International Standard

ISO/IEC 29110-5-1-1

First edition
2025-02

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

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

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

COPYRIGHT PROTECTED DOCUMENT

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 cancels and replaces ISO/IEC TR 29110-5-1-1:2012, which has been technically revised.

The main changes are as follows:

— Many task statements have been reworded to facilitate their understanding. Some task statements are deleted to make this Entry profile light-weighted and suitable for streamlining to the Basic profile.

— Conditional tasks have been added to develop optional work products (e.g. change requests) that have been requested by a customer. This notation replaces the 'Optional' notation [e.g. *(optional) used in the first edition 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 ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15289.

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

A list of all parts in the ISO/IEC 29110 series can be found on the ISO website.

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 a 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.

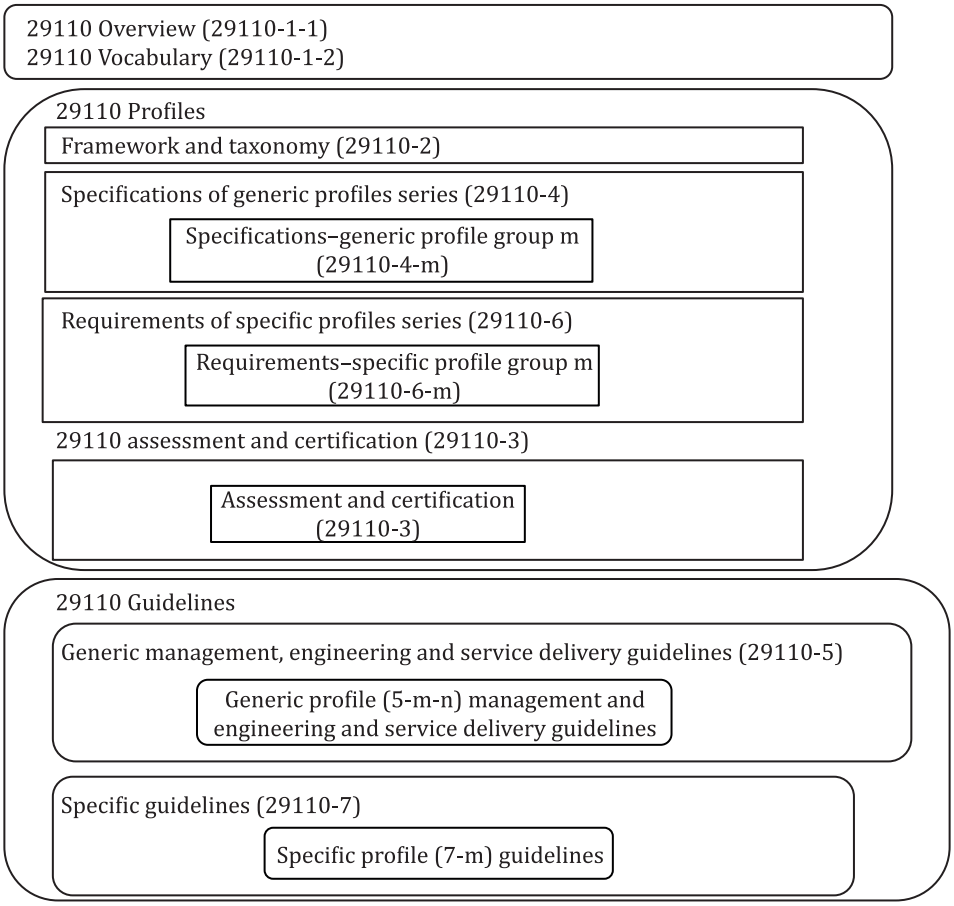

Figure 1 — The ISO/IEC 29110 series

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 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 first software profile of a four-profile software engineering roadmap (i.e. Entry, Basic, Intermediate and Advanced).

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.

For a start-up VSE that does not have customers yet, anybody who acts on behalf of customers can play the role of the customer.

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 Entry profile guidelines of the ISO/IEC 29110 series (i.e. ISO/IEC 29110-5-6-1).

In this document, Annex A describes the deployment packages for the software Entry 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 5-1-1:
Software engineering guidelines for the generic Entry profile

1 Scope

This document provides the management and engineering guidelines to the software Entry 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 applies to start-up VSEs (e.g. VSEs that started their operation less than three years ago) and/or VSEs working on small projects (e.g. projects with a size of less than six person-months).

