International Standard

ISO/IEC 29110-5-1-2

First edition
2025-02

Systems and software engineering — Life cycle profiles for very small entities (VSEs) —

Part 5-1-2: Software engineering guidelines for the generic Basic profile

Reference number
ISO/IEC 29110-5-1-2:2025(en)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.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 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 or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

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.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*.

This first edition replaces the ISO/IEC TR 29110-5-1-2:2011, which has been technically revised.

The main changes are as follows:

— Many task statements have been reworded to facilitate their understanding.

— Conditional tasks have been added to develop optional work products (e.g. operation guide) that have been requested by a customer. This notation replaces the 'Optional' notation [e.g. *(optional)] used in the TR that caused ambiguities.

— Terms have been added to Clause 3 such that this document is self-contained.

— A few terms have been modified to align this document with the updated version of standards such as the ISO/IEC/IEEE 12207 and the ISO/IEC/IEEE 15289.

— Texts have been added for giving additional information intended to assist the understanding or use of the text of the document.

— Annex A has been added to describe a process, which can be added to the software Basic profile, to enable VSEs to support the software product which they developed.

— Annex B has been added to describe a set of tasks which can be added to the software Basic profile for VSEs with the aim to better support software testing.

— Annex C has been added to describe a set of tasks, which can be added to the software Basic profile, to enable VSEs to add accessibility tasks to the software implementation process.

— Annex D has been added to describe a set of tasks which can be added to the Basic profile to integrate security related tasks.

A list of all parts in the ISO/IEC 29110 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

0.1 Introduction to the ISO/IEC 29110 series

For the purpose of the ISO/IEC 29110 series, a very small entity VSE is an enterprise, organization (e.g. government agency, non-profit organization), department or project having up to 25 people. Many VSEs develop and/or maintain systems and the software components used in those systems, either as independent products or incorporated into the larger system. Due to this, a recognition of VSEs as suppliers of high-quality products is required.

VSEs around the world are creating valuable products and services. According to the World Bank, small and medium enterprises (SMEs) account for about 90 % of enterprises worldwide. According to the Organisation for Economic Co-operation and Development (OECD), SMEs represent 99 % of all businesses and generate about 60 % of employment. Almost one person out of three is employed in a micro firm with less than 10 employees. The European Union reports that micro firms, with fewer than 10 persons, account for 93,5 % of all enterprises and small firms, with 10 to 49 employees, account for 5,5 % of all enterprises. The challenge facing OECD governments is to provide a business environment that supports the competitiveness of this large heterogeneous business population and that promotes a vibrant entrepreneurial culture.

From studies and surveys conducted, it is clear that the majority of International Standards do not address the needs of VSEs. Implementation of and conformity with these standards is difficult, if not impossible.

Consequently, VSEs have no, or very limited, ways to be recognized as entities that produce quality systems/system elements including software in their domain. Therefore, VSEs are excluded from some economic activities.

It has been found that VSEs find it difficult to relate International Standards to their business needs and to justify the effort required to apply standards to their business practices. Most VSEs can neither afford the resources, in terms of number of employees, expertise, budget and time, nor do they see a net benefit in establishing over-complex systems or software life cycle processes. To address some of these difficulties, a set of guidelines has been developed based on a set of VSE characteristics. The guidelines are based on subsets of appropriate standards processes, activities, tasks, and outcomes, referred to as Profiles. The purpose of a profile is to define a subset of International Standards relevant to the VSEs' context; for example, processes, activities, tasks, and outcomes of ISO/IEC/IEEE 12207 for software; and processes, activities, tasks, and outcomes of ISO/IEC/IEEE 15288 for systems; and information products (documentation) of ISO/IEC/IEEE 15289 for software and systems.

VSEs can achieve recognition through implementing a profile and by being audited against ISO/IEC 29110 specifications.

The ISO/IEC 29110 series can be applied at any phase of system or software development within a life cycle. The ISO/IEC 29110 series is intended to be used by VSEs that do not have experience or expertise in adapting/tailoring ISO/IEC/IEEE 12207 or ISO/IEC/IEEE 15288 standards to the needs of a specific project. VSEs that have expertise in adapting/tailoring ISO/IEC/IEEE 12207 or ISO/IEC/IEEE 15288 are encouraged to use those standards instead of the ISO/IEC 29110 series.

The ISO/IEC 29110 series is intended to be used with any life cycle, such as waterfall, iterative, incremental, evolutionary or agile.

Systems, in the context of the ISO/IEC 29110 series, are typically composed of hardware and software components.

The ISO/IEC 29110 series, targeted by audience, has been developed to improve system or software and/or service quality, and process performance. Figure 1 describes the ISO/IEC 29110 series and positions the parts within the framework of reference.

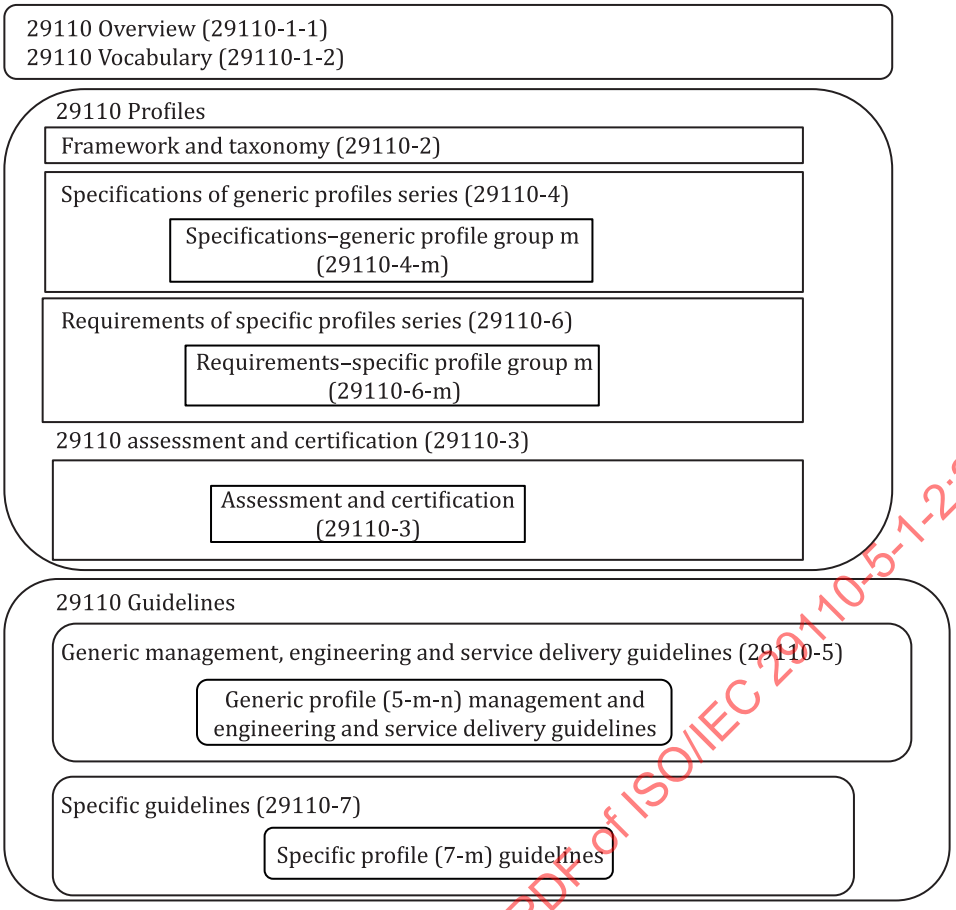

Figure 1 — The ISO/IEC 29110 series

ISO/IEC 29110-1-1 introduces processes, life cycle and standardization concepts, the taxonomy (catalogue) of ISO/IEC 29110 profiles and the ISO/IEC 29110 series. ISO/IEC 29110-1-1 also introduces the characteristics and needs of a VSE, and clarifies the rationale for specific profiles, documents, standards and guidelines. ISO/IEC 29110-1-2 defines the terms common to the ISO/IEC 29110 series. ISO/IEC 29110-1-1 and ISO/IEC 29110-1-2 are targeted at VSEs and their customers, assessors, standards producers, tool vendors and methodology vendors.

ISO/IEC 29110-2 introduces the concepts for systems and software engineering profiles for VSEs. It establishes the logic behind the definition and application of profiles. For standardized profiles, it specifies the elements common to all profiles (structure, requirements, conformity, and assessment). For domain-specific profiles (profiles that are not standardized and developed outside of the ISO process), it provides general guidance adapted from the definition of standardized profiles. ISO/IEC 29110-2 is targeted at profile producers, tool vendors and methodology vendors.

ISO/IEC 29110-3 defines certification schemes, assessment guidelines and compliance requirements for process capability assessment, conformity assessments, and self-assessments for process improvements. ISO/IEC 29110-3 also contains information that can be useful to developers of certification and assessment methods and developers of certification and assessment tools. ISO/IEC 29110-3 is addressed to people who have direct involvement with the assessment process, for example, the auditor, certification and accreditation bodies and the sponsor of the audit, who need guidance on ensuring that the requirements for performing an audit have been met. ISO/IEC 29110-3 is targeted at VSEs and their customers, assessors, accreditation bodies.

ISO/IEC 29110-4 provides the specifications for all generic profiles of the Generic profile group that are based on subsets of appropriate standards elements. ISO/IEC 29110-4 is targeted at VSEs, customers, standards producers, tool vendors and methodology vendors.

ISO/IEC TR 29110-5 provides a management, engineering and service delivery guidelines for profiles of the Generic profile group. ISO/IEC 29110-5 is targeted at VSEs and their customers.

ISO/IEC 29110-6 provides the specifications for Specific profiles that are based on subsets of appropriate standards elements. ISO/IEC 29110-6 is targeted at VSEs, customers, standards producers, tool vendors and methodology vendors.

ISO/IEC 29110-7 provides a guideline for each profile of the specific profile group. ISO/IEC 29110-7 is targeted at VSEs and their customers.

If a new profile is needed, ISO/IEC 29110-4, ISO/IEC 29110-6, ISO/IEC 29110-7 or ISO/IEC 29110-5, or all, can be developed with minimal impact to existing documents.

Since a VSE may be an enterprise, a project or a department of an organization, a customer of a VSE can be internal or external to the organization.

0.2 Introduction to this document

This document is the second software profile of a four-profile software engineering roadmap (i.e. Entry, Basic, Intermediate and Advanced).

This document is intended to be used by VSEs to support the software product which they developed.

This document is intended to be used by VSEs to add accessibility tasks to the software implementation process.

This document is intended to be used with any processes, techniques and methods that enhance the VSE's customer satisfaction and productivity.

The life cycle processes described in the ISO/IEC 29110 series are not intended to preclude or discourage their use by organizations larger than VSEs.

Using this document, a VSE can obtain the following benefits:

— an agreed set of project requirements and expected work products is delivered to the customer;

— a disciplined management process that provides project visibility and corrective actions of project problems and deviations is performed;

— a systematic software implementation process that satisfies customer needs and ensures quality work products is followed.

VSEs that develop systems that have software components are invited to use the systems engineering Basic profile guideline of the ISO/IEC 29110 series (i.e. ISO/IEC 29110-5-6-2).

In this document, Annex A describes a process, which can be added to the Basic profile, to enable a VSE to support the software product which they developed. Annex B adds a set of tasks, one role and several work products to the software Basic profile for a VSE with the aim to better support software testing. Annex C describes a set of additional tasks for a VSE that should add accessibility to a software product. Annex D integrates security-related tasks and work products in the Basic profile for software engineering. Annex E describes the deployment packages for the software Basic profile.

Conformity requirements for implementations of this document can be found in ISO/IEC 29110-4-1.

Systems and software engineering — Life cycle profiles for very small entities (VSEs) —

Part 5-1-2:
Software engineering guidelines for the generic Basic profile

1 Scope

This document provides management and engineering guidelines to the software Basic profile specified in ISO/IEC 29110-4-1 through project management and software implementation processes.

This document applies to VSEs that do not develop safety-critical software.

This document applies for software development projects, which can fulfil an external or internal agreement.

This document is applicable to VSEs developing a single product by a single work team.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

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

ISO and IEC 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/

3.1
acceptance testing
testing (3.47) conducted to determine whether a system satisfies its acceptance criteria and to enable the *customer* (3.10) to determine whether to accept the system

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.34]

3.2
accessibility
extent to which products, systems, services, environments and facilities can be used by people from a population with the widest range of characteristics and capabilities to achieve a specified goal in a specified context of use

3.3
activity
set of cohesive *tasks* (3.41) of a *process* (3.18)

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.3]

3.4
adaptive strategies
techniques that people with disabilities use to improve interaction with the web

EXAMPLE Increasing the font size in a common browser.

[SOURCE: W3C Web Accessibility Initiative – Planning and Policies – Involving Users]

3.5
agreement
mutual acknowledgement of terms and conditions under which a working relationship is conducted

EXAMPLE Contract, memorandum of agreement.

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.5]

3.6
assistive technology
equipment, product system, hardware, software or service that is used to increase, maintain or improve capabilities of individuals

Note 1 to entry: Assistive technology is an umbrella term that is broader than assistive products.

Note 2 to entry: Assistive technology can include assistive services, and professional services needed for assessment, recommendation and provision.

[SOURCE: ISO/IEC GUIDE 71:2014, 2.16]

3.7
baseline
formally approved version of a configuration item, regardless of media, formally designated and fixed at a specific time during the configuration item's life cycle

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.11]

3.8
Basic profile
profile (3.21) targeted at *VSEs* (3.52) developing a single product by a single work team

[SOURCE: ISO/IEC 29110-1-2:2024, 4.14]

3.9
conditional task
task (3.37) that can be mandatory under some specified condition(s), can be optional under other specified conditions, and can be out of scope or not applicable under other specified conditions

Note 1 to entry: These are to be observed if the specified condition(s) apply.

3.10
customer
person or organization that could or does receive a product or a service that is intended for or required by this person or organization

EXAMPLE Consumer, client, end-user, retailer, receiver of product or service from an internal *process* (3.18), beneficiary and purchaser.

Note 1 to entry: A customer can be internal or external to the organization.

[SOURCE: ISO 9000:2015, 3.2.4]

3.11
defect
imperfection or deficiency in a *work product* (3.53) where that work product does not meet its *requirements* (3.26) or specifications and needs to be either repaired or replaced

[SOURCE: IEEE 1044:2009]

3.12
deployment package
DP
set of artefacts developed to facilitate the implementation of a set of practices, of the selected framework, in a *very small entity* (3.52)

[SOURCE: ISO/IEC 29110-1-2:2024, 3.35]

3.13
expected result
observable predicted behaviour of the *test item* (3.43) under specified conditions based on its specification or another source

[SOURCE: ISO/IEC/IEEE 29119-1:2022, 3.35]

3.14
generic profile group
profile (3.21) group applicable to *VSEs (very small entities)* (3.52) that do not develop safety-critical systems or *software products* (3.32) and have typical situational factors

[SOURCE: ISO/IEC 29110-1-2:2024, 4.28]

3.15
incident
unplanned interruption to a service, a reduction in the quality of a service or an event that has not yet impacted the service to the *customer* (3.10) or user

[SOURCE: ISO/IEC 20000-10:2018, 3.2.5]

3.16
integration testing
testing (3.47) in which software components, hardware components, or both are combined and tested to evaluate the interaction among them

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.2034, modified — Note 1 to entry has been removed.]

