

---

---

**Software engineering — Lifecycle profiles  
for Very Small Entities (VSEs) —**

Part 5-1-2:

**Management and engineering guide:  
Generic profile group: Basic profile**

*Ingénierie du logiciel — Profils de cycle de vie pour très petits  
organismes (TPO) —*

*Partie 5-1-2: Guide d'ingénierie et de gestion: Groupe de profil  
générique: Profil basique*



**COPYRIGHT PROTECTED DOCUMENT**

© ISO/IEC 2011

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
Case postale 56 • CH-1211 Geneva 20  
Tel. + 41 22 749 01 11  
Fax + 41 22 749 09 47  
E-mail [copyright@iso.org](mailto:copyright@iso.org)  
Web [www.iso.org](http://www.iso.org)

Published in Switzerland

## Contents

Page

Foreword .....	v
Introduction.....	vi
1 Scope .....	1
1.1 Fields of application .....	1
1.2 Target audience .....	1
2 Normative references .....	1
3 Terms and definitions .....	2
4 Conventions and abbreviated terms .....	2
4.1 Naming, diagramming and definition conventions.....	2
4.2 Abbreviated terms .....	3
5 Overview.....	3
6 Project Management (PM) process.....	4
6.1 PM purpose .....	4
6.2 PM objectives.....	5
6.3 PM input products .....	7
6.4 PM output products.....	8
6.5 PM internal products.....	8
6.6 PM roles involved .....	8
6.7 PM diagram .....	9
6.7.1 PM activities .....	10
6.7.2 PM incorporation to <i>Project Repository</i> .....	15
7 Software Implementation (SI) process .....	16
7.1 SI purpose .....	16
7.2 SI objectives.....	16
7.3 SI input products .....	19
7.4 SI output products.....	20
7.5 SI internal products.....	20
7.6 SI roles involved .....	21
7.7 SI diagram .....	21
7.7.1 SI activities .....	23
7.7.2 SI incorporation to the <i>Project Repository</i> .....	31
8 Roles.....	32
9 Product description.....	33
10 Software tools requirements.....	40
10.1 Project Management process .....	40
10.2 Software Implementation process.....	41
Annex A (informative) Deployment Packages.....	42
Bibliography.....	44

### Table of illustrations

Figure 1 — ISO/IEC 29110 series.....	vii
Figure 2 — Basic profile guide processes (diagram notation is explained in 4.1) .....	4

Figure 3 — Project Management process diagram. ....	9
Figure 4 — Software Implementation process diagram. ....	22

**Table of tables**

Table 1 — ISO/IEC 29110 target audience .....	vi
Table 2 — PM input products .....	7
Table 3 — PM output products .....	8
Table 4 — PM internal products .....	8
Table 5 — PM roles involved .....	8
Table 6 — PM.1 task list .....	10
Table 7 — PM.2 task list .....	13
Table 8 — PM.3 task list .....	14
Table 9 — PM.4 task list .....	15
Table 10 — PM repository products .....	15
Table 11 — SI input products .....	19
Table 12 — SI output products .....	20
Table 13 — SI internal products .....	20
Table 14 — SI roles involved .....	21
Table 15 — SI.1 task list .....	23
Table 16 — SI.2 task list .....	24
Table 17 — SI.3 task list .....	26
Table 18 — SI.4 task list .....	27
Table 19 — SI.5 task list .....	28
Table 20 — SI.6 task list .....	30
Table 21 — SI repository products .....	31
Table 22 — Roles .....	32
Table 23 — Product Descriptions .....	33
Table 24 — Project Management tools .....	40
Table 25 — Software Implementation tools .....	41
Table A.1 — Table of Content of a Deployment Package .....	42

## Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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

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

In exceptional circumstances, when the joint technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example), it may decide to publish a Technical Report. A Technical Report is entirely informative in nature and shall be subject to review every five years in the same manner as an International Standard.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC TR 29110-5-1-2 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*.

ISO/IEC 29110 consists of the following parts, under the general title *Software engineering — Lifecycle profiles for Very Small Entities (VSEs)*:

- *Part 1: Overview* [Technical Report]
- *Part 2: Framework and taxonomy*
- *Part 3: Assessment guide* [Technical Report]
- *Part 4-1: Profile specifications: Generic profile group*
- *Part 5-1-2: Management and engineering guide: Generic profile group: Basic profile* [Technical Report]

Parts 4 and 5 can be developed to accommodate new profile specifications and management and engineering guides as follows:

- *Part 4-m: Profile specifications: Profile group aaaaa*
- *Part 5-m-n: Management and engineering guide: Profile group aaaaa: Profile bbbbbb* [Technical Report]

## Introduction

The software industry recognizes the value of Very Small Entities (VSEs) in contributing valuable products and services. For the purpose of ISO/IEC 29110, a VSE is an entity (enterprise, organization, department or project) having up to 25 people. VSEs also develop and/or maintain software that is used in larger systems; therefore, recognition of VSEs as suppliers of high quality software is often required.

According to the Organisation for Economic Co-operation and Development (OECD) SME and Entrepreneurship Outlook report (2005) "SMEs constitute the dominant form of business organisation in all countries world-wide, accounting for over 95 % and up to 99 % of the business population depending on country". 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. Conformance with these standards is difficult, if not impossible, giving VSEs no way, or very limited ways, to be recognized as entities that produce quality software in their domain. Therefore, VSEs are often cut off from some economic activities.

It has been found that VSEs find it difficult to relate International Standards to their business needs and to justify their application to their business practices. Most VSEs can neither afford the resources, in terms of number of employees, budget and time, nor do they see a net benefit in establishing software lifecycle processes. To rectify some of these difficulties, a set of guides has been developed according to a set of VSE characteristics. The guides are based on subsets of appropriate standards elements, referred to as VSE profiles. The purpose of a VSE profile is to define a subset of International Standards relevant to the VSE context, for example, processes and outcomes of ISO/IEC 12207 and products of ISO/IEC 15289.

ISO/IEC 29110, targeted by audience, has been developed to improve product and/or service quality, and process performance. See Table 1. ISO/IEC 29110 is not intended to preclude the use of different lifecycles such as: waterfall, iterative, incremental, evolutionary or agile.

**Table 1 — ISO/IEC 29110 target audience**

ISO/IEC 29110	Title	Target audience
Part 1	Overview	VSEs, assessors, standards producers, tool vendors and methodology vendors.
Part 2	Framework and taxonomy	Standards producers, tool vendors and methodology vendors. Not intended for VSEs.
Part 3	Assessment guide	Assessors and VSEs
Part 4	Profile specifications	Standards producers, tool vendors and methodology vendors. Not intended for VSEs.
Part 5	Management and engineering guide	VSEs

If a new profile is needed, ISO/IEC 29110-4 and ISO/IEC TR 29110-5 can be developed without impacting existing documents and they become ISO/IEC 29110-4-m and ISO/IEC 29110-5-m-n, respectively, through the ISO/IEC process.

ISO/IEC TR 29110-1 defines the business terms common to the VSE Profile Set of Documents. It introduces processes, lifecycle and standardization concepts, and the ISO/IEC 29110 series. It also introduces the characteristics and requirements of a VSE, and clarifies the rationale for VSE-specific profiles, documents, standards and guides.

ISO/IEC 29110-2 introduces the concepts for software engineering standardized profile for VSEs, and defines the terms common to the VSE Profile Set of Documents. It establishes the logic behind the definition and application of standardized profiles. It specifies the elements common to all standardized profiles (structure, conformance, assessment) and introduces the taxonomy (catalogue) of ISO/IEC 29110 profiles.

ISO/IEC TR 29110-3 defines the process assessment guidelines and compliance requirements needed to meet the purpose of the defined VSE Profiles. ISO/IEC TR 29110-3 also contains information that can be useful to developers of assessment methods and assessment tools. ISO/IEC TR 29110-3 is addressed to people who have direct relation with the assessment process, e.g. the assessor and the sponsor of the assessment, who need guidance on ensuring that the requirements for performing an assessment have been met.

ISO/IEC 29110-4-1 provides the specification for all the profiles of the Generic Profile Group. The Generic Profile Group is applicable to VSEs that do not develop critical software products. The profiles are based on subsets of appropriate standards elements. VSEs' Profiles apply and are targeted to authors/providers of guides and authors/providers of tools and other support material.

This part of ISO/IEC 29110 provides an implementation management and engineering guide for the Basic Profile of the Generic Profile Group specified in ISO/IEC 29110-4-1. The Basic Profile describes software development of a single application by a single project team with no special risk or situational factors.

Figure 1 describes the ISO/IEC 29110 series and positions the parts within the framework of reference. Overviews and guides are published as Technical Reports (TR), and profiles are published as International Standards (IS).

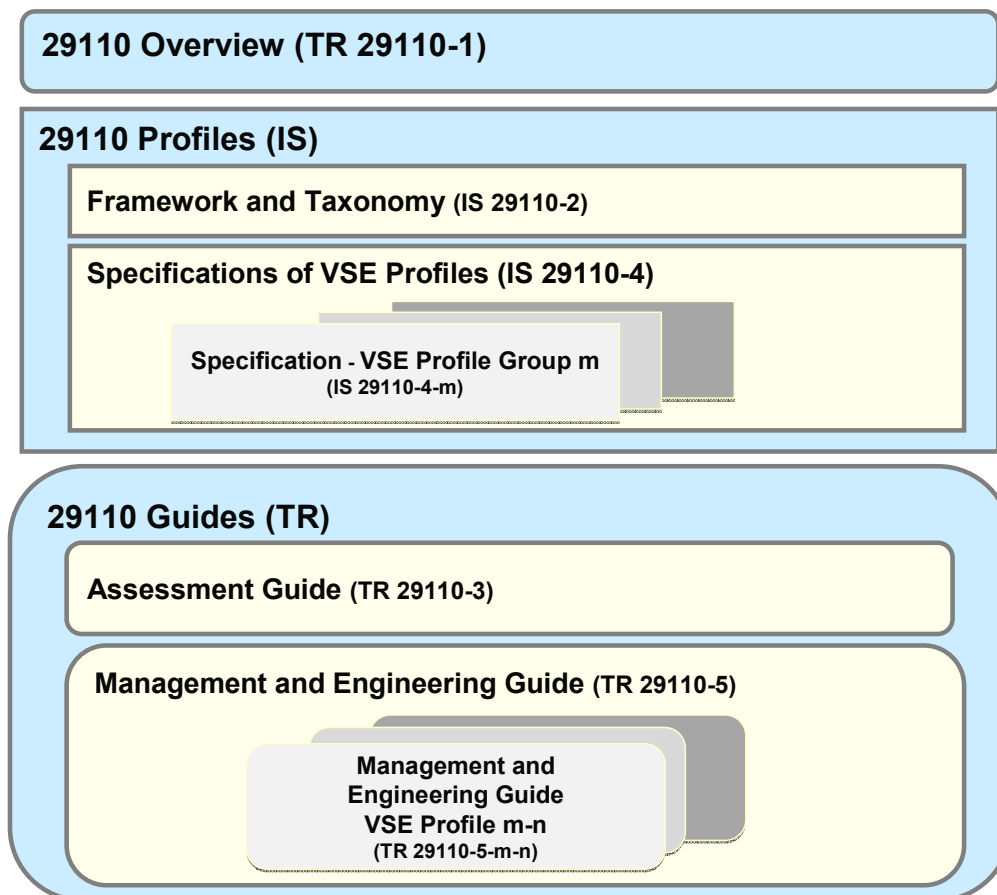


Figure 1 — ISO/IEC 29110 series





# Software engineering — Lifecycle profiles for Very Small Entities (VSEs) —

## Part 5-1-2:

### Management and engineering guide: Generic profile group: Basic profile

## 1 Scope

### 1.1 Fields of application

This part of ISO/IEC 29110 is applicable to Very Small Entities (VSEs). VSEs are enterprises, organizations, departments or projects of up to 25 people. The lifecycle processes described in ISO/IEC 29110 are not intended to preclude or discourage their use by organizations bigger than VSEs.

