



German Association of the Automotive Industry
Quality Management Center

Joint Quality Management in the Supply Chain

Automotive SPICE[®]

Process Reference Model

Process Assessment Model

Version 4.1 (Preview)

Title:	Automotive SPICE [®] Process Assessment / Reference Model
Author(s):	VDA Working Group 13
Version:	4.1
Date:	2026-04-16
Status:	Preview

Copyright notice

This document is a revision of the Automotive SPICE® process assessment model and process reference model 4.1, which has been developed by the Working Group 13 of the Quality Management Center (QMC) in the German Association of the Automotive Industry.

This document reproduces relevant material from:

- **ISO/IEC 33020:2019**
Information technology -- Process assessment -- Process measurement framework for assessment of process capability

ISO/IEC 33020:2019 provides the following copyright release statement:

'Users of this International Standard may reproduce subclauses 5.2, 5.3, 5.4 and 5.6 as part of any process assessment model or maturity model so that it can be used for its intended purpose.'

- **ISO/IEC 15504-5:2006**
Information Technology – Process assessment – Part 5: An exemplar Process Assessment Model

ISO/IEC 15504-5:2006 provides the following copyright release statement:

'Users of this part of ISO/IEC 15504 may freely reproduce the detailed descriptions contained in the exemplar assessment model as part of any tool or other material to support the performance of process assessments, so that it can be used for its intended purpose.'

Relevant material from one of the mentioned standards is incorporated under the copyright release notice.

Acknowledgement

The VDA, the VDA QMC and the Project Group 13 explicitly acknowledge the high-quality work carried out by the members of the intacs® working groups. We would like to thank all involved people who have contributed to the development and publication of Automotive SPICE®.

Derivative works

You may not alter, transform, or build upon this work without the prior consent of the VDA Quality Management Center. Such consent may be given provided ISO copyright is not infringed.

The detailed descriptions contained in this document may be incorporated as part of any tool or other material to support the performance of process assessments, so that this process assessment model can be used for its intended purpose, provided that any such material is not offered for sale.

All distribution of derivative works shall be made at no cost to the recipient.

Document distribution

The Automotive SPICE® process assessment model may only be obtained by download from the www.vda-qmc.de web site. It is not permitted for the recipient to further distribute the document.

Change requests

Any problems or change requests should be reported through the defined mechanism at the www.vda-qmc.de web site.

Trademark notice

Automotive SPICE® is a registered trademark of the Verband der Automobilindustrie e.V. (VDA)

For further information about Automotive SPICE® visit www.vda-qmc.de.

Document history

Version	Date	By	Notes
2.0	2005-05-04	AutoSIG / SUG	DRAFT RELEASE, pending final editorial review
2.1	2005-06-24	AutoSIG / SUG	Editorial review comments implemented Updated to reflect changes in FDIS 15504-5
2.2	2005-08-21	AutoSIG / SUG	Final checks implemented: FORMAL RELEASE
2.3	2007-05-05	AutoSIG / SUG	Revision following CCB: FORMAL RELEASE
2.4	2008-08-01	AutoSIG / SUG	Revision following CCB: FORMAL RELEASE
2.5	2010-05-10	AutoSIG / SUG	Revision following CCB: FORMAL RELEASE
3.0	2015-07-16	VDA QMC WG13	Changes: See release notes
3.1	2017-11-01	VDA QMC WG13	Changes: See www.automotivespice.com
4.0	2023-10-20	VDA QMC WG13	Complete revision of PAM
4.1	2026-04-16	VDA QMC WG13	t.b.c.

Table of contents

Copyright notice	2
Acknowledgement.....	2
Derivative works.....	2
Document distribution	2
Change requests.....	2
Trademark notice	3
Document history	3
Table of contents	4
List of Figures	6
List of Tables	6
1. Introduction	7
1.1. Scope.....	7
1.2. Terminology.....	8
1.3. Abbreviations.....	11
2. Statement of compliance.....	12
3. Process capability determination.....	13
3.1. Process reference model.....	14
3.1.1. Primary processes category.....	14
3.1.2. Supporting processes category.....	16
3.1.3. Organizational processes category	16
3.2. Measurement framework.....	17
3.2.1. Process capability levels and process attributes	17
3.2.2. Process attribute rating	18
3.2.3. Rating and aggregation method.....	20
3.2.4. Process capability level model	22
3.3. Process assessment model.....	23
3.3.1. Assessment indicators	23
3.3.2. Understanding information Items and work products.....	24
3.3.3. Understanding the level of abstraction of a PAM.....	25
3.3.4. Why a PRM and PAM are not a lifecycle model or development process blueprint	26
4. Process reference model and performance indicators (Level 1).....	28
4.1. Acquisition process group (ACQ).....	29
4.1.1. ACQ.4 Supplier Monitoring	29
4.2. Supply process group (SPL).....	30
4.2.1. SPL.2 Product Release.....	30
4.3. System Engineering process group (SYS).....	33
4.3.1. SYS.1 Requirements Elicitation	33
4.3.2. SYS.2 System Requirements Analysis.....	35
4.3.3. SYS.3 System Architectural Design	37
4.3.4. SYS.4 System Integration and Integration Verification	39
4.3.5. SYS.5 System Verification	42
4.4. Software Engineering process group (SWE).....	44

4.4.1.	SWE.1 Software Requirements Analysis	44
4.4.2.	SWE.2 Software Architectural Design	46
4.4.3.	SWE.3 Software Detailed Design and Unit Construction	48
4.4.4.	SWE.4 Software Unit Verification	50
4.4.5.	SWE.5 Software Component Verification and Integration Verification	52
4.4.6.	SWE.6 Software Verification	55
4.5.	Validation process group (VAL)	58
4.5.1.	VAL.1 Validation	58
4.6.	Machine Learning Engineering process group (MLE)	61
4.6.1.	MLE.1 Machine Learning Requirements Analysis	61
4.6.2.	MLE.2 Machine Learning Architecture	63
4.6.3.	MLE.3 Machine Learning Training	65
4.6.4.	MLE.4 Machine Learning Model Testing	67
4.7.	Hardware Engineering process group (HWE)	70
4.7.1.	HWE.1 Hardware Requirements Analysis	70
4.7.2.	HWE.2 Hardware Design	72
4.7.3.	HWE.3 Verification against Hardware Design	74
4.7.4.	HWE.4 Verification against Hardware Requirements	76
4.8.	Supporting process group (SUP)	79
4.8.1.	SUP.1 Quality Assurance	79
4.8.2.	SUP.8 Configuration Management	81
4.8.3.	SUP.9 Problem Resolution Management	83
4.8.4.	SUP.10 Change Request Management	85
4.8.5.	SUP.11 Machine Learning Data Management	87
4.9.	Management process group (MAN)	89
4.9.1.	MAN.3 Project Management	89
4.9.2.	MAN.5 Risk Management	91
4.9.3.	MAN.6 Measurement	93
4.10.	Process Improvement process group (PIM)	95
4.10.1.	PIM.3 Process Improvement	95
4.11.	Reuse process group (REU)	97
4.11.1.	REU.2 Reuse of Products	97
5.	Process capability levels and process attributes	99
5.1.	Process capability level 0: Incomplete process	99
5.2.	Process capability Level 1: Performed process	100
5.2.1.	PA 1.1 Process performance process attribute	100
5.3.	Process capability Level 2: Managed process	100
5.3.1.	PA 2.1 Process performance management process attribute	101
5.3.2.	PA 2.2 Work product management process attribute	104
5.4.	Process capability Level 3: Established process	106
5.4.1.	PA 3.1 Process definition process attribute	106
5.4.2.	PA 3.2 Process deployment process attribute	108
5.5.	Process capability Level 4: Predictable process	110
5.5.1.	PA 4.1 Quantitative analysis process attribute	110
5.5.2.	PA 4.2 Quantitative control process attribute	112

5.6.	Process capability Level 5: Innovating process.....	114
5.6.1.	PA 5.1 Process innovation process attribute.....	114
5.6.2.	PA 5.2 Process innovation implementation process attribute.....	116
Annex A	Conformity statements.....	118
Annex A.1	Introduction.....	118
Annex A.2	Conformance to the requirements for process reference models.....	118
Annex A.3	Conformance to the requirements for process assessment models.....	119
Annex A.4	Conformance to the requirements for measurement frameworks.....	121
Annex B	Information Item Characteristics.....	122
Annex C	Key concepts.....	153
Annex C.1	The "Plug-in" concept.....	153
Annex C.2	Integration of Machine Learning Engineering Processes.....	154
Annex C.3	Example of an ML Architecture.....	156
Annex C.4	Traceability and consistency.....	157
Annex D	Reference standards.....	159

List of Figures

Figure 1	— Process assessment model relationship.....	13
Figure 2	— Automotive SPICE® process reference model - Overview.....	14
Figure 3	— Possible levels of abstraction for the term "process".....	26
Figure 4	— Performing a process assessment for determining process capability.....	27
Figure 5	— Annex C.1: The "Plug-in" concept.....	153
Figure 6	— Annex C.3: Integration of MLE Processes.....	154
Figure 7	— Annex C.3: Interdependencies within MLE and SUP.11.....	155
Figure 8	— Annex C.4: Example of an ML Architecture.....	156
Figure 9	— Annex C.4: Consistency and traceability between system and software work products.....	157
Figure 10	— Annex C.4: Consistency and traceability between system and hardware work products.....	157
Figure 11	— Annex C.4: Consistency and traceability between ML work products.....	158

List of Tables

Table 1	— Terminology.....	11
Table 2	— Abbreviation List.....	11
Table 3	— Process capability levels.....	17
Table 4	— Process attributes.....	18
Table 5	— Rating scale.....	19
Table 6	— Rating scale percentage values.....	19
Table 7	— Refinement of rating scale.....	19
Table 8	— Refined rating scale percentage values.....	19
Table 9	— Capability levels and corresponding process attribute ratings.....	22
Table 10	— Template for the process description.....	28
Table 11	— Template for the process description.....	99
Table B.1	— Structure of information item characteristics (IIC) table.....	122
Table B.2	— Information Item Characteristics.....	152
Table D.1	— Referenced standards.....	159

1. Introduction

1.1. Scope

Process assessment is a disciplined evaluation of an organizational unit's processes for product development against a process assessment model.

The Automotive SPICE® process assessment model (PAM) is intended for use when performing conformant assessments of the process capability on the development of embedded automotive systems. It was developed in accordance with the requirements of ISO/IEC 33004:2015.

The Automotive SPICE® process reference model (PRM), which was developed based on the ISO/IEC 12207, has further developed and tailored to the specific needs of the automotive industry. If processes beyond the scope of Automotive SPICE® are needed, appropriate processes from other process reference models such as ISO/IEC 12207 or ISO/IEC/IEEE 15288 may be added based on the business needs of the organization.

The PRM is incorporated in this document and is used in conjunction with the Automotive SPICE® process assessment model when performing an assessment.

This Automotive SPICE® process assessment model contains a set of indicators to be considered when interpreting the intent of the Automotive SPICE® process reference model. These indicators may also be used when implementing a process improvement program.

1.2. Terminology

Automotive SPICE® follows the following precedence for use of terminology:

- ISO/IEC 33001 for assessment related terminology
- ISO/IEC/IEEE 24765
- Terms introduced by Automotive SPICE®
- PMBOK® Guide – Fourth Edition
- PAS 1883:2020
- Oxford Dictionary

Term	Origin	Description
Activity	Automotive SPICE® V4.0	Execution of a task by a stakeholder or an involved party.
Application parameter	Automotive SPICE® V4.0	<p>An application parameter is a software variable containing data that can be changed at the system or software levels; they influence the system's or software behavior and properties. The notion of application parameter is expressed in two ways:</p> <ul style="list-style-type: none"> • The specification (including variable names, the domain value range, technical data types, default values, physical unit (if applicable), the corresponding memory maps, respectively). • The actual quantitative data value it receives by means of data application. <p>Application parameters are not requirements. They are a technical implementation solution for configurability-oriented requirements.</p>
Approval	Automotive SPICE® V4.0	Written statement that a deliverable is fit for its intended use, and compliant with defined criteria.
Baseline	Automotive SPICE® V4.0	A defined and coherent set of read-only information, serving as an input information for the affected parties.
Deliverable	PMBOK® Guide – Fourth Edition	Any unique and verifiable product, result, or capability to perform a service that must be produced to complete a process, phase, or project. Often used more narrowly in reference to an external deliverable, which is a deliverable that is subject to approval by the project sponsor or customer.
Functional requirement	ISO/IEC/IEEE 24765	A statement that identifies what a product or process must accomplish to produce required behavior and/or results.
Hardware	Automotive SPICE® V4.0	Assembled and interconnected electrical or electronic hardware components or parts which perform analog or digital functions or operations.
Hardware component	Automotive SPICE® V4.0	<p>Logical (e.g., functional block) or physical group of hardware parts realizing a functionality, which</p> <ul style="list-style-type: none"> • cannot be realized by any of its hardware parts alone, e.g., voltage monitoring, power supply. • may be organized hierarchically, i.e., a hardware component can contain lower-level hardware components. <p><i>NOTE: Depending on the application, e.g., the populated PCB, a system-on-chip, a microcontroller, or an SBC can be considered a HW component.</i></p>

Hardware element	Automotive SPICE® V4.0	Generic term; can represent a hardware component, a hardware part, a hardware interface, or the hardware.
Hardware part	Automotive SPICE® V4.0	Fundamental HW element the purpose and functionality of which cannot be further subdivided or separated. <i>NOTE: Examples are transistors, resistors, diodes, non-populated PCB.</i> <i>NOTE: Depending on the application, e.g., a system-on-chip, a microcontroller or an SBC can be considered a HW part.</i> <i>NOTE: the term 'unit' is considered to apply to the software domain only. The term 'hardware part' can be viewed as the hardware counterpart of 'software unit'.</i>
Hyperparameter	Automotive SPICE® V4.0	In machine learning, a hyperparameter is a parameter whose value is used to control the training of the ML model. Its value must be set between training iterations. Examples: learning rate, loss function, model depth, regularization constants.
Information need	Automotive SPICE® V4.0	The need for characterizing process or product related effectiveness and efficiency (used by MAN.6 and PA 4.1).
Machine Learning (ML)	Automotive SPICE® V4.0	In Automotive SPICE® Machine Learning (ML) describes the ability of software to learn from specific training data and to apply this knowledge to other similar tasks.
Measure	Automotive SPICE® V4.0	An activity to achieve a certain intent.
Measurement	Oxford Dictionary	The activity to find the size, quantity or degree of something.
Metric	Automotive SPICE® V4.0	A quantitative or qualitative measurable indicator that matches defined information needs.
Operational Design Domain	PAS 1883:2020	Operational Design Domain (ODD) is operating conditions under which a given overall system or feature thereof is specifically designed to function. This includes, but is not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristics.
Project	ISO/IEC/IEEE 24765	Endeavor with defined start and finish dates undertaken to create a product or service in accordance with specified resources and requirements.
Release	Automotive SPICE® V4.0	A product delivered to a customer, including a defined set of functionalities and properties.
Regression verification	Automotive SPICE® V4.0	Selective re-verification of elements to verify that modifications have not caused unintended effects.
Risk	ISO/IEC/IEEE 24765	The combination of the probability of occurrence and the consequences of a given future undesirable event.

Software component	Automotive SPICE® V4.0	<p>Software component in design and implementation-oriented processes: The software architecture decomposes the software into software components across appropriate hierarchical levels down to the lowest-level software components in a conceptual model.</p> <p>Software component in verification-oriented processes: The implementation of a SW component under verification is represented, e.g., as source code, object files, library file, executable, or executable model.</p>
Software element	Automotive SPICE® V4.0	A software element is subject to integration within the software. Can represent a →SOFTWARE COMPONENT or →SOFTWARE UNIT
Software unit	Automotive SPICE® V4.0	<p>Software unit in design and implementation-oriented processes: As a result of the decomposition of a software component in a conceptual model, the software is decomposed into software units. A software unit is decided not to be further subdivided.</p> <p>Software unit in verification-oriented processes: An implemented SW unit under verification is represented, e.g., as source code files, or an object file.</p>
Stakeholder requirements	Automotive SPICE® V4.0	Any type of requirement for the stakeholders in the given context, e.g., customer requirement, supplier internal requirements (product-specific, platform etc.), legal requirements, regulatory requirements, statutory requirements, industry sector requirements, international standards, codes of practice etc. ...
System	Automotive SPICE® V4.0	A collection of interacting components organized to accomplish a specific function or set of functions within a specific environment.
System Element	Automotive SPICE® V4.0	<p>System elements can be:</p> <ul style="list-style-type: none"> • Logical and structural objects at the architectural and design level. System elements can be further decomposed into more fine-grained system elements of the architecture or design across appropriate hierarchical levels. • Physical representations of these objects, or a combination, e.g., peripherals, sensors, actuators, mechanical parts, software executables.
Task	Automotive SPICE® V4.0	A definition, but not the execution, of a coherent set of atomic actions.
Validation measure	Automotive SPICE® V4.0	<p>Validation measure can be:</p> <ul style="list-style-type: none"> • Operational use case testing under real-life conditions • Highly accelerated life testing (HALT) • Simulations under real-life conditions • End user trials • Panel or blind tests • Expert panels

Verification measure	Automotive SPICE® V4.0	<p>Verification measure can be:</p> <ul style="list-style-type: none"> • Test cases • Measurements • Calculations • Simulations • Reviews • Analyses <p>Note, that in particular domains certain verification measures may not be applicable, e.g., software units generally cannot be verified by means of calculations or analyses.</p>
----------------------	------------------------	---

Table 1 — Terminology

1.3. Abbreviations

BP	Base Practice
CCB	Change Control Board
CPU	Central Processing Unit
ECU	Electronic Control Unit
EEPROM	Electrically Erasable Programmable Read Only Memory
EOL	End-of-Line
FMEA	Failure Mode and Effect Analysis
FTA	Fault Tree Analysis
GP	Generic Practice
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
MISRA	Motor Industry Software Reliability Association
ML	Machine Learning
ODD	Operational Design Domain
PA	Process Attribute
PAM	Process Assessment Model
PRM	Process Reference Model
RAM	Random Access Memory
ROM	Read Only Memory
SPICE	Systems/Software Process Improvement and Capability dEtermination
WP	Work Product

Table 2 — Abbreviation List

2. Statement of compliance

The Automotive SPICE® process reference model and process assessment model are conformant with the ISO/IEC 33004:2015 and can be used as the basis for conducting an assessment of process capability.

An ISO/IEC 33003:2015 compliant Measurement Framework is defined in section 5.

A statement of compliance of the process assessment model and process reference model with the requirements of ISO/IEC 33004:2015 is provided in Annex A.

A statement of compliance of the measurement framework with the requirements of ISO/IEC 33003:2015 is provided in Annex A.

3. Process capability determination

The concept of process capability determination by using a process assessment model is based on a two-dimensional framework. The first dimension is provided by processes defined in a process reference model (process dimension). The second dimension consists of capability levels that are further subdivided into process attributes (capability dimension). The process attributes provide the measurable characteristics of process capability.

The process assessment model selects processes from a process reference model and supplements with indicators. These indicators support the collection of objective evidence which enable an assessor to assign ratings for processes according to the capability dimension.

The relationship is shown in Figure 1.

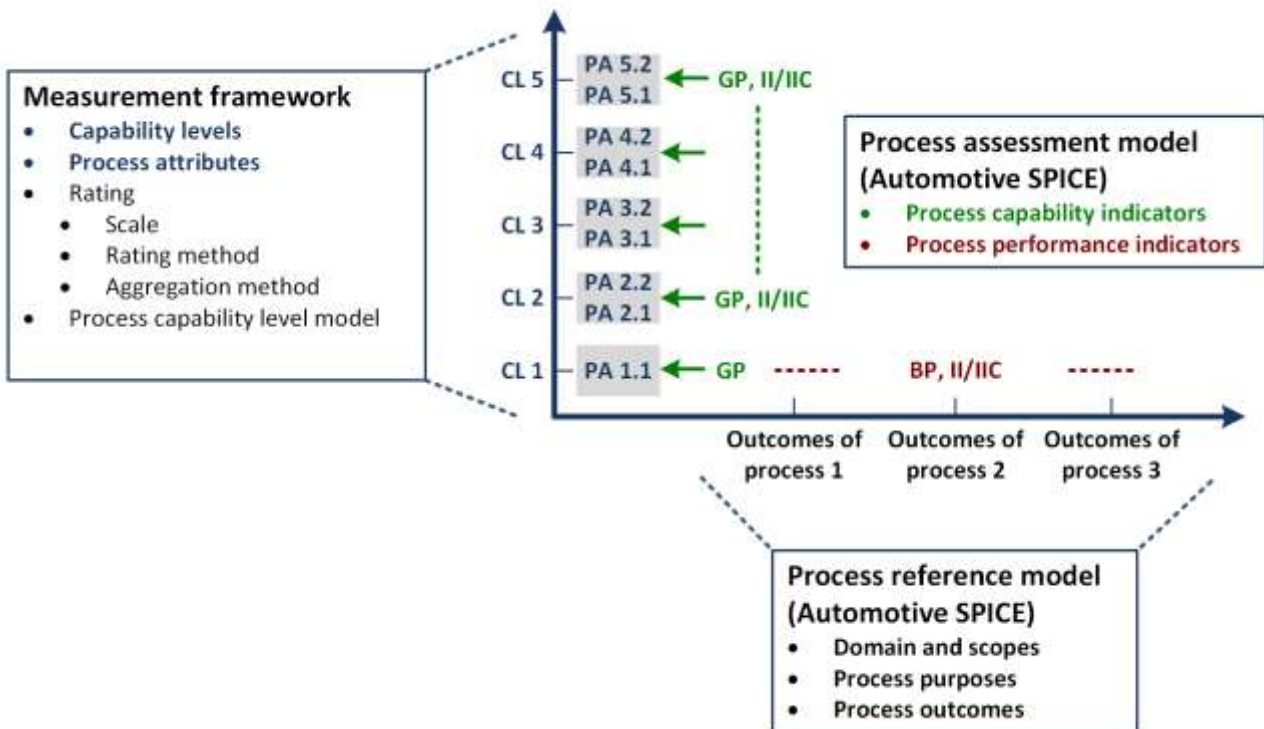


Figure 1 — Process assessment model relationship

3.1. Process reference model

Processes are clustered into process groups according to the domain of activities they address.

These process groups are organized into the process categories: Primary processes, Organizational processes and Supporting processes.

For each process a purpose statement is formulated that contains the unique functional objectives of the process when performed in a particular environment. For each purpose statement a list of specific outcomes is associated, as a list of expected positive results of the process performance.

For the process dimension, the Automotive SPICE® process reference model provides the set of processes shown in Figure 2.

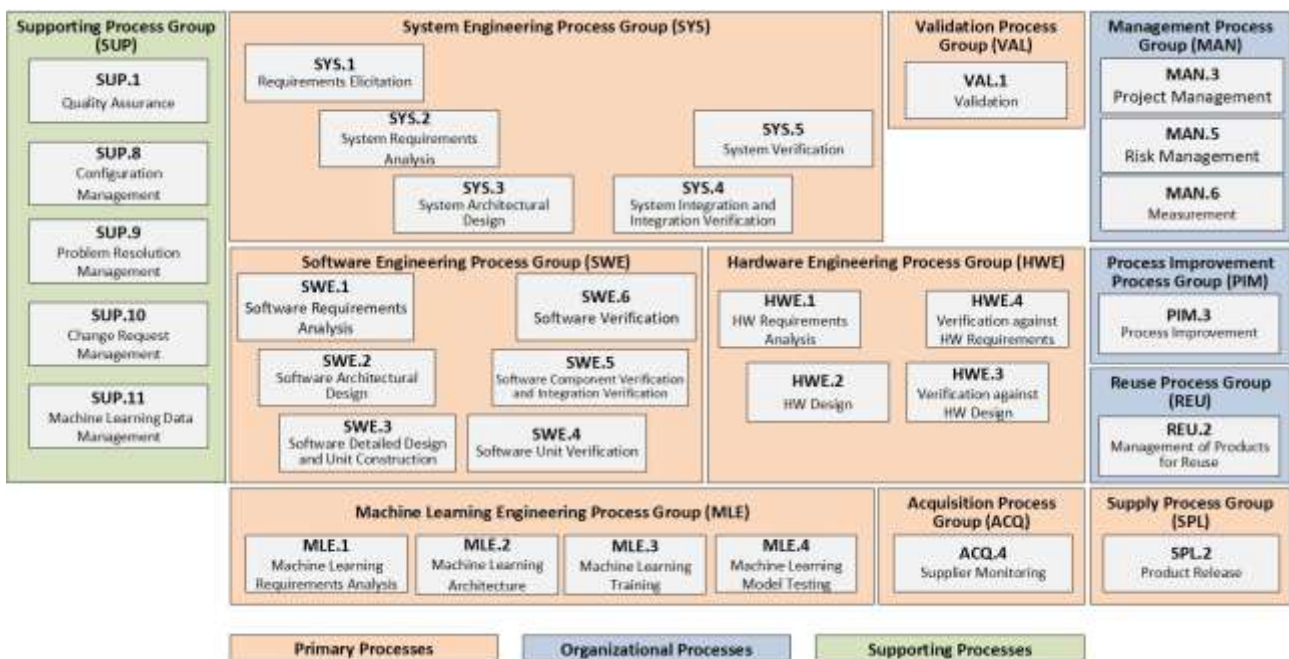


Figure 2 — Automotive SPICE® process reference model - Overview

3.1.1. Primary processes category

The primary processes category consists of processes that may apply for an acquirer of products from a supplier or may apply for product development when responding to stakeholder needs and delivering products including the engineering processes needed for specification, design, implementation, integration and verification.

The primary processes category consists of the following groups:

- the Acquisition process group
- the Supply process group
- the System Engineering process group
- the Validation process group
- the Software Engineering process group
- the Machine Learning Engineering process group
- the Hardware Engineering process group

The Acquisition process group (ACQ) consists of one process that is performed by the customer, or by the supplier when acting as a customer for its own suppliers, in order to acquire a product and/or service.

ACQ.4 Supplier Monitoring

The Supply process group (SPL) consists of one process performed by the supplier in order to supply a product and/or a service.

SPL.2 Product Release

The System Engineering process group (SYS) consists of processes addressing the elicitation and management of external and internal requirements, the definition of the system architecture and the integration and verification on the system level.

SYS.1 Requirements Elicitation

SYS.2 System Requirements Analysis

SYS.3 System Architectural Design

SYS.4 System Integration and Integration Verification

SYS.5 System Verification

The Validation process group (VAL) consists of one process that is performed to provide evidence that the product to be delivered satisfies the expectations for its intended use.

VAL.1 Validation

The Software Engineering process group (SWE) consists of processes addressing the management of software requirements derived from the system requirements, the development of the corresponding software architecture and design as well as the implementation, integration and verification of the software.

SWE.1 Software Requirements Analysis

SWE.2 Software Architectural Design

SWE.3 Software Detailed Design and Unit Construction

SWE.4 Software Unit Verification

SWE.5 Software Component Verification and Integration Verification

SWE.6 Software Verification

The Machine Learning Engineering process group (MLE) consists of processes addressing the management of ML requirements derived from the software requirements, the development of the corresponding ML architecture, the training of ML model, and testing of ML model against ML requirements.

MLE.1 Machine Learning Requirements Analysis

MLE.2 Machine Learning Architecture

MLE.3 Machine Learning Training

MLE.4 Machine Learning Model Testing

The Hardware Engineering process group (HWE) consists of processes addressing the management of hardware requirements derived from the system requirements, the development of the corresponding hardware architecture and design as well as the verification of the hardware.

HWE.1	Hardware Requirements Analysis
HWE.2	Hardware Design
HWE.3	Verification against Hardware Design
HWE.4	Verification against Hardware Requirements

3.1.2. Supporting processes category

The supporting processes category consists of processes that may be employed by any of the other processes at various points in the **lifecycle**.

SUP.1	Quality Assurance
SUP.8	Configuration Management
SUP.9	Problem Resolution Management
SUP.10	Change Request Management
SUP.11	Machine Learning Data Management

3.1.3. Organizational processes category

The organizational processes category consists of processes that develop process, product, and resource assets which, when used by projects in the organization, may help the organization achieve its business goals.

The organizational processes category consists of the following groups:

- the Management process group
- the Process Improvement process group
- the Reuse process group

The Management process group (MAN) consists of processes that may be used by anyone who manages any type of project or process within the **lifecycle**.

MAN.3	Project Management
MAN.5	Risk Management
MAN.6	Measurement

The Process Improvement process group (PIM) covers one process that contains practices to improve the processes performed in the organizational unit.

