INTERNATIONAL STANDARD

ISO/IEC 30130

First edition
2016-02-01

Software engineering — Capabilities of software testing tools

Ingénierie du logiciel — Capacités des outils d'essai de logiciel

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

Reference number
ISO/IEC 30130:2016(E)

© ISO/IEC 2016

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2016, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

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

 © ISO/IEC 2016 – All rights reserved

Contents

Page

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

© ISO/IEC 2016 – All rights reserved

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

 © ISO/IEC 2016 – All rights reserved

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.

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 document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

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).

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

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, SC 7, *Software and systems engineering*.

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

© ISO/IEC 2016 – All rights reserved

Introduction

This International Standard defines the framework to which capabilities of software testing tools are allocated in order to identify the capabilities of products being used by any project for software testing. To develop high-quality software with reasonable time and budget, the use of software testing tools is required. The increase in the size and complexity of software is complemented by an increase in the difficulty and complexity of software testing. This created a larger demand for the support of tools in order to test software effectively and efficiently.

Testing tools are highly diverse due to their contexts of use. Testing itself varies by objective, such as functional testing or nonfunctional testing, and granularity, such as unit testing or system testing. Testing tool vendors vary by providing tools with a different function or combination of functions. And despite vendor provided explanations for the type of testing support functions, there is little common understanding of these functions. In this environment, it is difficult to utilize a testing tool that is suitable for a project without common understanding of tool functions, proper acquisition of the needed tools, and efficient training.

The framework defined by this International Standard consists of objectives of testing, granularity of software to be tested and capabilities. In Clause 4, an object model for software testing tools as basis for the framework is defined. In Clause 5, three of the categories in that software testing model (Dynamic Test Execution, Code Analysis, and Test Management) are specified. In Clause 6, quality characteristics, granularity, and other aspects of characteristics are defined and in Clause 7, tool capabilities are mapped onto those categories and characteristics.

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

© ISO/IEC 2016 – All rights reserved

Software engineering — Capabilities of software testing tools

1 Scope

This International Standard defines the framework to which capabilities of software testing tools are allocated in order to identify the capabilities of products being used by any project for software testing. Software testing processes are identified in ISO/IEC/IEEE 29119-2 and software verification processes are identified in ISO/IEC 12207. This International Standard is fully harmonized with these existing standards in terms of software testing processes.

This International Standard focuses on the following areas that the existing ISO/IEC standards do not deal with the following:

— categorization of software test entities and software testing tools (Clauses 4 and 5);

— characterization of each software testing tool category (Clauses 5 and 6);

— mapping of software testing tool capabilities and characteristics (Clauses 6 and 7).

2 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 25010, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*

3 Terms and definitions

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

3.1
quality record report
report which is generated through dynamic test execution and code analysis to record test results and other output

Note 1 to entry: Including Test Result, Static Code Analysis Report, Test Incident Report, and Metrics.

3.2
test target version
specific version of test target which is used for one-time execution of Dynamic Test Execution or Code Analysis

3.3
testing tool
specific or generic tool which is used for test execution and test management such as test results recording, test results display, test results interpretation, generation of test data, generation of test procedure, generation of test scripts, test modelling, etc.

3.4
package
namespace for the grouped elements

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

© ISO/IEC 2016 – All rights reserved

4 Object model for software testing tools

4.1 Overview of the object model

A software testing tool is described by its input, process, output and environment. To define software testing tools, the object model for software testing is identified.

The object model for software testing tools consists of the following elements:

a) test process, which represents processes of dynamic test;

b) code analysis process, which represents processes of code analysis;

c) test entity, which represents entities that appear in the process;

d) test tool, which supports inputs, processes, outputs and test entity.

Test Process comprises multiple subprocesses. Each process has input and output. These input and output are basically test entities. Test processes are specified in detail by ISO/IEC/IEEE 29119-2.

Code Analysis Process comprises multiple sub processes. Each process has input and output. These input and output are basically test entities.

Test Entity comprises multiple entities. The entity is referred to as Dynamic Test Execution Entity that is required or created in Dynamic Test Process. The entity is referred to as Code Analysis Entity that is required or created in Code Analysis Process. A test entity represents a single performance of a test, in which the target of the test is tested in various respects. Testing has two modes of execution. Dynamic Test Execution involves actual execution of the code and Code Analysis involves examination of the source code of the target.

The testing tools take test entities as input and produce test entities. By producing test entities, testing tools effectively support the testing process.

The object model diagrams, Figures 1 to 5, are described using UML 2 (ISO/IEC 19505-2).

Test Process includes Dynamic Test Process and Test Management Process.

Code Analysis Process includes Code Analysis Process and Code Analysis Management Process.

Test Entity includes Test Management Entity, Dynamic Test Execution Entity and Code Analysis Entity, and Test Target.

Test Tool includes Test Execution Tool and Test Management Tool.