This part of ISO/IEC 29110 provides the management and engineering guide to the Basic VSE Profile specified in ISO/IEC 29110-4-1 through project management and software implementation processes. This part of ISO/IEC 29110 is a standalone guide; it is not intended for a VSE to use the standardized profile to implement this part of ISO/IEC 29110.

This part of ISO/IEC 29110 applies for software development projects, which can be to fulfil an external or internal contract. The internal contract need not be explicit between the project team and their Customer.

Using this part of ISO/IEC 29110, a VSE can obtain the following benefits.

- An agreed set of project requirements and expected 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 products is followed.

### 1.2 Target audience

This part of ISO/IEC 29110 is targeted at VSEs.

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

## 2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC TR 29110-1, *Software engineering — Lifecycle profiles for Very Small Entities (VSEs) — Part 1: Overview*

### 3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC TR 29110-1 apply.

### 4 Conventions and abbreviated terms

#### 4.1 Naming, diagramming and definition conventions

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

**Name** – process identifier, followed by its abbreviation in brackets “( )”.

**Purpose** – general goals and results expected of the effective implementation of the process. The implementation of the process should provide tangible benefits to the stakeholders. The purpose is identified by the abbreviation of the process name.

**Objectives** – specific goals to ensure the accomplishment of the process purpose. The objectives are identified by the abbreviation of the process name, followed by the letter “O” and a consecutive number, for example PM.O1, SI.O2, etc. Each objective is followed by the square box which includes a list of the chosen processes for the basic profile from ISO/IEC 12207:2008 and its outcomes related to the objective.

**Input Products** – products required to perform the process and its corresponding source, which can be another process or an external entity to the project, such as the Customer. Identified by the abbreviation of the process name and showed as two column table of product names and sources.

**Output Products** – products generated by the process and its corresponding destination, which can be another process or an external entity to the project, such as Customer or Organizational Management. Identified by the abbreviation of the process name and showed as two column table of product names and destinations.

**Internal Products** – products generated and consumed by the process. Identified by the abbreviation of the process name and showed as one column table of the product names.

All products' names are printed in cursive and initiate with capital letters. Some products have one or more statuses attached to the product name surrounded by square brackets “[ ]” and separated by “,”. The product status may change during the process execution. See Clause 9 for the alphabetical list of the products, its descriptions, possible statuses and the source of the 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, its abbreviations and required 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 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 products. The notation used in the diagrams does not imply the use of any specific process lifecycle.

**Activity** – a set of cohesive tasks. Task is a requirement, recommendation, or permissible action, intended to contribute to the achievement of one or more objectives of a process. 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 objectives surrounded by brackets “( )”. For example PM.1 Project Planning (PM.O1, PM.O5, PM.O6, PM.O7) means that the activity PM.1 Project Planning contributes to the achievement of the listed objectives: PM.O1, PM.O5, PM.O6 and PM.O7. The activity description begins with the task summary and is followed by the task descriptions table. 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.

Tasks description table 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 PM1.1, PM1.2, and so on.
- Input Products - products needed to execute the task.
- Output Products - products created or modified by the execution of the task.

**Incorporation to *Project Repository*** – list of products to be saved in *Project Repository*; the *Version Control Strategy* has to be applied to some of them (see Clause 6.7.2 and 7.7.2). It is useful as a checklist for project manager and technical leader.

NOTE: Tables used in process description are for presentation purpose only.

## 4.2 Abbreviated terms

VSE      Very Small Entity

## 5 Overview

The Basic VSE Profile Management and Engineering Guide applies to a Very Small Entity (VSE) (enterprise, organisation, department or project up to 25 people) dedicated to software development. The project may fulfil an external or internal contract. The internal contract between the project team and its Customer need not be explicit.

The Guide provides Project Management and Software Implementation processes which integrate practices based on the selection of ISO/IEC 12207:2008, *Systems and software engineering — Software life cycle processes* and ISO/IEC 15289:2006, *Systems and software engineering — Content of systems and software life cycle process information products (Documentation)* standards elements. Annex A provides information about Deployment Packages which will facilitate the implementation of these processes.

This part of ISO/IEC 29110 is intended to be used by the VSE to establish processes to implement any development approach or methodology including, e.g., agile, evolutionary, incremental, test driven development, etc. based on the VSE organization or project needs.

Using the Guide, VSE can obtain benefits in the following aspects:

- An agreed set of project requirements and expected 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 products is followed.

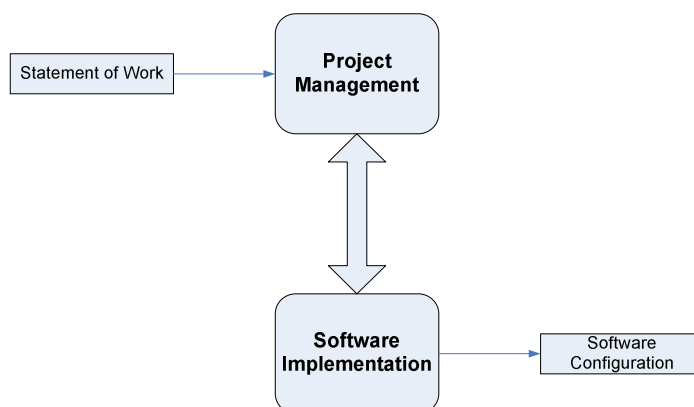
To use the Guide the VSE needs to fulfil the following entry conditions:

- project *Statement of Work* is documented;
- feasibility of the project was performed before its start;
- project team, including project manager, is assigned and trained; and
- goods, services and infrastructure to start the project are available.

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 cost.

The purpose of the Software Implementation process is the systematic performance of the analysis, design, construction, integration and tests activities for new or modified software products according to the specified requirements.

Both processes are interrelated (see Figure 2).



NOTE Diagram notation is explained in 4.1.

**Figure 2 — Basic profile guide processes**

PM process uses the Customer's *Statement of Work* to elaborate the *Project Plan*. 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 Configuration*, produced by SI, and gets the Customer's acceptance to formalize 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*. SI process starts with an initiation activity of the *Project Plan* revision. *Project Plan* will guide the execution of the software requirements analysis, software architectural and detailed design, software construction, software integration and test, and product delivery activities.

To remove product's defects verification, validation and test *Tasks* are included in the activities workflow.

The Customer provides a *Statement of Work* as an input to Project Management process and receives a *Software Configuration* as a result of Software Implementation process execution (see Figure 2).

## 6 Project Management (PM) process

### 6.1 PM 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 part of ISO/IEC 29110 is intended to be used by the VSE to establish processes to implement any development approach or methodology including, e.g., agile, evolutionary, incremental, test driven development, etc. based on the VSE organization or project needs.

## 6.2 PM objectives

PM.O1. The *Project Plan* for the execution of the project is developed according to the *Statement of Work* and reviewed and accepted by the Customer. The *Tasks* and *Resources* necessary to complete the work are sized and estimated.

### 6.3.1 Project Planning Process

- a) *the scope of the work for the project is defined;*
- c) *the tasks and resources necessary to complete the work are sized and estimated;*
- e) *plans for the execution of the project are developed; and*
- f) *plans for the execution of the project are activated.*

### 6.3.7 Measurement Process

- a) *the information needs of technical and management processes are identified.*

[ISO/IEC 12207:2008, 6.3.1, 6.3.7]

PM.O2. Progress of the project is monitored against the *Project Plan* and recorded in the *Progress Status Record*. 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*.

### 6.3.2 Project Assessment and Control Process

- a) *progress of the project is monitored and reported;*
- c) *actions to correct deviations from the plan and to prevent recurrence of problems identified in the project, are taken when project targets are not achieved; and*
- d) *project objectives are achieved and recorded.*

### 6.3.7 Measurement Process

- d) *the required data are collected, stored, analyzed, and the results interpreted; and*
- e) *information products are used to support decisions and provide an objective basis for communication.*

### 6.4.8 Software Acceptance Support Process

- a) *the product is completed and delivered to the acquirer;*

*7.2.8 Software Problem Resolution Process*

- b) problems are recorded, identified and classified; and*
- e) problems are tracked to closure.*

*[ISO/IEC 12207:2008, 6.3.2, 6.3.7, 6.4.8, 7.2.8]*

PM.O3. The *Change Requests* are addressed through their reception and analysis. Changes to software requirements are evaluated for cost, schedule and technical impact.

*7.1.2 Software Requirements Analysis Process*

- g) changes to the software requirements are evaluated for cost, schedule and technical impact.*

*[ISO/IEC 12207:2008, 7.1.2]*

PM.O4. Review meetings with the Work Team and the Customer are held. Agreements are registered and tracked.

*7.2.6 Software Review Process*

- a) management and technical reviews are held based on the needs of the project;*
- c) review results are made known to all affected parties;*
- d) action items resulting from reviews are tracked to closure.*

*[ISO/IEC 12207:2008, 7.2.6]*

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

*6.3.4 Risk Management Process*

- c) risks are identified as they develop and during the conduct of the project;*

*7.2.6 Software Review Process*

- e) risks and problems are identified and recorded.*

*[ISO/IEC 12207:2008, 6.3.4, 7.2.6]*

PM.O6. A software *Version Control Strategy* is developed. Items of *Software Configuration* 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.

