actions necessary to manage the divergence of actual testing from planned testing shall be identified.

NOTE 1 These control actions can require changes to the testing, the test plan, test data, test environment, staffing and/or changes in other areas, such as development.

d) Means of treating newly identified and changed risks shall be identified.

NOTE 2 This can include assigning more staff to specific tasks and changing test completion criteria.

e) As appropriate:

1) control directives shall be issued to make changes to the way testing is performed;

2) changes to the test plan shall be in the form of test plan updates; and

3) recommended changes shall be communicated to the relevant stakeholders.

EXAMPLE 3 IT Support for test environments.

f) Readiness for commencing any assigned test activity shall be established before commencing that activity, if not already done.

NOTE 3 This can typically be performed by checking against entry criteria described in the test plan.

NOTE 4 Assigned test activity can be test execution.

NOTE 5 Readiness can have been established in the test design and implementation process and/or the test environment and data management process.

g) Approval shall be granted at the completion of assigned test activities.

EXAMPLE 4 Completion of a lower level of testing.

NOTE 6 This will typically be performed by checking against exit criteria described in the test plan.

h) When the testing has met its completion criteria, approval for the test completion decision shall be obtained.

#### 7.3.4.5 Report (TMC4)

This activity consists of the following tasks:

a) Testing progress against the test plan shall be communicated to stakeholders in a test status report for the specified reporting period.

b) New risks and changes to existing risks shall be updated in the risk register and communicated to the relevant stakeholders.

### 7.3.5 Information items

As a result of carrying out this process, the following information items shall be produced:

a) test status reports;

b) test plan updates;

c) control directives (e.g. changes to the testing, the test plan, test data, test environment and staffing);

d) project and product risk information.

NOTE Risk information can be held in the project risk register or locally in the test plan.

## 7.4 Test completion process

### 7.4.1 Overview

The test completion process as shown in Figure 8 is performed when agreement has been obtained that the testing activities are complete. It will be performed to complete the testing carried out at a specific test level (e.g. system testing) or test type (e.g. performance testing) and to complete the testing for a complete project.

Typical inputs to this process include:

— project test plan;

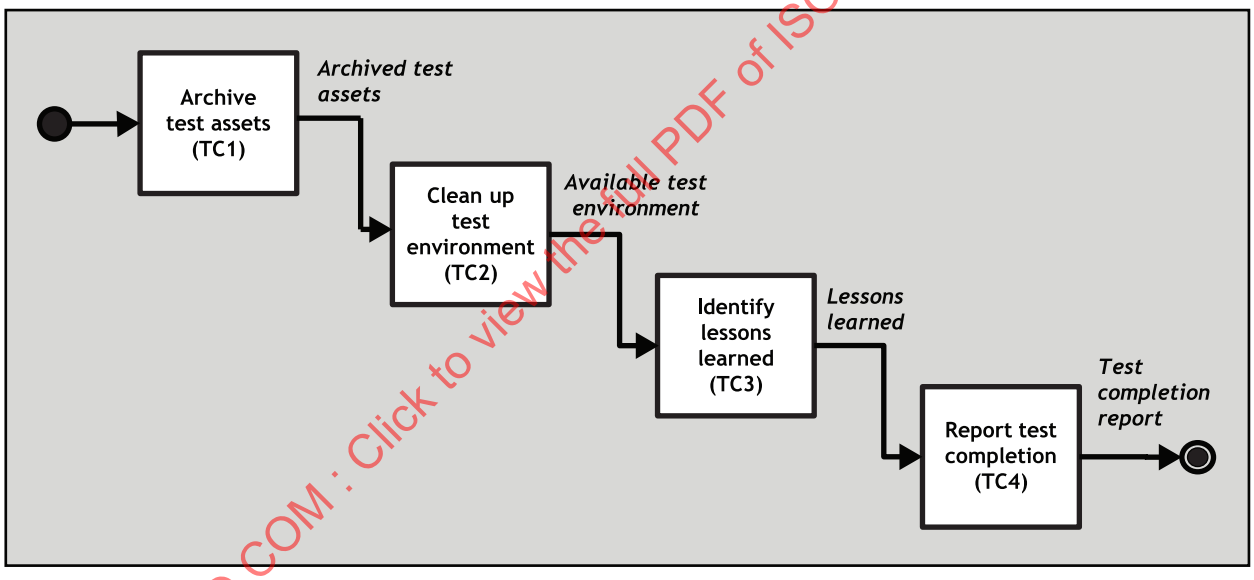
— phase test plans;

— incident reports;

— project test status reports;

— phase/type test completion reports; and

— organizational test practices (if relevant).




NOTE The process is shown as purely sequential, but in practice it can be carried out iteratively, with some activities being revisited.

**Figure 8 — Test completion process**



### 7.4.2 Purpose

The purpose of the test completion process is to make available useful test assets for later use, leave the test environment in a satisfactory condition, and record and communicate the results of the testing to relevant stakeholders. Test assets include test plans, test case specifications, test scripts, test tools, test data and test environment infrastructure.

### 7.4.3 Outcomes

As a result of the successful implementation of the test completion process:

a) Test assets are either archived or passed directly to the relevant stakeholders.

b) The test environment is in its agreed state (e.g. so that it is available for any following testing).

c) The test completion report is recorded.

d) The test completion report is approved.

e) The test completion report is communicated to relevant stakeholders.

### 7.4.4 Activities and tasks

#### 7.4.4.1 General

The persons responsible for test completion will implement the activities and tasks specified in 7.4.4.2 to 7.4.4.5 in accordance with applicable organization policies and procedures with respect to the test completion process.

#### 7.4.4.2 Archive test assets (TC1)

This activity consists of the following tasks:

a) Those test assets that may be useful in future or are expected to be reused at a later date should be identified, made available using appropriate means and archived.