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
agreement
mutual acknowledgment 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]

3.2
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]

3.3
defect
imperfection or deficiency in a *work product* (3.14) where that work product does not meet its *requirements* (3.8) or specifications and needs to be either repaired or replaced

[SOURCE: IEEE 1044:2009]

3.4
Entry profile
profile (3.5) targeted at start-up VSEs (i.e. VSEs who started their operation fewer than three years ago) and/or at VSEs working on a single small *project* (3.6) (e.g. project size of less than 6 person-months)

[SOURCE: ISO/IEC 29110-1-2:2024, 3.40]

3.5
profile
subset of appropriate standards' processes and their outcomes, activities and tasks combined to accomplish a particular function

Note 1 to entry: The base standards used to develop profiles for VSEs are ISO/IEC/IEEE 12207, ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 15289.

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

3.6
project
endeavour with defined start and finish dates undertaken to create a product or service in accordance with specified resources and *requirements* (3.8)

Note 1 to entry: A project is sometimes viewed as a unique process comprising coordinated and controlled activities 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]

3.7
report
information item that describes the results of activities such as investigations, observations, assessments, or tests

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

3.8
requirement
statement that translates or expresses a need and its associated constraints and conditions

Note 1 to entry: A constraint is an externally imposed limitation on the software, its design, or implementation or on the process 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.]

3.9
small and medium enterprise
SME
enterprise with less than 250 persons employed

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

3.10
software product
set of computer programs, procedures, and possibly associated documentation and data

[SOURCE: ISO/IEC/IEEE 12207:2017, 3.1.45, modified — Note 1 to entry has been removed.]

3.11
task
requirement (3.8), recommendation, or permissible action intended to contribute to the achievement of one or more outcomes of a process

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

3.12
validation
confirmation, through the provision of objective evidence, that the *requirements* (3.8) 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 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 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]

3.13
verification
confirmation, through the provision of objective evidence, that specified *requirements* (3.8) have been fulfilled

Note 1 to entry: Verification is a set of activities 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.]

3.14
work product
artefact produced by a process

EXAMPLE *Project* (3.6) plan, *requirements* (3.8) specification, design documentation, source code, test plan, test meeting minutes, schedules, budgets, and incident *reports* (3.7).

Note 1 to entry: A subset of the work products can be baselined to be used as the basis of further work, and some will 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 that 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.

All work products' names are initiated with capital letters. Some work products 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 abbreviations 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 shown as a two-column table. See Clause 8 for the alphabetical list of the roles and the description of the required competencies.

Diagram – Graphical representation of the processes. The large round-edged rectangles indicate processes 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 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 outcomes: 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 into 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 roles 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 followed with this text: "Conditional task".

Incorporation to project repository – list of work products to be saved in a project repository. 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 purpose only.

NOTE 3 The term 'Entry' is using a capital 'E' to indicate an ISO/IEC 29110 profile (e.g. the Entry profile) while the term 'entry' is used when referring to the act of entering a place or something written or printed in a dictionary, account book.

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

4.3 Abbreviated terms

CUS customer

PM project management

PJM project manager

SI software implementation

WT work team

5 Overview

5.1 General

This document provides project management and software implementation processes which integrate practices 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 approach or methodology, e.g. waterfall, iterative, incremental, evolutionary or agile.

5.2 Entry conditions to use the software Entry profile

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

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

— development cycle(s) has/have been documented (e.g. waterfall, iterative, incremental, evolutionary, agile).

5.3 Processes and activities of the software Entry profile

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

Both processes of the software Entry 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.