PIM.3	Process Improvement
--------------	---------------------

The Reuse process group (REU) covers one process to systematically exploit reuse opportunities in organization's product portfolio.

REU.2	Management of Products for Reuse
--------------	----------------------------------

3.2. Measurement framework

The measurement framework provides the necessary requirements and rules for the capability dimension. It defines a schema which enables an assessor to determine the Capability Level of a given process. These capability levels are defined as part of the measurement framework.

To enable the rating, the measurement framework provides process attributes defining a measurable property of process capability. Each process attribute is assigned to a specific capability level. The extent of achievement of a certain process attribute is represented by means of a rating based on a defined rating scale. The rules from which an assessor can derive a final capability level for a given process are represented by a process capability level model.

Automotive SPICE® defines its own measurement framework.

Note: The Automotive SPICE® measurement framework is an adaption of ISO/IEC 33020:2019. Text incorporated from ISO/IEC 33020 within this chapter is written in italic font and marked with a left side bar.

3.2.1. Process capability levels and process attributes

The process capability levels, and associated process attributes are described in detail in chapter 5.

Process attributes are features of a process that can be evaluated on a scale of achievement, providing a measurement of the capability of the process. They are applicable to all processes.

A capability level is characterized by one or more process attributes whose implementation results in a significant improvement in the capability to perform a process. Each attribute addresses a specific aspect of the capability level. The levels constitute a rational way of progressing through improvement of the capability of any process.

There are six capability levels as listed in Table 3:

<i>Level 0: Incomplete process</i>	<i>The process is not implemented or fails to achieve its process purpose.</i>
<i>Level 1: Performed process</i>	<i>The implemented process achieves its process purpose.</i>
<i>Level 2: Managed process</i>	<i>The previously described performed process is now implemented in a managed fashion (planned, monitored and adjusted) and its work products are appropriately established, controlled and maintained.</i>
<i>Level 3: Established process</i>	<i>The previously described managed process is now implemented using a defined process that is capable of achieving its process outcomes.</i>
<i>Level 4: Predictable process</i>	<i>The previously described established process now operates predictively within defined limits to achieve its process outcomes. Quantitative management needs are identified, measurement data are collected and analyzed to identify assignable causes of variation. Corrective action is taken to address assignable causes of variation.</i>
<i>Level 5: Innovating process</i>	<i>The previously described predictable process is now continually improved to respond to organizational change.</i>

Table 3 — Process capability levels

Within this process assessment model, the determination of capability is based upon the nine process attributes (PA) as listed in Table 4 — Process attributes.

Attribute ID	Process Attributes
<i>Level 0: Incomplete process</i>	
<i>Level 1: Performed process</i>	
PA 1.1	<i>Process performance process attribute</i>
<i>Level 2: Managed process</i>	
PA 2.1	<i>Performance management process attribute</i>
PA 2.2	<i>Work product management process attribute</i>
<i>Level 3: Established process</i>	
PA 3.1	<i>Process definition process attribute</i>
PA 3.2	<i>Process deployment process attribute</i>
<i>Level 4: Predictable process</i>	
PA 4.1	<i>Quantitative analysis process attribute</i>
PA 4.2	<i>Quantitative control process attribute</i>
<i>Level 5: Innovating process</i>	
PA 5.1	<i>Process innovation process attribute</i>
PA 5.2	<i>Process innovation implementation process attribute</i>

Table 4 — Process attributes

3.2.2. Process attribute rating

To support the rating of process attributes, the measurement framework provides a defined rating scale with an option for refinement, different rating methods and different aggregation methods depending on the class of the assessment (e.g., required for organizational maturity assessments).

3.2.2.1. Rating scale

Within this process measurement framework, a process attribute is a measurable property of process capability. A process attribute rating is a judgement of the degree of achievement of the process attribute for the assessed process.

The rating scale is shown in Table 5 — Rating scale.

Note: The rating scale is identical to ISO/IEC 33020:2019.

N	<i>Not achieved</i>	<i>There is little or no evidence of achievement of the defined process attribute in the assessed process.</i>
P	<i>Partially achieved</i>	<i>There is some evidence of an approach to, and some achievement of, the defined process attribute in the assessed process. Some aspects of achievement of the process attribute may be unpredictable.</i>

L	Largely achieved	There is evidence of a systematic approach to, and significant achievement of, the defined process attribute in the assessed process. Some weaknesses related to this process attribute may exist in the assessed process.
F	Fully achieved	There is evidence of a complete and systematic approach to, and full achievement of, the defined process attribute in the assessed process. No significant weaknesses related to this process attribute exist in the assessed process.

Table 5 — Rating scale

The ordinal scale defined above shall be understood in terms of percentage achievement of a process attribute. The corresponding percentages shall be:

N	Not achieved	0 to ≤ 15% achievement
P	Partially achieved	> 15% to ≤ 50% achievement
L	Largely achieved	> 50% to ≤ 85% achievement
F	Fully achieved	> 85% to ≤ 100% achievement

Table 6 — Rating scale percentage values

The ordinal scale may be further refined for the measures P and L as defined below.

P-	Partially achieved:	There is some evidence of an approach to, and some achievement of, the defined process attribute in the assessed process. Many aspects of achievement of the process attribute may be unpredictable.
P+	Partially achieved:	There is some evidence of an approach to, and some achievement of, the defined process attribute in the assessed process. Some aspects of achievement of the process attribute may be unpredictable.
L-	Largely achieved:	There is evidence of a systematic approach to, and significant achievement of, the defined process attribute in the assessed process. Many weaknesses related to this process attribute may exist in the assessed process.
L+	Largely achieved:	There is evidence of a systematic approach to, and significant achievement of, the defined process attribute in the assessed process. Some weaknesses related to this process attribute may exist in the assessed process.

Table 7 — Refinement of rating scale

The corresponding percentages shall be:

P-	Partially achieved -	> 15% to ≤ 32.5% achievement
P+	Partially achieved +	> 32.5 to ≤ 50% achievement
L-	Largely achieved -	> 50% to ≤ 67.5% achievement
L+	Largely achieved +	> 67.5% to ≤ 85% achievement

Table 8 — Refined rating scale percentage values

3.2.3. Rating and aggregation method

Rating and aggregation methods are taken from ISO/IEC 33020:2019, which provides the following definitions:

A process outcome is the observable result of successful achievement of the process purpose.

A process attribute outcome is the observable result of achievement of a specified process attribute.

Process outcomes and process attribute outcomes may be characterised as an intermediate step to providing a process attribute rating.

When performing rating, the rating method employed shall be specified relevant to the class of assessment. The following rating methods are defined.

The use of rating method may vary according to the class, scope and context of an assessment. The lead assessor shall decide which (if any) rating method to use. The selected rating method(s) shall be specified in the assessment input and referenced in the assessment report.

ISO/IEC 33020:2019 provides the following 3 rating methods:

Rating method R1

The approach to process attribute rating shall satisfy the following conditions:

- a) Each process outcome of each process within the scope of the assessment shall be characterized for each process instance, based on validated data;*
- b) Each process attribute outcome of each process attribute for each process within the scope of the assessment shall be characterized for each process instance, based on validated data;*
- c) Process outcome characterizations for all assessed process instances shall be aggregated to provide a process performance attribute achievement rating;*
- d) Process attribute outcome characterizations for all assessed process instances shall be aggregated to provide a process attribute achievement rating.*

Rating method R2

The approach to process attribute rating shall satisfy the following conditions:

- a) Each process attribute for each process within the scope of the assessment shall be characterized for each process instance, based on validated data;*
- b) Process attribute characterizations for all assessed process instances shall be aggregated to provide a process attribute achievement rating.*

Rating method R3

Process attribute rating across assessed process instances shall be made without aggregation.

In principle the three rating methods defined in ISO/IEC 33020:2019 depend on

- a) whether the rating is made only on process attribute level (Rating method 3 and 2) or – with more level of detail – both on process attribute and process attribute outcome level (Rating method 1); and
- b) the type of aggregation ratings across the assessed process instances for each process

If a rating is performed for both process attributes and process attribute outcomes (Rating method 1), the result will be a process performance attribute outcome rating on level 1 and a process attribute achievement rating on higher levels.

Depending on the class, scope and context of the assessment an aggregation within one process (one-dimensional, vertical aggregation), across multiple process instances (one-dimensional, horizontal aggregation) or both (two-dimensional, matrix aggregation) is performed.

ISO/IEC 33020:2019 provides the following examples:

When performing an assessment, ratings may be summarized across one or two dimensions.

For example, when rating a

- *process attribute for a given process, one may aggregate ratings of the associated process (attribute) outcomes – such an aggregation will be performed as a vertical aggregation (one dimension).*
- *process (attribute) outcome for a given process attribute across multiple process instances, one may aggregate the ratings of the associated process instances for the given process (attribute) outcome such an aggregation will be performed as a horizontal aggregation (one dimension)*
- *process attribute for a given process, one may aggregate the ratings of all the process (attribute) outcomes for all the processes instances – such an aggregation will be performed as a matrix aggregation across the full scope of ratings (two dimensions)*

The standard defines different methods for aggregation. Further information can be taken from ISO/IEC 33020:2019.

3.2.4. Process capability level model

The process capability level achieved by a process shall be derived from the process attribute ratings for that process according to the process capability level model defined in Table 9 — Capability levels.

The process capability level model defines the rules how the achievement of each level depends on the rating of the process attributes for the assessed and all lower levels.

As a general rule the achievement of a given level requires a largely or fully achievement of the corresponding process attributes and a full achievement of any lower lying process attribute.

Scale	Process attribute	Rating
Level 1	PA 1.1: Process performance process attribute	Largely or fully
Level 2	PA 1.1: Process performance process attribute PA 2.1: Process performance management process attribute PA 2.2: Work product management process attribute	Fully Largely or fully Largely or fully
Level 3	PA 1.1: Process performance process attribute PA 2.1: Process performance management process attribute PA 2.2: Work product management process attribute PA 3.1: Process definition process attribute PA 3.2: Process deployment process attribute	Fully Fully Fully Largely or fully Largely or fully
Level 4	PA 1.1: Process performance process attribute PA 2.1: Process performance management process attribute PA 2.2: Work product management process attribute PA 3.1: Process definition process attribute PA 3.2: Process deployment process attribute PA 4.1: Quantitative analysis process attribute PA 4.2: Quantitative control process attribute	Fully Fully Fully Fully Fully Largely or fully Largely or fully
Level 5	PA 1.1: Process performance process attribute PA 2.1: Process performance management process attribute PA 2.2: Work product management process attribute PA 3.1: Process definition process attribute PA 3.2: Process deployment process attribute PA 4.1: Quantitative analysis process attribute PA 4.2: Quantitative control process attribute PA 5.1: Process innovation process attribute PA 5.2: Process innovation implementation process attribute	Fully Fully Fully Fully Fully Fully Fully Largely or fully Largely or fully

Table 9 — Capability levels and corresponding process attribute ratings

3.3. Process assessment model

The process assessment model offers indicators in order to identify whether the process outcomes and the process attribute outcomes (achievements) are present or absent in the instantiated processes of projects and organizational units. These indicators provide guidance for assessors in accumulating the necessary objective evidence to support judgments of capability. They are not intended to be regarded as a mandatory set of checklists to be followed.

3.3.1. Assessment indicators

According to ISO/IEC 33004, a process assessment model needs to define a set of assessment indicators:

Assessment Indicators

A process assessment model shall be based on a set of assessment indicators that:

- a) explicitly address the purpose and process outcomes, as defined in the selected process reference model, of each of the processes within the scope of the process assessment model;*
- b) demonstrate the achievement of the process attributes within the scope of the process assessment model;*
- c) demonstrate the achievement (where relevant) of the process quality levels within the scope of the process assessment model.*

The assessment indicators generally fall into three types:

- a) **practices** that support achievement of either the process purpose or the specific process attribute.*
- b) **information items** and their characteristics that demonstrate the respective achievements.*
- c) **resources and infrastructure** that support the respective achievements.*

[ISO/IEC 33004:2015, 6.3.1]

In this assessment model, only practices and information items are used.

Practices represent activity-oriented indicators, whereas information items represent result-oriented indicators. Both practices and information items are used for judging objective evidence to be collected and accumulated in the performance of an assessment.

As a first type of assessment indicator, practices are provided, which can be divided into two types:

1. Base practices (BP), applying to capability level 1

They provide an indication of the extent of achievement of the process outcomes. Base practices relate to one or more process outcomes, thus being always process-specific and not generic.

2. Generic practices (GP), applying to capability levels 1 to 5

They provide an indication of the extent of process attribute achievement. Generic practices relate to one or more process attribute achievements, thus applying to any process.

As a second type of assessment indicators, **information items (II)** including their **characteristics (IIC)** are provided in Annex B.

These are meant to offer a good practice and state-of-the-art knowledge guide for the assessor. Therefore, information items including their characteristics are supposed to be a quickly accessible information source during an assessment.

Information item characteristics shall not be interpreted as a required structure of a corresponding work product, which is defined by the project and organization, respectively.

Please refer to chapter 3.3.2 for understanding the difference between information items and work products.

ISO 33004:2015 requires the mapping of assessment indicators to process attributes as shown in Figure 1.

The capability of a process on level 1 is only characterized by the measure of the extent to which the process outcomes are achieved. According to ISO 33003:2015, a measurement framework requires each level to reveal a process attribute. Therefore, the only process performance attribute for capability Level 1 (PA.1.1) has a single generic practice (GP 1.1.1) pointing as an editorial reference to the respective process performance indicators (see Figure 1 and chapter 4).

The detailed mapping of base practices / indicators and generic practices / indicators to process outcomes and achievements, is provided in corresponding tables in chapter 4 and 5, respectively.

3.3.2. Understanding information Items and work products

To judge the presence or absence of process outcomes and process attribute achievements, an assessment obtains objective evidence. All such evidence comes either from the examination of work products related to a specific output of the processes assessed, or from statements made by the performers and managers of the processes. Sources for such evidence are either repository content of the assessed processes, or testimony provided by the performers and managers of the assessed processes.

As described in chapter 3.3.1, this process assessment model provides information items serving as indicators to guide the assessor when judging a process attribute achievement.

3.3.2.1. Information items versus work products

ISO/IEC 33001 provides the following definition of the term “information item”:

information item

separately identifiable body of information that is produced, stored, and delivered for human use

Note 1 to entry: An information item can be produced in several versions during a system, software, or service lifecycle. Syn: information product.

[ISO/IEC 33001:2015, 3.1.4]

Note: Human use includes the information stored, managed and processed by a tool.

One common definition of the term “work product” is:

work product

artifact resulting from the execution of a process

[ISO/IEC/IEEE 24765:2017]

Both terms are used in different context in an assessment:

- Information items are defining relevant pieces of information used by the assessors to judge the achievement of process attributes.
- Work products are produced by the organization assessed when performing, managing, establishing, analyzing and innovating processes.

Information items (together with their characteristics) are provided as guidance for “what to look for” when examining the work products available in the assessed organization. The extent of implementation of an information item (in line with its defined characteristics) in a related work product serves as objective evidence supporting the assessment of a particular process. A documented process and assessor judgment is needed to ensure that the process context (application domain, business purpose, development methodology, size of the organization, etc.) is considered when using this information.

Information items shall therefore not be mistaken for the work product generated by the assessed organization itself. There is no 1:1 relationship between an information item and the work product taken as sample evidence by the assessor when assessing the achievement of a process outcome and process attribute achievements. An output generated by a process may comprise multiple information item characteristics, and multiple outputs may also contain the same information item characteristics.

Information item characteristics should be considered as indicators when considering whether, given the context, a work product is contributing to the intended purpose of the process. Context-sensitivity means that assessor judgment is needed to ensure that the actual context (application domain, business purpose, development methodology, size of the organization, etc.) is considered when using the information items.

3.3.2.2. Types of work products

A work product to be considered as evidence when rating a process attribute may not necessary be outputs from the processes assessed but can also be originated from other processes of the organization. Once such a work product is used in the performance of a process under assessment, it may be considered by the assessor as objective evidence.

In a lot of cases work products are comprising documentation aspects, such as specifications, reports, records, architectural designs, software code etc.

Examples of work products not comprising any documentation aspects are software binaries, raw data, or a physical electronic hardware.

3.3.3. **Understanding the level of abstraction of a PAM**

The term "process" can be understood at three levels of abstraction. Note that these levels of abstractions are not meant to define a strict black-or-white split, nor is it the aim to provide a scientific classification schema – the message here is to understand that, in practice, when it comes to the term "process" there are different abstraction levels, and that a PAM resides at the highest.

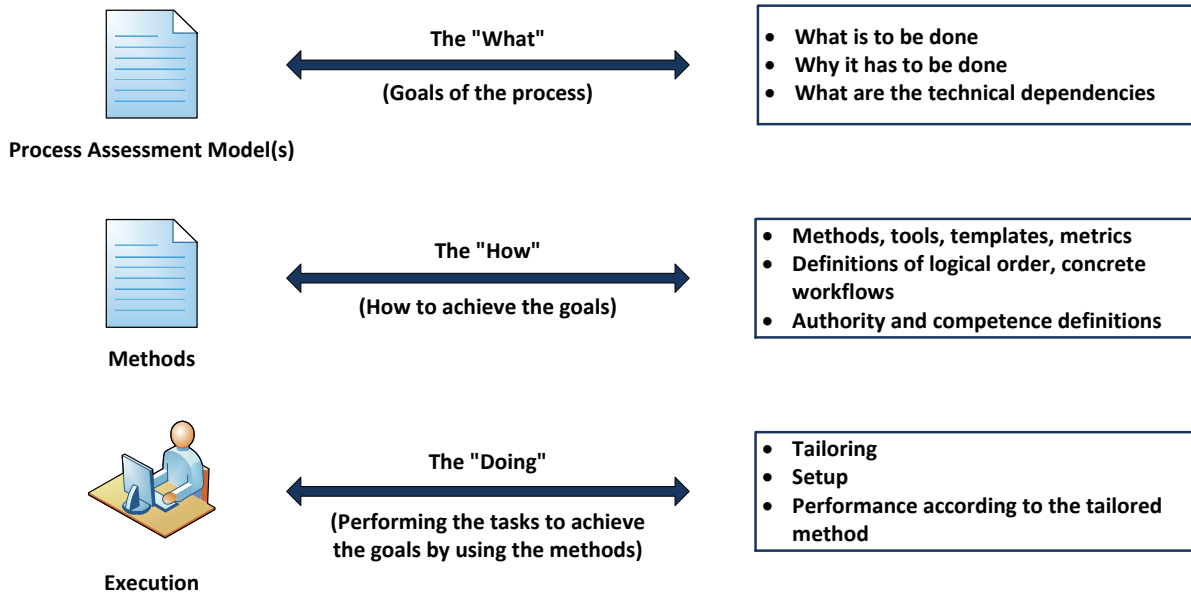


Figure 3 — Possible levels of abstraction for the term "process"

Capturing experience acquired during product development (i.e., at the DOING level) in order to share this experience with others means creating a HOW level. However, a HOW is always specific to a particular context such as a company, an organizational unit, or a product line. For example, the HOW of a project, organizational unit, or company A is potentially not applicable as is to a project, organizational unit, or company B. However, both might be expected to adhere the principles represented by PAM indicators for process outcomes and process attribute achievements. These indicators are at the WHAT level while deciding on solutions for concrete templates, proceedings, and tooling etc. is left to the HOW level.

3.3.4. Why a PRM and PAM are not a lifecycle model or development process blueprint

A lifecycle model defines phases and activities in a logical timely order, possibly including cycles or loops, and parallelization. For example, some standards such as ISO 26262 or ISO/SAE 21434 are centered around a lifecycle model (neither of these standards in fact represents a PRM according to ISO/IEC 33004). Companies, organizational units, or projects will interpret such general lifecycle models given in standards, and then detail it out into roles, organizational interactions and interfaces, tools or tool chains, work instructions, and artifacts. Lifecycle models therefore are a concept at the HOW level (see section 3.3.3).

In contrast, a PRM/PAM according to ISO/IEC 33004 (formerly ISO/IEC 15504-2) is at the level of the WHAT by abstracting from any HOW level, see Figure 3 in section 3.3.3. In Automotive SPICE®, this has been, and is, indicated by the process MAN.3 Project Management requiring in BP2 "Define the project **lifecycle**". A PRM/PAM groups a set of coherent and related characteristics of a particular technical topic and calls it 'process'. In different terms, a process in a PRM represents a 'distinct conceptual silo'. In this respect, a PRM/PAM

- neither predefines, nor discourages, any order in which PRM processes or Base Practices are to be performed. Ultimately, in Automotive SPICE® consistency must be fulfilled as

required by the traceability/consistency Base Practices in MAN.3 or SYS.x, SWE.x, and HWE.x;

- does not predefine any particular work product structure, or work product blueprints. For example, the process SYS.2 does not mean that there shall be exactly one system requirements specification or repository containing everything derived from the stakeholders.

As a consequence, it is the assessor’s responsibility to map the elements in such a HOW level to the Assessment Indicators in the PAM, see Figure 4.

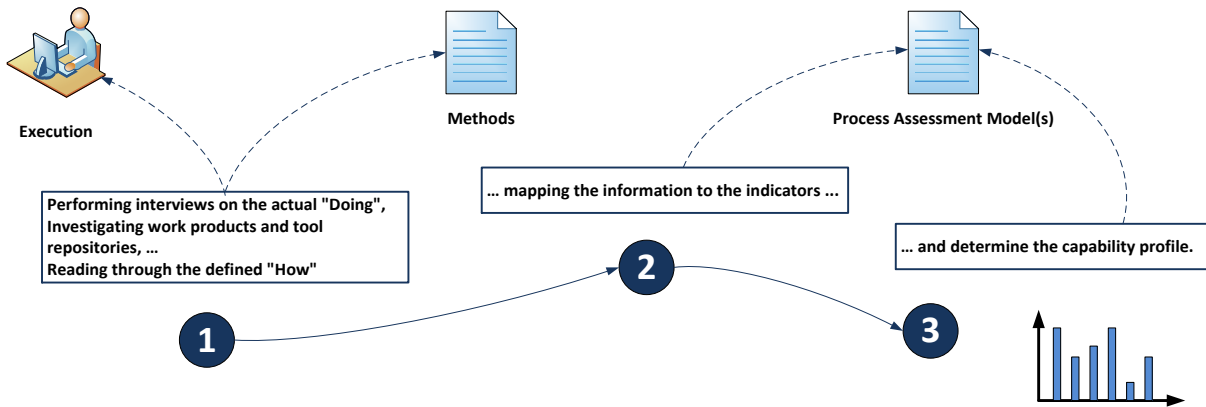


Figure 4 — Performing a process assessment for determining process capability

In this respect, a PRM or PAM further is not supposed to represent a product element hierarchy either.

4. Process reference model and performance indicators (Level 1)

The processes in the process dimension can be drawn from the Automotive SPICE® process reference model, which is incorporated in the tables below indicated by a red bar at the left side.

Each table related to one process in the process dimension contains the process reference model (indicated by a red bar) and the process performance indicators necessary to define the process assessment model. The process performance indicators consist of base practices (indicated by a green bar) and output information items (indicated by a blue bar).

Process reference model	Process ID Process name Process purpose Process outcomes	<p>The individual processes are identified with a unique process identifier and a process name. A process purpose statement is provided, and process outcomes are defined to represent the process dimension of the Automotive SPICE® process reference model. The background coloring of process ID's and names are indicating the assignment to the corresponding process group.</p>
	Base Practices	<p>A set of base practices for the process providing a definition of the activities to be performed to accomplish the process purpose and fulfill the process outcomes.</p> <p>The base practice headers are summarized at the end of a process to demonstrate their relationship to the process outcomes.</p>
Process performance indicators	Output information items	<p>The output information items that are relevant to accomplish the process purpose and fulfill the process outcomes summarized at the end of a process to demonstrate their relationship to the process outcomes.</p> <p><i>Note: Refer to Annex B for the characteristics of each information item.</i></p>

Table 10 — Template for the process description

4.1. Acquisition process group (ACQ)

4.1.1. ACQ.4 Supplier Monitoring

Process ID
ACQ.4
Process name
Supplier Monitoring
Process purpose
The purpose is to track and assess the performance of an external contract-based supplier company against the agreed commitments.
Process outcomes
<ol style="list-style-type: none"> 1) The joint activities, as agreed between the customer and the supplier, are performed. 2) All information, agreed upon for exchange, is communicated regularly between the customer and the supplier. 3) Performance of the supplier is monitored against the agreements. 4) The changes to the agreement, if needed, are negotiated between the customer and the supplier and documented in the agreement.
Base Practices
<p>ACQ.4.BP1: Agree on and maintain joint activities, joint interfaces, and information to be exchanged. Establish and maintain an agreement on information to be exchanged, on joint activities, joint interfaces, responsibilities, type and frequency of joint activities, communications, meetings, status reports, and reviews.</p>
<p>ACQ.4.BP2: Exchange all agreed information. Use the defined joint interfaces between customer and supplier for the exchange of all agreed information.</p>
<p>ACQ.4.BP3: Review development work products with the supplier. Review development work products with the supplier on the agreed regular basis, covering technical aspects, problems and risks. Track the open measures.</p> <p><i>Note 1: See SUP.9 for management of problems.</i></p>
<p>ACQ.4.BP4: Review progress of the supplier. Review progress of the supplier regarding schedule, quality, and cost on the agreed regular basis. Track the open measures to closure and perform the risk mitigation activities.</p> <p><i>Note 2: See MAN.5 for management of risks.</i></p>
<p>ACQ.4.BP5: Act to correct the deviations. Take action when agreed objectives are not achieved. Negotiate the changes to the objectives and document them in the agreements.</p>

ACQ.4 Supplier Monitoring	Outcome 1	Outcome 2	Outcome 3	Outcome 4
Output Information Items				
02-01 Commitment/Agreement	X	X	X	X
13-52 Communication evidence	X	X	X	
13-09 Meeting support evidence	X	X		
13-14 Progress status		X	X	
13-16 Change request				X
13-19 Review evidence		X		
14-02 Corrective action				X
15-51 Analysis results			X	
Base Practices				
BP1: Agree on and maintain joint processes, joint interfaces, and information to be exchanged	X	X		X
BP2: Exchange all agreed information	X	X	X	
BP3: Review development work products with the supplier	X		X	X
BP4: Review progress of the supplier	X		X	X
BP5: Act to correct the deviations			X	X

4.2. Supply process group (SPL)

4.2.1. SPL.2 Product Release

Process ID
SPL.2
Process name
Product Release
Process purpose
The purpose is to control the release of a product to the intended customer.
Process outcomes
<ol style="list-style-type: none"> 1) The contents of the product releases are determined. 2) The release package is assembled from configured items. 3) The release documentation is defined and produced. 4) The release approval is performed against the defined criteria. 5) The release package is made available to the intended customer.