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

© ISO/IEC 2016 – All rights reserved

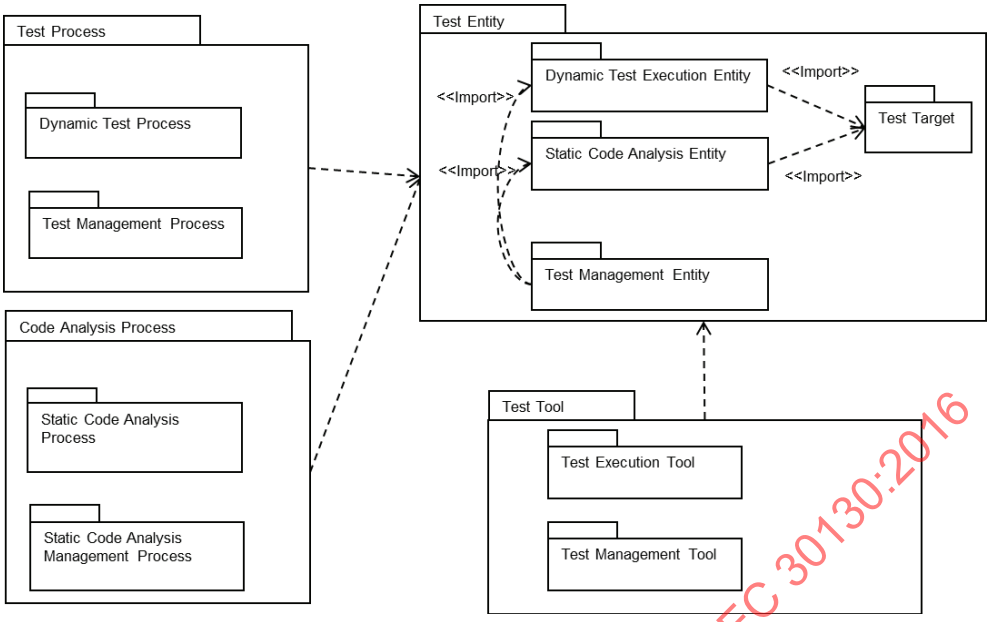

Figure 1 — Object model of software testing

Test Process and Test Tool have indirect relationships which are described in Annex A.

4.2 Test target

Test Target is a set of source code and executable code of software that is in development. It has a hierarchical structure, and any part of the structure or the whole structure is tested.

Test Target Version is a specific version of the software.

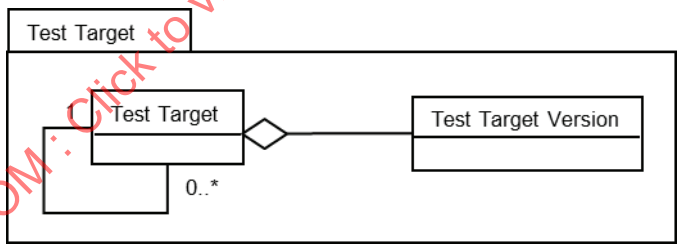

Figure 2 — Object model of the package "Test Target"

4.3 Dynamic test execution entity

Dynamic Test Execution Entity is prepared to identify necessary entities which are used in one-time dynamic test execution. Test Data, Test Target Version, Test Result, and Expected Result are identified in this package.

In one-time dynamic test execution, Test Environment is used.

Test Target Version which is used in one-time dynamic test execution is extracted from an appropriate granularity and version of Test Target.

Test Data which is used in one-time dynamic test execution is designed for Test Case which is derived from Specification and is stored on Test Data Collections in Test Data Repository.

Specification, Test Case, and Test Target are referred to as input for Dynamic Test Execution.

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

© ISO/IEC 2016 – All rights reserved

Test Data Repository has Test Data which is used by all Dynamic Test Execution and is classified by Objective and Technique.

