



**International  
Standard**

**ISO/IEC 29110-5-4**

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

**Part 5-4:  
Agile software development  
guidelines**

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

*Partie 5-4: Lignes directrices relatives au développement de  
logiciels agiles*

**First edition  
2025-09**



**COPYRIGHT PROTECTED DOCUMENT**

© ISO/IEC 2025

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

ISO copyright office  
CP 401 • Ch. de Blandonnet 8  
CH-1214 Vernier, Geneva  
Phone: +41 22 749 01 11  
Email: [copyright@iso.org](mailto:copyright@iso.org)  
Website: [www.iso.org](http://www.iso.org)

Published in Switzerland

# Contents

	Page
<b>Foreword</b> .....	<b>v</b>
<b>Introduction</b> .....	<b>vi</b>
<b>1 Scope</b> .....	<b>1</b>
<b>2 Normative references</b> .....	<b>1</b>
<b>3 Terms, definitions and abbreviated terms</b> .....	<b>1</b>
3.1 Terms and definitions.....	1
3.2 Abbreviated terms.....	6
<b>4 Naming, diagramming and definition conventions</b> .....	<b>6</b>
<b>5 Overview</b> .....	<b>7</b>
5.1 General.....	7
5.2 Project management process of Basic profile and the agile events performed to manage the project.....	7
5.3 Software implementation process of Basic profile and the agile events performed to implement the project.....	8
5.4 Roles involved.....	9
<b>6 Agile software development process</b> .....	<b>10</b>
6.1 Agile project management process.....	10
6.1.1 Agile project management process purpose.....	10
6.1.2 Agile project management process outcomes.....	10
6.1.3 Overview of the agile project management process.....	10
6.2 Agile software implementation process.....	11
6.2.1 Agile software implementation process purpose.....	11
6.2.2 Agile software implementation process outcomes.....	11
6.2.3 Overview of the agile software implementation process.....	12
6.3 Event 1 - project vision meeting (contributes to outcome PM.01).....	13
6.3.1 Justification.....	13
6.3.2 Overview of event 1- project vision meeting.....	14
6.3.3 Tasks of event 1 - project vision meeting.....	14
6.4 Event 2 - estimation meeting (contributes to outcomes PM.01, PM.05).....	15
6.4.1 Justification.....	15
6.4.2 Overview of the event 2 - estimation meeting.....	15
6.4.3 Tasks of event 2 - estimation meeting.....	16
6.5 Event 3 - sprint planning (contributes to outcomes PM.01, PM.02, PM.04, PM.07, SI.01, SI.03, SI.05).....	17
6.5.1 Justification.....	17
6.5.2 Overview of event 3 - sprint planning.....	18
6.5.3 Tasks of event 3 - sprint planning.....	18
6.6 Event 4 - sprint (contributes to outcomes PM.02 PM.03, SI.03, SI.04, SI.05, SI.07).....	21
6.6.1 Justification.....	21
6.6.2 Overview of event 4 - sprint.....	21
6.6.3 Tasks of event 4 - sprint.....	22
6.7 Event 5 - daily scrum (contributes to outcomes PM.02, PM.03, PM.04, PM.05, PM.07).....	25
6.7.1 Justification.....	25
6.7.2 Overview of event 5 - daily scrum.....	25
6.7.3 Tasks of event 5 - daily scrum.....	25
6.8 Event 6 - sprint review (contributes to outcomes PM.02, PM.03, PM.04, SI.05, SI.07).....	27
6.8.1 Justification.....	27
6.8.2 Overview of event 6 - sprint review.....	27
6.8.3 Task of event 6 - sprint review.....	27
6.9 Event 7 - sprint retrospective (contributes to outcomes PM.02, PM.03, PM.04, PM.07).....	30
6.9.1 Justification.....	30
6.9.2 Overview of event 7 - sprint retrospective.....	30
6.9.3 Task of event 7 - sprint retrospective.....	30

<b>7</b>	<b>Role descriptions</b> .....	<b>32</b>
<b>8</b>	<b>Work products description</b> .....	<b>35</b>
<b>Annex A</b> (informative)	<b>Example of work products templates</b> .....	<b>59</b>
<b>Annex B</b> (informative)	<b>Overview of popular agile methods</b> .....	<b>65</b>
<b>Annex C</b> (informative)	<b>Development of the guidelines for the reinforcement of an agile environment</b> .....	<b>67</b>
<b>Annex D</b> (informative)	<b>Development of the guidelines for the reinforcement of an agile environment</b> .....	<b>70</b>
	<b>Bibliography</b> .....	<b>72</b>

## 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](http://www.iso.org/directives) or [www.iec.ch/members\\_experts/refdocs](http://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](http://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](http://www.iso.org/iso/foreword.html). In the IEC, see [www.iec.ch/understanding-standards](http://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*.

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](http://www.iso.org/members.html).

## 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. Since many VSEs develop and/or maintain system and software components used in systems, either as independent products or incorporated in larger systems, 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-sized 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 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, 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 find it helpful to have detailed, specified, procedures when beginning agile projects, rather than attempting to interpret and tailor the more flexible, high-level process requirements of ISO/IEC/IEEE. To address some of these difficulties, a set of guidelines have been developed based on a set of VSEs 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 systems; 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.

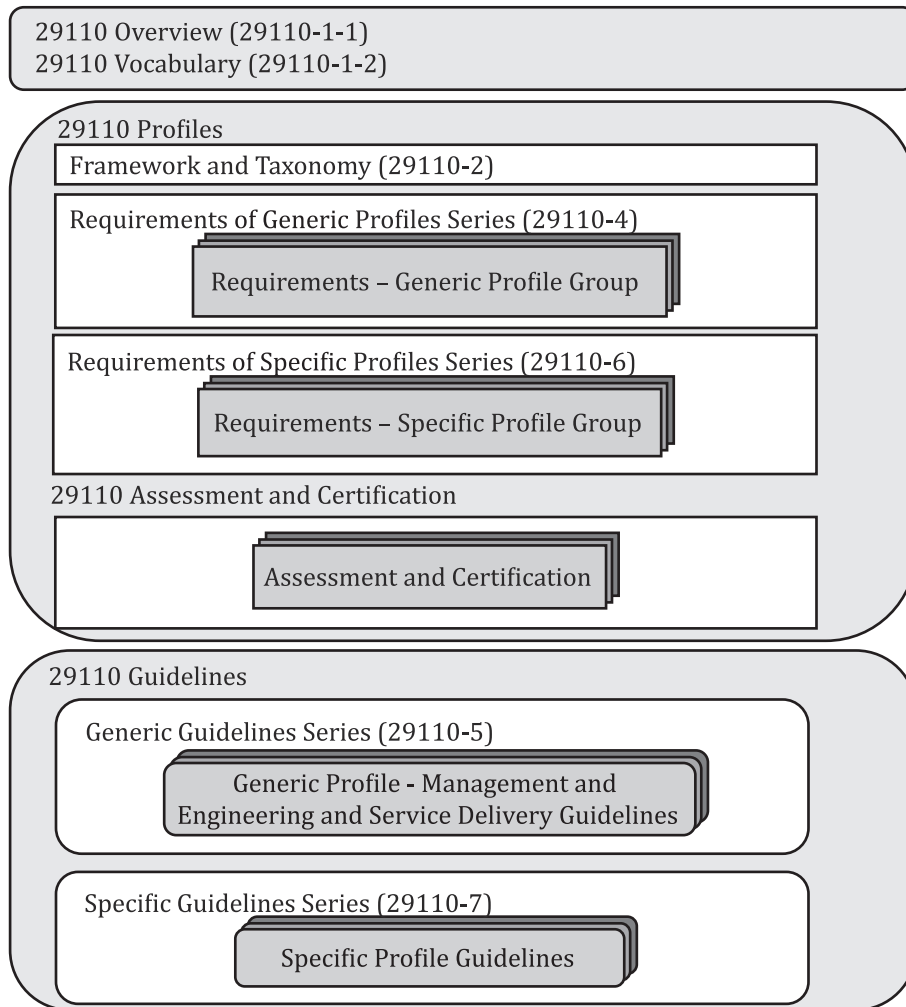
The ISO/IEC 29110 series can be applied at any phase of system or software development within a life cycle. The ISO/IEC 29110 series is intended to be used by VSEs that do not have experience or expertise in adapting/tailoring ISO/IEC/IEEE 12207 or ISO/IEC/IEEE 15288 standards to the needs of a specific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

## ISO/IEC 29110-5-4:2025(en)



**Figure 1 — ISO/IEC 29110 series**

ISO/IEC 29110-1-1 introduces processes, life cycle and standardization concepts, the taxonomy (catalog) 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, e.g. 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 29110-5-4:2025(en)

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 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 guidelines 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 or ISO/IEC 29110-6 and or ISO/IEC 29110-7, ISO/IEC 29110-5 can be developed with minimal impact to existing documents.

### 0.2 Introduction to this document

This document aims to help VSEs that want to reinforce their agile environment to develop software products using an agile approach with practices of the ISO/IEC 29110 series. Therefore, it can be implemented by organizations or projects that are implementing and using an agile environment and want to reinforce it using the processes and products recommended by this document.

This document provides examples of work products templates (see [Annex A](#)).

This document is based mainly on the scrum and XP agile methods (see [Annex B](#)).

This document also aims to help VSEs that want to initiate a certification of the two processes of the software Basic profile of the ISO/IEC 29110, as specified in ISO/IEC 29110-4-1, while performing their agile practices (see [Annex C](#)).

This document has been developed, using the management and engineering guidelines of the ISO/IEC 29110-5-1-2 software Basic profile, by modifying and adding elements (e.g. task, work product, role) for VSEs involved in the development of software using an agile approach (see [Annex C](#)). However, this document does not contain any requirements; and organizations cannot claim conformity to this document.

Using this document, VSEs can obtain the following benefits:

- project planning: by providing a set of events to enable the team to track the work and know what an impediment can be and assign tasks to each member of the team;
- effort estimation: by providing estimation techniques for the sprint planning event;
- progress tracking: by providing tools (e.g. burndown chart, product backlog, daily scrum) to update and track the work, the budget and the schedule until the project's closure;
- management of changes and artefacts: by providing a set of tasks to control the changes to the software and management work products;
- clarity of each role: by providing a set of tasks for each role;
- agile risk management: by providing work products to identify and manage risks;
- reduced rework: by providing a set of tasks that minimize errors and handle the detection and correction of defects effectively.

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

## Part 5-4: Agile software development guidelines

### 1 Scope

This document provides guidelines for very small entities (VSEs) that want to reinforce their agile environment to develop software using an agile approach with practices of the ISO/IEC 29110 series.

This document is applicable to VSEs that do not develop business- or safety-critical products.

### 2 Normative references

The following documents are referred to in the text in such a way that some or all their content constitutes requirements 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 29110-1-2, *Systems and software engineering — Lifecycle profiles for Very Small Entities (VSEs) — Part 1-2: Vocabulary*

### 3 Terms, definitions and abbreviated terms

#### 3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 29110-1-2 and the following apply.

ISO and IEC maintain terminological 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/>

##### 3.1.1

##### **acceptance criteria**

working *agreement* (3.1.6) on the required quality attribute that work meets before it completes a specified story or feature, item or artefact

##### 3.1.2

##### **agile**

approach to development, delivery and maintenance of products and services by enabling rapid response to feedback

Note 1 to entry: The required specification for the software under development is elaborated with specific requirements only when the work is started. This lean principle is meant to avoid waste of work and to provide an *agile team* (3.1.5) with means of prioritizing their work.

[SOURCE: ISO/IEC 33202:2024, 3.2, modified — "need specification" has been replaced by "required specification".]

### 3.1.3

#### **agile development**

development approach based on iterative development, frequent inspection and adaptation, and incremental deliveries in which requirements and solutions evolve through collaboration in cross-functional teams and through continuous stakeholder feedback

[SOURCE: ISO/IEC/IEEE 26515:2018, 3.1, modified — Note 1 to entry has been removed.]

### 3.1.4

#### **agile environment**

organizational culture, infrastructure, and methodologies that support *agile development* ([3.1.3](#))

[SOURCE: ISO/IEC/IEEE 26515:2018, 3.2]

### 3.1.5

#### **agile team**

organization or team using *agile development* ([3.1.3](#)) methods and approaches

Note 1 to entry: Typically, with roles such as team lead, project manager, user or user representative, software and information developers, and testers

[SOURCE: ISO/IEC/IEEE 26515:2018, 3.3]

### 3.1.6

#### **agreement**

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

EXAMPLE Contract, memorandum of agreement.

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

### 3.1.7

#### **backlog**

collection of *agile* ([3.1.2](#)) *features* ([3.1.15](#)) or stories of both functional and non-functional requirements that are typically sorted in an order based on value priority

[SOURCE: ISO/IEC/IEEE 26515:2018, 3.4]

### 3.1.8

#### **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]

### 3.1.9

#### **build**

operational version of a system or component that incorporates a specified subset of the capabilities that the final product will provide

[SOURCE: IEEE 828-2012, 2.1, IEEE dictionary]

### 3.1.10

#### **burndown chart**

graph that represents the work remaining to do on a project

[SOURCE: ISO/IEC/IEEE 26511:2018, 3.1.6]

### 3.1.11

#### **burnup chart**

graphical representation of the work completed toward the release of a product

[SOURCE: Agile Practice Guide, Project Management Institute, 2017]

### 3.1.12

#### **continuous integration**

technique that continually merges artefacts, including source code updates from all developers on a team, into a shared mainline to build and test the developed system

[SOURCE: ISO/IEC/IEEE 32675:2022, 3.1]

### 3.1.13

#### **definition of done**

##### **DoD**

statement on the required quality attributes that work meets before the work completes a specified life cycle activity or task and is ready for use

[SOURCE: ISO/IEC 33202:2024, 3.13]

### 3.1.14

#### **event**

ceremony

key meeting in which a set of tasks are performed to help the *scrum leader* (3.1.28), *product owner* (3.1.20), and developer to organize, evaluate, and adjust the work to be done to develop a product

EXAMPLE *Sprint* (3.1.31) planning, *daily scrum* (3.1.29), sprint review and sprint retrospective.

### 3.1.15

#### **feature**

functional or non-functional distinguishing characteristic of a system

Note 1 to entry: A feature, item or artefact is usually an enhancement to an existing system.

[SOURCE: ISO/IEC/IEEE 26515:2018, 3.7, modified — The original note 1 to entry has been replaced by a new one.]

### 3.1.16

#### **increment**

tested, deliverable version of a *software product* (3.1.30) that provides new or modified capabilities

[SOURCE: ISO/IEC 33202:2024, 3.17]

### 3.1.17

#### **pair programming**

practice from eXtreme Programming (XP) that typically involves one developer writing source code or configuring a system while a second developer observes the work to look for errors and to determine whether the work completed is accurate and meets its requirements

Note 1 to entry: Pairs alternate roles periodically so a different developer writes code while the other developer observes.

### 3.1.18

#### **planning poker**

team-based estimation approach whereby relative estimates of the effort required to develop and test each *user story* (3.1.37) are set via consensus

Note 1 to entry: The approach uses poker cards that are typically based on a Fibonacci sequence (i.e. 1, 2, 3, 5, 8, 13).

### 3.1.19

#### **product backlog**

It is an emergent, ordered list of what is needed to improve the product. It is the single source of work undertaken by the *agile team* (3.1.5)

### 3.1.20

#### **product owner**

designated stakeholder accountable for defining and accepting outcomes of the work and managing *backlog* (3.1.7), while aligning with the stakeholder needs

Note 1 to entry: A product owner is also the 'voice of the user' or the user representative on the team.

[SOURCE: ISO/IEC 33202:2024, 3.2, modified — Note 1 to entry has been added.]

### 3.1.21

#### **process purpose**

high-level objective of performing the process and the likely outcomes of effective implementation of the process

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

### 3.1.22

#### **process outcome**

observable result of the successful achievement of the process

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

### 3.1.23

#### **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.

### 3.1.24

#### **repository**

organized and persistent data storage that allows data retrieval

[SOURCE: ISO/IEC/IEEE 26511:2018, 3.1.24]

### 3.1.25

#### **safety-critical product**

software whose failure or malfunction can result in one (or more) of the following outcomes:

- death or serious injury to people
- loss or severe damage to equipment or property
- environmental harm

[SOURCE: ISO/IEC 23643:2020, 3.15, modified — EXAMPLE has been removed.]

### 3.1.26

#### **self-organizing team**

group composed of motivated individuals working together toward meeting a goal

Note 1 to entry: They have the ability and authority for decision-making. They manage their work and readily adapt for changing demands.

Note 2 to entry: Principles of self-organizing teams are competency, collaboration, motivation, trust, respect and continuity.

### 3.1.27

#### **scrum**

iterative project management framework used in *agile development* (3.1.3), in which a team agrees on development items from *product backlog* (3.1.19) and produces them within a short duration of a few weeks

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.3637]

**3.1.28**

**scrum leader**

person who facilitates the *scrum* (3.1.27) process within a team or project

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.3638, modified — The term has been changed from "scrum master" to "scrum leader".]

**3.1.29**

**daily scrum**

in *agile development* (3.1.3) methods, brief daily meeting where team members share accomplishments from the prior daily meeting, plans for the current day and identification of impediments

Note 1 to entry: The *product backlog* (3.1.19) is sorted in prioritized order.

**3.1.30**

**software product**

set of computer programs, procedures, and possibly associated documentation and data

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

**3.1.31**

**sprint**

short time frame, in which a set of software *features* (3.1.15) is developed, leading to a working product that can be demonstrated to stakeholders

Note 1 to entry: In some organizations, a sprint is known as an iteration.

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.3914]

**3.1.32**

**sprint backlog**

iteration backlog

sprint catalogue

set of *backlog* (3.1.7) items [e.g. *user stories* (3.1.37)] that are selected for inclusion in a given *sprint* (3.1.31)

**3.1.33**

**story points**

units of measure for expressing an estimate of the overall effort required to fully implement a *product backlog* (3.1.19) item or any other piece of work

**3.1.34**

**time-boxed**

having prescribed duration limit for a project task

[SOURCE: Adapted from Software Extension to the PMBOK Guide Fifth Edition]

**3.1.35**

**task board**

kanban board

scrum board

visual representation of an *agile team's* (3.1.5) progress within an iteration

**3.1.36**

**unit test**

testing of individual routines and modules by the developer or an independent tester

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.4429, definition 1]

**3.1.37**

**user story**

brief description of desired functionality

Note 1 to entry: User story can describe the stakeholder roles, goals, benefits, and motivation.

Note 2 to entry: Work in *backlog* (3.1.7) can be referred in other ways like epics, features, items or artefacts.

[SOURCE: ISO/IEC 33202:2024, 3.1.54, modified — In Note 2 to entry, "items or artefacts" has been added.]

### 3.1.38

#### **velocity**

rate of current work unit completion, measured as work units completed per fixed time period, such as *story points* (3.1.33), delivered *features* (3.1.15), functions, function points, *user stories* (3.1.37), use cases, or requirements completed in a given time period

Note 1 to entry: Used as a measure of burndown rate or burnup rate.

[SOURCE: ISO/IEC/IEEE 32675:2022, 3.1]

## 3.2 Abbreviated terms

DT	developers
CUS	customer
PM	project management
PJM	project manager
PO	product owner
SI	software implementation
SL	scrum leader
TE	task event
TL	technical leader
WP	work product

## 4 Naming, diagramming and definition conventions

This clause provides a description of the process structure, and the notation used to describe the processes.