**7.2.2 Software Configuration Management Process**

- a) a software configuration management strategy is developed;
- b) items generated by the process or project are identified, defined and baselined;
- c) modifications and releases of the items are controlled;
- d) modifications and releases are made available to affected parties;
- g) the storage, handling and delivery of the items are controlled.

[ISO/IEC 12207:2008, 7.2.2]

PM.O7. Software Quality Assurance is performed to provide assurance that work products and processes comply with the *Project Plan* and *Requirements Specification*.

NOTE: The implementation of the Software Quality Assurance process is through the performance of the verifications, validations and review *Tasks* performed in Project Management and Software Implementation processes.

**7.2.3 Software Quality Assurance Process**

- a) a strategy for conducting quality assurance is developed;
- b) evidence of Software quality assurance is produced and maintained;
- c) problems and/or non-conformance with requirements are identified and recorded; and
- d) adherence of products, processes and activities to the applicable standards, procedures and requirements are verified.

[ISO/IEC 12207:2008, 7.2.3]

**6.3 PM input products****Table 2 — PM input products**

Name	Source
<i>Statement of Work</i>	Customer
<i>Software Configuration</i>	Software Implementation
<i>Change Request</i>	Customer Software Implementation

## 6.4 PM output products

**Table 3 — PM output products**

<b>Name</b>	<b>Destination</b>
<i>Project Plan</i>	Software Implementation
<i>Acceptance Record</i>	Organizational Management
<i>Project Repository</i>	Software Implementation
<i>Meeting Record</i>	Customer
<i>Software Configuration</i>	Customer

## 6.5 PM internal products

**Table 4 — PM internal products**

<b>Name</b>
<i>Change Request</i>
<i>Correction Register</i>
<i>Meeting Record</i>
<i>Verification Results</i>
<i>Progress Status Record</i>
<i>Project Repository Backup</i>

## 6.6 PM roles involved

**Table 5 — PM roles involved**

<b>Role</b>	<b>Abbreviation</b>
Customer	CUS
Project Manager	PM
Technical Leader	TL
Work Team	WT



## 6.7 PM diagram

The following diagram shows the flow of information between the Project Management Process activities including the most relevant work products and their relationship.

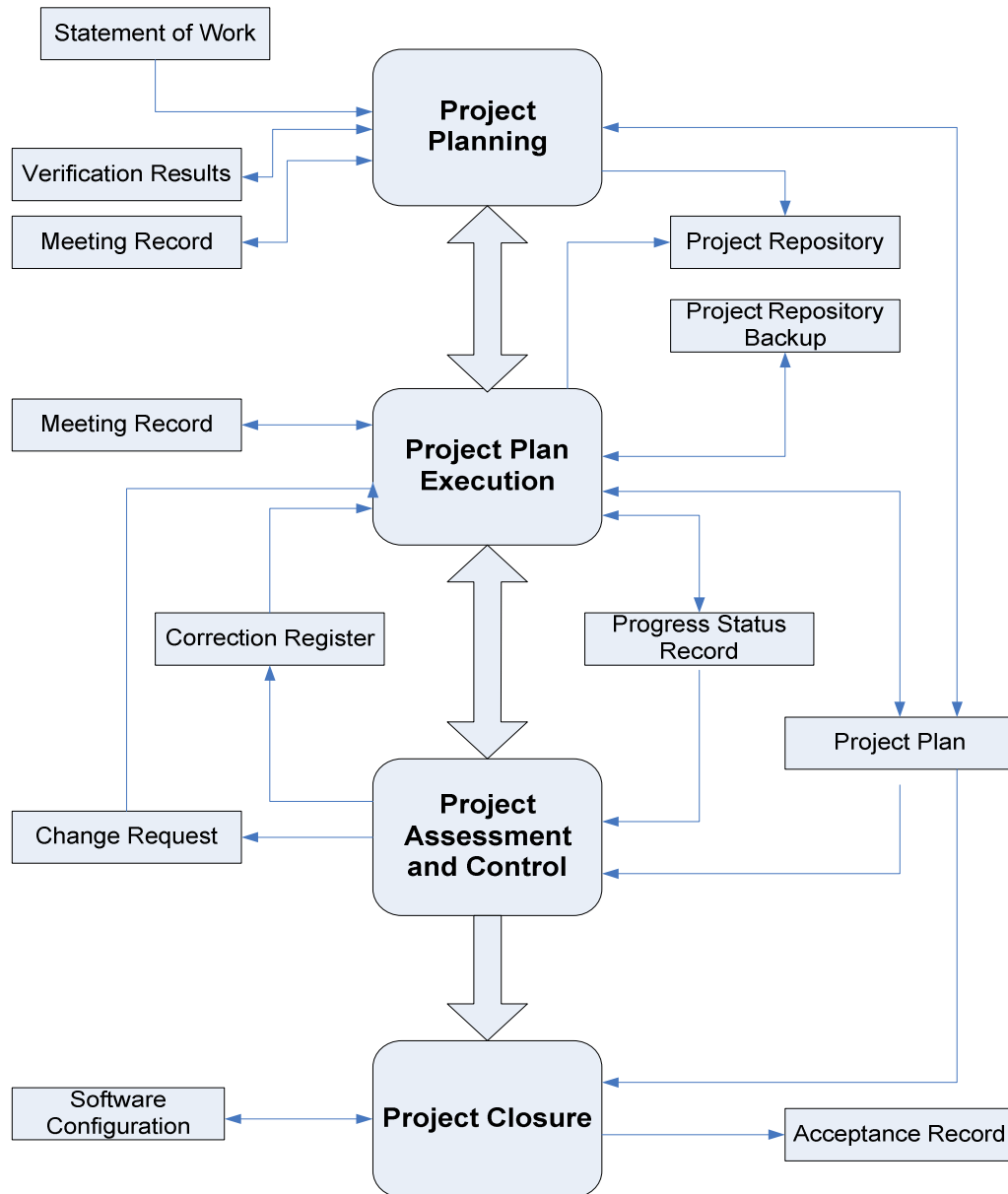


Figure 3 — Project Management process diagram

### 6.7.1 PM activities

The Project Management Process has the following activities:

- PM.1 Project Planning
- PM.2 Project Plan Execution
- PM.3 Project Assessment and Control
- PM.4 Project Closure

#### PM.1 Project Planning, (PM.O1, PM.O5, PM.O6, PM.O7)

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

- Reviewed *Statement of Work* and the *Tasks* needed to provide the contract *Deliverables* and to satisfy Customer requirements.
- Project lifecycle, 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 and baseline strategy.
- *Project Repository* to store, handle and deliver controlled product and document versions and baselines.

**Table 6 — PM.1 task list**

Role	Task List	Input Products	Output Products
PM TL	PM.1.1 Review the <i>Statement of Work</i>	<i>Statement of Work</i>	<i>Statement of Work</i> [reviewed]
PM CUS	PM.1.2 Define with the Customer the <i>Delivery Instructions</i> of each one of the <i>Deliverables</i> specified in the <i>Statement of Work</i> .	<i>Statement of Work</i> [reviewed]	<i>Project Plan</i> - <i>Delivery Instructions</i>
PM TL	PM.1.3 Identify the specific <i>Tasks</i> to be performed in order to produce the <i>Deliverables</i> and their <i>Software Components</i> identified in the <i>Statement of Work</i> . Include <i>Tasks</i> in the SI process along with verification, validation and reviews with Customer and Work Team <i>Tasks</i> to assure the quality of work products. Identify the <i>Tasks</i> to perform the <i>Delivery Instructions</i> . Document the <i>Tasks</i> .	<i>Statement of Work</i> [reviewed]	<i>Project Plan</i> - <i>Tasks</i>

Role	Task List	Input Products	Output Products
PM TL	PM.1.4 Establish the <i>Estimated Duration</i> to perform each task.	<i>Project Plan</i> - <i>Tasks</i>	<i>Project Plan</i> - <i>Estimated Duration</i>
PM TL	PM.1.5 Identify and document the <i>Resources</i> : human, material, equipment and tools, standards, including the required training of the Work Team to perform the project. Include in the schedule the dates when <i>Resources</i> and training will be needed.	<i>Statement of Work</i>	<i>Project Plan</i> - <i>Resources</i>
PM TL	PM.1.6 Establish the <i>Composition of Work Team</i> assigning roles and responsibilities according to the <i>Resources</i> .	<i>Project Plan</i> - <i>Resources</i>	<i>Project Plan</i> - <i>Composition of Work Team</i>
PM TL	PM.1.7 Assign estimated start and completion dates to each one of the <i>Tasks</i> in order to create the <i>Schedule of the Project Tasks</i> taking into account the assigned <i>Resources</i> , sequence and dependency of the <i>Tasks</i> .	<i>Project Plan</i> - <i>Tasks</i> - <i>Estimated Duration</i> - <i>Composition of Work Team</i>	<i>Project Plan</i> - <i>Schedule of the Project Tasks</i>
PM	PM.1.8 Calculate and document the project <i>Estimated Effort and Cost</i> .	<i>Project Plan</i> - <i>Schedule of the Project Tasks</i> - <i>Resources</i>	<i>Project Plan</i> - <i>Estimated Effort and Cost</i>
PM TL	PM.1.9 Identify and document the risks which may affect the project.	<i>All elements previously defined</i>	<i>Project Plan</i> - <i>Identification of Project Risks</i>
PM TL	PM.1.10 Document the <i>Version Control Strategy</i> in the <i>Project Plan</i> .		<i>Project Plan</i> - <i>Version Control Strategy</i>
PM	PM.1.11 Generate the <i>Project Plan</i> integrating the elements previously identified and documented.	<i>All elements previously defined</i>	<i>Project Plan</i> - <i>Tasks</i> - <i>Estimated Duration</i> - <i>Resources</i> - <i>Composition of Work Team</i> - <i>Schedule of the Project Task</i> - <i>Estimated Effort and Cost</i> - <i>Identification of Project Risks</i> - <i>Version Control Strategy</i> - <i>Delivery Instructions</i>

Role	Task List	Input Products	Output Products
PM TL	PM.1.12 Include <i>Product Description</i> , <i>Scope</i> , <i>Objectives</i> and <i>Deliverables</i> in the <i>Project Plan</i> .	<i>Statement of Work</i> - <i>Product Description</i> - <i>Scope</i> - <i>Objectives</i> - <i>Deliverables</i>	<i>Project Plan</i> - <i>Product Description</i> - <i>Scope</i> - <i>Objectives</i> - <i>Deliverables</i>
PM TL	PM.1.13 Verify and obtain approval of the <i>Project Plan</i> .  Verify that all <i>Project Plan</i> elements are viable and consistent. The results found are documented in a <i>Verification Results</i> and corrections are made until the document is approved by PM.	<i>Project Plan</i>	<i>Verification Results</i> <i>Project Plan [verified]</i>
PM CUS	PM.1.14 Review and accept the <i>Project Plan</i> .  Customer reviews and accepts the <i>Project Plan</i> , making sure that the <i>Project Plan</i> elements match with the <i>Statement of Work</i> .	<i>Project Plan [verified]</i>	<i>Meeting Record</i> <i>Project Plan [accepted]</i>
PM TL	PM.1.15 Establish the <i>Project Repository</i> using the <i>Version Control Strategy</i> .	<i>Version Control Strategy</i>	<i>Project Repository</i>

## PM.2 Project Plan Execution (PM.O2, PM.O3, PM.O4, PM.O5, PM.O7)

The Project Plan Execution activity implements the documented plan on the project. The activity provides:

- *Progress Status Record* of the project updated.
- Analyzed 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.