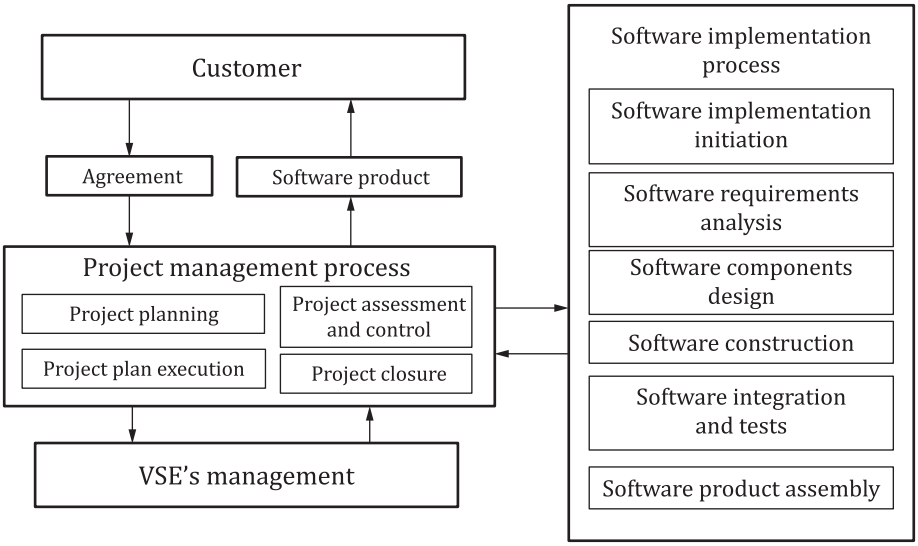

Figure 2 — Processes and activities of the software Entry profile

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 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 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 is initiated with a revision of the project plan. Project plan will guide the execution of the software requirements analysis, software components specification, software construction, software integration and test, and software product assembly activities.

To remove work product's defects, verification and validation by 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).

6 Project management (PM) process

6.1 PM process purpose

The purpose of the project management process is to establish and carry out in a systematic way the tasks of the software implementation project, which allows complying with the project's objectives in the expected quality, time, and costs.

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 approved 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. Closure of the project is performed to get the customer acceptance documented in the acceptance record.

PM.O3. The change requests are addressed, evaluated and tracked.

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

PM.O5. Items of software configuration are identified and controlled.

PM.O6. Software quality assurance is performed to provide assurance that delivered software products comply with requirements specifications and customer requirements.

NOTE Software quality assurance is implemented through the performance of the verification and validation by test tasks performed in software implementation processes.

6.3 PM roles involved

The roles involved in the PM process are:

— customer;

— project manager;

— 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.

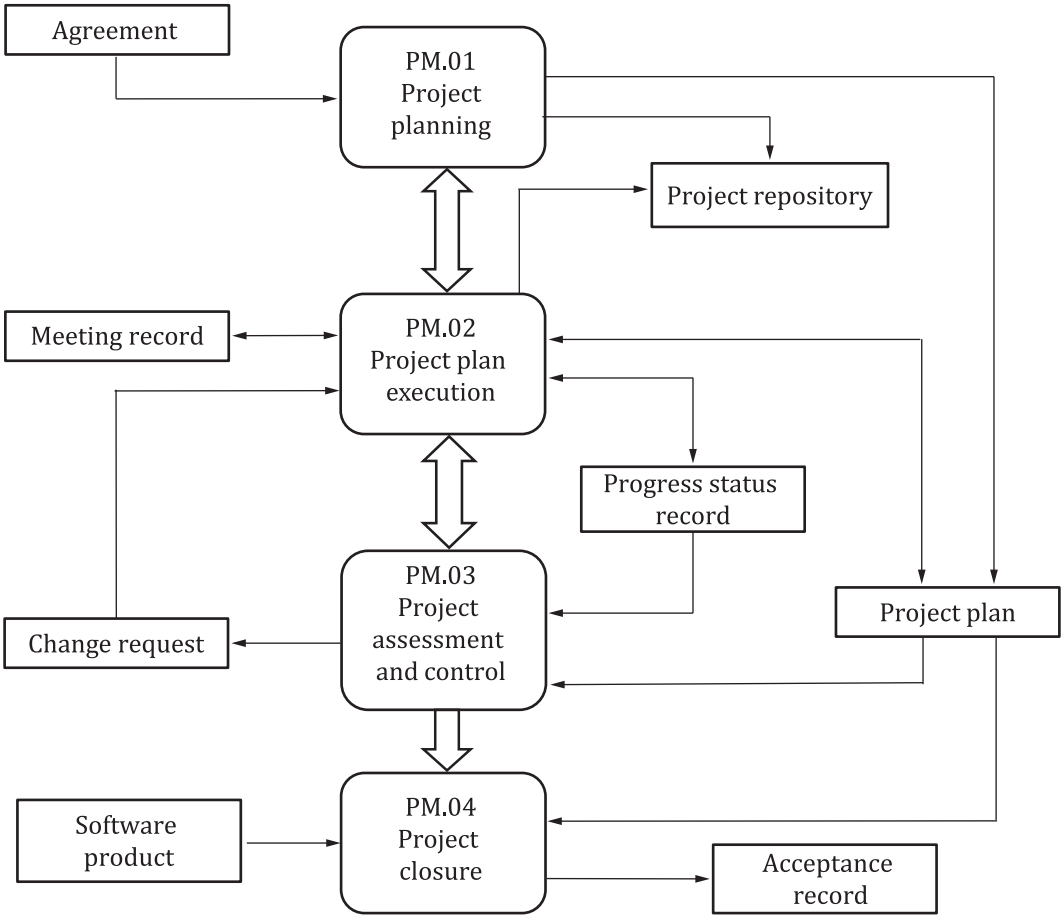

Figure 3 — Project management process

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 has 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)