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

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.

Events – key meetings throughout the software development to organize, evaluate, and adjust the work to be done to develop a product in an agile environment. For the purpose of this guide, an event is composed of a set of tasks to be performed. An event is the first level of process workflow decomposition, and the second one is a task. Events are identified by a process name abbreviation followed by a consecutive number and the event name. Each event description is identified by the event name and the list of related outcomes surrounded by parentheses “()”. For example, PM.O1 Project planning (PM.O1, PM.O5, PM.O6, PM.O7) means the event PM.O1 Project planning contributes to the achievement of the listed outcomes: PM.O1, PM.O5, PM.O6 and PM.O7. The event description begins with the task summary and is followed by the task descriptions table. The selection of the techniques or methods is left to the VSE or project team.

Input work products – work 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. They are identified by

the abbreviation of the process name and shown as two column tables of work product names and sources. Parentheses after the work product name show the state of the work product.

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 or organizational management. They are identified by the abbreviation of the process name and shown as two column tables of work product names and destinations. Parentheses after the work product name show the state of the work product.

Internal work products – work products generated and consumed by the process itself. An internal work product is not reviewed or approved by the customer. They are identified by the abbreviation of the process name and shown as one column table of the work product names. Parentheses after the work product name show the state of the work product.

Roles involved – names and abbreviations for the sets of tasks and responsibilities which project team members are assigned. 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 [Table 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 an event, and the smaller square-edged rectangles indicate the work products. The directional or bidirectional thick arrows indicate the major flow of information between events. 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.

Task – description of the task to be performed. The agile practices to be performed from an agile method such as scrum or XP. Therefore, an agile task is identified by indicating that is a task event TE; PM or SI that specifies if it is a project management event or a software implementation event, and the event number with a consecutive number, for example TE.PM.1.2, TE.PM.2.1, TE.SI.1.1.

Input work products - work products needed to execute a task.

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

Incorporation to organizational repository – list of work products to be saved in organizational repository.

NOTE 1 Tables used in the process description are for presentation purpose only.

NOTE 2 The term ‘Basic’ is using a capital ‘B’ to indicate an ISO/IEC 29110 profile (e.g. the Basic profile) while the term ‘basic’ is used when referring to the most important part of something (e.g. basic principles).

## 5 Overview

### 5.1 General

This clause provides guidelines for VSEs that want to reinforce their agile environment to develop software using an agile approach with practices of the ISO/IEC 29110 series. See [Annex C](#) for more detail on how this document was developed.

### 5.2 Project management process of Basic profile and the agile events performed to manage the project

[Figure 2](#) presents an overview of the project management process of the software Basic profile compared with the events of the scrum method performed to manage the project. The events not related to project management are greyed. Each event is described in detail in [Clause 6](#).

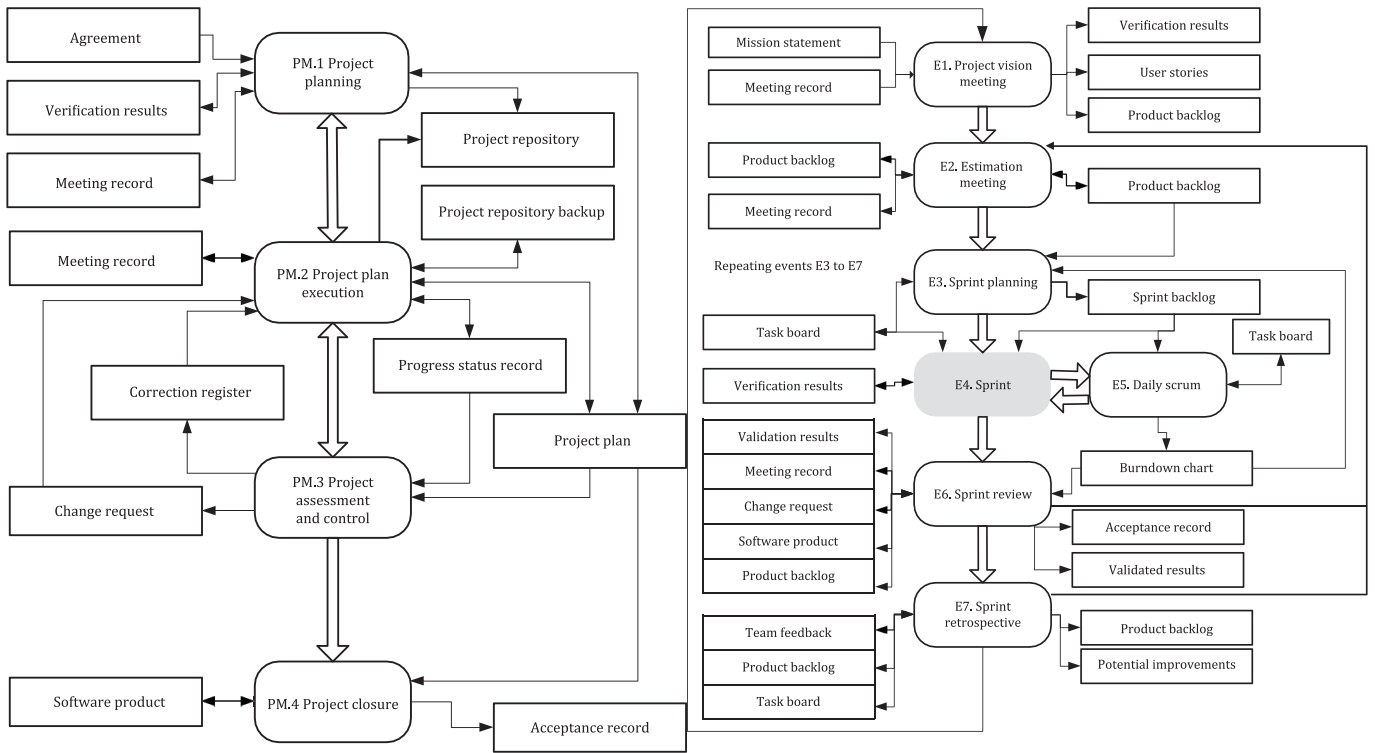


Figure 2 — Overview of the project management process of the Basic profile and the agile events

### 5.3 Software implementation process of Basic profile and the agile events performed to implement the project

Figure 3 presents an overview of the software implementation process of the software Basic profile compared with the events of the scrum method performed to implement the project. The events not related to software implementation are greyed. Each event is described in detail in Clause 6.

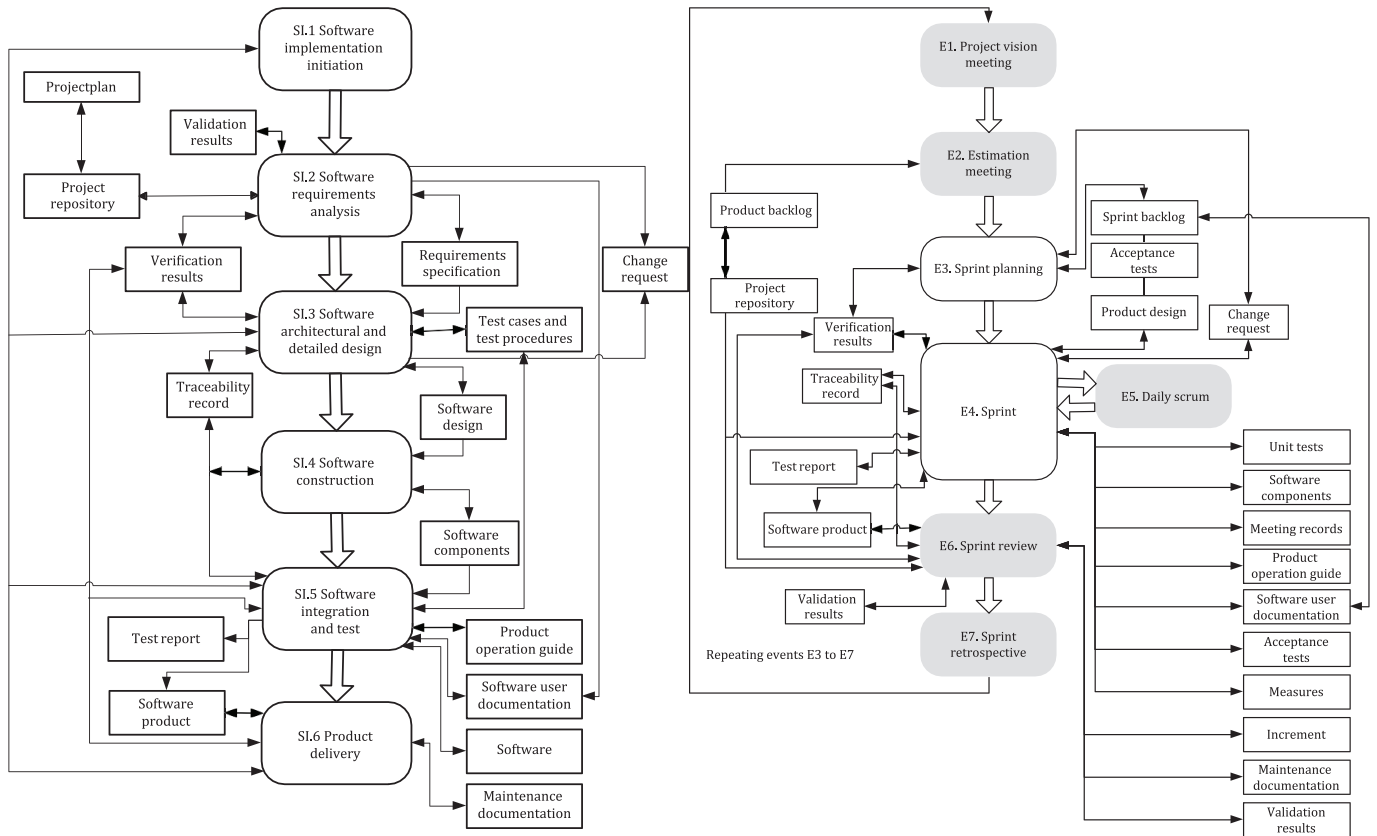


Figure 3 — Overview of the software implementation process of the Basic profile and the agile events

## 5.4 Roles involved

The list of roles involved for the project management events are:

- Customer
- Product owner
- Scrum leader
- Developers

The list of roles involved for the software implementation events are:

- Customer
- Developers
- Product owner
- Scrum leader

## 6 Agile software development process

### 6.1 Agile project management process

#### 6.1.1 Agile project management process purpose

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

This document is intended to be used by the VSE to establish processes to implement agile approach based on the VSE or project needs

#### 6.1.2 Agile project management process outcomes

PM.01. The project plan for the execution of the project is developed according to the agreement and reviewed and accepted by the customer. The tasks and resources necessary to complete the work are sized and estimated.

PM.02. The progress of the project is monitored against the project plan and recorded in the progress status record. The customer approves changes to the project plan. Corrections to remediate problems and deviations from the plan are taken when project targets are not achieved. Closure of the project is performed to get the customer acceptance documented in the acceptance record.

PM.03. The change requests are addressed through their reception and analysis. Changes to software requirements are evaluated for cost, schedule and technical impact. Change requests that impact the project plan, the requirements or the agreement are approved by the customer.

PM.04. Review meetings with the work team and the customer are held. Agreements are registered and tracked.

PM.05. Risks are identified as they develop and during the conduct of the project.

PM.06. A software version control strategy is developed. Items of software product are identified, defined and baselined. Modifications and releases of the items are controlled and made available to the customer and work team. The storage, handling and delivery of the items are controlled.

PM.07. Software quality assurance is performed to provide assurance that work products and processes comply with the project plan and requirements specification.

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

#### 6.1.3 Overview of the agile project management process

[Figure 4](#) presents an overview of agile project management. The events not related to project management are greyed. Each event is described in detail in [6.3](#) to [6.9](#).

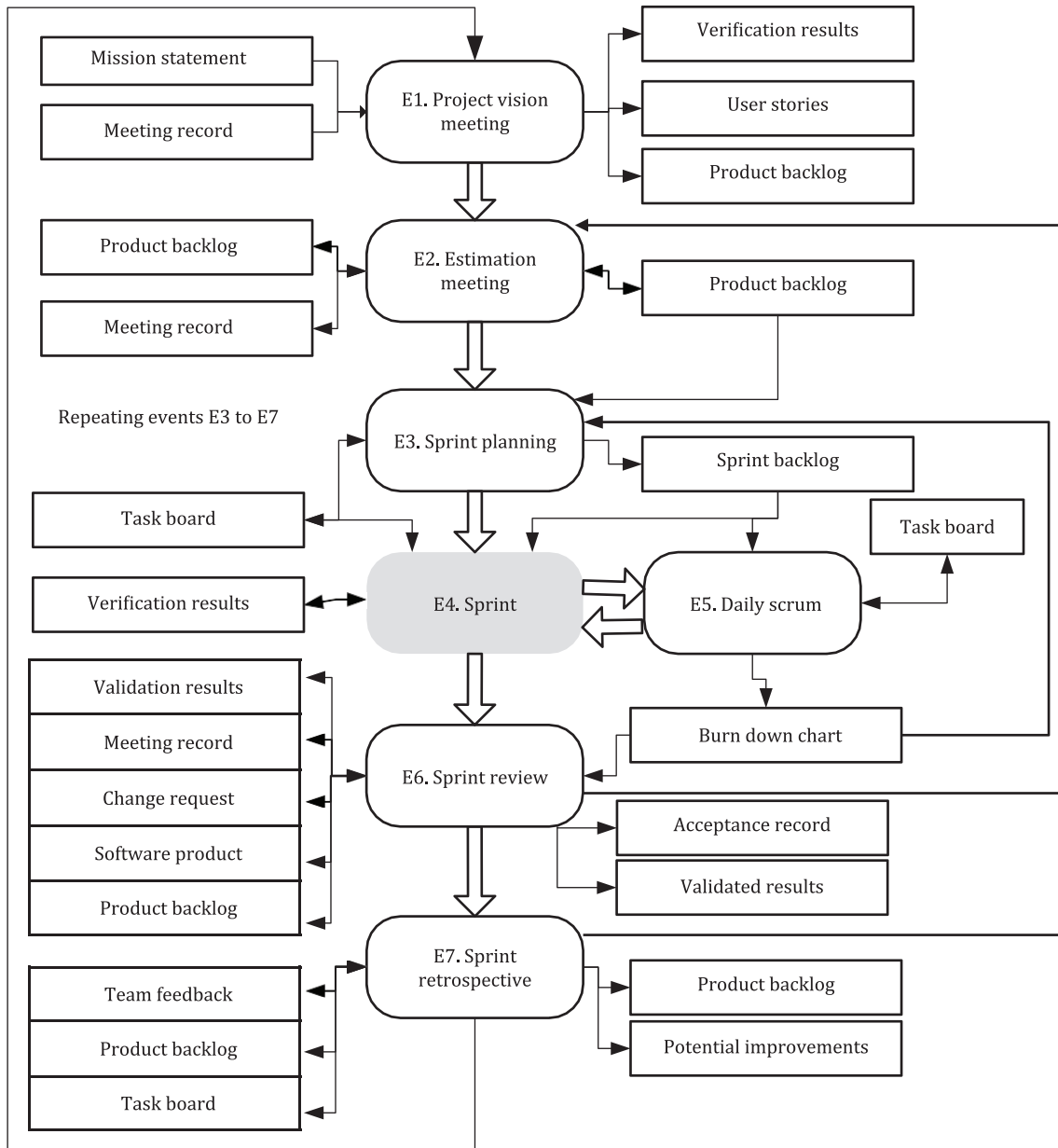


Figure 4 — Overview of the agile project management process

## 6.2 Agile software implementation process

### 6.2.1 Agile software implementation process purpose

The purpose of the agile software implementation process is the systematic performance of the analysis, design, construction, integration, testing activities and assembly of work products for a new or modified software product according to the specified requirements. This document is intended to be used by the VSE to establish processes to implement agile approach based on the VSE or project needs.

### 6.2.2 Agile software implementation process outcomes

SI.01. Tasks of the activities are performed through the accomplishment of the current project plan.

SI.02. Software requirements are defined, analysed for correctness and testability, approved by the customer, baselined and communicated. Traceability between customer requirements and software requirements is established.

SI.03. 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.

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

SI.04. Software components defined by the design are produced. Unit tests are defined and performed to verify the consistency with requirements and the design. Traceability to the requirements and design are established.

SI.05. Software is produced performing integration of software components and verified using test cases and test procedures. Results are recorded in the test report. Defects are corrected and consistency and traceability to software design are established.

SI.06. A software product, that meets the requirements specification as agreed to with the customer, which includes user, operation and maintenance documentations is integrated, baselined and stored at the project repository. Needs for changes to the software product are detected and related change requests are initiated.

SI.07. Verification and validation tasks of all required work products are performed using the defined criteria to achieve consistency among output work products and input work products in each activity. Defects are identified and corrected; records are stored in the verification/validation records.

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

### 6.2.3 Overview of the agile software implementation process

[Figure 5](#) presents an overview of the agile software implementation process. The events not related to software implementation are greyed. Each event is described in detail in [6.3](#) to [6.9](#).



6.3.2 Overview of event 1- project vision meeting

Figure 6 shows event 1, the project vision meeting. The project vision meeting event receives the mission statement as an input and produces a reviewed mission statement, the prioritized product backlog and the number of sprints needed. It also selects the development team.

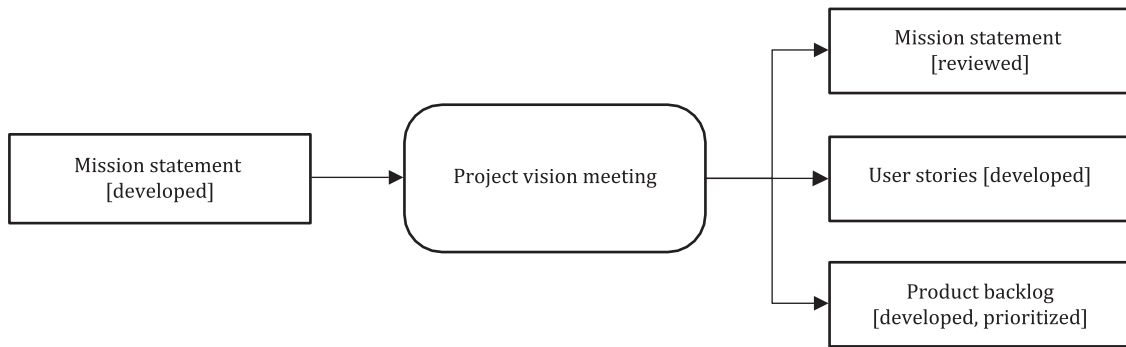


Figure 6 — Overview of event 1 - project vision meeting

6.3.3 Tasks of event 1 - project vision meeting

Figure 7 shows the tasks of event 1.