Table 7 — PM.2 task list

Role	Task List	Input Products	Output Products
PM TL WT	PM.2.1 Monitor the <i>Project Plan</i> execution and record actual data in <i>Progress Status Record</i> .	<i>Project Plan</i>	<i>Progress Status Record</i>
PM TL	PM 2.2 Analyse and evaluate the <i>Change Request</i> for cost, schedule and technical impact.  The <i>Change Request</i> can be initiated externally by the Customer or internally by the Work Team. Update the <i>Project Plan</i> , if the accepted change does not affect agreements with Customer.  <i>Change Request</i> , which affects those agreements, needs to be negotiated by both parties (see PM.2.4).	<i>Change Request [initiated]</i>  <i>Project Plan</i>	<i>Change Request [evaluated]</i>  <i>Project Plan [updated]</i>
PM TL WT	PM.2.3 Conduct revision meetings with the Work Team, identify problems, review risk status, record agreements and track them to closure.	<i>Project Plan</i> <i>Progress Status Record</i>  <i>Correction Register</i> <i>Meeting Record</i>	<i>Meeting Record [updated]</i>
PM CUS TL WT	PM.2.4 Conduct revision meetings with the Customer, record agreements and track them to closure.  <i>Change Request</i> initiated by Customer or initiated by Work Team, which affects the Customer, needs to be negotiated to reach acceptance of both parties.  If necessary, update the <i>Project Plan</i> according to new agreement with Customer.	<i>Project Plan</i> <i>Progress Status Record</i>  <i>Change Request [evaluated]</i>  <i>Meeting Record</i>	<i>Meeting Record [updated]</i>  <i>Change Request [accepted]</i> <i>Project Plan [updated]</i>
PM	PM.2.5 Perform backup according to the <i>Version Control Strategy</i> .	<i>Version Control Strategy</i>	<i>Project Repository Backup</i>
PM	PM.2.6 Perform <i>Project Repository</i> recovery using the <i>Project Repository Backup</i> , if necessary.	<i>Project Repository Backup</i>	<i>Project Repository [recovered]</i>

**PM.3 Project Assessment and Control (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.
- 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.

**Table 8 — PM.3 task list**

Role	Task List	Input Products	Output Products
PM TL WT	PM.3.1 Evaluate project progress with respect to the <i>Project Plan</i> , comparing: <ul style="list-style-type: none"> <li>- actual <i>Tasks</i> against planned <i>Tasks</i></li> <li>- actual results against established project <i>Objectives</i></li> <li>- actual resource allocation against planned <i>Resources</i></li> <li>- actual cost against budget estimates</li> <li>- actual time against planned schedule</li> <li>- actual risk against previously identified</li> </ul>	<i>Project Plan</i> <i>Progress Status Record</i>	<i>Progress Status Record [evaluated]</i>
PM TL WT	PM.3.2 Establish actions to correct deviations or problems and identified risks concerning the accomplishment of the plan, as needed, document them in <i>Correction Register</i> and track them to closure.	<i>Progress Status Record [evaluated]</i>	<i>Correction Register</i>
PM TL WT	PM.3.3 Identify changes to requirements and/or <i>Project Plan</i> to address major deviations, potential risks or problems concerning the accomplishment of the plan, document them in <i>Change Request</i> and track them to closure.	<i>Progress Status Record [evaluated]</i>	<i>Change Request [initiated]</i>

**PM.4 Project Closure (PM.O2)**

The Project Closure activity provides the project's documentation and products in accordance with contract requirements. The activity provides:

- Delivery of the product as specified in the *Delivery Instructions*.
- Support of Customer product acceptance in accordance to *Delivery Instructions*.
- Completion of the project and sign of the *Acceptance Record*.

Table 9 — PM.4 task list

Role	Task List	Input	Output
PM CUS	PM.4.1. Formalize the completion of the project according to the <i>Delivery Instructions</i> established in the <i>Project Plan</i> , providing acceptance support and getting the <i>Acceptance Record</i> signed.	<i>Project Plan</i> - <i>Delivery Instructions</i>  <i>Software Configuration [delivered]</i>	<i>Acceptance Record</i>  <i>Software Configuration [accepted]</i>
PM	PM.4.2 Update <i>Project Repository</i> .	<i>Software Configuration [accepted]</i>  <i>Project Repository</i>	<i>Project Repository [updated]</i>

### 6.7.2 PM incorporation to *Project Repository*

The list of products to be saved in *Project Repository*. After the incorporation, *Version Control Strategy* has to be applied to: *Project Plan*.

Table 10 — PM repository products

Product
<i>Project Plan</i>
<i>Change Request</i>
<i>Acceptance Record</i>
<i>Meeting Record</i>
<i>Correction Register</i>
<i>Progress Status Record</i>
<i>Verification Results</i>

## 7 Software Implementation (SI) process

### 7.1 SI purpose

The purpose of the Software Implementation process is the systematic performance of the analysis, design, construction, integration and tests activities for new or modified software products according to the specified requirements.

This part of ISO/IEC 29110 is intended to be used by the VSE to establish processes to implement any development approach or methodology including, e.g., agile, evolutionary, incremental, test driven development, etc. based on the VSE organization or project needs.

### 7.2 SI objectives

SI.O1. *Tasks of the activities are performed through the accomplishment of the current Project Plan.*

SI.O2. Software requirements are defined, analyzed for correctness and testability, approved by the Customer, baselined and communicated.

#### 6.4.1 Stakeholder Requirements Definition Process

a) *the required characteristics and context of use of services are specified.*

#### 7.1.2 Software Requirements Analysis Process

a) *the requirements allocated to the software elements of the system and their interfaces are defined;*

b) *software requirements are analyzed for correctness and testability;*

f) *the software requirements are approved and updated as needed; and*

h) *the software requirements are baselined and communicated to all affected parties.*

*[ISO/IEC 12207:2008, 6.4.1, 7.1.2]*

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

#### 7.1.3 Software Architectural Design Process

a) *a software architectural design is developed and baselined that describes the software items that will implement the software requirements;*

b) *internal and external interfaces of each software item are defined; and*

c) *consistency and traceability are established between software requirements and software design.*



**7.1.4 Software Detailed Design Process**

- a) *a detailed design of each software component, describing the software units to be built, is developed;*
- b) *external interfaces of each software unit are defined; and*
- c) *consistency and traceability are established between the detailed design and the requirements and architectural design.*

**7.1.5 Software Construction Process**

- a) *verification criteria are defined for all software units against their requirements;*

*[ISO/IEC 12207:2008, 7.1.3, 7.1.4, 7.1.5]*

NOTE: 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 test are defined and performed to verify the consistency with requirements and the design. Traceability to the requirements and design are established.

**7.1.5 Software Construction Process**

- b) *software units defined by the design are produced;*
- c) *consistency and traceability are established between software units and requirements and design; and*
- d) *verification of the software units against the requirements and the design is accomplished.*

*[ISO/IEC 12207:2008, 7.1.5]*

SI.O5. *Software* is produced performing integration of *Software Components* and verified using *Test Cases and Test Procedures*. Results are recorded at the *Test Report*. Defects are corrected and consistency and traceability to *Software Design* are established.

**7.1.6 Software Integration Process**

- c) *software items are verified using the defined criteria;*
- d) *software items defined by the integration strategy are produced;*
- e) *results of integration testing are recorded;*
- f) *consistency and traceability are established between software design and software items;*

#### 7.1.7 Software Qualification Testing Process

- a) *criteria for the integrated software is developed that demonstrates compliance with the software requirements;*
- b) *integrated software is verified using the defined criteria; and*
- c) *test results are recorded.*

[ISO/IEC 12207:2008, 7.1.6, 7.1.7]

SI.O6. A *Software Configuration*, 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 Configuration* are detected and related change requests are initiated.

#### 6.1.2 Supply Process

- d) *a product and/or service that meets the agreed requirements are developed by the supplier;*
- e) *the product and/or service is delivered to the acquirer in accordance with the agreed requirements; and*
- f) *the product is installed in accordance with the agreed requirements.*

#### 7.2.1 Software Documentation Management Process

- a) *a strategy identifying the documentation to be produced during the life cycle of the software product or service is developed;*
- c) *documentation to be produced by the process or project is identified; and*
- e) *documentation is developed and made available in accordance with identified standards.*

[ISO/IEC 12207:2008, 6.1.2, 7.2.1]

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

#### 7.2.4 Software Verification Process

- a) *a verification strategy is developed and implemented;*
- b) *criteria for verification of all required software work products is identified;*
- c) *required verification activities are performed;*
- d) *defects are identified and recorded; and*
- e) *results of the verification activities are made available to the customer and other involved parties.*

**7.2.5 Software Validation Process**

- a) *a validation strategy is developed and implemented;*
- b) *criteria for validation of all required work products are identified;*
- c) *required validation activities are performed;*
- d) *problems are identified and recorded;*
- f) *results of the validation activities are made available to the customer and other involved parties.*

*[ISO/IEC 12207:2008, 7.2.4, 7.2.5]*

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

**7.3 SI input products****Table 11 — SI input products**

<b>Name</b>	<b>Source</b>
<i>Project Plan</i>	Project Management
<i>Project Repository</i>	Project Management

## 7.4 SI output products

**Table 12 — SI output products**

<b>Name</b>	<b>Destination</b>
<i>Software Configuration</i> <ul style="list-style-type: none"> <li>• <i>Requirements Specification</i></li> <li>• <i>Software Design</i></li> <li>• <i>Traceability Record</i></li> <li>• <i>Software Components</i></li> <li>• <i>Software</i></li> <li>• <i>Test Cases and Test Procedures</i></li> <li>• <i>Test Report</i></li> <li>• <i>Product Operation Guide</i></li> <li>• <i>Software User Documentation</i></li> <li>• <i>Maintenance Documentation</i></li> </ul>	Project Management
<i>Change Request</i>	Project Management