The project planning 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;

— product quality assurance strategy through verification and validation of work products/deliverables by testing techniques;

— work team and customer roles and responsibilities;

— project resources and training needs;

— estimates of effort, cost, and schedule;

— project repository to store, handle, and deliver controlled work products 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 WT	PM.01.01 Review the agreement to select 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 document as listed in Clause 5. 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 WT	PM.01.02 Identify the tasks to produce the work products to be delivered to the customer. 1) Tasks of the SI processes, with verification and validation by testing 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 WT	PM.01.03 Establish the estimated duration to perform each task.	Project plan [initiated] — Tasks	Project plan [initiated] — Estimated duration
PJM WT	PM.01.04 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 has been approved and uploaded in the project repository.	Agreement [approved, baselined] Project plan [initiated]	Project plan [initiated] — Resources
PJM WT	PM.01.05 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 WT	PM.01.06 Create the schedule of the project tasks.	Project plan [initiated]	Project plan [initiated]

Table 1 *(continued)*

Roles	Task list	Input Work products	Output Work products
	1) Estimated start and completion dates to each task are assigned. 2) The assigned resources, sequence and dependency of the tasks are considered.	— Tasks — Estimated duration — Composition of work team	— Schedule of the project tasks
PJM	PM.01.07 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	PM.01.08 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 — Deliveryinstructions
PJM	PM.01.09 Review and obtain approval of the project plan.	Project plan [initiated]	Project plan [approved]
PJM WT	PM.01.10 Establish the project repository.	Project plan [approved]	Project repository [established]

6.4.2.3 PM.02 Project plan execution (contributes to PM.O2, PM.O3, PM.O4, PM.O6)

The project plan execution 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;

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

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 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 CUS WT	PM.02.02 Conduct review meetings with the customer: — Initiate a change request if needed. — Record agreements between customer and PJM and tracked them to closure. 1) A change request that affects the customer needs to be negotiated to reach the acceptance of the customer and PJM. 2) If a change request is initiated, update the project plan according to the new agreement between the customer and PJM.	Project plan [approved] Progress status record [published] Meeting record [published]	Meeting record [published] Change request [approved] Project plan [updated]

6.4.2.4 PM.03 Project assessment and control (contributes to PM.O2)

The project assessment and control activity evaluates the performance of the plan against documented commitments.

The activity provides:

— evaluation of actual plan performance and progress against targets;

— tracked change requests;

— 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 WT	PM.03.01 Evaluate project progress with respect to the project plan, comparing: — Actual tasks against planned tasks — Actual resource allocation against planned resources — Actual cost against budget estimates — Actual time against planned schedule	Project plan [approved] Progress status record [published]	Progress status record [evaluated]
PJM WT	PM.03.02 Evaluate and track change request. 1) A change request can be initiated externally by the customer or internally by the PJM or WT.	Change request [initiated]	Change request [evaluated]
PJM WT	PM.03.03 Establish actions to correct deviations or problems: — Document actions in the progress status record. — Track actions to closure. 1) Conditional task: Track actions to closure.	Progress status record [evaluated]	Progress status record [updated]

6.4.2.5 PM.04 Project closure (contributes to PM.O2)

The project closure 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, have been assembled as the software product in activity SI.06. 2) The delivery instructions, signed by the CUS and PJM, are used to conduct the acceptance of the work products by the CUS. 3) The completion of the project that requires "signature of the acceptance record" may not be applicable to a start-up and mass-market customers.	Project plan [approved] — Delivery instructions Software product [delivered]	Acceptance record [signed] Software product [accepted]
PJM	PM.04.02 Store the acceptance record in the project repository.	Acceptance record [signed]	Project repository [updated]

6.4.3 Incorporation to project repository

The work products to be saved in PM project repository are:

— project plan;

— change request;

— acceptance record;

— meeting record;

— progress status record.