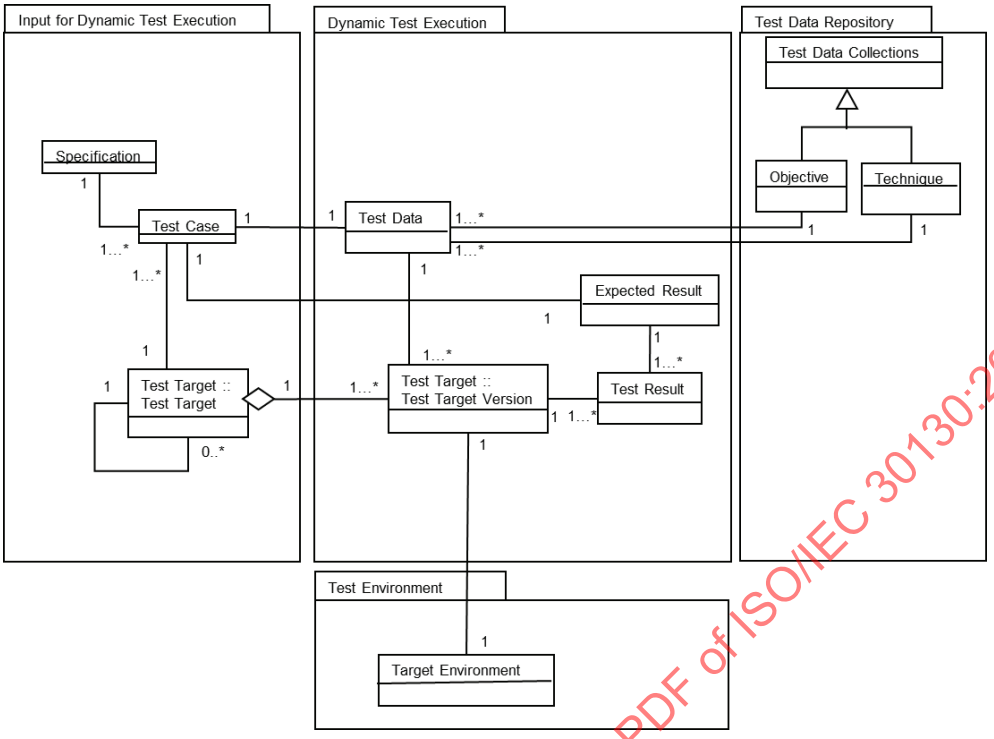

Figure 3 — Object model of the package "Dynamic Test Execution Entity"

4.4 Code analysis entity

Code Analysis Entity is prepared to identify necessary entities which are used in one-time code analysis. Test Target Version and Code Analysis Result are identified in this package.

Test Target Version which is used in one-time code analysis is extracted from an appropriate granularity and version of Test Target.

Test Target and Check List which are used in Code Analysis are referred to as Input for Code Analysis. Check List usually does not depend on Specification.

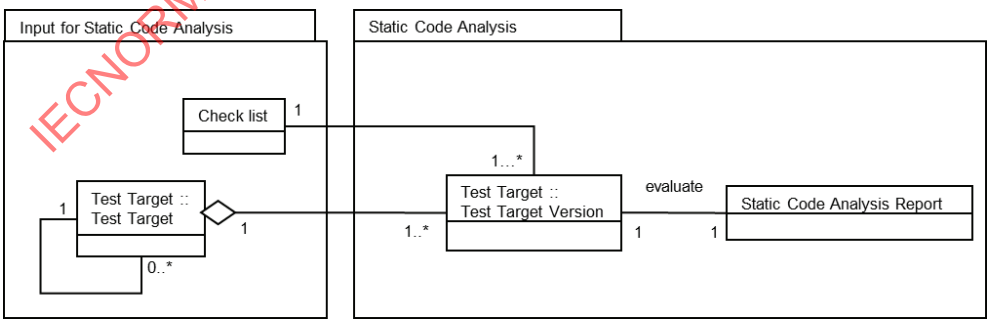

Figure 4 — Object model of the package "Code Analysis Entity"

4.5 Test management entity

Test Management Entity is prepared to identify necessary packages and entities which are used to manage the entire test processes. Test Plan, Test Asset, Quality Record Report, Test Completion Report ,Verification and Validation Report,and Test Status Report are identified in this package.

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

© ISO/IEC 2016 – All rights reserved

Dynamic Test Execution Documentation and Static Code Analysis Report are referred to as Quality Record Report.

Dynamic Test Execution Documentation is composed by Metric, Test Result, and Test Incident Report. Static Code Analysis Report is composed by Metric, Static Code Analysis Report, and Test Incident Report.

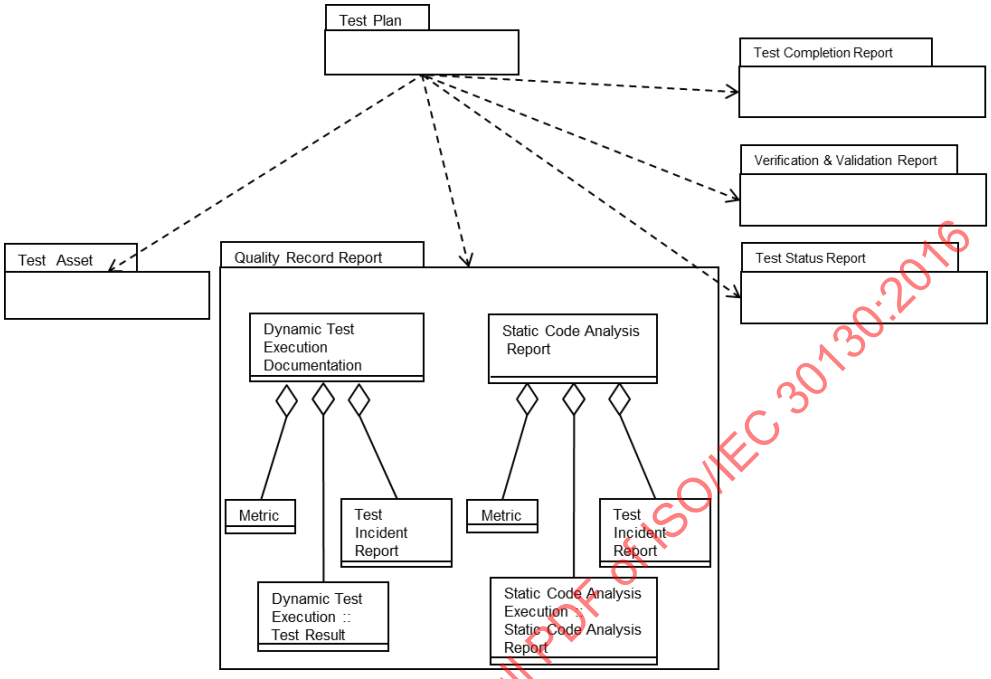

Figure 5 — Object model of the package "Test Management Entity"

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

5 Category of test entity

5.1 Overview

In this Clause, categories of test entities are defined to allocate and specify the capabilities of software tools. Each category is identified by analyzing objective, life history, and usage of each test entity.