3.17
persona
representation of a type of user that includes a concise summary of the characteristics and the behaviour of the user that is most informative to the design or illustrative of specific user *requirements* (3.26)

[SOURCE: ISO/IEC 25063:2014, modified — Note 1 to entry has been removed.]

3.18
process
set of interrelated or interacting *activities* (3.3) that transforms inputs into outputs

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.33]

3.19
process purpose
high-level objective of performing the *process* (3.18) and the likely outcomes of effective implementation of the process

[SOURCE: ISO/IEC/IEEE 24774:2021, 3.12]

3.20
process outcome
observable result of the successful achievement of the *process* (3.18)

[SOURCE: ISO/IEC/IEEE 24774:2021, 3.11]

3.21
profile
subset of appropriate standards' *processes* (3.18) and their outcomes, *activities* (3.3) and *tasks* (3.37) combined to accomplish a particular function

Note 1 to entry: The base standards used to develop profiles for *VSEs* (3.52) are the ISO/IEC/IEEE 12207, the ISO/IEC/IEEE 15288 and the ISO/IEC/IEEE 15289

[SOURCE: ISO/IEC 29110-1-2:2024, 3.70]

3.22
project
endeavour with defined start and finish dates undertaken to create a product or service in accordance with specified resources and *requirements* (3.26)

Note 1 to entry: A project is sometimes viewed as a unique *process* (3.18) comprising coordinated and controlled *activities* (3.3) and composed of activities from the technical management processes and technical processes defined in this document.

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.37]

3.23
report
information item that describes the results of *activities* (3.3) such as investigations, observations, assessments, or tests

[SOURCE: ISO/IEC/IEEE 15289:2019, 3.1.22]

3.24
review
process (3.18) or meeting during which a *work product* (3.53), or set of work products, is presented to *project* (3.22) personnel, managers, users, *customers* (3.10), or other interested parties for comment or approval

[SOURCE: ISO/IEC 29110-1-2:2024, 4.51]

3.25
risk
effect of uncertainty on objectives

Note 1 to entry: An effect is a deviation from the expected — positive and/or negative.

Note 2 to entry: Objectives can have different aspects (such as financial, health and safety, and environmental goals) and can apply at different levels (such as strategic, organization-wide, *project* (3.22), product and *process* (3.18)).

Note 3 to entry: Risk is often characterized by reference to potential events and consequences, or a combination of these.

Note 4 to entry: Risk is often expressed in terms of a combination of the consequences of an event (including changes in circumstances) and the associated likelihood of occurrence.

Note 5 to entry: Uncertainty is the state, even partial, of deficiency of information related to, understanding or knowledge of, an event, its consequence, or likelihood.

[SOURCE: ISO 31073:2022, 3.1.1]

3.26
requirement
statement that translates or expresses a need and its associated constraints and conditions

Note 1 to entry: A constraint is externally imposed limitation on the software, its design, or implementation or on the *process* (3.18) used to develop or modify a software.

Note 2 to entry: A condition is a measurable qualitative or quantitative attribute that is stipulated for a requirement and that indicates a circumstance or event under which a requirement applies.

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.44, modified — Notes to entry have been added.]

3.27
safety-critical software
software whose failure or malfunction can result in death or serious injury to people, loss of or severe damage to equipment or property, or damage to the natural environment

3.28
security
protection against intentional subversion or forced failure, containing a composite of four attributes: confidentiality, integrity, availability, and accountability, plus aspects of a fifth, usability, all of which have the related issue of their assurance

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.49]

3.29
security control
safeguard or countermeasure prescribed for an information system or an organization, designed to protect the confidentiality, integrity, and availability of its information and to meet a set of defined *security* (3.28) *requirements* (3.26)

[SOURCE: IEEE 7002:2022, 3, IEEE dictionary]

3.30
service level agreement
SLA
documented *agreement* (3.5) between the organization and the *customer* (3.10) that identifies services and their agreed performance

Note 1 to entry: A service level agreement can also be established between the organization and an external supplier, an internal supplier or a customer acting as supplier.

Note 2 to entry: A service level agreement can be included in a contract or another type of documented agreement.

[SOURCE: ISO/IEC 20000-10:2018, 3.2.20]

3.31
small and medium enterprise
SME
enterprise with less than 250 persons employed

[SOURCE: ISO/IEC 29110-1-2:2024, 3.92]

3.32
software product
set of computer programs, procedures, and possibly associated documentation and data

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.54, modified — Note 1 to entry has been removed.]

3.33
software support
continued provision of services and material necessary for the use and improvement of an implemented system

3.34
stakeholder
individual or organization having a right, share, claim, or interest in a system or in its possession of characteristics that meet their needs and expectations

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.59, modified — EXAMPLE and note 1 to entry have been removed.]

3.35
system testing
testing (3.47) conducted on a complete, integrated system to evaluate the system's compliance with its specified *requirements* (3.26)

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.4122]

3.36
support manager
role responsible for production support in the *VSE* (3.52) who oversees *customer* (3.10) relationships and change management within production environments

3.37
task
required, recommended, or permissible action, intended to contribute to the achievement of one or more outcomes of a *process* (3.18)

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.66]

3.38
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* (3.47)

[SOURCE: ISO/IEC/IEEE 29119-1:2022, 3.88, modified — Note 1 to entry has been removed.]

3.39
test data
data created or selected to satisfy the input *requirements* (3.26) for executing one or more test cases

Note 1 to entry: Test data can be stored within the *test item* (3.43) (e.g. in arrays or flat files), or can come from external sources, such as other systems, hardware devices, or human operators.

[SOURCE: ISO/IEC/IEEE 29119-1:2022, 3.91]

3.40
test design technique
test technique
procedure used to create or select a test model, identify test coverage items, and derive corresponding test cases

EXAMPLE Equivalence partitioning, boundary value analysis, decision table *testing* (3.47), branch testing.

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

Note 2 to entry: Some test practices, such as exploratory testing or model-based testing are sometimes referred to as “test techniques”. Following the definition in the ISO/IEC/IEEE 29119 series, 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.

[SOURCE: ISO/IEC/IEEE 29119-1:2022, 3.94]

3.41
test environment
environment containing facilities, hardware, software, firmware, procedures, needed to conduct a test

Note 1 to entry: A test environment can contain multiple environments to accommodate specific test level or test 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.

[SOURCE: ISO/IEC/IEEE 29119-1:2022, 3.95]

3.42
test execution
process (3.18) of running a test on the *test item* (3.43), producing actual result(s)

[SOURCE: ISO/IEC/IEEE 29119-1:2022, 3.99]

3.43
test item
test object
work product (3.53) to be tested

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

[SOURCE: ISO/IEC/IEEE 29119-1:2022, 3.107]

3.44
test plan
detailed description of test objectives to be achieved and the means and schedule for achieving them, organized to coordinate *testing* (3.47) *activities* (3.3) for some *test item* (3.43) or set of test items

[SOURCE: ISO/IEC/IEEE 29119-1:2022, 3.117, modified — Notes to entry have been removed.]

3.45
test sub-process
test management and dynamic (and static) test *process* (3.18) used to perform a specific test level [e.g. *system testing* (3.35), *acceptance testing* (3.1)] or test type (e.g. usability testing, performance testing) normally within the context of an overall test process for a test *project* (3.22)

3.46
test result
indication of whether or not a specific test case has passed or failed, i.e. if the actual result corresponds to the *expected result* (3.13) or if deviations were observed

[SOURCE: ISO/IEC/IEEE 29119-1:2022, 3.122]

3.47
testing
set of *activities* (3.3) conducted to facilitate discovery and evaluation of properties of one or more *test items* (3.43)

[SOURCE: ISO/IEC/IEEE 29119-1:2022, 3.131, modified — "one or more" has been added; note 1 to entry has been removed.]

3.48
traceability
degree to which a relationship can be established among two or more logical entities, especially entities having a predecessor-successor or master-subordinate relationship to one another, such as *requirements* (3.26), system elements, *verifications* (3.50), or *tasks* (3.37)

EXAMPLE Software features and test cases are typically traced to software requirements.

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.69]

3.49
validation
confirmation, through the provision of objective evidence, that the *requirements* (3.26) for a specific intended use or application have been fulfilled

Note 1 to entry: A system is able to accomplish its intended use, goals and objectives (i.e. meet *stakeholder* (3.34) requirements) in the intended operational environment. The right system was built.

Note 2 to entry: In a life cycle context, validation involves the set of *activities* (3.3) for gaining confidence that a system is able to accomplish its intended use, goals and objectives in an environment like the operational environment.

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.71]

3.50
verification
confirmation, through the provision of objective evidence, that specified *requirements* (3.26) have been fulfilled

Note 1 to entry: Verification is a set of *activities* (3.3) that compares a system or system element against the required characteristics. This includes, but is not limited to specified requirements, design, descriptions, and the system itself. The system was built right.

[SOURCE: ISO 9000:2015, 3.8.12, modified — Notes 1 to 3 to entry have been replaced by a new note 1 to entry.]

3.51
version control
establishment and maintenance of *baselines* (3.7) and the identification and control of changes to baselines that make it possible to return to the previous baseline

[SOURCE: ISO/IEC/IEEE 26511:2018, 3.1.40]

3.52
very small entity
VSE
enterprise, organization (e.g. government agency, non-profit organization), department or *project* (3.22) having up to 25 people

[SOURCE: ISO/IEC 29110-1-2:2024, 4.71, modified — "(e.g. government agency, non-profit organization)" has been added.]

3.53
work product
WP
artefact produced by a *process* (3.18)

EXAMPLE *Project* (3.22) plan, *requirements* (3.26) specification, design documentation, source code, *test plan* (3.44), test meeting minutes, schedules, budgets, and *incident* (3.15) *reports* (3.23).

Note 1 to entry: A subset of the work products can be baselined to be used as the basis of further work and some form the set of project deliverables.

[SOURCE: ISO/IEC 20246:2017, 3.20]

4 Naming, diagramming and definitions conventions

4.1 General

Conventions for naming, diagramming, describing and defining profiles are defined in ISO/IEC 29110-2-1:2015.

4.2 Naming, diagramming and definition conventions

The following process structure description and notation are used to describe the processes.

Process name – process identifier, followed by its abbreviation in parenthesis “( )”.

Process purpose – high level objective of performing the process and the likely outcomes of effective implementation of the process.

Process outcomes – observable result of the successful achievement of the process purpose. Outcomes are identified by the abbreviation of the process name, followed by the letter "O" and a consecutive number, for example PM.O1, SI.O2.

Input work products – Work products (WP) which can be used to perform the process and its corresponding source, which can be another process or an external entity to the project, such as the customer. They are identified by the abbreviation of the process name.

Output work products – Work products generated by the process and its corresponding destination, which can be another process or an external entity to the project, such as customer. They are identified by the abbreviation of the process name.

Internal work products – Work products generated (i.e. output work product) and consumed (i.e. input work product) by the process. An internal work product is not reviewed or approved by the customer.

All work products' names initiate with capital letters. Some work products should have one or more statuses attached to the work product name surrounded by square brackets "[ ]" and separated by ",". The work product state may change during the process execution. See Clause 9 for the alphabetical list of the work products, its descriptions, possible states and the source of the work product. The source can be another process or an external entity to the project, such as the customer.

Roles involved – Names and abbreviation of the functions to be performed by project team members. Several roles may be played by a single person and one role may be assumed by several persons. Roles are assigned to project participants based on the characteristics of the project. The role list is identified by the abbreviation of the process name and showed as two column table. See Clause 8 for the alphabetical list of the roles and recommended competencies description.

Diagram – Graphical representation of the processes. The large round-edged rectangles indicate process or activities, and the smaller square-edged rectangles indicate the work products. The directional or bidirectional thick arrows indicate the major flow of information between processes or activities. The thin directional or bidirectional arrows indicate the input or output work products. The notation used in the diagrams does not imply the use of any specific process life cycle.

Activity – A set of cohesive tasks of a process. The task statements in this document are not imperative. A process activity is the first level of process workflow decomposition, and the second one is a task. Activities are identified by process name abbreviation followed by consecutive number and the activity name.

Activity description – Each activity description is identified by the activity name and the list of related process outcomes surrounded by parenthesis "( )". For example, PM.01 project planning (PM.O1, PM.O5, PM.O6, PM.O7) means that the activity PM.01 project planning contributes to the achievement of the listed objectives: PM.O1, PM.O5, PM.O6 and PM.O7.

Task description – Each task description begins with an active verb (e.g. assign, test) and is followed by an object (e.g. review the project plan). To facilitate their implementation, a few tasks are broken down in elementary tasks. The task description doesn't impose any technique or method to perform it. The selection of the techniques or methods is left to the VSE or project team.

Task description tables contain four columns corresponding to:

— Role - the abbreviation of role(s) involved in the task execution.

— Task - description of the task to be performed. Each task is identified by activity ID and consecutive number, for example PM.01.01, PM.01.02. A few numbered items are added to provide additional information intended to assist the understanding or use of tasks.

— Input work products – work products needed to execute a task.

— Output work products – work products created or modified by the execution of a task.

NOTE 1 A conditional task is executed if its associated work product (e.g. software user documentation) is required by the customer and listed in the delivery instructions. The conditional task statement is preceded with this text: " x) Conditional task:".

Incorporation to project repository – list of work products to be saved in project repository; the version control strategy should be applied to some of them. A few work products (e.g. requirements) are baselined, a change to a baselined work product can be done through an approved change request.

NOTE 2 Tables used in process description are for presentation purposes only.

NOTE 3 The term 'Basic' is using a capital 'B' to indicate an ISO/IEC 29110 profile (e.g. the Basic profile) while the term 'basic' is used when referring to the most important part of something (e.g. basic principles).

NOTE 4 A work product is baselined when it has been approved and uploaded in the repository.

4.3 Abbreviated terms

AN	analyst
ARIA	accessible rich Internet applications
CMMI	Capability Maturity Model Integration
CSS	cascading style sheets
CUS	customer
DES	designer
DP	deployment package
HTML	hypertext mark-up language
IM	incident management
PCM	production change management
PM	project management
PJM	project manager
PR	programmer
SI	software implementation
SLM	service level management
SM	support manager
SMIL	synchronized multimedia integration language
SS	software support
ST	support team
TL	technical leader
W3C	World Wide Web Consortium
WCAG	Web Content Accessibility Guidelines
WT	work team

5 Overview

5.1 General

This document provides project management (PM) and software implementation (SI) processes which integrate tasks based on the selection of elements from ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15289.

This document is intended to be used by the VSE to establish processes to implement any development approach or methodology including, for example, agile, evolutionary, incremental, test-driven development, based on the VSE or project needs.

5.2 Entry conditions to use the software Basic profile

To use this document, a VSE should fulfil the following conditions:

— project agreement was documented, approved by the management of the VSE and baselined before the start of the project;

— a project feasibility study was performed before the project start;

— the project team, including project manager, is assigned and trained;

— development cycle(s) should be documented (e.g. waterfall, iterative, incremental, evolutionary, agile);

— goods, services and infrastructure to start the project are available.

5.3 Processes and activities of the software Basic profile

This document provides two processes: a project management (PM) process and a software implementation (SI) process.

Both processes of the software Basic profile are interrelated (see Figure 2).