Base Practices
<p>SPL.2.BP1: Define and agree the functional content of the releases. Define and agree the functionality to be included with the intended customer, and define the release criteria for each release.</p> <p><i>Note 1: This may include the hardware elements, software elements, and extra application parameter files (influencing the identified system functionality) that are needed for the release.</i></p> <p><i>Note 2: The intended customer may be an internal organizational unit or an external organization.</i></p>
<p>SPL.2.BP2: Define the release package. Define the release as well as supporting tools and information.</p> <p><i>Note 3: The release package may include also programming tools.</i></p>
<p>SPL.2.BP3: Ensure a unique identification of each release. Ensure a unique identification of the releases based upon the intended purpose and expectations of the releases.</p> <p><i>Note 4: Unique identification may be realized by a classification and numbering scheme for product releases.</i></p>
<p>SPL.2.BP4: Build the release from items under configuration control. Build the release from items under configuration control to ensure integrity.</p> <p><i>Note 5: This practice may be supported by the SUP.8 Configuration Management Process.</i></p>
<p>SPL.2.BP5: Ensure the release approval before the delivery. The criteria for the release are satisfied before the delivery takes place.</p>
<p>SPL.2.BP6: Provide a release note. Each release is accompanied by information detailing the key characteristics of the release.</p> <p><i>Note 6: The release note may include information about legal aspects like relevant target markets, legislation that is considered etc. See also VAL.1 Validation.</i></p>
<p>SPL.2.BP7: Communicate the type, service level and duration of support for each release. Identify and communicate the type, service level and duration of support for a release.</p>
<p>SPL.2.BP8: Deliver the release package to the intended customer. Deliver the release package to the intended customer as per the agreement.</p>

SPL.2 Product Release	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5
Output Information Items					
11-03 Release note	X		X	X	X
11-04 Product release package		X	X		
13-06 Delivery evidence			X		X
13-13 Product release approval				X	X

18-06 Product release criteria	X	X		X	
Base Practices					
BP1: Define and agree the functional content of the releases.	X				
BP2: Define the release package	X				
BP3 Ensure a unique identification of each release			X		
BP4: Build the release from items under configuration control		X			
BP5: Ensure the release approval before the delivery				X	
BP6: Provide a release note			X		X
BP7: Communicate the type, service level and duration of support for each release.			X		X
BP8: Deliver the release package to the intended customer					X

4.3. System Engineering process group (SYS)

4.3.1. SYS.1 Requirements Elicitation

Process ID
SYS.1
Process name
Requirements Elicitation
Process purpose
The purpose is to gather, analyze, and track evolving stakeholder needs and requirements throughout the lifecycle of the product and/or service to establish a set of agreed requirements.
Process outcomes
<ol style="list-style-type: none"> 1) Relevant stakeholders are identified. 2) Continuing communication with the stakeholder is established. 3) The stakeholder expectations are understood, and the requirements derived from them are defined and agreed upon. 4) The stakeholder requirements changes arising from stakeholder needs are analyzed to enable associated risk assessment and impact management. 5) The status of each stakeholder requirement is transparent for all affected parties.
Base Practices
<p>SYS.1.BP1: Identify stakeholders and obtain their expectations and requests. Identify relevant stakeholders. Obtain and define the stakeholder expectations and requests through direct solicitation of stakeholder input, and through review of stakeholder business proposals (where relevant) and other documents containing inputs to the stakeholder requirements, and consideration of the target operating and hardware environment.</p> <p><i>Note 1: Documenting the stakeholder, or the source of a stakeholder requirement, supports the stakeholder requirements agreement and change analysis (see BP2 and BP3).</i></p>
<p>SYS.1.BP2: Agree on the requirements. Formalize the stakeholder’s expectations and requests into requirements. Reach a common understanding and agree on the set of stakeholder requirements among the affected parties.</p> <p><i>Note 2: Examples of affected parties are customers, suppliers, design partners, joint venture partners, or outsourcing parties.</i></p> <p><i>Note 3: The agreed stakeholder requirements may be based on feasibility studies and/or cost and schedule impact analyses.</i></p>

SYS.1.BP3: Analyze the stakeholder requirements changes. Analyze all changes made to the agreed stakeholder requirements. Assess the impact and risks, and initiate appropriate change control and mitigation actions.

Note 4: Requirements changes may arise from different sources such as changing technology, stakeholder needs, or legal constraints.

Note 5: Refer to SUP.10 Change Request Management, if required.

SYS.1.BP4: Communicate the requirements' status. Ensure all affected parties can be aware of the status and disposition of the requirements concerning them including the changes and can communicate necessary information and data.

SYS.1 Requirements Elicitation	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5
Output Information Items					
15-51 Analysis Results				X	
13-52 Communication Evidence		X	X		
14-50 Stakeholder groups	X				
17-00 Requirement			X		
17-54 Requirement Attribute			X	X	X
Base Practices					
BP1: Identify stakeholders and obtain their expectations and requests	X	X			
BP2: Agree on the requirements			X		
BP3: Analyze the stakeholder requirements changes				X	
BP4: Communicate the requirements status		X			X

4.3.2. SYS.2 System Requirements Analysis

Process ID
SYS.2
Process name
System Requirements Analysis
Process purpose
The purpose is to establish a structured and analyzed set of system requirements consistent with the stakeholder requirements.
Process outcomes
<ol style="list-style-type: none"> 1) The system requirements are specified. 2) The system requirements are structured and prioritized. 3) The system requirements are analyzed for correctness and technical feasibility. 4) The impact of the system requirements on the operating environment is analyzed. 5) Consistency and bidirectional traceability are established between the system requirements and the stakeholder requirements. 6) The system requirements are agreed and communicated to all affected parties.

Base Practices
<p>SYS.2.BP1: Specify the system requirements. Use the stakeholder requirements to identify and document the functional and non-functional requirements for the system according to the defined characteristics for requirements.</p> <p><i>Note 1: Characteristics of requirements are defined in standards such as ISO IEEE 29148, ISO 26262-8:2018, or the INCOSE Guide For Writing Requirements.</i></p> <p><i>Note 2: Examples for defined characteristics of requirements shared by technical standards are verifiability (i.e., verification criteria being inherent in the requirements text), unambiguity/comprehensibility, freedom from design and implementation, and not contradicting any other requirement).</i></p>
<p>SYS.2.BP2: Structure the system requirements. Structure and prioritize the system requirements.</p> <p><i>Note 3: Examples for structuring can be grouping by functionality or product variants identification.</i></p> <p><i>Note 4: Prioritization can be done according to project or stakeholder needs via, e.g., definition of release scopes. Please refer to SPL.2.BP1.</i></p>
<p>SYS.2.BP3: Analyze the system requirements. Analyze the specified system requirements including their interdependencies to ensure correctness, technical feasibility, and to support project management regarding project estimates.</p> <p><i>Note 5: See MAN.3.BP3 for project feasibility and MAN.3.BP5 for project estimates.</i></p> <p><i>Note 6: Technical feasibility can be evaluated based on, e.g., platform or product line, or by means of prototype development or product demonstrators.</i></p>

SYS.2.BP4: Analyze the impact on the system context. Analyze the impact that the system requirements will have on the elements in the relevant system context.

SYS.2.BP5: Ensure consistency and establish bidirectional traceability. Ensure consistency and establish bidirectional traceability between the system requirements and the stakeholder requirements.

Note 7: Bidirectional traceability supports consistency, facilitates the impact analyses of change requests, and supports the demonstration of coverage of the stakeholder requirements. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent.

Note 8: There may be non-functional stakeholder requirements that the system requirements do not trace to. Examples are process requirements. Such stakeholder requirements are still subject to verification.

SYS.2.BP6: Communicate the agreed system requirements and the impact on the system context. Communicate the agreed system requirements, and the results of the impact analysis on the system context, to all affected parties.

SYS.2 System Requirements Analysis	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6
Output Information Items						
17-00 Requirement	X	X				
17-54 Requirement Attribute		X	X			
15-51 Analysis Results			X	X		
13-51 Consistency Evidence					X	
13-52 Communication Evidence						X
Base Practices						
BP1: Specify the system requirements	X					
BP2: Structure the system requirements		X				
BP3: Analyze the system requirements			X			
BP4: Analyze the impact on the system context				X		
BP5: Ensure consistency and establish bidirectional traceability					X	
BP6: Communicate the agreed system requirements and the impact on the system context						X

4.3.3. SYS.3 System Architectural Design

Process ID
SYS.3
Process name
System Architectural Design
Process purpose
The purpose is to establish an analyzed system architecture, comprising the static and dynamic aspects, consistent with the system requirements.
Process outcomes
<ol style="list-style-type: none"> 1) A system architecture is designed including a definition of the system elements with their behavior, their interfaces, their relationships, and their interactions. 2) The system architecture is analyzed against the defined criteria, and the special characteristics are identified. 3) Consistency and bidirectional traceability are established between the system architecture and the system requirements. 4) The agreed system architecture and the special characteristics are communicated to all affected parties.

Base Practices
SYS.3.BP1: Specify the static aspects of the system architecture. Specify and document the static aspects of the system architecture with respect to the functional and non-functional system requirements, including the external interfaces and a defined set of system elements with their interfaces and relationships.
SYS.3.BP2: Specify the dynamic aspects of the system architecture. Specify and document the dynamic aspects of the system architecture with respect to the functional and non-functional system requirements including the behavior of the system elements and their interaction in different system modes. <i>Note 1: Examples of interactions of system elements are timing diagrams reflecting inertia of mechanical components, processing times of ECUs, and signal propagation times of bus systems.</i>
SYS.3.BP3: Analyze the system architecture. Analyze the system architecture regarding relevant technical design aspects related to the product lifecycle, and to support the project management regarding project estimates, and derive the special characteristics for non-software system elements. Document the rationales for the system architectural design decisions. <i>Note 2: See MAN.3.BP3 for project feasibility and MAN.3.BP5 for project estimates.</i> <i>Note 3: Examples of product lifecycle phases are production, maintenance & repair, decommissioning.</i> <i>Note 4: Examples of technical aspects are manufacturability for production, suitability of pre-existing system elements to be reused, or availability of system elements.</i> <i>Note 5: Examples of suitable methods for analyzing technical aspects are prototypes, simulations, and qualitative analyses (e.g., FMEA approaches).</i>

Note 6: Examples of design rationales are proven-in-use, reuse of a product platform or product line), a make-or-buy decision, or found in an evolutionary way (e.g., set-based design).

SYS.3.BP4: Ensure consistency and establish bidirectional traceability. Ensure consistency and establish bidirectional traceability between the elements of the system architecture and the system requirements that represent properties or characteristics of the physical end product.

Note 7: Bidirectional traceability further supports consistency, and facilitates the impact analysis of change requests, and the demonstration of coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent.

Note 8: There may be non-functional requirements that the system architectural design does not trace to. Examples are requirements that do not address, or represent, direct properties or characteristics of the physical end product. Such requirements are still subject to verification.

SYS.3.BP5: Communicate the agreed system architecture. Communicate the agreed system architecture, including the special characteristics, to all affected parties.

SYS.3 System Architectural Design	Outcome 1	Outcome 2	Outcome 3	Outcome 4
Output Information Items				
04-06 System Architecture	X			
13-51 Consistency Evidence			X	
13-52 Communication Evidence				X
15-51 Analysis Results		X		
17-57 Special Characteristics		X		
Base Practices				
BP1: Specify the static aspects of the system architecture	X			
BP2: Specify the dynamic aspects of the system architecture	X			
BP3: Analyze the system architecture		X		
BP4: Ensure consistency and establish bidirectional traceability			X	
BP5: Communicate the agreed system architecture				X

4.3.4. SYS.4 System Integration and Integration Verification

Process ID
SYS.4
Process name
System Integration and Integration Verification
Process purpose
The purpose is to integrate the system elements and to verify that the integrated system elements are consistent with the system architecture.
Process outcomes
<ol style="list-style-type: none"> 1) Verification measures are specified for the system integration verification of the integrated system elements based on the system architecture, including the interfaces of, and interactions between, the system elements. 2) The system elements are integrated up to a complete integrated system consistent with the release scope. 3) Verification measures are selected according to the release scope considering criteria, including criteria for regression verification. 4) The integrated system elements are verified using the selected verification measures, and the results of the system integration verification are recorded. 5) Consistency and bidirectional traceability are established between verification measures and the elements of the system architecture. 6) Bidirectional traceability between the verification results and the verification measures is established. 7) Results of the system integration and integration verification are summarized and communicated to all affected parties.

Base Practices

SYS.4.BP1: Specify verification measures for the system integration. Specify verification measures, based on a defined sequence and preconditions for the integration of system elements against the static and dynamic aspects of the system architecture, including

- techniques for the verification measures,
- pass/fail criteria for the verification measures,
- a definition of entry and exit criteria for the verification measures,
- necessary sequence of the verification measures, and
- the required verification infrastructure and environment setup.

Note 1: Examples of what a verification measure may focus on are the timing dependencies of the correct signal flow between interfacing system elements, or interactions between hardware and software, as specified in the system architecture. The system integration test cases may focus on

- *the correct signal flow between system items,*
- *the timeliness and timing dependencies of signal flow between system items,*
- *the correct interpretation of signals by all system items using an interface, and/or*
- *the dynamic interaction between system items.*

SYS.4.BP2: Select verification measures. Document the selection of verification measures for each integration step considering selection criteria including criteria for the regression verification. The documented selection of verification measures shall have sufficient coverage according to the release scope.

Note 2: Examples of selection criteria can be prioritization of requirements, the need for regression verification (due to, e.g., changes to the system architectural design or to system components), or the intended use of the delivered product release (e.g., test bench, test track, public road etc.).

SYS.4.BP3: Integrate the system elements and perform the integration verification. Integrate the system elements until the system is fully integrated according to the specified interfaces and interactions between the system elements, and according to the defined sequence and the defined preconditions. Perform the selected system integration verification measures. Record the verification measure data including pass/fail status and the corresponding verification measure data.

Note 3: Examples for preconditions for starting system integration can be successful system element verification or qualification of pre-existing system elements.

Note 4: See SUP.9 for handling the verification results which deviate from expected results.

SYS.4.BP4: Ensure consistency and establish bidirectional traceability. Ensure consistency and establish bidirectional traceability between the verification measures and the system architecture. Establish bidirectional traceability between the verification results and the verification measures.

Note 5: Bidirectional traceability supports consistency, and facilitates the impact analysis of change requests, and the demonstration of coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent.

SYS.4.BP5: Summarize and communicate the results. Summarize the system integration and integration verification results and communicate them to all affected parties.

Note 6: Providing all necessary information from the test case execution in a summary enables other parties to judge the consequences.

SYS.4 System Integration and Integration Verification	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6	Outcome 7
Output Information Items							
08-60 Verification Measure	X						
06-50 Integration Sequence Instruction		X					
03-50 Verification Measure Data				X			
08-58 Verification Measure Selection Set			X				
15-52 Verification Results				X			
13-51 Consistency Evidence					X	X	
13-52 Communication Evidence							X
11-06 Integrated System		X					
Base Practices							
BP1: Specify verification measures for the system integration	X						
BP2: Select verification measures			X				
BP3: Integrate system elements and perform the integration verification		X		X			
BP4: Ensure consistency and establish bidirectional traceability					X	X	
BP5: Summarize and communicate the results							X

4.3.5. SYS.5 System Verification

Process ID
SYS.5
Process name
System Verification
Process purpose
The purpose is to ensure that the system is verified to be consistent with the system requirements.
Process outcomes
<ol style="list-style-type: none"> 1) Verification measures are specified for the verification of the system based on the system requirements. 2) Verification measures are selected according to the release scope considering criteria, including criteria for regression verification. 3) The integrated system is verified using the selected verification measures and the results of the system verification are recorded. 4) Consistency and bidirectional traceability are established between verification measures and the system requirements. 5) Bidirectional traceability is established between the verification results and the verification measures. 6) Verification results are summarized and communicated to all affected parties.

Base Practices
<p>SYS.5.BP1: Specify verification measures for the system verification. Specify the verification measures for the system verification suitable to provide evidence for compliance with the functional and non-functional system requirements, including</p> <ul style="list-style-type: none"> • techniques for the verification measures, • pass/fail criteria for the verification measures, • a definition of entry and exit criteria for the verification measures, • necessary sequence of the verification measures, and • the required verification infrastructure and environment setup. <p><i>Note 1: The system verification measures may cover aspects such as thermal, environmental, robustness/lifetime, and EMC.</i></p>

SYS.5.BP2: Select verification measures. Document the selection of verification measures considering the selection criteria including criteria for the regression verification. The documented selection of verification measures shall have sufficient coverage according to the release scope.

Note 2: Examples for criteria for selection can be prioritization of requirements, the need for regression verification (due to, e.g., changes to the system requirements), the intended use of the delivered product release (test bench, test track, public road etc.).

SYS.5.BP3: Perform the verification of the integrated system. Perform the verification of the integrated system using the selected verification measures. Record the verification results including pass/fail status and the corresponding verification measure data.

Note 3: See SUP.9 for handling the verification results which deviate from expected results.

SYS.5.BP4: Ensure consistency and establish bidirectional traceability. Ensure consistency and establish bidirectional traceability between the verification measures and the system requirements. Establish bidirectional traceability between the verification results and the verification measures.

Note 4: Bidirectional traceability supports consistency, and facilitates impact analysis of change requests, and demonstration of coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent.

SYS.5.BP5: Summarize and communicate the results. Summarize the system verification results and communicate them to all affected parties.

Note 5: Providing all necessary information from the test case execution in a summary enables other parties to judge the consequences.

SYS.5 System Verification	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6
Output Information Item						
08-60 Verification Measure	X					
03-50 Verification Measure Data			X			
08-58 Verification Measure Selection Set		X				
15-52 Verification Results			X			
13-51 Consistency Evidence				X	X	
13-52 Communication Evidence						X
Base Practices						
BP1: Specify verification measures for the system verification	X					
BP2: Select verification measures		X				
BP3: Perform the verification of the integrated system			X			
BP4: Ensure consistency and establish bidirectional traceability				X	X	
BP5: Summarize and communicate the results						X

4.4. Software Engineering process group (SWE)

4.4.1. SWE.1 Software Requirements Analysis

Process ID
SWE.1
Process name
Software Requirements Analysis
Process purpose
The purpose is to establish a structured and analyzed set of software requirements consistent with the system requirements, and the system architecture.
Process outcomes
<ol style="list-style-type: none"> 1) The software requirements are specified. 2) The software requirements are structured and prioritized. 3) The software requirements are analyzed for correctness and technical feasibility. 4) The impact of the software requirements on the operating environment is analyzed. 5) Consistency and bidirectional traceability are established between the software requirements and the system requirements. 6) Consistency and bidirectional traceability are established between the software requirements and the system architecture. 7) The software requirements are agreed and communicated to all affected parties.
Base Practices
<p>SWE.1.BP1: Specify the software requirements. Use the system requirements and the system architecture to identify and document the functional and non-functional requirements for the software according to the defined characteristics for requirements.</p> <p><i>Note 1: Characteristics of requirements are defined in standards such as ISO IEEE 29148, ISO 26262-8:2018, or the INCOSE Guide for Writing Requirements.</i></p> <p><i>Note 2: Examples for defined characteristics of requirements shared by technical standards are verifiability (i.e., verification criteria being inherent in the requirements text), unambiguity/comprehensibility, freedom from design and implementation, and not contradicting any other requirement).</i></p> <p><i>Note 3: In case of software-only development, the system requirements and the system architecture refer to a given operating environment. In that case, the stakeholder requirements can be used as the basis for identifying the required functions and capabilities of the software.</i></p> <p><i>Note 4: The hardware-software-interface (HSI) definition puts in context hardware and therefore it is an interface decision at the system design level. If such a HSI exists, then it may provide input to software requirements.</i></p>

SWE.1.BP2: Structure the software requirements. Structure and prioritize the software requirements.

Note 5: Examples for structuring criteria can be grouping (e.g., by functionality) or expressing product variants.

Note 6: Prioritization can be done according to project or stakeholder needs via, e.g., definition of release scopes. Refer to SPL.2.BP1.

SWE.1.BP3: Analyze the software requirements. Analyze the specified software requirements including their interdependencies to ensure correctness, technical feasibility, and to support project management regarding project estimates.

Note 7: See MAN.3.BP3 for project feasibility and MAN.3.BP5 for project estimates.

Note 8: Technical feasibility can be evaluated based on, e.g., platform or product line, or by prototyping.

SWE.1.BP4: Analyze the impact on the operating environment. Analyze the impact that the software requirements will have on the elements in the operating environment.

SWE.1.BP5: Ensure consistency and establish bidirectional traceability. Ensure consistency and establish bidirectional traceability between the software requirements and the system architecture. Ensure consistency and establish bidirectional traceability between the software requirements and the system requirements.

Note 9: Redundant traceability is not intended.

Note 10: There may be non-functional system requirements that the software requirements do not trace to. Examples are process requirements or requirements related to later software product lifecycle phases such as incident handling. Such requirements are still subject to verification.

Note 11: Bidirectional traceability supports consistency, and facilitates impact analysis of change requests, and demonstration of coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent.

Note 12: In case of software development only, the system requirements and system architecture refer to a given operating environment. In that case, consistency and bidirectional traceability can be ensured between stakeholder requirements and software requirements.

SWE.1.BP6: Communicate the agreed software requirements and the impact on the operating environment. Communicate the agreed software requirements, and the results of the analysis of impact on the operating environment, to all affected parties.

SWE.1 Software Requirements Analysis	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6	Outcome 7
Output Information Items							
17-00 Requirement	X	X					
17-54 Requirement Attribute		X	X				
15-51 Analysis Results			X	X			
13-51 Consistency Evidence					X	X	
13-52 Communication Evidence							X

Base Practices							
BP1: Specify the software requirements	X						
BP2: Structure the software requirements		X					
BP3: Analyze the software requirements			X				
BP4: Analyze the impact on the operating environment				X			
BP5: Ensure consistency and establish bidirectional traceability					X	X	
BP6: Communicate agreed the software requirements and the impact on the operating environment							X

4.4.2. SWE.2 Software Architectural Design

Process ID
SWE.2
Process name
Software Architectural Design
Process purpose
The purpose is to establish an analyzed software architecture, comprising the static and dynamic aspects, consistent with the software requirements.
Process outcomes
<ol style="list-style-type: none"> 1) A software architecture is designed including the static and dynamic aspects. 2) The software architecture is analyzed against the defined criteria. 3) Consistency and bidirectional traceability are established between the software architecture and the software requirements. 4) The software architecture is agreed and communicated to all affected parties.

Base Practices
<p>SWE.2.BP1: Specify the static aspects of the software architecture. Specify and document the static aspects of the software architecture with respect to the functional and non-functional software requirements, including the external interfaces and a defined set of software components with their interfaces and relationships.</p> <p><i>Note 1: The hardware-software-interface (HSI) definition puts in context the hardware design and therefore is an aspect of system design (SYS.3).</i></p>
<p>SWE.2.BP2: Specify the dynamic aspects of the software architecture. Specify and document the dynamic aspects of the software architecture with respect to the functional and non-functional software requirements, including the behavior of the software components and their interaction in different software modes, and concurrency aspects.</p> <p><i>Note 2: Examples for concurrency aspects are application-relevant interrupt handling, preemptive processing, multi-threading.</i></p> <p><i>Note 3: Examples for behavioral descriptions are natural language or semi-formal notation (e.g., SysML, UML).</i></p>

SWE.2.BP3: Analyze the software architecture. Analyze the software architecture regarding relevant technical design aspects, and to support the project management regarding project estimates. Document the rationales for the software architectural design decisions.

Note 4: See MAN.3.BP3 for project feasibility and MAN.3.BP5 for project estimates.

Note 5: The analysis may include the suitability of pre-existing software components for the current application.

Note 6: Examples of methods suitable for analyzing technical aspects are prototypes, simulations, qualitative analyses.

Note 7: Examples of technical aspects are functionality, timings, and resource consumption (e.g., ROM, RAM, external / internal EEPROM or Data Flash or CPU load).

Note 8: Design rationales can include arguments such as proven-in-use, reuse of a software framework or software product line, a make-or-buy decision, or found in an evolutionary way (e.g., set-based design).

SWE.2.BP4: Ensure consistency and establish bidirectional traceability. Ensure consistency and establish bidirectional traceability between the software architecture and the software requirements.

Note 9: There may be non-functional software requirements that the software architectural design does not trace to. Examples are development process requirements. Such requirements are still subject to verification.

Note 10: Bidirectional traceability supports consistency, and facilitates impact analysis of change requests, and demonstration of coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent.

SWE.2.BP5: Communicate the agreed software architecture. Communicate the agreed software architecture to all affected parties.

SWE.2 Software Architectural Design	Outcome 1	Outcome 2	Outcome 3	Outcome 4
Output Information Items				
04-04 Software Architecture	X			
13-51 Consistency Evidence			X	
13-52 Communication Evidence				X
15-51 Analysis Results		X		
Base Practices				
BP1: Specify the static aspects of the software architecture	X			
BP2: Specify the dynamic aspects of the software architecture	X			
BP3: Analyze the software architecture		X		
BP4: Ensure consistency and establish bidirectional traceability			X	
BP5: Communicate the agreed software architecture				X

4.4.3. SWE.3 Software Detailed Design and Unit Construction

Process ID
SWE.3
Process name
Software Detailed Design and Unit Construction
Process purpose
The purpose is to establish a software detailed design, comprising the static and dynamic aspects, consistent with the software architecture, and to construct the software units consistent with the software detailed design.
Process outcomes
<ol style="list-style-type: none"> 1) A detailed design is specified including the static and dynamic aspects. 2) Software units as specified in the software detailed design are constructed. 3) Consistency and bidirectional traceability are established between the software detailed design and the software architecture. 4) Consistency and bidirectional traceability are established between the source code and the software detailed design. 5) Consistency and bidirectional traceability are established between the software detailed design and the software requirements. 6) The source code and the agreed software detailed design are communicated to all affected parties.

Base Practices
<p>SWE.3.BP1: Specify the static aspects of the detailed design. For each software component specify the behavior of its software units, their static structure and relationships, their interfaces including</p> <ul style="list-style-type: none"> • valid data value ranges for the inputs and outputs (from the application domain perspective), and • physical or measurement units applicable to the inputs and outputs (from the application domain perspective). <p><i>Note 1: The boundary of a software unit is independent from the software unit's representation in the source code, code file structure, or model-based implementation, respectively. It is rather driven by the semantics of the application domain perspective. Therefore, a software unit may be, at the code level, represented by a single subroutine or a set of subroutines.</i></p> <p><i>Note 2: Examples of valid data value ranges with applicable physical units from the application domain perspective are '0..200 [m/s]', '0..3.8 [A]' or '1..100 [N]'. For mapping such application domain value ranges to programming language-level data types (such as unsigned Integer with a value range of 0..65535) refer to BP2.</i></p> <p><i>Note 3: Examples of a measurement unit are '%' or '‰'.</i></p> <p><i>Note 4: A counter is an example of a parameter, or a return value, to which neither a physical nor a measurement unit is applicable.</i></p> <p><i>Note 5: The hardware-software-interface (HSI) definition puts in context the hardware design and</i></p>

therefore is an aspect of the system design (SYS.3).

SWE.3.BP2: Specify the dynamic aspects of the detailed design. Specify and document the dynamic aspects of the detailed design with respect to the software architecture, including the interactions between relevant software units to fulfill the component's dynamic behavior.

Note 6: Examples for behavioral descriptions are natural language or semi-formal notation (e.g., SysML, UML).

SWE.3.BP3: Develop the software units. Develop and document the software units consistent with the detailed design, and according to the coding principles.

Note 7: Examples for coding principles at capability level 1 are not to use implicit type conversions, only one entry and one exit point in subroutines, and range checks (design-by-contract, defensive programming). Further examples see, e.g., ISO 26262-6 clause 8.4.5 together with table 6.

SWE.3.BP4: Ensure consistency and establish bidirectional traceability. Ensure consistency and establish bidirectional traceability between the software detailed design and the software architecture. Ensure consistency and establish bidirectional traceability between the developed software units and the software detailed design. Ensure consistency and establish traceability between the software detailed design and the software requirements.

Note 8: Redundancy should be avoided by establishing a combination of these approaches.

Note 9: Examples for tracing a software unit in the detailed design to a software requirement directly are communication matrices or basis software aspects such as a list of diagnosis identifiers inherent in an Autosar configuration.

Note 10: Bidirectional traceability supports consistency, and facilitates impact analysis of change requests, and demonstration of coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent.

SWE.3.BP5: Communicate the agreed software detailed design and the developed software units. Communicate the agreed software detailed design and the developed software units to all affected parties.

SWE.3 Software Detailed Design and Unit Construction	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6
Output Information Items						
04-05 Software Detailed Design	X					
11-05 Software Unit	X	X				
13-51 Consistency Evidence			X	X	X	
13-52 Communication Evidence						X
Base Practices						
BP1: Specify the static aspects of the detailed design	X					
BP2: Specify the dynamic aspects of the detailed design	X					
BP3: Develop the software units		X				

BP4: Ensure consistency and establish bidirectional traceability			X	X	X	
BP5: Communicate the agreed software detailed design and the developed software units						X

4.4.4. SWE.4 Software Unit Verification

Process ID
SWE.4
Process name
Software Unit Verification
Process purpose
The purpose is to verify that software units are consistent with the software detailed design.
Process outcomes
<ol style="list-style-type: none"> 1) Verification measures for the software unit verification are specified. 2) Software unit verification measures are selected according to the release scope, including criteria for regression verification. 3) The Software units are verified using the selected verification measures, and the results are recorded. 4) Consistency and bidirectional traceability are established between verification measures and the software units. 5) Bidirectional traceability is established between the verification results and the verification measures. 6) Results of the software unit verification are summarized and communicated to all affected parties.

Base Practices
<p>SWE.4.BP1: Specify software unit verification measures. Specify verification measures for each software unit defined in the software detailed design, including</p> <ul style="list-style-type: none"> • pass/fail criteria for verification measures, • entry and exit criteria for verification measures, and • the required verification infrastructure. <p><i>Note 1: Examples for unit verification measures are static analysis, code reviews, and unit tests.</i> <i>Note 2: Static analysis can be done based on MISRA rulesets and other coding standards.</i></p>
<p>SWE.4.BP2: Select software unit verification measures. Document the selection of verification measures considering selection criteria including criteria for the regression verification. The documented selection of verification measures shall have sufficient coverage according to the release scope.</p>

SWE.4.BP3: Verify the software units. Perform the software unit verification using the selected verification measures. Record the verification results including pass/fail status and corresponding verification measure data.

