INTERNATIONAL STANDARD

ISO/IEC/ IEEE 29119-2

First edition
2013-09-01

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 29119-2:2013(E)

ISO copyright office
Case postale 56
CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
E-mail inmail@iec.ch
Web www.iec.ch

Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York
NY 10016-5997, USA
E-mail stds.ipr@ieee.org
Web 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. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of ISO/IEC JTC 1 is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is called to the possibility that implementation of this standard may require the use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. ISO/IEEE is not responsible for identifying essential patents or patent claims for which a license may be required, for conducting inquiries into the legal validity or scope of patents or patent claims or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance or a Patent Statement and Licensing Declaration Form, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from ISO or the IEEE Standards Association.

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 Software & Systems Engineering Standards Committee of the IEEE Computer Society, under the Partner Standards Development Organization cooperation agreement between ISO and IEEE.

ISO/IEC 29119 consists of the following standards, under the general title *Software and systems engineering — Software testing*:

— *Part 1: Concepts and definitions*

— *Part 2: Test processes*

— *Part 3: Test documentation*

— *Part 4: Test techniques*

Introduction

The purpose of the ISO/IEC/IEEE series of software testing standards 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 supports dynamic testing, functional and non-functional testing, manual and automated testing, and scripted and unscripted testing. The processes defined in this series of international standards 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 24774:2010 Guidelines for Process Description, and covers the purpose, outcomes, activities, tasks and information items of each test process.

Testing is a key approach to risk mitigation in software development. This part of ISO/IEC/IEEE 29119 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.

The concepts and vocabulary that support this series of international standards are defined in ISO/IEC/IEEE 29119-1 Concepts and definitions. Templates and examples of test documentation that are produced during the testing process are defined in ISO/IEC/IEEE 29119-3 Test documentation. Software test design techniques that can be used during testing are defined in ISO/IEC/IEEE 29119-4 Test techniques.

This series of international standards aims to provide those responsible for software testing with the information required to manage and perform software testing in any organization.

Software and systems engineering — Software testing —

Part 2:
Test processes

1 Scope

This part of ISO/IEC/IEEE 29119 specifies test processes that can be used to govern, manage and implement software testing for any organization, project or smaller testing activity. It comprises generic test process descriptions that define the software testing processes. Supporting informative diagrams describing the processes are also provided.

This part of ISO/IEC/IEEE 29119 is applicable to testing in all software development lifecycle models.

This part of ISO/IEC/IEEE 29119 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 Conformance

2.1 Intended usage

The requirements in this part of ISO/IEC/IEEE 29119 are contained in Clauses 6 to 8. This part of ISO/IEC/IEEE 29119 provides requirements for a number of test processes suitable for use during the complete software lifecycle. It is recognized that particular projects or organizations may not need to use all of the processes defined by this part of ISO/IEC/IEEE 29119. Therefore, implementation of this part of ISO/IEC/IEEE 29119 typically involves selecting a set of processes suitable for the organization or project. There are two ways that an organization can claim to conform to the provisions of this part of ISO/IEC/IEEE 29119.

The organization shall assert whether it is claiming full or tailored conformance to this part of ISO/IEC/IEEE 29119:

2.1.1 Full conformance

Full conformance is achieved by demonstrating that all of the requirements (i.e. shall statements) of the full set of processes defined in this part of ISO/IEC/IEEE 29119 have been satisfied.

2.1.2 Tailored conformance

When this part of ISO/IEC/IEEE 29119 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, 7 and 8 of this part of ISO/IEC/IEEE 29119 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 (Information and documentation - Records management) or ISO 9001 (Quality management systems - Requirements) or

use similar internal organizational processes, they can decide to use those processes in place of the information item management tasks defined in this part of ISO/IEC/IEEE 29119.