7 Software implementation (SI) process

7.1 SI process purpose

The purpose of the software implementation process is the systematic performance of the requirements analysis, design components identification, construction, integration and 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, for example, agile, evolutionary, incremental, and 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 and communicated.

SI.O3. Software components and their interfaces are identified.

SI.O4. Software components are produced. Unit tests are performed to verify the consistency with software requirements.

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.

SI.O6. Verification and Validation by testing tasks with customer and work team are performed to ensure that the software products fulfil customer requirements.

SI.O7. A software product, that meets the requirements specification, as agreed to with the customer, is prepared for delivery.

7.3 SI roles involved

The roles involved in the SI process are:

— customer;

— project manager;

— 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.

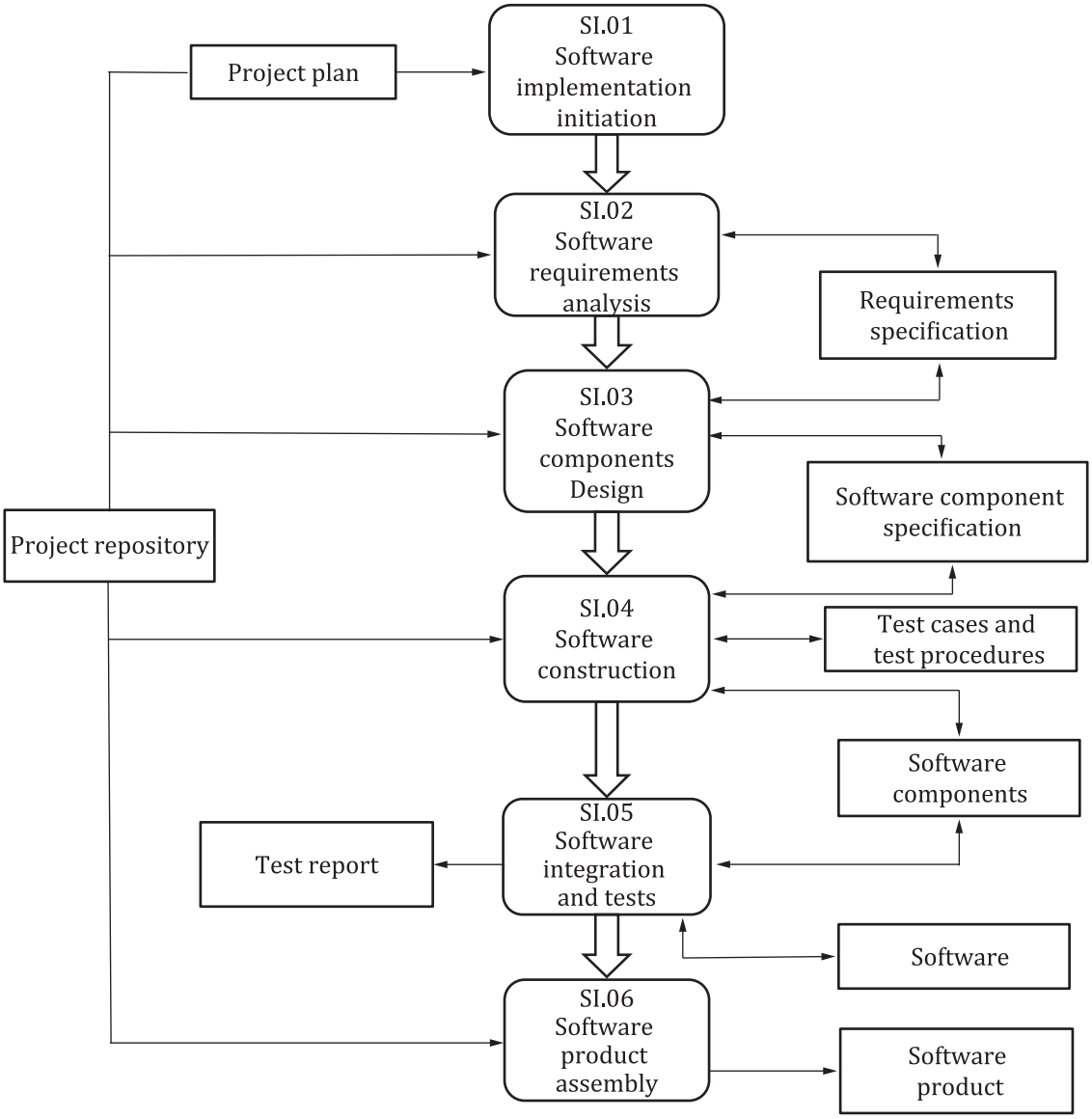

Figure 4 — Software implementation process

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 has the following activities:

— SI.01 Software implementation initiation

— SI.02 Software requirements analysis

— SI.03 Software components 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)

The software implementation initiation 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 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 and PJM to the project plan. 2) ISO/IEC 20246 defines different types of work product reviews.	Project plan [approved]	Project plan [approved]
WT	SI.01.02 Set up the implementation environment.	Project plan [approved]	Implementation environment [established]

7.4.2.3 SI.02 Software requirements analysis (contributes to SI.O2, SI.O6, SI.O7)

The software requirements analysis 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;

— validation to obtain agreement on the customer requirements.

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
PJM WT	SI.02.01 Assign tasks to the work team members. 1) Tasks are assigned to WT in accordance with their role, based on the approved project plan.	Project plan [approved] — Tasks	Project plan [approved] — Tasks
WT CUS	SI.02.02 Document the software requirements specification: — 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.	Agreement [baselined] Project plan [approved] — Product description	Requirements specification [initiated]
CUS PJM	SI.02.03 Validate and obtain approval from the customer of the software requirements specification: — Make corrections until the document is approved by the customer. 1) Software requirements specification must satisfy the needs and the agreed expectations, including the user interface usability. 2) Conditional task: initiate a change request if significant changes (those affecting project delivery schedule, project costs, etc.) were needed.	Requirements specification [initiated]	Requirements specification [validated, approved] Change request [initiated]
WT	SI.02.04 Add the software requirements specification to the project repository.	Requirements specification [approved]	Project repository [updated] — Requirements specification [baselined]

7.4.2.4 SI.03 Software component design (contributes to SI.03, SI.06, SI.07)

The software component design activity identifies and designs the software components which are required to deliver the software requirements.

The activity provides:

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

— work team review of the requirements specification;

— work team identifies and designs software components for the requirements specification.

The task list for SI.03 software component design is given in Table 7.

Table 7 — SI.03 task list

Roles	Task list	Input Work products	Output Work products
PJM WT	SI.03.01 Assign tasks to the members of the work team. 1) Tasks are related to their roles, according to the project plan.	Project plan [approved] — Tasks	Project plan [approved] — Tasks
WT	SI.03.02 Review the software requirements specification. 1) The WT 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 [approved]	Requirements specification [approved]
WT	SI.03.03 Document or update the software components specification: — Analyse the requirements specification to identify software components. — Describe the interfaces, based on the requirements specification. — Provide the detail of software components and their interfaces to allow their construction.	Requirements specification [approved]	Software components specification [completed]
WT	SI.03.04 Incorporate the software components specification to the project repository.	Software components specification [completed]	Project repository [updated] — Software components specification [baselined]

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

The software construction activity develops the code modules and data from the software components specification.

The activity provides:

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

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

— code modules and applied unit tests.

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
PJM WT	SI.04.01 Assign tasks to the work team members.	Project plan [approved] — Tasks	Project plan [approved] — Tasks
WT	SI.04.02 Develop the software components based on software components specification.	Software components specification [completed]	Software components [initiated]
WT	SI.04.03 Develop the unit test cases.	Software components specification [completed] Software components [initiated] Requirements specification [approved]	Test cases and test procedures — Unit test cases [initiated]
WT	SI.04.04 Execute the unit test cases: — Correct the defects detected by the unit tests.	Software components [initiated] Test cases and test procedures — Unit test cases [initiated]	Software components [tested]
WT	SI.04.05 Add the software components to the project repository.	Software components [tested]	Project repository [updated] — Software components [baselined]

7.4.2.6 SI.05 Software integration and tests (contributes to SI.O5, SI.O6, SI.O7)

The software integration and tests 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.