EXAMPLE 1 Test plans, manual and/or automated test procedures, test environment infrastructure.

EXAMPLE 2 The test assets to be reused are appropriately labelled within the configuration management system.

b) The availability of reusable test assets shall be recorded in the test completion report and communicated to the relevant stakeholders.

EXAMPLE 3 Those responsible for maintenance testing (to achieve successful transition) and the project test manager.

#### 7.4.4.3 Clean up test environment (TC2)

This activity consists of the following task: the test environment shall be restored to a pre-defined state on completion of all testing activities.

EXAMPLE Restore settings and hardware to original state.

#### 7.4.4.4 Identify lessons learned (TC3)

This activity consists of the following tasks:

a) Lessons learned during the project shall be recorded.

NOTE 1 This can be achieved by recording:

— what went well during testing and associated activities;

— what did not go well during testing and associated activities;

— recommended improvements to the testing and other processes, such as the development process.

NOTE 2 In agile, lessons learned can be identified during retrospectives. In traditional projects, they are sometimes identified at post-project review meetings or lessons learned meetings.

b) The outcomes shall be recorded for inclusion in the test completion report and communicated to the relevant stakeholders.

EXAMPLE Outcomes include feedback to the organizational level, such as test process improvements.

#### 7.4.4.5 Report test completion (TC4)

This activity consists of the following tasks:

a) Relevant information shall be collected from the following documents, but not limited to:

   1) test plans (e.g. project test plan, system test plan, or performance test plan);

   2) test results;

   3) test status reports;

   4) test completion reports from test level or test type; and

      EXAMPLE From unit testing, performance testing, acceptance testing, etc., if this is reporting on the completion of testing for the whole project.

   5) incident reports.

b) The collected information shall be evaluated and summarized in the test completion report.

c) Approval for the test completion report shall be obtained from the responsible stakeholders.

d) The approved test completion report shall be distributed to the relevant stakeholders.

### 7.4.5 Information items

As a result of carrying out this process, the following information item shall be produced: test completion report.

## 8 Dynamic test processes

### 8.1 General

The dynamic test processes are used to carry out dynamic testing within a particular level of testing (e.g. unit, integration, system and acceptance) or type of testing (e.g. performance testing, security testing, usability testing). The processes for the management of this dynamic testing are described in Clause 7.

There are four dynamic test processes (as shown in Figure 9):

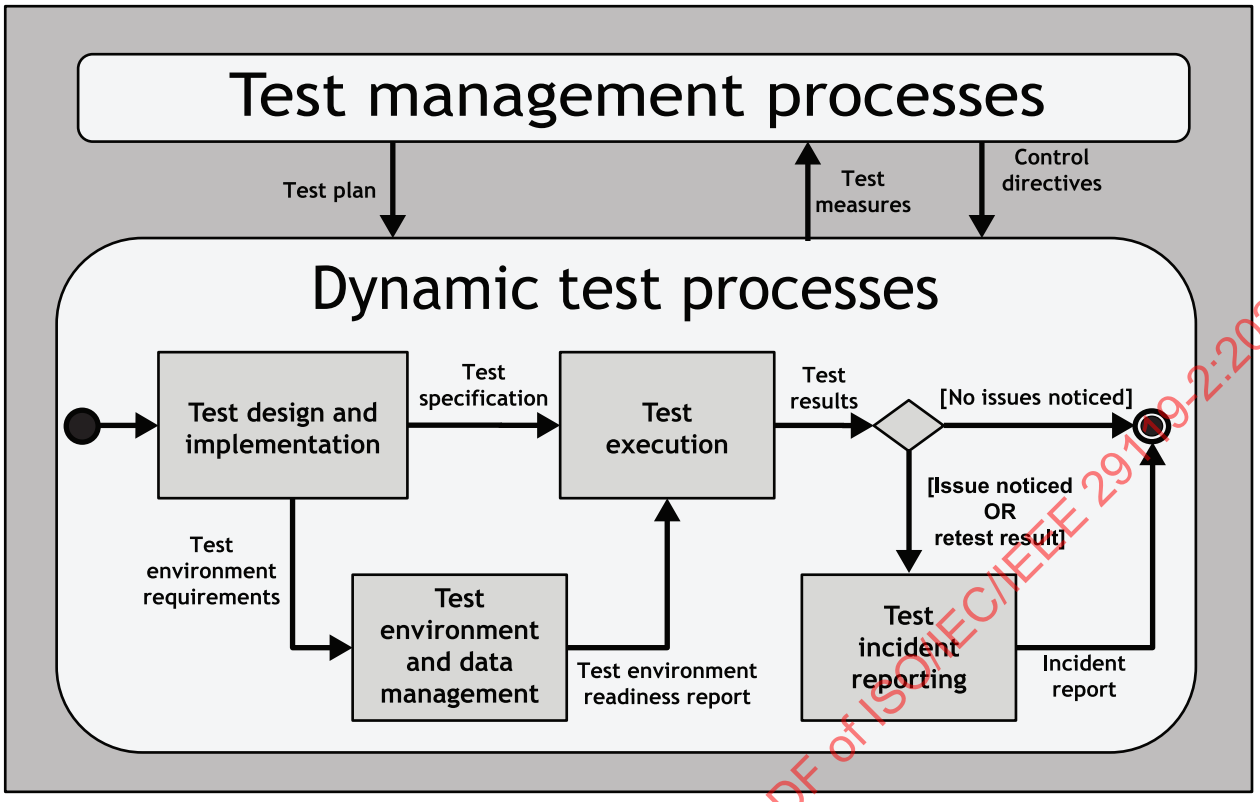
a) test design and implementation;

b) test environment and data management;

c) test execution; and

d) test incident reporting.