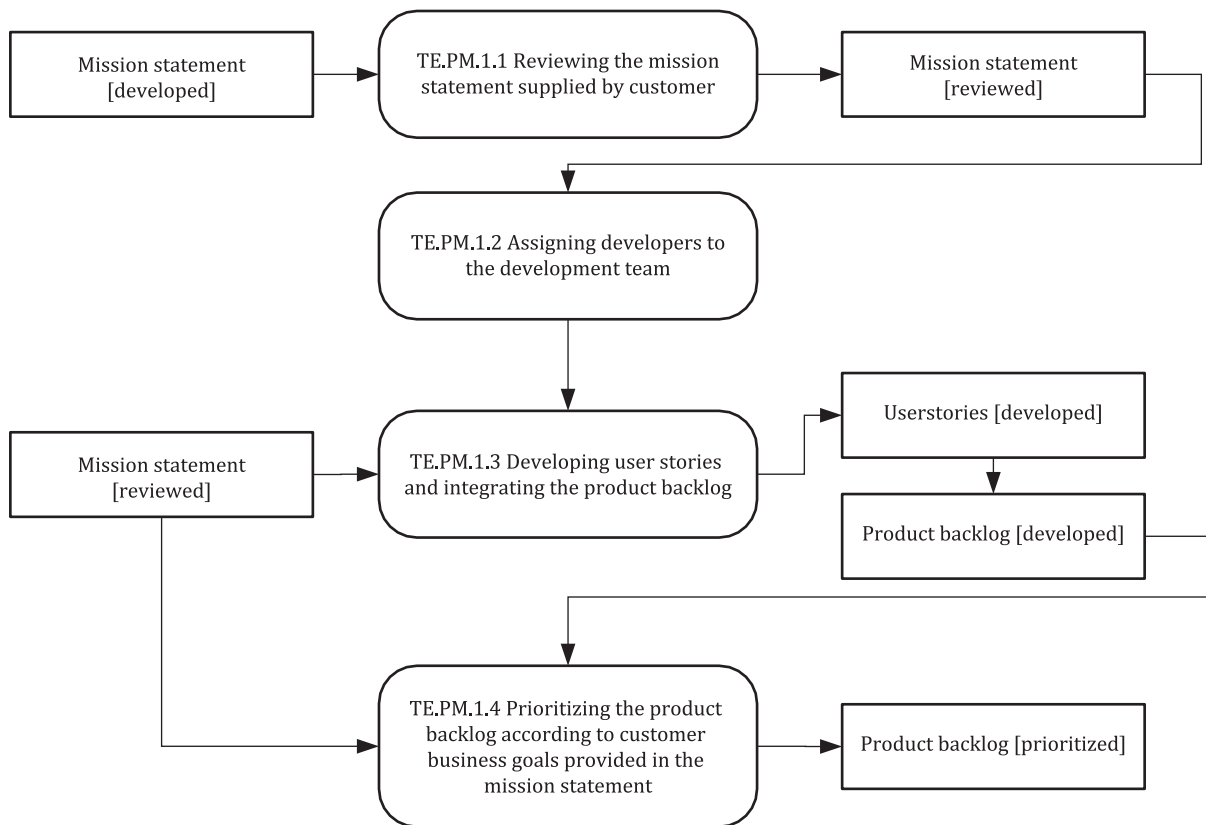


Figure 7 — Tasks of event 1 - project vision meeting

Table 1 provides the list of tasks of event 1.

Table 1 — Event 1 - project vision meeting task list

Role	Task list	Input work products	Output work products
CUS PO SL	TE.PM.1.1 Reviewing the mission statement supplied by the customer. 1) Using the mission statement, the developers and the product owner get an understanding of the requested product. 2) If the VSE has not documented its mission statement, a template is provided in <a href="#">Annex A</a> .	Mission statement [Developed]	Mission statement [Reviewed]
PO SL	TE.PM.1.2 Assigning developers to the development team. 1) During the project vision meeting the developers are integrated as a development team and the name of developers is listed in the project plan.	Mission statement [Reviewed]	Project plan: — Developers assigned [Reviewed]
CUS PO SL	TE.PM.1.3 Developing user stories and integrating the product backlog.	Mission statement [Reviewed]	User stories [Developed]  Product backlog [Developed]
CUS PO SL	TE.PM.1.4 Prioritizing the product backlog according to customer business goals provided in the mission statement. 1) Delivery instructions are reinforced by the definition of 'done', defined for each sprint backlog item.	Mission statement [Reviewed]  User stories [Developed]  Product backlog [Developed]	Product backlog [Prioritized]

6.4 Event 2 - estimation meeting (contributes to outcomes PM.01, PM.05)

6.4.1 Justification

An estimation meeting is time-boxed to a pre-agreed duration in which the members of the developers gain an understanding of the user stories listed in the product backlog and obtain a clearer picture of the overall product.

6.4.2 Overview of the event 2 - estimation meeting

Figure 8 shows event 2, the estimation meeting. The estimation meeting event receives the prioritized product backlog as input and produces an estimation and an update of the product backlog and an update (if necessary) of the user stories.

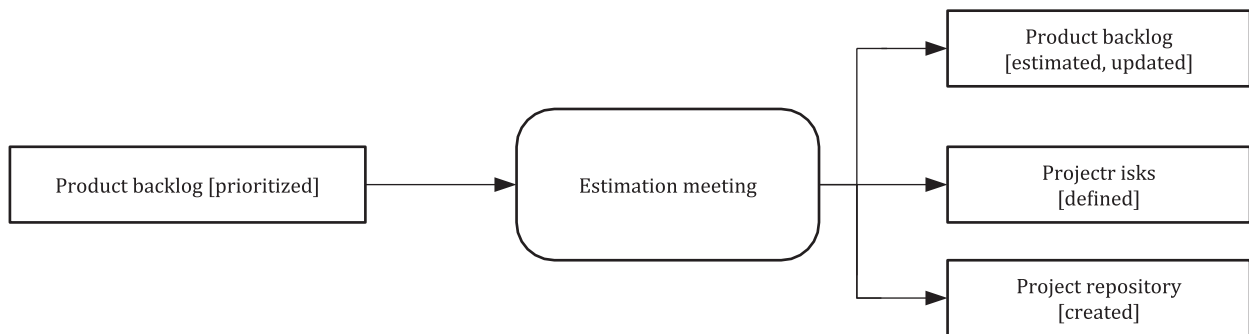


Figure 8 — Overview of event 2 - estimation meeting

6.4.3 Tasks of event 2 - estimation meeting

Figure 9 shows the tasks of event 2.

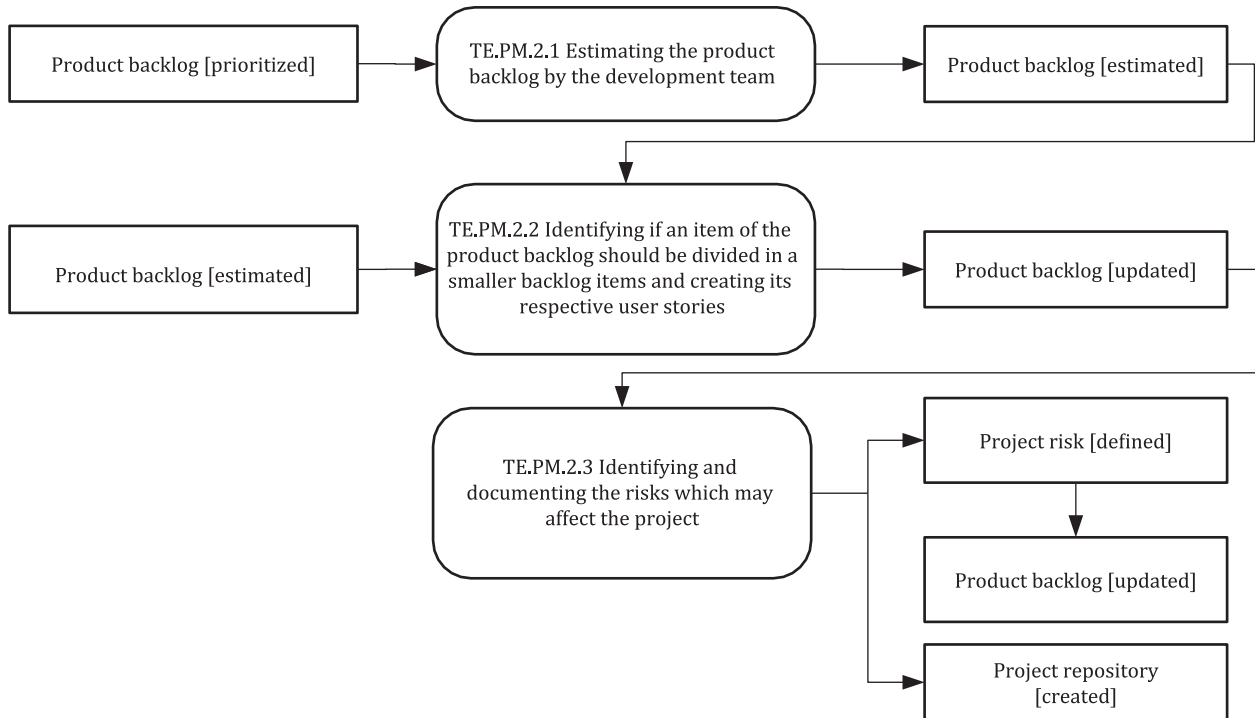


Figure 9 — Tasks of event 2 - estimation meeting

Table 2 provides the list of tasks of event 2.

Table 2 — Event 2 - estimation meeting task list

Role	Task list	Input work products	Output work products
CUS PO SL DT	<p>TE.PM.2.1 Estimating the product backlog by the developers.</p> <ol style="list-style-type: none"> <li>1) Estimating each item of the product backlog to get a picture of the overall effort.</li> <li>2) It is recommended a team-based estimation technique such as planning poker be used to estimate the product backlog.</li> <li>3) The estimation of effort can be in staff hours or in the size of tasks. The most used measure is the story points.</li> <li>4) Estimation should be done only to the level of fidelity currently required, the entire backlog does not should be estimated in detail. There will always be un-estimated or only roughly estimated items in a backlog.</li> </ol>	Product backlog [Prioritized]	Product backlog [Estimated]
CUS PO SL DT	<p>TE.PM.2.2 Identifying if an item of the product backlog should be divided in smaller backlog items and creating the user stories.</p> <ol style="list-style-type: none"> <li>1) The developers decide if a product backlog item is too large, in which case it is broken down into smaller backlog items. A user story for each one is developed as described in task TE.PM.2</li> <li>2) A list is made of the user stories that should be clarified by the product owner or the customer.</li> </ol>	Product backlog [Estimated]	Product backlog [Updated]
CUS PO SL DT	<p>TE.PM.2.3 Identifying and documenting the risks which may affect the project.</p> <ol style="list-style-type: none"> <li>1) Risk is a combination of the effect, or impact, of uncertainty on project objectives. An effect is a positive and/or negative deviation from what is expected.</li> <li>2) The common risks in agile projects are: <ul style="list-style-type: none"> <li>— Financial (e.g. budget)</li> <li>— Scope</li> <li>— Bad implementation of agile principles</li> <li>— Lack of change management</li> <li>— Lack of knowledge in the agile method</li> <li>— Technical debt</li> <li>— Agile is not suitable for a project</li> <li>— Security</li> </ul> </li> </ol>	Product backlog [Updated]	<p>Project risks [Defined]</p> <p>Product backlog [Updated]</p> <p>Project repository [Created]</p>

**6.5 Event 3 - sprint planning (contributes to outcomes PM.01, PM.02, PM.04, PM.07, SI.01, SI.03, SI.05)**

**6.5.1 Justification**

Sprint planning meeting is time-boxed at 8 h or less for one-month sprint. This determines the deliverables for the current sprint (selection of product backlog items).

### 6.5.2 Overview of event 3 – sprint planning

Figure 10 shows event 3, the sprint planning. The sprint planning event receives the updated version of the product backlog and the user stories developed by the developers as inputs.

Event 3 produces:

- the definition of a goal for the current sprint;
- the selection of the product backlog items to be developed in the current sprint;
- the reviewed and approved user stories, the definition of the user acceptance tests;
- the definition of the criteria that user stories should meet to be considered as “done”;
- the design of the functionality to be done in the current sprint.

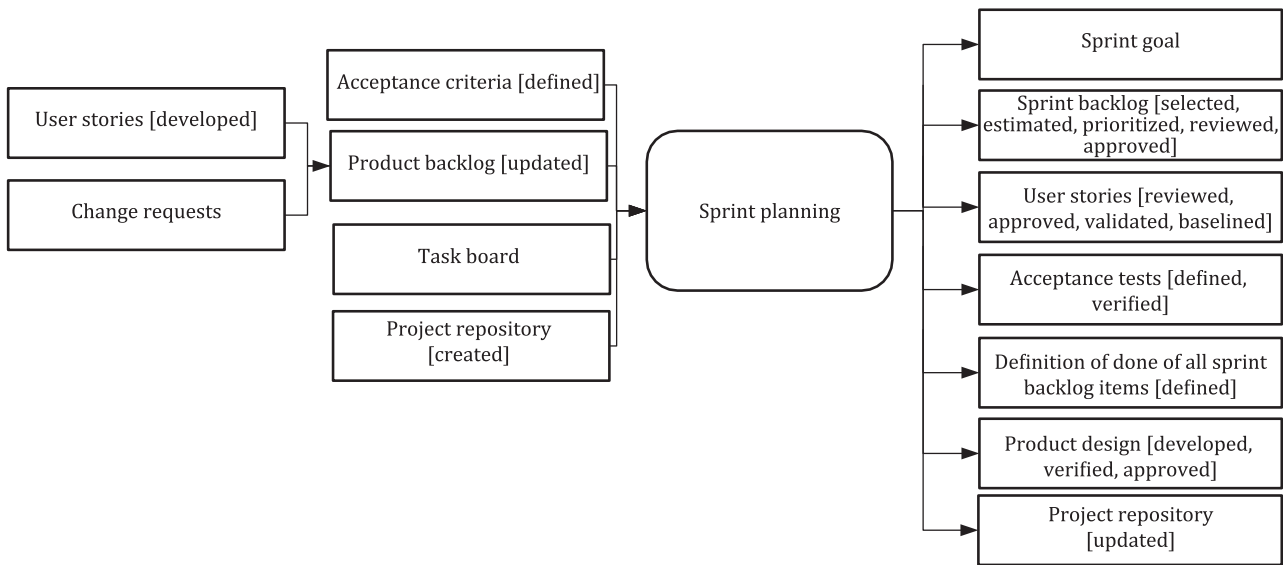


Figure 10 — Overview of event 3 - sprint planning

### 6.5.3 Tasks of event 3 – sprint planning

Figure 11 shows the tasks of event 3.

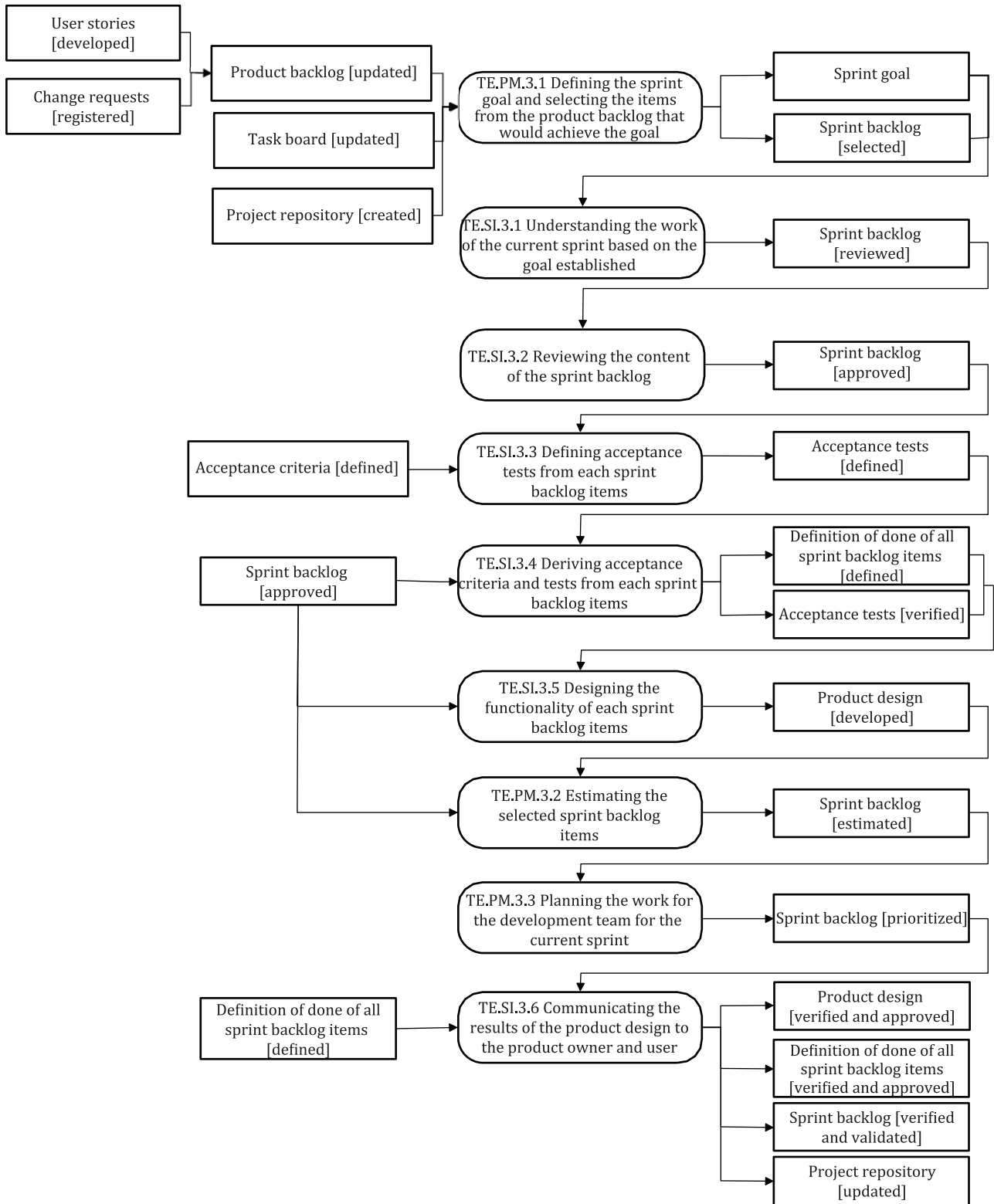


Figure 11 — Tasks of event 3 - sprint planning

Table 3 provides the list of tasks of event 3.