3 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. 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 29119-1, *Software and systems engineering — Software testing — Part 1: Concepts and definitions*

ISO/IEC/IEEE 29119-3, *Software and systems engineering — Software testing — Part 3: Test documentation*

ISO/IEC/IEEE 29119-4, *Software and systems engineering — Software testing — Part 4: Test techniques*[1]

ISO/IEC 12207:2008, *Systems and software engineering — Software life cycle processes*

Other standards useful for the implementation and interpretation of this document are listed in the Bibliography.

4 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC/IEEE 24765 and the following apply.

NOTE Use of the terminology in this part of ISO/IEC/IEEE 29119 is for ease of reference and is not mandatory for conformance with this part of ISO/IEC/IEEE 29119. The following terms and definitions are provided to assist with the understanding and readability of this part of ISO/IEC/IEEE 29119. Only terms critical to the understanding of this part of ISO/IEC/IEEE 29119 are included. This clause is not intended to provide a complete list of testing terms. The Systems and Software Engineering Vocabulary ISO/IEC/IEEE 24765 can be referenced for terms not defined in this clause. This source is available at the following web site: http://www.computer.org/sevocab. All terms defined in this clause are also intentionally included in ISO/IEC/IEEE 29119-1, as that international standard includes all terms that are used in ISO/IEC/IEEE 29119-1,- 2, -3 and -4.

4.1
actual results
set of behaviours or conditions of a test item, or set of conditions of associated data or the test environment, observed as a result of test execution

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

4.2
completion criteria
conditions under which the testing activities are considered complete

4.3
coverage item
see test coverage item (4.33)

4.4
dynamic testing
testing that requires the execution of program code

---

[1] To be published.

4.5
equivalence partition
subset of the range of values of a variable, or set of variables, within a test item or at its interfaces such that all the values in the partition can reasonably be expected to be treated similarly by the test item (i.e. they may be considered "equivalent")

4.6
equivalence partition coverage
proportion of identified equivalence partitions of a test item that are covered by a test set

Note 1 to entry: In many cases, the identification of equivalence partitions is subjective (especially in the sub-partitioning of "invalid" partitions), so a definitive count of the number of equivalence partitions in a test item can be impossible.

4.7
equivalence partitioning
test design technique in which test cases are designed to exercise equivalence partitions by using one or more representative members of each partition

4.8
expected result
observable predicted behaviour of the test item under specified conditions based on its specification or another source

4.9
exploratory testing
type of unscripted experience-based testing in which the tester spontaneously designs and executes tests based on the tester's existing relevant knowledge, prior exploration of the test item (including the results of previous tests), and heuristic "rules of thumb" regarding common software behaviours and types of failure

Note 1 to entry: Exploratory testing hunts for hidden properties (including hidden behaviours) that, while quite possibly benign by themselves, can interfere with other properties of the software under test, and so constitute a risk that the software will fail.

4.10
feature set
logical subset of the test item(s) that could be treated independently of other feature sets in the subsequent test design activities

Note 1 to entry: This could be the set of all features for the item (its full feature set), or a subset identified for a specific purpose (the functional feature set, etc.).

4.11
Incident Report
documentation of the occurrence, nature, and status of an incident

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.

4.12
performance testing
type of testing conducted to evaluate the degree to which a test item accomplishes its designated functions within given constraints of time and other resource

4.13
Organizational Test Process
test process for developing and managing organizational test specifications

4.14
Organizational Test Policy
see Test Policy

4.15
Organizational Test Specification
document that provides information about testing 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 and Organizational Test Strategy.

4.16
Organizational Test Strategy
document that expresses the generic requirements for the testing to be performed on all the projects run within the organization, providing detail on how the testing is to be performed

Note 1 to entry: The Organizational Test Strategy is aligned with the Organizational Test Policy.

Note 2 to entry: An organization can have more than one Organizational Test Strategy to cover markedly different project contexts.

4.17
product risk
risk that a product can be defective in some specific aspect of its function, quality, or structure

4.18
project risk
risk related to the management of a project

EXAMPLE Lack of staffing, strict deadlines, changing requirements.

4.19
regression testing
testing following modifications to a test item or to its operational environment, to identify whether regression failures occur

Note 1 to entry: The sufficiency of a set of regression test cases depends on the item under test and on the modifications to that item or its operational environment.