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
PJM WT	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
WT	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) WT 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 — Integration tests [initiated]	Test cases and test procedures — Integration tests [verified]
WT	SI.50.03 Integrate the software using the software components.	Software components [verified]	Software [implemented]
WT	SI.05.04 Perform software integration tests. 1) Results are documented in the test report.	Software [implemented] Test cases and test procedures — Integration tests [verified]	Software [tested] Test report [initiated]
WT	SI.05.05 Correct the defects detected: — Perform tests until defects are corrected.	Software [tested] Test report [initiated] Test cases and test procedures — Integration tests [verified]	Software [corrected, completed] Test report [published]
WT PJM CUS	SI.05.06 Validate the software with customer. 1) Conditional task for start-up VSEs. 2) Usually, a start-up needs to quickly build a first version of the software, i.e. minimum viable product (MVP), in order to get customer feedbacks. This is an important part of the start-up life cycle.	Software [completed]	Software [validated]

Table 9 *(continued)*

Roles	Task list	Input Work products	Output Work products
WT	SI.05.07 Incorporate the work products to the project repository. 1) Add test cases and test procedures, software, test report to the software product.	Test cases and test procedures [verified] Software [validated] Test report [published]	Project repository [updated] Software product — Test cases and test procedures [baselined] — Software [baselined] — Test report [baselined]

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

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

The activity provides:

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

— 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
PJM 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
PJM WT	SI.06.02 Prepare the delivery according to delivery instructions. 1) WT 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 has been approved and uploaded to the project repository.	Software product [baselined] Project plan [approved] — Delivery instructions	Software product [delivered]

7.4.3 Incorporation to the project repository

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

— requirements specification;

— software design components;

— test cases and test procedures;

— software components;

— software;

— test report.

8 Description of roles

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

For a start-up VSE that does not have customers, anybody who can represent the potential customer's assumed needs can play the role of the customer (e.g. project manager).

Table 11 — Description of roles

	Role	Competency
1	Customer	Knowledge of the customer processes and ability to explain the customer requirements. The customer (representative) has the authority to approve the requirements and their changes. The customer includes user representatives in order to ensure that the operational environment is addressed. Knowledge and experience in the application domain.
2	Project manager	Leadership capability with experience making decisions, planning, personnel management, delegation and supervision, finances and software development.
3	Work team	Knowledge and experience according to their roles on the project. 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 has provided 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.
Completed	A work product is considered complete.
Corrected	Defect(s) identified in a work product has(ve) been removed.
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.
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.
Reviewed	A work product was presented to the authorized customer or stakeholder(s) as applicable for comment.

Table 12 *(continued)*

Name of State	Definition of states
Signed	A work product is signed by the authorized customer or stakeholder(s) as applicable.
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 it will fulfil its requirements for its intended use and was approved by authorized stakeholder(s).

Tables 13 to 27 provide an alphabetical list of the input and output 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, tested, baselined) give 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 input.

Work products are identified with a unique code WP.XX, where XX is a sequential number. These codes are 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
- Identifies any open issues (if applicable)
- Signed by customer and project manager

The acceptance record includes the “acceptance criteria”.

The development team asks for clarification if needed.

The applicable states are: initiated and signed.

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 — Identifies requester contact information — 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 — Implementation environment

WP ID	Name	Source
WP.04	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 17 — Meeting record

WP ID	Name	Source
WP.05	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 18 — Progress status record

WP ID	Name	Source
WP.06	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
- Record of any deviations from planned tasks and reason why.
- Signature of the project manager

The applicable state is: published.

Table 19 — Project plan

WP ID	Name	Source
WP.07	Project plan	Project management

Description

Presents how the project processes and activities will be 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 – i.e. list of work products to be delivered to customer
- Tasks, including verification and validation by testing and review with customer, 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
- 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: accepted, updated and reviewed.

Table 20 — Project repository

WP ID	Name	Source
WP.08	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 state is: updated.

Table 21 — Requirements specification

WP ID	Name	Source
WP.09	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 the 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. coexistence, 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 testable or can be assessed.

The applicable states are: validated and baselined.

Table 22 — Software

WP ID	Name	Source
WP.10	Software	Software implementation
Description Software item (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 23 — Software components

WP ID	Name	Source
WP.11	Software components	Software implementation
Description A set of related code units. The applicable states are: unit tested, corrected and baselined.		

Table 24 — Software components specification

WP ID	Name	Source
WP.12	Software components specification	Software implementation

Description

Textual and graphical information on the software structure and its components.

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 (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: completed, baselined.

Table 25 — Software product

WP ID	Name	Source
WP.13	Software product	Software implementation

Description

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

- Requirements specification
- Software components specification
- Software components
- Software
- Test cases and test procedures
- Test report
- Product operation guide
- Software user documentation

The applicable states are: delivered and accepted.

Table 26 — Test cases and test procedures

WP ID	Name	Source
WP.14	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 state is: baselined.

Table 27 — Test report

WP ID	Name	Source
WP.15	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.		

10 Software tools

10.1 General

Tables 28 and 29 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 is given in Table 28.