Figure 9 illustrates how the dynamic test processes interact and the relationship with the test management processes. These dynamic test processes would normally be invoked as part of the implementation of the test strategy documented within the test plan for the test level (e.g. system testing) or test type (e.g. performance testing) being performed.




NOTE The process is shown as purely sequential, but in practice it can be carried out iteratively, with some activities being revisited.

**Figure 9 — Dynamic test processes**

For any particular test, the dynamic test processes will execute in the order shown in Figure 9, but these processes will typically be invoked a number of times to complete the testing for a given test level (e.g. system testing) or test type (e.g. performance testing). This is because as tests are designed and run, the overseeing test management process (test monitoring and control) monitors test progress (through test measures) and may require further tests (through control directives) to be designed and run until the test completion criterion for this test activity is achieved.

Test measures, which are an output of the dynamic test processes and an input into the test monitoring and control process (see Figure 7), can be produced during any activity of the dynamic test processes. Test measures are used to report the status and progress of testing to test management staff. For example, test measures can be used to indicate to test management how many test cases have been designed by the testing team. Test measures can also be used by the test management staff to evaluate the quality of the test item, and to measure the efficiency and effectiveness of the testing and defect fixing activities.

Similarly, control directives are an output of the test management process and an input to the dynamic test process (see Figure 7) and can be acted upon during any activity of the dynamic test processes. Control directives correspond to instructions from test management staff that dictate how dynamic testing should be conducted by the test team. For example, a control directive can be given to a test team that instructs them to design additional test cases for new program features that have been allocated to their team by their test manager.

Since test measures can be produced during any activity of the dynamic test processes, and since control directives can be acted upon during any activity of these processes, the production of measures and handling of directives are not shown as tasks in any specific activity of these processes.

## 8.2 Test design and implementation process

### 8.2.1 Overview

The test design and implementation process, shown in Figure 10, is used to derive test cases and test procedures.

Typical inputs to this process include:

— test basis;

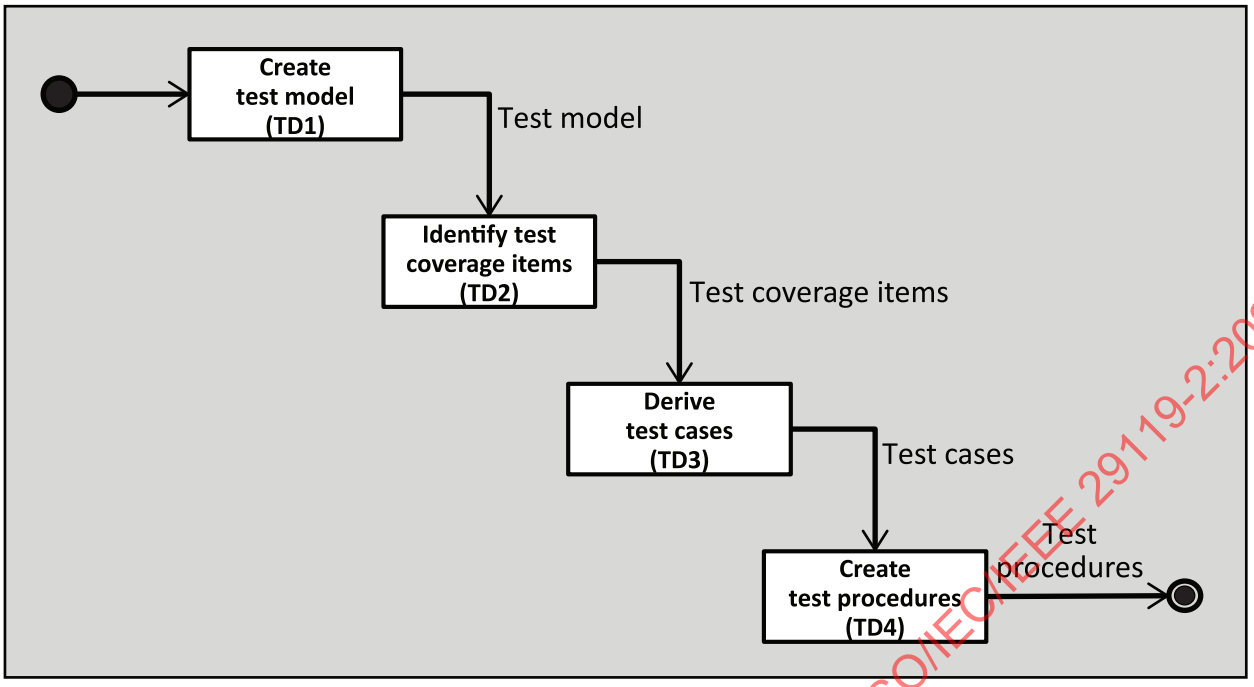
— required test coverage; and

— risk information.

The test design and implementation process may be exited and re-entered for a number of reasons. For instance, it will typically be re-entered if, after executing a test procedure, it is realized that additional test cases are needed to meet the required test completion criteria. Thus, it is possible, indeed probable, that only a subset of all the test cases that are needed to meet the test completion criteria will be derived during any one implementation of this process.

This process requires testers to apply a test design technique to derive test cases and test procedures with the ultimate aim of achieving the test completion criteria typically described in terms of test coverage measures. These test completion criteria and test design techniques are specified in the test plan. The test design and implementation process will typically be run for each of the test completion criteria in the test plan. Test design techniques and test measures are defined in ISO/IEC/IEEE 29119-4. In some situations, the derivation of test cases (and test procedures) can be achieved using a form of test automation in the form of model-based testing.

A number of situations can cause iteration between activities in this process. Such situations include stakeholders failing to agree with the results of an activity, such as the initial validity of the test model. Similarly, situations can occur where the results of an activity make it clear that test strategy decisions, such as the choice of test completion criteria, are incompatible with constraints on project timelines, requiring test management processes to be revisited.