Note 3: See SUP.9 for handling the verification results which deviate from expected results.

SWE.4.BP4: Ensure consistency and establish bidirectional traceability. Ensure consistency and establish bidirectional traceability between the verification measures and the software units defined in the detailed design. Establish bidirectional traceability between the verification results and the verification measures.

Note 4: Bidirectional traceability supports consistency, and facilitates impact analysis of change requests, and demonstration of coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent.

SWE.4.BP5: Summarize and communicate the results. Summarize the results of software unit verification and communicate them to all affected parties.

Note 5: Providing all necessary information from the test case execution in a summary enables other parties to judge the consequences.

SWE.4 Software Unit Verification	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6
Output Information Items						
08-60 Verification Measure	X					
03-50 Verification Measure Data			X			
08-58 Verification Measure Selection Set		X				
15-52 Verification Results			X			
13-51 Consistency Evidence				X	X	
13-52 Communication Evidence						X
Base Practices						
BP1: Specify software unit verification measures	X					
BP2: Select software unit verification measures		X				
BP3: Verify the software units			X			
BP4: Ensure consistency and establish bidirectional traceability for software unit verification				X	X	
BP5: Summarize and communicate the results						X

4.4.5. SWE.5 Software Component Verification and Integration Verification

Process ID
SWE.5
Process name
Software Component Verification and Integration Verification
Process purpose
The purpose is to verify that the software components are consistent with the software architectural design, and to integrate the software elements and verify that the integrated software elements are consistent with the software architecture and the software detailed design.
Process outcomes
<ol style="list-style-type: none"> 1) Verification measures are specified for the software integration verification of the integrated software elements based on the software architecture and the detailed design, including the interfaces of, and interactions between, the software components. 2) Verification measures for the software components are specified to provide evidence for compliance of the software components with the software components' behavior and interfaces. 3) The Software elements are integrated up to a complete integrated software. 4) Verification measures are selected according to the release scope considering criteria, including criteria for regression verification. 5) The software components are verified using the selected verification measures, and the results of the integration verification are recorded. 6) The integrated software elements are verified using the selected verification measures, and the results of the integration verification are recorded. 7) Consistency and bidirectional traceability are established between the verification measures and the software architecture as well as the detailed design. 8) Bidirectional traceability is established between the verification results and the verification measures. 9) The results of the software component verification and software elements integration verification are summarized and communicated to all affected parties.
Base Practices
<p>SWE.5.BP1: Specify software integration verification measures. Specify verification measures, based on a defined sequence and preconditions for the integration of software elements, against the defined static and dynamic aspects of the software architecture, including</p> <ul style="list-style-type: none"> • techniques for the verification measures, • pass/fail criteria for verification measures, • entry and exit criteria for verification measures, • necessary sequence of the verification measures, and

- the required verification infrastructure and environment setup.

Note 1: Examples on which the software integration verification measures may focus on are the correct dataflow and dynamic interaction between software components together with their timing dependencies, the correct interpretation of data by all software components using an interface, and the compliance to resource consumption objectives.

Note 2: The software integration verification measure may be supported by using hardware debug interfaces or simulation environments (e.g., Software-in-the-Loop-Simulation).

SWE.5.BP2: Specify verification measures for verifying the software component behavior. Specify verification measures for the software component verification against the defined software components' behavior and their interfaces in the software architecture, including

- techniques for the verification measures,
- entry and exit criteria for verification measures,
- pass/fail criteria for verification measures, and
- the required verification infrastructure and environment setup.

Note 3: Verification measures are related to software components but not to the software units since the software unit verification is addressed in the process SWE.4 Software Unit Verification.

SWE.5.BP3: Select verification measures. Document the selection of integration verification measures for each integration step considering selection criteria including criteria for the regression verification. The documented selection of verification measures shall have sufficient coverage according to the release scope.

Note 4: Examples for selection criteria can be the need for continuous integration / continuous development regression verification (due to, e.g., changes to the software architectural or detailed design), or the intended use of the delivered product release (e.g., test bench, test track, public road etc.).

SWE.5.BP4: Integrate the software elements and perform the integration verification. Integrate the software elements until the software is fully integrated according to the specified interfaces and interactions between the software elements, and according to the defined sequence and the defined preconditions. Perform the selected integration verification measures. Record the verification measure data including pass/fail status and the corresponding verification measure data.

Note 5: Examples for preconditions for starting the software integration are qualification of pre-existing software components, off-the-shelf software components, open-source-software, or auto-code generated software.

Note 6: Defined preconditions may allow, e.g., big-bang-integration of all software components, continuous integration, as well as stepwise integration (e.g., across software units and/or software components up to the fully integrated software) with accompanying verification measures.

Note 7: See SUP.9 for handling the verification results which deviate from expected results.

SWE.5.BP5: Perform the software component verification. Perform the selected verification measures for verifying the software component behavior. Record the verification results including pass/fail status and the corresponding verification measure data.

Note 8: See SUP.9 for handling the verification results which deviate from expected results.

SWE.5.BP6: Ensure consistency and establish bidirectional traceability. Ensure consistency and establish bidirectional traceability between the verification measures and the static and dynamic aspects of the software architecture as well as the detailed design. Establish bidirectional traceability between the verification results and the verification measures.

Note 9: Bidirectional traceability supports consistency, and facilitates impact analysis of change requests, and demonstration of coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent.

SWE.5.BP7: Summarize and communicate the results. Summarize the software component verification and the software integration verification results and communicate them to all affected parties.

Note 10: Providing all necessary information from the execution of the verification measures in a summary enables other parties to judge the consequences.

SWE.5 Software Component Verification and Integration Verification	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6	Outcome 7	Outcome 8	Outcome 9
Output Information Items									
08-60 Verification Measure	X	X							
06-50 Integration Sequence Instruction			X						
03-50 Verification Measure Data					X				
08-58 Verification Measure Selection Set				X					
15-52 Verification Results					X	X			
13-51 Consistency Evidence							X	X	
13-52 Communication Evidence									X
01-03 Software Component			X						
01-50 Integrated Software			X						
Base Practices									
BP1: Specify software integration verification measures	X								
BP2: Specify verification measures for verifying the software component behavior		X							
BP3: Select verification measures				X					
BP4: Integrate the software elements and perform the integration verification			X			X			
BP5: Perform the software component verification					X				
BP6: Ensure consistency and establish bidirectional traceability							X	X	
BP7: Summarize and communicate the results									X

4.4.6. SWE.6 Software Verification

Process ID
SWE.6
Process name
Software Verification
Process purpose
The purpose is to ensure that the integrated software is verified to be consistent with the software requirements.
Process outcomes
<ol style="list-style-type: none"> 1) Verification measures are specified for the verification of the software based on the software requirements. 2) Verification measures are selected according to the release scope considering criteria, including criteria for regression verification. 3) The integrated software is verified using the selected verification measures and the results of the software verification are recorded. 4) Consistency and bidirectional traceability are established between verification measures and software requirements. 5) Bidirectional traceability is established between the verification results and the verification measures. 6) The results of the software verification are summarized and communicated to all affected parties.

Base Practices
<p>SWE.6.BP1: Specify verification measures for the software verification. Specify the verification measures for the software verification suitable to provide evidence for compliance of the integrated software with the functional and non-functional software requirements, including</p> <ul style="list-style-type: none"> • techniques for the verification measures, • pass/fail criteria for the verification measures, • a definition of entry and exit criteria for the verification measures, • necessary sequence of the verification measures, and • the required verification infrastructure and environment setup. <p><i>Note 1: The selection of appropriate techniques for verification measures may depend on the content of the respective software requirement (e.g., boundary values and equivalence classes for data range-oriented requirements, positive/sunny-day-test vs. negative testing such as fault injection), or on requirements-based testing vs. "error guessing based on knowledge or experience".</i></p>

SWE.6.BP2: Select verification measures. Document the selection of verification measures considering the selection criteria including criteria for the regression verification. The documented selection of verification measures shall have sufficient coverage according to the release scope.

Note 2: Examples for selection criteria can be prioritization of requirements, continuous development, the need for regression verification (due to, e.g., changes to the software requirements), or the intended use of the delivered product release (test bench, test track, public road etc.).

SWE.6.BP3: Verify the integrated software. Perform the verification of the integrated software using the selected verification measures. Record the verification results including pass/fail status and the corresponding verification measure data.

Note 3: See SUP.9 for handling the verification results which deviate from expected results.

SWE.6.BP4: Ensure consistency and establish bidirectional traceability. Ensure consistency and establish bidirectional traceability between the verification measures and the software requirements. Establish bidirectional traceability between the verification results and the verification measures.

Note 4: Bidirectional traceability supports consistency, and facilitates impact analysis of change requests, and demonstration of coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent.

SWE.6.BP5: Summarize and communicate the results. Summarize the software verification results and communicate them to all affected parties.

Note 5: Providing all necessary information from the test case execution in a summary enables other parties to judge the consequences.

SWE.6 Software Verification	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6
Output Information Items						
08-60 Verification Measure	X					
03-50 Verification Measure Data			X			
08-58 Verification Measure Selection Set		X				
15-52 Verification Results			X			
13-51 Consistency Evidence				X	X	
13-52 Communication Evidence						X
Base Practices						

BP1: Specify verification measures for the software verification	X					
BP2: Select verification measures		X				
BP3: Verify the integrated software			X			
BP4: Ensure consistency and establish bidirectional traceability.				X	X	
BP5: Summarize and communicate the results						X

4.5. Validation process group (VAL)

4.5.1. VAL.1 Validation

Process ID
VAL.1
Process name
Validation
Process purpose
The purpose is to provide evidence that the end product, allowing direct end user interaction, satisfies the intended use expectations in its operational target environment.
Process outcomes
<ol style="list-style-type: none"> 1) Validation measures are selected considering criteria for the regression verification. 2) The product is validated using the selected validation measures, and the results of validation are recorded. 3) Consistency and bidirectional traceability are established between validation measures and the stakeholder requirements, if applicable; and bidirectional traceability is established between the validation results and the validation measures. 4) The results of the validation are summarized and communicated to all affected parties.
Base Practices
<p>VAL.1.BP1: Specify validation measures for the product validation. Specify the validation measures for the end product based on the stakeholder requirements to provide evidence that it fulfills its intended use expectations in its operational target environment, and this may include</p> <ul style="list-style-type: none"> • techniques for the validation measures, • pass/fail criteria for the validation measures, if applicable • a definition of entry criteria for the validation measures, if applicable • necessary sequence of the validation measures, and • the required validation infrastructure and environment setup. <p><i>Note 1: An example for validation-relevant stakeholder requirements are homologation or legal type approval requirements. Further examples of sources of intended use expectations are technical risks (see MAN.5, SYS.3.BP4, SWE.2.BP3, HWE.2.BP6).</i></p> <p><i>Note 2: Where stakeholder requirements cannot be specified comprehensively or change frequently, repeated validation of (often rapidly developed) increments in product evolution may be employed to refine stakeholder requirements, and to mitigate risks in the correct identification of needs.</i></p> <p><i>Note 3: Validation may also be conducted to confirm that the product also satisfies the often less formally expressed, but sometimes overriding, attitudes, experience, and subjective tests that comprise stakeholder or end user satisfaction.</i></p>

VAL.1.BP2: Select validation measures. Document the selection of validation measures considering the selection criteria including criteria for regression validation. The documented selection of validation measures shall have sufficient coverage according to the release scope.

Note 4: Examples for criteria for selection can be the release purpose of the delivered product (such as test bench, test track, validation on public roads, field use by end users), homologation/ type approval, confirmation of requirements, or the need for regression due to, e.g., changes to stakeholder requirements and needs.

VAL.1.BP3: Perform the validation and evaluate the results. Perform the validation of the integrated end product using the selected validation measures. Record the validation results including pass/fail status. Evaluate the validation results.

Note 5: Validation results can be used as a means for identifying stakeholder or system requirements, e.g., in the case of mock-ups or concept studies.

Note 6: See SUP.9 for handling the validation results which deviate from expected results.

VAL.1.BP4: Ensure consistency and establish bidirectional traceability. Ensure consistency and establish bidirectional traceability, if applicable, from validation measures to the stakeholder requirements from which they are derived. Establish bidirectional traceability between the validation results and the validation measures.

Note 7: Examples of sources of validation measures from which they can be derived are legal requirements, homologation requirements, results of technical risk analyses, or stakeholder and system requirements (see SYS.1 and SYS.2).

Note 8: If sources of validation measures are, e.g., legal or homologation requirements, then direct bidirectional traceability from those sources to the validation measures are not possible. In such a case, unidirectional traceability is sufficient.

Note 9: Bidirectional traceability supports consistency, and facilitates impact analyses of change requests, and demonstration of validation coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent.

VAL.1.BP5: Summarize and communicate the results. Summarize the validation results and communicate them to all affected parties.

Note 10: Providing all necessary information from the test case execution in a summary enables other parties to judge the consequences.

VAL.1 Validation	Outcome 1	Outcome 2	Outcome 3	Outcome 4
Output Information Items				
08-59 Validation Measure	X			
08-57 Validation Measure Selection Set	X			
13-24 Validation Results		X		
13-51 Consistency Evidence			X	
13-52 Communication Evidence				X
Base Practices				

BP1: Specify validation measures for the product validation	X			
BP2: Select validation measures	X			
BP3: Perform the validation and evaluate the results		X		
BP4: Ensure consistency and establish traceability.			X	
BP5: Summarize and communicate the results				X

4.6. Machine Learning Engineering process group (MLE)

4.6.1. MLE.1 Machine Learning Requirements Analysis

Process ID
MLE.1
Process name
Machine Learning Requirements Analysis
Process purpose
The purpose is to refine the machine learning-related software requirements into a set of ML requirements.
Process outcomes
<ol style="list-style-type: none"> 1) The ML requirements including ML data requirements are identified and specified based on the software requirements and the components of the software architecture. 2) The ML requirements are structured and prioritized. 3) The ML requirements are analyzed for correctness and verifiability. 4) The impact of the ML requirements on the ML operating environment is analyzed. 5) Consistency and bidirectional traceability are established between the ML requirements and the software requirements, and between the ML requirements and the software architecture. 6) The ML requirements are agreed and communicated to all affected parties.
Base Practices
<p>MLE.1.BP1: Specify the ML requirements. Use the software requirements and the software architecture to identify and specify the functional and non-functional ML requirements, as well as the ML data requirements specifying the data characteristics and their expected distributions.</p> <p><i>Note 1: Non-functional requirements may include relevant characteristics of the ODD and aspects such as robustness, performance, and level of trustworthiness.</i></p> <p><i>Note 2: The ML data requirements are input for SUP.11 Machine Learning Data Management but also for other MLE processes.</i></p> <p><i>Note 3: In case of ML development only, stakeholder requirements represent the software requirements.</i></p> <p><i>Note 4: Data characteristics can be, e.g., gender, weather conditions, street conditions within the ODD.</i></p>
<p>MLE.1.BP2: Structure the ML requirements. Structure and prioritize the ML requirements.</p> <p><i>Note 5: Examples for structuring criteria can be grouping (e.g., by functionality) or variants identification.</i></p> <p><i>Note 6: Prioritization can be done according to project or stakeholder needs via, e.g., the definition of release scopes. Refer to SPL.2.BP1.</i></p>

MLE.1.BP3: Analyze the ML requirements. Analyze the specified ML requirements including their interdependencies to ensure correctness, technical feasibility, and ability for machine learning model testing, and to support project management regarding project estimates.

Note 7: See MAN.3.BP3 for project feasibility and MAN.3.BP5 for project estimates.

MLE.1.BP4: Analyze the impact on the ML operating environment. Analyze the impact that the ML requirements will have on the interfaces of the software components and the ML operating environment.

Note 8: The ML operating environment is defined as the infrastructure and information which both the trained ML model and the deployed ML model need for execution.

MLE.1.BP5: Ensure consistency and establish bidirectional traceability. Ensure consistency and establish bidirectional traceability between the ML requirements and the software requirements and between the ML requirements and the software architecture.

Note 9: Bidirectional traceability supports consistency, facilitates impact analyses of change requests, and verification coverage demonstration. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent.

Note 10: Redundant traceability is not intended, but at least one out of the given traceability paths is required.

MLE.1.BP6: Communicate the agreed ML requirements and the impact on the operating environment. Communicate the agreed ML requirements, and the results of the impact analysis on the ML operating environment to all affected parties.

MLE.1 Machine Learning Requirements Analysis	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6
Output Information Items						
17-00 Requirement	X	X				
17-54 Requirement attribute		X	X			
13-52 Communication evidence						X
13-51 Consistency evidence					X	
15-51 Analysis results			X	X		
Base Practices						
BP1: Specify the ML requirements	X					
BP2: Structure the ML requirements		X				
BP3: Analyze the ML requirements			X			
BP4: Analyze the impact on the ML operating environment				X		
BP5: Ensure consistency and establish bidirectional traceability					X	
BP6: Communicate the agreed ML requirements and the impact on the operating environment						X

4.6.2. MLE.2 Machine Learning Architecture

Process ID
MLE.2
Process name
Machine Learning Architecture
Process purpose
The purpose is to establish a ML architecture supporting training and deployment of a ML model, consistent with the ML requirements, and to evaluate the ML architecture against defined criteria.
Process outcomes
<ol style="list-style-type: none"> 1) A ML architecture is developed. 2) The hyperparameter ranges and the initial values are determined as a basis for the training. 3) An analysis of the ML architectural elements is conducted based on defined criteria. 4) The interfaces of the ML architectural elements are defined. 5) Consistency and bidirectional traceability are established between the ML architectural elements and the ML requirements. 6) The ML architecture is agreed and communicated to all affected parties.

Base Practices
<p>MLE.2.BP1: Develop the ML architecture. Develop and document the ML architecture that specifies the ML architectural elements including the details of the ML model, pre- and postprocessing, and hyperparameters which are required to create, train, test, and deploy the ML model.</p> <p><i>Note 1: Necessary details of the ML model may include layers, activation functions, and backpropagation. The level of detail of the ML model may not need to cover aspects like single neurons.</i></p> <p><i>Note 2: The details of the ML model may differ between the ML model used during training and the deployed ML model.</i></p>
<p>MLE.2.BP2: Determine the hyperparameter ranges and the initial values. Determine and document the hyperparameter ranges and the initial values as a basis for the training.</p>
<p>MLE.2.BP3: Analyze the ML architectural elements. Analyze the ML architecture regarding relevant technical design aspects, and to support the project management regarding project estimates. Document the rationales for the ML architectural design decision. Document the rationales for the ML architectural design decisions</p> <p><i>Note 3: Examples of technical design aspects are functionality, trustworthiness, explainability and resource consumption during training and deployment.</i></p>
<p>MLE.2.BP4: Define the interfaces of the ML architectural elements. Determine and document the internal and external interfaces of each ML architectural element including its interfaces to the related software components.</p>

MLE.2.BP5: Ensure consistency and establish bidirectional traceability. Ensure consistency and establish bidirectional traceability between the ML architectural elements and the ML requirements.

Note 4: Bidirectional traceability supports consistency, and facilitates impact analyses of change requests, and verification coverage demonstration. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent.

Note 5: The bidirectional traceability should be established on a reasonable level of abstraction to the ML architectural elements.

MLE.2.BP6: Communicate the agreed ML architecture. Inform all affected parties about the agreed ML architecture including the details of the ML model and the initial hyperparameter values.

MLE.2 Machine Learning Architecture	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6
Output Information Items						
04-51 ML architecture	X	X	X	X		
13-52 Communication evidence						X
13-51 Consistency evidence					X	
01-54 Hyperparameter	X	X				
15-51 Analysis results	X		X			
Base Practices						
BP1: Develop the ML architecture	X					
BP2: Determine the hyperparameter ranges and the initial values		X				
BP3: Analyze the ML architectural elements			X			
BP4: Define the interfaces of the ML architectural elements				X		
BP5: Ensure consistency and establish bidirectional traceability					X	
BP6: Communicate the agreed ML architecture						X

4.6.3. MLE.3 Machine Learning Training

Process ID
MLE.3
Process name
Machine Learning Training
Process purpose
The purpose is to optimize the ML model to meet the defined ML requirements.
Process outcomes
<ol style="list-style-type: none"> 1) An ML training and validation approach is specified. 2) The data set for ML training and ML validation is created. 3) The ML model, including the hyperparameter values, is optimized to meet the defined ML requirements. 4) Consistency and bidirectional traceability are established between the ML training and validation data set and the ML data requirements. 5) The results of the optimization are summarized, and the trained ML model is agreed and communicated to all affected parties.

Base Practices
<p>MLE.3.BP1: Specify the ML training and validation approach. Specify an approach which supports the training and validation of the ML model to meet the defined ML requirements. The ML training and validation approach includes</p> <ul style="list-style-type: none"> • entry and exit criteria of the training and validation, • approaches for the hyperparameter tuning / optimization, • approach for the data set creation and modification, and • the training and validation environment <p><i>Note 1: The ML training and validation approach may include random dropout and other robustification methods.</i></p> <p><i>Note 2: ML validation is one step of optimizing the model. If the model performance does not meet expectations, hyperparameter values or training and validation data may be changed. The term “validation” has a different meaning than VAL.1.</i></p>
<p>MLE.3.BP2: Create the ML training and validation data set. Select data from the ML data collection provided by SUP.11 and assign them to the data set for training and validation of the ML model according to the specified ML training and validation approach.</p> <p><i>Note 4: The ML training and validation data set may include corner cases, unexpected cases, and normal cases depending on the ML requirements.</i></p> <p><i>Note 5: A separated data set for training and validation might not be required in some cases (e.g., k-fold cross validation, no optimization of hyperparameters).</i></p>
<p>MLE.3.BP3: Create and optimize the ML model. Create the ML model according to the ML architecture and train it, using the identified ML training and validation data set according to the</p>

ML training and validation approach to meet the defined ML requirements, and training and validation exit criteria.

MLE.3.BP4: Ensure consistency and establish bidirectional traceability. Ensure consistency and establish bidirectional traceability between the ML training and validation data set and the ML data requirements.

Note 6: Bidirectional traceability supports consistency and facilitates impact analyses of change requests. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent.

MLE.3.BP5: Summarize and communicate the agreed trained ML model. Summarize the results of the optimization and inform all affected parties about the agreed trained ML model.

MLE.3 Machine Learning Training	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5
Output Information Items					
08-65 ML training and validation approach	X				
03-51 ML data set		X			
01-53 Trained ML model			X		
01-54 Hyperparameter			X		
13-51 Consistency evidence				X	
13-52 Communication evidence					X
Base Practices					
BP1: Specify the ML training and validation approach	X				
BP2: Create the ML training and validation data set		X			
BP3: Create and optimize the ML model			X		
BP4: Ensure consistency and establish bidirectional traceability				X	
BP5: Summarize and communicate the agreed trained ML model					X

4.6.4. MLE.4 Machine Learning Model Testing

Process ID
MLE.4
Process name
Machine Learning Model Testing
Process purpose
The purpose is to ensure the compliance of the trained ML model and the deployed ML model with the ML requirements.
Process outcomes
<ol style="list-style-type: none"> 1) An ML test approach is defined. 2) An ML test data set is created. 3) The trained ML model is tested. 4) The deployed ML model is derived from the trained ML model and tested. 5) Consistency and bidirectional traceability are established between the ML test approach and the ML requirements, and the ML test data set and the ML data requirements; and bidirectional traceability is established between the ML test approach and the ML test results. 6) Results of the ML model testing are summarized and communicated with the deployed ML model to all affected parties.

Base Practices

MLE.4.BP1: Specify an ML test approach. Specify an ML test approach suitable to provide evidence for compliance of the trained ML model and the deployed ML model with the ML requirements. The ML test approach includes

- ML test scenarios with distribution of data characteristics (e.g., gender, weather conditions, street conditions within the ODD) defined by ML requirements,
- the distribution and frequency of each ML test scenario inside the ML test data set,
- the expected test result per test datum,
- the pass/fail criteria of the testing,
- the entry and exit criteria of the testing,
- the approach for data set creation and modification, and
- the required testing infrastructure and environment setup.

Note 1: An expected test result per test datum might require the labeling of test data to support the comparison of the output of the ML model with the expected output.

Note 2: Test datum is the smallest amount of ML data which is processed by the ML model into only one output. E.g., one image in photo processing or an audio sequence in voice recognition.

Note 3: Data characteristic is one property of the ML data that may have different expressions in the ODD. E.g., weather condition may contain expressions like sunny, foggy or rainy.

Note 4: An ML test scenario is a combination of expressions of all defined data characteristics, e.g., weather conditions = sunny, street conditions = gravel road.

MLE.4.BP2: Create the ML test data set. Create the ML test data set needed for testing of the trained ML model and testing of the deployed ML model from the ML data collection provided by SUP.11 considering the ML test approach. The ML test data set shall not be used for training.

Note 5: The ML test data set for the trained ML model might differ from the test data set of the deployed ML model.

Note 6: Additional data sets might be used for special purposes like assurance of safety, fairness, robustness.

MLE.4.BP3: Test the trained ML model. Test the trained ML model according to the ML test approach using the created ML test data set. Record and evaluate the ML test results.

Note 7: Evaluation of test logs might include pattern analysis of failed test data to support, e.g., trustworthiness.

MLE.4.BP4: Derive the deployed ML model. Derive the deployed ML model from the trained ML model according to the ML architecture. The deployed ML model shall be used for testing and delivery to software integration.

Note 8: The deployed ML model will be integrated into the target system and may differ from the trained ML model which often requires powerful hardware and uses interpretative languages.

MLE.4.BP5: Test the deployed ML model. Test the deployed ML model according to the ML test approach using the created ML test data set. Record and evaluate the ML test results.

MLE.4.BP6: Ensure consistency and establish bidirectional traceability. Ensure consistency and establish bidirectional traceability between the ML test approach and the ML requirements, and the ML test data set and the ML data requirements; and bidirectional traceability is established between the ML test approach and the ML test results.

Note 9: Bidirectional traceability supports consistency, and facilitates impact analyses of change requests, and verification coverage demonstration. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent.

MLE.4.BP7: Summarize and communicate the results. Summarize the ML test results of the ML model. Inform all affected parties about the agreed results and the deployed ML model.

MLE.4 Machine Learning Model Testing	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6
Output Information Items						
08-64 ML test approach	X					
03-51 ML data set		X				
13-50 ML test results			X	X		
11-50 Deployed ML model				X		
13-51 Consistency evidence					X	
13-52 Communication evidence						X
Base Practices						
BP1: Specify an ML test approach	X					
BP2: Create the ML test data set		X				
BP3: Test the trained ML model			X			
BP4: Derive the deployed ML model				X		
BP5: Test the deployed ML model				X		
BP6: Ensure consistency and establish bidirectional traceability					X	
BP7: Summarize and communicate the results						X

4.7. Hardware Engineering process group (HWE)

4.7.1. HWE.1 Hardware Requirements Analysis

Process ID
HWE.1
Process name
Hardware Requirements Analysis
Process purpose
The purpose is to establish a structured and analyzed set of hardware requirements consistent with the system requirements, and the system architectural design.
Process outcomes
<ol style="list-style-type: none"> 1) The hardware requirements are specified. 2) The hardware requirements are structured and prioritized. 3) The hardware requirements are analyzed for correctness and technical feasibility. 4) The impact of the hardware requirements on the operating environment is analyzed. 5) Consistency and bidirectional traceability are established between the hardware requirements and the system requirements. 6) Consistency and bidirectional traceability are established between the hardware requirements and the system architectural design. 7) The hardware requirements are agreed and communicated to all affected parties.

Base Practices
<p>HWE.1.BP1: Specify the hardware requirements. Use the system requirements, and the system architecture including the interface definitions, to identify and document the functional and non-functional requirements of the hardware according to the defined characteristics for the requirements.</p> <p><i>Note 1: Characteristics of requirements are defined in standards such as ISO IEEE 29148, ISO/IEC IEEE 24765, ISO 26262-8:2018, or the INCOSE Guide For Writing Requirements.</i></p> <p><i>Note 2: Examples for defined characteristics of requirements shared by the above-mentioned standards are verifiability (i.e., verification criteria being inherent in the requirements text), unambiguity/comprehensibility, freedom from design and implementation, and not contradicting any other requirement.</i></p> <p><i>Note 3: In case of hardware-only development, the system requirements and the system architecture refer to a given operating environment. In that case, stakeholder requirements can be used as the basis for identifying the required functions and capabilities of the hardware.</i></p> <p><i>Note 4: The hardware-software-interface (HSI) definition puts in context software and therefore is an interface decision at the system design level. If such a HSI exists, then it may provide input to hardware requirements.</i></p>

HWE.1.BP2: Structure the hardware requirements. Structure and prioritize the hardware requirements.