4.20
retesting
re-execution of test cases that previously returned a "fail" result, to evaluate the effectiveness of intervening corrective actions

Note 1 to entry: Retesting is often combined with regression testing.

Note 2 to entry: Retesting is also known as confirmation testing.

4.21
risk-based testing
testing in which the management, selection, prioritisation, and use of testing activities and resources is consciously based on corresponding types and levels of analyzed risk

4.22
security testing
type of testing conducted to evaluate the degree to which a test item, 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

4.23
scripted testing
testing performed based on a documented test script

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

4.24
static testing
testing in which a test item is examined against a set of quality or other criteria without code being executed

EXAMPLE Reviews or static analysis.

4.25
stress testing
type of performance efficiency testing conducted to evaluate a test item's behaviour under conditions of loading above anticipated or specified capacity requirements, or of resource availability below minimum specified requirements

4.26
test basis
body of knowledge used as the basis for the design of tests and test cases

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.

Note 2 to entry: For specification-based testing the test basis is used to derive both test inputs and expected results, whereas for structure-based testing, the test basis is used solely for deriving expected results.

4.27
test case
set of test case preconditions, inputs (including actions, where applicable), and expected results, developed to drive the execution of a test item to meet test objectives, including correct implementation, error identification, checking quality, and other valued information

Note 1 to entry: A test case is the lowest level of test input (i.e. test cases are not made up of other test cases).

4.28
Test Case Specification
documentation of a set of one or more test cases

4.29
Test Completion Process
Test Management Process for ensuring that useful test assets are made available for later use, test environments are left in a satisfactory condition, and the results of testing are recorded and communicated to relevant stakeholders

4.30
Test Completion Report
report that provides a summary of the testing that was performed

4.31
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

Note 1 to entry: Test conditions can be used to derive coverage items, or can themselves constitute coverage items.

4.32
test coverage
degree, expressed as a percentage, to which specified test coverage items have been exercised by a test case or test cases

4.33
test coverage item
attribute or combination of attributes that is derived from one or more test conditions by using a test design technique that enables the measurement of the thoroughness of the test execution

4.34
test data
data created or selected to satisfy the input requirements for executing one or more test cases, which can be defined in the Test Plan, test case or test procedure

Note 1 to entry: Test data can be stored within the product under test (e.g., in arrays, flat files, or a database), or can be available from or supplied by external sources, such as other systems, other system components, hardware devices, or human operators.

4.35
Test Data Readiness Report
document describing the status of each test data requirement

4.36
Test Design and Implementation Process
test process for deriving and specifying test cases and test procedures

4.37
Test Design Specification
document specifying the features to be tested and their corresponding test conditions

4.38
test design technique
activities, concepts, processes, and patterns used to construct a test model that is used to identify test conditions for a test item, derive corresponding test coverage items, and subsequently derive or select test cases

4.39
test environment
facilities, hardware, software, firmware, procedures, and documentation intended for or used to perform testing of software

4.40
test environment readiness report
document that describes the status of each environment requirement

4.41
Test Environment Requirements
description of the necessary properties of the test environment

Note 1 to entry: All or parts of the test environment requirements can reference where the information can be found, e.g. in the appropriate Organizational Test Strategy, Test Plan, and/or Test Specification.

4.42
Test Environment Set-up Process
dynamic test process for establishing and maintaining a required test environment

4.43
test execution
process of running a test on the test item, producing actual results

4.44
Test Execution Log
document that records details of the execution of one or more test procedures

4.45
Test Execution Process
dynamic test process for executing test procedures created in the Test Design and Implementation Process in the prepared test environment, and recording the results

4.46
Test Incident Reporting Process
dynamic test process for reporting to the relevant stakeholders issues requiring further action that were identified during the test execution process

4.47
test item
work product that is an object of testing

EXAMPLE System, software item, requirements document, design specification, user guide.