The processes illustrated in Figure 2 are intended to be used with any life cycle such as: waterfall, iterative, incremental, evolutionary or agile.

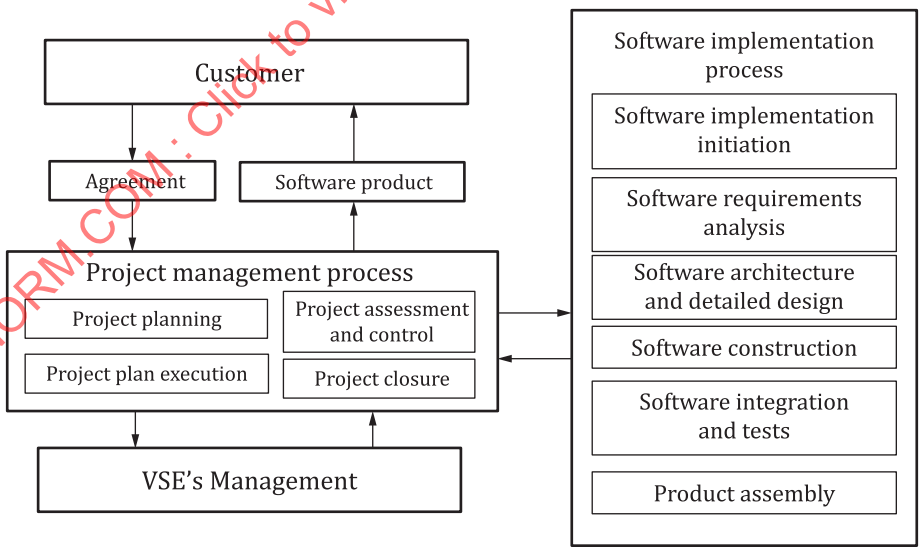

Figure 2 — Processes and activities of the software Basic profile

A customer can be external or internal to an organization (e.g. a client, a project or department of an organization that develops products composed of hardware and software components), therefore the agreement may come from an external or internal customer.

The PM process uses the customer's agreement to plan the project. As the project progresses, the PM project assessment and control tasks compare the project progress against the project plan and actions are taken to eliminate deviations or incorporate changes to the project plan. The customer approves changes to the project plan, agreement and change requests that impact the project plan. The PM project closure activity delivers the software product, produced by the SI process, and gets the customer's acceptance to formalise the end of the project. A project repository is established to save the work products and to control its versions during the project.

The execution of the SI process is driven by the project plan. The SI process starts with an initiation activity of the project plan revision. The project plan guides the execution of the software requirements analysis, software architectural and detailed design, software construction, software integration and test, and software product assembly activities.

To remove work product's defects verification, validation and test tasks are included in the activity's workflow.

The customer provides an agreement as an input to project management process and receives a software product as a result of software implementation process execution (see Figure 2).

The Basic profile uses baselines and traceability for the control of work products:

— the requirements baseline is traceable to the agreement approved by the management of the VSE;

— the specification baseline is traceable to the requirements specification work product;

— the design is traceable to the software architecture and detailed design work products;

— the product baseline is traceable to all the work products listed in the delivery instructions signed by the customer and the project manager.

6 Project management (PM) process

6.1 PM process purpose

The purpose of the project management process is to achieve the project's time, quality and cost objectives by establishing and executing the tasks of the software implementation project systematically. It also carries out administrative tasks relating to the storage, handling, protection and delivery of work products and configuration items.

This document is intended to be used by the VSE to establish processes to implement any development approach or methodology including, for example, agile, evolutionary, incremental, test-driven development, based on the VSE or project needs.

6.2 PM process outcomes

PM.O1. The project plan for the execution of the project is developed according to the agreement and reviewed and accepted by the customer. The tasks and resources necessary to complete the work are sized and estimated.

PM.O2. Progress of the project is monitored against the project plan and recorded in the progress status record. Changes to the project plan are approved by the customer. Corrections to remediate problems and deviations from the plan are taken when project targets are not achieved. Closure of the project is performed to get the customer acceptance documented in the acceptance record.

PM.O3. The change requests are addressed through their reception and analysis. Changes to software requirements are evaluated for cost, schedule, and technical impact. Change requests that impact the project plan, the requirements or the agreement are approved by the customer.

PM.O4. Review meetings with the work team and the customer are held. Agreements are registered and tracked.

PM.O5. Risks are identified as they develop and during the conduct of the project.

PM.O6. A software version control strategy is developed. Items of software product are identified, defined, and baselined. Modifications and releases of the items are controlled and made available to the customer and work team. The storage, handling and delivery of the items are controlled.

PM.O7. Software quality assurance is performed to provide assurance that work products and processes comply with the project plan and requirements specification.

NOTE Software quality assurance is implemented through the performance of the verification, validation and review tasks performed in project management and software implementation processes.

6.3 PM roles involved

The roles involved in the PM process are:

— customer;

— project manager;

— technical leader;

— work team.

6.4 PM activities and tasks

6.4.1 Overview of the PM process

Figure 3 shows the flow of information between the project management process activities including the most relevant work products and their relationship.

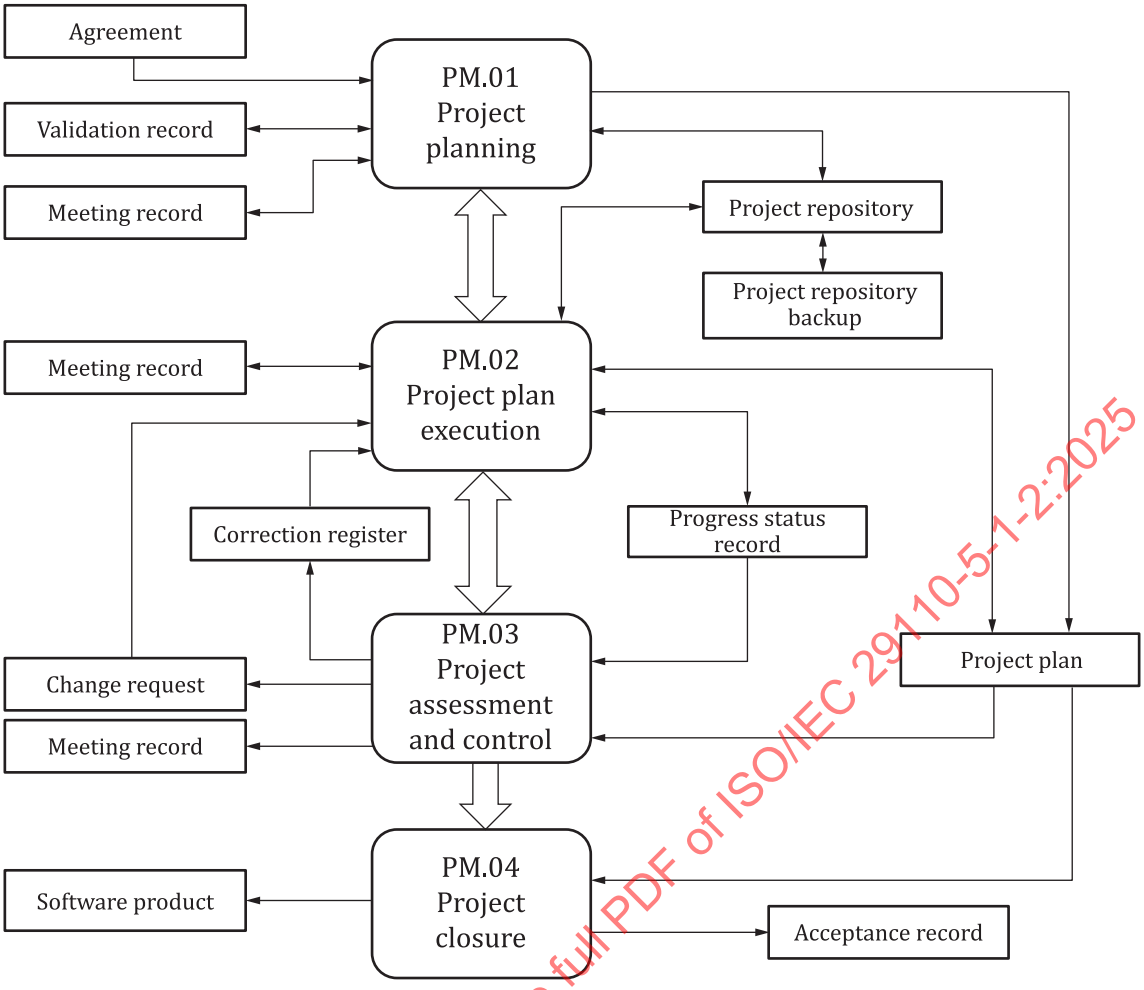

Figure 3 — Project management process

NOTE 1 The process illustrated in Figure 3 is intended to be used with any life cycle such as waterfall, iterative, incremental, evolutionary or agile.

NOTE 2 The agreement is provided by an internal or external customer.

6.4.2 PM activities

6.4.2.1 Overview

The project management process should have the following activities:

— PM.01 Project planning

— PM.02 Project plan execution

— PM.03 Project assessment and control

— PM.04 Project closure

6.4.2.2 PM.01 Project planning (contributes to PM.O1, PM.O5, PM.O6, PM.O7)

This activity documents the planning details needed to manage the project. The activity provides:

— reviewed agreement and the tasks needed to provide the work products of the agreement and to satisfy other customer requirements;

— project life cycle, including task dependencies and duration;

— project quality assurance strategy through verification and validation of work products/deliverables, customer and work team reviews;

— work team and customer roles and responsibilities;

— project resources and training needs;

— estimates of effort, cost and schedule;

— identified project risks;

— project version control strategy;

— project repository to store, handle and deliver controlled work product and work product versions and baselined work products.

The task list for PM.01 project planning is given in Table 1.

Table 1 — PM.01 task list

Roles	Task list	Input work products	Output work products
PJM TL	PM.01.01 Review the agreement for the development cycle of the project (e.g. waterfall, iterative, incremental, evolutionary, agile). 1) The agreement is an entry condition to the use of this guideline as listed in Clause 6. 2) The agreement had been approved and baselined by the management of the VSE before the start of the project. 3) ISO/IEC 20246 defines different types of work product reviews.	Agreement [approved]	Project plan [initiated] — Development cycle
PJM CUS	PM.01.02 Define with the customer the delivery instructions. 1) The delivery instructions list all the work products (e.g. code; documentation) to be delivered to the customer as specified in the agreement.	Agreement [approved] Project plan [initiated]	Project plan [initiated] — Delivery instructions
PJM TL	PM.01.03 Identify the tasks to produce the work products to be delivered to the customer. 1) Tasks of the PM and SI processes, with verification, validation and reviews with customer and work team, are added to assure the quality of work products. 2) Tasks are identified according to the development cycle selected.	Agreement [approved] Project plan [initiated] — Delivery instructions	Project plan [initiated] — Tasks
PJM TL	PM.01.04 Establish the estimated duration to perform each task.	Project plan [initiated] — Tasks	Project plan [initiated] — Estimated duration

Table 1 *(continued)*

Roles	Task list	Input work products	Output work products
PJM TL	PM.01.05 Identify and document the resources to perform the project in the project plan. 1) Resources include human, material, equipment and tools, standards. 2) Schedule and dates when resources will be needed are added to the project plan. 3) A work product is baselined when it should be approved and uploaded in the project repository.	Agreement [approved] Project plan [initiated]	Project plan [initiated] — Resources
PJM TL	PM.01.06 Establish the composition of work team. 1) Roles and responsibilities are assigned according to the resources available.	Project plan [initiated] — Resources	Project plan [initiated] — Composition of work team
PJM TL	PM.01.07 Create the schedule of the project tasks. 1) Estimated start and completion dates to each task are assigned. 2) The assigned resources, sequence and dependency of the tasks are considered.	Project plan [initiated] — Tasks — Estimated duration — Composition of work team	Project plan [initiated] — Schedule of the project tasks
PJM	PM.01.08 Calculate and document the project estimated effort and cost.	Project plan [initiated] — Schedule of the project tasks — Resources	Project plan [initiated] — Estimated effort and cost
PJM TL	PM.01.09 Identify and document the risks which can affect the project. 1) Risk is a combination of the effect, or impact, of uncertainty on project objectives. 2) An effect is a positive and/or negative deviation from the expected. 3) The management of deviations from the expected is performed to achieve the project and/or product objectives (e.g. financial, security).	Project plan [initiated]	Project plan [initiated] — Identification of project risks
PJM TL	PM.01.10 Document the version control strategy in the project plan. 1) Baselines selected for the project are described in the version control strategy.	Project plan [initiated]	Project plan [initiated] — Version control strategy

Table 1 *(continued)*

Roles	Task list	Input work products	Output work products
PJM	PM.01.11 Generate the project plan. 1) All the elements previously documented are integrated to the project plan.	Project plan [initiated]	Project plan [initiated] — Tasks — Estimated duration — Resources — Composition of work team — Schedule of the project task — Estimated effort and cost — Identification of project risks — Version control strategy — Delivery instructions
PJM TL	PM.01.12 Add product description, scope, objectives and deliverables listed in the agreement to the project plan.	Agreement [approved] — Product description — Scope — Objectives — Deliverables Project plan [initiated]	Project plan [initiated] — Product description — Scope — Objectives — Deliverables
PJM TL	PM.01.13 Validate the project plan: — Make corrections until the document is approved by PJM. 1) All elements of the project plan are feasible and consistent. 2) The results of the validation are documented in a verification record.	Project plan [initiated]	Project plan [validated] Validation record [approved]
PJM CUS	PM.01.14 Obtain approval of the customer of the project plan. 1) The customer reviews and signs the project plan, making sure that the elements of the project plan match with the agreement.	Project plan [validated]	Project plan [approved] Meeting record [published]
PJM TL	PM.01.15 Establish the project repository using the version control strategy.	Project plan [approved] — Version control strategy	Project repository [established]

Table 1 *(continued)*

Roles	Task list	Input work products	Output work products
PJM	PM.01.16 Add the agreement and the project plan to the project repository. 1) The approved agreement and the project plan are stored in the project repository as the baselined agreement and project plan. 2) Changes to the baselined agreement and to the project plan are done by approving change requests.	Project plan [approved] Agreement [approved]	Project repository [updated]

6.4.2.3 PM.02 Project plan execution (contributes to PM.O2, PM.O3, PM.O4, PM.O5, PM.O7)

This activity implements the documented plan on the project.

The activity provides:

— progress status record of the project updated;

— analysed and evaluated change requests to the plan impacting cost, schedule and technical requirements;

— approved changes to the plan;

— reviews and agreements with the work team (WT) and customer (CUS);

— back up of the project repository, and its recovery if necessary.

The task list for PM.02 project plan execution is given in Table 2.