Note 5: Examples for structuring criteria can be grouping (e.g., by functionality) or variants identification.

Note 6: Prioritization can be done according to project or stakeholder needs via, e.g., the definition of release scopes. Refer to SPL.2.BP1.

HWE.1.BP3: Analyze the hardware requirements. Analyze the specified hardware requirements including their interdependencies to ensure correctness, technical feasibility, and to support project management regarding project estimates.

Note 7: See MAN.3.BP3 for project feasibility and MAN.3.BP5 for project estimates.

Note 8: The analyses of technical feasibility can be done based on a given hardware design (e.g., platform) or by prototype development.

HWE.1.BP4: Analyze the impact on the operating environment. Identify the interfaces between the specified hardware and other elements of the operating environment. Analyze the impact that the hardware requirements will have on these interfaces and the operating environment.

HWE.1.BP5: Ensure consistency and establish bidirectional traceability. Ensure consistency and establish traceability between the hardware requirements and the system architecture. Ensure consistency and establish traceability between the hardware requirements and the system requirements.

Note 9: Redundant traceability is not intended.

Note 10: There may be non-functional hardware requirements that the hardware design does not trace to. Examples are development process requirements. Such requirements are still subject to verification.

Note 11: Bidirectional traceability supports consistency, and facilitates impact analysis of change requests, and demonstration of coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent.

Note 12: In case of hardware development only, the system requirements and the system architecture refer to a given operating environment. In that case, consistency and bidirectional traceability can be ensured between the stakeholder requirements and the hardware requirements.

HWE.1.BP6: Communicate the agreed hardware requirements and the impact on the operating environment. Communicate the agreed hardware requirements and the results of the analysis of impact on the operating environment to all affected parties.

HWE.1 Hardware Requirements Analysis	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6	Outcome 7
Output Information Items							
13-52 Communication Evidence							X
13-51 Consistency Evidence					X	X	
17-00 Requirement	X	X					

17-54 Requirement Attribute		X	X				
15-51 Analysis Results			X	X			
Base Practices							
BP1: Specify the hardware requirements	X						
BP2: Structure the hardware requirements		X					
BP3: Analyze the hardware requirements			X				
BP4: Analyze the impact on the operating environment				X			
BP5: Ensure consistency and establish bidirectional traceability					X	X	
BP6: Communicate the agreed hardware requirements							X

4.7.2. HWE.2 Hardware Design

Process ID
HWE.2
Process name
Hardware Design
Process purpose
The purpose is to provide an analyzed design, including dynamic aspects, that is consistent with the hardware requirements and suitable for manufacturing, and to derive production-relevant data.
Process outcomes
<ol style="list-style-type: none"> 1) A hardware architecture and hardware detailed design is developed that identifies the elements of the hardware and describes their behavior as well as their interfaces, and the dynamic interactions of the hardware elements. 2) The hardware architecture and the hardware detailed design is analyzed, and the special characteristics are identified. 3) Consistency and bidirectional traceability are established between the hardware requirements and the hardware design. 4) Hardware production data are derived from the hardware detailed design and communicated to all affected parties. 5) The hardware architecture and hardware detailed design and the special characteristics are agreed and communicated to all affected parties.

Base Practices

HWE.2.BP1: Specify the hardware architecture. Develop the hardware architecture that identifies the hardware components. Document the rationale for the defined hardware architecture.

Note 1: Examples for aspects reflected in the hardware architecture are ground concept, supply concept, EMC concept.

Note 2: Examples for a design rationale can be implied by the reuse of a standard hardware, platform, or product line, respectively, or by a make-or-buy decision, or found in an evolutionary way.

HWE.2.BP2: Specify the hardware detailed design. Based on the components identified in the hardware architecture, specify the detailed design description and the schematics for the intended hardware variants, including the interfaces between the hardware elements. Derive the hardware layout, the hardware bill of materials, and the production data.

Note 3: The identification of the hardware parts and their suppliers in the hardware bill of materials may be subject to a pre-defined repository (see also IATF 16949:2016, clause 8.4.1.2.).

Note 4: The hardware detailed design may be subject to constraints such as availability of hardware parts on the market, hardware design rules, layout rules, creepage and clearance distances, compliance of HW parts with industry standards such as AEC-Q, REACH.

HWE.2.BP3: Specify the dynamic aspects of the hardware architecture and the hardware detailed design. Evaluate and document the dynamic behavior of the relevant hardware elements and the interaction between them.

Note 5: Not all hardware elements have dynamic behavior that needs to be described.

HWE.2.BP4: Analyze the hardware architecture and the hardware detailed design. Analyze the hardware architecture and the hardware detailed design regarding relevant technical aspects, and support project management regarding project estimates. Identify the special characteristics.

Note 6: Examples for technical aspects are manufacturability for production, suitability of pre-existing hardware components to be reused, or availability of hardware elements.

Note 7: Examples of methods suitable for analyzing technical aspects are simulations, calculations, quantitative or qualitative analyses such as FMEA.

HWE.2.BP5: Ensure consistency and establish bidirectional traceability. Ensure consistency and establish traceability between the hardware elements and the hardware requirements. Ensure consistency and establish traceability between the hardware detailed design and the components of the hardware architecture.

Note 8: There may be non-functional hardware requirements that the hardware design does not trace to. Examples are development process requirements. Such requirements are still subject to verification.

Note 9: Bidirectional traceability further supports consistency, and facilitates impact analysis of change requests, and demonstration of coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent.

HWE.2.BP6: Communicate the agreed hardware architecture and hardware detailed design. Communicate the agreed hardware architecture and the hardware detailed design, including the special characteristics and relevant production data, to all affected parties.

HWE.2 Hardware Design	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5
Output Information Items					
04-52 Hardware Architecture	X				
04-53 Hardware Detailed Design	X				
15-51 Analysis Results		X			
13-51 Consistency Evidence			X		
17-57 Special Characteristics		X			
13-52 Communication Evidence					X
04-54 Hardware Schematics	X			X	
14-54 Hardware Bill of Materials	X			X	
04-55 Hardware Layout	X			X	
03-54 Hardware Production Data	X			X	
04-56 Hardware Element Interface	X				
Base Practices					
BP1: Specify the hardware architecture	X			X	
BP2: Specify the hardware detailed design	X			X	
BP3: Specify the dynamic aspects of the hardware architecture and the hardware detailed design	X				
BP4: Analyze the hardware architecture and the hardware detailed design		X			
BP5: Ensure consistency and establish bidirectional traceability			X		
BP6: Communicate the agreed hardware architecture and hardware detailed design				X	X

4.7.3. HWE.3 Verification against Hardware Design

Process ID
HWE.3
Process name
Verification against Hardware Design
Process purpose
The purpose is to ensure that the production data compliant hardware is verified to provide evidence for compliance with the hardware design.
Process outcomes

- 1) Verification measures are specified for the verification of the hardware against the hardware design, including the interfaces between the hardware elements and the dynamic aspects.
- 2) Verification measures are selected according to the release scope-considering criteria, including criteria for regression verification.
- 3) The verification is performed on production data compliant samples using the selected verification measures, and the verification results are recorded.
- 4) Consistency and bidirectional traceability are established between the hardware elements and the verification measures.
- 5) Bidirectional traceability is established between the verification measures and the verification results.
- 6) The verification results are summarized and communicated to all affected parties.

Base Practices

HWE.3.BP1: Specify verification measures for the verification against the hardware design. Specify the verification measures suitable to provide evidence for compliance of the hardware with the hardware design and its dynamic aspects. This includes

- techniques for the verification measures,
- pass/fail criteria for the verification measures,
- a definition of entry and exit criteria for the verification measures,
- necessary sequence of the verification measures, and
- the required verification infrastructure and environment setup.

Note 1: Examples on what a verification measure may focus on are the timeliness and timing dependencies of the correct signal flow between interfacing hardware elements, or interactions between hardware components.

Note 2: Measuring points can be used for stepwise testing of hardware elements.

HWE.3.BP2: Ensure the use of compliant samples. Ensure that the samples used for the verification against the hardware design are compliant with the corresponding production data, including the special characteristics. Ensure that deviations are documented and that they do not alter the verification results.

Note 3: Examples of compliance are sample reports, record of visual inspection, ICT report.

HWE.3.BP3: Select verification measures. Document the selection of verification measures considering the selection criteria including regression criteria. The documented selection of verification measures shall have sufficient coverage according to the release scope.

Note 4: Examples for selection criteria can be prioritization of requirements, the need for regression due to changes to the hardware design, or the intended use of the delivered hardware release (e.g., test bench, test track, public road etc.).

HWE.3.BP4: Verify against the hardware design. Verify the hardware design using the selected verification measures. Record the verification results including pass/fail status and the corresponding verification measure output data.

Note 5: See SUP.9 for handling the verification results which deviate from expected results.

HWE.3.BP5: Ensure consistency and establish bidirectional traceability. Ensure consistency and establish bidirectional traceability between the hardware elements and the verification measures. Establish bidirectional traceability between the verification measures and the verification results.

Note 6: Bidirectional traceability supports consistency, and facilitates impact analysis of change requests, and demonstration of coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent.

HWE.3.BP6: Summarize and communicate the results. Summarize the verification results and communicate them to all affected parties.

Note 7: Providing all necessary information from the test case execution in a summary enables other parties to judge the consequences.

HWE.3 Verification against Hardware Design	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6
Output Information Items						
08-60 Verification Measure	X					
03-50 Verification Measure Data			X			
08-58 Verification Measure Selection Set		X				
15-52 Verification Results			X			
13-51 Consistency Evidence				X	X	
13-52 Communication Evidence						X
Base Practices						
BP1: Specify verification measures for the verification against the hardware design	X					
BP2: Ensure the use of compliant samples			X			
BP3: Select verification measures		X				
BP4: Verify against the hardware design			X			
BP5: Ensure consistency and establish bidirectional traceability				X	X	
BP6: Summarize and communicate the results						X

4.7.4. HWE.4 Verification against Hardware Requirements

Process ID
HWE.4
Process name
Verification against Hardware Requirements
Process purpose

The purpose is to ensure that the complete hardware is verified to be consistent with the hardware requirements.

Process outcomes

- 1) Verification measures are specified for the verification of the hardware against the hardware requirements.
- 2) Verification measures are selected according to the release scope considering criteria, including the criteria for the regression verification.
- 3) The verification is performed, if applicable on production data compliant samples, using the selected verification measures, and the verification results are recorded.
- 4) Consistency and bidirectional traceability are established between the verification measures and the hardware requirements.
- 5) Bidirectional traceability is established between the verification measures and the verification results.
- 6) The verification results are summarized and communicated to all affected parties.

Base Practices

HWE.4.BP1: Specify verification measures for the verification against the hardware requirements. Specify the verification measure to provide evidence for compliance with the hardware requirements. This includes

- techniques for the verification measures,
- pass/fail criteria for the verification measures,
- a definition of entry and exit criteria for the verification measures,
- necessary sequence of the verification measures, and
- the required verification infrastructure and environment setup

Note 1: The verification measures may cover aspects such as thermal, environmental, robustness/lifetime, and EMC.

HWE.4.BP2: Ensure the use of compliant samples. Ensure that the samples used for the verification against the hardware requirements are compliant with the corresponding production data, including special characteristics, provided by the hardware design.

Note 2: Examples of compliance are sample reports, record of visual inspection, ICT report.

HWE.4.BP3: Select verification measures. Document the selection of verification measures considering selection criteria including regression criteria. The documented selection of verification measures shall have sufficient coverage according to the release scope.

Note 3: Examples for selection criteria can be prioritization of requirements, the need for regression due to changes to the hardware requirements, or the intended use of the delivered hardware release (e.g., test bench, test track, public road etc.).

HWE.4.BP4: Verify the compliant hardware samples. Verify the compliant hardware samples using the selected verification measures. Record the verification results including pass/fail status and the corresponding verification measure output data.

Note 4: See SUP.9 for handling the verification results which deviate from expected results.

HWE.4.BP5: Ensure consistency and establish bidirectional traceability. Ensure consistency between the hardware requirements and the verification measures. Establish bidirectional traceability between the hardware requirements and the verification measures. Establish bidirectional traceability between the verification measures and the verification results.

Note 5: Bidirectional traceability supports consistency, and facilitates impact analysis of change requests, and demonstration of coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent.

HWE.4.BP6: Summarize and communicate the results. Summarize the verification results and communicate them to all affected parties.

Note 6: Providing all necessary information from the test case execution in a summary enables other parties to judge the consequences.

HWE.4 Verification against Hardware Requirements	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6
Output Information Items						
08-60 Verification Measure	X					
03-50 Verification Measure Data			X			
08-58 Verification Measure Selection Set		X				
15-52 Verification Results			X			
13-51 Consistency Evidence				X	X	
13-52 Communication Evidence						X
Base Practices						
BP1: Specify verification measures for the verification against the hardware requirements	X					
BP2: Ensure the use of compliant samples			X			
BP3: Select verification measures		X				
BP4: Verify the compliant hardware samples			X			
BP5: Ensure consistency and establish bidirectional traceability				X	X	
BP6: Summarize and communicate the results						X

4.8. Supporting process group (SUP)

4.8.1. SUP.1 Quality Assurance

Process ID
SUP.1
Process name
Quality Assurance
Process purpose
The purpose is to provide independent and objective assurance that the work products and processes comply with the defined criteria and that the non-conformances are resolved and further prevented.
Process outcomes
<ol style="list-style-type: none"> 1) Quality assurance is performed independently and objectively without conflicts of interest. 2) Criteria for the quality of work products and process performance are defined. 3) Conformance of work products and process performance with the defined criteria and targets is verified, documented and communicated to all affected parties. 4) The non-conformances are tracked, resolved, and further prevented. 5) The non-conformances are escalated to appropriate levels of management. 6) Management ensures that the escalated non-conformances are resolved.
Base Practices
<p>SUP.1.BP1: Ensure independence of quality assurance. Ensure that quality assurance is performed independently and objectively without conflicts of interest.</p> <p><i>Note 1: Possible inputs for evaluating the independence may be assignment to financial and/or organizational structure as well as responsibility for processes that are subject to quality assurance (no self-monitoring).</i></p>
<p>SUP.1.BP2: Define the criteria for quality assurance. Define the quality criteria for work products as well as for process tasks and their performance.</p> <p><i>Note 2: Quality criteria may consider internal and external inputs such as customer requirements, standards, milestones, etc.</i></p>
<p>SUP.1.BP3: Assure quality of the work products. Identify the work products subject to quality assurance according to the quality criteria. Perform appropriate activities to evaluate the work products against the defined quality criteria and document the results.</p> <p><i>Note 3: Quality assurance activities may include reviews, problem analysis and lessons learned that improve the work products for further use.</i></p>

SUP.1.BP4: Assure quality of the process activities. Identify the processes subject to quality assurance according to the quality criteria. Perform appropriate activities to evaluate the processes against their defined quality criteria and associated target values and document the results.

NOTE 4: Quality assurance activities may include process assessments, problem analysis, regular check of methods, tools, and the adherence to defined processes, and consideration of lessons learned.

SUP.1.BP5: Summarize and communicate quality assurance activities and results. Regularly report performance, non-conformances, and trends of quality assurance activities to all affected parties.

SUP.1.BP6: Ensure the resolution of non-conformances. Analyze, track, correct, resolve, and further prevent non-conformances found in quality assurance activities.

NOTE 5: Non-conformances detected in work products may be entered into the problem resolution management process (SUP.9).

NOTE 6: Non-conformances detected in the process definition or implementation may be entered into a process improvement process (PIM.3).

SUP.1.BP7: Escalate non-conformances. Escalate relevant non-conformances to appropriate levels of management and other relevant stakeholders to facilitate their resolution.

NOTE 7: The decision whether to escalate non-conformances may be based on criteria such as delay of resolution, urgency, and risk.

SUP.1 Quality Assurance	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6
Output Information Items						
16-50 Organizational structure	X				X	
18-52 Escalation path					X	X
18-07 Quality criteria		X	X	X		
13-52 Communication evidence			X	X	X	
13-18 Quality conformance evidence			X	X		
13-19 Review evidence			X	X		
14-02 Corrective action				X		X
Base Practices						
BP1: Ensure independence of quality assurance.	X					
BP2: Define the criteria for quality assurance.		X				
BP3: Assure quality of the work products.			X	X		
BP4: Assure quality of the process activities.			X	X		

BP5: Summarize and communicate quality assurance activities and results.			X	X	X	
BP6: Ensure the resolution of non-conformances.				X		X
BP7: Escalate non-conformances.					X	X

4.8.2. SUP.8 Configuration Management

Process ID
SUP.8
Process name
Configuration Management
Process purpose
The purpose is to establish and maintain the integrity of relevant configuration items and baselines, and to make them available to all affected parties.
Process outcomes
<ol style="list-style-type: none"> 1) Selection criteria for configuration items are defined and applied. 2) The configuration item properties are defined. 3) Configuration management is established. 4) The modifications are controlled. 5) Baselining is applied. 6) The status of the configuration items is recorded and reported. 7) The completeness and consistency of the baselines is ensured. 8) The availability of backup and recovery mechanisms is verified.

Base Practices
<p>SUP.8.BP1: Identify the configuration items. Define selection criteria for identifying relevant work products to be subject to configuration management. Identify and document the configuration items according to the defined selection criteria.</p> <p><i>NOTE 1: Configuration items represent work products or group of work products which are subject to configuration management as a single entity.</i></p> <p><i>NOTE 2: Configuration items may vary in complexity, size, and type, ranging from an entire system including all system, hardware, and software documentation down to a single element or document.</i></p> <p><i>NOTE 3: The selection criteria may be applied to single work products or a group of work products.</i></p>
<p>SUP.8.BP2: Define the configuration item properties. Define the necessary properties needed for the modification and control of configuration items.</p> <p><i>NOTE 4: The configuration item properties may be defined for single configuration items or a group of items.</i></p> <p><i>NOTE 5: Configuration item properties may include a status model (e.g., Under Work, Tested, Released, etc.), storage location, access rights, etc.</i></p> <p><i>NOTE 6: The application of properties may be implemented by attributes of configuration items.</i></p>

SUP.8.BP3: Establish the configuration management. Establish configuration management mechanisms for the control of identified configuration items including the configuration item properties, including mechanisms for controlling parallel modifications of configuration items.

NOTE 7: This may include specific mechanisms for different configuration item types, such as branch and merge management, or checkout control.

SUP.8.BP4: Control the modifications. Control the modifications using the configuration management mechanisms.

NOTE 8: This may include the application of a defined status model for configuration items.

SUP.8.BP5: Establish baselines. Define and establish baselines for internal purposes, and for external product delivery, for all relevant configuration items.

SUP.8.BP6: Summarize and communicate the configuration status. Record, summarize, and communicate the status of the configuration items and established baselines to all affected parties in order to support the monitoring of progress and status.

NOTE 9: Regular communication of the configuration status, e.g., based on a defined status model supports project management, quality activities, and dedicated project phases such as software integration.

SUP.8.BP7: Ensure completeness and consistency. Ensure that the information about configuration items is correct and complete including the configuration item properties. Ensure the completeness and consistency of the baselines.

NOTE 10: Completeness and consistency of a baseline means that all required configuration items are included and consistent, and have the required status. This can be used to support, e.g., project gate approval.

SUP.8.BP8: Verify the availability of backup and recovery mechanisms. Verify the availability of appropriate backup and recovery mechanisms for the configuration management including the controlled configuration items. Initiate measures in case of insufficient backup and recovery mechanisms.

NOTE 11: Backup and recovery mechanisms may be defined and implemented by organizational units outside the project team. This may include references to corresponding procedures or regulations.

SUP.8 Configuration Management	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6	Outcome 7	Outcome 8
Output Information Items								
18-53 Configuration item selection criteria	X							
01-52 Configuration item list	X	X					X	
16-03 Configuration management system			X	X	X			
13-08 Baseline					X		X	

14-01 Change history			X	X		X		
15-56 Configuration status						X		
13-51 Consistency Evidence							X	
06-52 Backup and recovery mechanism information								X
Base Practices								
BP1: Identify the configuration items	X							
BP2: Define the configuration item properties		X						
BP3: Establish the configuration management			X	X				
BP4: Control the modifications				X				
BP5: Establish baselines					X			
BP6: Summarize and communicate the configuration status						X		
BP7: Ensure completeness and consistency							X	
BP8: Verify the availability of backup and recovery mechanisms								X

4.8.3. SUP.9 Problem Resolution Management

Process ID
SUP.9
Process name
Problem Resolution Management
Process purpose
The purpose is to ensure that problems are identified, recorded, analyzed, and their resolution is managed and controlled.
Process outcomes
<ol style="list-style-type: none"> 1) The problems are uniquely identified, recorded and classified. 2) The problems are analyzed and assessed to determine an appropriate solution. 3) Problem resolution is initiated. 4) The problems are tracked to closure. 5) The status of the problems including the trends identified are reported to the stakeholders.
Base Practices

SUP.9.BP1: Identify and record the problems. Each problem is uniquely identified, described and recorded. A status is assigned to each problem to facilitate tracking. Supporting information is provided to reproduce and diagnose the problems.

NOTE 1: Problems may relate to, e.g., product, resources, or methods.

NOTE 2: Example values for the problem status are “new”, “solved”, “closed”, etc.

NOTE 3: Supporting information may include, e.g., the origins of the problems, such as failed verification, how they can be reproduced, environmental information, by whom they have been detected.

NOTE 4: Unique identification supports traceability to changes made as needed by the change request management process (SUP.10).

SUP.9.BP2: Determine the causes and the impacts of the problems. Analyze the problems, determine their causes, including common causes if existing, and impacts. Involve the relevant parties. Categorize the problems.

NOTE 5: Problem categorization (e.g., light, medium, severe) may be based on severity, criticality, urgency, etc.

SUP.9.BP3: Authorize urgent resolution actions. Obtain authorization for immediate action if a problem requires an urgent resolution according to the categorization.

SUP.9.BP4: Raise alert notifications. If, according to the categorization, problems have a high impact on other systems or other affected parties, alert notifications need to be raised accordingly.

SUP.9.BP5: Initiate problem resolution. Initiate appropriate actions according to the categorization to resolve the problems long-term and prevent recurrence, including reviews of those actions, or initiate change requests. This includes synchronization and consistency with short-term urgent resolution actions, if applicable.

SUP.9.BP6: Track the problems to closure. Track the statuses of the problems to closure including all related change requests. The closure of the problems is accepted by the relevant stakeholders.

SUP.9.BP7: Report the status of problem resolution activities. Collect and analyze problem resolution management data, identify trends, and initiate related actions. Regularly report the results of data analysis, the identified trends and the status of problem resolution activities to the affected parties.

NOTE 6: Collected data may contain information about where the problems occurred, how and when they were found, what their impacts were, etc.

SUP.9 Problem Resolution Management	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5
Output Information Items					
13-07 Problem	X	X	X	X	

15-55 Problem analysis evidence		X			
15-12 Problem status					X
Base Practices					
BP1: Identify and record the problems	X			X	
BP2: Determine the causes and the impacts of the problems	X	X			
BP3: Authorize urgent resolution actions			X		
BP4: Raise alert notifications			X		
BP5: Initiate the resolution of each problem			X		
BP6: Track the problems to closure				X	X
BP7: Report the status of problem resolution activities					X

4.8.4. SUP.10 Change Request Management

Process ID
SUP.10
Process name
Change Request Management
Process purpose
The purpose is to ensure that change requests are recorded, analyzed, tracked, approved, and implemented.
Process outcomes
<ol style="list-style-type: none"> 1) The requests for changes are recorded and identified. 2) The change requests are analyzed, dependencies and relationships to other change requests are identified, and the impact is estimated. 3) The change requests are approved before implementation and prioritized accordingly. 4) Bidirectional traceability is established between the change requests and the affected work products. 5) The implementation of change requests is confirmed. 6) The change requests are tracked to closure and the status of the change requests is communicated to all affected parties.
Base Practices
<p>SUP.10.BP1: Identify and record the change requests. The scope for application of change requests is identified. Each change request is uniquely identified, described, and recorded, including the initiator and reason of the change request. A status is assigned to each change request to facilitate tracking.</p> <p><i>NOTE 1: Change requests may be used for changes related to, e.g., product, process, methods.</i></p> <p><i>NOTE 2: Example values for the change request status are “open”, “under investigation”, “implemented”, etc.</i></p>

NOTE 3: The change request handling may differ across the product life cycle, e.g., during prototype construction and series development.

SUP.10.BP2: Analyze and assess the change requests. The change requests are analyzed by relevant parties according to the analysis criteria. The work products affected by the change requests and dependencies on other change requests are determined. The impact of the change requests is assessed.

NOTE 4: Examples for analysis criteria are: resource requirements, scheduling issues, risks, benefits, etc.

SUP.10.BP3: Approve the change requests before implementation. The change requests are prioritized and approved for implementation based on the analysis results and the availability of resources.

NOTE 5: A Change Control Board (CCB) is an example mechanism used to approve change requests.

NOTE 6: Prioritization of change requests may be done by allocation to releases.

SUP.10.BP4: Establish bidirectional traceability. Establish bidirectional traceability between the change requests and the work products affected by the change requests. In case that the change requests are initiated by problems, establish bidirectional traceability between the change requests and the corresponding problem reports.

SUP.10.BP5: Confirm the implementation of the change requests. The implementation of the change requests is confirmed before closure by relevant stakeholders.

SUP.10.BP6: Track the change requests to closure. The change requests are tracked to closure. The status of the change requests is communicated to all affected parties.

NOTE 7: Examples for informing affected parties can be daily standup meetings or tool-supported workflows.

SUP.10 Change Request Management	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6
Output Information Items						
18-57 Change analysis criteria		X				
13-16 Change request	X	X	X		X	X
13-51 Consistency evidence				X		
Base Practices						
BP1: Identify and record the change requests	X					
BP2: Analyze and assess the change requests		X				
BP3: Approve the change requests before implementation			X			
BP4: Establish bidirectional traceability				X		

BP5: Confirm the implementation of the change requests					X	
BP6: Track the change requests to closure						X

4.8.5. SUP.11 Machine Learning Data Management

Process ID
SUP.11
Process name
Machine Learning Data Management
Process purpose
The purpose is to define and align ML data with ML data requirements, maintain the integrity and quality of the ML data, and to make them available to all affected parties.
Process outcomes
<ol style="list-style-type: none"> 1) An ML data management system including an ML data lifecycle is established. 2) An ML data quality approach is developed including ML data quality criteria. 3) The collected ML data are processed for consistency with the ML data requirements. 4) The ML data are verified against the defined ML data quality criteria and updated as needed. 5) The ML data are agreed and communicated to all affected parties.
Base Practices
<p>SUP.11.BP1: Establish an ML data management system. Establish an ML data management system which supports</p> <ul style="list-style-type: none"> • ML data management activities, • relevant sources of ML data, • ML data lifecycle including a status model, and • interfaces to affected parties. <p><i>Note 1: Supported ML data management activities may include data collection, labeling/annotation, and structuring.</i></p>
<p>SUP.11.BP2: Develop an ML data quality approach. Develop an approach to ensure that the quality of ML data are analyzed based on the defined ML data quality criteria and that activities are performed to support the avoidance of biases of data.</p> <p><i>Note 2: Examples of ML data quality criteria are relevant data sources, reliability and consistency of labelling, completeness against ML data requirements.</i></p> <p><i>Note 3: The ML data management system should support the quality criteria and activities of the ML data quality approach.</i></p> <p><i>Note 4: Biases to avoid may include sampling bias (e.g., gender, age) and feedback loop bias.</i></p> <p><i>Note 5: For the creation of ML data sets see MLE.3.BP2 and MLE.4.BP2.</i></p>

SUP.11.BP3: Collect the ML data. Relevant sources for the raw data are identified and continuously monitored for changes. The raw data are collected according to the ML data requirements.

Note 6: The identification and collection of ML data might be an organizational responsibility.

Note 7: Continuous monitoring should include the ODD and may lead to changes of the ML requirements.

SUP.11.BP4: Process the ML data. The raw data are processed (annotated, analyzed, and structured) according to the ML data requirements.

SUP.11.BP5: Assure the quality of the ML data. Perform the activities according to the ML data quality approach to ensure that the ML data meet the defined ML data quality criteria.

Note 8: These activities may include sample-based reviews or statistical methods.