## 7.5 SI internal products

**Table 13 — SI internal products**

<b>Name</b>
<i>Validation Results</i>
<i>Verification Results</i>

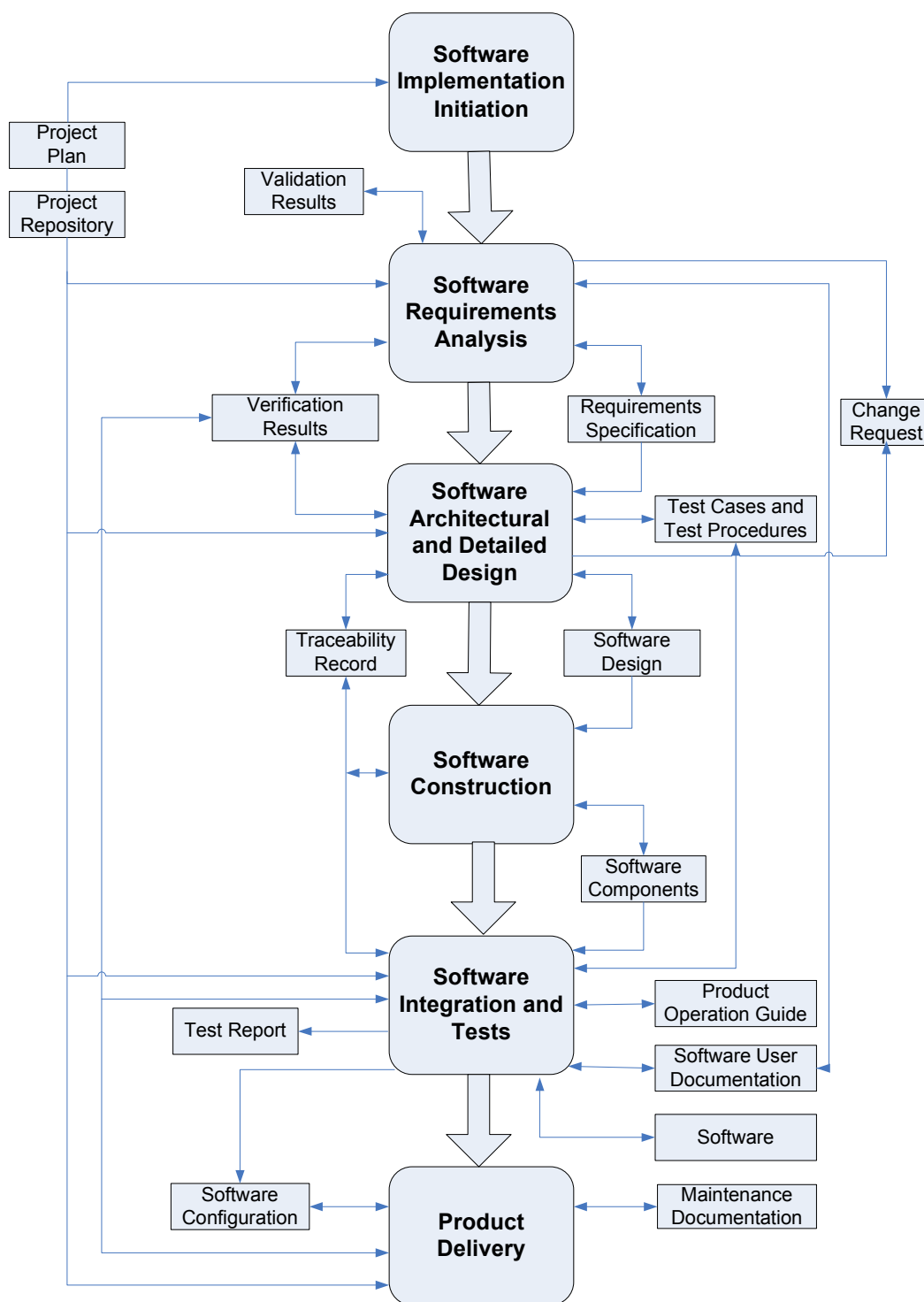
## 7.6 SI roles involved

**Table 14 — SI roles involved**

<b>Role</b>	<b>Abbreviation</b>
Customer	CUS
Analyst	AN
Designer	DES
Programmer	PR
Project Manager	PM
Technical Leader	TL
Work Team	WT

## 7.7 SI diagram

The following diagram shows the flow of information between the Software Implementation Process activities including the most relevant work products and their relationship.



**Figure 4 — Software Implementation process diagram.**

### 7.7.1 SI activities

The Software Implementation Process has the following activities:

- SI.1 Software Implementation Initiation
- SI.2 Software Requirements Analysis
- SI.3 Software Architectural and Detailed Design
- SI.4 Software Construction
- SI.5 Software Integration and Tests
- SI.6 Product Delivery

#### SI.1 Software Implementation Initiation (SI.O1)

The Software Implementation Initiation activity ensures that the *Project Plan* established in Project Planning activity is committed to by the Work Team. The activity provides:

- Review of the *Project Plan* by the Work Team to determine task assignment.
- Commitment to *Project Plan* by the Work Team and Project Manager.
- An implementation environment established.

**Table 15 — SI.1 task list**

Role	Task List	Input Products	Output Products
PM TL WT	SI.1.1 Revision of the current <i>Project Plan</i> with the Work Team members in order to achieve a common understanding and get their engagement with the project.	<i>Project Plan</i>	<i>Project Plan</i> [reviewed]
TL WT	SI.1.2 Set or update the implementation environment.	<i>Project Plan</i> [reviewed]	

#### SI.2 Software Requirements Analysis (SI.O2, SI.O6, SI.O7)

The Software Requirements Analysis activity analyzes 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 products.

Table 16 — SI.2 task list

Role	Task List	Input Products	Output Products
TL WT	SI.2.1 Assign <i>Tasks</i> to the Work Team members in accordance with their role, based on the current <i>Project Plan</i> .	<i>Project Plan [reviewed]</i> - <i>Tasks</i>	
AN CUS	SI.2.2 Document or update the <i>Requirements Specification</i> .  Identify and consult information sources (Customer, users, previous systems, documents, etc.) in order to get new requirements.  Analyze the identified requirements to determinate the <i>Scope</i> and feasibility.  Generate or update the <i>Requirements Specification</i> .	<i>Project Plan</i> - <i>Product Description</i>	<i>Requirements Specification</i>
AN TL	SI.2.3 Verify and obtain approval of the <i>Requirements Specification</i> .  Verify the correctness and testability of the <i>Requirements Specification</i> and its consistency with the <i>Product Description</i> . Additionally, review that requirements are complete, unambiguous and not contradictory. The results found are documented in a <i>Verification Results</i> and corrections are made until the document is approved by AN. If significant changes were needed, initiate a <i>Change Request</i> .	<i>Requirements Specification</i>  <i>Project Plan</i> - <i>Product Description</i>	<i>Verification Results</i>  <i>Requirements Specification [verified]</i>  <i>Change Request [initiated]</i>
CUS AN	SI.2.4 Validate and obtain approval of the <i>Requirements Specification</i>  Validate that <i>Requirements Specification</i> satisfies needs and agreed upon expectations, including the user interface usability. The results found are documented in a <i>Validation Results</i> and corrections are made until the document is approved by the CUS.	<i>Requirements Specification [verified]</i>	<i>Validation Results</i>  <i>Requirements Specification [validated]</i>
AN	SI.2.5 Document the preliminary version of the <i>*Software User Documentation</i> or update the present manual, if appropriate.  <i>*(Optional)</i>	<i>Requirements Specification [validated]</i>	<i>*Software User Documentation [preliminary]</i>



Role	Task List	Input Products	Output Products
AN TL	<p>SI.2.6 Verify and obtain approval of the *<i>Software User Documentation</i>, if appropriate.</p> <p>Verify consistency of the *<i>Software User Documentation</i> with the <i>Requirements Specification</i>. The results found are documented in a <i>Verification Results</i> and corrections are made until the document is approved by AN. If significant changes were needed, initiate a <i>Change Request</i>.</p> <p>*(Optional)</p>	<p>*<i>Software User Documentation</i> [preliminary]</p> <p><i>Requirements Specification</i></p>	<p><i>Verification Results</i></p> <p>*<i>Software User Documentation</i> [preliminary, verified]</p> <p><i>Change Request</i> [initiated]</p>
TL	<p>SI.2.7 Incorporate the <i>Requirements Specification</i>, and *<i>Software User Documentation</i> to the <i>Software Configuration</i> in the baseline.</p> <p>*(Optional)</p>	<p><i>Requirements Specification</i> [validated]</p> <p>*<i>Software User Documentation</i> [preliminary, verified]</p>	<p><i>Software Configuration</i></p> <ul style="list-style-type: none"> <li>- <i>Requirements Specification</i> [validated, baselined]</li> <li>- *<i>Software User Documentation</i> [preliminary, verified, baselined]</li> </ul>

### SI.3 Software Architectural and Detailed Design (SI.O3, SI.O6, SI.O7)

The Software Architectural and Detailed Design activity transforms the software requirements to the system software architecture and software detailed design. 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 products and documents under version control.

NOTE Software Architecture and Detailed Design can be performed separately according to the project schedule.