Note The process is shown as purely sequential, but in practice it can be carried out iteratively, with some activities being revisited.

**Figure 10 — Test design and implementation process**

### 8.2.2 Purpose

The purpose of the test design and implementation process is to derive test procedures that will be executed during the test execution process. As part of this process the test basis is analysed and a test model, test coverage items, test cases and test procedures are derived.

### 8.2.3 Outcomes

As a result of the successful implementation of the test design and implementation process:

a) The test basis for each test item is analysed.

b) A test model is created.

c) Test coverage items are identified.

d) Test cases are derived.

e) Test procedures are created.

### 8.2.4 Activities and tasks

#### 8.2.4.1 General

The persons responsible for test design and implementation will implement the activities and tasks specified in 8.2.4.2 to 8.2.4.5 in accordance with applicable organization policies and procedures with respect to the test design and implementation process.

#### 8.2.4.2 Create test model (TD1)

This activity consists of the following tasks:

a) The test basis shall be analysed to understand the requirements for the test item.

NOTE 1 If defects in the test basis are uncovered during this analysis these are reported using an appropriate incident management system.

b) The test strategy shall be used to identify those parts and/or characteristics of the test item that are to be tested.

EXAMPLE For state transition testing (STT), identify those parts of the test item for which STT is applicable.

c) The notation for the test model shall be decided based on the required test coverage.

NOTE 2 The required test coverage will normally be specified as part of the test completion criteria in the test plan.

EXAMPLE Notations can include requirements statements (including those in natural language), equivalence partitions, state transition diagram, use case description, decision table, input syntax, source code, control flow graph, parameters and values, classification tree, among others.

d) The test model shall be created for the test item.

NOTE 3 In some situations the test model can already be available as an artefact of the development process, such as a design specification or source code.

e) Additional prioritization information may be added to the test model using relevant risk exposure levels.

f) The test model should be agreed with the stakeholders.

NOTE 4 Where necessary, tasks a) to d) are revisited in order to achieve stakeholder agreement.

g) The test model shall be recorded in the test model specification.

h) The traceability between the test basis and the test model shall be recorded.

#### 8.2.4.3 Identify test coverage items (TD2)

This activity consists of the following tasks:

a) The test coverage items to be exercised by the testing shall be derived by applying test design techniques to the test model to achieve the test completion coverage criteria specified in the test plan.

NOTE 1 Where a test completion criterion for the test item is specified as less than 100 % of a test coverage measure, a subset of the test coverage items required to achieve 100 % coverage needs to be selected to be exercised by the testing.

NOTE 2 There can be criteria provided to aid in this selection in the test plan or organizational test practices (e.g. discard test coverage items associated with lower risk exposures). This selection will sometimes need to be revisited based on the results of later activities.

b) The test coverage items shall be prioritized using the risk exposure levels documented in the identify and analyse risks activity (TP3).

c) The test coverage items shall be recorded in the test case specification.

d) The traceability between the test basis, test model and test coverage items shall be recorded.

#### 8.2.4.4 Derive test cases (TD3)

This activity consists of the following tasks:

a) One or more test cases shall be derived by determining pre-conditions, selecting input values and, where necessary, actions to exercise the selected test coverage items, and by determining the corresponding expected results.

NOTE 1 When deriving the test cases, one test case can exercise more than one test coverage item and thus there is the opportunity to combine coverage of multiple test coverage items in a single test case. This can reduce test execution times but can also increase debugging times.

NOTE 2 More test cases can be derived for those test coverage items with higher risk exposure levels to increase the rigour of testing for those test coverage items.

b) The test cases shall be prioritized using the risk exposure levels documented in the identify and analyse risks activity (TP3).

c) The test cases shall be recorded in the test case specification.

d) The traceability between the test basis, test model, test coverage items and test cases shall be recorded.

e) The test case specification shall be approved by the stakeholders.

NOTE 3 This can require repeating tasks a) and b), and in some cases, first repeating the create test model activity (TD1) and/or the identify test coverage items activity (TD2) in order to achieve stakeholder agreement.

#### 8.2.4.5 Create test procedures (TD4)

This activity consists of the following tasks:

a) Test procedures shall be derived by ordering test cases according to dependencies described by pre-conditions and post-conditions and other testing requirements.

EXAMPLE Risk priorities can be used to order test cases.

NOTE 1 Any other required actions can be included in the test procedure, such as those necessary to set up the pre-conditions for a test case.

NOTE 2 Where test procedures are to be executed using tools, it can be necessary to further elaborate them by adding extra detail to create automated test scripts.

b) Any test data and test environment requirements not already included in the test plan shall be identified.

NOTE 3 Although it's possible that this activity is not finalized until the derivation of test procedures is complete, this task can often start far earlier in the process, sometimes even as early as when the test model is agreed.

c) The test procedures shall be prioritized using the risk exposure levels documented in the identify and analyse risks activity (TP3).

d) The test procedures shall be recorded in the test procedure specification.

e) The traceability between the test basis, test model, test coverage items, test cases, and test procedures (and/or automated test scripts) shall be recorded.

f) The test procedure specification shall be approved by the stakeholders.

NOTE 4 This can require repeating tasks a) to e) in order to achieve stakeholder agreement.

### 8.2.5 Information items

As a result of carrying out this process, the following information items shall be produced:

a) test specifications (test model specification, test case specifications and test procedure specifications) and related traceability information;

b) test data requirements;

c) test environment requirements.

## 8.3 Test environment and data management process

### 8.3.1 Overview

The test environment and data management process, shown in Figure 11, is used to establish and maintain the environment in which tests are executed and manage the corresponding test data. Maintenance of the test environment and test data may involve changes based on the results of previous tests. Where change and configuration management processes exist, changes to the test environments may be managed using these processes.