Table 2 — PM.02 task list

Roles	Task list	Input work products	Output work products
PJM TL WT	PM.02.01 Monitor the project plan execution. 1) Actual data are recorded in the progress status record. 2) Actual project record data to monitor includes: — Tasks — Results — Resources and effort — Cost — Elapsed time — Defects	Project plan [approved]	Progress status record [published]
PJM TL	PM.02.02 Analyse and evaluate change request for cost, schedule and technical impact. 1) Conditional task: Analyse and evaluate change request for cost, schedule and technical impact. 2) A change request may be initiated by the customer or internally by the PJM, TL or the WT. 3) A change request, which affects the agreement, should be negotiated by customer and PJM (see task PM.02.04).	Change request [initiated] Project plan [approved] Traceability record [initiated]	Change request [evaluated]
PJM CUS	PM.02.03 Obtain approval of change request: — If a change request is approved by customer or PJM, update the project plan if necessary. 1) Conditional task: Obtain approval of change request	Change request [evaluated]	Change request [approved or rejected]
PJM TL WT	PM.02.04 Conduct review meetings with the work team: — Track problems to closure. 1) Conditional task: Track problems to closure. 2) Meetings are conducted to identify problems, review risk status, record agreements. 3) If a problem should be detected, a correction register is initiated.	Project plan [approved] Progress status record [published]	Meeting record [published] Correction register [initiated]

Table 2 *(continued)*

Roles	Task list	Input work products	Output work products
PJM CUS TL WT	PM.02.05 Conduct review meetings with the customer: — Initiate a change request if needed. — Record agreements between customer and PJM and tracked them to closure. — If a change request is initiated, update if necessary, the project plan according to new agreement between customer and PJM. 1) A change request which affects the customer, should be negotiated to reach acceptance of customer and PJM.	Project plan [approved] Progress status record [published] Meeting record [published]	Meeting record [published] Change request [approved] Project plan [updated]
TL	PM.02.06 Add project management artefacts to the project repository.	Project plan [approved] — Version control strategy	Project repository backup [established]
PJM TL	PM.02.07 Verify the project repository recovery using the project repository backup. 1) The verification is performed to check that the work products that should be backed up can be recovered.	Project repository backup [established]	Project repository [recovered]

6.4.2.4 PM.03 Project assessment and control (contributes to PM.02)

This activity evaluates the performance of the plan against documented commitments.

The activity provides:

— evaluation of actual plan performance and progress against targets;

— approval of customer of changes to the requirements or to the project plan;

— identified and evaluated significant cost, schedule and technical performance deviations and problems;

— review of project risks and identification of new risks;

— documented change requests, appropriate corrective action defined, and changes tracked to closure.

The task list for PM.03 Project assessment and control is given in Table 3.

Table 3 — PM.03 task list

Roles	Task list	Input work products	Output work products
PJM TL WT	PM.03.01 Evaluate project progress with respect to the project plan, comparing: — Actual tasks against planned tasks — Actual results against established project objectives — Actual resource allocation against planned resources — Actual cost against budget estimates — Actual time against planned schedule — Actual risk against previously identified	Project plan [approved] Progress status record [published]	Progress status record [evaluated]
PJM TL WT	PM.03.02 Establish actions to correct deviations or problems and identified risks concerning the accomplishment of the project plan, as needed: — Document actions in the correction register. — Track actions to closure. 1) Conditional task: Track actions to closure.	Progress status record [evaluated]	Correction register [initiated]
PJM TL WT	PM.03.03 Identify changes to requirements or project plan: — Document actions in change request. — Track actions to closure. 1) Conditional task: Document actions in change request and track actions to closure. 2) To address major deviations, potential risks or problems concerning the accomplishment of the plan.	Progress status record [evaluated]	Change request [initiated]
CUS PJM	PM.03.04 Obtain approval from the customer about the changes to the requirements or to the project plan. 1) Conditional task: This task is not executed if there is no change to the requirements or to the project plan.	Progress status record [evaluated] Change request [initiated] Correction register [initiated] Agreement [approved]	Change request [approved] Meeting record [approved]

6.4.2.5 PM.04 Project closure (contributes to PM.02)

This activity provides the project's work products in accordance with the requirements of the agreement.

The activity provides:

— delivery of the work product as specified in the delivery instructions;

— support of customer work products acceptance in accordance with the delivery instructions;

— completion of the project and signature of the acceptance record;

— updated project repository.

The task list for PM.04 Project closure is given in Table 4.

Table 4 — PM.04 task list

Roles	Task list	Input work products	Output work products
PJM CUS	PM.04.01. Obtain signature of customer of the acceptance record. 1) The work products, listed in the delivery instructions, should be assembled as the software product in activity SI.06. 2) The delivery instructions, signed by the customer and PJM, is used to conduct the acceptance of the work products by the customer. 3) The completion of the project that requires "signature of the acceptance record" cannot be applicable to mass-market customers. 4) The customer may be internal or external to the organization.	Project plan [approved] — Delivery instructions Software product [baselined]	Acceptance record [approved] Software product [delivered]
PJM	PM.04.02 Store the acceptance record in the project repository.	Acceptance record [approved]	Project repository [updated]

6.4.3 Incorporation to project repository

After the incorporation of the work products to project repository, a version control strategy should be applied to the project plan. The list of work products to be saved in the project repository are:

— project plan;

— change request;

— acceptance record;

— meeting record;

— correction register;

— progress status record;

— validation record.

7 Software implementation (SI) process

7.1 SI process purpose

The purpose of the software implementation process is the systematic performance of the analysis, design, construction, integration, testing activities and assembly of work products for a new or modified software product according to the specified requirements.

This document is intended to be used by the VSE to establish processes to implement any development approach or methodology including, for example, agile, evolutionary, incremental, test-driven development, based on the VSE or project needs.

7.2 SI process outcomes

SI.O1. Tasks of the activities are performed through the accomplishment of the current project plan.

SI.O2. Software requirements are defined, analysed for correctness and testability, approved by the customer, baselined and communicated. Traceability between customer requirements and software requirements is established.

SI.O3. Software architectural and detailed design is developed. Software architecture is baselined. It describes the software components and internal and external interfaces of them. Consistency and traceability to software requirements are established.

NOTE 1 Software architecture and detailed design can be performed separately according to the project schedule.

SI.O4. Software components defined by the design are produced. Unit tests are defined and performed to verify the consistency with requirements and the design. Traceability to the requirements and design are established.

SI.O5. Software is produced performing integration of software components and verified using test cases and test procedures. Results are recorded in the test report. Defects are corrected and consistency and traceability to software design are established.

SI.O6. A software product, that meets the requirements specification as agreed to with the customer, which includes user, operation and maintenance documentations is integrated, baselined and stored at the project repository. Needs for changes to the software product are detected and related change requests are initiated.

SI.O7. Verification and validation tasks of all required work products are performed using the defined criteria to achieve consistency among output work products and input work products in each activity. Defects are identified and corrected; records are stored in the verification/validation records.

NOTE 2 It's not the intention that all verification activities and work products are made available to the customer. Verifications are performed by individuals that have organizational freedom, authority, to permit objective evaluation, and to initiate, effect, resolve and verify problem resolution.

7.3 SI Roles involved

The roles involved in the SI process are:

— customer;

— analyst;

— designer;

— programmer;

— project manager;

— technical leader;

— work team.

7.4 SI activities and tasks

7.4.1 Overview of the SI process

Figure 4 shows the flow of information between the software implementation process activities including the most relevant work products and their relationships.

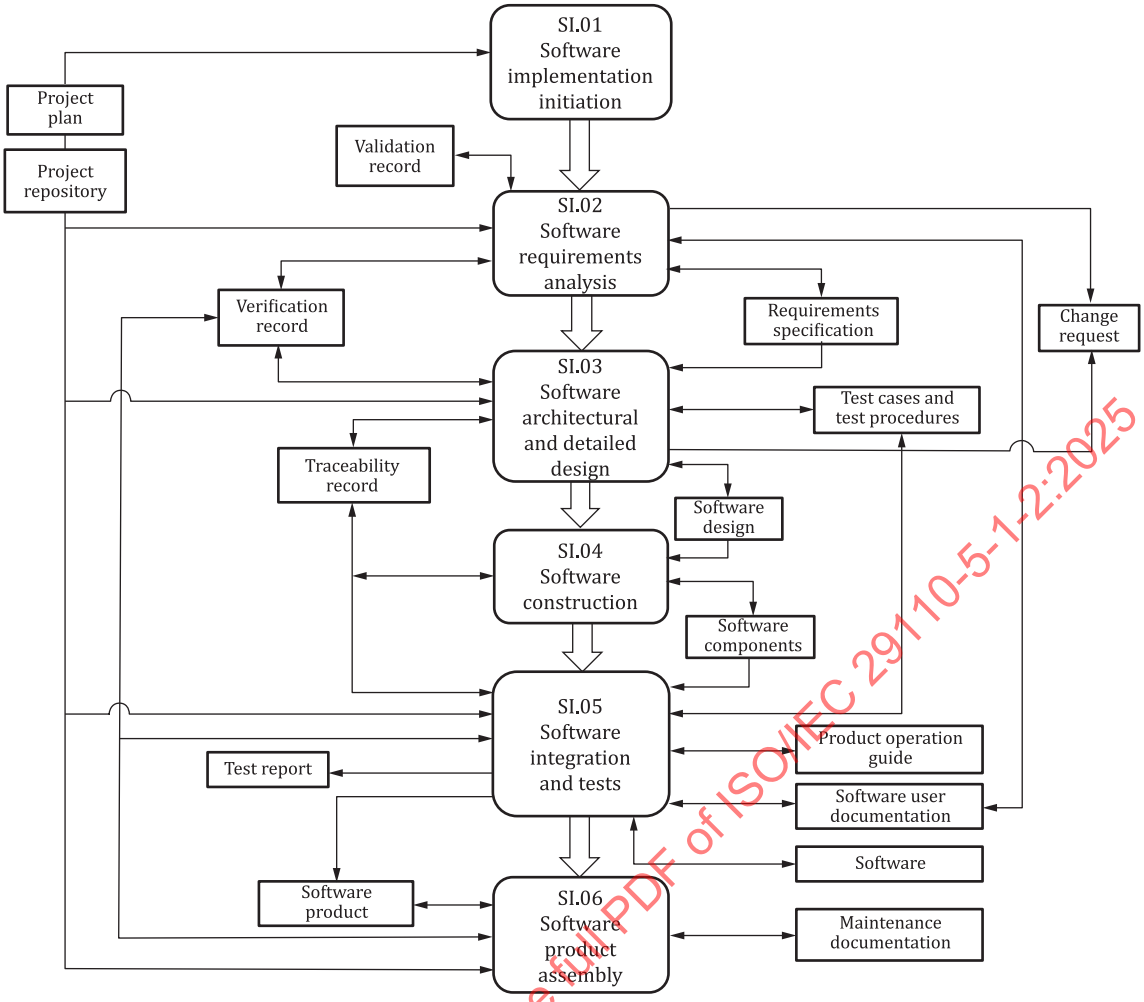

Figure 4 — Software implementation process

NOTE The process illustrated in Figure 4 is intended to be used with any life cycle such as waterfall, iterative, incremental, evolutionary or agile.

7.4.2 SI activities

7.4.2.1 Overview

The software implementation process should have the following activities:

— SI.01 Software implementation initiation

— SI.02 Software requirements analysis

— SI.03 Software architectural and detailed design

— SI.04 Software construction

— SI.05 Software integration and tests

— SI.06 Software product assembly

7.4.2.2 SI.01 Software implementation initiation (contributes to SI.O1)

This activity ensures that the project plan, established in the project planning activity, is committed to by the work team.

The activity provides:

— review of task assignments of the work team established in the project plan;

— commitment to project plan by the work team, team leader and project manager;

— an implementation environment is established.

The task list for SI.01 software implementation initiation is given in Table 5.

Table 5 — SI.01 task list

Roles	Task list	Input work products	Output work products
PJM TL WT	SI.01.01 Review the current project plan with the work team. 1) A review is performed to achieve a common understanding and get the commitment of the WT, TL and PJM to the project plan. 2) ISO/IEC 20246 defines different types of work product reviews.	Project plan [approved]	Project plan [approved]
TL WT	SI.01.02 Set up the implementation environment.	Project plan [approved]	Implementation environment [established]

7.4.2.3 SI.02 Software requirements analysis (SI.02, SI.06, SI.07)

This activity analyses the agreed customer's requirements and establishes the validated project requirements.

The activity provides:

— work team review of the project plan to determine task assignment;

— elicitation, analysis and specification of customer's requirements;

— agreement on the customer requirements;

— verification and validation of requirements;

— version control of the software requirements work product;

— traceability between the requirements of the agreement and software requirements specification.

The task list for SI.02 software requirements analysis is given in Table 6.

Table 6 — SI.02 task list

Roles	Task list	Input work products	Output work products
TL WT	SI.02.01 Assign tasks to the work team members. 1) Tasks are assigned to work team in accordance with their role, based on the approved project plan.	Project plan [approved] — Tasks	Project plan [approved] — Tasks
AN CUS	SI.02.02 Document the software requirements specification:	Agreement [baselined]	Requirements specification [initiated]

Table 6 *(continued)*

Roles	Task list	Input work products	Output work products
	— Analyse the requirements defined in the agreement to determinate the scope and feasibility. — Identify and consult information sources (e.g. customer, users, previous software products, documents) to get new or missing requirements. 1) The agreement may have been provided by an internal or an external customer. 2) The content of the work product titled 'Requirements specification' is not limited to the list of attributes listed (e.g. reliability). Additional attributes (e.g. dependability) can be added.	Project plan [approved] — Product Description	
AN	SI.02.03 Document the traceability between the requirements defined in the agreement and the software requirements specification.	Agreement [baselined] Requirements specification [initiated]	Traceability record [initiated]
AN TL	SI.02.04 Verify and obtain approval of the software requirements specification from the AN: — Make corrections until the document is approved by AN.- — Verify the correctness and testability of the software requirements specification and its consistency with the product description. — Review to check that requirements are complete, unambiguous and not contradictory. 1) Conditional task: Initiate a change request if significant changes (those affecting project delivery schedule, project costs, etc.) were needed. 2) The results of the verification are documented in the verification record. 3) ISO/IEC 20246 defines different types of work product reviews.	Traceability record [initiated] Project plan [approved] — Product description	Requirements specification [approved] Verification record [published] Change request [initiated]
CUS AN	SI.02.05 Validate and obtain approval from the customer of the software requirements specification: — Make corrections until the document is approved by the customer.	Requirements specification [approved]	Requirements specification [approved]

Table 6 *(continued)*

Roles	Task list	Input work products	Output work products
	1) Conditional task: Initiate a change request if significant changes (i.e. those affecting project delivery schedule, project costs, etc.) were needed. 2) Software requirements specification should satisfy the needs and the agreed expectations, including the user interface usability. 3) The results of the validation are documented in the validation record.	Traceability record [initiated]	Validation record [published] Change request [initiated]
AN	SI.02.06 Document the preliminary version of the software user documentation. 1) Conditional task: Software user documentation is developed if it is listed in the delivery instructions.	Requirements specification [approved]	Software user documentation [initiated]
AN TL	SI.02.07 Verify and obtain approval of the software user documentation from AN: — Verify consistency of the software user documentation with the requirements specification. — Make corrections until the document is approved by AN. 1) Conditional task: Software user documentation is verified and approved, if it is listed in the delivery instructions. 2) Conditional task: Initiate a change request if significant changes (i.e. those affecting project delivery schedule, project costs, etc.) were needed. 3) The results of the verification are documented in the verification record. 4) Conditional task: Initiate a change request if significant changes (i.e. those affecting project delivery schedule, project costs, etc.) were needed. 5) ISO/IEC 20246 defines different types of work product reviews.	Software user documentation [initiated] Requirements specification [approved] Traceability record [initiated]	Software user documentation [verified] Verification record [published] Change request [initiated]
TL	SI.02.08 Add the software requirements specification, traceability record and software user documentation to the software product in the specification baseline. 1) Software user documentation is added to the baseline if this document is listed in the delivery instructions.	Requirements specification [approved] Software user documentation [verified] Traceability record [initiated]	Software product — Requirements specification [approved] — Software user documentation [preliminary, verified] — Traceability record [initiated]

7.4.2.4 SI.03 Software architectural and detailed design (contributes to SI.03, SI.06, SI.07)

This activity creates the software architecture and identifies and designs the software components which are required to deliver the requirements.