Table 3 — Event 3 - sprint planning task list

Role	Task list	Input work products	Output work products
CUS PO SL DT	TE.PM.3.1 Defining the sprint goal and selecting the items based on current priority or customer need.  1) The product owner discusses the objective to be achieved in the current sprint and the backlog items to be developed in the current sprint to achieve the objective.	Product backlog [Updated]  Task board [Updated]  Project repository [Baselined]	Sprint goal [Defined]  Sprint backlog [Selected]
CUS PO SL DT	TE.SI.3.1 Understanding the work of the current sprint based on the goal established.  1) The product owner can help to clarify the selected items for the sprint backlog.  2) If the developers consider there is too much or too little work in the sprint, it can be renegotiated with the product owner.  3) Team size may be renegotiated if goals and capacity are incompatible.	Sprint backlog [Selected]	Sprint backlog [Reviewed]
CUS PO SL DT	TE.SI.3.2 Reviewing the content of the Sprint backlog.  1) Those tasks are related to review of the sprint backlog.	Sprint backlog [Reviewed]	Sprint backlog [Approved]
CUS PO SL DT	TE.SI.3.3 Defining acceptance tests for each element of the Sprint backlog.  1) The developers with the scrum leader and the product owner write the user acceptance tests for each user story.  2) Constraints and acceptance criteria as well as sketches of the functionality, are established.  3) Together with the user acceptance tests, the developers have a test procedure defined.	Sprint backlog [Reviewed]  Acceptance criteria [Defined]	Acceptance tests [Defined]
CUS PO SL DT	TE.SI.3.4 Deriving acceptance criteria and tests from each element of the Sprint backlog.  1) Based on the user acceptance tests, the developers decide how the functionality of user stories will be built into “done” a product increment.  2) If tasks of <a href="#">Table 15</a> are not going to be implemented, the user acceptance test are approved in this task.	Sprint backlog [Approved]  Acceptance tests [Defined]	Definition of done [Defined]  Acceptance tests [Verified]
SL DT	TE.SI.3.5 Designing the functionality of each element of the product backlog.  1) Design the functionality (back-end and front-end) to transform each element of the product backlog in a working product increment.  2) The developers with the scrum leader discuss how each item of the sprint backlog can be implemented and designed (including interfaces, databases, architecture and components).	User stories [Reviewed and approved]  Definition of done [Defined]  User acceptance tests [Verified]	Product design [Developed]

Table 3 (continued)

Role	Task list	Input work products	Output work products
CUS PO SL DT	<p>TE.PM.3.2 Estimating the selected sprint backlog items.</p> <ol style="list-style-type: none"> <li>1) Estimate each item of the sprint backlog to be developed in the current sprint.</li> <li>2) It is possible to use an agile estimation technique such as planning poker. The effort can be estimated in staff hours or in size of tasks to be done or by the cost.</li> <li>3) While the agile estimation is performed, the following elements are identified: <ul style="list-style-type: none"> <li>— The material, equipment and tools are required to build each sprint backlog item.</li> <li>— The training required by the developers to perform the project.</li> </ul> </li> </ol>	<p>Sprint backlog [Approved]</p> <p>Product design [Developed]</p>	<p>Sprint backlog [Estimated]</p>
CUS PO SL DT	<p>TE.PM.3.3 Planning the work for the current sprint.</p> <ol style="list-style-type: none"> <li>1) Planning should be done based on the estimation of the all-backlog items and the established sprint goal.</li> <li>2) The work to be performed during the current sprint planned by the development team.</li> <li>3) The work to be performed during the current sprint is prioritized by the product owner.</li> </ol>	<p>Sprint backlog [Estimated]</p>	<p>Sprint backlog [Prioritized]</p>
CUS PO SL DT	<p>TE.SI.3.6 Communicating the results of the product design to the product owner and customer.</p> <ol style="list-style-type: none"> <li>1) The developers with the scrum leader register the agreements of the first meeting with the product owner and customer in a meeting record.</li> </ol>	<p>Sprint backlog [Prioritized]</p> <p>Definition of done [Defined]</p>	<p>Product design [Verified and approved]</p> <p>Definition of done [Verified and approved]</p> <p>Sprint backlog [Verified and validated]</p> <p>Project repository [Updated]</p>

**6.6 Event 4 – sprint (contributes to outcomes PM.02 PM.03, SI.03, SI.04, SI.05, SI.07)**

**6.6.1 Justification**

The sprint is a time-box in which the developers develop a done, usable and potentially releasable product.

**6.6.2 Overview of event 4 – sprint**

Figure 12 shows event 4, the sprint. The sprint event receives the user stories reviewed and approved, the definition of done established by the developers to determine if a user story is done, the product design verified and approved, and the user acceptance tests verified and approved as inputs.

Event 4 produces:

- the verified and approved unit tests and the software components that are ready to be released;

- the list of defects of the individual software components and of the overall increment;
- Unit tests that meet a minimum code coverage threshold have been executed and passed;
- the user acceptance test results;
- the increment of the current sprint.

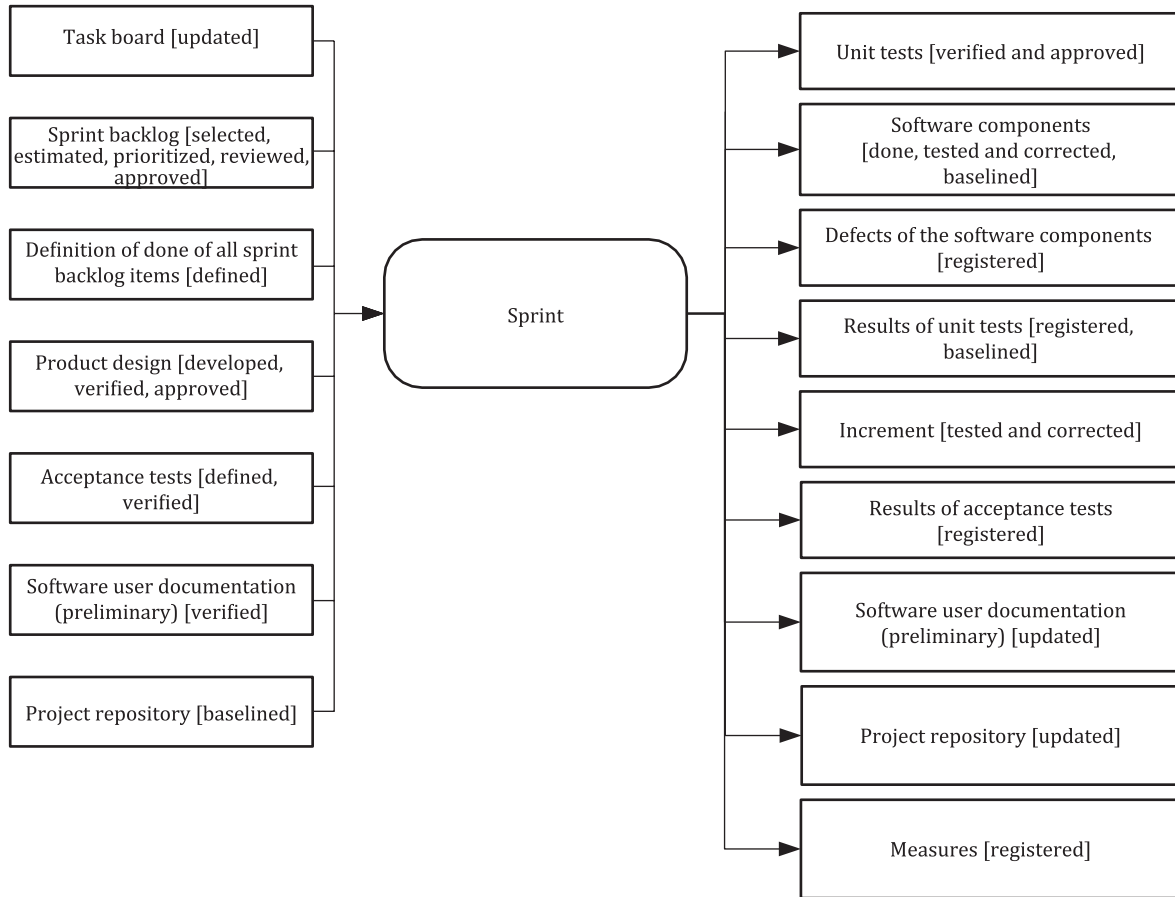


Figure 12 — Overview of event 4 - sprint

### 6.6.3 Tasks of event 4 - sprint

Figure 13 shows the tasks of event 4.

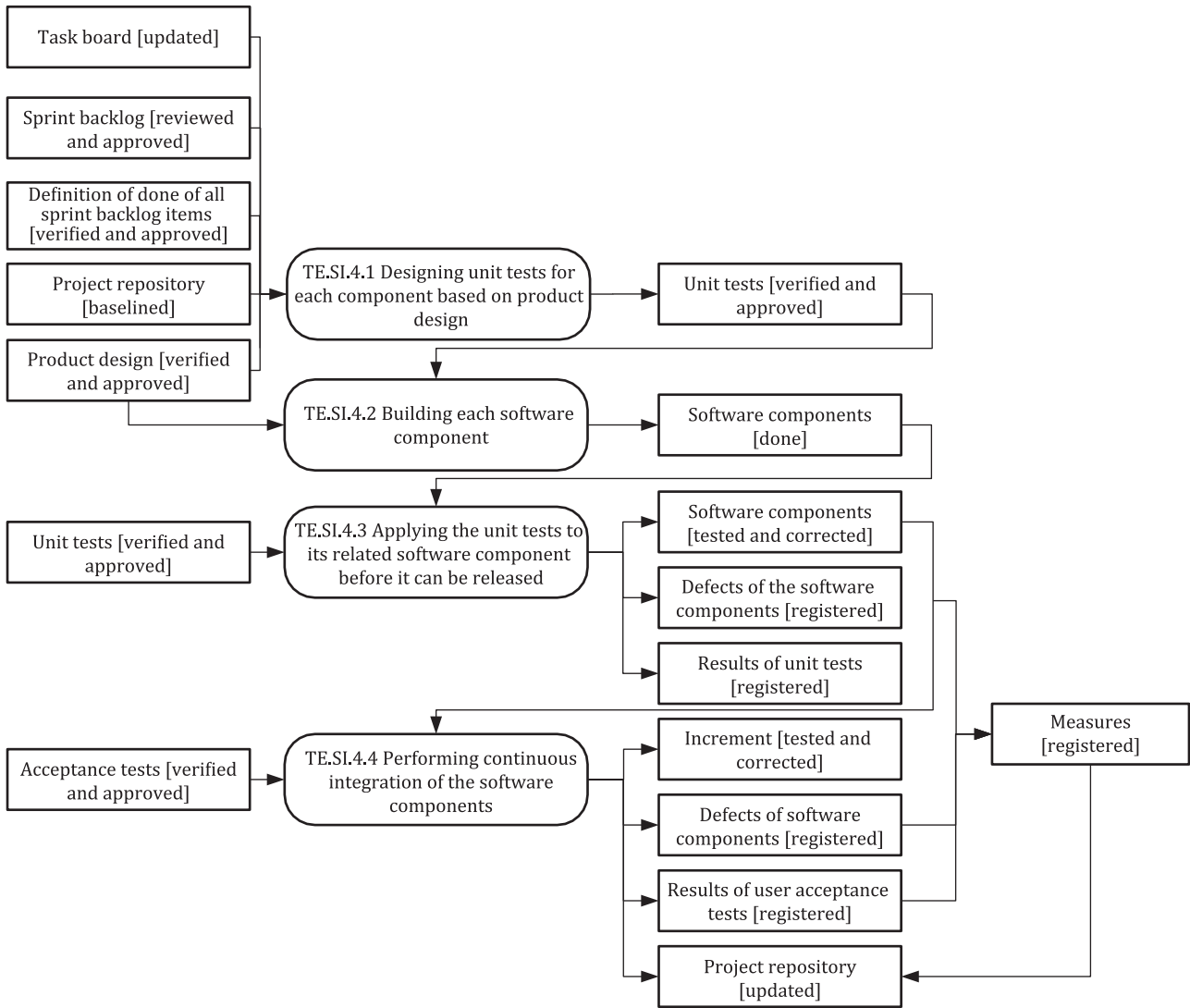


Figure 13 — Tasks of event 4 - sprint

Table 4 provides the list of tasks of event 4.

Table 4 — Event 4 – sprint task list

Role	Task list	Input work products	Output work products
DT	<p>TE.SI.4.1 Designing unit tests for each component of the Sprint backlog based on product design.</p> <ol style="list-style-type: none"> <li>1) Unit tests are designed and produced before software is developed.</li> <li>2) The definition of unit test is important because it verifies the component under test correctness.</li> <li>3) It is important to ensure the correctness of the unit tests by discussing its content and getting the agreement on it.</li> </ol>	<p>Task board [Updated]</p> <p>Sprint backlog [Reviewed and approved]</p> <p>Definition of done [Verified and approved]</p> <p>Product design [Verified and approved]</p> <p>Project repository [Baselined]</p>	<p>Unit tests [Verified and approved]</p>
DT	<p>TE.SI.4.2 Building each software component.</p> <ol style="list-style-type: none"> <li>1) Building each software component, consistent with the product design and the unit tests.</li> <li>2) Coding the software components using XP technique pair programming could be useful.</li> </ol>	<p>Product design [Verified and approved]</p> <p>Unit tests [Verified and approved]</p>	<p>Software components [Done]</p>
DT	<p>TE.SI.4.3 Applying the unit tests to its related software component before it may be released.</p> <ol style="list-style-type: none"> <li>1) When pair programming technique is performed, the code may be reviewed in real time, encouraging a higher level of quality than that achieved in a single inspection.</li> <li>2) If a defect is found, create unit tests that address the desired behaviour, correct the defect(s) and continue applying the unit tests until they are passed.</li> <li>3) More information about agile measures is provided in <a href="#">Annex D</a>.</li> </ol>	<p>Software component [Done]</p> <p>Unit tests [Verified and approved]</p>	<p>Software components [Tested and corrected]</p> <p>Defects of the software components [Registered]</p> <p>Results of unit tests [Registered]</p> <p>Measures [Registered]</p>

Table 4 (continued)

Role	Task list	Input work products	Output work products
DT	<p>TE.SI.4.4 Performing continuous integration of the software components.</p> <ol style="list-style-type: none"> <li>1) To ensure all the code works, the developer integrates software components as often and as early as possible.</li> <li>2) To integrate software components, it is possible to use the XP technique continuous integration by installing a continuous integration server (CI) that pulls the base code, builds and runs automated tests (to ensure the build is not broken). The report including the number of tests that passed and failed is produced.</li> <li>3) If a defect is found, register it as part of the results of the user acceptance test, correct the defect(s) and continue running it in the CI until the user acceptance tests are approved.</li> <li>4) Other tests, in addition to the user acceptance test, should be required such as integration or regression tests. If performed, ensure to register the tests and the test results.</li> </ol>	<p>Software component [Tested and corrected]</p> <p>User acceptance test [Verified and approved]</p>	<p>Increment [Tested and corrected]</p> <p>Defects of software component [Registered]</p> <p>Results of user acceptance tests [Registered]</p> <p>Project repository [Updated]</p> <p>Measures [Registered]</p>

6.7 Event 5 - daily scrum (contributes to outcomes PM.02, PM.03, PM.04, PM.05, PM.07)

6.7.1 Justification

The daily scrum is a 15 min time-box meeting in which the developer inspects the project progress towards the sprint goal and to inspect how progress is trending towards completing the work in the sprint backlog.

6.7.2 Overview of event 5 - daily scrum

Figure 14 shows event 5, the daily scrum. The daily scrum event receives the task board and user stories updated until the previous daily meeting, as inputs and produces an updated version of the: task board, burndown chart, project risks and sprint backlog.

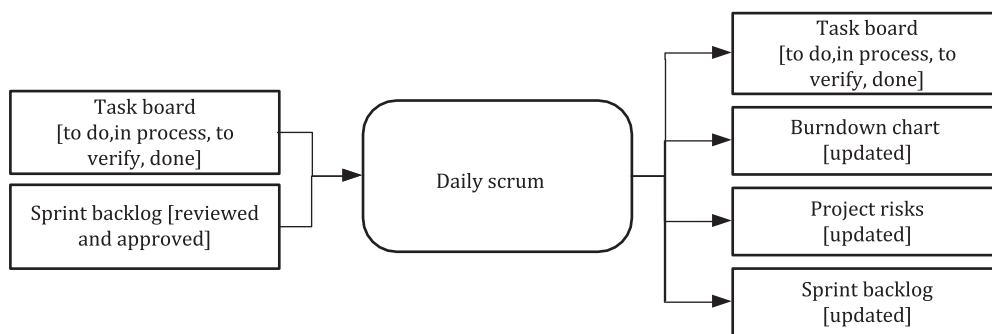


Figure 14 — Overview of event 5 - daily scrum

6.7.3 Tasks of event 5 - daily scrum

Figure 15 shows the tasks of event 5.

ISO/IEC 29110-5-4:2025(en)

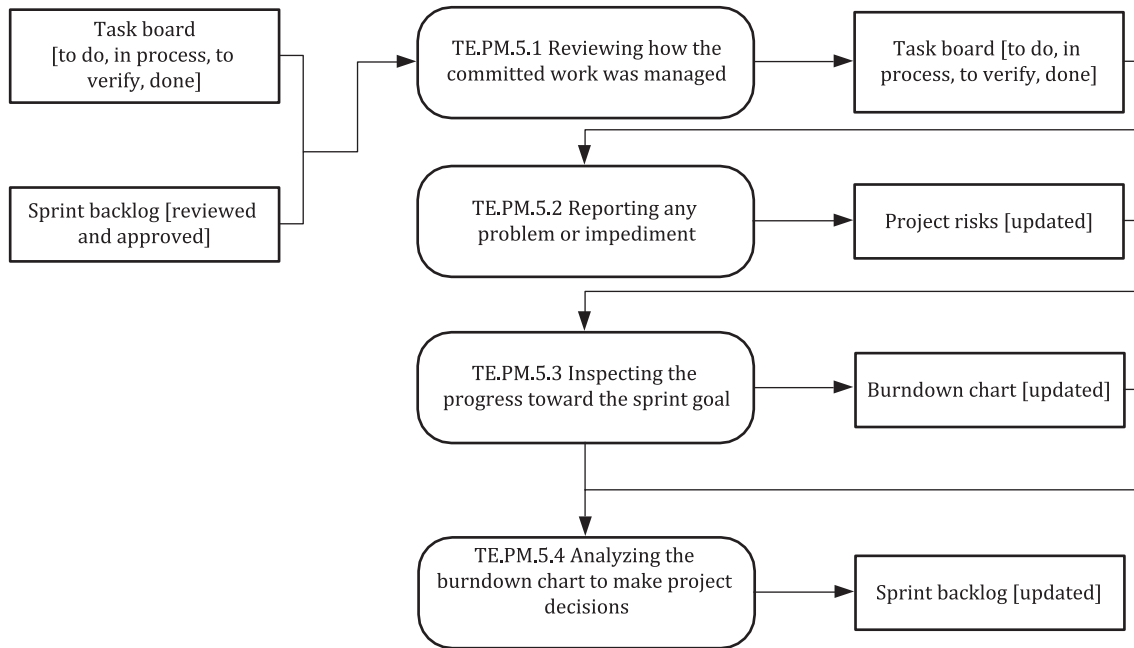


Figure 15 — Tasks of event 5 - daily scrum

Table 5 provides the list of tasks of event 5.