4.48
test level
specific instantiation of a test sub-process

EXAMPLE The following are common test levels that can be instantiated as test sub-processes: component testing, integration testing, system testing and acceptance testing.

Note 1 to entry: Test levels are also known as test phases.

4.49
test management
planning, estimating, monitoring, reporting, control and completion of test activities

4.50
Test Management Process
test process for containing the sub-processes that are required for the management of a test project

Note 1 to entry: See test planning process, test monitoring and control process, test completion process.

4.51
Test Monitoring and Control Process
test management process for ensuring that testing is performed in line with a Test Plan and with organizational test specifications

4.52
test phase
specific instantiation of test sub-process

Note 1 to entry: Test phases are synonymous with test levels, therefore examples of test phases are the same as for test levels (e.g. system test phase/sub-process).

4.53
Test Plan
detailed description of test objectives to be achieved and the means and schedule for achieving them, organised to coordinate testing activities for some test item 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 activities can be defined in one or more test sub-process plans (i.e. a system test plan or a performance test plan).

Note 2 to entry: Test plans can also be written for non-project activities, for example a maintenance test plan.

4.54
Test Planning Process
Test Management Process used to complete test planning and develop Test Plans

4.55
Test Policy
an executive-level document that describes the purpose, goals, principles and scope of testing 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 organisations testing.

4.56
test procedure
sequence of test cases in execution order, and any associated actions that may be required to set up the initial preconditions and any wrap up activities post execution

Note 1 to entry: Test procedures include detailed instructions for how to run a set of one or more test cases selected to be run consecutively, including set up of common preconditions, and providing input and evaluating the actual results for each included test case.

4.57
Test Procedure Specification

document specifying one or more test procedures, which are collections of test cases to be executed for a particular objective

Note 1 to entry: The test cases in a test set are listed in their required order in the test procedure

Note 2 to entry: Also known as a manual test script. A test procedure specification for an automated test run is usually called a test script.

4.58
test process
used to provide information on the quality of a software product, often comprised of a number of activities, grouped into one or more test sub-processes

4.59
test result
indication of whether or not a specific test case has passed or failed, i.e. if the actual results corresponds to the expected results or if deviations were observed

4.60
test requirement
see test condition (4.31)

4.61
test script
test procedure specification for manual or automated testing

4.62
test set
collection of test cases for the purpose of testing a specific test objective

Note 1 to entry: The test sets will typically reflect the feature sets, but they could contain test cases for a number of feature sets.

Note 2 to entry: Test cases for a test set could be selected based on the identified risks, test basis, retesting and/or regression testing.

4.63
test specification
complete documentation of the test design, test cases and test procedures for a specific test item

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.

4.64
test specification technique
see test design technique (4.38)

4.65
Test Status Report
report that provides information about the status of the testing that is being performed in a specified reporting period

4.66
test strategy
part of the Test Plan that describes the approach to testing for a specific test project or test sub-process or sub-processes

Note 1 to entry: The test strategy is a distinct entity from the Organizational Test Strategy.

Note 2 to entry: The test strategy usually describes some or all of the following; the test sub-processes to be implemented; the retesting and regression testing to be employed; the test design techniques and corresponding test completion criteria to be used; test data; test environment and testing tool requirements; and expectations for test deliverables.

4.67
test sub-process
test management and dynamic (and static) test processes used to perform a specific test level (e.g. system testing, acceptance testing) or test type (e.g. usability testing, performance testing) normally within the context of an overall test process for a test project

Note 1 to entry: A test sub-process can comprise one or more test types. Depending on the life cycle model used, test sub-processes are also typically called test phases, test levels, test stages or test tasks.

4.68
test technique
see test design technique (4.38)

4.69
test type
group of testing activities that are focused on specific quality characteristics

Note 1 to entry: A test type can be performed in a single test sub-process or can be performed across a number of test sub-processes (e.g. performance testing completed at a component test sub-process and also completed at a system test sub-process).

EXAMPLE Security testing, functional testing, usability testing, and performance testing.