The activity provides:

— work team review of the project plan to determine task assignment;

— design software architecture, software components and associated interfaces;

— detailed design of the software components and interfaces;

— work team review of the requirements specification;

— software design verified and defects corrected;

— verified test cases and test procedures for integration testing;

— traceability of the software requirements to the software design, test cases and test procedures;

— design work products are under version control.

NOTE Software architecture and detailed design can be performed separately according to the project schedule.

The task list for SI.03 software architectural and detailed design is given in Table 7.

Table 7 — SI.03 task list

Roles	Task list	Input work products	Output work products
TL AN DES	SI.03.01 Assign tasks to the work team members. 1) Tasks of AN and DES are related to their roles, according to the project plan.	Project plan [approved] — Tasks	Project plan [approved] — Tasks
AN DES	SI.03.02 Review the software requirements specification. 1) The analyst and the designer review the requirements specification to verify that the requirements specification is correct, complete, unambiguous and not contradictory. 2) ISO/IEC 20246 defines different types of work product reviews.	Requirements specification [baselined] Traceability record [initiated]	Requirements specification [baselined] Traceability record [initiated]
DES AN	SI.03.03 Document the software design:	Requirements specification [baselined]	Software design [initiated]

Table 7 *(continued)*

Roles	Task list	Input work products	Output work products
	— Analyse the requirements specification to generate the architectural design, to define the software components, the internal and external interfaces. — Describe in detail, the appearance and the behaviour of the interface, based on the requirements specification, in a way that resources for its implementation can be planned. — Provide the detail of software components and their interfaces to allow their construction. 1) The AN helps the DES to understand the requirements.	Traceability record [updated]	
AN	SI.03.04 Update the traceability record. 1) Traceability between the requirements the software requirements specification and the software design	Requirements specification [baselined] Software design [initiated] Traceability record [updated]	Traceability record [updated]
DES AN	SI.03.05 Verify and obtain approval of the software design from DES: — Verify correctness of software design documentation, its feasibility and consistency with their requirement specification. — Verify that the traceability record contains the adequate relationships between requirements and the software design elements. — Make corrections are until the document is approved by DES. 1) Conditional task: If significant changes were needed, initiate a change request. 2) The results of the verification are documented in the verification record. 3) Conditional task: If significant changes were needed, initiate a change request. 4) The AN helps the DES to assure that the design meets the requirements.	Software design [initiated] Traceability record [updated] Requirements specification [baselined]	Software design [approved] Verification record [published] Traceability record [updated] Change request [initiated]

Table 7 *(continued)*

Roles	Task list	Input work products	Output work products
DES	SI.03.06 Develop the test cases and test procedures for integration testing. 1) Test cases and test procedures are based on requirements specification and software design. 2) Customer provides testing data, if needed. 3) Test procedures, test cases, unit tests and integration tests are developed.	Requirements specification [baselined] Software design [approved] Traceability record [updated]	Test cases and test procedures [initiated]
DES	SI.03.07 Update the traceability record: — Add the test cases and test procedures to the traceability record.	Test cases and test procedures [initiated] Traceability record [updated]	Traceability record [updated]
DES AN	SI.03.08 Verify and obtain approval of the test cases and test procedures by AN: — Verify consistency among requirements specification, software design and test cases and test procedures. — Make corrections until the document is approved by AN. 1) The results of the verification are documented in a verification record 2) Traceability record is updated, if necessary.	Test cases and test procedures [initiated] Requirements specification [validated] Software design [approved] Traceability record [updated]	Test cases and test procedures [approved] Verification record [published] Traceability record [updated]
TL	SI.03.09 Incorporate work products to the design: — Add the software design, traceability record, test cases, and test procedures to the project repository.	Software design [approved] Test cases and test procedures [approved] Traceability record [updated]	Project repository [updated] — Software design [approved] — Test cases and test procedures [verified] — Traceability record [updated]

7.4.2.5 SI.04 Software construction (contributes to SI.04, SI.06, SI.07)

This activity develops the software code and data from the software design.

The activity provides:

— work team review of the project plan to determine task assignment;

— work team review of the software design to determine software construction sequence;

— coded software components and applied unit tests;

— traceability between software components and software design.

The task list for SI.04 software construction is given in Table 8.

Table 8 — SI.04 task list

Roles	Task list	Input work products	Output work products
TL PR	SI.04.01 Assign tasks to the work team members. 1) Tasks of TL and PR are related to their role, according to the project plan.	Project plan [approved] — Tasks	Project plan [approved] — Tasks
PR DES	SI.04.02 Review the software design. 1) The programmer and the designer review the software design to verify that it is complete, unambiguous. 2) ISO/IEC 20246 defines work product reviews.	Software design [approved] Traceability record [updated]	Software design [approved]
PR	SI.04.03 Develop the software components. 1) The software components are based on the detailed part of the software design.	Software design [approved]	Software components [initiated]
PR	SI.04.04 Develop the unit test cases. 1) To verify that the software components implement the detailed software design.	Software design [approved] Traceability record [updated]	Test cases and test procedures [initiated] — Unit test cases
PR	SI.04.05 Execute the unit test cases: — Correct the defects detected by the unit tests.	Test cases and test procedures [initiated] — Unit test cases Software components [initiated] Traceability record [updated]	Software components [verified]
PR	SI.04.06 Update the traceability record: — Add the software components to the traceability record. — Add the tests cases to the traceability record. — Add the results (e.g. pass, fail) of the test cases to the traceability record.	Software components [verified] Traceability record [updated] Test cases and test procedures	Traceability record [updated]
TL	SI.04.07 Add the software components and traceability record to the software product as part of the baseline. 1) The name of the baseline of the product (e.g. product baseline) should be defined in the version control strategy. 2) Software components and traceability record are stored in the project repository.	Software components [verified] Traceability record [updated]	Project repository [updated] Software product — Software components [baselined] — Traceability record [updated]

7.4.2.6 SI.05 Software integration and tests (contributes to SI.05, SI.06, SI.07)

This activity ensures that the integrated software components satisfy the software requirements.

The activity provides:

— work team review of the project plan to determine task assignment;

— understanding of test cases and procedures and the integration environment;

— integrated software components, corrected defects and documented results;

— traceability of requirements and design to the integrated software work product;

— documented and verified operational and software user documentation;

— verified software baseline.

The task list for SI.05 software integration and tests is given in Table 9.

Table 9 — SI.05 task list

Roles	Task list	Input work products	Output work products
TL PR	SI.05.01 Assign tasks to the work team members. 1) Tasks are related to their role, according to the project plan.	Project plan [approved] — Tasks	Project plan [approved] — Tasks
PR	SI.05.02 Review the test cases and test procedures: — Set or update the testing environment. — Update the integration tests of the test cases and test procedures if needed. 1) The programmer reviews the test cases and test procedures to verify that they are complete. 2) ISO/IEC 20246 defines work product reviews.	Test cases and test procedures [approved] — Integration tests Implementation environment [established] Traceability record [updated]	Test cases and test procedures [verified] — Integration tests
PR DES	SI.05.03 Integrate the software using the software components. 1) DES participates to this task, since DES is responsible of the test cases and procedures.	Software components [verified] Test cases and test procedures [verified] — Integration tests Traceability record [updated]	Software [implemented]
PR CUS	SI.05.04 Perform software integration tests. 1) Results are documented in the test report.	Software [implemented] Test cases and test procedures [verified] — Integration tests	Software [verified] Test report [initiated]
PR	SI.05.05 Correct the defects detected:	Software [verified]	Software [verified]

Table 9 *(continued)*

Roles	Task list	Input work products	Output work products
	— Perform tests until defects are corrected.	Test report [initiated] Test cases and test procedures [verified] — Integration tests Traceability record [updated]	Test report [published]
PR	SI.05.06 Update the traceability record. 1) Traceability record is updated with integration test cases and results of integration tests (e.g. pass, fail).	Software [verified] Traceability record [updated] Test cases and test procedures [initiated]	Traceability record [updated]
PR	SI.05.07 Document the product operation guideline or update the current guideline. 1) Conditional task: Product operation guideline is developed if it is listed in the delivery instructions	Software [verified]	Product operation guideline [initiated]
PR DES	SI.05.08 Verify and obtain approval of the product operation guideline: — Verify consistency of the product operation guideline with the software. — Validate that the documentation is usable for the intended user to perform the tasks. — Make corrections until the document is approved by DES. 1) The results of the verification are documented in the Verification Record.	Product operation guideline [initiated] Software [verified]	Product operation guideline [verified] Verification record [published]
AN	SI.05.09 Develop the software user documentation. 1) Conditional task: Software user documentation is developed if it is listed in the delivery instructions.	Software [verified] Software user documentation [initiated]	Software user documentation [initiated]

Table 9 *(continued)*

Roles	Task list	Input work products	Output work products
AN CUS	SI.05.10 Verify and obtain approval of the software user documentation: — Verify consistency of the software user documentation with the software. — Make corrections are made until the document is approved by CUS. 1) Conditional task: Software user documentation is developed if it is listed in the delivery instructions. 2) The results of the verification are documented in the verification record.	Software user documentation initiated] Software [verified]	Software user documentation [verified] Verification record [published]
TL	SI.05.11 Incorporate the software work products to the baseline: — Add test cases and test procedures, software, traceability record, test report, product operation guideline and software user documentation to the software product. 1) The name of the baseline of the product (e.g. product baseline) should be defined in the version control strategy. 2) Project repository is updated.	Test cases and test procedures [verified] Software [verified] Test report [published] Traceability record [updated] Product operation guideline [verified] Software user documentation [verified]	Project repository [updated] Software work products — Test cases and test procedures [verified] — Software [verified] — Traceability record [updated] — Test report published] — Product operation guideline [verified] — Software user documentation [verified]

7.4.2.7 SI.06 Software product assembly (contributes to SI.O6, SI.O7)

This activity provides the software product to the project manager. The formal delivery and acceptance by the customer of the work products is conducted in PM.04.

The activity provides:

— work team review of the project plan to determine task assignment;

— verified maintenance documentation;

— internal delivery to the PJM of the work products and applicable documentation in accordance with the delivery instructions.

The task list for SI.06 product assembly is given in Table 10.

Table 10 — SI.06 task list

Roles	Task list	Input work products	Output work products
TL WT	SI.06.01 Assign tasks to the work team members. 1) Tasks are related to their role, according to the project plan.	Project plan [approved] — Tasks	Project plan [approved] — Tasks
DES	SI.06.02 Document the maintenance documentation.	Software product [baselined]	Maintenance documentation [initiated]
DES TL	SI.06.03 Verify and obtain approval of the maintenance documentation by the TL: — Verify consistency of maintenance documentation with software product. — Validate that the documentation is usable for the intended user to perform the tasks. — Make corrections until the document is approved by TL. 1) The results are documented in the verification record.	Maintenance documentation [initiated] Software product [baselined]	Maintenance documentation [verified] Verification records [published]
TL	SI.06.04 Add the maintenance documentation to the baseline.	Software product [baselined] Maintenance documentation [verified]	Software product [verified] — Maintenance documentation
TL	SI.06.05 Identify all work products to be stored in the product baseline. 1) TL identifies the final version of all work products to be stored in the project repository. 2) Project repository is updated.	Project plan [approved] — Delivery instructions Software product [baselined] Traceability record [updated]	Software product [baselined] Project repository [updated]
TL PJM	SI.06.06 Assemble the work products according to delivery instructions. 1) TL assembles all the work products for delivery to the PJM. 2) The delivery of the work products to the customer is performed in activity PM.04 by the PJM. 3) A work product is baselined when it should be approved and uploaded in the project repository.	Software product [baselined] Delivery instructions	Software product [baselined]

7.4.3 Incorporation to the project repository

After the incorporation of the work products to the project repository, a version control strategy should be applied.

The work products to be saved in the SI project repository are:

— requirements specification;

— software user documentation;

— software design;

— traceability record;

— test cases and test procedures;

— software components;

— software;

— product operation guideline;

— maintenance documentation;

— test report;

— verification record;

— validation record.

8 Description of roles

An alphabetical list of the roles and suggested competencies description is given in Table 11.

Table 11 — Description of roles

	Role	Competency
1	Analyst	Knowledge and experience eliciting, specifying and analysing the requirements. Knowledge in designing user interfaces and ergonomic criteria. Knowledge of the revision techniques. Knowledge of the editing techniques. Experience on the software development and maintenance.
2	Customer	Knowledge of the customer processes and ability to explain the Customer requirements. The customer (representative) should have the authority to approve the requirements and their changes. The customer includes user representatives to ensure that the operational environment is addressed. Knowledge and experience in the application domain.
3	Designer	Knowledge and experience in the software components and architecture design. Knowledge of the revision techniques. Knowledge and experience in the planning and performance of integration tests. Knowledge of the editing techniques. Experience on the software development and maintenance.
4	Programmer	Knowledge and/or experience in programming, integration and unit tests. Knowledge of the revision techniques. Knowledge of the editing techniques. Experience on the software development and maintenance.
5	Project manager	Leadership capability with experience making decisions, planning, personnel management, delegation and supervision, finances, and software development.
6	Technical leader	Knowledge and experience in the software process domain.
7	Work team	Knowledge and experience according to their roles on the project: TL, AN, DES, and/or PR. Knowledge on the standards used by the customer and/or by the VSE.

9 Description of work products

Table 12 provides an alphabetical list and a definition of the states of work products.

Table 12 — Definition of states of work products

Name of State	Definition of states
Approved	An authorized stakeholder has checked that a work product is ready for delivery and should provide a sign-off for the work product.
Baselined	A work product has been approved and uploaded in the repository. The work product is a stable base for further development.
Cancelled	A planned work product or an element of a work product has been deleted from the project plan.
Completed	A work product is considered complete.
Corrected	Defect(s) identified in a work product has(ve) been removed.
Deleted	A planned work product or an element of a work product has been deleted from the baseline.

Table 12 *(continued)*

Name of State	Definition of states
Delivered	A work product or a set of work products, that has been approved by the customer or authorized stakeholders, has been delivered to a customer.
Established	A tool, an environment or a project repository is operational.
Evaluated	A work product or an element of a work product that has been verified and validated.
Initiated	The workflow for a work product has been initiated.
Implemented	Code has been written to implement the requirement, and the requirement's corresponding design elements have been traced into the code.
Postponed/deferred	A work product or an element of a work product is planned for implementation in a later release.
Preliminary	A work product that has not yet been submitted for approval.
Published	A work product is stored in the repository and made available to all members of the project.
Rejected	A work product or an element of a work product that has not been approved.
Recovered	A work product stored in a repository has been successfully restored.
Reviewed	A work product was presented to the authorized customer or stakeholder(s) as applicable for comment.
Updated	A new version of a work product has been produced and stored in a repository.
Verified	A work product was evaluated (e.g. via test, review) to confirm, through the provision of objective evidence, that it fulfils its specified applicable requirements.
Validated	A work product was evaluated to confirm, through the provision of objective evidence, that fulfils its requirements for its intended use and was approved by authorized stakeholder(s).