Table 28 — Project management tools

Activity	Resource list
Project planning Project plan execution Project assessment and control Project closure	Tools allowing document, manage 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 is given in Table 29.

Table 29 — Software implementation tools

Activity	Resource list
Software implementation initiation Software requirements analysis Software component 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)

Deployment packages for the software Entry profile

In order to facilitate the implementation, by VSEs, of a profile, a set of deployment packages is available. A deployment package is a set of artifacts developed to facilitate the implementation of a set of practices, of the selected framework, in a VSE. But a deployment package is not a complete process reference model. Deployment packages are not intended to preclude or discourage the use of additional guidelines that VSEs find useful.

The elements of a typical deployment package are: technical description, relationships with the ISO/IEC 29110 series, key definitions, detailed description of processes, activities, tasks, roles and products, template, checklist, example, references and mapping to standards and models, and a list of tools. The mapping is only given as information to show that a Deployment Package has explicit links to this document, ISO standards, such as ISO/IEC/IEEE 12207, or models. Hence by deploying and implementing a package, a VSE can see its concrete step to achieve or demonstrate coverage to this document. Deployment packages are designed such that a VSE can implement its content, without having to implement the complete framework at the same time.

A deployment package includes the following items, in the order given below:

- Technical description
  - Purpose of this document
  - Why this topic is important?
- Definitions
- Relationships with ISO/IEC 29110
- Overview of processes, activities, tasks, roles and products
- Description of processes, activities, tasks, steps, roles and products
  - Role description
  - Product description
  - Artefact description
- Template(s)
- Example(s)
- Checklist(s)
- Tool(s)
- References to other standards and models (e.g. ISO 9001, ISO/IEC/IEEE 12207)
- References
- Evaluation form

For the Entry profile, a set of deployment packages is available at no cost on the Internet.

a) Project management

b) Software implementation

c) Self-assessment

Bibliography

[1] ISO 9000:2015, *Quality management systems — Fundamentals and vocabulary*

[2] ISO 9001*Quality management systems — Requirements*

[3] ISO/IEC/IEEE 12207:2017, *Systems and software engineering — Software life cycle processes*

[4] ISO/IEC/IEEE 15288, *Systems and software engineering — System life cycle processes*

[5] ISO/IEC/IEEE 15289:2019, *Systems and software engineering — Content of life-cycle information items (documentation)*

[6] ISO/IEC 20246:2017, *Software and systems engineering — Work product reviews*

[7] ISO/IEC 25010, *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Product quality model*

[8] ISO/IEC 29110-1-1, *Systems and software engineering — Lifecycle profiles for very small entities (VSEs) — Part 1-1: Overview*

[9] ISO/IEC 29110-1-2:2024, *Systems and software engineering — Lifecycle profiles for Very Small Entities (VSEs) — Part 1-2: Vocabulary*

[10] ISO/IEC 29110-4-1, *Systems and software engineering — Lifecycle profiles for Very Small Entities (VSEs) — Part 4-1: Software engineering - Profile specifications: Generic profile group*

[11] ISO/IEC 29110-5-6-1[1)], *Systems and software engineering — Life cycle profiles for very small entities (VSEs) — Part 5-6-1: Systems engineering — Management and engineering guide: Generic profile group: Entry profile*

[12] IEEE 1044:2009, *IEEE Standard Classification for Software Anomalies*

[13] World Bank. https://www.worldbank.org/en/topic/smefinance

[14] OECD SME and Entrepreneurship Outlook. 2019 Edition, Organisation for Economic Co-operation and Development, Paris, 2019. Available at: https://www.oecd-ilibrary.org/industry-and-services/oecd-sme-and-entrepreneurship-outlook-2019_34907e9c-en

[15] Di Bella, L., Katsinis, A., Lagüera-González, J., Odenthal, L., Hell, M., Lozar, B. Annual Report on European SMEs 2022/2023, Publications Office of the European Union, Luxemburg, 2023

[16] Laporte, C.Y., Munoz, M., Mejia Miranda, J., O'Connor, R.V., Applying Software Engineering Standards in Very Small Entities-From Startups to Grownups, IEEE Software, Vol. 35, Issue 1, pp 99-103, 2018

[17] Laporte, C.Y., Verret, G., Munoz, M., A Software Project That Partially Failed, Computer, vol. 56, Issue 5, May 2023

1) Under development.

ICS 35.080
Price based on 36 pages

iso.org