5.2 Categories of dynamic test execution entities

5.2.1 Input for dynamic test execution

Entities in this category are used in Dynamic Test Execution as inputs, such as Specification, Test Case, and Test Target.

5.2.2 Dynamic test execution

Entities in this category are used in one-time dynamic testing. It comprises Test Data, Test Result, Expected Result, and Test Target Version.

5.2.3 Test data repository

Entities in this category are used in Dynamic Test Execution. It comprises Test Data Collections, in which there are a large number of test data for the overall Dynamic Test Execution. Test Data Collections can be subsets which depend on each objective and techniques.

© ISO/IEC 2016 – All rights reserved

5.2.4 Test environment

An entity, Target Environment, in this category is required to define a condition to evaluate Test Target Version by one-time dynamic testing, such as hardware, operating system, middleware and test drivers, etc.

5.3 Categories of code analysis entities

5.3.1 Input for code analysis

Entities in this category are used in Code Analysis as inputs, such as Check List and Test Target.

5.3.2 Code analysis

Entities in this category are used in one-time code analysis. It comprises of Test Target Version and Code Analysis Result.

5.4 Categories of test management entities

5.4.1 Test plan

This category is used for testing that can be performed given constraints on personnel, material and budget.

5.4.2 Test asset

Entities in this category is identified as a list of Test Data Collections, Test Target Version, and Check List, which may be reused (e.g. for regression testing).

5.4.3 Quality record report

The entities in this category are created thorough test executions. It comprises Dynamic Test Execution Documentation and Static Code Analysis Report.

5.4.4 Test completion report

The entities in this category provides a summary of the testing that was performed. This may be for the project/programme as a whole or for the particular test sub-process.

5.4.5 Verification and validation report

This category is used for reporting verification and validation items, on which the final version of the Test Completion Report is based.

5.4.6 Test status report

This category is used for reporting the status of testing that is performed in a specified reporting period.

6 Characteristics of software testing tools

6.1 Overview

Test objectives and test granularity are the key drivers of test scope. Other drivers of test scope, such as test personnel, cost, and schedule constraints also have an effect on the scope of testing.

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

 © ISO/IEC 2016 – All rights reserved

The most important aspect of characteristics is objectives of the software test, software quality. This aspect is defined by SQuaRE (ISO/IEC 25010) and top level of the characteristics is explained in 6.2.

The second important aspect of characteristics is a size of component to be tested, granularity. This aspect is defined in 6.3.

For other aspects of characteristics, the reader is suggested to add those aspects if necessary.

6.2 Quality characteristics

Quality characteristics which a test target shall satisfy are defined as described below by SQuaRE (ISO/IEC 25010).

a) Functional Suitability

degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions:

— functional completeness;

— functional correctness;

— functional appropriateness.

b) Reliability

degree to which a system, product or component performs specified functions under specified conditions for a specified period of time:

— maturity;

— availability;

— fault tolerance;

— recoverability.

c) Usability

degree to which a product or system can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

— appropriateness recognizability;

— learnability;

— operability;

— user error protection;

— user interface aesthetics;

— accessibility.

d) Performance Efficiency

performance relative to the amount of resources used under sated conditions:

— time behavior;

— resource utilization;

— capacity.

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

© ISO/IEC 2016 – All rights reserved

e) Maintainability

degree of effectiveness and efficiency with which a product or system can be modified by the intended maintainers.

— modularity;

— reusability;

— analysability;

— modifiability;

— testability.

f) Portability

degree of effectiveness and efficiency with which a system, product or component can be transferred from one hardware, software or other operational or usage environment to another.

— adaptability;

— installability;

— replaceability.

g) Compatibility

degree to which product, system or component can exchange information with other products, system or components, and/or perform its required functions, while sharing the same hardware or software environment:

— co-existence;

— interoperability.

h) Security

degree to which a product or system protects information and data so that persons or other products or systems have the degree of data access appropriate to their types and levels of authorization.

— confidentiality;

— integrity;

— non-repudiation;

— accountability;

— authenticity.

6.3 Granularity

Granularity which is defined in this international standard is a size of a test target. The size of a test target through test phases varies from small pieces to a whole system.

a) Smallest Unit

Smallest Unit is defined as a unit for smallest subsets of the test target.

Modules, components, functions, subroutines or other such atomic units of description are often used as the Smallest Unit.