Tables 13 to 35 provide an alphabetical list of the input, output and internal process work products, their descriptions, possible states and the source of the work products. The source can be another process or an external entity to the project, such as the customer.

Work product items are based on ISO/IEC/IEEE 15289 information items with some exceptions.

Information items may be combined or subdivided consistent with the project, service, or processes, phases, and stakeholder needs by a VSE.

The states of the work product (e.g. evaluated, verified, tested, baselined) give the information to the project team about the type of work (tasks) already done on the work product This information can be used to start the next task that uses the work product as an input.

Work products are identified with a unique code WP.XX where XX is a sequential number. These codes should not be used in the descriptions of activities and tasks to facilitate readability.

A work product is available on the media identified in the project plan. As an example, a work product can be available as a printed document, as an item of an email, or as an item of an electronic tool.

Table 13 — Acceptance record

WP ID	Name	Source
WP.01	Acceptance record	Project management

Description

Documents the customer acceptance of the deliverables (i.e. list of work products to be delivered to customer) of the project.

It may have the following characteristics:

- Record of the receipt of the delivery
- Identifies the date received
- Identifies the delivered elements
- Records the validation of any customer acceptance criteria defined
- Identifies any open issues (if applicable)
- Signed by customer and project manager

The applicable states are: initiated and signed.

NOTE 1 The acceptance record includes the “acceptance criteria”.
NOTE 2 The development team asks for clarification if needed.

Table 14 — Agreement

WP ID	Name	Source
WP.02	Agreement	Customer Project management

Description

Description of work to be done related to software development.

It may Include:

- Product description
  - Purpose
  - General customer requirements
- Scope description of what is included and what is not included
- Objectives of the project
- List of work products to be delivered to customer (i.e. deliverables)
- Signed by customer and project manager

The applicable states are: approved, reviewed and baselined.

Table 15 — Change request

WP ID	Name	Source
WP.03	Change request	Software implementation Customer Project management

Description

Identifies a software, or documentation problem or desired improvement, and requests modifications.

It may have the following characteristics:

- Identifies purpose of change
- Identifies request states (e.g. initiated, evaluated, accepted and rejected)
- Identifies requester contact information (e.g. customer, project manager)
- Impacted software product, component
- Impact to operations of existing software defined
- Impact to associated documentation defined
- Criticality of the request, date needed

The applicable states are: initiated, evaluated, accepted and rejected.

Table 16 — Correction register

WP ID	Name	Source
WP.04	Correction register	Project management

Description

Identifies activities established to correct a deviation or problem concerning the accomplishment of a plan.

It may have the following characteristics:

- Identifies the initial problem
- Defines a solution
- Identifies corrective actions taken
- Identifies the person responsible for completion of defined actions
- Identifies the open date and target closure date
- Contains a state indicator
- Indicates follow up actions

The applicable states are: initiated, approved and completed.

Table 17 — Implementation environment

WP ID	Name	Source
WP.05	Implementation environment	Software implementation

Description

Describes the tools selected (e.g. compilers, design tools, construction and tests) for the project.

The applicable state is: established

Table 18 — Maintenance documentation

WP ID	Name	Source
WP.06	Maintenance documentation	Software implementation

Description

Describes the software product and the environment used for development and testing (e.g. compilers, design tools, construction and tests).

It may have the following characteristics:

— Includes or refers to all software product elements developed during implementation

— Identifies environment used for development and testing (e.g. compilers, design tools, construction and tests tools)

It is written in terms that maintenance personnel can understand.

The applicable states are: validated and baselined.

Table 19 — Meeting record

WP ID	Name	Source
WP.07	Meeting record	Project management

Description

Records the agreements established with customer and/or work team.

It may have the following characteristics:

— Purpose of meeting

— Name of attendees

— Date, place held

— Reference to previous minutes

— What was accomplished

— Identifies issues raised

— Any open issues

— Agreements

— Next meeting, if any.

— Signed by customer and project manager

The applicable state is: approved.

Table 20 — Product operation guideline

WP ID	Name	Source
WP.08	Product operation guideline	Software implementation

Description
Contains the necessary information to install and manage the software.
It may have the following characteristics:

- Criteria for operational use
- A description of how to operate the product including:
  - Operational environment required
  - Supporting tools and material (e.g. user manuals) required
  - Possible safety warnings
  - Start-up preparations and sequence
  - Frequently asked questions (FAQ)
  - Sources of further information and help to operate the product
- Certification and safety approvals
- Warranty and replacement instructions

It is written in terms that the personnel responsible for the operation can understand.

The applicable states are: verified and baselined.

Table 21 — Progress status record

WP ID	Name	Source
WP.09	Progress status record	Project management

Description
Records the status of the project against the project plan.
It may have the following characteristics:

- Status of actual tasks against planned tasks
- Status of actual results against established objectives/goals
- Status of actual resource allocation against planned resources
- Status of actual cost against budget estimates
- Status of actual time against planned schedule
- Status of actual risk and mitigation against previously identified
- Record of any deviations from planned tasks and reason why.
- Signature of the project manager

The applicable state is: published.

Table 22 — Project plan

WP ID	Name	Source
WP.10	Project plan	Project management

Description

Presents how the project processes and activities are executed to assure the project's successful completion, and the quality of the work products to be delivered according to the schedule.

It includes the following elements which may have the characteristics as follows:

- Product description
  - Purpose
  - General Customer requirements
- Scope description of what is included and what is not included
- Objectives of the project
- Development cycle selected (e.g. waterfall, iterative, incremental, evolutionary, agile)
- Deliverables - list of work products to be delivered to customer
- Tasks, including verification, validation and reviews with customer and work team, to assure the quality of work products. tasks may be represented as a work breakdown structure (WBS).
- Estimated duration of tasks
- Resources (humans, materials, standards, equipment and tools) including the required training, and the schedule when the resources are needed.
- Composition of work team
- Schedule of the project tasks, the expected start and completion date for each task, and the relationship and dependencies of the tasks.
- Estimated effort and cost
- Identification of project risks
- Version control strategy
  - Work product repository tools or mechanism identified
  - Location and access mechanisms for the repository specified
  - Version identification and control defined
  - Description of the baselines selected for the project: e.g. requirements baseline (i.e. agreement), design baseline and product baseline
  - Backup and recovery mechanisms defined
  - Storage, handling and delivery (including archival and retrieval) mechanisms specified

Table 22 *(continued)*

WP ID	Name	Source
— Delivery Instructions — Elements required for product release identified (i.e. hardware, software, documentation, etc.) — Delivery requirements — Sequential ordering of tasks to be performed — Applicable releases identified — Applicable media identified (e.g. paper, electronic) — Identifies all delivered software components and work products (e.g. software user documentation) with version information — Identify any necessary backup and recovery procedures Signed by customer and project manager. The applicable states are: validated, accepted, updated and reviewed.		

Table 23 — Project repository

WP ID	Name	Source
WP.11	Project repository	Project management
Description Electronic container to store project work products and deliverables. It may have the following characteristics: — Stores project work products — Stores released work products — Storage and retrieval capabilities — Ability to browse content — Listing of contents with description of attributes — Sharing and transfer of work products between affected groups — Effective controls over access — Maintain work product descriptions — Recovery of archive versions of work products — Ability to report work products status — Changes to work products are tracked to change requests The applicable states are: recovered and updated.		

Table 24 — Project repository backup

WP ID	Name	Source
WP.12	Project repository backup	Project management
Description Repository used to back up the project repository and, if necessary, to recover the information. The applicable state is: recovered		

Table 25 — Requirements specification

WP ID	Name	Source
WP.13	Requirements specification	Software implementation

Description

Provides the software requirements such as functions, performance, design constraints, and attributes.

It may have the following characteristics:

- Introduction – general description of software and its use within the context of the customer.
- Requirements description:
  - Functional suitability – Degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions:
    - Functional completeness (i.e. degree to which the set of functions covers all the specified tasks and user objectives)
    - Functional correctness (i.e. degree to which a product or system provides the correct results with the needed degree of precision)
    - Functional appropriateness (i.e. degree to which the functions facilitate the accomplishment of specified tasks and objectives. As an example: a user is only presented with the necessary steps to complete a task, excluding any unnecessary steps)
  - Performance efficiency - Performance relative to the amount of resources used under stated conditions (e.g. time behaviour, resource utilisation, capacity)
  - Compatibility - Degree to which a product or component can exchange information with other products, systems or components, and/or perform its required functions, while sharing the same hardware or software environment (e.g. co-existence, interoperability)
  - Usability - Degree to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use (e.g. appropriateness recognizability, learnability, operability, user error protection, user interface aesthetics, accessibility)
  - Reliability – Degree to which a product or component performs specified functions under specified conditions for a specified period of time (e.g. maturity, availability, fault tolerance, recoverability)
  - Maintainability – Degree of effectiveness and efficiency with which a product can be modified by the intended maintainers (e.g. modularity, reusability, analysability, modifiability, testability)
  - Portability – Degree of effectiveness and efficiency with which a product, or component can be transferred from one hardware, software or other operational or usage environment to another (e.g. adaptability, instability, replaceability)
  - Security - Degree to which a product 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 (e.g. confidentiality, integrity, non-repudiation, accountability, authenticity)
  - Design and construction limitations/constraints – needs imposed by the customer
  - Legal and regulative – needs imposed by laws, regulations, etc.

Each requirement is identified (e.g. Req. #01), unique and it is verifiable or can be assessed.

The applicable states are: verified, validated and baselined.

Table 26 — Software

WP ID	Name	Source
WP.14	Software	Software implementation
Description Software item (e.g. software source and executable code) for a customer, constituted by a collection of integrated software components. The applicable states are: tested, corrected and baselined.		

Table 27 — Software components

WP ID	Name	Source
WP.15	Software components	Software implementation
Description A set of related code units. The applicable states are: unit tested, corrected and baselined.		

Table 28 — Software design

WP ID	Name	Source
WP.16	Software design	Software implementation

Description

Textual and graphical information on the software structure.

This structure may include the following parts:

- Architectural or high-level software design
  - Describes the overall software structure:
  - Identifies the required software components
  - Identifies the relationship between software components
  - Consideration is given to any required:
    - Software performance characteristics
    - Hardware, software and human interfaces
    - Security characteristics
    - Database design requirements
    - Error handling and recovery attributes
- Low-level or detailed software design - includes details of the software components to facilitate its construction and test within the programming environment:
  - Provides detailed design (i.e. can be represented as a prototype, flow chart, entity-relationship diagram, pseudo-code, etc.)
  - Provides format of input/output data
  - Provides specification of data storage needs
  - Establishes required data naming conventions
  - Defines the format of required data structures
  - Defines the data fields and purpose of each required data element
  - Provides the specifications of the program structure

The applicable states are: verified, approved.

Table 29 — Software product

WP ID	Name	Source
WP.17	Software product	Software implementation

Description

Provides a uniquely identified and consistent set of software products including:

- Requirements specification
- Software design
- Traceability record
- Software components
- Software
- Test cases and test procedures
- Test report
- Product operation guideline
- Software user documentation
- Maintenance documentation

The applicable states are: delivered and accepted.

Table 30 — Software user documentation

WP ID	Name	Source
WP.18	Software user documentation	Software implementation

Description

Describes the way of using the software based on the user interface.

It may have the following characteristics:

- User procedures for performing specified tasks using the software
- Installation and de-installation procedures
- Brief description of the intended use of the software (the concept of operations)
- The supplied and required resources
- Needed operational environment
- Availability of problem reporting and assistance
- Procedures to access and exit the software
- Lists and explains software commands and messages to the user
- As appropriate for the identified risk, it includes warnings, cautions, and notes, with corrections
- It includes troubleshooting and error correction procedures.

It is written in terms understandable by users.

The applicable states are: preliminary, verified and baselined.

Table 31 — Test cases and test procedures

WP ID	Name	Source
WP.19	Test cases and test procedures	Software implementation

Description

Provides elements needed to test code.

Test cases may include:

- Identifies the test case
- Test items (unit test cases, integration tests)
- Input specifications
- Output specifications
- Environmental needs
- Special procedural requirements
- Interface dependencies

Test procedures may include:

- Identifies: test name, test description and test completion date
- Identifies potential implementation issues
- Identifies the person who completed the test procedure
- Identifies prerequisites
- Identifies procedure steps including the step number, the required action by the tester and the expected results

The applicable states are: verified and baselined.

Table 32 — Test report

WP ID	Name	Source
WP.20	Test report	Software implementation

Description

Documents the tests execution.

It may include:

- A summary of each defect
- Identifies the related test case
- Identifies the tester who found each defect
- Identifies the severity for each defect
- Identifies the affected function(s) for each defect
- Identifies the date when each defect originated
- Identifies the date when each defect was resolved
- Identifies the person who resolved each defect

The applicable states are: initiated, [defects] corrected, baselined.

Table 33 — Traceability record

WP ID	Name	Source
WP.21	Traceability record	Software implementation

Description

Documents the relationship between the requirements included in the requirements specification, software design elements, software components, test cases and test procedures.

It may include:

- Identification of the individual requirements of the requirements specification to be traced
- Provides forward and backward mapping of requirements to software design elements, software components, test cases, test procedures and test results.

The applicable states are: verified, baselined and updated.

Table 34 — Validation record

WP ID	Name	Source
WP.22	Validation record	Software implementation

Description

Documents the validation execution.

It may include the record of:

- Participants
- Date
- Place
- Duration
- Validation checklist
- Passed items of validation
- Failed items of validation
- Pending items of validation
- Defects identified during validation
- Signature of project manager
- Signature of customer

The applicable states are: initiated, approved.

Table 35 — Verification record

WP ID	Name	Source
WP.23	Verification record	Project management Software implementation

Description
Documents the verification execution.
It may include the record of:

- Participants
- Date
- Place
- Duration
- Verification checklist
- Passed items of verification
- Failed items of verification
- Pending items of verification
- Defects identified during verification
- Signature of author

The applicable states are: initiated, approved.

10 Software tools

10.1 General

Tables 36 and Table 37 provide software tools that can be used to perform process activities.

10.2 Project management process tools

The list of software tools that can be used to perform project management activities are given in Table 36.

Table 36 — Project management tools

Activity	Resource list
Project planning Project plan execution project assessment and control Project closure	Tools allowing documentation, management and control the project plan and the use and management of the project repository.

10.3 Software implementation process tools

The list of software tools that can be used to perform software implementation activities are given in Table 37.

Table 37 — Software implementation tools

Activity	Resource list
Software implementation initiation Software requirements analysis Software architectural and detailed design Software construction Software integration and tests Product delivery	Documentation tools
Software requirements analysis	Requirements specification tools
Software architectural and detailed design	Software design tools
Software construction	Construction tools
Software integration and tests	Tests tools, defect tracking tools

Annex A
(informative)

Support process for the software Basic profile

A.1 General

This annex describes a process which can be added to the software Basic profile to enable VSEs to support the software product which they developed in a controlled way. The Basic profile is targeted at VSEs developing a single product by a single work team.