Typical inputs to this process include:

— test plans;

— test environment requirements;

— test data requirements;

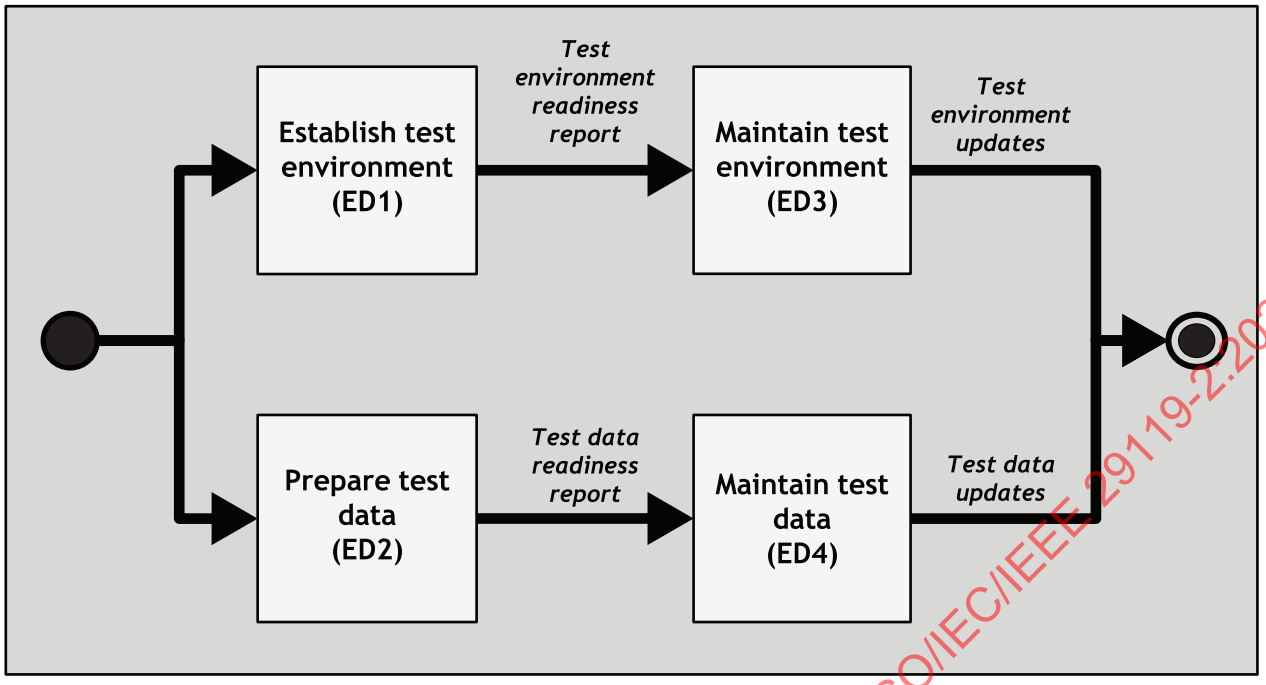
— intended/operational environment;

— test basis;

— test procedures; and

— test results (where available).

The requirements for the test environment and test data will initially be described in the test plan, but the detailed composition of the test environment and test data will normally only become clear once the test design and implementation process has started.




Note The process is shown as purely sequential, but in practice it can be carried out iteratively, with some activities being revisited.

**Figure 11 — Test environment and data management process**

### 8.3.2 Purpose

The purpose of the test environment and data management process is to establish and maintain the required test environment and test data and to communicate their status to all relevant stakeholders.

### 8.3.3 Outcomes

As a result of the successful implementation of the test environment and data management process:

a) The test environment is set-up in a state ready for testing.

b) The status of the test environment is communicated to all relevant stakeholders.

c) The test environment is maintained.

d) The test data is prepared and is in a state ready for testing.

e) The status of the test data is communicated to all relevant stakeholders.

f) The test data is maintained.

### 8.3.4 Activities and tasks

#### 8.3.4.1 General

The persons responsible for test environment and test data management (such as IT support technicians) will implement the activities and tasks specified in 8.3.4.2 to 8.3.4.5 in accordance with applicable organization policies and procedures with respect to the test environment and data management process.

#### 8.3.4.2 Establish test environment (ED1)

This activity consists of the following tasks:

a) Based on the test plan, the detailed requirements generated as a result of the test design and implementation process, the requirement for test tools, and the scale/ formality of the testing, the following shall be performed:

1) plan the set-up of the test environment;

EXAMPLE Requirements, interfaces, schedules and costs.

2) design the test environment;

3) determine the degree of configuration management to be applied (where appropriate);

4) build the test environment;

NOTE 1 This can include hardware and software items, as appropriate.

NOTE 2 Design and implementation of the test environment can include the use of various software/system life cycle processes (refer to ISO/IEC/IEEE 12207 or ISO/IEC/IEEE 15288 for more information).

5) set up test tools to support the testing (where appropriate);

6) install and configure the test item on the test environment;

7) verify that the test environment meets the test environment requirements; and

8) where required, provide assurance that the test environment meets the defined requirements.

b) The status of the test environment shall be recorded and communicated through the test environment readiness report to the relevant stakeholders.

NOTE 3 Relevant stakeholders can include testers and the test manager.

c) The test environment readiness report shall include a description of the known differences between the test environment and the operational environment.

#### 8.3.4.3 Prepare test data (ED2)

This activity consists of the following tasks:

a) Based on the test plan, the detailed requirements generated as a result of the test design and implementation process, and the scale/ formality of the testing, the following shall be performed:

1) plan the preparation of the test data;

2) prepare the test data;

3) set up test data to support the testing;

4) verify that the test data meets the test data requirements; and

5) where required, provide assurance that the test data meets the defined requirements.

b) The status of the test data shall be recorded and communicated through the test data readiness report to the relevant stakeholders.

NOTE Relevant stakeholders can include testers and the test manager.