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

 © ISO/IEC 2016 – All rights reserved

b) Intermediate Unit

Intermediate Unit is defined as a unit for combination of Smallest Units of the test target.

c) Largest Unit

Largest Unit is defined as a unit for whole system of the test target.

NOTE Test levels usually correspond to the granularity, such as unit test, integration test, system test and acceptance test. This standard covers any test level and test type.

6.4 Other aspects of characteristics

In addition to quality characteristics and granularity, other aspects of characteristics may have impact on a scope of testing. For example, if a system to be tested is required to be of very high quality, many iterations of regression testing are required. In such case, priority for tool capabilities is different from ordinary projects. The reader is encouraged to investigate additional aspects of characteristics such as integrity levels, costs, and schedule constraints.

7 Capabilities of software testing tools

7.1 Overview

This Clause describes capabilities of software testing tools. Each capability is mapped onto the categories which are defined in Clause 5 and the characteristics which are defined in Clause 6.

NOTE Capabilities defined in this Clause are sourced from International Software Testing Qualifications Board (ISTQB), IEEE 1012, and ISO/IEC/IEEE 29119-1.

7.2 Dynamic test execution

7.2.1 Input for dynamic test execution

This includes the following capabilities.

a) Test design

This includes capabilities that support test design. This capability creates test specifications (test design specifications, test case specifications and test procedure specifications).

This capability is applicable to all eight Quality characteristics.

This capability is applicable to all three levels of granularity.

b) Risk-based priority

This includes capabilities that manage test priorities based on risks of the software.

This capability is applicable to the following Quality characteristics:

— functional suitability;

— reliability;

— security.

This capability is applicable to the following level of granularity:

— largest unit.

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

© ISO/IEC 2016 – All rights reserved

7.2.2 Dynamic Test Execution

This includes the following capabilities.

a) Test execution control and automated test execution

This includes capabilities for testing of test targets, and generating the actual results.

This capability is applicable to all eight Quality characteristics.

This capability is applicable to all three levels of granularity.

b) Capture and playback

This includes capabilities that capture input during manual testing and repeat same testing automatically. The capabilities are usually used for automated regression testing.

This capability is applicable to all eight Quality characteristics.

This capability is applicable to all three levels of granularity.

c) Keyword driven test case

This includes capabilities that separate the documentation of test cases and test data from the prescription of the way the test cases are executed

This capability is applicable to the following Quality characteristics:

— functional suitability.

This capability is applicable to all three levels of granularity.

d) Test comparator

This includes capabilities that compare actual result of testing and expected result.

This capability is applicable to the following Quality characteristics:

— functional suitability.

This capability is applicable to all three levels of granularity.

e) Debugging

This includes capabilities that halt an executed program and run a program step by step. This capability is applicable to reproducing failures and discovering defects.

This capability is applicable to all eight Quality characteristics.

This capability is applicable to all three levels of granularity.

f) Dynamic analysis

This includes capabilities for providing information on the state of the software code at runtime. It is mostly used for unassigned pointer detection, checkpoint calculation, memory allocation, and memory leak detection.

This capability is applicable to the following Quality characteristics:

— performance efficiency.

This capability is applicable to all three levels of granularity.

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

 © ISO/IEC 2016 – All rights reserved

g) Monitoring

This includes capabilities for monitoring, recording and analyzing the behavior of the test target.

This capability is applicable to the following Quality characteristics:

— functional suitability;

— reliability;

— performance efficiency.

This capability is applicable to all three levels of granularity.

h) Coverage measurement

This includes capabilities that report coverage of tested code.

This capability is applicable to the following Quality characteristics:

— functional suitability.

This capability is applicable to the following level of granularity:

— smallest unit.

NOTE Coverage measurement: ISO/IEC/IEEE 29119-4.

i) Security testing

This includes capabilities that support testing for security vulnerabilities.

This capability is applicable to the following Quality characteristics:

— security.

This capability is applicable to all three levels of granularity.

7.2.3 Test data repository

This includes the following capabilities.

a) Test data preparation and test data generation

This includes capabilities that prepare data for testing by selecting (actual data from a database, etc.), creating, generating or editing such data.

This capability is applicable to all eight Quality characteristics.

This capability is applicable to all three levels of granularity.

b) Stress testing and load testing

This includes capabilities that evaluate a system beyond the limits of its specified workloads.

This capability is applicable to the following Quality characteristics:

— reliability;

— performance efficiency.

This capability is applicable to all three levels of granularity.

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

© ISO/IEC 2016 – All rights reserved

c) Performance testing

This includes capabilities that generate workload and measure response time of transactions to evaluate software product performance.

This capability is applicable to the following Quality characteristics:

— performance efficiency.

This capability is applicable to all three levels of granularity.

d) Data validation and verification

This includes capabilities for determining the validity and correctness of data.

This capability is applicable to the following Quality characteristics:

— functional suitability;

— userbility;

— performance efficiency;

— security.

This capability is applicable to the following level of granularity:

— intermediate units;

— largest unit.

e) Database validation and verification

This includes capabilities for determining the validity and correctness of databases such as checking for character code, database linking, and compatibility between sets of data, etc.