SUP.11.BP6: Communicate the agreed processed ML data. Inform all affected parties about the agreed processed ML data and provide them to the affected parties.

SUP.11 Machine Learning Data Management	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5
Output Information Items					
16-52 ML data management system	X				
19-50 ML data quality approach		X			
03-53 ML data			X	X	
13-52 Communication evidence					X
Base Practices					
BP1: Establish an ML data management system	X				
BP2: Develop an ML data quality approach		X			
BP3: Collect the ML data			X		
BP4: Process the ML data			X		
BP5: Assure the quality of the ML data				X	
BP6: Communicate the agreed processed ML data					X

4.9. Management process group (MAN)

4.9.1. MAN.3 Project Management

Process ID
MAN.3
Process name
Project Management
Process purpose
The purpose is to identify and control the activities, and to establish resources necessary for a project to develop a product, in the context of the project's requirements and constraints.
Process outcomes
<ol style="list-style-type: none"> 1) The scope of the work for the project is defined. 2) The feasibility of achieving the goals of the project with available resources and constraints is evaluated. 3) The activities and resources necessary to complete the work are sized and estimated. 4) Interfaces within the project, and with other projects and organizational units, are identified and monitored. 5) Plans for the execution of the project are developed, implemented and maintained. 6) Progress of the project is monitored and reported. 7) Adjustment is performed when project goals are not achieved.

Base Practices
MAN.3.BP1: Define the scope of work. Identify the project's goals, motivation and boundaries.
<p>MAN.3.BP2: Define the project lifecycle. Define the lifecycle for the project, which is appropriate to the scope, context, and complexity of the project. Define a release scope for relevant milestones.</p> <p><i>Note 1: This may include the alignment of the project lifecycle with the customer's development process.</i></p>
<p>MAN.3.BP3: Evaluate the feasibility of the project. Evaluate the feasibility of achieving the goals of the project with respect to time, project estimates, and available resources.</p> <p><i>Note 2: The evaluation of feasibility may consider technical constraints of the project.</i></p>
<p>MAN.3.BP4: Define and monitor the work packages. Define and monitor adequately sized work packages and their dependencies according to defined project lifecycle and estimations.</p> <p><i>Note 3: The structure and the size of the work packages support an adequate progress monitoring.</i></p> <p><i>Note 4: Work packages may be organized in a work breakdown structure.</i></p>

MAN.3.BP5: Define and monitor the project estimates and resources. Define and monitor the project estimates of effort and resources based on the project's goals, risks, motivation and boundaries.

Note 5: Examples of necessary resources are budget, people, product samples, or infrastructure.

Note 6: Estimations and resources may include engineering, management and supporting processes.

MAN.3.BP6: Define and monitor the required skills, knowledge, and experience. Identify and monitor the required skills, knowledge, and experience for the project in line with the estimates and work packages.

Note 7: Training, mentoring or coaching of individuals may be applied to resolve deviations from required skills and knowledge.

MAN.3.BP7: Define and monitor the project interfaces and agreed commitments. Identify and agree the interfaces of the project with affected stakeholders and monitor the agreed commitments. Define an escalation mechanism for commitments that are not fulfilled.

Note 8: Affected stakeholders may include other projects, organizational units, sub-contractors, and service providers.

MAN.3.BP8: Define and monitor the project schedule. Allocate resources to the work packages and schedule each activity of the project. Monitor the performance of the activities against the schedule.

MAN.3.BP9: Ensure consistency. Regularly adjust the estimates, resources, skills, work packages and their dependencies, schedules, plans, interfaces, and commitments for the project to ensure consistency with the scope of work.

Note 9: Critical dependencies, that are an input for risk management may also be considered.

MAN.3.BP10: Review and report the progress of the project. Regularly review and report the status of the project and the fulfillment of the work packages against estimated effort and duration to all affected parties.

Note 10: Project reviews may be executed at regular intervals by the management. Project reviews may contribute to identify best practices and lessons learned.

MAN.3 Project Management	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6	Outcome 7
Output Information Items							
08-53 Scope of work	X						
08-54 Feasibility analysis		X		X			
14-10 Work package			X	X	X		
13-52 Communication evidence		X	X				

13-16 Change request							X
13-51 Consistency evidence		X					X
14-02 Corrective action						X	X
18-52 Escalation path				X		X	X
08-56 Schedule			X		X		X
14-50 Stakeholder groups				X			
15-06 Project status			X	X		X	
Base Practices							
BP1: Define the scope of work	X						
BP2: Define the project lifecycle	X	X					
BP3: Evaluate the feasibility of the project		X					
BP4: Define and monitor the work packages			X	X	X		
BP5: Define and monitor the project estimates and resources		X	X				
BP6: Define and monitor the required skills, knowledge, and experience			X				
BP7: Define and monitor the project interfaces and agreed commitments			X	X			
BP8: Define and monitor the project schedule						X	
BP9: Ensure consistency			X	X	X		X
BP10: Review and report the progress of the project						X	

4.9.2. MAN.5 Risk Management

Process ID
MAN.5
Process name
Risk Management
Process purpose
The purpose is to regularly identify, analyze, treat and monitor the process related risks and the product related risks.
Process outcomes
<ol style="list-style-type: none"> 1) The sources of risks are identified and regularly updated. 2) Potential undesirable events are identified as they develop during the conduct of the project. 3) The risks are analyzed and the priority in which to apply resources to treatment of these risks is determined. 4) The risk measures are defined, applied, and assessed to determine changes in the statuses of the risks and the progress of the risk treatment activities.

5) Appropriate treatment is taken to correct or avoid the impact of risk based on its priority, probability, and consequence or other defined risk threshold.

Base Practices

MAN.5.BP1: Identify sources of risks. Identify and regularly update the sources of risks with affected parties.

Note 1: Risks may include technical, economical, and schedule risks.

Note 2: Risks may include the suppliers' deliverables and services.

Note 3: The risk sources may vary across the entire project lifecycle.

MAN.5.BP2: Identify the potential undesirable events. Identify the potential undesirable events within the scope of the risk management for the project.

MAN.5.BP3: Determine the risks. Determine the risks by analyzing the probability, consequence and severity of the potential undesirable events to support priorities for the mitigation of the risks.

Note 4: Different methods may be used to analyze technical risks of a system, for example, functional analysis, simulation, FMEA, FTA etc.

MAN.5.BP4: Define risk treatment options. For each risk define a treatment option to accept, mitigate, avoid, or share (transfer) the risk.

MAN.5.BP5: Define and perform risk treatment activities. Define and perform risk activities for the risk treatment options.

MAN.5.BP6: Monitor the risks. Regularly re-evaluate the risks to determine changes in the status of a risk and to evaluate the progress of the risk treatment activities.

Note 5: Risks of high priority may need to be communicated to and monitored by higher levels of management.

MAN.5.BP7: Take corrective action. When risk treatment activities are not effective, take appropriate corrective action.

Note 6: Corrective actions may involve reevaluation of risks, developing and implementing new mitigation concepts or adjusting the existing concepts.

MAN.5 Risk Management	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5
Output Information Items					
15-51 Analysis results	X	X	X		X
15-09 Risk status	X		X	X	X
08-55 Risk measure				X	X
14-02 Corrective action				X	X

Base Practices					
BP1: Identify sources of risks	X				
BP2: Identify the potential undesirable events		X			
BP3: Determine the risks			X		
BP4: Define risk treatment options				X	X
BP5: Define and perform risk treatment activities.				X	X
BP6: Monitor the risks				X	
BP7: Take corrective action					X

4.9.3. MAN.6 Measurement

Process ID
MAN.6
Process name
Measurement
Process purpose
The purpose is to collect and analyze data relating to the development results and processes implemented within the organization and its projects, to support an effective management of the processes.
Process outcomes
<ol style="list-style-type: none"> 1) The measurement information needs that are necessary to evaluate the achievement of process objectives and the achievement of desired work products are identified. 2) An appropriate set of metrics, driven by the information needs, are identified and/or developed. 3) Measurement activities are identified and performed. 4) The required metrics are collected, stored, analyzed, and the results interpreted. 5) Metrics are used to support decisions and provide an objective basis for communication.

Base Practices
<p>MAN.6.BP1: Identify the information needs. Identify the measurement information needs that are necessary to evaluate the achievement of process objectives and work products.</p> <p><i>Note 1: Information needs may change over time. Therefore, the measurement process may be used in an iterative way.</i></p> <p><i>Note 2: The information needs should be aligned with the stakeholders and decision makers.</i></p>
<p>MAN.6.BP2: Specify metrics. Identify and develop an appropriate set of metrics based on the measurement information needs.</p> <p><i>Note 3: Metrics may be related to processes or development results.</i></p>

MAN.6.BP3: Collect and store the metrics. Collect and store both base and derived metrics, including any context information necessary to verify and understand the metrics.

Note 4: Base metrics in the context of this process are directly gathered metrics like “number of defects found” or “number of lines of code”, where derived metrics are two or more metrics that are brought in relation to each other like “number of defects found per line of code”.

MAN.6.BP4: Analyze the collected metrics. Analyze, interpret and review the measured values to support decision-making.

MAN.6.BP5: Communicate the analysis results. Communicate the analysis results to all affected parties.

MAN.6.BP6: Use metrics for decision-making. Make accessible and use information from the collected metrics and analysis results for any decision-making process for which it is relevant.

MAN.6 Measurement	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5
Output Information Items					
03-03 Benchmarking data				X	X
03-04 Customer satisfaction data				X	X
03-06 Process performance information				X	X
07-51 Measurement result			X	X	X
15-51 Analysis results	X			X	X
Base Practices					
BP1: Identify the information needs	X				
BP2: Specify metrics		X	X		
BP3: Collect and store the metrics			X	X	
BP4: Analyze the collected metrics				X	X
BP5: Communicate the measurement information					X
BP6: Use metrics for decision-making					X

4.10. Process Improvement process group (PIM)

4.10.1.PIM.3 Process Improvement

Process ID
PIM.3
Process name
Process Improvement
Process purpose
The purpose is to continually improve the organization's effectiveness and efficiency through the processes used and to ensure alignment of the processes with the business needs.
Process outcomes
<ol style="list-style-type: none"> 1) A commitment is established to provide resources to sustain improvement measures. 2) Issues arising from the organization's internal or external environment are identified as improvement opportunities and justified as reasons for change. 3) An analysis of the current status of the existing processes is performed. 4) Improvement goals are identified and prioritized, and consequent changes to the processes are defined, documented and implemented. 5) The effects of process implementations are monitored, measured and confirmed in relation to the identified improvement goals. 6) The knowledge gained from the improvement is communicated within the organization.

Base Practices
<p>PIM.3.BP1: Establish a commitment. Establish a commitment to support the process improvement staff, to provide resources and further enablers to sustain improvement actions.</p> <p><i>Note 1: The process improvement process is a generic process, which can be used at all levels (e.g., organizational level, process level, project level, etc.) and which can be used to improve all processes.</i></p> <p><i>Note 2: Commitment at all levels of management may support process improvement.</i></p> <p><i>Note 3: Enablers for improvement measures may include trainings, methods, infrastructure, etc.</i></p>
<p>PIM.3.BP2: Identify improvement measures. Identify issues from the analysis of process performance and derive improvement opportunities with justified reasons for change.</p> <p><i>Note 4: Analysis may include problem report trend analyses (see SUP.9), analyses from Quality Assurance and Verification results and records (see SUP.1), validation results and records, and product quality metrics like defect rate.</i></p> <p><i>Note 5: Issues and improvement suggestions may be addressed by the customer.</i></p> <p><i>Note 6: Sources for identification of issues may include: process assessment results, audits, customer's satisfaction reports, measurements of organizational effectiveness/efficiency, costs of quality.</i></p>
<p>PIM.3.BP3: Establish process improvement goals. Analyze the current status of the existing processes and establish improvement goals.</p>

Note 7: The current status of processes may be determined by a process assessment.

PIM.3.BP4: Prioritize the improvements. Prioritize the improvement goals and improvement measures.

PIM.3.BP5: Define process improvement measures. Process improvement measures are defined.

Note 8: Improvements may be documented in incremental steps.

PIM.3.BP6: Implement the process improvement measures. Implement and apply the improvements to the processes. Update the process documentation and train people as needed.

Note 9: Process application can be supported by establishing policies, adequate process infrastructure, process training, process coaching and tailoring processes to local needs.

Note 10: Improvements may be piloted before rolling out within the organization.

PIM.3.BP7: Confirm the process improvements. The effects of process implementation are monitored and measured, and the achievement of defined improvement goals is confirmed.

PIM.3.BP8: Communicate the results of the improvements. Knowledge gained from the improvements and progress of the improvement implementation is communicated to all affected parties.

PIM.3 Process Improvement	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6
Output Information Items						
02-01 Commitment/agreement	X					
06-04 Training material				X		X
07-04 Process metric					X	X
10-00 Process description				X		
13-52 Communication evidence						X
13-16 Change request		X				
15-51 Analysis result		X	X	X	X	
15-13 Assessment/audit result			X		X	
15-16 Improvement opportunity		X	X	X		
16-06 Process repository				X		
Base Practices						
BP1: Establish a commitment	X					
BP2: Identify improvement measures		X	X			
BP3: Establish process improvement goals				X		
BP4: Prioritize the improvements				X		

BP5: Define process improvement measures				X		
BP6: Implement the process improvement measures				X		
BP7: Confirm the process improvements					X	
BP8: Communicate the results of the improvements						X

4.11. Reuse process group (REU)

4.11.1.REU.2 Reuse of Products

Process ID
REU.2
Process name
Reuse of Products
Process purpose
The purpose is to ensure that the reused (work) products are analyzed, verified, and approved for their target context.
Process outcomes
<ol style="list-style-type: none"> 1) Products for reuse are selected using defined criteria. 2) The products for reuse are analyzed for portability and interoperability. 3) The limitations for reuse are defined and communicated. 4) The products for reuse are verified. 5) The products for reuse are provided to affected parties. 6) A communication mechanism is established with the reuse product provider.

Base Practices
<p>REU.2.BP1: Select products for reuse. Select the products to be reused using defined criteria.</p> <p><i>Note 1: Products for reuse may be systems, hardware or software components, third party components or legacy components.</i></p>
<p>REU.2.BP2: Analyze the reuse capability of the products. Analyze the designated target architecture and the products to be reused to determine their applicability in the target architecture according to relevant criteria.</p> <p><i>Note 2: The target architecture may be, e.g., system architecture, hardware architecture or software architecture.</i></p> <p><i>Note 3: Examples for criteria can be requirements compliance, verifiability of the products to be reused in the target architecture, or portability/interoperability.</i></p>
<p>REU.2.BP3: Define the limitations for reuse. Define and communicate the limitations for the products to be reused.</p> <p><i>Note 4: Limitations may address parameters of the operational environment.</i></p>

REU.2.BP4: Ensure the qualification of the products for reuse. Provide evidence that the products for reuse are qualified for the intended use of the deliverable.

Note 5: Qualification may be demonstrated by verification evidence.

Note 6: Verification may include the appropriateness of documentation.

REU.2.BP5: Provide the products for reuse. Make available the products to be reused to affected parties.

Note 7: Refer to HWE.3, SWE.5 or SYS.4 for more information on integration of hardware, software, or system components.

REU.2.BP6: Communicate information about the effectiveness of the reuse activities. Establish a communication and notification mechanism about experiences and technical outcomes to the provider of reused products.

Note 8: The communication with the provider of a reused product may depend on whether the product is under development or not.

REU.2 Management of Products for Reuse	Outcome 1	Outcome 2	Outcome 3	Outcome 4	Outcome 5	Outcome 6
Output Information Items						
12-03 Reuse candidate	X				X	
04-02 Domain architecture		X				
13-52 Communication evidence						X
15-07 Reuse analysis evidence		X	X			
13-53 Qualification evidence				X		
Base Practices						
BP1: Select products for reuse	X					
BP2: Analyze the reuse capability of the products		X				
BP3: Define the limitations for reuse			X			
BP4: Ensure the qualification of the products for reuse				X		
BP5: Provide the products for reuse					X	
BP6: Communicate information about the effectiveness of the reuse activities						X

5. Process capability levels and process attributes

The definition of process capability indicators for each process attribute is an integral part of a measurement framework. Process capability indicators such as generic practices and information items are the means to support the judgment of the degree of achievement of the associated process attribute.

This chapter defines the generic practices and information items and their mapping to the process attributes for each capability level defined in the measurement framework [3.2].

Note: Due to lack of a defined process attribute for process capability level 0, no generic practices and information items are defined.

Process capability level	<p>Process attribute ID</p> <p>Process attribute name</p> <p>Process attribute scope</p> <p>Process attribute achievements</p>	<p>Each process attribute is identified with a unique identifier and name. A process attribute scope statement is provided, and process achievements are defined.</p>
	<p>Generic practices</p>	<p>A set of generic practices for the process attribute providing a definition of the activities to be performed to accomplish the process attribute scope and fulfill the process achievements.</p> <p>The generic practice headers are summarized at the end of a process to demonstrate their relationship to the process attribute achievements.</p>
Process attribute achievement indicators	<p>Output information items</p>	<p>The output information items that are relevant to accomplish the process attribute scope and fulfill the process achievements are summarized at the end of a process attribute section to demonstrate their relationship to the process achievements.</p> <p><i>Note: Refer to Annex B for the characteristics of each information item.</i></p>

Table 11 — Template for the process description

5.1. Process capability level 0: Incomplete process

The process is not implemented or fails to achieve its process purpose. At this level there is little or no evidence of any systematic achievement of the process purpose.

5.2. Process capability Level 1: Performed process

The implemented process achieves its process purpose. The following process attribute demonstrates the achievement of this level.

5.2.1. PA 1.1 Process performance process attribute

Process attribute ID	
PA 1.1	
Process attribute name	
Process performance	
Process attribute scope	
The process performance process attribute is a measure of the extent to which the process purpose is achieved.	
Process attribute achievements	
1) The process achieves its defined outcomes.	
Generic practices	
GP 1.1.1 Achieve the process outcomes	
Achieve the intent of the base practices. Produce work products that evidence the process outcomes.	
PA 1.1 Process performance process attribute	Achievement a
Output Information Items	
Process specific information items, as described in chapter 4	X
Generic practices	
GP 1.1.1 Achieve the process outcomes	X

5.3. Process capability Level 2: Managed process

The following process attributes, together with the previously defined process attribute, demonstrate the achievement of this level.

5.3.1. PA 2.1 Process performance management process attribute

Process attribute ID
PA 2.1
Process attribute name
Process performance management
Process attribute scope
The performance management process attribute is a measure of the extent to which the performance of the process is managed.
Process attribute achievements
<ol style="list-style-type: none"> 1) A strategy for the performance of the process is defined based on the identified objectives. 2) The performance of the process is planned. 3) The performance of the process is monitored and adjusted to meet the planning. 4) The needs for human resources including responsibilities and authorities for performing the process are determined. 5) The needs for physical and material resources are determined. 6) The persons performing the process are prepared for executing their responsibilities. 7) The physical and material resources for performing the process are identified, made available, allocated and used. 8) The interfaces between the involved parties are managed to ensure both effective communication and the assignment of responsibilities.

Generic practices
<p>GP 2.1.1: Identify the objectives and define a strategy for the performance of the process.</p> <p>The scope of the process activities including the management of process performance and the management of work products are determined.</p> <p>Corresponding results to be achieved are determined.</p> <p>The process performance objectives and associated criteria are identified.</p> <p><i>Note 1: Process performance objectives are not identical to project goals, nor are they a repetition of process outcomes at capability level 1. Examples:</i></p> <ul style="list-style-type: none"> • SUP.8: Max. ratio of configuration items in status “in work” at certain milestones before next delivery / release date • SYS.2: End of req writing = release date - x weeks • SWE.6: Test coverage of Functional Safety Requirements at baseline y is 100% • SUP.10: all incoming CRs are analyzed within z days <p><i>Note 2: Performance objectives are the basis for planning and monitoring.</i></p> <p>Assumptions and constraints are considered when identifying the performance objectives.</p> <p>The approach and methodology for the process performance is determined.</p> <p><i>Note 3: A process performance strategy may not necessarily be documented specifically for each process. Elements applicable for multiple processes may be documented jointly, e.g., as part of a</i></p>

common project handbook or in a joint test strategy.

GP 2.1.2: Plan the performance of the process.

The planning for the performance of the process is established according to the defined objectives, criteria, and strategy.

The process activities and work packages are defined.

The estimates for the work packages are identified using appropriate methods.

Note 4: Schedule and milestones are defined.

GP 2.1.3: Determine the resource needs.

The required amount of human resources, and experience, knowledge and skill needs for the process performance are determined based on the planning.

The needs for physical and material resources are determined based on the planning.

Note 5: Physical and material resources may include equipment, laboratories, materials, tools, licenses etc.

The required responsibilities and authorities to perform the process, and to manage the corresponding work products are determined.

Note 6: The definition of responsibilities and authorities does not necessarily require formal role descriptions.

GP 2.1.4: Identify and make available the resources.

The individuals performing and managing the process are identified and allocated according to the determined needs.

The individuals performing and managing the process are being qualified to execute their responsibilities.

Note 7: Qualification of individuals may include training, mentoring, or coaching.

The other resources, necessary for performing the process are identified, made available, allocated and used according to the determined needs.

GP 2.1.5: Monitor and adjust the performance of the process.

The process performance is monitored to identify deviations from the planning.

Appropriate actions in case of deviations from the planning are taken.

The planning is adjusted as necessary.

GP 2.1.6: Manage the interfaces between involved parties.

The individuals and groups including required external parties involved in the process performance are determined.

The responsibilities are assigned to the relevant individuals or parties.

The communication mechanisms between the involved parties are determined.

An effective communication between the involved parties is established and maintained.

PA 2.1 Process performance management process attribute	Achievement 1	Achievement 2	Achievement 3	Achievement 4	Achievement 5	Achievement 6	Achievement 7	Achievement 8
Output Information Items								
19-01 Process performance strategy	X							
18-58 Process performance objectives	X							
14-10 Work package		X						
08-56 Schedule		X	X					
13-14 Progress status			X					
17-55 Resource needs				X	X			
08-61 Resource allocation						X	X	
08-62 Communication matrix								X
13-52 Communication evidence								X
Generic Practices								
GP 2.1.1: Identify the objectives and define a strategy for the performance of the process	X							
GP 2.1.2: Plan the performance of the process		X						
GP 2.1.3: Determine the resource needs				X	X			
GP 2.1.4: Identify and make available the resources						X	X	
GP 2.1.5: Monitor and adjust the performance of the process			X					
GP 2.1.6: Manage the interfaces between involved parties								X

5.3.2. PA 2.2 Work product management process attribute

Process attribute ID
PA 2.2
Process attribute name
Work product management
Process attribute scope
The work product management process attribute is a measure of the extent to which the work products produced by the process are appropriately managed.
Process attribute achievements
<ol style="list-style-type: none"> 1) The requirements for the work products of the process are defined. 2) The requirements for storage and control of the work products are defined. 3) The work products are appropriately identified, stored, and controlled. 4) The work products are reviewed and adjusted as necessary to meet requirements.
Generic practices
<p>GP 2.2.1 Define the requirements for the work products.</p> <p>The requirements for the content and structure of the work products to be produced are defined. Quality criteria for the work products are identified. Appropriate review and approval criteria for the work products are defined.</p> <p><i>Note 1: Possible sources of documentation requirements may be, e.g., best practices or lessons learned from other projects, standards, organization requirements, customer requirements, etc.</i></p> <p><i>Note 2: There may be types of work products for which no review or approval is required, thus then there would be no need to define the corresponding criteria.</i></p>
<p>GP 2.2.2 Define the requirements for storage and control of the work products.</p> <p>The requirements for the storage and control of the work products are defined, including their identification and distribution.</p> <p><i>Note 3: Possible sources for the identification of requirements for storage and control may be, e.g., legal requirements, data policies, best practices from other projects, tool related requirements, etc.</i></p> <p><i>Note 4: Examples for work product storage are files in a file system, tickets in a tool, Wiki entries, paper documents etc.</i></p> <p><i>Note 5: Where the status of a work product is required in base practices, this should be managed via a defined status model.</i></p>

GP 2.2.3 Identify, store and control the work products.

The work products to be controlled are identified.

The work products are stored and controlled in accordance with the requirements.

A change control is established for work products.

Versioning and baselining of the work products is performed in accordance with the requirements for storage and control of the work products.

The work products including the revision status are made available through appropriate mechanisms.

GP 2.2.4 Review and adjust work products.

The work products are reviewed against the defined requirements and criteria.

Resolution of issues arising from work products reviews is ensured.

PA 2.2 Work product management process attribute	Achievement 1	Achievement 2	Achievement 3	Achievement 4
Output Information Items				
17-05 Requirements for work products	X	X		
18-59 Review and approval criteria for work products	X			
18-07 Quality criteria	X			
13-19 Review evidence				X
13-08 Baseline			X	
16-03 Configuration management system			X	
Generic Practices				
GP 2.2.1 Define the requirements for the work products	X			
GP 2.2.2 Define the requirements for storage and control of the work products		X		
GP 2.2.3 Identify, store and control the work products			X	
GP 2.2.4 Review and adjust work products.				X

5.4. Process capability Level 3: Established process

The following process attributes, together with the previously defined process attributes, demonstrate the achievement of this level.

5.4.1. PA 3.1 Process definition process attribute

Process attribute ID
PA 3.1
Process attribute name
Process definition
Process attribute scope
The process definition process attribute is a measure of the extent to which a standard process is maintained to support the deployment of the defined process.
Process attribute achievements
<ol style="list-style-type: none"> 1) A standard process is developed, established, and maintained that describes the fundamental elements that must be incorporated into a defined process. 2) The required inputs and the expected outputs for the standard process are defined. 3) The roles, responsibilities, authorities, and required competencies for performing the standard process are defined. 4) Tailoring guidelines for deriving the defined process from the standard process are defined. 5) The required physical and material resources and process infrastructure needs are determined as part of the standard process. 6) Suitable methods and required activities for monitoring the effectiveness, suitability and adequacy of the process are determined.
Generic practices
<p>GP 3.1.1 Establish and maintain the standard process.</p> <p>A suitable standard process is developed including the required activities and their interactions comprising:</p> <ul style="list-style-type: none"> - The inputs and outputs of the standard process are defined including the interactions and sequence with other processes. - The process performance roles are identified and assigned to the standard process activities including their type of involvement, responsibilities, and authorities. <p><i>Note 1: An example for describing the involvement of the process roles in the activities is a RASI/RASIC representation.</i></p> <p>Sufficiently detailed guidance, procedures, and templates are provided to support the execution of the process as needed.</p> <p><i>Note 2: Procedures may also include descriptions of specific methods to be used.</i></p>

Appropriate tailoring guidelines including predefined unambiguous criteria as well as predefined and unambiguous proceedings are defined based on identified deployment needs and the context of the standard process.

The standard process is maintained according to corresponding feedback from the monitoring of the deployed processes.

Note 3: For guidance on how to perform process improvements see the Process Improvement process (PIM.3).

GP 3.1.2 Determine the required competencies.

The required competencies, skills, and experience for performing the standard process are determined for the identified roles.

Appropriate qualification methods to acquire the necessary competencies and skills are determined, maintained, and made available for the identified roles.

Note 4: Qualification methods are, e.g., trainings, mentoring, self-study.

Note 5: Preparation includes, e.g., identification or definition of trainings, mentoring concepts, self-learning material.

GP 3.1.3 Determine the required resources.

The required physical and material resources and process infrastructure needs for performing the standard process are determined.

Note 6: This may include, e.g., facilities, tools, licenses, networks, services, and samples supporting the establishment of the required work environment.

GP 3.1.4 Determine suitable methods to monitor the standard process.

Methods and required activities for monitoring the effectiveness and adequacy of the standard process are determined.

Note 7: Methods and activities to gather feedback regarding the standard process may be lessons learned, process compliance checks, internal audits, management reviews, change requests, reflection of state-of-the-art such as applicable international standards, etc.

Appropriate criteria and information needed to monitor the standard process are defined.

Note 8: Information about process performance may be of qualitative or quantitative nature.