Table 17 — SI.3 task list

Role	Task List	Input Products	Output Products
TL AN DES	SI.3.1 Assign <i>Tasks</i> to the Work Team members related to their role according to the current <i>Project Plan</i> .	<i>Project Plan</i> - <i>Tasks</i>	
AN DES	SI.3.2 Understand <i>Requirements Specification</i> .	<i>Requirements Specification</i> [validated, baselined]	
AN DES	SI.3.3 Document or update the <i>Software Design</i> .  Analyze the <i>Requirements Specification</i> to generate the architectural design, its arrangement in subsystems and <i>Software Components</i> defining the internal and external interfaces. Describe in detail, the appearance and the behaviour of the interface, based on the <i>Requirements Specification</i> in a way that <i>Resources</i> for its implementation can be foreseen.  Provide the detail of <i>Software Components</i> and their interfaces to allow the construction in an evident way.  Generate or update the <i>Traceability Record</i> .	<i>Requirements Specification</i> [validated, baselined]	<i>Software Design</i>  <i>Traceability Record</i>
AN DES	SI.3.4 Verify and obtain approval of the <i>Software Design</i> .  Verify correctness of <i>Software Design</i> documentation, its feasibility and consistency with their <i>Requirement Specification</i> . Verify that the <i>Traceability Record</i> contains the adequate relationships between requirements and the <i>Software Design</i> elements. The results found are documented in a <i>Verification Results</i> and corrections are made until the document is approved by DES. If significant changes were needed, initiate a <i>Change Request</i> .	<i>Software Design</i>  <i>Traceability Record</i>  <i>Requirements Specification</i> [validated, baselined]	<i>Verification Results</i>  <i>Software Design</i> [verified]  <i>Traceability Record</i> [verified]  <i>Change Request</i> [initiated].
DES	SI.3.5 Establish or update <i>Test Cases and Test Procedures</i> for integration testing based on <i>Requirements Specification</i> and <i>Software Design</i> .  Customer provides testing data, if needed.	<i>Requirements Specification</i> [validated, baselined]  <i>Software Design</i> [verified, baselined]	<i>Test Cases and Test Procedures</i>

Role	Task List	Input Products	Output Products
DES AN	<p>SI.3.6 Verify and obtain approval of the <i>Test Cases and Test Procedures</i>.</p> <p>Verify consistency among <i>Requirements Specification</i>, <i>Software Design</i> and <i>Test Cases and Test Procedures</i>. The results found are documented in a <i>Verification Results</i> and corrections are made until the document is approved by AN.</p>	<p><i>Test Cases and Test Procedures</i></p> <p><i>Requirements Specification</i> [validated, baselined]</p> <p><i>Software Design</i> [verified, baselined]</p>	<p><i>Verification Results</i></p> <p><i>Test Cases and Test Procedures</i> [verified]</p>
DES	<p>SI.3.7 Update the <i>Traceability Record</i> incorporating the <i>Test Cases and Test Procedures</i>.</p>	<p><i>Test Cases and Test Procedures</i> [verified]</p> <p><i>Traceability Record</i> [updated]</p>	<p><i>Traceability Record</i> [updated]</p>
TL	<p>SI.3.8 Incorporate the <i>Software Design</i>, and <i>Traceability Record</i> to the <i>Software Configuration</i> as part of the baseline.</p> <p>Incorporate the <i>Test Cases</i>, and <i>Test Procedures</i> to the <i>Project Repository</i>.</p>	<p><i>Software Design</i> [verified]</p> <p><i>Test Cases and Test Procedures</i> [verified]</p> <p><i>Traceability Record</i> [verified]</p>	<p><i>Software Configuration</i></p> <ul style="list-style-type: none"> <li>- <i>Software Design</i> [verified, baselined]</li> <li>- <i>Test Cases and Test Procedures</i> [verified]</li> <li>- <i>Traceability Record</i> [verified, baselined]</li> </ul>

#### SI.4 Software Construction (SI.O4, SI.O6, SI.O7)

The Software Construction 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*,

**Table 18 — SI.4 task list**

Role	Task List	Input Products	Output Products
TL PR	<p>SI.4.1 Assign <i>Tasks</i> to the Work Team members related to their role, according to the current <i>Project Plan</i>.</p>	<p><i>Project Plan</i></p> <ul style="list-style-type: none"> <li>- <i>Tasks</i></li> </ul>	
PR	<p>SI.4.2 Understand <i>Software Design</i>.</p>	<p><i>Software Design</i> [verified, baselined]</p>	
PR	<p>SI.4.3 Construct or update <i>Software Components</i> based on the detailed part of the <i>Software Design</i>.</p>	<p><i>Software Design</i> [verified, baselined],</p> <p><i>Traceability Record</i> [verified, baselined]</p>	<p><i>Software Components</i></p>

Role	Task List	Input Products	Output Products
PR	SI.4.4 Design or update unit test cases and apply them to verify that the <i>Software Components</i> implements the detailed part of the <i>Software Design</i> .	<i>Software Components</i>	<i>Software Components [unit tested]</i>
PR	SI.4.5 Correct the defects found until successful unit test (reaching exit criteria) is achieved.	<i>Software Components [unit tested]</i>	<i>Software Components [corrected]</i>
PR	SI.4.6 Update the <i>Traceability Record</i> incorporating <i>Software Components</i> constructed or modified.	<i>Software Components [corrected]</i>  <i>Traceability Record [verified, baselined]</i>	<i>Traceability Record [updated]</i>
TL	SI.4.7 Incorporate <i>Software Components</i> and <i>Traceability Record</i> to the <i>Software Configuration</i> as part of the baseline.	<i>Software Components [corrected]</i>  <i>Traceability Record [updated]</i>	<i>Software Configuration</i> - <i>Software Components [corrected, baselined]</i> - <i>Traceability Record [updated baselined]</i>

### SI.5 Software Integration and Tests (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.
- Traceability of requirements and design to the integrated software product.
- Documented and verified operational and software user documentations.
- Verified *Software* baseline.

**Table 19 — SI.5 task list**

Role	Task List	Input Products	Output Products
TL PR	SI.5.1 Assign <i>Tasks</i> to the work team members related to their role, according to the current <i>Project Plan</i> .	<i>Project Plan</i> - <i>Tasks</i>	
PR	SI.5.2 Understand <i>Test Cases and Test Procedures</i> .  Set or update the testing environment.	<i>Test Cases and Test Procedures [verified]</i>	

Role	Task List	Input Products	Output Products
PR	SI.5.3 Integrates the <i>Software</i> using <i>Software Components</i> and updates <i>Test Cases</i> and <i>Test Procedures</i> for integration testing, as needed.	<i>Software Components</i> [corrected, baselined]  <i>Test Cases and Test Procedures</i> [verified]  <i>Traceability Record</i> [updated, baselined]	<i>Software</i>  <i>Test Cases and Test Procedures</i>
PR CUS	SI.5.4 Perform <i>Software</i> tests using <i>Test Cases</i> and <i>Test Procedures</i> for integration and document results in <i>Test Report</i> .	<i>Software</i>  <i>Test Cases and Test Procedures</i>	<i>Software</i> [tested]  <i>Test Report</i>
PR	SI.5.5 Correct the defects found and perform regression test until exit criteria is achieved.	<i>Software</i> [tested]  <i>Test Report</i>  <i>Test Cases and Test Procedures</i>  <i>Traceability Record</i> [updated, baselined]	<i>Software</i> [corrected]  <i>Test Report</i> [defects eliminated]
PR	SI.5.6 Updates the <i>Traceability Record</i> , if appropriate.	<i>Software</i> [corrected]  <i>Traceability Record</i> [updated, baselined]	<i>Traceability Record</i> [updated]
PR	SI.5.7 Document the <i>*Product Operation Guide</i> or update the current guide, if appropriate.  *(Optional)	<i>Software</i> [tested]	<i>*Product Operation Guide</i>
PR DES	SI.5.8 Verify and obtain approval of the <i>*Product Operation Guide</i> , if appropriate (see SI.5.7)  Verify consistency of the <i>Product Operation Guide</i> with the <i>Software</i> . The results found are documented in a <i>Verification Results</i> and corrections are made until the document is approved by DES.  *(Optional)	<i>*Product Operation Guide</i>  <i>Software</i> [tested]	<i>Verification Results</i>  <i>*Product Operation Guide</i> [verified]
AN	SI.5.9 Document the <i>*Software User Documentation</i> or update the current one, if appropriate.  *(Optional)	<i>Software</i> [tested]  <i>*Software User Documentation</i> [preliminary]	<i>*Software User Documentation</i>

Role	Task List	Input Products	Output Products
AN CUS	<p>SI.5.10 Verify and obtain approval of the *<i>Software User Documentation</i>, if appropriate (see SI.5.9)</p> <p>Verify consistency of the <i>Software User Documentation</i> with the <i>Software</i>. The results found are documented in a <i>Verification Results</i> and corrections are made until the document is approved by CUS.</p> <p>*(Optional)</p>	<p>*<i>Software User Documentation</i></p> <p><i>Software [tested]</i></p>	<p><i>Verification Results</i></p> <p>*<i>Software User Documentation [verified]</i></p>
TL	<p>SI.5.11 Incorporate the <i>Test Cases and Test Procedures</i>, <i>Software</i>, <i>Traceability Record</i>, <i>Test Report</i>, *<i>Product Operation Guide</i> and *<i>Software User Documentation</i> to the <i>Software Configuration</i> as part of the baseline.</p> <p>*(Optional)</p>	<p><i>Test Cases and Test Procedures</i></p> <p><i>Software [tested]</i></p> <p><i>Test Report</i></p> <p><i>Traceability Record [updated]</i></p> <p>*<i>Product Operation Guide [verified]</i></p> <p>*<i>Software User Documentation [verified]</i></p>	<p><i>Software Configuration</i></p> <ul style="list-style-type: none"> <li>- <i>Test Cases and Test Procedures [baselined]</i></li> <li>- <i>Software [tested, baselined]</i></li> <li>- <i>Traceability Record [updated, baselined]</i></li> <li>- <i>Test Report [baselined]</i></li> <li>- *<i>Product Operation Guide [verified, baselined]</i></li> <li>- *<i>Software User Documentation [verified, baselined]</i></li> </ul>

## SI.6 Product Delivery (SI.O6, SI.O7)

The Product Delivery activity provides the integrated software product to the Customer. The activity provides:

- Work Team review of the *Project Plan* to determine task assignment.
- Verified *Maintenance Documentation*.
- Delivery of the software product and applicable documentation in accordance with the *Delivery Instructions*.

**Table 20 — SI.6 task list**

Role	Task List	Input Products	Output Products
TL WT	SI.6.1 Assign <i>Tasks</i> to the work team members related to their role, according to the current <i>Project Plan</i> .	<p><i>Project Plan</i></p> <ul style="list-style-type: none"> <li>- <i>Tasks</i></li> </ul>	
DES	SI.6.2 Understand <i>Software Configuration</i> .	<i>Software Configuration</i>	
DES	SI.6.3 Document the <i>Maintenance Documentation</i> or update the current one.	<i>Software Configuration</i>	<i>Maintenance Documentation</i>

Role	Task List	Input Products	Output Products
DES TL	SI.6.4 Verify and obtain approval of the <i>Maintenance Documentation</i> .  Verify consistency of <i>Maintenance Documentation</i> with <i>Software Configuration</i> . The results found are documented in a <i>Verification Results</i> and corrections are made until the document is approved by TL.	<i>Maintenance Documentation</i>  <i>Software Configuration</i>	<i>Verification Results</i>  <i>Maintenance Documentation [verified]</i>
TL	SI.6.5 Incorporate the <i>Maintenance Documentation</i> as baseline for the <i>Software Configuration</i> .	<i>Software Configuration</i>  <i>Maintenance Documentation [verified]</i>	<i>Software Configuration</i> - <i>Maintenance Documentation [verified, baselined]</i>
TL	SI.6.6 Perform delivery according to <i>Delivery Instructions</i> .	<i>Project Plan</i> - <i>Delivery Instructions</i>  <i>Software Configuration</i>	<i>Software Configuration [delivered]</i>

### 7.7.2 SI incorporation to the *Project Repository*

The list of products to be saved in *Project Repository*. After the incorporation, *Version Control Strategy* has to be applied to: *Requirements Specification*, *Software Design*, *Traceability Record*, *Test Cases and Test Procedures*, *Software Components*, *Software*, *Product Operation Guide*, *Software User Documentation* and *Maintenance Documentation*.