This capability is applicable to the following Quality characteristics:

— functional suitability;

— performance efficiency;

— security.

This capability is applicable to the following level of granularity:

— intermediate units;

— largest unit.

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

7.2.4 Test Environment

This includes the following capabilities.

a) Emulators and simulators

This includes capabilities that produce the same output as a particular system by accepting the same input.

This capability is applicable to the following Quality characteristics:

— functional suitability;

— performance efficiency.

© ISO/IEC 2016 – All rights reserved

This capability is applicable to all three levels of granularity.

b) Unit test framework

This includes capabilities that provide an environment for unit testing in which a unit can be tested with suitable stubs and drivers.

This capability is applicable to the following Quality characteristics:

— functional suitability.

This capability is applicable to the following level of granularity:

— smallest unit.

c) Automated environment setup

This includes capabilities that automatically install software environment including test data.

This capability is applicable to all of eight Quality characteristics.

This capability is applicable to the following level of granularity:

— intermediate units;

— largest unit.

d) Runtime environment management

This includes capabilities that manage the information of multiple hardware and software configurations for testing.

This capability is applicable to all of eight Quality characteristics.

This capability is applicable to the following level of granularity:

— intermediate units;

— largest unit.

7.3 Code analysis

7.3.1 Input for code analysis

This includes the following capabilities.

a) Code review

This includes capabilities that support review processes such as management review, informal review, technical review, inspection, and walkthrough.

This capability is applicable to all of eight Quality characteristics.

This capability is applicable to all three levels of granularity.

7.3.2 Code analysis

This includes the following capabilities.

a) Code analyzer

This includes capabilities that analyse source code without executing the software product.

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

© ISO/IEC 2016 – All rights reserved

This capability is applicable to all eight Quality characteristics.

This capability is applicable to all three levels of granularity.

b) Code-based security testing

This includes capabilities that support testing for security vulnerabilities without executing the software product.

This capability is applicable to the following Quality characteristics:

— security.

This capability is applicable to the following level of granularity:

— smallest unit.

7.4 Test management

7.4.1 Test plan

This includes the following capabilities.

a) Test management

This includes capabilities that support test planning, monitoring, and control such as testware management, scheduling of tests, the logging of results, progress tracking and test reporting.

This capability is applicable to all eight Quality characteristics.

This capability is applicable to all three levels of granularity.

7.4.2 Test asset

This includes the following capabilities.

a) Test Asset configuration management

This includes capabilities that support maintenance of test asset such as test cases ,test data, check list, and test target.

This capability is applicable to all eight Quality characteristics.

This capability is applicable to all three levels of granularity.

7.4.3 Quality record report

This includes the following capabilities.

a) Incident management

This includes capabilities that support incident recording and status tracking. This capability usually has workflow-oriented functions.

This capability is applicable to all eight Quality characteristics.

This capability is applicable to all three levels of granularity.

b) Defect management, defect tracking, bug tracking

This includes capabilities that perform defect recording, classification, identification and handling of defects.

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

 © ISO/IEC 2016 – All rights reserved

This capability is applicable to all eight Quality characteristics.

This capability is applicable to all three levels of granularity.

c) Test management

This includes capabilities that support test planning, monitoring, and control such as testware management, scheduling of tests, logging of results, progress tracking and test reporting.

This capability is applicable to all eight Quality characteristics.

This capability is applicable to all three levels of granularity.

NOTE This capability is used during the project period.

7.4.4 Test completion report

This includes the following capabilities.

a) Test management

This includes capabilities that support test planning, monitoring, and control such as testware management, scheduling of tests, logging of results, progress tracking and test reporting.

This capability is applicable to all eight Quality characteristics.

This capability is applicable to all three levels of granularity.

NOTE This capability provides information for the test completion report at the end of the project period.

7.4.5 Verification and validation report

This includes the following capabilities.

a) Relating of data that serves as the basis for Verification and Validation Reports

This includes capabilities that gather information item of the Verification and Validation Report on which the final version of the Test Completion Report is based.

This capability is applicable to all eight Quality characteristics.

This capability is applicable to all three levels of granularity.

b) Verification and Validation Report

This includes capabilities that provide preparation of comprehensive test evaluation results.

This capability is applicable to all eight Quality characteristics.

This capability is applicable to all three levels of granularity.

7.4.6 Test status report

This includes the following capabilities.

a) Test monitoring

This includes capabilities that support test management tasks for dealing with activities related to periodic checking of test project status such as comparing actual activities with planned activities.

This capability is applicable to all eight Quality characteristics.

This capability is applicable to all three levels of granularity.

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

© ISO/IEC 2016 – All rights reserved

7.5 Summary of capabilities with characteristics

The categories and characteristics for the entire set of capabilities are summarized as below.