#### 8.3.4.4 Maintain test environment (ED3)

This activity consists of the following tasks:

a) The test environment shall be maintained as defined by the test environment requirements.

NOTE This can require making changes based on the results of previous tests.

b) Changes to the status of the test environment shall be communicated to the relevant stakeholders.

EXAMPLE The testers and the test manager.

#### 8.3.4.5 Maintain test data (ED4)

This activity consists of the following tasks:

a) The test data shall be maintained as defined by the test data requirements.

NOTE This can require making changes based on the results of previous tests.

b) Changes to the status of the test data shall be communicated to the relevant stakeholders.

EXAMPLE The testers and the test manager.

### 8.3.5 Information items

As a result of carrying out this process, the following information items shall be produced:

a) test environment;

b) test environment readiness report;

c) test environment updates (where applicable);

d) test data;

e) test data readiness report;

f) test data updates (where applicable).

## 8.4 Test execution process

### 8.4.1 Overview

The test execution process, shown in Figure 12, is used to run the test procedures generated as a result of the test design and implementation process on the test environment established by the test environment and data management process. The test execution process may need to be performed a number of times as it may not be possible to execute all the available test procedures in a single iteration. If an issue is fixed it should be retested by re-entering the test execution process.

Typical inputs to this process include:

— test plans;

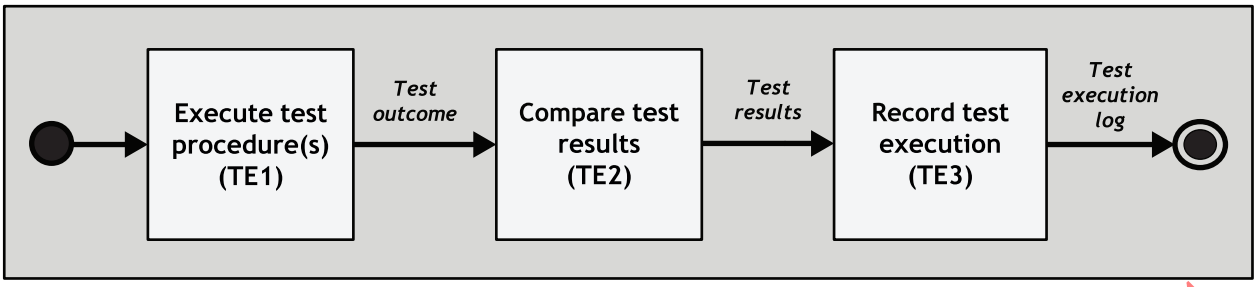
— test procedures;

— test item;

— test basis;

— test environment readiness report (where available); and

— test environment updates (where available).




NOTE The process is shown as purely sequential, but in practice it can be carried out iteratively, with some activities being revisited.

**Figure 12 — Test execution process**

Comparison of test results and recording of test execution details will normally be interleaved with the execution of test procedures.

### 8.4.2 Purpose

The purpose of the test execution process is to execute the test procedures created in the test design and implementation process in the prepared test environment and record the results.

### 8.4.3 Outcomes

As a result of the successful implementation of the test execution process:

a) The test procedures are executed.

b) The actual results are recorded.

c) The actual and expected results are compared.

d) The test results are determined.

### 8.4.4 Activities and tasks

#### 8.4.4.1 General

The persons responsible for test execution will implement the activities and tasks specified in 8.4.4.2 to 8.4.4.4 in accordance with applicable organization policies and procedures with respect to the test execution process.

NOTE 1 The test execution process can be interrupted by circumstances such as when a defect is discovered in a test case, a problem is discovered in the test environment or changes are made to the test plan (e.g. due to project cost or time modifications) or circumstances that have been specified under suspension criteria. In these circumstances the process is resumed at the appropriate task or cancelled altogether.

NOTE 2 The test execution process will be re-entered if, after executing one or more test cases, it is realized that additional test cases need to be executed in order to meet the required test completion criteria. Thus, only a subset of all test cases for a test item can be executed during any one iteration of this process.

#### 8.4.4.2 Execute test procedures (TE1)

This activity consists of the following tasks:

a) One or more test procedures shall be executed in the prepared test environment.

NOTE 1 The test procedures can have been scripted for automated execution, or can have been recorded in a test specification for manual test execution, or can be executed immediately as they are designed as in the case of exploratory testing.

b) The actual results for each test case in the test procedure shall be observed.

c) The actual results shall be recorded.

NOTE 2 This can be in a test tool or manually, as indicated in the test case specification.

NOTE 3 Where exploratory testing is performed, actual results can be observed, and not recorded when the actual results match the expected results.

#### 8.4.4.3 Compare test results (TE2)

This activity consists of the following tasks:

a) The actual and expected results for each test case in the test procedure shall be compared.

NOTE 1 Expected results can have been recorded in the test specification or can be an undocumented expectation in the case of exploratory testing. In the case of automated testing the expected results are normally embedded in the automated test script (or in an associated file) and the test tool performs the comparison automatically.

b) The test result of executing the test cases in the test procedure shall be determined. If a retest is passed, this will require an update to an incident report by the test incident reporting process.

NOTE 2 Failures and unexpected changes to the test environment will result in issues (potential incidents) being passed to the test incident reporting process.

#### 8.4.4.4 Record test execution (TE3)

This activity consists of the following task: details of test execution shall be recorded, as specified in the test plan.

NOTE This would normally be recorded in the execution log.

### 8.4.5 Information items

As a result of carrying out this process, the following information items shall be produced:

a) actual results;

NOTE The actual results are compared with the expected results to determine the test result.

b) test results;

EXAMPLE Test passes, failures or the result of the test is inconclusive.

c) test execution log.