**Table 21 — SI repository products**

Product
<i>Requirements Specification</i>
<i>Software User Documentation</i>
<i>Software Design</i>
<i>Traceability Record</i>
<i>Test Cases and Test Procedures</i>
<i>Software Components</i>
<i>Software</i>
<i>Product Operation Guide</i>
<i>Maintenance Documentation</i>
<i>Test Report</i>
<i>Verification Results</i>
<i>Validation Results</i>

## 8 Roles

This is an alphabetical list of the roles, its abbreviations and suggested competencies description. This list is showed as a four-column table for presentation purpose only.

**Table 22 — Roles**

	<b>Role</b>	<b>Abbreviation</b>	<b>Competency</b>
1.	Analyst	AN	<p>Knowledge and experience eliciting, specifying and analyzing the requirements.</p> <p>Knowledge in designing user interfaces and ergonomic criteria.</p> <p>Knowledge of the revision techniques.</p> <p>Knowledge of the editing techniques.</p> <p>Experience on the software development and maintenance.</p>
2.	Customer	CUS	<p>Knowledge of the Customer processes and ability to explain the Customer requirements.</p> <p>The Customer (representative) must have the authority to approve the requirements and their changes.</p> <p>The Customer includes user representatives in order to ensure that the operational environment is addressed.</p> <p>Knowledge and experience in the application domain.</p>
3.	Designer	DES	<p>Knowledge and experience in the <i>Software Components</i> and architecture design.</p> <p>Knowledge of the revision techniques.</p> <p>Knowledge and experience in the planning and performance of integration tests.</p> <p>Knowledge of the editing techniques.</p> <p>Experience on the software development and maintenance.</p>
4.	Programmer	PR	<p>Knowledge and/or experience in programming, integration and unit tests.</p> <p>Knowledge of the revision techniques.</p> <p>Knowledge of the editing techniques.</p> <p>Experience on the software development and maintenance.</p>
5.	Project Manager	PM	<p>Leadership capability with experience making decisions, planning, personnel management, delegation and supervision, finances and software development.</p>
6.	Technical Leader	TL	<p>Knowledge and experience in the software process domain.</p>



	Role	Abbreviation	Competency
7.	Work Team	WT	<p>Knowledge and experience according to their roles on the project: TL, AN, DES, and/or PR.</p> <p>Knowledge on the standards used by the Customer and/or by the VSE.</p>

## 9 Product description

This is an alphabetical list of the input, output and internal process products, its descriptions, possible states and the source of the product. The source can be another process or an external entity to the project, such as the Customer. This list is showed as a four-column table for presentation purpose only. Product items in the following tables are based on ISO/IEC 15289 Information Items with some exceptions.

**Table 23 — Product Descriptions**

	Name	Description	Source
1.	<i>Acceptance Record</i>	<p>Documents the Customer acceptance of the <i>Deliverables</i> of the project. It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>- Record of the receipt of the delivery</li> <li>- Identifies the date received</li> <li>- Identifies the delivered elements</li> <li>- Records the verification of any Customer acceptance criteria defined</li> <li>- Identifies any open issues (if applicable)</li> <li>- Signed by receiving Customer</li> </ul>	Project Management
2.	<i>Change Request</i>	<p>Identifies a <i>Software</i>, or documentation problem or desired improvement, and requests modifications. It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>- Identifies purpose of change</li> <li>- Identifies request status</li> <li>- Identifies requester contact information</li> <li>- Impacted system(s)</li> <li>- Impact to operations of existing system(s) defined</li> <li>- Impact to associated documentation defined</li> <li>- Criticality of the request, date needed</li> </ul> <p>The applicable statuses are: initiated, evaluated, and accepted.</p>	Software Implementation  Customer  Project Management
3.	<i>Correction Register</i>	<p>Identifies activities established to correct a deviation or problem concerning the accomplishment of a plan. It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>- Identifies the initial problem</li> <li>- Defines a solution</li> <li>- Identifies corrective actions taken</li> <li>- Identifies the ownership for completion of defined actions</li> <li>- Identifies the open date and target closure date</li> <li>- Contains a status indicator</li> <li>- Indicates follow up actions</li> </ul>	Project Management

	Name	Description	Source
4.	<i>Maintenance Documentation</i>	<p>Describes the <i>Software Configuration</i> and the environment used for development and testing (compilers, design tools, construction and tests). It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>- Includes or refers to all <i>Software Configuration</i> elements developed during implementation</li> <li>- Identifies environment used for development and testing (compilers, design tools, construction and tests tools)</li> </ul> <p>It is written in terms that maintenance personnel can understand.</p> <p>The applicable statuses are: verified and baselined.</p>	Software Implementation
5.	<i>Meeting Record</i>	<p>Records the agreements established with Customer and/or Work Team. It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>- Purpose of meeting</li> <li>- Attendees</li> <li>- Date, place held</li> <li>- Reference to previous minutes</li> <li>- What was accomplished</li> <li>- Identifies issues raised</li> <li>- Any open issues</li> <li>- Agreements</li> <li>- Next meeting, if any.</li> </ul> <p>The applicable status is: updated.</p>	Project Management
6.	<i>Product Operation Guide</i>	<p>Contains the necessary information to install and manage the <i>Software</i>. It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>- Criteria for operational use</li> <li>- A description of how to operate the product including: <ul style="list-style-type: none"> <li>- operational environment required</li> <li>- supporting tools and material (e.g. user manuals) required</li> <li>- possible safety warnings</li> <li>- start-up preparations and sequence</li> <li>- frequently asked questions (FAQ)</li> <li>- sources of further information and help to operate the product</li> </ul> </li> <li>- Certification and safety approvals</li> <li>- Warranty and replacement instructions</li> <li>- It should be written in terms that the personnel responsible for the operation can understand.</li> </ul> <p>The applicable statuses are: verified and baselined.</p>	Software Implementation

	Name	Description	Source
7.	<i>Progress Status Record</i>	<p>Records the status of the project against the <i>Project Plan</i>. It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>- Status of actual <i>Tasks</i> against planned <i>Tasks</i></li> <li>- Status of actual results against established <i>Objectives</i> / goals</li> <li>- Status of actual resource allocation against planned <i>Resources</i></li> <li>- Status of actual cost against budget estimates</li> <li>- Status of actual time against planned schedule</li> <li>- Status of actual risk against previously identified</li> <li>- Record of any deviations from planned <i>Tasks</i> and reason why.</li> </ul> <p>The applicable status is: evaluated.</p>	Project Management
8.	<i>Project Plan</i>	<p>Presents how the project processes and activities will be executed to assure the project's successful completion, and the quality of the deliverable products. It includes the following elements which may have the characteristics as follows:</p> <ul style="list-style-type: none"> <li>- <i>Product Description</i> <ul style="list-style-type: none"> <li>- Purpose</li> <li>- General Customer requirements</li> </ul> </li> <li>- <i>Scope</i> description of what is included and what is not</li> <li>- <i>Objectives</i> of the project</li> <li>- <i>Deliverables</i> - list of products to be delivered to Customer</li> <li>- <i>Tasks</i>, including verification, validation and reviews with Customer and Work Team, to assure the quality of work products. <i>Tasks</i> may be represented as a Work Breakdown Structure (WBS).</li> <li>- <i>Estimated Duration</i> of tasks</li> <li>- <i>Resources</i> (humans, materials, standards, equipment and tools) including the required training, and the schedule when the <i>Resources</i> are needed.</li> <li>- <i>Composition of Work Team</i></li> <li>- <i>Schedule of the Project Tasks</i>, the expected start and completion date for each task, and the relationship and dependencies of the <i>Tasks</i>.</li> <li>- <i>Estimated Effort and Cost</i></li> <li>- <i>Identification of Project Risks</i></li> <li>- <i>Version Control Strategy</i> <ul style="list-style-type: none"> <li>- Product repository tools or mechanism identified</li> <li>- Location and access mechanisms for the repository specified</li> <li>- Version identification and control defined</li> <li>- Backup and recovery mechanisms defined</li> <li>- Storage, handling and delivery (including archival and retrieval) mechanisms specified</li> </ul> </li> <li>- <i>Delivery Instructions</i> <ul style="list-style-type: none"> <li>- Elements required for product release identified (i.e., hardware, software, documentation etc.)</li> <li>- Delivery requirements</li> <li>- Sequential ordering of <i>Tasks</i> to be performed</li> <li>- Applicable releases identified</li> <li>- Identifies all delivered <i>Software Components</i> with version information</li> </ul> </li> </ul>	Project Management

	Name	Description	Source
		<ul style="list-style-type: none"> <li>- Identifies any necessary backup and recovery procedures</li> </ul> <p>The applicable statuses are: verified, accepted, updated and reviewed.</p>	
9.	<i>Project Repository</i>	<p>Electronic container to store project work products and deliveries. It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>- Stores project work products</li> <li>- Stores released <i>Deliverables</i> products</li> <li>- Storage and retrieval capabilities</li> <li>- Ability to browse content</li> <li>- Listing of contents with description of attributes</li> <li>- Sharing and transfer of work products between affected groups</li> <li>- Effective controls over access</li> <li>- Maintain work products descriptions</li> <li>- Recovery of archive versions of work products</li> <li>- Ability to report work products status</li> <li>- Changes to work products are tracked to <i>Change Requests</i></li> </ul> <p>The applicable statuses are: recovered and updated.</p>	Project Management
10.	<i>Project Repository Backup</i>	Repository used to backup the <i>Project Repository</i> and, if necessary, to recover the information.	Project Management
11.	<i>Requirements Specification</i>	<p>Identifies the software requirements. It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>- Introduction –general description of <i>Software</i> and its use within the <i>Scope</i> of the Customer business;</li> <li>- Requirements description: <ul style="list-style-type: none"> <li>- Functionality – established needs to be satisfied by the <i>Software</i> when it is used in specific conditions. Functionality must be adequate, accurate and safe</li> <li>- User interface – definition of those user interface characteristics that allow to understand and learn the <i>Software</i> easily so the user be able to perform his/her <i>Tasks</i> efficiently including the interface exemplar description</li> <li>- External interfaces – definition of interfaces with other software or hardware</li> <li>- Reliability – specification of the software execution level concerning the maturity, fault tolerance and recovery</li> <li>- Efficiency – specification of the software execution level concerning the time and use of the <i>Resources</i></li> <li>- Maintenance – description of the elements facilitating the understanding and execution of the future <i>Software</i> modifications</li> <li>- Portability – description of the <i>Software</i> characteristics that allow its transfer from one place to other</li> </ul> </li> </ul>	Software Implementation