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

© ISO/IEC 2016 – All rights reserved

Table 1 — Summary of capabilities with characteristics

Capabilities	Categories												Characteristics										
	Dynamic test execution			Test environment	Code analysis		Test management						Software quality characteristics								Granularity		
	Input for dynamic test execution	Dynamic test execution	Test data repository	Test environment	Input for code analysis	Code analysis	Test plan	Test asset	Quality record report	Test completion report	Verification and validation	Test status report	Functional suitability	Reliability	Usability	Performance efficiency	Maintainability	Portability	Compatibility	Security	Smallest unit	Intermediate units	Largest unit
Test design	0												0	0	0	0	0	0	0	0	0	0	0
Risk Based Priority	0												0	0		0				0	0	0	0
Test execution controll, Automated test execution		0											0	0	0	0	0	0	0	0	0	0	0
Capture, Playback		0											0	0	0	0	0	0	0	0	0	0	0
Keyword driven test case		0											0									0	0
Test comparator		0											0										0
Debugging		0											0								0	0	0
Dynamic analysis		0											0			0					0	0	0
Monitoring		0											0	0		0					0	0	0
Coverage measurement		0											0								0		
Security testing		0																		0			0
Test data preparation, Test data generation			0										0	0	0	0	0	0	0	0	0	0	0
Stress testing, Load testing			0											0		0							0
Performance testing			0													0					0	0	0
Data validation and verification			0										0			0				0		0	0
Database validation and verification			0										0			0				0		0	0
Emulators, Simulators				0									0			0					0	0	0
Unit test framework				0									0								0		

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

© ISO/IEC 2016 – All rights reserved

Table 1 *(continued)*

Capabilities	Categories												Characteristics										
	Dynamic test execution			Test environment	Code analysis		Test management						Software quality characteristics								Granularity		
	Input for dynamic test execution	Dynamic test execution	Test data repository	Test environment	Input for code analysis	Code analysis	Test plan	Test asset	Quality record report	Test completion report	Verification and validation	Test status report	Functional suitability	Reliability	Usability	Performance efficiency	Maintainability	Portability	Compatibility	Security	Smallest unit	Intermediate units	Largest unit
Automated environment setup				0									0	0	0	0	0	0	0	0	0	0	0
Runtime envrionment management				0									0	0	0	0	0	0	0	0		0	0
Review					0								0	0	0	0	0	0	0	0	0	0	0
Code analyzer						0								0			0	0		0	0		
Code-based security testing						0														0	0		
Test management							0			0			0	0	0	0	0	0	0	0	0	0	0
Test Asset configuration management								0					0	0	0	0	0	0	0	0	0	0	0
Incident management									0				0	0	0	0	0	0	0	0	0	0	0
Defect management, Defect tracking, Bug tracking									0				0	0	0	0	0	0	0	0	0	0	0
Test monitoring												0	0	0	0	0	0	0	0	0	0	0	0
Relating of data that serves as the basis for Verification and Validation Reports											0		0	0	0	0	0	0	0	0	0	0	0
Verification and Validation Report											0		0	0	0	0	0	0	0	0	0	0	0

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

© ISO/IEC 2016 – All rights reserved

Annex A
(informative)

Capabilities mapping to test process

A.1 Overview

This Annex describes a mapping of capabilities of software testing tools and the processes of ISO/IEC/IEEE 29119-2.

This Annex uses an abbreviated name of activity for all names of activities in the processes which are defined in ISO/IEC/IEEE 29119-2. They are listed in Table A.1.

Table A.1 — Abbreviated name of activities

Name of process	Name of activity	Abbreviated name of activity
Test Planning Process	Understand Context	TP1
	Organize Test Plan Development	TP2
	Identify and Analyze Risks	TP3
	Identify Risk Mitigation Approaches	TP4
	Design Test Strategy	TP5
	Determine Staffing and Scheduling	TP6
	Record Test Plan	TP7
	Gain Consensus on Test Plan	TP8
	Communicate Test Plan and Make Available	TP9
Test Monitoring and Control Process	Set-Up	TMC1
	Monitor	TMC2
	Control	TMC3
	Report	TMC4
Test Completion Process	Archive Test Assets	TC1
	Clean Up Test Environment	TC2
	Identify Lessons Learned	TC3
	Report Test Completion	TC4
Test Design and Implementation Process	Identify Feature Sets	TD1
	Derive Test Conditions	TD2
	Derive Test Coverage Items	TD3
	Derive Test Cases	TD4
	Assemble Test Sets	TD5
	Derive Test Procedures	TD6
Test Environment Set-Up and Maintenance Process	Establish Test Environment	ES1
	Maintain Test Environment	ES2

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

© ISO/IEC 2016 – All rights reserved

Table A.1 *(continued)*

Name of process	Name of activity	Abbreviated name of activity
Test Execution Process	Execute Test Procedures	TE1
	Compare Test Results	TE2
	Record Test Execution	TE3
Test Incident Reporting Process	Analyze Test Results	IR1
	Create/Update Incident Report	IR2

A.2 Capability mapping to " Organizational Test Process "

There is no capability mapping to "Organizational Test Process".

A.3 Capability mapping to "Test Planning Process"

Capability mapping to "Test Planning Process" is shown as Table A.2. 'X' is marked in the cell when a capability of software testing tools is used in an activity of "Test Planning Process".