The Basic profile concludes at the delivery of the software product. This means that any customer support required, such as fixing software problems, can be done by the VSE without controls. Uncontrolled changes to the software product can impact on product quality and product stability. VSEs who do not manage ongoing customer support effectively risk damage to their reputation, which can affect their business and their ability to retain current customers and acquire new customers.

Uncontrolled and unplanned customer support work can also be disruptive to a VSE's development schedules. It is often be prioritised above development work. Those with the skills and knowledge may be working on a project.

Before a VSE delivers the software product (e.g. documentation, code), its management discusses with the customer if and how the VSE will support the software product. The VSE decides whether support resources will be dedicated or whether they will also be expected to work on an approved project. If support is to be supplied by those working on a project, an estimate of the time needed for support is included in the project plan to reduce the risk of project slippage due to support requirements.

The annex describes a support process and its 3 activities which can be added to the project management and software implementation processes of the software Basic profile to enable a VSE to deliver effective software support after the software product should be delivered to the customer.

A.2 Scope

To use this annex, the VSE should fulfil the following entry conditions:

— should have implemented or is implementing the software Basic profile;

— should have completed or is completing the development of a software product;

— should have delivered or is delivering a software product to a customer.

The software support process is composed of the following three activities:

— service level management;

— incident management;

— production change management.

These activities are interlinked and are also linked into the project management (PM) process and the software implementation (SI) process of the software Basic profile. The software support process is performed at the same time as the PM process for the VSE's current projects, continuing support for deliverables from previous projects.

The scope of software support (SS) is determined by the VSE as part of a service level agreement. The scope is normally restricted to defect correction and minor changes in requirements. Larger software changes are excluded from support and handled as a project through the PM and SI processes of the Basic profile.

A.3 SS process purpose

The purpose of the software support process is to establish and carry out in a systematic way the ongoing support of the software product delivered to a VSE's customer to deliver a support service to the customer. This service may be delivered as part of an agreement.

A.4 SS process outcomes

— SS.O1. Definition, review and acceptance by the customer of the service level agreement for the software support is included as part of the agreement.

— SS.O2 Ongoing communications with customer on the software support service.

— SS.O3. Manage incidents to restore service to the customer in accordance with the SLA.

— SS.O4 Control and manage change to production software.

— SS.O5 Develop, test and deploy production changes.

A.5 Overview of the SS process

Figure A.1 provides an overview of the processes of the Basic profile illustrating the interactions of the additional software support process.

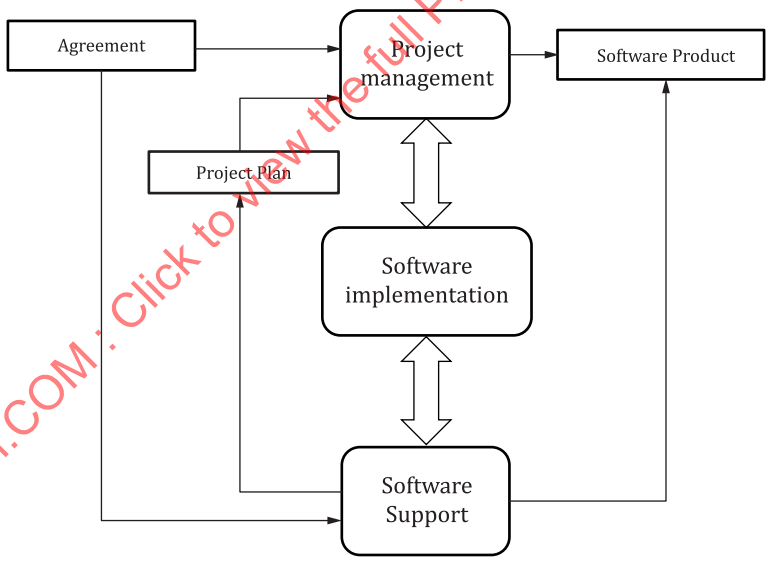

Figure A.1 — Addition of software support to Basic profile processes

The software support process commences for each project after the product delivery should be completed and continues for the life of the software product. The software support process is managed by a support manager who may be the project manager of the software product or someone who may take this role on a part-time basis in a VSE. It is important that roles are allocated to enable the support manager to focus on customer support when required. A VSE may have a single team performing SI activities and SS activities or it may choose to have a separate SS team. In the case of a single team, the support manager should handle prioritisation conflicts between project and support activities, if any.

A.6 Roles

Table A.1 provides a list of roles for the software support process.

Table A.1 — SS roles

Role	Competency
Customer	As per the Basic profile
Project manager	As per the Basic profile
Support manager	Leadership capability with experience of delivery of customer service and prioritisation.
Support team	Knowledge and experience of software processes according to their role which may include software development or may be software deployment and support

A.7 Overview of the SS process

Figure A.2 provides an overview of the software support process, its set of 3 activities and associated work products.

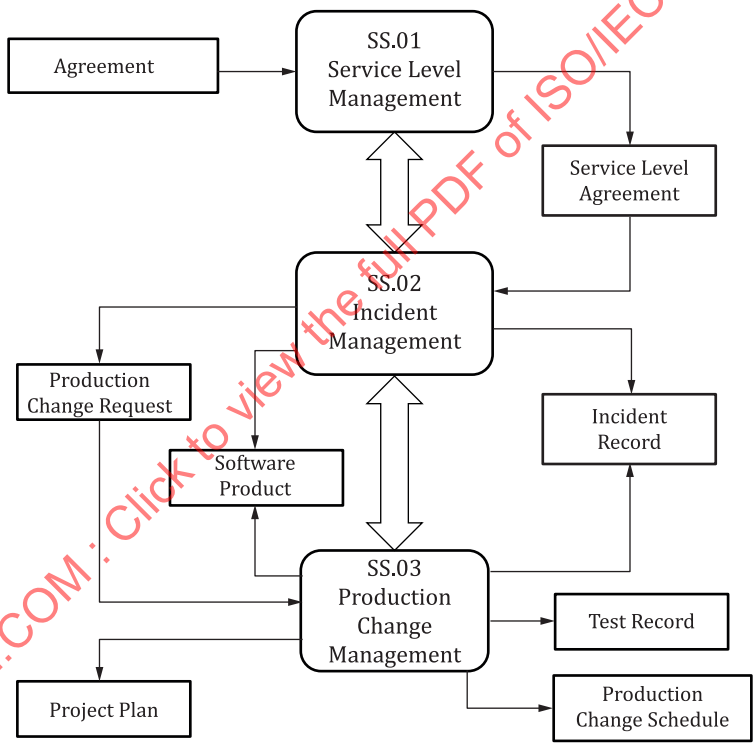

Figure A.2 — Overview of the activities of the software support process

A.8 SS Work products

Table A.2 provides a list of the work products specific to the software support process.

Table A.2 — SS Work products

Name	Activity	Input and output work products	Source
Agreement – service requirements	Service level management	Input	Customer
Service level agreement	Service level management	Output	Support manager
Incident record	Incident management Production change management	Output Output	Support team member Support team member
Software product	Incident management Production change management	Input Output	Support team member Support team member
Production change request	Incident management Production change management	Output Input	Support team member Support team member
Production change schedule	Production change management	Output	Support manager
Test record	Production change management	Output	Support team member
Project plan	Production change management	Output	Support manager

A.9 SS activities

A.9.1 General

The software support process should have the following activities:

— SS.01 Service level management (SLM)

— SS.02 Incident management (IM)

— SS.03 Production change management (PCM)

A.9.2 SS.01 Service level management (SLM)

A.9.2.1 General

Service level management is the activity which establishes and supervises the software support service which is delivered to the customer. Its key work product is the service level agreement which specifies how software support is delivered and defines the scope of support.

A.9.2.2 SLM purpose

The purpose of this activity is to specify the support services that are being delivered by the VSE to customers and to maintain good customer relationships by ensuring the services provided match the communicated needs and expectations of customers.

Table A.3 provides a list of tasks for the SLM activity.

Table A.3 — SS.01 task list

Role	Task list	Input work products	Output work products
SM	SLM.01.01 Develop the service level agreement. 1) To achieve a team understanding of the service required. The service level agreement includes service support hours, incident classifications and emergency change criteria as defined in Table A.6.	Agreement [approved] Service level agreement [initial] Requirements specification [baselined]	Service level agreement [initiated]
SM CUS	SLM.01.02 Obtain an approval of the service level agreement of the customer. 1) The service level agreement can be included into an agreement (e.g. contract) if required.	Service level agreement [initiated]	Service level agreement [approved]
SM CUS	SLM.01.03 Manage escalations from the Incident management activity. 1) The SM also manages regular communications with the customer.	Service level agreement [approved]	Meeting record [published]

A.9.3 SS.02 Incident management (IM)

A.9.3.1 General

Incident management is the activity which handles customer support communications, recording and restoring service on an ongoing basis. Its key work product is the incident record. VSEs decides the scope of this activity. Incident management always includes software break/fix support and may also include support for minor improvements. A VSE may wish to use incident management to record customer queries or may decide to handle them informally. The scope of incident management is documented in the service level agreement.

Important concepts of incident management are:

— the recording of all incidents;

— the priority placed on restoring customer service rather than fixing the cause of the incident. Many incidents can be fixed by workaround, meaning that a software change can be planned, tested and scheduled with lower risk than an immediate software deployment.

A.9.3.2 IM purpose

The purpose of this activity is to restore the customer's ability to use the software after a failure of the software or a customer misunderstanding of the software, or to prevent failures occurring. This may be achieved through customer education, a software change or an interim workaround followed by a software change delivered through the production change management activity.

Table A.4 provides a list of tasks for the IM activity.

Table A.4 — SS.02 task list

Role	Task list	Input work products	Output work products
ST CUS	IM.01.01 Record all reported or identified incidents.	Meeting record [published]	Incident record [published]
PJM CUS	IM.01.02 Classify incidents according to the criteria set in the service level agreement and link to other necessary records.	Service level agreement [approved] Software product [delivered]	Incident record [updated]
ST CUS	IM.01.03 Investigate incidents, communicate to customer and identify any required software changes.	Incident record [updated]	Incident record [updated] Production change request [initiated] Meeting record [updated]
ST SM CUS	IM.01.04 Restore service: — Restore service either via workaround or the production change management activity if a fix is required urgently, notify customer and obtain customer agreement. — If customer agreement cannot be obtained escalate to the SLM activity.	Incident record [updated] Production change request [approved]	Incident record [updated] Meeting record [updated]
ST	IM.01.05 Close incident. 1) The incident is closed with a temporary solution, e.g. a workaround and a future production change request.	Incident record [updated]	Incident record [approved] Production change request [initiated]

A.9.4 SS.03 Production change management (PCM)

A.9.4.1 General

Production change management is the set of activities to deploy a change into a software product which should already be delivered to a customer.

There are two types of change which the VSE manages:

— Normal change to the software which can be planned and managed to minimize risk to the software deployment.

  — EXAMPLE: A normal change can include delivering a set of tested software fixes into the production environment.

— Emergency change to the software – an unplanned/unscheduled change for immediate deployment, usually to resolve a high severity incident.

  — An emergency change is in principle handled in the same way as a normal change, but the schedule is accelerated in accordance with the urgency of the change and the output products may be completed products after the change should be deployed into production. Emergency changes are used when necessary as they reduce productivity and increase risk. Testing is completed for all changes including emergency changes.

A.9.4.2 PCM purpose

The purpose of this activity is to manage/minimise the risks associated with delivering production software changes through planning, testing, control and communication.

Table A.5 provides a list of tasks for the PCM activity.

Table A.5 — SS.03 task list

Role	Task list	Input work products	Output work products
SM CUS	PCM.01.01 Define and approve the production change build, test & deployment procedure(s): — Define criteria for emergency change and normal change. — Include build, test and deployment and the planned service change schedule.	Software product [delivered] Production change schedule [initiated]	Production change schedule [approved]
SM ST	PCM.02.02 Evaluate production change request. 1) The VSE decides that the support manager does not have to be involved for every change. Criteria for involvement can be documented in PCM.01.01.	Incident record [updated] Production change request [initiated] Production change schedule [approved] Software product [baselined] Project plan [approved]	Incident record [updated] Production change request [evaluated] Production change schedule [updated]
ST CUS	PCM.02.03 Update production change schedule to include the proposed deployment date.	Incident record [updated] Production change request [evaluated]	Incident record [updated] Production change schedule [updated]
ST PJM	PCM.02.04 Design and build the software change and documentation: — Consult with ongoing project management and software implementation process for overlaps. 1) Resources are incorporated into the project plan, or the project plan is changed to accommodate redeployment of resources to deliver the change.	Software product [baselined] Project plan [approved]	Software product [updated] Project plan [updated] Meeting record [updated]

Table A.5 *(continued)*

Role	Task list	Input work products	Output work products
ST	PCM.02.05 Test the software change and document results. 1) Test records can be kept in the same repository as project test records.	Incident record [updated] Software product [updated]	Test record [published]
SM ST	PCM.02.06 Approve and deploy the software change: — Ensure appropriate support is provided.	Software product [updated] Test record [published]	Software product [updated] Production change request [approved] Incident record [approved]

A.10 Work product description

Table A.6 to A.9 provide an alphabetical list of only the SS work products (i.e. work product not already defined in the software Basic profile).

Table A.6 — Incident record

WP ID	Name	Source
WP.A.01	Incident record	Software support
Description Contains the details of an incident. It may include: — Incident identifier — Name and contact details of the reporter of the event — Impacted software product, software component — Details of the incident — Severity of the incident The applicable states are: initiated, evaluated, approved, rejected.		

Table A.7 — Production change request

WP ID	Name	Source
WP.A.02	Production change request	Software support

Description

Request to implement a change into the production software.

It may include:

- Change identifier
- Identification of purpose of change
- Identification of date change needed by
- Identification of change requester
- Impacted software product, software component
- Customer impact of change

The applicable states are: draft, approved, rejected.

Table A.8 — Production change schedule

WP ID	Name	Source
WP.A.03	Production change schedule	Software support

Description

List of approved changes to be deployed into production with planned dates.

It may include:

- Future planned releases of changes
- Customer business events where changes are not to be scheduled
  - e.g. financial year end

The applicable states are: draft, approved, rejected.

Table A.9 — Service level agreement

WP ID	Name	Source
WP.A.04	Service level agreement	Software support

Description

Description of work to be done related to the support of the software product.

It may include:

- Service hours, contact details and accepted ways of contacting the VSE
- Description of what is included in the support service, in particular whether minor improvements are included and treated as incidents
- Definition of incident types and severities. There may be different types of incident defined for break/fix, answering customer queries and minor improvements
- Initial response times for incidents by type and severity
- Target resolution times for incidents by type and severity
- Definition of change types and criteria
- Customer escalation process with contact details
- Warranty/pricing/penalty arrangements

The applicable states are: initiated, approved, and baselined.

Annex B
(informative)

Additional testing tasks to the software Basic profile

B.1 General

This annex adds a set of tasks, one role and several work products to the software Basic profile for VSEs with the aim to better support software testing. Rigorous testing, of the software and the associated documentation, can help reduce the risk of failures occurring during operation. When defects are detected, and subsequently fixed, this contributes to the quality of the software product. In addition, software testing can fulfil to meet contractual or legal requirements or industry-specific standards.