	Name	Description	Source
		<ul style="list-style-type: none"> <li>- Design and construction limitations/constraints – needs imposed by the Customer</li> <li>- Interoperability – capability for two or more systems or <i>Software Components</i> be able to change information each other and use it</li> <li>- Reusability – feature of any product/sub-product, or a part of it, so that it can be used by several users as an end product, in the own software development, or in the execution of other software products</li> <li>- Legal and regulative – needs imposed by laws, regulations, etc.</li> </ul> <p>Each requirement is identified, unique and it is verifiable or can be assessed.</p> <p>The applicable statuses are: verified, validated and baselined.</p>	
12.	<i>Software</i>	<p>Software item (<i>Software</i> source and executable code) for a Customer, constituted by a collection of integrated <i>Software Components</i>.</p> <p>The applicable statuses are: tested and baselined.</p>	Software Implementation
13.	<i>Software Components</i>	<p>A set of related code units.</p> <p>The applicable statuses are: unit tested, corrected and baselined.</p>	Software Implementation
14.	<i>Software Configuration</i>	<p>A uniquely identified and consistent set of software products including:</p> <ul style="list-style-type: none"> <li>- <i>Requirements Specification</i></li> <li>- <i>Software Design</i></li> <li>- <i>Traceability Record</i></li> <li>- <i>Software Components</i></li> <li>- <i>Software</i></li> <li>- <i>Test Cases and Test Procedures</i></li> <li>- <i>Test Report</i></li> <li>- <i>Product Operation Guide</i></li> <li>- <i>Software User Documentation</i></li> <li>- <i>Maintenance Documentation</i></li> </ul> <p>The applicable statuses are: delivered and accepted.</p>	Software Implementation
15.	<i>Software Design</i>	<p>Textual and graphical information on the <i>Software</i> structure. This structure may include the following parts:</p> <p>Architectural high level software design – Describes the overall <i>Software</i> structure:</p> <ul style="list-style-type: none"> <li>- Identifies the required <i>Software Components</i></li> <li>- Identifies the relationship between <i>Software Components</i></li> <li>- Consideration is given to any required: <ul style="list-style-type: none"> <li>- <i>Software</i> performance characteristics</li> <li>- hardware, software and human interfaces</li> <li>- security characteristics</li> <li>- database design requirements</li> <li>- error handling and recovery attributes</li> </ul> </li> </ul>	Software Implementation

	Name	Description	Source
		<p>Detailed low level software design – includes details of the <i>Software Components</i> to facilitate its construction and test within the programming environment;</p> <ul style="list-style-type: none"> <li>- Provides detailed design (could be represented as a prototype, flow chart, entity relationship diagram, pseudo code, etc.)</li> <li>- Provides format of input / output data</li> <li>- Provides specification of data storage needs</li> <li>- Establishes required data naming conventions</li> <li>- Defines the format of required data structures</li> <li>- Defines the data fields and purpose of each required data element</li> <li>- Provides the specifications of the program structure</li> </ul> <p>The applicable statuses are: verified and baselined.</p>	
16.	<i>Software User Documentation</i>	<p>Describes the way of using the <i>Software</i> based on the user interface. It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>- User procedures for performing specified <i>Tasks</i> using the <i>Software</i></li> <li>- Installation and de-installation procedures</li> <li>- Brief description of the intended use of the <i>Software</i> (the concept of operations)</li> <li>- The supplied and required <i>Resources</i></li> <li>- Needed operational environment</li> <li>- Availability of problem reporting and assistance</li> <li>- Procedures to access and exit the <i>Software</i></li> <li>- Lists and explains <i>Software</i> commands and system-provided messages to the user</li> <li>- As appropriate for the identified risk, it includes warnings, cautions, and notes, with corrections</li> <li>- It includes troubleshooting and error correction procedures.</li> </ul> <p>It is written in terms understandable by users.</p> <p>The applicable statuses are: preliminary, verified and baselined.</p>	Software Implementation
17.	<i>Statement of Work</i>	<p>Description of work to be done related to <i>Software</i> development. It may Include:</p> <ul style="list-style-type: none"> <li>- <i>Product Description</i> <ul style="list-style-type: none"> <li>- Purpose</li> <li>- General Customer requirements</li> </ul> </li> <li>- <i>Scope</i> description of what is included and what is not</li> <li>- <i>Objectives</i> of the project</li> <li>- <i>Deliverables</i> list of products to be delivered to Customer</li> </ul> <p>The applicable status is: reviewed.</p>	Customer

	Name	Description	Source
18.	<i>Test Cases and Test Procedures</i>	<p>Elements needed to test code. Test Case may include:</p> <ul style="list-style-type: none"> <li>- Identifies the test case</li> <li>- Test items</li> <li>- Input specifications</li> <li>- Output specifications</li> <li>- Environmental needs</li> <li>- Special procedural requirements</li> <li>- Interface dependencies</li> </ul> <p>Test Procedures may include:</p> <ul style="list-style-type: none"> <li>- Identifies: test name, test description and test completion date</li> <li>- Identifies potential implementation issues</li> <li>- Identifies the person who completed the test procedure</li> <li>- Identifies prerequisites</li> <li>- Identifies procedure steps including the step number, the required action by the tester and the expected results</li> </ul> <p>The applicable statuses are: verified and baselined.</p>	Software Implementation
19.	<i>Test Report</i>	<p>Documents the tests execution. It may include:</p> <ul style="list-style-type: none"> <li>- A summary of each defect</li> <li>- Identifies the related test case</li> <li>- Identifies the tester who found each defect</li> <li>- Identifies the severity for each defect</li> <li>- Identifies the affected function(s) for each defect</li> <li>- Identifies the date when each defect originated</li> <li>- Identifies the date when each defect was resolved</li> <li>- Identifies the person who resolved each defect</li> </ul> <p>The applicable status is: baselined.</p>	Software Implementation
20.	<i>Traceability Record</i>	<p>Documents the relationship among the requirements included in the <i>Requirements Specification</i>, <i>Software Design</i> elements, <i>Software Components</i>, <i>Test Cases</i> and <i>Test Procedures</i>. It may include:</p> <ul style="list-style-type: none"> <li>- Identifies requirements of <i>Requirements Specification</i> to be traced</li> <li>- Provides forward and backward mapping of requirements to <i>Software Design</i> elements, <i>Software Components</i>, <i>Test Cases</i> and <i>Test Procedures</i>.</li> </ul> <p>The applicable statuses are: verified, baselined and updated.</p>	Software Implementation

	Name	Description	Source
21.	<i>Verification Results</i>	Documents the verification execution. It may include the record of: <ul style="list-style-type: none"> <li>- Participants</li> <li>- Date</li> <li>- Place</li> <li>- Duration</li> <li>- Verification check-list</li> <li>- Passed items of verification</li> <li>- Failed items of verification</li> <li>- Pending items of verification</li> <li>- Defects identified during verification</li> </ul>	Project Management  Software Implementation
22.	<i>Validation Results</i>	Documents the validation execution, It may include the record of: <ul style="list-style-type: none"> <li>- Participants</li> <li>- Date</li> <li>- Place</li> <li>- Duration</li> <li>- Validation check-list</li> <li>- Passed items of validation</li> <li>- Failed items of validation</li> <li>- Pending items of validation</li> <li>- Defects identified during validation</li> </ul>	Software Implementation

## 10 Software tools requirements

Software tools that could be used to perform process activities.

### 10.1 Project Management process

**Table 24 — Project Management tools**

Activity	Resource List
Project Planning Project Plan Execution Project Assessment and Control Project Closure	Tools allowing document, manage and control the <i>Project Plan</i> and the use and management of the <i>Project Repository</i>



## 10.2 Software Implementation process

**Table 25 — Software Implementation tools**

<b>Activity</b>	<b>Resource List</b>
Software Implementation Initiation Software Requirements Analysis Software Architectural and Detailed Design Software Construction Software Integration and Tests Product Delivery	Documentation tools
Software Requirements Analysis	<i>Requirements Specification</i> tools
Software Architectural and Detailed Design	<i>Software Design</i> tools
Software Construction	Construction Tools
Software Integration and Tests	Tests tools, bug tracking tools

## Annex A (informative)

### Deployment Packages

In order to facilitate the implementation, by VSEs, of a Profile, a set of Deployment Packages are available. A deployment package is a set of artefacts 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 ISO/IEC 29110, 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 Part 5, ISO standards, such as ISO/IEC 12207, or models such as the CMMI developed by the Software Engineering Institute. Hence by deploying and implementing a package, a VSE can see its concrete step to achieve or demonstrate coverage to Part 5. Deployment Packages are designed such that a VSE can implement its content, without having to implement the complete framework at the same time. The table of content of a deployment package is illustrated in Table A-1.

**Table A.1 — Table of Content of a Deployment Package.**

1.	Technical Description
	Purpose of this document
	Why this Topic is important?
2.	Definitions
3.	Relationships with ISO/IEC 29110
4.	Overview of Processes, Activities, Tasks, Roles and Products
5.	Description of Processes, Activities, Tasks, Steps, Roles and Products
	Role Description
	Product Description
	Artefact Description
6.	Template(s)
7.	Example(s)
8.	Checklist(s)
9.	Tool(s)
10.	References to Other Standards and Models (e.g. ISO 9001, ISO/IEC 12207, CMMI)
11.	References
12.	Evaluation Form

For the Basic VSE Profile, a set of Deployment Packages are available, at no cost, on the Internet:

- a) Requirements Analysis
- b) Architecture and Detailed Design
- c) Construction and Unit Testing
- d) Integration and Test
- e) Verification and Validation
- f) Version Control

- g) Project Management
- h) Product Delivery
- i) Self-Assessment

## Bibliography

- [1] ISO/IEC 12207:2008, *Systems and software engineering — Software life cycle processes*
- [2] ISO/IEC 15289:2006, *Systems and software engineering — Content of systems and software life cycle process information products (Documentation)*
- [3] ISO/IEC TR 29110-1, *Software engineering — Lifecycle profiles for Very Small Entities (VSEs) — Part 1: Overview*
- [4] ISO/IEC 29110-4-1, *Software engineering — Lifecycle profiles for Very Small Entities (VSEs) — Part 4-1: Profile specifications — Generic profile group*
- [5] *OECD SME and Entrepreneurship Outlook*, 2005 Edition, Organisation for Economic Co-operation and Development, Paris, 2005