## 8.5 Test incident reporting process

### 8.5.1 Overview

The test incident reporting process, shown in Figure 13, is used for the reporting of test incidents. This process will be entered as a result of the identification of test failures, instances where something unusual or unexpected occurred during test execution, or when a retest passes.

Typical inputs to this process include:

— test results;

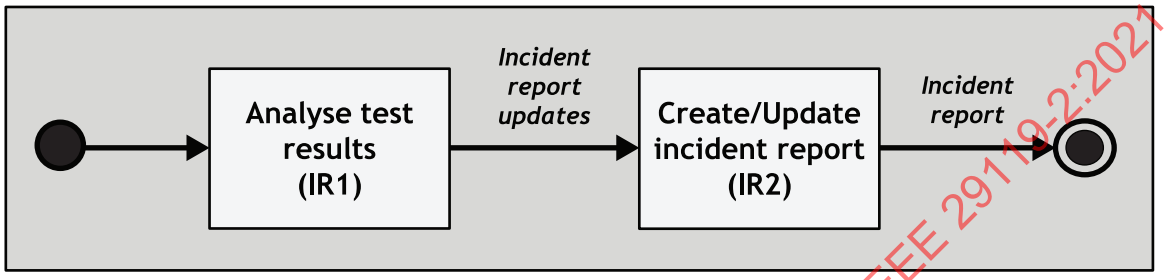
— test procedures;

— test cases;

— test items;

— test basis; and

— test execution logs (where available).




Note The process is shown as purely sequential, but in practice it can be carried out iteratively, with some activities being revisited.

**Figure 13 — Test incident reporting process**

### 8.5.2 Purpose

The purpose of the test incident reporting process is to report to the relevant stakeholders those incidents requiring further action identified as a result of test execution. In the case of a new test this will require an incident report to be created. In the case of a retest, this will require the status of a previously raised incident report to be updated but may also require a new incident report to be raised where further incidents are identified.

### 8.5.3 Outcomes

As a result of the successful implementation of the test incident reporting process:

a) Test results are analysed.

b) New incidents are confirmed.

c) New incident report details are created.

d) The status and details of previously raised incidents are determined.

e) Previously raised incident report details are updated as appropriate.

f) New and/or updated incident reports are communicated to the relevant stakeholders.

### 8.5.4 Activities and tasks

#### 8.5.4.1 General

The persons responsible for test incident reporting will implement the activities and tasks specified in 8.5.4.2 and 8.5.4.3 in accordance with applicable organization policies and procedures with respect to the test incident reporting process.

#### 8.5.4.2 Analyse test results (IR1)

This activity consists of the following tasks:

a) Where a test result relates to a previously raised incident, the test result shall be analysed and the incident details shall be updated.

b) Where a test result indicates that a new issue has been identified, the test result shall be analysed and it will be determined whether it is an incident that requires reporting, an action item that will be resolved without incident reporting, or requires no further action to be taken.

NOTE Where appropriate, the decision not to raise an incident report is discussed with the originator to aid mutual understanding of this decision.

c) Action items shall be assigned to an appropriate person for resolution.

#### 8.5.4.3 Create/update incident report (IR2)

This activity consists of the following task:

a) The information that needs to be recorded about the incident shall be identified and reported/updated.

NOTE 1 Incident reports can be raised against test items and other items such as test procedures, the test basis and the test environment.

NOTE 2 Following a successful retest, the incident report can be updated and closed.

b) The status of new and/or updated incidents shall be communicated to the relevant stakeholders.

### 8.5.5 Information items

As a result of carrying out this process, the following information item shall be produced: incident report.

# Annex A
(informative)

# Example application of the test design and implementation process



## A.1 General

This annex provides an example application of activities TD1 to TD4 of the test design and implementation process.

## A.2 Fragment of test basis

*"The insurance quote system shall generate an accept message, a reject message or an accept with excess warning message.*

*It will accept insurance applicants from the age of 18 and up to the age of 80 years on the day of application based on their input age in whole years; all other inputs shall be rejected.*

*Accepted applicants of 70 and over shall be accepted but with a warning that in the event of a claim they shall pay an excess."*

## A.3 Test completion criterion

*"The test completion criterion is that 100 % equivalence partition coverage is achieved, and all test cases must result in a "pass" status on execution."*

## A.4 Test model (TD1)

Based on the test completion criterion, the test model in Figure A.1 showing the equivalence partitions (with the black outlines) for the described system behaviour can be created.