This annex aligns to the software test processes of the ISO/IEC/IEEE 29119 series.

This annex can be used by VSEs that want to achieve desired quality of software test processes.

B.2 Naming, diagramming and conventions

Task - each testing task is identified by an activity ID and a consecutive number. An additional notation is added to highlight the testing tasks (TT). For example, if two testing tasks are added to task PM.01.10, they are identified as follows: PM.01.10.TT1 and PM.01.10.TT2.

Work products – testing work products are identified as WP. CXX. A new state has been added.

Roles – A new role, tester, has been added.

B.3 Additional tasks to the project management (PM) process

B.3.1 Project planning activity (PM.01)

Table B.1 provides the tasks for PM.01.

Table B.1 — Testing task list for PM.01

Roles	Task list	Input work products	Output work products
PJM TL TES	PM.01.10.TT1. Define the test plan: — Identify risk mitigation approaches; — Design test strategy including test phases, test types, features to be tested, test design techniques, test completion criteria; — Identify metrics to be used for test monitoring; — Identify test data requirements; — Identify test environment requirements; — Identify test deliverables; — Estimate the required resources; — Schedule each required test activity	Agreement [approved] Project plan [initiated]	Test plan [initiated]
PJM TL TES	PM.01.10.TT2. Verify the test plan: — Make corrections until the document is approved by PJM. 1) The results of the verification are documented in a verification record.	Test plan [initiated]	Test plan [verified] Verification record [approved]

B.3.2 Project plan execution activity (PM.02)

Table B.2 provides the tasks for PM.02.

Table B.2 — Testing task list for PM.02

Roles	Task list	Input work products	Output work products
PJM TL TES	PM.02.01.TT1. Monitor testing: — Collect and record test measures. — Monitor progress against the test plan. — Identify and record divergence from planned testing activities. — Identify new risks. 1) The results of monitoring are documented in a test status report.	Test plan [verified]	Test status report [initiated]

B.3.3 Project assessment and control activity (PM.03)

Table B.3 provides the tasks for PM.03.

Table B.3 — Testing task list for PM.03

Roles	Task list	Input work products	Output work products
PJM TL TES	PM.03.01.TT1. Control testing: — Perform actions necessary to implement the test plan. — Perform actions necessary to implement control directives. — Identify actions necessary to manage the divergence of actual testing from planned testing. — Identify means of treating newly identified and changed risks. — Update test plan if appropriate.	Test plan [verified]	Test plan [updated]

B.3.4 Project closure activity (PM.04)

Table B.4 provides the tasks for PM.04.

Table B.4 — Testing task list for PM.04

Roles	Task list	Input work products	Output work products
PJ TL TES	PM.04.01.TT1. Complete testing: — Clean up test environment.	Test plan [verified] Test environment [updated]	Test environment [updated]
PJM TL TES	PM.04.02.TT1. Archive test assets: — Identify test assets which may be of use later and save them in the project repository	Project repository [established]	Project repository [updated]

B.4 Additional tasks to the software implementation (SI) process

B.4.1 Software implementation initiation activity (SI.01)

Table B.5 provides the task for SI.01.

Table B.5 — Testing task list for SI.01

Roles	Task list	Input work products	Output work products
TL TES	SI.01.02.TT1 Set up the test environment: — Set up test data to support the testing. — Set up test tools to support the testing — Install and configure the test item on the test environment. — Verify that the test environment meets the test environment requirements.	Test plan [verified]	Test environment [established]

B.4.2 Software requirements analysis activity (SI.02)

Table B.6 provides the task for SI.02.

Table B.6 — Testing task list for SI.02

Roles	Task list	Input work products	Output work products
TL TES	SI.02.04.TT1 Identify test cases: — Perform test analysis which includes identifying features to be tested. — Define and prioritize test conditions for each feature based on analysis of the test basis. — Identify test cases (i.e. ID and name) and record them in preliminary test cases.	Test plan [verified] Requirements specification [baselined]	Test cases [preliminary]

B.4.3 Software architectural and detailed design activity (SI.03)

Table B.7 provides the tasks for SI.03.

Table B.7 — Testing task list for SI.03

Roles	Task list	Input work products	Output work products
TL TES	SI.03.05.TT1 Develop test cases: — Develop test cases according to the test strategy defined in the test plan. 1) Test cases are aimed at integration, system (e.g. functional, performance, security) and acceptance testing. 2) Test design techniques and test completion criteria to use are specified in the Test Plan. 3) Test design techniques and measures are defined in ISO/IEC/IEEE 29119-4: Test techniques.	Test plan [verified] Requirements specification [baselined] Software design [verified] Traceability record [verified] Test cases [preliminary]	Test cases [updated]
TL TES PJM	SI.03.05.TT2 Verify test cases: — Review test cases. — Make corrections until test cases are approved by PJM 1) ISO/IEC 20246 defines work product reviews. 2) The results of the verification are documented in a verification record.	Test plan [verified] Requirements specification [baselined] Software design [baselined] Test cases [updated]	Test cases [verified] Verification record [published]
TL TES	SI.03.05.TT3 Develop test procedures: — Order test cases according to dependencies described by pre-conditions and post-conditions and other testing requirements. — Identify any test data and test environment requirements that are not already included in the test plan. — Prioritize the test procedures. 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. 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.	Test plan [verified] Requirements specification [baselined] Software design [verified] Traceability record [verified] Test cases [verified]	Test procedures [initiated]
TL TES	SI.03.06.TT1. Update the traceability record: — Add the test cases and test procedures to the traceability record. 1) The traceability between the test basis, feature sets, test cases, and test procedures (and/or automated test scripts) can be recorded.	Test procedures [initiated] Test cases [verified] Traceability record [verified]	Traceability record [updated]

Table B.7 *(continued)*

Roles	Task list	Input work products	Output work products
AN DES TES	SI.03.07.TT1. Verify the test procedures: — Verify consistency among requirements specification, software design and test cases and test procedures. 1) The results of the verification are documented in a verification record. 2) Traceability record is updated, if necessary. 3) ISO/IEC 20246 defines work product reviews	Test cases [verified] Test procedures [initiated] Requirements specification [baselined] Software design [baselined] Traceability record [updated]	Test cases [verified] Test procedures [verified] Verification record [published] Traceability record [updated]
TL	SI.3.8.TT1. Incorporate work products to the design baseline: — Add the software design, traceability record, test cases, and test procedures to the project repository.	Software design [approved] Test cases [verified] Test procedures [verified] Traceability record [updated]	Project repository [updated] — Software design [baselined] — Test cases [verified] — Test procedures [verified] — Traceability record [updated]

B.4.4 Software construction activity (SI.04)

Table B.8 provides the tasks for SI.04.

Table B.8 — Testing task list for SI.04

Roles	Task list	Input work products	Output work products
PR	SI.04.04 Develop the unit test cases. 1) To verify that the software components implement the detailed Software Design.	Software design [baselined] Test cases [verified] Traceability record [verified]	Test cases — Unit test cases [implemented]
PR	SI.04.05 Execute the unit test cases: — Execute the unit tests and correct detected defects.	Test cases — Unit test cases [implemented] Software components [initiated] Traceability record [updated]	Test cases — Unit test cases [completed] Software components [verified]

B.4.5 Software integration and tests activity (SI.05)

Table B.9 provides the tasks for SI.05.

Table B.9 — Testing task list for SI.05

Roles	Task list	Input work products	Output work products
TES TL	SI.05.02.TT1 Maintain the test environment: — Verify that the test environment meets the test environment requirements and do corrections.	Test plan [verified] Test environment [established]	Test environment [updated]
PR TES TL	SI.05.04.TT1 Perform integration tests: — Verify that the units together operate as specified in the software design. — Verify that the interfaces operate as specified in the software design. 1) Results are documented in the test execution documentation. 2) Test incidents are documented in the test incident report.	Test plan [verified] Test environment [updated] Software [implemented] Test cases — integration tests [verified] Test procedures — integration tests [verified] Software design [verified]	Software [verified] Test cases — integration tests [completed] Test procedures — integration tests [completed] Test execution documentation [published] Test incident report [published]
TES TL	SI.05.04.TT2 Perform system tests: — Verify that the system operates as specified in the requirements specification. — Perform system tests defined in the test plan (e.g. functional, performance, security) 1) Results are documented in the test execution documentation. 2) Test incidents are documented in the test incident report. 3) The test procedures can have been scripted for automated execution or recorded in a test specification for manual test execution.	Test plan [verified] Test environment [updated] Software [implemented] Test cases — System tests [verified] Test procedures — System tests [verified]	Software [verified] Test cases — System tests [completed] Test procedures — System tests [completed Test execution documentation [published] Test incident report [published]
PR	SI.05.05.TT1. Manage test incidents to closure:	Software [verified]	Software [verified]

Table B.9 *(continued)*

Roles	Task list	Input work products	Output work products
TES TL	— Decide disposition of incidents. — Perform appropriate action to fix the test incidents, re-test and close the test incidents or defer the incident(s) to a future release. 1) Status of test incidents is recorded in the test incident report.	Test cases [verified] Test procedures [verified] Traceability record [updated] Test execution documentation [published] Test incident report [published]	Test execution documentation [published] Test incident report [updated]
TL TES	SI.05.06 Update the traceability record.	Software [verified] Traceability record [updated]	Traceability record [updated]

B.4.6 Software product assembly activity (SI.06)

Table B.10 provides the task for SI.06.

Table B.10 — Testing task list for SI.06

Roles	Task list	Input work products	Output work products
TL TES CUS	SI.06.05.TT1 Perform validation: — Verify whether a system satisfies the acceptance criteria. — Validate whether the system is 'fit for use'. — Determine whether to accept the system. 1) Results are documented in the test execution documentation. 2) Test incidents are documented in the test incident report.	Test plan [verified] Test environment [established] Test cases [verified] Test procedures [verified] Traceability record [updated]	Test cases — Acceptance tests [completed] Test procedures — Acceptance tests [completed] Test execution documentation [published] Test incident report [published]

B.5 Testing work products

Table B.11 to B.17 provide an alphabetical description of the work products of this annex.

Table B.11 — Test plan

WP ID	Name	Source
WP.T01	Test plan	Project management

Description

The test plan provides a test planning and test management document. It is detailed description of test objectives to be achieved and the means and schedule for achieving them, organized to coordinate testing activities for some test item or set of test items.

It may include:

- Context of the testing
- Test item
- Test scope
- Test strategy – Describes the approach to testing.
- Test sub-processes
- Test deliverables.
- Test design techniques – Specifies which test design techniques are to be applied.
- Test completion criteria – Describes the conditions under which the relevant test organization considers test execution activities to be complete.
- Metrics to be collected – Describes the metrics for which values are to be collected during the test activities.
- Test data requirements – Specifies all relevant test data requirements for the project or test sub-process.
- Test environment requirements – Specifies the necessary and desired properties of the test environment.
- Testing activities and estimates.
- Staffing.
- Schedule.

The applicable states are: initiated, verified, updated.

Table B.12 — Test status report

WP ID	Name	Source
WP.T02	Test status report	Project management

Description

The test status report provides information about the status of the testing that is performed in a specific reporting period.

It may include:

- Test status - Provides information on the status of the testing for the reporting period.
- Reporting period - Specifies the time period, covered by the report.
- Progress against test plan.
- Factors blocking progress.
- Test measures - Presents the collated test measures related to the end of the reporting period.
- New and changed risks.
- Planned testing – Describes the planned testing for the next reporting period.

The applicable states are: initiated, published.

Table B.13 — Test cases

WP ID	Name	Source
WP.T03	Test cases	Software implementation

Description

The test case is a 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.

Test Case may include:

- Unique identifier.
- Objective – Identifies and briefly describes the special focus or objective of the test case.
- Priority – Defines the priority for the testing.
- Preconditions – Describes the required state of the test environment and any special constraints pertaining to the execution of the test case.
- Inputs – Specifies each action required to bring the test item into a state where the expected result can be compared to the actual results. The detail of the descriptions is tailored to fit the knowledge of the test executors. This can require provision of input data and/or events to the test item. The actions may be numbered within the test case, if needed.
- Expected results – Specifies the expected outputs and behaviour required of the test item in response to the inputs that are given to the test item when it is in its precondition state. Provides the expected value(s) (with tolerances where appropriate) for each required output.

The applicable states are: preliminary, updated, verified, implemented, completed and baselined.

Table B.14 — Test procedures

WP ID	Name	Source
WP.T04	Test procedures	Software implementation

Description

The test procedure specifies the order in which the test cases are executed according to dependencies described by pre-conditions and post-conditions and other testing requirements.

Test procedure may include:

- Unique identifier.
- Objective – Identifies and briefly describes the special focus or objective of the test procedure.
- Priority.
- Start-up – Describes the necessary actions to prepare for execution of the test cases specified in the test procedure.
- Test cases to be executed (traceability) – lists the test cases in the order in which they are to be executed.
- Relationship to other procedures.
- Stop and wrap up - Describes the actions necessary to bring execution to an orderly halt and the actions necessary after the execution of the procedure should be completed.

The applicable states are: initiated, updated, verified, completed and baselined.

Table B.15 — Test execution documentation

WP ID	Name	Source
WP.T05	Test execution documentation	Software implementation
Description Test execution documentation documents the result of the execution of a test case in a test procedure. It may include: — Test results - records of whether or not a specific test case execution should have passed or failed, they are usually recorded directly in the test procedure in a placeholder reserved for this purpose. — Test logs. The applicable state is: published.		

Table B.16 — Test incident report

WP ID	Name	Source
WP.T06	Test incident report	Software implementation
Description Test incident report documents an incident recognized during testing. Test incident report may include: — Timing information – The date (and possibly also the time) when the incident was first observed. — Originator - Specifies the name(s) and title(s) of the individual(s) who identified the incident. — Context - Identifies the context in which the incident was observed - configuration item, the test procedure and test case, information about the test environment and/or test data. — Description of the incident - Provides a detailed description of the incident. Indicates if the incident is reproducible, and, if so, provides enough information to reproduce it. — Originator's assessment of severity. — Originator's assessment of priority. — Risk - Provides information on the introduction of new risks or changes to the status of existing risks, where applicable. — Status of the incident - Identifies the current incident status, which can be: "open", "approved for resolution", "assigned for resolution", "fixed", "retested with the fix confirmed", and "closed". other possible status values can be "rejected" or "withdrawn". The applicable states are: published, updated.		

Table B.17 — Test environment

WP ID	Name	Source
WP.T07	Test environment	Software implementation
Description Test environment is the environment used to perform testing of software. Test environment requirements are specified in the test plan. It may include: — Facilities. — Hardware. — Software tools. — Simulators. — Other support elements needed to conduct tests. — Supporting documentation, e.g. installation, operation and maintenance documentation for test environment. The applicable states are: established, updated, cleaned up.		

B.6 Testing roles

Table B.18 provides the description of an additional role of this annex.

Table B.18 — Description of testing role

Role	Competency
Tester	Knowledge and experience eliciting, specifying and analysing the requirements. Knowledge in designing user interfaces and ergonomic criteria. Knowledge of interfacing techniques between modules, systems and components and integration testing techniques Knowledge of principles, techniques, infrastructure and tools to be used in the testing process Knowledge of the review techniques.