Table 5 — Event 5 - daily scrum task list

Role	Task list	Input work products	Output work products
DT SL	TE.PM.5.1 Reviewing how the committed work was managed. <ol style="list-style-type: none"> <li>1) Each member explains what they did regarding the committed work. The user stories are moved into the appropriate column of the task board.</li> <li>2) The state of the task board depends on the sprint progress:               <ul style="list-style-type: none"> <li>— “To do”: ready to start working on it.</li> <li>— “In process”: currently working on it.</li> <li>— “To verify”: ready to check the work done.</li> <li>— “Done”: software product (including documentation) is finished.</li> </ul> </li> </ol>	Task board [To do, In process, To verify, Done]  Sprint backlog [Reviewed and approved]	Task board [To do, In process, To verify, Done]
DT SL	TE.PM.5.2 Reporting any problem or impediment. <ol style="list-style-type: none"> <li>1) Report any impediment or detected problem that could impact on achieving the sprint goal.</li> <li>2) The scrum leader should take note to resolve any impediments</li> </ol>	Task board [To do, In process, To verify, Done]	Project risks [Updated]
DT SL	TE.PM.5.3 Inspecting the progress toward the sprint goal. <ol style="list-style-type: none"> <li>1) Another charts alternative to the burndown is the burnup chart or the product burnup to register the project progress.</li> </ol>	Project risks [Updated]	Burndown chart [Updated]
DT SL	TE.PM.5.4 Analysing the burndown chart to make project decisions. <ol style="list-style-type: none"> <li>1) The rest of the sprint work should be decided. It can be to adapt or re-plan the rest of the sprint.</li> </ol>	Burndown chart [Updated]	Sprint backlog [Updated]

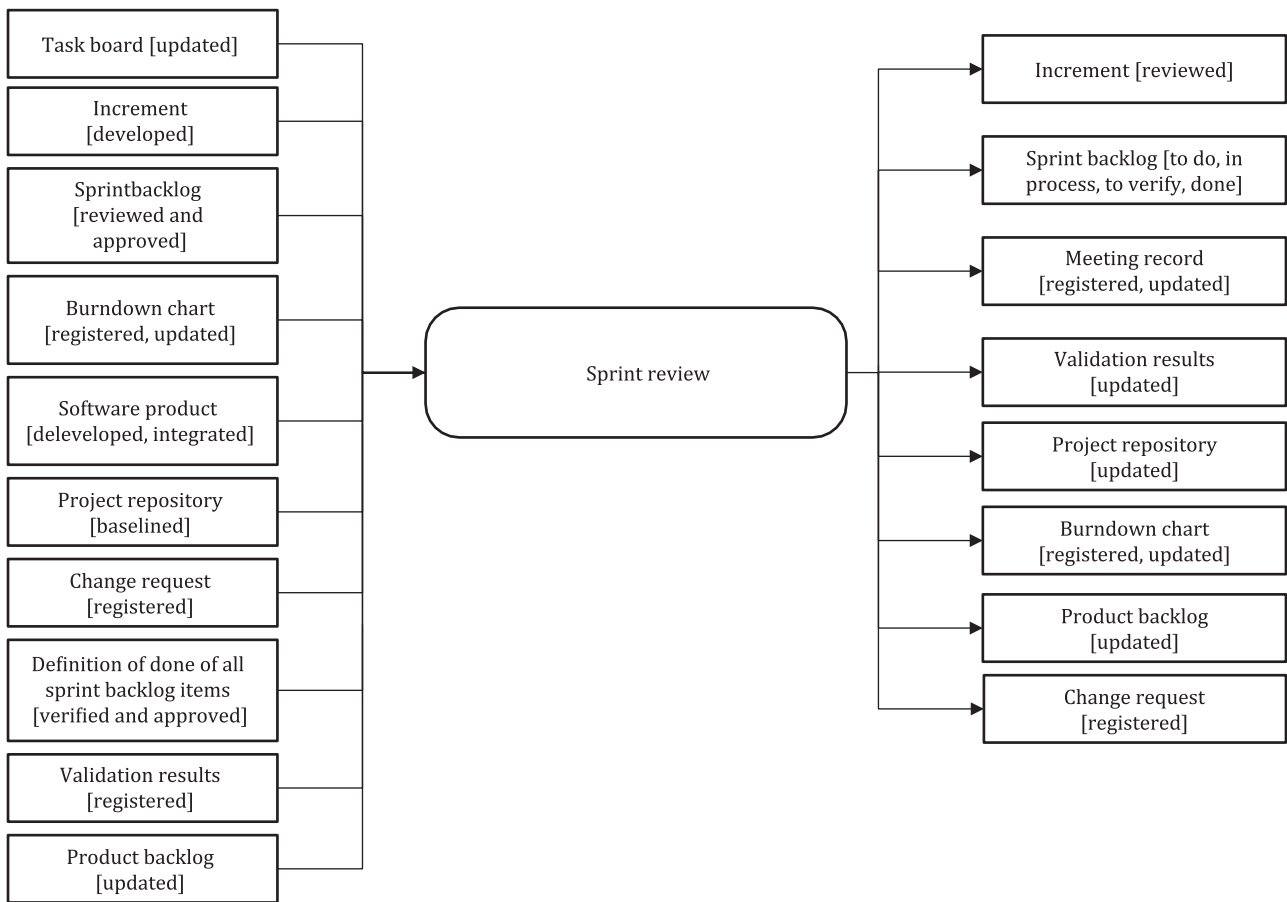
**6.8 Event 6 - sprint review (contributes to outcomes PM.02, PM.03, PM.04, SI.05, SI.07)**

**6.8.1 Justification**

A sprint review is held at the end of the sprint to inspect the increment and adapt the product backlog if required. The developer understands what they will be able to deliver to the customer or user in the sprint. Results of the past sprint are shown to the product owner.

**6.8.2 Overview of event 6 – sprint review**

Figure 16 shows event 6, the sprint review. The sprint review event receives the increment(s) delivered in the current sprint, the sprint backlog of the current sprint and the burndown chart of the current sprint as inputs and produces an update of the increment and the sprint backlog. The meeting record is documented.



**Figure 16 — Overview of event 6 - sprint review**

**6.8.3 Task of event 6 – sprint review**

Figure 17 shows the tasks of event 6.

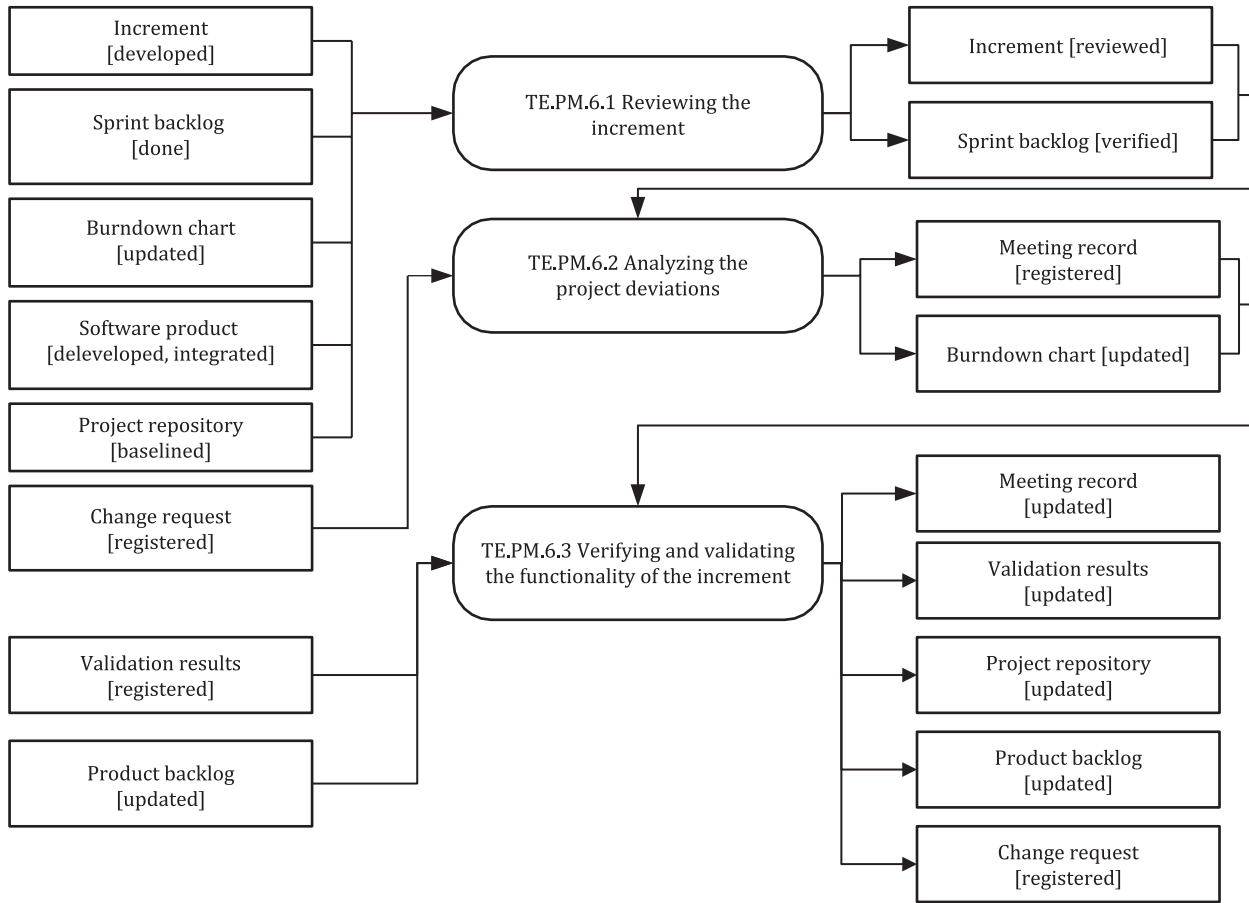


Figure 17 — Tasks of event 6 - sprint review

Table 6 provides the list of tasks of event 6.

Table 6 — Event 6 – sprint review task list

Role	Task list	Input work products	Output work products
DT PO SL	TE.PM.6.1 Reviewing the increment based on: <ul style="list-style-type: none"> <li>— A reminder of the definition of done.</li> <li>— Sprint backlog items that have been “done”.</li> <li>— Sprint backlog items that have not been “done”.</li> <li>— State of the sprint and product backlog.</li> </ul> 1) The state of the sprint backlog depends on the sprint progress: <ul style="list-style-type: none"> <li>— “Done”: software product (including documentation) is finished.</li> </ul>	Increment [Developed]  Sprint backlog [Done]  Burndown chart [Updated]  Software product [Developed]  Project repository [Baselined]  Software product [Integrated]  Definition of done of all sprint backlog items [Verified and approved]	Increment [Reviewed]  Sprint backlog [Verified]
DT PO SL	TE.PM.6.2 Analysing project deviations. <ol style="list-style-type: none"> <li>1) Analysing project deviations from the schedule and the budget.</li> <li>2) Analysing unexpected change requests and the overall performance of the sprint.</li> </ol>	Increment [Reviewed]  Sprint backlog [Updated]  Change Request [Registered]	Meeting record [Registered]  Burndown chart [Updated]
DT PO SL	TE.PM.6.3 Verifying and validating the functionality of the increment. <ol style="list-style-type: none"> <li>1) It is possible the developer requests the product owner to validate the increment and document feedback.</li> </ol>	Meeting record [Registered]  Validated results [Registered]	Meeting record [Updated]  Validated results [Updated]  Project repository [Updated]  Product backlog [Updated]  Change request [Registered]

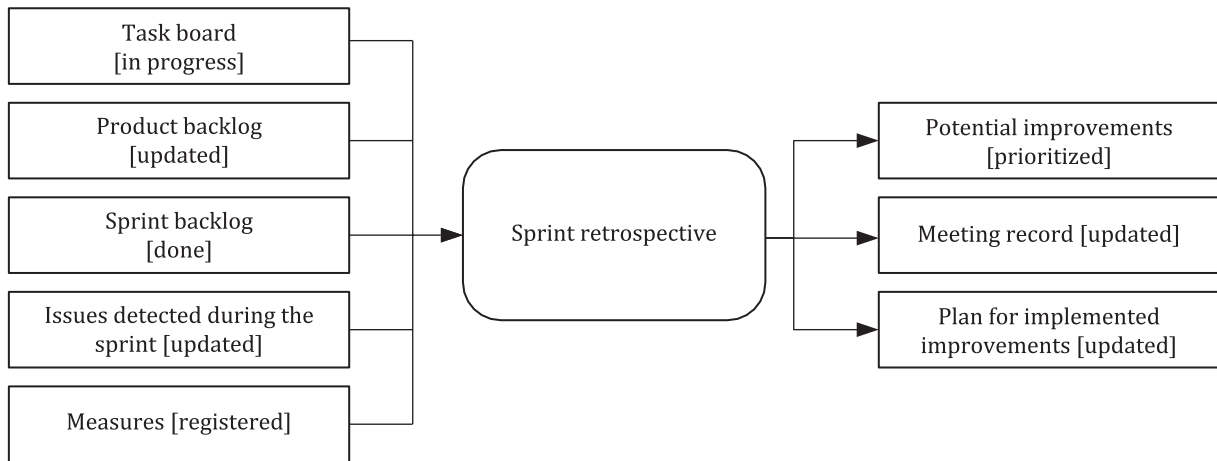
**6.9 Event 7 - sprint retrospective (contributes to outcomes PM.02, PM.03, PM.04, PM.07)**

**6.9.1 Justification**

The sprint retrospective is an opportunity for the scrum team to find what events and “things” the team is doing well, what events are continued, and what “more” can be done to improve the next sprint to be more productive.

**6.9.2 Overview of event 7 – sprint retrospective**

[Figure 18](#) shows event 7, the sprint retrospective.



**Figure 18 — Overview of event 7 - sprint retrospective**

**6.9.3 Task of event 7 – sprint retrospective**

[Figure 19](#) shows the tasks of event 7.

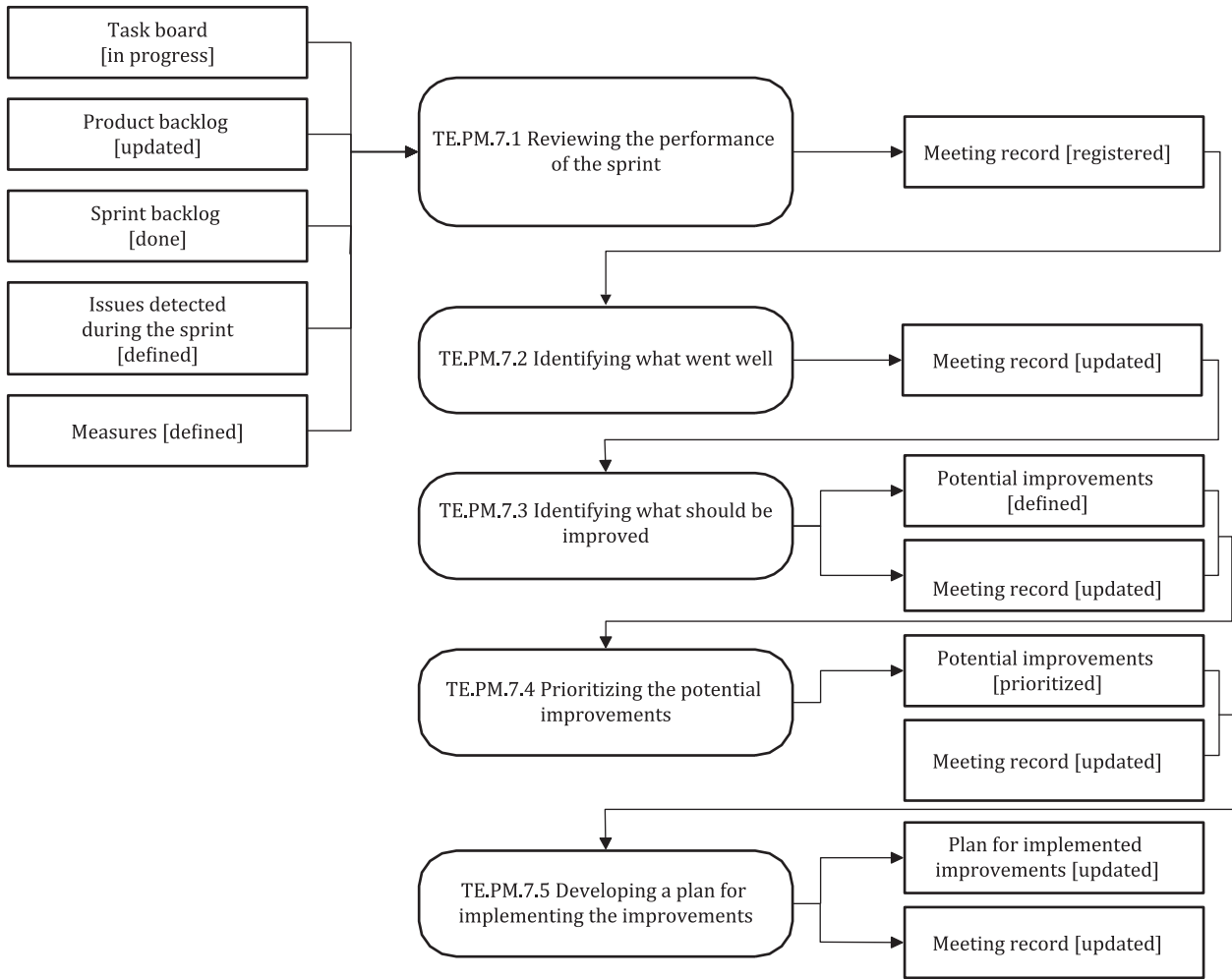


Figure 19 — Tasks of event 7 - sprint retrospective

Table 7 provides the list of tasks of event 7.

Table 7 — Event 7 - sprint retrospective list of tasks

Role	Task list	Input Work products	Output Work products
DT SL	TE.PM.7.1 Reviewing the performance of the sprint. 1) Reviewing the performance of the sprint focusing on people, relationships, process and tools 2) The scrum leader and developers participate together to analyse their accountability over the scrum process. 3) Team should account for changes in organizational or team practices and procedures	Task board [In progress]  Product backlog [Updated]  Sprint backlog [Done]  Issues detected during the sprint [Defined]  Measures [Defined]	Meeting record [Registered]
DT SL	TE. PM.7.2 Identifying what went well.	Task board [Updated]  Sprint backlog [Updated]	Meeting record [updated]
DT SL	TE.PM.7.3 Identifying what should be improved.	Task board [Updated]  Sprint backlog [Updated]	Potential improvements [Defined]  Meeting record [Updated]
DT SL	TE.PM.7.4 Prioritizing the potential improvements.	Potential improvements [Defined]	Potential improvements [Prioritized]  Meeting record [Updated]
DT SL	TE.PM.7.5 Developing a plan to implement the improvements.	Potential improvements [Prioritized]	Plan for implementing improvements [Developed]  Meeting record [Updated]

## 7 Role descriptions

Table 8 shows a list of the roles, abbreviations and suggested competency descriptions for the software Basic profile and the agile environment. There are some specific roles with more of one function.

For instance, the members of a project, which are working on the delivery of a releasable increment of a product at the end of a sprint, should perform, at specific moments of the project, the equivalent functions of analyst, designer and programmer of the software Basic profile.