PA 3.1 Process definition process attribute	Achievement ₁	Achievement ₂	Achievement ₃	Achievement ₄	Achievement ₅	Achievement ₆
Output Information Items						
06-51 Tailoring guideline				X		
08-63 Process monitoring method						X
10-00 Process description	X	X				
10-50 Role description			X			
10-51 Qualification method description			X			

10-52 Process resource and infrastructure description					X	
Generic Practices						
GP 3.1.1 Establish and maintain the standard process	X	X	X	X		
GP 3.1.2 Determine the required competencies			X			
GP 3.1.3 Determine the required resources					X	
GP 3.1.4 Determine suitable methods to monitor the standard process						X

5.4.2. PA 3.2 Process deployment process attribute

Process attribute ID
PA 3.2
Process attribute name
Process deployment
Process attribute scope
The process deployment process attribute is a measure of the extent to which the standard process is deployed as a defined process to achieve its process outcomes.
Process attribute achievements
<ol style="list-style-type: none"> 1) A defined process is deployed based upon an appropriately selected and/or tailored standard process. 2) An assignment of persons necessary for performing the defined process to roles is performed and communicated. 3) The required education, training and experience is ensured and monitored for the persons assigned to the roles. 4) The required resources for performing the defined process are made available, allocated, and maintained. 5) Appropriate information is collected and analyzed as a basis for understanding the behavior of the process.
Generic practices
<p>GP 3.2.1 Deploy a defined process that satisfies the context specific requirements of the use of the standard process.</p> <p>The defined process is appropriately selected and/or tailored from the standard process.</p> <p>The conformance of the defined process with standard process requirements and tailoring criteria is verified.</p> <p>The defined process is used as managed process to achieve the process outcomes.</p> <p><i>Note 1: Changes in the standard process may require updates of the defined process.</i></p>
<p>GP 3.2.2 Ensure the required competencies for the defined roles.</p>

The human resources are allocated to the defined roles according to the required competencies and skills.

The assignment of persons to roles and corresponding responsibilities and authorities for performing the defined process is communicated.

The gaps in competencies and skills are identified, and the corresponding qualification measures are initiated and monitored.

Availability and usage of the project staff are measured and monitored.

GP 3.2.3 Ensure the required resources to support the performance of the defined process.

The required information to perform the defined process is made available, allocated and used.

The required physical and material resources, process infrastructure and work environment are made available, allocated and used.

Availability and usage of the resources are measured and monitored.

GP 3.2.4 Monitor the performance of the defined process.

Information is collected and analyzed according to the determined process monitoring methods to understand the effectiveness and adequacy of the defined process.

The results of the analysis are made available to all involved parties and used to identify where continual improvement of the standard and/or defined process can be made.

Note 2: For guidance on how to perform process improvements see the Process Improvement process (PIM.3).

	Achievement 1	Achievement 2	Achievement 3	Achievement 4	Achievement 5
PA 3.2 Process deployment process attribute					
Output Information Items					
10-00 Process description	X				
15-54 Tailoring documentation	X				
14-53 Role assignment		X	X		
13-55 Process resource and infrastructure documentation				X	
03-06 Process performance information					X
Generic Practices					
GP 3.2.1 Deploy a defined process	X				
GP 3.2.2 Ensure the required competencies		X	X		
GP 3.2.3 Ensure the required resources				X	
GP 3.2.4 Monitor the performance of the defined process					X

5.5. Process capability Level 4: Predictable process

The following process attributes, together with the previously defined process attributes, demonstrate the achievement of this level.

5.5.1. PA 4.1 Quantitative analysis process attribute

Process attribute ID
PA 4.1
Process attribute name
Quantitative analysis
Process attribute scope
The quantitative analysis process attribute is a measure of the extent to which information needs are defined, relationships between process elements are identified and data are collected.
Process attribute achievements
<ol style="list-style-type: none"> 1) The process information needs in support of relevant defined quantitative business goals are established. 2) Measurable relationships between process elements that contribute to the process performance, and data collection techniques and data collection frequency, are identified. 3) Process measurement objectives are derived from the process information needs. 4) Techniques for analyzing the collected data are selected. 5) Quantitative control limits for the process performance in support of relevant business goals are established. 6) The results of measurement are collected, validated and reported in order to monitor the extent to which the quantitative targets/objectives for the process performance are met. <p><i>Note: Information needs typically reflect management, technical, project, process or product needs.</i></p>

Generic practices
<p>GP 4.1.1 Identify the business goals.</p> <p>The business goals are identified that are supported by the quantitatively measured process.</p>
<p>GP 4.1.2 Establish the process information needs.</p> <p>The stakeholders of the identified business goals and the quantitatively measured process are identified, and their information needs are defined and agreed.</p>
<p>GP 4.1.3 Identify the measurable relationships between process elements.</p> <p>Identify the relationships between process elements, or sets of process elements, which contribute to the process information needs.</p> <p><i>Note 1: Examples of process elements are work products, activities, tasks.</i></p>

GP 4.1.4 Derive a process measurement approach and select the analysis techniques.

Based on the measurable relationships of process elements, or set of process elements, the process measurement metrics are derived to satisfy the established process information needs.

The frequency of data collection is defined.

Select the analysis techniques, appropriate to the collected data.

Algorithms and methods to create derived measurement results from base measures are defined, as appropriate.

The verification mechanism for base and derived measures is defined.

Note 2: Typically, the standard process definition is extended to include the collection of data for process measurement.

GP 4.1.5 Establish quantitative control limits.

Establish quantitative control limits for the derived metrics. An agreement with the process stakeholders is established.

GP 4.1.6 Collect product and process measurement results through performing the defined process.

Data collection mechanisms are created for all identified metrics.

The required data are collected across process instances with the defined frequency and recorded.

The measurement results are analyzed and reported to the identified stakeholders.

Note 3: A product measure can contribute to a process measure, e.g., the productivity of testing characterized by the number of defects found in a given timeframe in relation to the product defect rate in the field.

PA 4.1 Quantitative analysis process attribute	Achievement 1	Achievement 2	Achievement 3	Achievement 4	Achievement 5	Achievement 6
Output Information Items						
18-70 Business goals	X	X				
07-61 Quantitative process metric		X	X			
07-62 Process analysis techniques				X		
07-63 Process control limits					X	
07-64 Process measurement data						X
Generic Practices						
GP 4.1.1 Identify the business goals	X					
GP 4.1.2 Establish the process information needs	X					

GP 4.1.3 Identify the measurable relationships between process elements		X				
GP 4.1.4 Derive a process measurement approach and select the analysis techniques			X	X		
GP 4.1.5 Establish quantitative control limits					X	
GP 4.1.6 Collect product and process measurement results through performing the de-fined process						X

5.5.2. PA 4.2 Quantitative control process attribute

Process attribute ID
PA 4.2
Process attribute name
Quantitative control
Process attribute scope
The quantitative control process attribute is a measure of the extent to which objective data are used to manage process performance that is predictable.
Process attribute achievements
<ol style="list-style-type: none"> 1) The variations in process performance are identified. 2) The assignable causes of process variations are determined through analysis of the collected quantitative data. 3) The distributions that characterize the performance of the process are established. 4) Corrective actions are taken to address assignable causes of variation.
Generic practices
GP 4.2.1 Identify the variations in process performance.
The deviations in the performance of process instances from the established quantitative control limits are determined based on the collected quantitative measurement data.
GP 4.2.2 Identify the causes of variation.
The determined deviations in the process performance are analyzed to identify potential causes of variation using the defined analysis techniques.
Distributions are used to quantitatively understand the variation of process performance under the influence of potential causes of variation.
The consequences of process variation are analyzed.

GP 4.2.3 Identify and implement corrective actions to address assignable causes.

The results of the variation analyses are provided to those responsible for taking action.

Corrective actions are determined and implemented to address each assignable cause of variation.

The corrective action results are monitored and evaluated to determine their effectiveness.

Note 1: Assignable cause may indicate a possible problem in the defined process.

PA 4.2 Quantitative control process attribute	Achievement 1	Achievement 2	Achievement 3	Achievement 4
Output Information Items				
15-57 Quantitative process analysis results	X	X	X	
08-66 Measures against deviations in quantitative process analysis				X
Generic Practices				
GP 4.2.1 Identify the variations in process performance	X			
GP 4.2.2 Identify the causes of variation		X	X	
GP 4.2.3 Identify and implement corrective actions to address assignable causes				X

5.6. Process capability Level 5: Innovating process

The following process attributes, together with the previously defined process attributes, demonstrate the achievement of this level.

5.6.1. PA 5.1 Process innovation process attribute

Process attribute ID
PA 5.1
Process attribute name
Process innovation
Process attribute scope
The process innovation process attribute is a measure of the extent to which changes to the process are identified from investigations of innovative approaches to the definition and deployment of the process.
Process attribute achievements
<ol style="list-style-type: none"> 1) The process innovation objectives are defined that support the relevant business goals. 2) The quantitative data are analyzed to identify opportunities for innovation. 3) Innovation opportunities derived from new technologies and process concepts are identified.
Generic practices
<p>GP 5.1.1 Define the process innovation objectives for the process that support the relevant business goals.</p> <p>New business visions and goals are analyzed to give guidance for new process objectives and potential areas of process innovation.</p> <p>The quantitative and qualitative process innovation objectives are defined and documented.</p>
<p>GP 5.1.2 Analyze the quantitative data of the process.</p> <p>The common causes of variation in process performance across process instances are identified and analyzed to get a quantitative understanding of their impact.</p>
<p>GP 5.1.3 Identify the innovation opportunities.</p> <p>Identify the opportunities for innovation based on the quantitative understanding of the analyzed data.</p> <p>Industry best practices, new technologies and process concepts are identified and evaluated. Feedback on opportunities for innovation is actively sought.</p> <p>Emergent risks are considered in evaluating the improvement opportunities.</p>

PA 5.1 Process innovation process attribute	Achievement 1	Achievement 2	Achievement 3
Output Information Items			
18-80 Process improvement opportunity	X		X
15-58 Common cause of variation analysis results		X	
Generic Practices			
GP 5.1.1 Define the process innovation objectives for the process that support the relevant business goals	X		
GP 5.1.2 Analyze the quantitative data of the process		X	
GP 5.1.3 Identify the innovation opportunities			X

5.6.2. PA 5.2 Process innovation implementation process attribute

Process attribute ID
PA 5.2
Process attribute name
Process innovation implementation
Process attribute scope
The process innovation process implementation attribute is a measure of the extent to which changes to the definition, management and performance of the process achieves the relevant process innovation objectives.
Process attribute achievements
<ol style="list-style-type: none"> 1) The impact of all proposed changes is assessed against the objectives of the defined process and standard process. 2) The implementation of all agreed changes is managed to ensure that any disruption to the process performance is understood and acted upon. 3) The effectiveness of process changes on the basis of quantitative performance and innovation feedback is evaluated.
Generic practices
<p>GP 5.2.1 Define and assess the impact of the proposed changes.</p> <p>The specified changes are assessed against product quality and process performance requirements and goals.</p> <p>The impact of changes on other defined and standard processes is considered.</p> <p>Objective priorities for process innovation are established.</p> <p>The commitment to innovation is demonstrated by organizational management including other relevant stakeholders.</p>
<p>GP 5.2.2 Implement the agreed process changes.</p> <p>A mechanism is established for incorporating accepted changes into the defined and standard process(es) effectively and completely.</p> <p>The process changes are implemented and effectively communicated to all involved parties.</p>

GP 5.2.3 Evaluate the effectiveness of the process changes.

Performance and capability of the changed process are measured and compared with historical data.

Performance and capability of the changed process are analyzed to determine whether the process performance has improved with respect to common causes of variations.

Other feedback is recorded, such as opportunities for further innovation of the standard process.

A mechanism is available for documenting and reporting the analysis results to stakeholders of the standard and defined process.

PA 5.2 Process innovation implementation process attribute	Achievement 1	Achievement 2	Achievement 3
Output Information Items			
18-81 Improvement evaluation results	X		X
08-66 Measures against deviations in quantitative process analysis		X	X
Generic Practices			
GP 5.2.1 Define and assess the impact of the proposed changes	X		
GP 5.2.2 Implement the agreed process changes		X	
GP 5.2.3 Evaluate the effectiveness of the process changes			X

Annex A Conformity statements

Annex A.1 Introduction

The Automotive SPICE® process assessment and process reference model meet the requirements for conformance defined in ISO/IEC 33004:2015. The process assessment model can be used in the performance of assessments that meet the requirements of ISO/IEC 33002:2015.

This clause serves as the statement of conformance of the process assessment and process reference models to the requirements defined in ISO/IEC 33004:2015.

| *[ISO/IEC 33004:2015, 5.5 and 6.4]*

Due to copyright reasons each requirement is only referred by its number. The full text of the requirements can be drawn from ISO/IEC 33004:2015.

Annex A.2 Conformance to the requirements for process reference models

Clause 5.3, "Requirements for process reference models"

The following information is provided in chapter 1 and 3 of this document:

- the declaration of the domain of this process reference model;
- the description of the relationship between this process reference model and its intended context of use; and
- the description of the relationship between the processes defined within this process reference model.

The descriptions of the processes within the scope of this process reference model meeting the requirements of ISO/IEC 33004:2015 clause 5.4 is provided in chapter 4 of this document.

| *[ISO/IEC 33004:2015, 5.3.1]*

The relevant communities of interest and their mode of use and the consensus achieved for this process reference model is documented in the copyright notice and the scope of this document.

| *[ISO/IEC 33004:2015, 5.3.2]*

The process descriptions are unique. The identification is provided by unique names and by the identifier of each process of this document.

| *[ISO/IEC 33004:2015, 5.3.3]*

Clause 5.4, "Process descriptions"

These requirements are met by the process descriptions in chapter 4 of this document.

| *[ISO/IEC 33004:2015, 5.4]*

Annex A.3 Conformance to the requirements for process assessment models

Clause 6.1, "Introduction"

The purpose of this process assessment model is to support assessment of process capability within the automotive domain using the defined process measurement framework.

| [ISO/IEC 33004:2015, 6.1]

Clause 6.2, "Process assessment model scope"

The process scope of this process assessment model is defined in the process reference model included in chapter 3.1 of this document. The Automotive SPICE® process reference model is satisfying the requirements of ISO/IEC 33004:2015, clause 5 as described in Annex A.2.

The process capability scope of this process assessment model is defined in the process measurement framework, which defines a process measurement framework for process capability satisfying the requirements of ISO/IEC 33003:2015.

| [ISO/IEC 33004:2015, 6.2]

Clause 6.3, "Requirements for process assessment models"

The Automotive SPICE® process assessment model is related to process capability.

| [ISO/IEC 33004:2015, 6.3.1]

This process assessment model incorporates the defined process measurement framework, which satisfies the requirements of ISO/IEC 33003:2015.

| [ISO/IEC 33004:2015, 6.3.2]

This process assessment model is based on the Automotive SPICE® Reference Model included in this document.

This process assessment model is based on the defined Measurement Framework.

| [ISO/IEC 33004:2015, 6.3.3]

The processes included in this process assessment model are identical to those specified in the Process Reference Model

| [ISO/IEC 33004:2015, 6.3.4]

For all processes in this process assessment model all levels defined in the process measurement framework are addressed.

| [ISO/IEC 33004:2015, 6.3.5]

This process assessment model defines

- the selected process quality characteristic;
- the selected process measurement framework;
- the selected process reference model(s);
- the selected processes from the process reference model(s)

in chapter 3 of this document.

| [ISO/IEC 33004:2015, 6.3.5 a-d]

In the capability dimension, this process assessment model addresses all of the process attributes and capability levels defined in the process measurement framework.

| [ISO/IEC 33004:2015, 6.3.5 e]

Clause 6.3.1, "Assessment indicators"

Note: Due to an error in numbering in the published version of ISO/IEC 33004:2015 the following reference numbers are redundant to those stated above. To refer to the correct clauses from ISO/IEC 33004:2015, the text of clause heading is additionally specified for the following three requirements.

The Automotive SPICE® process assessment model provides a two-dimensional view of process capability for the processes in the process reference model, through the inclusion of assessment indicators as defined in chapter 3.3. The assessment indicators used are:

- Base practices and information items

| [ISO/IEC 33004:2015, 6.3.1 a, "Assessment indicators"]

- Generic practices and information items

| [ISO/IEC 33004:2015, 6.3.1 b, "Assessment indicators"]

Clause 6.3.2, "Mapping process assessment models to process reference models"

The mapping of the assessment indicators to the purpose and process outcomes of the processes in the process reference model is included in the tables for each process in chapter 4.

The mapping of the assessment indicators to the process attributes in the process measurement framework including all of the process attribute achievements is included in the tables for each process attribute in chapter 5.

| [ISO/IEC 33004:2015, 6.3.2, "Mapping process assessment models"]

Clause 6.3.3, "Expression of assessment results"

The process attributes and the process attribute ratings in this process assessment model are identical to those defined in the measurement framework. As a consequence, results of assessments based upon this process assessment model are expressed directly as a set of process attribute ratings for each process within the scope of the assessment. No form of translation or conversion is required.

| [ISO/IEC 33004:2015, 6.3.3, "Expression of assessment results"]

Annex A.4 Conformance to the requirements for measurement frameworks

The measurement framework defined in Automotive SPICE® 4.0 is an adaption of the measurement framework defined in ISO/IEC 33020:2019. The following modifications have been performed:

- Renaming of the Process attribute titles.
- Changes in the generic practices.
- Assignments of indicators to process attribute achievements.

Conceptualization, Construct definition and Operationalization relevant for conformity to ISO/IEC 33003:2015 has been adopted from ISO/IEC 33020:2019.

The conformity of the Automotive SPICE® Measurement Framework is thereby confirmed based on the existing conformance statement of 33020:2019.

Annex B Information Item Characteristics

The characteristics of the information items are defined using the schema in table B.1. See section 3.3.2 on the definition and explanation on how to interpret information items and their characteristics.

Information item identifier	An identifier number for the information item which is used to reference the information item.
Information item name	Provides an example of a typical name associated with the information item characteristics. This name is provided as an identifier of the type of information item the practice or process might produce. These information items would be expected to be part of work products in a project or organization, see also Section 3.3.2. Therefore, the structure of the information items varies.
Information item characteristics	<p>Information items characteristic provide:</p> <ul style="list-style-type: none"> • Purpose and scope of an information item as description • Items and elements that it typically identifies • Characteristics that can be included • Examples of the characteristics, events when characteristics are observable • References to norms, standards and practitioners <p>The assessor may use these in evaluating the samples provided by the organizational unit. It is not intended to use the listed characteristics as a checklist. Some characteristics may be contained in other work products, as it would be found appropriate in the assessed organization.</p>

Table B.1 — Structure of information item characteristics (IIC) table

ID	Name	Characteristics
01-03	Software component	<p>See definition in section 0.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • The Software element in the software architecture above the software unit level, or an implementation of such a software element. <p>Includes:</p> <ul style="list-style-type: none"> • Design model element, or executable source code such as libraries, modules, services or scripts.
01-50	Integrated software	<p>A functional executable software which has been integrated from multiple software components.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • All configured software elements <p>Includes:</p> <ul style="list-style-type: none"> • Application parameter <p>Examples and references:</p> <ul style="list-style-type: none"> • Simulator with stubbing • Object code
01-52	Configuration item list	<p>List of configurable items under configuration control.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Name of configuration item • Work products • Unique reference to the configuration item <p>Includes:</p>

ID	Name	Characteristics
		<ul style="list-style-type: none"> • Configuration item attributes and properties
01-53	Trained ML model	Resulting output machine learning (ML) model of the ML training process Identifies: <ul style="list-style-type: none"> • Set of weights optimized during the training Includes: <ul style="list-style-type: none"> • Software representing the ML architecture • Set of hyperparameters
01-54	Hyperparameter	Settings or configurations that are used to control the learning process of a machine learning model. Identifies: <ul style="list-style-type: none"> • Description • Initial value • Value of the results of the ML training Includes: <ul style="list-style-type: none"> • Number of epochs • Scaling of network (number of layers or neurons per layer) Examples and references: <ul style="list-style-type: none"> • Learning rate of training, Loss function
02-01	Commitment / agreement	Mutual arrangement between two or more parties regarding terms of a transaction or relationship. Identifies: <ul style="list-style-type: none"> • All involved parties • Purpose of the collaboration • Definitions • Confidentiality • Termination conditions Includes: <ul style="list-style-type: none"> • Timings and durations • Resources, for example people and equipment • Efforts • Facilities • Responsibilities
03-03	Benchmarking data	Data resulting out of performance measurement to establish a comparison of performance values. Identifies: <ul style="list-style-type: none"> • Measurement of current performance • Historical and current values • Criteria and information to be benchmarked Includes: <ul style="list-style-type: none"> • References to metric • References to their purpose, for example: goals, process, product or involved market
03-04	Customer satisfaction data	Active and passive feedback information of a customer. Identifies: <ul style="list-style-type: none"> • Related products and services

ID	Name	Characteristics
		<ul style="list-style-type: none"> • Mechanism to collect data, for example field performance, survey, interview, meeting minutes, product demo <p>Includes:</p> <ul style="list-style-type: none"> • Notes • Observations • Trend data • Opportunities • Level of customer satisfaction
03-06	Process performance information	<p>Measurements about quantitative or qualitative measurable indicators, that match information for performance evaluation.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Measurement metrics for the calculation of the quantitatively or qualitatively measurable indicators • Efforts • Time • Data <p>Includes:</p> <ul style="list-style-type: none"> • Performance level definition, for example usage of defined capacity, agreements. • Targets definitions, for example throughput • Utilization data of resources, for example up time, service outage time • Time schedules • Task completion criteria, for example: job run time • Quality of process and work products, for example non conformances ratio <p>References and Examples:</p> <ul style="list-style-type: none"> • Process performance respective deliveries, service level, operational performance
03-50	Verification measure data	<p>Data recorded during the execution of a verification measure.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Logging data, for example raw data or tool generated output • Values of measurements and calculations • Protocols of simulations • Review data <p>Includes:</p> <ul style="list-style-type: none"> • Finding records
03-51	Machine Learning data set	<p>Selection of data for machine learning training and validation, or test of a machine learning model.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Patterns • Relationships • Features <p>Includes:</p> <ul style="list-style-type: none"> • Annotations and labels • ML Training and Validation Data Set • ML Test Data Set

ID	Name	Characteristics
03-53	ML data	<p>Data used for machine learning, attributed by metadata.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Unique ID • Attributes • Data characteristics, for example audio recordings, sensor data, pictures and videos • Data created by an algorithm <p>Includes:</p> <ul style="list-style-type: none"> • Data created by processing of other data, for example added noise, changed colors or merged pictures
03-54	Hardware production data	<p>Data used and created during hardware production.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Layout data, e.g., in GERBER format • Test coverage <p>Includes:</p> <ul style="list-style-type: none"> • Bill of material <p>Examples and References:</p> <ul style="list-style-type: none"> • In semiconductor development, layout data can be mask data, for example in GDS2 format.
04-02	Domain architecture	<p>High-level design and structure specific to a domain.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Domain model(s) tailored from • Asset specifications • Definition of boundaries and relationships with other domains, for example within a Domain Interface Specification • Domain vocabulary • Domain representation standard <p>Includes:</p> <ul style="list-style-type: none"> • Justifying rationale for the chosen architecture • Functions and features • Capability and concepts of the domain
04-04	Software architecture	<p>High-level design and structure of a software.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Overall software structure • Functional behavior of the software components • Nonfunctional characteristics • Settings for application parameters • Technical characteristics of interfaces, for example application programming interface (API) • Dynamic behavior within different states and operating modes, for example: start-up, shutdown, calibration • Timing definitions for example slices, cycle times • Priorities, for example of processes and threats • Events handling, for example interrupts with their priorities

ID	Name	Characteristics
		Includes: <ul style="list-style-type: none"> • Justifying rationale for the chosen architecture • Explanatory annotations, for example with natural language, for single elements, diagrams or models.
04-05	Software detailed design	Internal structure and behavior of software elements. Identifies: <ul style="list-style-type: none"> • Control flows • Format of input/output data • Algorithms • Data structures • Variables Includes: <ul style="list-style-type: none"> • Justifying rationale for the chosen design • Interfaces • Explanatory annotations • Elements of semi-formal language, for example UML, SysML Examples and references: <ul style="list-style-type: none"> • Expression in formal or semi-formal language may be used in model based developed design.
04-06	System architecture	High level design and structure of a system. Identifies: <ul style="list-style-type: none"> • Overview of the system structure • System elements • Behavior of system elements, for example operation modes, calibration sequences, performance. • Behavior with the environment • Interfaces between system elements and environment • Hardware-software interfaces, for example input-output pins of microcontrollers • Parameters, for example of resource and performance characteristics • Domain specific engineering references, for example mechanical engineering evaluations for thermal influence. Includes: <ul style="list-style-type: none"> • Justifying rationale for the chosen architecture • References to underlying architecture, for example Hardware architecture (04-52), Software architecture (04-06), or an ML-architecture (04-51) Examples and references: <ul style="list-style-type: none"> • Vehicle wiring overview with identification of wiring technologies, control units and wireless interfaces for connectivity.
04-51	ML architecture	High-level design and structure of a machine learning system. Identifies: <ul style="list-style-type: none"> • Overall structure of the ML-based software element

ID	Name	Characteristics
		<ul style="list-style-type: none"> • Elements including an ML model and other ML architectural elements • Interfaces within the ML-based software element • Interfaces to other software elements • Resource consumption objectives <p>Includes:</p> <ul style="list-style-type: none"> • Justifying rationale for the chosen architecture • Details of the ML model like used layers, activation functions, loss function, and backpropagation • Hyperparameter ranges and initial values for training • References to ML requirements
04-52	Hardware architecture	<p>High-level design and structure of physical components in an electronic device system.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Overall hardware structure • Hardware components • Own developed and supplied hardware components • Hardware component interfaces <p>Includes:</p> <ul style="list-style-type: none"> • Justifying rationale for the chosen architecture • Relationship and dependency between hardware components • Specifics for hardware variants • Power supply, thermal and grounding concepts <p>Examples and references:</p> <ul style="list-style-type: none"> • Integrated circuits (IC) components placement is documented on “floorplan” architecture document.
04-53	Hardware detailed design	<p>Internal structure and behavior of hardware subsystems.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Interconnections between the hardware parts • Interfaces of the hardware parts <p>Includes:</p> <ul style="list-style-type: none"> • Dynamic behavior in modes of operation and internal states, for example in power-up and power-down sequences. • Timing specifications, for example frequencies and delays • Technologies and their characteristics, for example modulations and filter settings. • Justifying rationale for the chosen design • References to datasheets, application notes • Constraints for layout
04-54	Hardware schematics	<p>Detailed diagrams of components and connections of a hardware system.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Hardware parts • Connections of the hardware parts <p>Includes:</p>

ID	Name	Characteristics
		<ul style="list-style-type: none"> • Identification information of hardware parts and variants.
04-55	Hardware layout	<p>Physical arrangement of components on a circuit board or within a device.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Placement of the hardware parts • Manufacturing data, for example circuit paths, testing points • material specific adaptations • Shapes, masks, labels • layout identification <p>Includes:</p> <ul style="list-style-type: none"> • Justifying rationale for the chosen design <p>Examples and references:</p> <ul style="list-style-type: none"> • Printed circuit boards (PCB) layouts include vias, different layers • Soldering mask
04-56	Hardware element interface	<p>Description of interactions between different physical components within a hardware system.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Electrical interconnections • Thermal interfaces • Electrical characteristics • Signal tolerance • Performance needs, for example bandwidth <p>Includes:</p> <ul style="list-style-type: none"> • Definition of output, input, type • heat dissipation • communication parameters • signal parameters <p>Examples and references:</p> <ul style="list-style-type: none"> • Standardized interfaces and protocols like SPI, I2C, CAN.
06-04	Training material	<p>Resources created or used to support learning.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Knowledge for operation, maintenance or development of systems. • Area of application <p>Includes:</p> <ul style="list-style-type: none"> • Course listings and availability • Procedures • Guidance • Training identification information
06-50	Integration sequence instruction	<p>Workflow definition to combine, assemble or connect different systems or components into an integrated configuration.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Physical elements, for example hardware, mechanical and wiring elements.