Table A.2 — Capability mapping to "Test Planning Process"

Activity / Capability	TP 1	TP 2	TP3	TP 4	TP 5	TP 6	TP 7	TP 8	TP 9
7.2.1 a) Test design									
7.2.1 b) Risk-based priority	—	—	X	—	X	X	—	—	—
7.2.2 a) Test execution control and automated test execution	—	—	—	—	—	—	—	—	—
7.2.2 b) Capture and playback	—	—	—	—	—	—	—	—	—
7.2.2 c) Keyword-driven test case	—	—	—	—	—	—	—	—	—
7.2.2 d) Test comparator	—	—	—	—	—	—	—	—	—
7.2.2 e) Debugging	—	—	—	—	—	—	—	—	—
7.2.2 f) Dynamic analysis	—	—	—	—	—	—	—	—	—
7.2.2 g) Monitoring	—	—	—	—	—	—	—	—	—
7.2.2 h) Coverage measurement	—	—	—	—	—	—	—	—	—
7.2.2 i) Security testing	—	—	—	—	—	—	—	—	—
7.2.3 a) Test data preparation and test data generation	—	—	—	—	—	—	—	—	—
7.2.3 b) Stress testing and load testing	—	—	—	—	—	—	—	—	—
7.2.3 c) Performance testing	—	—	—	—	—	—	—	—	—
7.2.3 d) Data validation and verification	—	—	—	—	—	—	—	—	—
7.2.3 e) Database validation and verification	—	—	—	—	—	—	—	—	—
7.2.4 a) Emulators and simulators	—	—	—	—	—	—	—	—	—
7.2.4 b) Unit test framework	—	—	—	—	—	—	—	—	—
7.2.4 c) Automated environment setup	—	—	—	—	—	—	—	—	—
7.2.4 d) Runtime environment management	—	—	—	—	—	—	—	—	—
7.3.1 a) Review	—	—	—	—	—	—	—	—	—
7.3.2 a) Code analyzer, analyzer	—	—	—	—	—	—	—	—	—
7.3.2 b) Code-based security testing	—	—	—	—	—	—	—	—	—

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

© ISO/IEC 2016 – All rights reserved

Table A.2 *(continued)*

Activity / Capability	TP 1	TP 2	TP3	TP 4	TP 5	TP 6	TP 7	TP 8	TP 9
7.4.1 a) Test management	—	X	—	—	X	X	X	—	—
7.4.2 a) Test Asset configuration management	—	—	—	—	—	—	—	—	—
7.4.3 a) Incident management	—	—	—	—	—	—	—	—	—
7.4.3 b) Defect management, defect tracking, bug tracking	—	—	—	—	—	—	—	—	—
7.4.3 c) Test management	—	—	—	—	—	—	—	—	—
7.4.4 a) Test management	—	—	—	—	—	—	—	—	—
7.4.5 a) Relating of data that serves as the basis for Verification and Validation Reports	—	—	—	—	—	—	—	—	—
7.4.5 b) Verification and Validation Report	—	—	—	—	—	—	—	—	—
7.4.6 a) Test monitoring	—	—	—	—	—	—	—	—	—

A.4 Capability mapping to "Test monitoring and control process" and "Test completion process"

Capability mapping to "Test monitoring and control process" and "Test completion process" is shown as Table A.3. 'X' is marked in the cell when a capability of software testing tools is used in an activity of "Test monitoring and control process" and "Test completion process".

Table A.3 — Capability mapping to "Test Monitoring and Control Process" and "Test Completion Process"

Activity / Capability	TMC 1	TMC 2	TMC 3	TMC 4	TMC 1	TMC 2	TMC 3	TMC 4
7.2.1 a) Test design	—	—	—	—	—	—	—	—
7.2.1 b) Risk-based priority	X	X	—	—	—	—	—	—
7.2.2 a) Test execution control and automated test execution	—	—	—	—	—	—	—	—
7.2.2 b) Capture and playback	—	—	—	—	—	—	—	—
7.2.2 c) Keyword-driven test case	—	—	—	—	—	—	—	—
7.2.2 d) Test comparator	—	—	—	—	—	—	—	—
7.2.2 e) Debugging	—	—	—	—	—	—	—	—
7.2.2 f) Dynamic analysis	—	—	—	—	—	—	—	—
7.2.2 g) Monitoring	—	X	—	—	—	—	—	—
7.2.2 h) Coverage measurement	—	—	X	X	—	—	—	—
7.2.2 i) Security testing	—	—	—	—	—	—	—	—
7.2.3 a) Test data preparation and test data generation	—	—	—	—	—	—	—	—
7.2.3 b) Stress testing and load testing	—	—	—	—	—	—	—	—
7.2.3 c) Performance testing	—	—	—	—	—	—	—	—
7.2.3 d) Data validation and verification	—	—	—	—	—	—	—	—
7.2.3 e) Database validation and verification	—	—	—	—	—	—	—	—
7.2.4 a) Emulators and simulators	—	—	—	—	—	—	—	—

IECNORM.COM : Click to view the full PDF of ISO/IEC 30130:2016

© ISO/IEC 2016 – All rights reserved