Table 8 — Role descriptions

Software Basic profile		Agile environment		Knowledge and competency
Role	Abbreviation	Role	Abbreviation	
Analyst	AN	Developers	DT	<ul style="list-style-type: none"> <li>— Knowledge and experience eliciting, specifying and analysing the requirements.</li> <li>— Knowledge in designing user interfaces and ergonomic criteria.</li> <li>— Knowledge of the revision techniques.</li> <li>— Knowledge of the editing techniques.</li> <li>— Experience in software development and maintenance.</li> </ul>
Customer	CUS	Customer Product owner	CUS PO	<ul style="list-style-type: none"> <li>— Knowledge of the customer processes and ability to explain the customer requirements.</li> <li>— The customer, or the customer representative, has the authority to approve the requirements and their changes.</li> <li>— The customer, or the customer representative, participates in defining acceptance criteria before development begins to ensure software addressed the user's needs.</li> <li>— The customer includes user representatives to ensure that the operational environment is addressed.</li> <li>— Knowledge and experience in the application domain.</li> </ul>
Designer	DES	Developers	DT	<ul style="list-style-type: none"> <li>— Knowledge and experience in the software components and architecture design.</li> <li>— Knowledge of user experience design, user interface design and front-end awareness.</li> <li>— Knowledge and experience in the planning and performance of integration tests.</li> <li>— Experience in software development and maintenance.</li> </ul>

**Table 8** (continued)

Software Basic profile		Agile environment		Knowledge and competency
Role	Abbreviation	Role	Abbreviation	
Programmer	PR	Developers	DT	<ul style="list-style-type: none"> <li>— Knowledge and/or experience in programming, integration and unit tests.</li> <li>— Knowledge of the revision techniques.</li> <li>— Knowledge of the editing techniques.</li> <li>— Experience in software development and maintenance.</li> </ul>
Project manager	PJM	Product owner Scrum leader	PO SL	<ul style="list-style-type: none"> <li>— Leadership capability with experience making decisions, planning, personnel management, delegation and supervision, finances and software development.</li> <li>— With Scrum:                             <ul style="list-style-type: none"> <li>— PO is the sole person responsible for:                                     <ol style="list-style-type: none"> <li>1) collecting and understanding the software requirements;</li> <li>2) defining good user stories;</li> <li>3) ensuring the product backlog understanding;</li> <li>4) setting acceptance criteria for each user story;</li> <li>5) prioritizing and ordering the product backlog;</li> <li>6) defining the minimum viable product;</li> <li>7) define according together with the developer the definition of done;</li> <li>8) defining the releases according with the plan;</li> <li>9) validating the deliveries;</li> <li>10) making visible the content of the Product backlog items and</li> <li>11) cancelling the sprint if an extreme unforeseen event occurs.</li> </ol> </li> <li>— SL is responsible for:                                     <ol style="list-style-type: none"> <li>1) solving impediments that can arise during the sprint;</li> <li>2) ensuring there is a definition of done and;</li> <li>3) keeping the update of the burndown chart, together with the developer.</li> </ol> </li> </ul> </li> </ul>

**Table 8** (continued)

Software Basic profile		Agile environment		Knowledge and competency
Role	Abbreviation	Role	Abbreviation	
Technical leader	TL	Scrum leader	SL	<ul style="list-style-type: none"> <li>— Knowledge and experience in the software process domain.</li> <li>— With scrum:                             <ul style="list-style-type: none"> <li>— SL is responsible for: 1) ensuring the use of Scrum (monitoring its execution and promoting scrum courses if needed); 2) helping PO understanding agility; 3) helping PO to prioritize and management the product backlog; 4) helping developer to become a self-organizing and multi-functional team; 5) ensuring and promoting coding best practices and 6) helping to implement improvements detected during the sprint retrospective.</li> </ul> </li> </ul>
Work team	WT	Developers	DT	<ul style="list-style-type: none"> <li>— Knowledge and experience according to their roles on the project: TL, AN, DES, and/or PR.</li> <li>— Knowledge on the standards used by the customer and/or by the VSE.</li> </ul>

## 8 Work products description

This clause provides a list of states of work products could have and an alphabetical list of the input, output and internal process work products (WP).

[Table 9](#) provides an alphabetical list and definitions of the states of work products (WP).

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

**Table 9 — Definition of states of work products**

Name of state	Definition of state
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.
Cancelled	A planned work product or an element of a work product has been deleted from the project plan.
Completed	A work product is considered complete.
Corrected	Defect(s) identified in a work product has(ve) been removed.

**Table 9** (continued)

Name of state	Definition of state
Done	A work product is considered complete. This means, it has been built and prepared to be delivered, documented and tested in order words it meets the definition of done.
Deleted	A planned work product or an element of a work product has been deleted from the baseline.
Defined	A work product or a set of work products, that has defined by the DT.
Developed	A work product or a set of work products, that has been done by the DT.
Delivered	A work product or a set of work products, that has been approved by the customer or authorized stakeholders, has(ve) 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.
In process	A work product or a set of work products in which the DT is working on. It will take a certain amount of time to be done.
Postponed/Deferred	A work product or an element of a work product is planned for implementation in a later release.
Prioritized	A work product or an element of a work product that has been prioritized by the DT together with the PO.
Preliminary	A work product that has not yet been submitted for approval.
Published	A work product is stored in the repository and made available to all members of the project.
Rejected	A work product or an element of a work product that has not been approved.
Recovered	A work product stored in a repository was successfully restored.
Registered	A work product or a set of work products that has been entered or recorded.
Reviewed	A work product was presented to the authorized customer or stakeholder(s) as applicable for comment.
Selected	A work product or a set of work products to be done that has been selected by the DT.
Tested	A work product or a set of work products that has been evaluated to find defects applying a set of tests.
To do	A work product or a set of work products that has not been started.
To verify	A work product or set of work products that are evaluated (e.g. via test, review) to confirm, through the provision of objective evidence, that it fulfils its specified applicable requirements.
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 10 to 36 provide an alphabetical list of the input, output and internal process work products, their descriptions, possible states and the source of the work products for the Basic profile and the agile approach. The source of work products can be another process or an external entity to the project, such as the customer.

## ISO/IEC 29110-5-4:2025(en)

Work products are identified with a unique code, i.e. WP.XX for the software Basic profile or WPA.XX for the agile approach where XX is a sequential number. These codes have not been used in the tables of 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 paper document, as an item of an email, or as an item of an electronic tool.

**Table 10 — Acceptance record description**

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.01	Acceptance record <sup>a</sup>	Project management	WP.01	Acceptance record	Project management
<p><b>Description</b></p> <p>Documents the customer acceptance of the deliverables (i.e. list of work products to be delivered to customer) of the project.</p> <p>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 customer and project manager</li> </ul> <p>The applicable states are: initiated and signed.</p>					
<p>NOTE 1 The acceptance record includes the “acceptance criteria”.</p> <p>NOTE 2 The developer asks for clarification if required.</p>					
<p><sup>a</sup> There is no artefact in the agile methods used; a work product from the software Basic profile should be used.</p>					

**Table 11 — Burndown chart description**

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.02	Burndown chart	Sprint Sprint review	WP.08	Progress status record	Project management
<p><b>Description</b>                      Records the states of the project against the project plan.                      It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>— Status of actual tasks against planned tasks</li> <li>— Status of actual results against established objectives/goals</li> <li>— Status of actual resource allocation against planned resources</li> <li>— Status of actual cost against budget estimates</li> <li>— Status of actual time against planned schedule</li> <li>— Status of actual risk and its mitigation against previously identified</li> <li>— Records of any deviations from planned tasks and reason why</li> <li>— Signature of the project manager</li> </ul> <p>The applicable state is: published.</p>					
<p>NOTE 1 It records the state of the total effort against the amount of work for each sprint.                      NOTE 2 The scrum leader has a record of any deviations from planned tasks and reason why.                      NOTE 3 A template of the burndown chart WP is provided in <a href="#">Annex A</a>.                      NOTE 4 A template of this WP is provided in <a href="#">Annex A</a>.</p>					

**Table 12 — Change request description**

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.03	Change request <sup>a</sup>	Customer Sprint event	WP.03	Change request	Software implementation Customer Project management Work team
<p><b>Description</b>                      Identifies a software, 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 states</li> <li>— Identifies requester contact information</li> <li>— Impacted software</li> <li>— Impact to operations of the existing software defined</li> <li>— Impact to associated documentation defined</li> <li>— Criticality of the request, date required</li> </ul> <p>The applicable states are: initiated, evaluated, approved and rejected.</p>					
<p><sup>a</sup> There is no artefact in the agile methods used; a work product from the software Basic profile should be used.</p>					

**Table 13 — Defects of the software component description**

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.04	Defects of the software components	Sprint Daily scrum	WP.04	Correction register	Project management
<p><b>Description</b>                      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 person responsible for completion of defined actions</li> <li>— Identifies the open date and target closure date</li> <li>— Contains a state indicator</li> <li>— Indicates follow-up actions</li> </ul> <p>The applicable states are: initiated, approved and completed.</p>					

**Table 14 — Increment description**

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.05	Increment	Sprint	WP.11	Software	Software implementation
<p><b>Description</b>                      Software item (software source and executable code) for a customer, constituted by a collection of software release prototypes.                      The applicable states are: tested, corrected and baselined.</p>					

**Table 15 — Maintenance documentation description**

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.06	Maintenance documentation <sup>a</sup>	Sprint	WP.05	Maintenance documentation	Software implementation
<p><b>Description</b>                      Describes the design and functions of the software elements to assist maintainers in correcting defects, adapting, or enhancing the software product. Includes or refers to the infrastructure used in software development and available test suites.                      It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>— Includes or refers to all software product elements developed during implementation</li> <li>— Identifies the environment used for development and testing (compilers, design tools, construction and tests tools)</li> <li>— It is written in terms maintenance personnel can understand</li> </ul> <p>The applicable states are: verified and baselined.</p>					
<p><sup>a</sup> There is no artefact in the agile methods used; a work product from the software Basic profile should be used.</p>					

Table 16 — Meeting record description

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.07	Meeting record <sup>a</sup>	Project vision meeting Estimation meeting Sprint Sprint review Sprint retrospective Team feedback	WP.06	Meeting record	Project management
<p><b>Description</b>                      Records the agreements established with the customer and/or work team.                      It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>— Purpose of meeting</li> <li>— Name of 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> <li>— Signed by customer and project manager</li> </ul> <p>The applicable state is: approved.</p>					
<p>NOTE Meeting records provide an historical record that can be used at future meetings for verification of decisions, and as a reminder of past events and actions.</p> <p><sup>a</sup> There is no artefact in the agile methods used; a work product from the software Basic profile should be used.</p>					

Table 17 — Mission statement description

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.08	Mission statement	Customer	WP.02	Agreement	Customer
<p><b>Description</b>                      Description of work to be done related to software development.                      It may include the following characteristics:</p> <ul style="list-style-type: none"> <li>— Product description</li> <li>— Purpose</li> <li>— General customer requirements</li> <li>— Scope description of what is included and what is not included</li> <li>— Objectives of the project</li> <li>— List of work products to be delivered to customers (i.e., deliverables)</li> <li>— Signed by customer and project manager</li> </ul> <p>The applicable states are: approved, reviewed and baselined.</p> <p>NOTE 1 A template of this WP is provided in <a href="#">Annex A</a>.                      NOTE 2 This WP can also be provided in a web page or in an email.</p>					

Table 18 — Potential improvements description

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.09	Potential improvements	Sprint retrospective	WP.04	Correction register	Project management
<p><b>Description</b>                      Identifies improvements with regards to people, relationships, process and tools.                      It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>— Identifies and order the major items that went well</li> <li>— Identifies and order the potential improvements that may be implemented in next sprint</li> <li>— Plans the alternatives to improve the product quality by improving the process or adaptability of definition of done</li> </ul> <p>The applicable states are: initiated, approved and completed.</p>					

Table 19 — Product design description

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.10	Product design	Project vision meeting Sprint planning	WP.13	Software design	Software implementation
<p><b>Description</b></p> <p>Provides textual and graphical information on the software structure. This structure may include the following parts:</p> <ul style="list-style-type: none"> <li>— Architectural high-level software design</li> <li>— Describes the overall software structure: <ul style="list-style-type: none"> <li>— Identifies the required software components</li> <li>— Identifies the relationship between software components</li> </ul> </li> <li>— Consideration is given to any required: <ul style="list-style-type: none"> <li>— Software 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> <p>Low-level or detailed software design – includes details of the software components to facilitate its construction and test within the programming environment:</p> <ul style="list-style-type: none"> <li>— Provides detailed design (can 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 <ul style="list-style-type: none"> <li>— Provides the specifications of the program structure</li> </ul> </li> </ul> <p>The applicable states are: verified and baselined.</p> <p>NOTE 1 This WP provides value to the customer. NOTE 2 This document can be replaced with use cases, user stories or epics, model-based architecture, or other functional depiction.</p>					

Table 20 — Product operation guide description

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.11	Product operation guide <sup>a</sup>	Sprint	WP.07	Product operation guide	Software implementation
<p><b>Description</b></p> <p>Contains the necessary information to install and manage the software.            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> <li>— Certification and safety approvals</li> <li>— Warranty and replacement instructions</li> </ul> </li> </ul> <p>It is written in terms the personnel responsible for the operation can understand.            The applicable states are: verified and baselined.</p>					
<p>NOTE This work product provides support to the customer during the software installation.  <sup>a</sup> There is no artefact in the agile methods used; a work product from the software Basic profile should be used.</p>					

Table 21 — Project plan description

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.12	Project plan <sup>a</sup> including: — Product backlog description — Product backlog — Integration of developers — User stories — Sprint goal	Project vision meeting Estimation meeting Sprint planning	WP.09	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
- Deliverables - list of work products to be delivered to customer
- Tasks, including verification, validation and reviews with the customer and work team, to assure the quality of work products. Tasks may be represented as a work breakdown structure (WBS)
- Estimated duration of tasks
- Resources (humans, materials, standards, equipment and tools) including the required training, and the schedule when the resources are needed
- Composition of developers
- Schedule of the project tasks, the expected start and completion date for each task, and the relationship and dependencies of the estimated effort and cost
- Identification, prioritization and mitigation of project risks
- Verification criteria
  - e.g. maximum estimated number of defects left in a software unit for each software unit (e.g. specifications, architecture)

**Table 21** (continued)

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
	<ul style="list-style-type: none"> <li>— Measures                             <ul style="list-style-type: none"> <li>— set of measures is listed including the unit of measurement, measures are stored in the project repository</li> </ul> </li> <li>— Version control strategy:                             <ul style="list-style-type: none"> <li>— Work 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>— State of items and their modifications recorded and reported</li> <li>— Completeness and consistency of items ensured</li> <li>— Storage, handling and delivery (including archival and retrieval) mechanisms specified</li> </ul> </li> <li>— Delivery instructions                             <ul style="list-style-type: none"> <li>— Elements required for product release identified (e.g. hardware, software, documentation)</li> <li>— Delivery requirements</li> <li>— Sequential ordering of tasks to be performed</li> <li>— Applicable versions identified</li> <li>— Applicable media identified (e.g. paper, electronic)</li> <li>— Identifies all delivered software components and work products (e.g. user documentation) with version information</li> </ul> </li> <li>— Identifies any necessary backup and recovery procedures</li> </ul>				
Signed by customer and project manager					
The applicable states are: initiated, verified, approved, updated and reviewed.					
NOTE A template of the product backlog WP is provided in <a href="#">Annex A</a> .					
<sup>a</sup> There is no artefact in the agile methods used; a work product from the software Basic profile should be used.					

Table 22 — Project repository description

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.13	Project repository <sup>a</sup>	Estimation meeting	WP.21	Project repository	Project management
<p><b>Description</b>                      Electronic container to store project work products and deliverables.                      It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>— Stores project work products</li> <li>— Stores released deliverables work 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 state</li> <li>— Changes to work products are tracked to change requests</li> </ul> <p>The applicable states are: recovered and updated.</p>					
<p><sup>a</sup> There is no artefact in the agile methods used; a work product from the software Basic profile should be used.</p>					

Table 23 — Project repository backup description

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.14	Project repository backup <sup>a</sup>	Sprint planning	WP.22	Project repository backup	Project management
<p><b>Description</b>                      Repository used to back-up the project repository and, if necessary, to recover the information.                      The applicable state: recovered.</p>					
<p><sup>a</sup> There is no artefact in the agile methods used; a work product from the software Basic profile should be used.</p>					

Table 24 — Project risk description

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.15	Project risk	Sprint Sprint review	WP.08	Progress status record	Project management
<p><b>Description</b>                      Records the states of the project against the project plan.                      It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>— Status of actual tasks against planned tasks</li> <li>— Status of actual results against established objectives/goals</li> <li>— Status of actual resource allocation against planned resources</li> <li>— Status of actual cost against budget estimates</li> <li>— Status of actual time against planned schedule</li> <li>— Status of actual risk and its mitigation against previously identified</li> <li>— Records of any deviations from planned tasks and reason why</li> <li>— Signature of the project manager</li> </ul> <p>The applicable state is: published.</p> <p>Note 1 It records the state of the total effort against the amount of work for each sprint.                      Note 2 The scrum leader has a record of any deviations from planned tasks and reason why.                      Note 3 A template of the burndown chart WP is provided in <a href="#">Annex A</a>.                      Note 4 A template of this WP is provided in <a href="#">Annex A</a>.</p>					

Table 25 — Software components description

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.16	Software component	Sprint	WP.12	Software components	Software implementation
<p><b>Description</b>                      A set of related code units.                      The applicable states are: tested, corrected and baselined.</p>					

Table 26 — Software product description

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.17	Software product <sup>a</sup>	Sprint review	WP.14	Software product	Software implementation
<p><b>Description</b></p> <p>Provides a uniquely identified and consistent set of software products including:</p> <ul style="list-style-type: none"> <li>— Requirements specification</li> <li>— Software design</li> <li>— Traceability record</li> <li>— Software components</li> <li>— Software</li> <li>— Test cases and test procedures</li> <li>— Test report</li> <li>— Product operation guide</li> <li>— Software user documentation</li> <li>— Maintenance documentation</li> </ul> <p>The applicable states are: delivered and approved.</p>					
<p><sup>a</sup> There is no artefact in the agile methods used; a work product from the software Basic profile should be used.</p>					

Table 27 — Software user documentation description

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.18	Software user documentation <sup>a</sup>	Sprint	WP.15	Software user documentation	Software implementation
<p><b>Description</b></p> <p>Describe the way of using the software based on the user interface.                      It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>— User procedures for performing specific tasks using the software</li> <li>— Installation and de-installation procedures</li> <li>— Brief description of the intended use of the software (e.g. the concept of operation)</li> <li>— The supplied and required resources</li> <li>— Required operational environment</li> <li>— Availability of problem reporting and assistance</li> <li>— Procedures to access and exit the software</li> <li>— Lists and explains software 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.                      The applicable states are: preliminary, verified and baselined.</p> <p>NOTE This WP provides value to the customer.</p> <p><sup>a</sup> There is no artefact in the agile methods used; a work product from the software Basic profile should be used.</p>					

Table 28 — Sprint backlog description

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.19	Sprint backlog	Project vision meeting Estimation meeting Sprint planning	WP.09	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
- Deliverables - list of work products to be delivered to customer
- Tasks may be represented as a work breakdown structure (WBS). Tasks, including verification, validation and reviews with the customer and work team, to assure the quality of work products.
- 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 developers
- Schedule of the project tasks, the expected start and completion date for each task, and the relationship and dependencies of the estimated effort and cost
- Identification, prioritization and mitigation of project risks
- Verification criteria
  - e.g. maximum estimated number of defects left in a software unit for each software unit (e.g. specifications, architecture)

**Table 28** (continued)

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
	<ul style="list-style-type: none"> <li>— Measures                             <ul style="list-style-type: none"> <li>— set of measures is listed including the unit of measurement, measures are stored in the project repository</li> </ul> </li> <li>— Version control strategy:                             <ul style="list-style-type: none"> <li>— Work 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>— State of items and their modifications recorded and reported</li> <li>— Completeness and consistency of items ensured</li> <li>— Storage, handling and delivery (including archival and retrieval) mechanisms specified</li> </ul> </li> <li>— Delivery instructions                             <ul style="list-style-type: none"> <li>— Elements required for product release identified (e.g. hardware, software, documentation)</li> <li>— Delivery requirements</li> <li>— Sequential ordering of tasks to be performed</li> <li>— Applicable versions identified</li> <li>— Applicable media identified (e.g. paper, electronic)</li> <li>— Identifies all delivered software components and work products (e.g. user documentation) with version information</li> <li>— Identifies any necessary backup and recovery procedures</li> </ul> </li> </ul>				
Signed by customer and project manager					
The applicable states are: initiated, verified, approved, updated and reviewed.					
NOTE 1 A template of the product backlog WP is provided in <a href="#">Annex A</a> .					
NOTE 2 A template of the project plan WP is provided in <a href="#">Annex A</a> .					
<sup>a</sup> There is no artefact in the agile methods used; a work product from the software Basic profile should be used.					