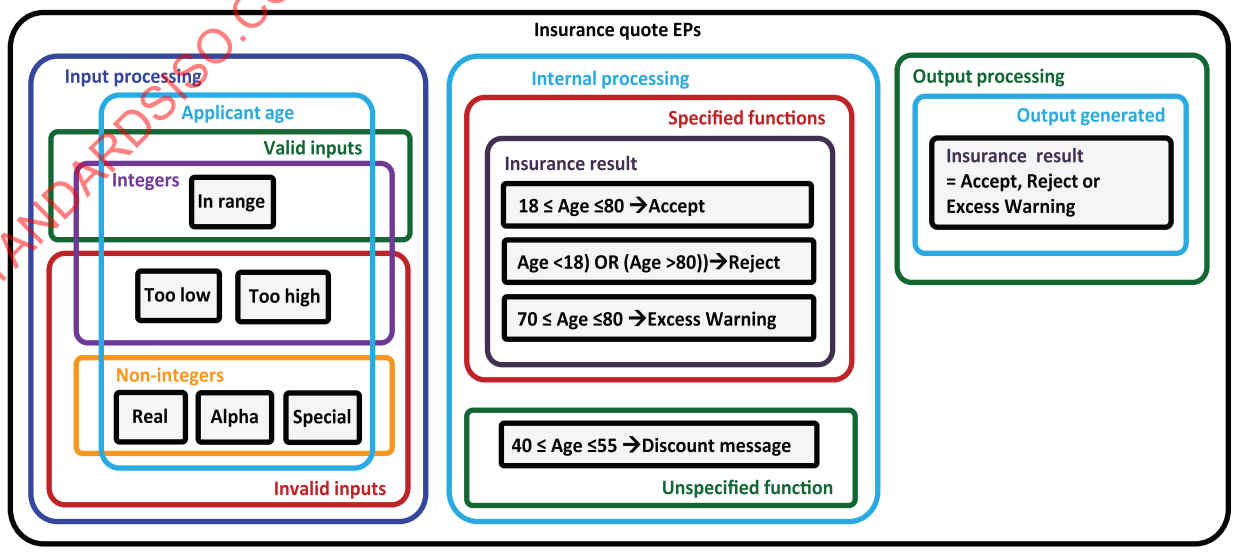



**Figure A.1 — Equivalence partitions for the 'insurance quote system'**

The test model in Figure A.1 is an Euler diagram showing the equivalence partitions.

Alternatively, the test model can be presented purely as text:

| | | |
|---|---|---|
| In range valid input: | 18 ≤ Age ≤ 80 | (PART-1) |
| Too low invalid input: | Age < 18 | (PART-2) |
| Too high invalid input: | Age > 80 | (PART-3) |
| Real number invalid input: | Age = real | (PART-4) |
| Alphabetic invalid input: | Age = alphabetic | (PART-5) |
| Special character invalid input: | Age = special char | (PART-6) |
| Derivation of 'Accept' message: | 18 ≤ Age ≤ 80 | (PART-7) |
| Derivation of 'Reject' message: | (Age < 18) OR (Age > 80) | (PART-8) |
| Derivation of 'Excess Warning' message: | 70 ≤ Age ≤ 80 | (PART-9) |
| Derivation of 'Discount' message: | 40 ≤ Age ≤ 55 | (PART-10) |
| Valid output generated: | Any input | (PART-11) |

The unspecified function can be any function other than those specified in the test basis. It can be challenging to identify unspecified functions; however, they shall be considered because if one can be caused to run, then defects in the test item, its test basis, or both, have been identified. For this example, just one unspecified function was identified (the unwanted requirement to provide a discount message for applicants between the ages of 40 and 55). Note that other testers may well derive quite different invalid functions.

## A.5 Test coverage items (TD2)

Using equivalence partitioning then the following eight test coverage items are derived that will cover all of the identified equivalence partitions:

| | | |
|---|---|---|
| TCI-1. | 18 ≤ Age ≤ 80 | (covers PART-1 / PART-7 / PART-11) |
| TCI-2. | Age < 18 | (covers PART-2 / PART-8 / PART-11) |
| TCI-3. | Age > 80 | (covers PART-3 / PART-8 / PART-11) |
| TCI-4. | Age = real | (covers PART-4 / PART-11) |
| TCI-5. | Age = alphabetic | (covers PART-5 / PART-11) |
| TCI-6. | Age = special char | (covers PART-6 / PART-11) |
| TCI-7. | 70 ≤ Age ≤ 80 | (covers PART-9 / PART-1 / PART-7 / PART-11) |
| TCI-8. | 40 ≤ Age ≤ 55 | (covers PART-10 / PART-1 / PART-7 / PART-11) |

## A.6 Test cases (TD3)

As long as test cases that exercise each of the eight test coverage items are generated, 100 % equivalence partition coverage is achieved.

When generating test cases, it can be seen that a single test case can sometimes exercise more than one test coverage item. There is an obvious benefit in minimizing the number of test cases as this reduces the test execution time, but this benefit can sometimes be counteracted by the extra time required to determine the minimal set and the potentially more complex debugging required when test cases target multiple test coverage items.

In this example two of the test cases exercise more than one test coverage item as shown below:

| | | | |
|---|---|---|---|
| CASE#1. | Input: 'Age =53' | Expected result: 'Accept'. | (exercises TCI-1 & TCI-8) |
| CASE#2. | Input: 'Age = 15' | Expected result: 'Reject'. | (exercises TCI-2) |
| CASE#3. | Input: 'Age = 89' | Expected result: 'Reject'. | (exercises TCI-3) |
| CASE#4. | Input: 'Age = 36.7' | Expected result: 'Reject'. | (exercises TCI-4) |
| CASE#5. | Input: 'Age = w' | Expected result: 'Reject'. | (exercises TCI-5) |
| CASE#6. | Input: 'Age =&' | Expected result: 'Reject'. | (exercises TCI-6) |
| CASE#7. | Input: 'Age =77' | Expected result: 'Excess Warning'. | (exercises TCI-7 & TCI-1) |

These seven test cases will demonstrate that all the test coverage items were exercised, so achieving the coverage part of the test completion criterion.

## A.7 Test procedure (TD4)

For the test procedure, it is simply necessary to put the test cases in a logical order. There are no dependencies, so the test cases can be run in any order.

Imagine that it is also decided that TCI-1 is higher risk as it will be run most often. Also, that TCI-6 is higher risk as experience with the developers shows that their handling of special characters is not always perfect. Thus, it is desirable to run more test cases covering TCI-1 and TCI-6.

A simple 14-step test procedure can be generated that covers all equivalence partitions, while also addressing the concerns with the risks, as follows:

STEP-1: Set up test environment

STEP-2: Run CASE#1a ('Age =53')

STEP-3: Run CASE#1b ('Age =27')

STEP-4: Run CASE#1c ('Age =66')

STEP-5: Run CASE#1d ('Age =38')

STEP-6: Run CASE#2

STEP-7: Run CASE#3

STEP-8: Run CASE#4

STEP-9: Run CASE#5

STEP-10: Run CASE#6a ('Age =&')

STEP-11: Run CASE#6b ('Age =#')

STEP-12: Run CASE#6c ('Age =!')

STEP-13: Run CASE#7