Table 29 — Task board description

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.20	Task board	Sprint Sprint review	WP.08	Progress status record	Project management
<p><b>Description</b>                      Records the states of the project against the project plan.                      It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>— Status of actual tasks against planned tasks</li> <li>— Status of actual results against established objectives/goals</li> <li>— Status of actual resource allocation against planned resources</li> <li>— Status of actual cost against budget estimates</li> <li>— Status of actual time against planned schedule</li> <li>— Status of actual risk and its mitigation against previously identified</li> <li>— Records of any deviations from planned tasks and reason why</li> <li>— Signature of the project manager</li> </ul> <p>The applicable state is: published.</p> <p>NOTE 1 It records the state of the total effort against the amount of work for each iteration.                      NOTE 2 The scrum leader has a record of any deviations from planned tasks and reason why.                      NOTE 3 A template of the burndown chart WP is provided in <a href="#">Annex A</a>.                      NOTE 4 A template of this WP is provided in <a href="#">Annex A</a>.</p>					

Table 30 — Test report description

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.21	Test report <sup>a</sup>	Sprint review	WP.17	Test report	Software implementation
<p><b>Description</b>                      Documents the test execution                      It may include:</p> <ul style="list-style-type: none"> <li>— A summary of each defect</li> <li>— Identifies the related unit tests</li> <li>— Identifies the tester or 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 states are: initiated, [defects] corrected, baselined.</p> <p><sup>a</sup> There is no artefact in the agile methods used; a work product from the software Basic profile should be used.</p>					

**Table 31 — Traceability record description**

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.22	Traceability record <sup>a</sup>	Sprint	WP.18	Traceability record	Software implementation
<p><b>Description</b>  Documents the relationship between the requirements included in the requirements specification, software design elements, software components, test cases and test procedures.  It may include:</p> <ul style="list-style-type: none"> <li>— Identification of the individual requirements of the requirements specification to be traced</li> <li>— Provides forward and backward mapping of requirements to software design elements, software components, test cases, test procedures and test results</li> </ul> <p>The applicable states are: verified, baselined and updated.</p> <p>NOTE In an agile environment, it documents the relationship between the user stories, product design elements, software components, acceptance tests and unit tests.  <sup>a</sup> There is no artefact in the agile methods used; a work product from the software Basic profile should be used.</p>					

**Table 32 — User acceptance test description**

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.23	User acceptance test	Sprint	WP.16	Test cases and test procedures	Software implementation
<p><b>Description</b>  Provides elements needed to test code.  User acceptance tests</p> <ul style="list-style-type: none"> <li>— It is defined to ensure the quality of the working software that will be delivered at the end of the sprint.</li> <li>— It is derived from the acceptance criteria defined by customer and testers during the story generation. It ensures the handled required tasks are covered according to its specifications.</li> <li>— Tasks to perform user acceptance tests are: 1) identify and develop test scenarios, 2) define existing criteria, the approach for test cases and test sceneries and the timelines for testing; 3) prepare the data for testing; and 4) run and record the results.</li> <li>— It may include: ID, scope (including out of scope), assumptions and constraints, risks, team roles and responsibilities, entry criteria, base requirements for the test cases, test results and signatures.</li> </ul> <p>Unit tests may include:</p> <ul style="list-style-type: none"> <li>— Identifies the unit tests</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>					

**Table 32** (continued)

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
Test procedures may include: <ul style="list-style-type: none"> <li>— Integration approach</li> <li>— Integration tests</li> <li>— Regression tests</li> <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> The applicable states are: verified and baselined.					

Table 33 — User stories description

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.24	User stories	Estimation meeting	WP.10	Requirements specification	Software implementation
<p><b>Description</b></p> <p>Provides the software requirements. It may have the following characteristics:</p> <ul style="list-style-type: none"> <li>— Introduction – General description of software and its use within the context of the customer</li> <li>— Requirements description:</li> <li>— Functional suitability – Degree to which a product or system provides functions that meet stated and implied needs when used under specified conditions (e.g. functional completeness, functional correctness, functional appropriateness)</li> <li>— Performance efficiency - Performance relative to the amount of resources used under stated conditions (e.g. time behaviour, resource utilization, capacity)</li> <li>— Compatibility - Degree to which a product or component can exchange information with other products, systems or components, and/or perform its required functions, while sharing the same hardware or software environment (e.g. co-existence, interoperability)</li> <li>— 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)</li> <li>— 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)</li> <li>— Maintainability – Degree of effectiveness and efficiency with which a product can be modified by the intended maintainers (e.g. modularity, reusability, analysability, modifiability, testability)</li> <li>— 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)</li> <li>— Security - Degree to which a product protects information and data so 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)</li> <li>— Design and construction limitations or constraints – needs imposed by the customer</li> <li>— Legal and regulative – needs imposed by laws, regulations, etc.</li> <li>— Each requirement is identified (e.g. REQ #01), unique and it is verifiable or can be assessed.</li> </ul> <p>The applicable states are: verified, validated and baselined.</p>					
<p>NOTE 1 In an agile environment, user stories are the most common technique used to write requirements.</p> <p>NOTE 2 A template of this WP is provided in <a href="#">Annex A</a>.</p>					

Table 34 — Unit test description

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.25	Unit test	Sprint	WP.16	Test cases and test procedures	Software implementation
<p><b>Description</b></p> <p>Provides elements required to test code.</p> <p>User acceptance tests</p> <ul style="list-style-type: none"> <li>— It is defined to ensure the quality of the working software that will be delivered at the end of the sprint.</li> <li>— It is derived from the acceptance criteria defined by customers and testers during the story generation. It ensures the handled required tasks are covered according to its specifications.</li> <li>— Tasks to perform user acceptance tests are: 1) identify and develop test scenarios, 2) define existing criteria, the approach for test cases and test sceneries and the timelines for testing; 3) prepare the data for testing; and 4) run and record the results.</li> <li>— It may include: ID, scope (including out of scope), assumptions and constraints, risks, team roles and responsibilities, entry criteria, base requirements for the test cases, test results and signatures.</li> </ul> <p>Unit tests may include:</p> <ul style="list-style-type: none"> <li>— Identifies the unit tests</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>— Integration approach</li> <li>— Integration tests</li> <li>— Regression tests</li> <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 states are: verified and baselined.</p>					

Table 35 — Verification record description

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.26	Verification record <sup>a</sup>	Project vision meeting Sprint planning Sprint Sprint review	WP.19	Verification record	Project management Software implementa- tion
<p><b>Description</b> Documents the verification execution. It may include the record of:</p> <ul style="list-style-type: none"> <li>— Participants</li> <li>— Date</li> <li>— Place</li> <li>— Duration</li> <li>— Verification checklist</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> <li>— Signature of author</li> </ul> <p>The applicable states are: initiated, approved.</p>					
<p>NOTE In an agile environment, it is part of the quality practices providing value to the customer. <sup>a</sup> There is no artefact in the agile methods used; a work product from the software Basic profile should be used.</p>					

Table 36 — Validation record description

Agile approach			Basic profile		
WPA ID	Name	Source	WP ID	Name	Source
WPA.27	Validation record <sup>a</sup>	Sprint review	WP.20	Validation record	Software implementation
<p><b>Description</b>  Documents the validation execution.  It may include the record of:</p> <ul style="list-style-type: none"> <li>— Participants</li> <li>— Date</li> <li>— Place</li> <li>— Duration</li> <li>— Validation checklist</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> <li>— Signature of author</li> <li>— Signature of customer</li> </ul> <p>The applicable states are: initiated, approved.</p>					
<p>NOTE In an agile environment, it is part of the quality practices providing value to the customer.  <sup>a</sup> There is no artefact in the agile methods used; a work product from the software Basic profile should be used.</p>					

## Annex A (informative)

### Example of work products templates

#### A.1 Mission statement

A mission statement helps the developers to get a description of the work to be done, so that they get a better product vision.

[Table A.1](#) provides a template of a mission statement.

**Table A.1 — Mission statement template**

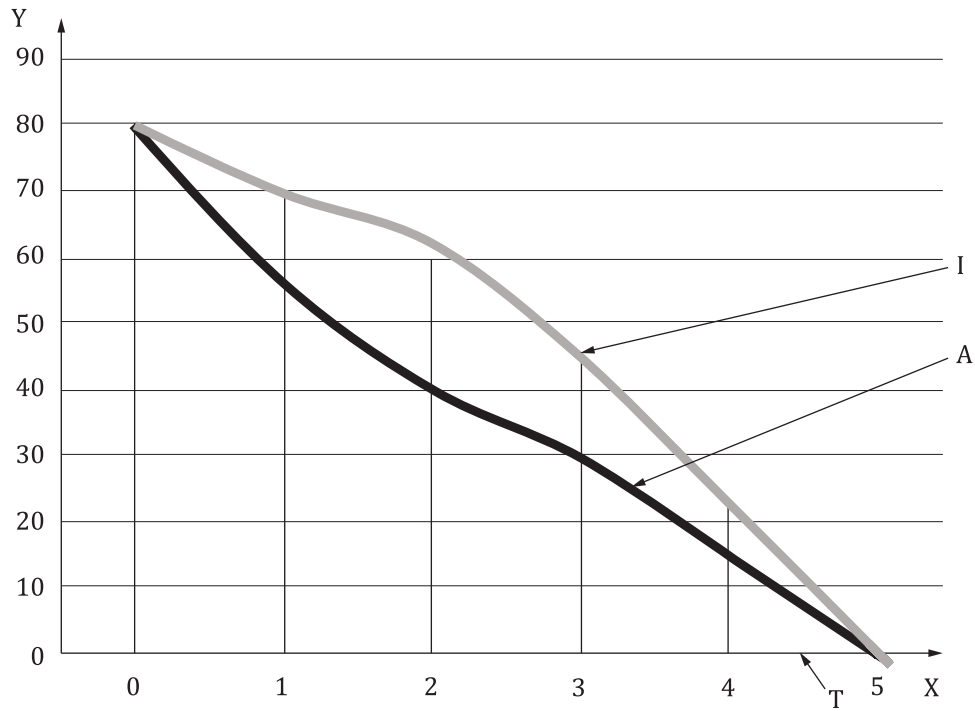
Mission statement		
<b>For whom:</b>	Customer	
<b>Where:</b>	Statement or a brief description of the need or the opportunity	
<b>What:</b>	Name of product and its category (e.g. document, software, type of product)	
<b>Why:</b>	Product goal	
<b>That should do:</b>	General functionalities of the product	
<b>Scope:</b>	What is included and what is not included in the project	
<b>To get:</b>	Work products to be delivered to the customer	
<b>State<sup>a</sup>:</b>	<b>Document version:</b>	<b>Reviewed by:</b>

<sup>a</sup> States: to do, in progress, completed, verified.

#### A.2 Burndown chart

A burndown chart is a visual tool for tracking the delivery of the expected solution within the desired schedule. It provides in a graphic way the remaining work to be completed and the remaining time units until the project deadline is achieved. When using a burndown chart, the total amount of work must be available. This quantity of work should not be modified during the timeline of the project.

[Figure A.1](#) shows the details of the product burndown.



**Key**

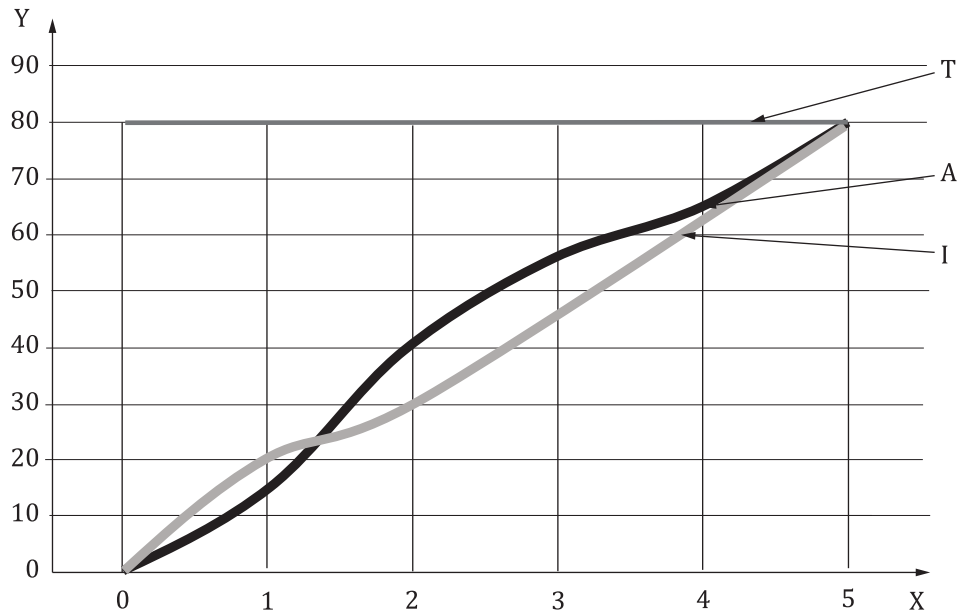
- A actual
- I ideal
- T target
- Y amount of work
- X time

**Figure A.1 — Simple burndown chart**

**A.3 Product burnup**

A product burnup is an alternative chart for tracking the delivery of the expected solution within the desired schedule. It is a visual tool that helps developers to show a comparison between completed work and the target achieved. It provides in a graphic way the cumulative sum of work that has been completed up to an actual iteration or point in time.

[Figure A.2](#) shows the details of the product burnup.



**Key**

- A actual
- I ideal
- T target
- Y amount of work
- X time

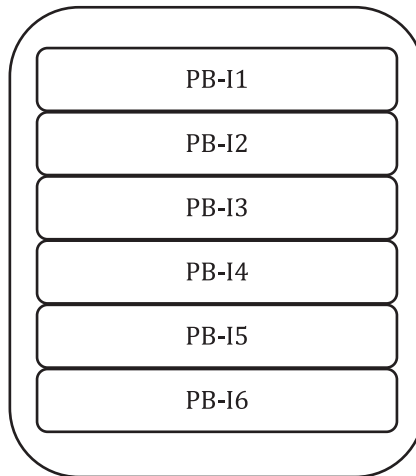
**Figure A.2 — Simple product burnup**

According to the official document of SCRUM (the Scrum Guide), the development team has the decision to use or not use one of them.

**A.4 Product backlog**

The product backlog is an ordered list which has the needs of the product, functions, requirements, improvements and repairs that constitute the changes made for future releases. It is a source of requirements for any changes made to the product. The product owner is responsible for the product backlog, as well as its content, availability and order.

[Figure A.3](#) shows an example of a product backlog list.



**Key**

- PB-I1 defect
- PB-I2 user story
- PB-I3 user case
- PB-I4 feature
- PB-I5 technical work
- PB-I6 free form text

**Figure A.3 — Simple Product Backlog**

[Table A.2](#) shows a template for a product backlog to be used in planning.

**Table A.2 — Template for a description of a product backlog to be used in a planning**

Product backlog						
Task name	Sprint	Priority <sup>a</sup>	State <sup>b</sup>	Story	Story points	Assigned to

<sup>a</sup> Priority: very high, high, normal, low.  
<sup>b</sup> States: to do, in progress, being tested, completed.

**A.5 Project plan**

A project plan presents how the project processes and activities will be executed to assure the project’s successful completion and the quality of the deliverable products. Links can be added to the tools which store data such as Trello, Gantt, or an Excel sheet.

Figure A.4 shows a template with the data that should be included in a simple project plan.

<p>Project plan [Project name]</p> <ol style="list-style-type: none"> <li>1. Product description; <ul style="list-style-type: none"> <li>— Purpose;</li> <li>— General customer requirements;</li> </ul> </li> <li>2. Scope description of what is included and what is not included;</li> <li>3. Objectives of the project;</li> <li>4. Deliverables - list of work products to be delivered to customer;</li> <li>5. Tasks, including verification, validation and reviews with the customer and work team, to assure the quality of work products. tasks may be represented as a work breakdown structure (WBS);</li> </ol>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

6.	Estimated duration of tasks;
7.	Resources (humans, materials, standards, equipment and tools) including the required training, and the schedule when the resources are needed;
8.	Composition of developers;
9.	Schedule of the project tasks, the expected start and completion date for each task, and the relationship and dependencies of the estimated effort and cost;
10.	Identification, prioritization and mitigation of project risks;
11.	Verification criteria <ul style="list-style-type: none"> <li>— e.g. maximum estimated number of defects left in a software unit for each software unit (e.g. specifications, architecture);</li> </ul>
12.	Measures <ul style="list-style-type: none"> <li>— set of measures is listed including the unit of measurement, measures are stored in the project repository;</li> </ul>
13.	Version control strategy: <ul style="list-style-type: none"> <li>— Work 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>— State of items and their modifications recorded and reported;</li> <li>— Completeness and consistency of items ensured;</li> <li>— Storage, handling and delivery (including archival and retrieval) mechanisms specified;</li> </ul>
14.	Delivery instructions; <ul style="list-style-type: none"> <li>— Elements required for product release identified (e.g. hardware, software, documentation);</li> <li>— Delivery requirements;</li> <li>— Sequential ordering of tasks to be performed;</li> <li>— Applicable versions identified;</li> <li>— Applicable media identified (e.g. paper, electronic)</li> <li>— Identifies all delivered software components and work products (e.g. user documentation) with version information;</li> <li>— Identifies any necessary backup and recovery procedures.</li> </ul>
State	Document version
Signed	
States: To do, in progress, completed, verified.	

Figure A.4 — Project plan template

## A.6 User stories

A user story contains enough information about the requirement to give the developers the ability to implement it. The user stories can be listed in a sprint backlog.

Figure A.5 shows a template of the user story.

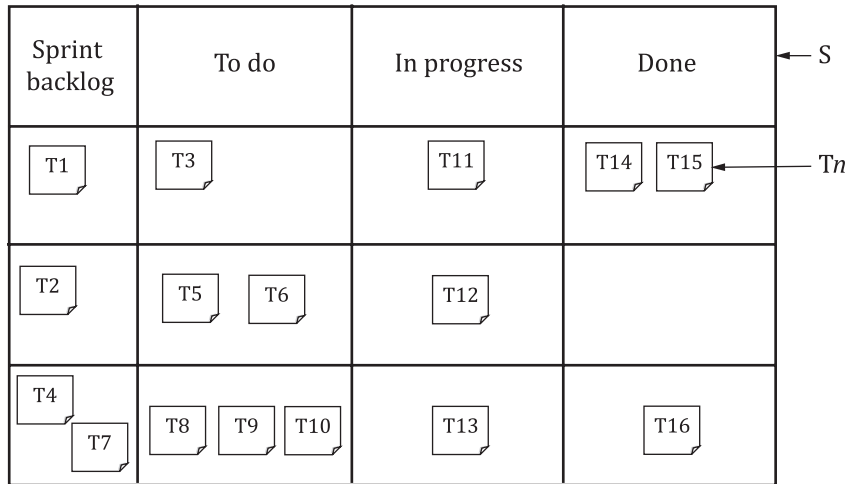
User story name:	Owner:	State:
Description:		
Estimated effort:	Priority:	
Acceptance criteria:		
NOTES:		

Figure A.5 — User story template

### A.7 Task board

In Scrum the task board is a visual display of the progress of the Scrum team during a sprint.

Figure A.6 presents a capture of the current sprint backlog, allowing everyone to review which tasks remain to be started, which are in progress, and which are completed.



**Key**

- S stage
- T<sub>n</sub> nth task

Figure A.6 — Task board template

## Annex B (informative)

### Overview of popular agile methods

#### B.1 Overview

According to the Agile Alliance,<sup>[17]</sup> being agile is the ability to create and respond to changes to succeed in an uncertain and turbulent environment. Agile software development aims to develop software faster, incrementally and produce satisfied customers. To achieve these goals, agile methods provide a conceptual framework of principles and practices. The agile methods define how to develop software using agile values and principles of the Agile Manifesto such as iterative development, frequent and early delivery of working software and simplicity. These methods emphasized a close collaboration between the developer and the stakeholders, frequent delivery and self-organizing teams <sup>[18]</sup>.

There are many agile methods described below. However, two of the most popular are Scrum and eXtreme Programming (XP).

#### B.2 Crystal methodologies

Crystal comprises a family of agile methodologies that focuses on principles such as people, interactions, community, skills, talent and communication, to deliver the best possible software development process and three main factors that characterize the project's team size, system criticality and project priorities<sup>[19][20]</sup>. Based on these principles, crystal offers 4 variants that should be implemented regarding the nature project needs regarding the validation and verification rules<sup>[21]</sup>:

- a) crystal clear: it is applied on teams of three to eight people;
- b) crystal yellow: it is applied on teams of 10 to 20 people;
- c) crystal orange: it is applied on teams of 20 to 50 people;
- d) crystal red: it is applied on teams to 50 to 100 people<sup>[23]</sup>.

Like other agile methods, this family promotes early, frequent delivery, high user involvement, adaptability, and the removal of bureaucracy or distractions <sup>[20]</sup>.

#### B.3 Dynamic software development method (DSDM)

Dynamic software development method is an agile method, which is based on nine key principles related to agile development process values that are focused on business needs/value, active user involvement, empowered teams, frequent deliveries, integrated testing, and the stakeholder collaboration.<sup>[19]</sup> The nine principles are focused on:

- a) the importance of the user involvement throughout the project development;
- b) the importance of the teams empowerment to make decisions;
- c) the importance of the frequent deliveries to detect errors;
- d) delivering good enough software to solve the business need and improving it in a latter interaction;
- e) it emphasizes the interactive and incremental development;
- f) all changes done during the software development should be reversible;

- g) The baseline requirements should be established at a level to limit the degree of freedom to which they can be altered during the development process;
- h) integrating testing throughout the development process;
- i) encouraging a collaborative and co-operative approach between technical and business staff [23].

This method requires an active involvement of users and gives team the power to make decisions[21]. Besides, the requirements are planned and delivered in short and fixed length time-boxes, named as iterations, and are prioritized using MoSCoW Rule [(M)must have requirements; (S) should have if at all possible; could have but not critical and (W) won't have this time, but potentially later][19].

## B.4 eXtreme programming (XP)

eXtreme Programming was designed to provide a framework that aims to produce higher quality software based on five values: simplicity, communication, feedback, courage and respect[22]. Besides, it focuses on a high quality of life for the Developer.

XP follows twelve engineering practices for software development so that it is considered the most specific of the agile methods. The engineering practices of XP are: planning games, small releases, customer acceptance tests, simple design, pair programming, test-driven development, refactoring, continuous integration, collective code overview, metaphor and sustainable pace [19][22].

## B.5 Feature driven development (FDD)

Feature driven development is an agile method focused on describing very specific model-driven and short two-week phases, in which a feature is designed and built[20]. The short phases include eight practices: domain object modelling, development focused on features, component or class ownership, feature teams, inspections, configuration management, regular builds and visibility of progress and results[19][20]. This method starts providing an overall model shape[19]. The execution of this method includes domain walkthrough, design, design inspection, code, code inspection, and promotion to build[19][20].

## B.6 Lean software development

This agile software development method is based on the principle of “just-in-time production” aiming on increasing the velocity to produce software while cost is decreased and delivering in small batches[19][20], by this way this method aims to ensure that the team members are productive the most time because it is concentrated in the current work and taking care of workflow dependencies[19]. Besides, it is based on seven main principles: eliminating waste, amplifying learning, deciding as late as possible, delivering as fast as possible, empowering the team, building integrity in and seeing the whole[19].

## B.7 Scrum

Scrum is one of the most popular agile methods that was designed to deliver meaningful value quickly throughout the project. According to the Scrum guide,[24][25] it has three features: lightweight, simple to understand and difficult to master.

Scrum provides a framework in which various processes and techniques can be implemented, so that it is possible to have a continuous improvement on the product, the team and the working environment [26].

Scrum strengths lie in the use of cross-functional, self-organized teams, and empowered people who divide their work into short and concentrated work cycles called sprints; this guarantees transparency in communication while creating an environment of collective responsibility and continuous progress.[27][28]

## Annex C (informative)

### Development of the guidelines for the reinforcement of an agile environment

A VSE implementing the Basic profile for software development of the ISO/IEC 29110 series should establish two processes: project management and software implementation. These processes are described in ISO/IEC 29110-5-1-2.

This document is based on two well-known agile methods, i.e. Scrum and eXtreme Programming.

[Table C.1](#) illustrates the correspondences between the software Basic profile of the ISO/IEC 29110 series and scrum. The correspondences were identified using the process elements description, listed in the left column, and using the process similarity study method.

**Table C.1 — Correspondence between elements of the software Basic profile and Scrum**

Process elements	SCRUM	Basic profile
<b>Process</b> A set of related activities to achieve a software project	1 process	2 processes
<b>Activity</b> 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 numbers and the activity name.	7 events containing a set of tasks to be done.	10 activities that containing a set of tasks to be done.
<b>Tasks</b> Tasks are key pieces of the process, then it can be defined as specific procedures to be performed at some point in the development. They usually require input artefacts as well as resources (technical and/or human), which can be associated through roles to generate another output artefact.	33 tasks for the scrum process	67 total of tasks, 26 for project management process and 41 for software implementation process
<b>Role</b> Who execute each process activity	3 roles	7 roles
<b>Artefact or work product</b> The work product that will be generated by each phase. They can be: a) outputs, they are products or services that result from a phase, or b) intermediate outputs, they become inputs of other phases of the process.	11 artefacts	22 work products

To develop this document, the following process and criteria were established.

- a) Process established to develop this document
  - A level of detail was defined for mapping scrum to a Basic profile of ISO/IEC 29110.
  - The elements to be analysed in the scrum and the software Basic profile of ISO/IEC 29110 were listed.
  - A template to perform the mapping was defined.
  - The elements of the software Basic profile of ISO/IEC 29110 were listed as a reference.

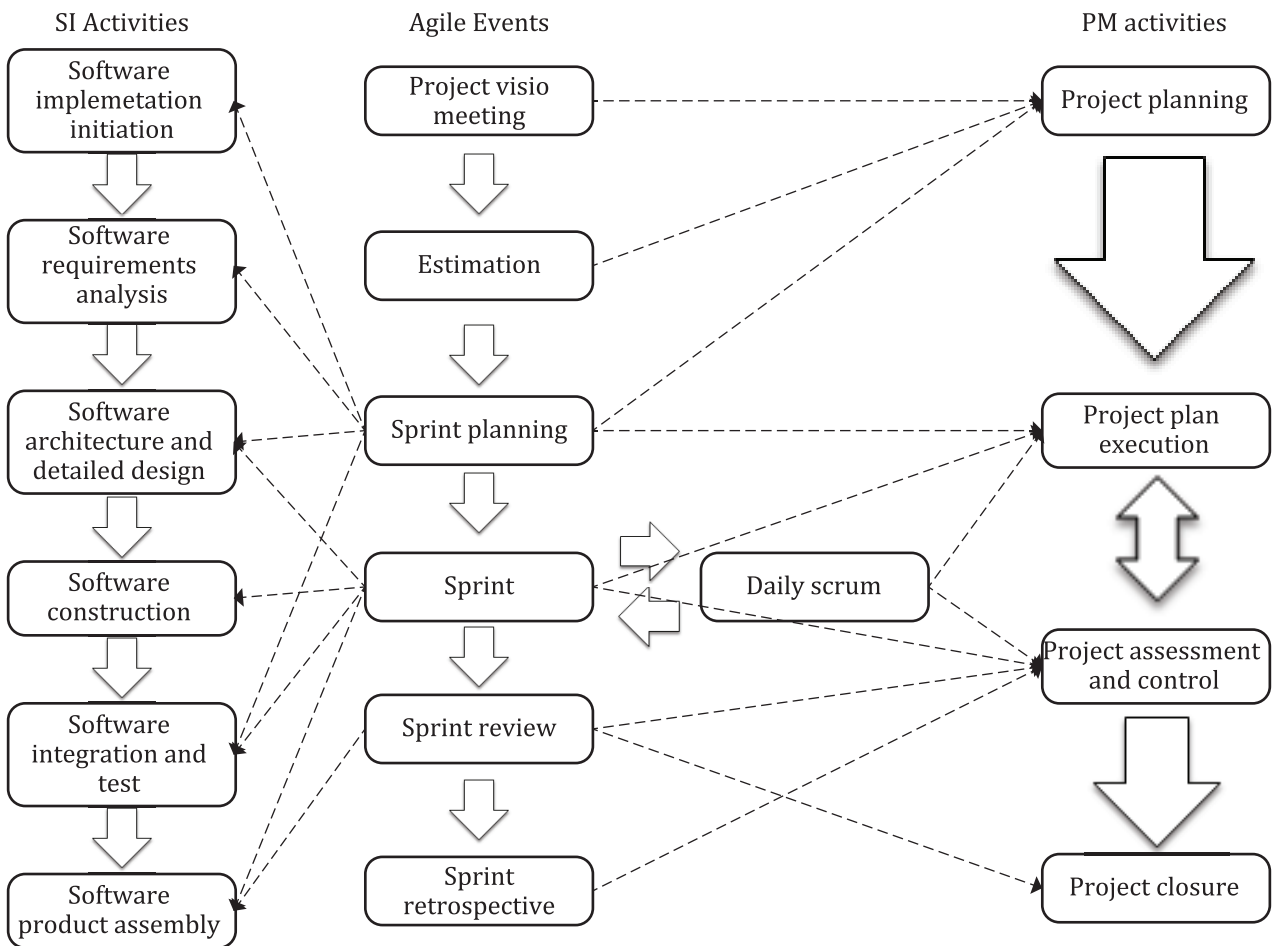
## ISO/IEC 29110-5-4:2025(en)

- It was identified how each practice of the software Basic profile of ISO/IEC 29110 is covered by performing an agile event, including input work products, tasks, roles and output work products [see procedure established in b)].
- The correspondence between the software Basic profile of ISO/IEC 29110 and Scrum and XP was established.

b) Procedure established to select a practice as part of this document.

- An event in scrum was identified, including input work products, tasks, involved roles and output work products, that helps to cover an ISO/IEC 29110 task of the software Basic profile;
- If there is no practice within an event of scrum that can cover the ISO/IEC 29110 task, an XP practice was identified that can help to cover an ISO/IEC 29110 task of the software Basic profile;
- If there is no XP practice that can cover the ISO/IEC 29110 task, the task proposed by a software Basic profile of ISO/IEC 29110 should be implemented.

Figure C.1 shows correspondence of the ISO/IEC 29110-5-4 agile events to the ISO/IEC 29110-5-1-2 Basic profile at activities/events level.



**Figure C.1 — ISO/IEC 29110-5-4 Agile Events correspondence to the activities of the Basic profile**

After performing this procedure, the results showed that more of the scrum practices were related to the project management process and more of the XP practices were related to the software implementation process. Therefore, this document took the events of SCRUM as a base to develop the guidelines, because it provides a good framework.

## ISO/IEC 29110-5-4:2025(en)

The following ground rules should be followed to use this document.

- A project should have a project repository created or established before its beginning or kick-off.
- The project closure should be formalized, and it is recommended to get the customer signature. It can be in an acceptance letter.
- All changes (requested by customers or identified by the developers) should be registered in a change request.
- A project should have documented the information necessary for the customer and the product maintainer for future product improvements.
- A project should have created a traceability matrix for improving the project monitoring and control of implemented requirements.

The 'definition of done' should be specified, such that it can be used as part of the delivery instructions, as well as to identify if a task is finished and should be moved to the baseline.

## Annex D (informative)

### Development of the guidelines for the reinforcement of an agile environment

#### D.1 General

This annex provides a list of measures commonly used in monitoring projects; in scrum, these measures should be collected during the sprint.

#### D.2 Productivity

- Lead time: The total time from when a task is requested until it is delivered to the user. Its objective is to minimize the delivery time of a functionality.
- Team velocity (agile velocity): The number of work units (story points, tasks, hours, completed user stories) completed in a sprint.

Velocity formula:

$$V = N_{wu} / N_s$$

where

$V$  is the velocity;

$N_{wu}$  is the total number of completed work units;

$N_s$  is the number of sprints.

- Work in progress (WIP): The number of tasks being worked on simultaneously. This measure is responsible for controlling the workload to prevent overload.

#### D.3 Software quality

- Percentage of code covered by tests: The proportion of code covered by unit or automated tests.

Coverage formula:

$$C = L_{exe} / L_t * 100$$

where

$C$  is the coverage;

$L_{exe}$  is the lines of code executed by tests;

$L_t$  is the total lines of code.

- Defect density: The number of defects found per unit of code or functionality.

Defect density formula:

$$D_d = N_d / (S_s * L)$$

where

- $D_d$  is the defect density;
- $N_d$  is the number of defects found;
- $S_s$  is the software size;
- $L$  is the lines of code or modules.

## Bibliography

- [1] World Bank, <https://www.worldbank.org/en/topic/smefinance>
- [2] 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](https://www.oecd-ilibrary.org/industry-and-services/oecd-sme-and-entrepreneurship-outlook-2019_34907e9c-en)
- [3] 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, Luxembourg, 2023, doi:10.2760/028705, JRC134336.
- [4] ISO/IEC/IEEE 12207:2017, *Systems and software engineering — Software life cycle processes*
- [5] ISO/IEC/IEEE 15288:2023, *Systems and software engineering — System life cycle processes*
- [6] ISO/IEC/IEEE 15289:2019, *Systems and software engineering — Content of life-cycle information items (documentation)*
- [7] ISO/IEC 33202:2024, *Software and systems engineering — Core agile practices*
- [8] ISO/IEC IEEE 26515, *Systems and software engineering — Developing information for users in an agile environment*
- [9] IEEE 828-2012, *Standard for Configuration Management in Systems and Software Engineering*
- [10] ISO/IEC/IEEE 26511:2018, *Systems and software engineering — Requirements for managers of information for users of systems, software, and services*
- [11] Project Management Institute. 2017. Agile practice guide. Project Management Institute
- [12] ISO/IEC/IEEE 32675:2022, *Information technology — DevOps — Building reliable and secure systems including application build, package and deployment*
- [13] ISO/IEC/IEEE 24774:2021, *Systems and software engineering — Life cycle management — Specification for process description*
- [14] ISO/IEC 23643:2020, *Software and systems engineering — Capabilities of software safety and security verification tools*
- [15] ISO/IEC/IEEE 24765:2017, *Systems and software engineering — Vocabulary*
- [16] ISO/IEC 29110-5-1-2:2025, *Systems and software engineering — Life cycle profiles for very small entities (VSEs) — Part 5-1-2: Software engineering guidelines for the generic Basic profile*
- [17] AGILE ALLIANCE. 2018. Agile glossary. <https://www.agilealliance.org/agile101/agile-glossary/>
- [18] MANIFESTO A. <https://agilemanifesto.org/>
- [19] COLLABNET. VersionOne (2020). What is Agile Methodology? Retrieved: <https://resources.collab.net/agile-101/agile-methodologies>
- [20] Guru99. (2020). Agile Methodology & Model: Guide for Software Development & Testing. Retrieved: <https://www.guru99.com/agile-scrum-extreme-testing.html>
- [21] Cockburn, Alistair. 2004. Crystal clear a human-powered methodology for small teams
- [22] LAMELAS A. (2018). Top 5 main Agile methodologies: advantages and disadvantages. Retrieved: <https://xpanse-it.com/en/blog/workplace-transformation/top-5-agile-methodologies/>
- [23] VOIGT B.J.J. "Dynamic System Development Method", January 2004, [online]. Available: [https://files.ifi.uzh.ch/rrerg/amadeus/teaching/seminars/seminar\\_ws0304/14\\_Voigt\\_DSMD\\_Ausarbeitung.pdf](https://files.ifi.uzh.ch/rrerg/amadeus/teaching/seminars/seminar_ws0304/14_Voigt_DSMD_Ausarbeitung.pdf).

- [24] SCHWABER K., SUTHERLAND J., BEEDLE M. (2020). The definitive guide to scrum: The rules of the game. Retrieved: <https://www.scrum.org/resources/blog/words-changed-scrum-guide-2020-update>
- [25] Blankenship J., Bussa M. & Millett S. (2011). Pro Agile. NET development with Scrum. Apress. ISBN 978-1-4302-3534-7 (eBook) pps. 372
- [26] Oktaba, H., & Ibarguengoitia González, G. (1998). Software process modeled with objects: Static view. *Computación y Sistemas*, 1(4), 228-238.
- [27] Tekin, Tekin, N., Yilmaz, M., & Clarke, P. (2021). A novel approach for visualization, monitoring, and control techniques for Scrum metric planning using the analytic hierarchy process. *Journal of Software: Evolution and Process*, e2420.
- [28] FEHLMANN T.M., KRANICH E. (2017, October). A new approach for continuously monitoring project deadlines in software development. In *Proceedings of the 27th International Workshop on Software Measurement and 12th International Conference on Software Process and Product Measurement* (pp. 161-169).
- [29] Calvo-Manzano, J.A., Cuevas, G., Muñoz, M., San Feliu, T. 2008. Process Similarity Study: Case Study on Project planning Practices Based on CMMI-DEV v1.2, EuroSPI 2008 Industrial Proceedings.
- [30] PHAM K.P., NEUMANN M. 2024. "How to Measure Performance in Agile Software Development? A Mixed-Method Study." 2024 50th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). IEEE.
- [31] RAM P., RODRIGUEZ P., OIVO M. 2018. "Software process measurement and related challenges in agile software development: A multiple case study." *Product-Focused Software Process Improvement: 19th International Conference, PROFES 2018, Wolfsburg, Germany, November 28–30, 2018, Proceedings 19*. Springer International Publishing.



**ICS 35.080**

Price based on 73 pages

© ISO/IEC 2025  
All rights reserved

**iso.org**