ID	Name	Characteristics
		<ul style="list-style-type: none"> • Software • Application parameters • Sequences and order of integration • Preconditions <p>Includes:</p> <ul style="list-style-type: none"> • Safety instructions • Recovery or fallback procedures • Verification activities
06-51	Tailoring guideline	<p>Set of rules to adapt processes, methods, documentation, or requirements to a specific project, product, or organizational context.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Criteria for tailoring • Proceeding of tailoring • Approval conditions • Integrity and consistency needs of the defined process <p>Includes:</p> <ul style="list-style-type: none"> • Tailoring matrix • Tailoring triggers • Assets of the defined process
06-52	Backup and recovery mechanism information	<p>Processes and tools used to create backups of data and restore it in case of loss or corruption.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Description and confirmation of backup and recovery mechanisms • References to corresponding procedures or regulations
07-04	Process metric	<ul style="list-style-type: none"> • Measurements about the process' performance: <ul style="list-style-type: none"> - ability to produce sufficient work products - adherence to the process - time it takes to perform process - defects related to the process • Measures the impact of process change • Measures the efficiency of the process
07-51	Measurement result	<p>Outcome of gathering qualitative or quantitative data.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Data sources • Data ownership • Evaluation and measurement workflow • Conformity requirements, for example on data protection • Approval and acceptance criteria <p>Includes:</p> <ul style="list-style-type: none"> • Benchmarking needs • Copyright • Metric references, for example to process, project or quality metric

ID	Name	Characteristics
		<p>Examples and References:</p> <ul style="list-style-type: none"> • Quality metrics could be identified in survey of the end customer, for example for usability, reliability and efficiency. • Field failure measurement for performance of 5D cycle time. • Project metrics for project management, for example resource utilization against plan.
07-61	Quantitative process metric	<p>Quantitative evaluation of measurable relationships and characteristics of a process.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Measurable indicators • Relationship of the quantitatively measurable indicators • Trend evaluation <p>Includes:</p> <ul style="list-style-type: none"> • Process elements and descriptions • Repositories and tools, for example to store historical data <p>Examples and References:</p> <ul style="list-style-type: none"> • Information needs derived from business goals.
07-62	Process analysis technique	<p>Workflow description to control process within specific limits.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Calculation specification, for example statistic algorithm • Mathematical references • Data structure <p>Includes:</p> <ul style="list-style-type: none"> • Frequency of data collection
07-63	Process control limits	<p>Boundaries used to monitor and manage a process.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Quantitative control limits • Relevant process metrics
07-64	Process measurement data	<p>Data collected across process instances.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Attributes of data, for example timestamps • Relation to process measurement metrics • Storage and retrieval • Controls over access
08-53	Scope of work	<p>Definition of tasks, responsibilities and deliverables of a project or program.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Deliverables and their intended use • Functional descriptions • Deliveries and delivery dates • Milestones • Work products • Activities

ID	Name	Characteristics
		<ul style="list-style-type: none"> • Out of scope definitions • Acceptance requirements <p>Includes</p> <ul style="list-style-type: none"> • Customer and market information • Applicable standards and legal requirements • Reuse options • Integration of third-party deliveries
08-54	Feasibility analysis	<p>An evaluation of the ability to achieve the project objectives within available time and resources.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Available resources • Efforts, for example variable and fixed duration activities <p>Includes:</p> <ul style="list-style-type: none"> • Dependencies to supporting and shared resources • Technical evaluation • Recovery options • Critical path analysis
08-55	Risk measure	<p>Quantitative or qualitative value to express the level of a risk.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Risk to be mitigated, avoided, or shared (transferred) • Related activities • Criteria for success or failure of activities • Frequency of monitoring <p>Includes:</p> <ul style="list-style-type: none"> • Exposure and detectability information • Originator • Alternative treatment options • Risk justifications, for example as claims
08-56	Schedule	<p>A plan that defines the times and sequences for events, tasks, or activities.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Activities • Timing requirements, for example start and due dates • Dependencies between activities • Critical path <p>Includes:</p> <ul style="list-style-type: none"> • Progress evaluation, for example completion of individual activities • Scheduled resources • Input data • Evaluation of resource workload • Shared resources
08-57	Validation measure selection set	<p>A set of validation measures selected for a specific purpose.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Validation measures

ID	Name	Characteristics
		<ul style="list-style-type: none"> • Scope of their application, for example a variant, regression or a subcomponent <p>Includes:</p> <ul style="list-style-type: none"> • Identification of the validation measures • Criteria for re-validation, for example in the case of changes • Dependencies
08-58	Verification measure selection set	<p>A set of verification measures selected for a specific purpose.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Verification measures • Scope of their application, for example a product variant, regression or a subcomponent <p>Includes:</p> <ul style="list-style-type: none"> • Identification of the verification measures • Criteria for re-verification, for example in the case of changes • Dependencies
08-59	Validation measure	<p>Tool or method used to ensure the accuracy and reliability of a measurement or test under realistic conditions to validate the intended use.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Description of test scope • Measurement to be performed during the test • Test steps description • Pass/fail criteria, for example from end user survey <p>Includes:</p> <ul style="list-style-type: none"> • Simulation scenarios • Emulation • Entry criteria • Techniques and resources • Validation environment setup and configuration • Sequences
08-60	Verification measure	<p>Tool or method used to ensure the accuracy and reliability of a measurement or test under specified conditions to verify specified functionality.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Description of test scope, for example fault injection safety case • Classifications, for example black-box or grey-box test • Measurement to be performed during the test, for example an optical inspection • Test steps description • Pass/fail criteria, for example from end user survey <p>Includes:</p> <ul style="list-style-type: none"> • Simulation scenarios • Emulation • Entry criteria • Equivalence classes and boundary values

ID	Name	Characteristics
		<ul style="list-style-type: none"> • Calculations and mathematical functions • Techniques and resources, for example review verification • Verification environment setup and configuration • Sequences
08-61	Resource allocation	<p>Distribution of resources to tasks and locations.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Resources identification information • Tasks • Locations • Timing needs • Roles • Restrictions of availability <p>Includes:</p> <ul style="list-style-type: none"> • Workload evaluation • Physical and material resources <p>Examples and references:</p> <ul style="list-style-type: none"> • Resource allocation for a test team, for example persons, tool licenses, test hardware
08-62	Communication matrix	<p>Relationships and dependencies of an organization to manage communication.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Affected internal and external stakeholders • Roles • Contact information • Definition of affected interfaces between stakeholders • Communication subject <p>Includes:</p> <ul style="list-style-type: none"> • Communication means and frequency • Documentation needs of the communication, for example type of record.
08-63	Process monitoring method	<p>Documentation of workflow to measure and to control boundaries of a process.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Methods of analysis • Measurements • Criteria for monitoring • Boundaries and limits of measures <p>Includes:</p> <ul style="list-style-type: none"> • Measurement frequency and technology
08-64	ML test approach	<p>Approach describing criteria and activities to test ML models.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • ML test scenarios • Distribution of data characteristics, for example gender of persons or weather conditions • Related ML requirements • Pass/fail criteria • Entry and exit criteria

ID	Name	Characteristics
		<ul style="list-style-type: none"> • Environment setup and configuration Includes: <ul style="list-style-type: none"> • References, for example with test data set
08-65	ML training and validation approach	Approach describing method for evaluating a trained ML model using data to confirm its accuracy and for the intended task. Identifies: <ul style="list-style-type: none"> • Entry and exit criteria • Approaches for hyperparameter tuning / optimization • Approach for data set creation and modification Includes: <ul style="list-style-type: none"> • Training environment setup and configuration • Resources • Provision of input data • Storage of output data • Methods to control robustness, for example random dropout
08-66	Measures against deviations in quantitative process analysis	Definition of corrective measures for quantitative process control. Identifies: <ul style="list-style-type: none"> • Corrective measures • Implementation boundaries • Causes of variation • Timing requirements • Evaluation requirements, for example data filtering.
10-00	Process description	Description of a standard or a defined process. Identifies: <ul style="list-style-type: none"> • Scope, purpose and intended use • Activities including their description and dependencies • Entry and exit criteria such as input information needed and expected outputs for activities • Templates • Guidance documents, for example guidelines, work instructions, standards or method descriptions Includes: <ul style="list-style-type: none"> • Roles assigned to process activities, for example as RASIC or to work products • Sequence and interaction of activities within the process • Sequence and interaction of the process to other processes • Tailoring guidelines
10-50	Role description	Definition of a process role. Identifies: <ul style="list-style-type: none"> • Name/identifier • Assigned activities, for example as RASIC • Responsibilities • Authorities

ID	Name	Characteristics
		<ul style="list-style-type: none"> • Required competencies, skills, and experience Includes: <ul style="list-style-type: none"> • Reference to qualification methods
10-51	Qualification method description	Description of a qualification method for human resources. Identifies: <ul style="list-style-type: none"> • Competence category, for example product, process or tools • Management level and title • Subject specific needs • Name • Required level of knowledge, for example to standards and norms • Type of method, for example as formal training, on the job, mentoring or coaching Includes: <ul style="list-style-type: none"> • Reference to courses and training material
10-52	Process resource and infrastructure description	Description of a non-human resource. Identifies: <ul style="list-style-type: none"> • Facilities • Tools and licenses • Networks • Services, for example by internal or external providers • Samples Includes: <ul style="list-style-type: none"> • Evidence for tool qualification if relevant, for example within safety critical usage.
11-03	Release note	Documentation of content provided in the release. Identifies: <ul style="list-style-type: none"> • Set of functionalities provided in the release • Limitations in relation to the committed scope • Known non-conformities • Application parameters • Configurations and variants • Unique name and version of the release. • Additional or removed functionalities • Changed or improved functionalities • Documentation of resolved non-conformities Includes: <ul style="list-style-type: none"> • Intended area and environment of usage • Impact to other components linked to the release • Upstream or downstream compatibility of delivered release • Copyright and license information • Result of verification and validation measures • Approval for delivery of the release by responsible roles • Dependencies to other linked products, for example hardware revision.

ID	Name	Characteristics
11-04	Product release package	<p>Set of Elements delivered in the product release. Identifies:</p> <ul style="list-style-type: none"> • Physical components • Elements necessary for intended use • Warnings and alerts • Application Parameters <p>Includes:</p> <ul style="list-style-type: none"> • Documentation to maintain or service the product • Information for the user, for example installation and setup instruction within a manual.
11-05	Software unit	<p>Software element not decomposed to a lower level. Identifies:</p> <ul style="list-style-type: none"> • Single functions • Set of functions <p>Includes:</p> <ul style="list-style-type: none"> • Intended use and defined functionality, for example within a specification or an agreement <p>Examples and References:</p> <ul style="list-style-type: none"> • Representation of a software element at the lowest level in a conceptual model.
11-06	Integrated system	<p>Set of elements composed from different units or integrated sub-systems. Identifies:</p> <ul style="list-style-type: none"> • The intended use • Boundaries, for example application parameters <p>Includes:</p> <ul style="list-style-type: none"> • All configured elements, for example references to requirements specification
11-50	Deployed ML model	<p>Deployed ML model Identifies:</p> <ul style="list-style-type: none"> • The source code derived from the trained ML model that shall be executed in the target system. <p>Examples and references:</p> <ul style="list-style-type: none"> • It may differ from the trained ML model which often requires powerful hardware and uses interpretative languages. • The deployed ML model is usually written in programming languages like C/C++.
12-03	Reuse candidate	<p>Description of a product that might be reused. Identifies:</p> <ul style="list-style-type: none"> • Name of the product • Person responsible for the product • Reuse goals and objectives • List of reuse assets • Issues/risks of reusing the product • Person responsible for the qualification of the product to be reused • Type of the product's provision, for example storage location or delivery details

ID	Name	Characteristics
		Includes: <ul style="list-style-type: none"> • Specific requirements
13-06	Delivery evidence	Evidence of items shipped or electronically delivered to customer. Identifies: <ul style="list-style-type: none"> • Receiver • Delivery address • Delivery date • Acknowledgment of receipt Includes: <ul style="list-style-type: none"> • Identification information
13-07	Problem	Description of an issue. Identifies: <ul style="list-style-type: none"> • Submitter • Version it was identified • Classification, for example criticality, urgency or relevance • Project phase in which problem is recorded • Owner • Status • Technical root cause • Potential effects on other systems • Expected closure date Includes: <ul style="list-style-type: none"> • Identification • References to other affected domains, subprojects, or programs • Relationships to dependencies like changes, preventive measures or other problems
13-08	Baseline	A coherent set of work products which are consistent and complete and may serve as basis for next process steps and/or delivery. A baseline must be unique and not changed any more. Identifies: <ul style="list-style-type: none"> • All work products belonging to the baseline • References using the configuration item properties Includes: <ul style="list-style-type: none"> • Name • Version • State
13-09	Meeting support evidence	Agenda and minutes of a meeting. Identifies: <ul style="list-style-type: none"> • Name • Purpose • Date and place held • Attendees • Outcomes, decisions Includes:

ID	Name	Characteristics
		<ul style="list-style-type: none"> • Reference to previous minutes • Open issues • Action points including responsibilities and due dates • Next meeting
13-13	Product release approval	<p>Evidence of approval for a product release.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Release date • Name and role of approver • Intended customer • Restrictions and limitations for usage <p>Includes:</p> <ul style="list-style-type: none"> • Delivery / shipping address • Release information of what is to be shipped or delivered • If applicable, supplier approval evidence(s)
13-14	Progress status	<p>Set of indicators to document the achievements created by performing process activities according to the committed work to the former phase.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Status of work activities, for example in respect to a plan • Evidence of the committed outcome • Fulfilment of criteria within the state model • Limitations of outcome or achievements in respect to the committed objectives, for example deviations of performed verification <p>Includes:</p> <ul style="list-style-type: none"> • Timing references, for example in relation to schedule, milestones • Efforts <p>Examples and references:</p> <ul style="list-style-type: none"> • Status of actual quality against planned quality
13-16	Change request	<p>Request to change scope or impacts on planned activities or outcomes.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Scope and purpose of the change activities • Originator of the change • Impacted operation environment • Process performance • Environment of usage for the process outcome • Infrastructure or resources • Documentation needs <p>Includes:</p> <ul style="list-style-type: none"> • Impacted outcomes and achievements • Process objectives • Approval for implementation of a change • Limitations to be considered • Criteria to verify the fulfillment of the implementation

ID	Name	Characteristics
13-18	Quality conformance evidence	<p>Documents the fulfillment of quality related criteria. Identifies:</p> <ul style="list-style-type: none"> • Process performance • Work product compliance • Verification criteria • Process outcomes • Achievements • Success factors • Performance objectives <p>Includes:</p> <ul style="list-style-type: none"> • References to quality activities • Data collected • Measurement and calculation
13-19	Review evidence	<p>A proof to demonstrate a structured check of a work product or a process has taken place. Identifies:</p> <ul style="list-style-type: none"> • Name and version of the review object • Names and roles of the people who performed the review • Date • Status • Review criteria, for example review method, coverage, justification • Non-conformances found • Improvement suggestions <p>Includes:</p> <ul style="list-style-type: none"> • Documents considered, for example checklists, guidelines, work instructions • Time spent • Applicable standards
13-24	Validation results	<p>Documented evidence confirming whether the end product meets the intended use expectations. Identifies:</p> <ul style="list-style-type: none"> • Data, logs, feedback, or documentation • Measures passed • Measures not passed • Measures not executed • Rationale, for example security claims • Information about the execution (date, participants, etc.) • Abstraction or summary of results
13-50	ML test results	<p>Results of ML test activities Identifies:</p> <ul style="list-style-type: none"> • Test data and logs • Test data with correct results • Test data with incorrect results • Test data not executed, and a rationale <p>Includes:</p> <ul style="list-style-type: none"> • Information about the test execution (date, participants, model version etc.)

ID	Name	Characteristics
13-51	Consistency evidence	<ul style="list-style-type: none"> • Abstraction or summary of ML test results <p>Evidence of information to be semantically coherent along relevant artifacts, ensuring completeness, purpose and maturity of processes, tasks and products throughout their lifecycle.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Traceability information, for example hyperlinks, repository location or editorial references. • Naming conventions • Relevant artifacts • Revision and revision history information • Change documentation and analysis information <p>Includes:</p> <ul style="list-style-type: none"> • Meta-information, for example database identifiers, notes in Git commits comments <p>Examples and References:</p> <ul style="list-style-type: none"> • Evidence of Definition of Done (DoD) adherence.
13-52	Communication evidence	<p>Evidence of interpersonal communication.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Scope of information • Need for feedback, for example an expected confirmation within one week • Meta data, for example time when communication was done or how information was distributed. <p>Includes:</p> <ul style="list-style-type: none"> • Personal information • Work-flows, for example within tools <p>Examples and References:</p> <ul style="list-style-type: none"> • E-mails and other forms of memos • Verbal statements • Meeting minutes, for example in standups • Electronic media, for example webcasts, blog posts, intranet forum • Chat protocols • Wiki pages • Photo protocol
13-53	Qualification evidence	<p>Evidence that the products for reuse are qualified for the intended use of the deliverable.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Qualification checklist • Passed items • Failed items • Pending items • Problems identified during qualification • Recommendation of actions • Conclusions of qualification <p>Includes:</p> <ul style="list-style-type: none"> • Risk analysis • Approval • Applicable standards

ID	Name	Characteristics
13-55	Process resource and infrastructure documentation	<p>Documentation that the required resources for performing the defined process are made available, allocated, used and maintained.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Information on availability, allocation and usage of resources • Facilities • Tools and corresponding licenses • Networks • Services • Needs for samples, for example prototype protection <p>Includes:</p> <ul style="list-style-type: none"> • Maintenance information, if applicable
14-01	Change history	<p>Structured, chronological record of modifications made to an object (document, file, software component, etc.) over its lifecycle.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Version information about changed object • Date of change • Description of change • Originator
14-02	Corrective action	<p>Activity required to resolve a problem.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Initial problem description • Ownership of activity • Definition of solution(s) • Series of actions <p>Includes:</p> <ul style="list-style-type: none"> • Timing needs, for example required closure or analysis date • Status indicator • Further activities needed, for example a follow up audit.
14-10	Work package	<p>Activities required to be performed in order to complete a set of work.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Activities to be performed • Ownership, for example to a specific domain like verification. • Dependencies to other activities and work products • Input and output work products • Required information <p>Includes:</p> <ul style="list-style-type: none"> • Estimation of efforts and duration <p>Examples and References:</p> <ul style="list-style-type: none"> • Work package can be a statement of work document, for example to outsource activities to an engineering service.

ID	Name	Characteristics
14-50	Stakeholder groups	<p>A group of people, relevant to a specific purpose or domain.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Persons and group of persons • Responsibilities of ownership • Representatives and roles <p>Includes:</p> <ul style="list-style-type: none"> • Information needs, for example reporting content and frequency. <p>Examples and references:</p> <ul style="list-style-type: none"> • Expert groups for specific domains, for example safety. • Executive board
14-53	Role assignment	<p>Assignment of person(s) to roles.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Required competencies • Existing competencies • Experiences • Training related to role <p>Includes:</p> <ul style="list-style-type: none"> • Gap identification and guidance • Approver
14-54	Hardware bill of materials	<p>Description of subcomponents, material and assemblies included within a hardware product.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Type of components • Number of hardware parts • Supplier name • Revision information <p>Includes:</p> <ul style="list-style-type: none"> • Structure and hierarchical information
15-06	Project status	<p>Evaluated result for the relative state of a project</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Progress and consistency to plans and schedule • Completed tasks • Relation to the intent of the project • Resources usage, for example hardware, material, human resources <p>Includes:</p> <ul style="list-style-type: none"> • Deviations and justification • Risks and opportunities to the planned progress and agreements • Issues and actions
15-07	Reuse analysis evidence	<p>Evaluation of suitability for an existing component, requirement, design element, or process to be used in a new project.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Constraints for reuse, for example known defects and vulnerabilities

ID	Name	Characteristics
		<ul style="list-style-type: none"> • Related change of infrastructure and environment • Functional fit • Compatibility • Dependencies • Maturity <p>Includes:</p> <ul style="list-style-type: none"> • Regulatory and standard differences • Characteristics of domain specific architecture <p>Examples and References:</p> <ul style="list-style-type: none"> • Reuse of an off-the-shelf component within a different product line.
15-09	Risk status	<p>Status or change of an identified risk.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Risk statement • Sources of the risk • Impact, for example in relation to severity and probability • Owner <p>Includes:</p> <ul style="list-style-type: none"> • Category • Threshold value(s) • Risk treatment activities and their progress
15-12	Problem status	<p>Progress of problem resolution.</p> <p>Identifies</p> <ul style="list-style-type: none"> • Containment and prevention activities • Dependencies status • References to status model <p>Includes:</p> <ul style="list-style-type: none"> • Problem categories and classification
15-13	Assessment/audit result	<p>Documentation of results of an audit or assessment activity.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Gaps, potentials, weaknesses or non-conformances that require corrective actions • Scope and purpose • Context to the evaluations • Results, for example rating by percentage or traffic lights • Dates the activity was performed • Requirements used <p>Includes:</p> <ul style="list-style-type: none"> • Methods used • References to evidence • Assumptions and limitations • Organizational unit • Sponsor information • Assessment team • Attendees

ID	Name	Characteristics
15-16	Improvement opportunity	<p>Suggestion or idea of an improvement related to a weakness or a new capability.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Related problem and cause • Value expected to benefit of, for example business opportunities or efficiency increases • Penalties of losses for not making the improvement <p>Includes:</p> <ul style="list-style-type: none"> • Owner • Decider and approve • Evaluation of the improvement
15-51	Analysis results	<p>Results of analysis of an object or task.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Object under analysis • Analysis criteria • Selection criteria or prioritization scheme • Decision criteria • Quality criteria <p>Includes:</p> <ul style="list-style-type: none"> • Decisions and selections performed • Assumptions and constraints • Evaluation criteria, for example correctness, completeness or consistency to a work product <p>Examples and references:</p> <ul style="list-style-type: none"> • Verifiability analysis results, for example when a test machine becomes defective • Results of a feasibility analysis
15-52	Verification results	<p>Results of verification activities.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Identification information • Log data • Passed and failed results • Not executed/blocked verification activities • Test execution details, for example tester name, role. • Timing information <p>Includes:</p> <ul style="list-style-type: none"> • Summary of verification results • References, for example to problem resolution actions.
15-54	Tailoring documentation	<p>Context specific use evaluation results and adaptations to a standard process.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Applied criteria for tailoring • Evidence of adaptations made to the standard process, for example meeting organization changes. • Reference to standard process and version, for example adaptation of lifecycle <p>Includes:</p> <ul style="list-style-type: none"> • Traceability information • Methods, tools and techniques

ID	Name	Characteristics
		<ul style="list-style-type: none"> Formalities for regulation compliance and conformities. <p>Examples and References:</p> <ul style="list-style-type: none"> Project management tailoring for development in an agile approach.
15-55	Problem analysis evidence	<p>Elements and documents proving a problem analysis is performed.</p> <p>Identifies:</p> <ul style="list-style-type: none"> Analyst and owner Affected and involved parties Context and root cause of the problem Analysis results and summary Potential impact, for example as severity or cost calculation <p>Includes:</p> <ul style="list-style-type: none"> Date of the analysis Potential solution(s)
15-56	Configuration status	<p>Summary of configuration management records and status.</p> <p>Identifies:</p> <ul style="list-style-type: none"> Version and change history information Baseline identification Status of configuration items (CI) integrity and consistency Identified deviations, for example missing CI, performance metric deviations
15-57	Quantitative process analysis results	<p>Quantitative analysis results of a process.</p> <p>Identifies:</p> <ul style="list-style-type: none"> Results to defined, measurable outputs, for example cycle time, defect density, process yield Distribution results <p>Includes:</p> <ul style="list-style-type: none"> Trend analysis Statistical control evaluation Quantitative control limits <p>Examples and references:</p> <ul style="list-style-type: none"> Lean Six Sigma analysis with evaluation of process capability indices (Cpk)
15-58	Common cause of variation analysis results	<p>Results of variation analysis specific for common cause.</p> <p>Identifies:</p> <ul style="list-style-type: none"> References of common causes Deviations of the quantitative performance of process instances Distributions of the quantitative performance of process instances <p>Includes:</p> <ul style="list-style-type: none"> Quantitative control limits
16-03	Configuration management system	<p>System to manage configurations throughout a project or product life cycle, providing integrity, consistency and control.</p>

ID	Name	Characteristics
		<p>Identifies:</p> <ul style="list-style-type: none"> • Configuration items to be controlled, for example documents, hardware or tools • Configuration information, for example to a status model • Repository • Baselines • Changes and change status <p>Includes:</p> <ul style="list-style-type: none"> • Reporting requirements • Workflow of recovery and other containment activities, for example security fire drills. <p>Examples and References:</p> <ul style="list-style-type: none"> • NASA defines it as technical management discipline that provides visibility and control over performance, functional, and physical characteristics of a product, preventing incorrect or unsafe configurations from being released.
16-06	Process repository	<p>Controlled storage location for configuration related artefacts of a process.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Artefacts, for example process descriptions • Assets of the process • Access control • Changes to artefacts • Version information • Meta data, for example name, attributes, reference links. <p>Includes:</p> <ul style="list-style-type: none"> • Storage and retrieval characteristics, for example in archiving processes • Status information
16-50	Organizational structure	<p>Hierarchical overview and structure of a set of groups and persons interacting with each other for a specific activity.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Hierarchical and functional reporting lines • Ownership and responsibilities • Organizational units <p>Includes:</p> <ul style="list-style-type: none"> • Interfaces • Locations • Communication details
16-52	ML data management system	<p>Data management system supporting machine learning engineering (MLE).</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Data management activities • Data collection workflow • Labeling, Annotations and description • Ingestion

ID	Name	Characteristics
		<ul style="list-style-type: none"> • Exploration and profiling workflows • Procedure of structuring and cleansing • Related phases of engineering, for example ML training <p>Includes:</p> <ul style="list-style-type: none"> • References to configuration management • Information of data sources <p>Examples and References:</p> <ul style="list-style-type: none"> • Supports the relevant sources of ML data
17-00	Requirement	<p>Statement that identifies an operational, functional, or design characteristic, which is unambiguous, testable, measurable, and necessary for product or process acceptability.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Expectation of functions and capabilities (e.g., non-functional requirements) • Owner and origin of the requirement • Identification information • Boundaries, for example target price range <p>Includes:</p> <ul style="list-style-type: none"> • Stakeholder viewpoint, for example black-box perspective • Design constraints, for example predefined and necessary implementation details. • Tolerances • Domain specific details, for example of safety and privacy standards <p>Examples and references:</p> <ul style="list-style-type: none"> • ISO/IEC directive quote: <i>“expression, in the content of a document, that conveys objectively verifiable criteria to be fulfilled and from which no deviation is permitted if conformance with the document is to be claimed”</i> • ISO/IEC 42010:2007 • Hardware level requirements for example mission profile, storage requirements of environment boundary definitions during environment test. • Operational expectations for example power consumption, crank behavior, heat dissipation boundaries. • Resource constraints, for example performance and memory space definition.
17-05	Requirements for work products	<p>Statement that identifies a textual, functional, or design characteristic, necessary for work products.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Expectation to content and structure • Owner and origin of the requirement • Control and approval mechanism • Identification information • Access rights • Status model references, for example as maintenance and disposal requirements <p>Includes:</p>

ID	Name	Characteristics
		<ul style="list-style-type: none"> • References to documentation templates • Table of content or other overview documentation • Design constraints, for example predefined and necessary implementation details. • Tolerances • Domain specific details, for example of safety and privacy standards
17-54	Requirement attribute	<p>Information supporting structuring of requirements for specific context or intent.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Context and intent of requirement <p>Includes:</p> <ul style="list-style-type: none"> • Meta-attributes • Solution intent
17-55	Resource needs	<p>Identification of required resources for process performance.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Human resources • Competencies • Skills and authorities • Material • Infrastructure and equipment • Tooling • Time • Budgets <p>Includes:</p> <ul style="list-style-type: none"> • Relation to work breakdown structure and planning information
17-57	Special characteristics	<p>Product or process characteristics that impact safety, legal compliance, fit, function, performance, or further processing, and therefore require special treatment.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Product characteristics • Production process parameters • Domain impact references, for example security. • References to regulations (compliance) • Conformity requirements <p>Includes:</p> <ul style="list-style-type: none"> • Processing and performance requirements <p>Examples and References:</p> <ul style="list-style-type: none"> • IATF 16949, VDA 6.x Guidelines, ISO 26262, ISO/SAE 21434 • FMEA rating within FMEA can consider special characteristics.
18-06	Product release criteria	<p>Conditions to be satisfied prior to authorization of a product for delivery.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Release type and status • Elements of the release • Completeness conditions

ID	Name	Characteristics
		<ul style="list-style-type: none"> • Documentation needs <p>Includes:</p> <ul style="list-style-type: none"> • Adequacy considerations • Coverage • Limits, for example the maximum of residing risk in a product. <p>Examples and References:</p> <ul style="list-style-type: none"> • Acceptance conditions for an engineering service product to a specific milestone.
18-07	Quality criteria	<p>Specified requirements to determine whether a product, process, or service conforms to quality objectives.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Measurement needs • Timings • Required elements • Needs for completeness • Accuracy • Performance needs <p>Includes:</p> <ul style="list-style-type: none"> • Thresholds • Tolerance level • Attributes • Compliance and conformance references <p>Examples and References:</p> <ul style="list-style-type: none"> • Conformance requirements specifying a specific set of values to be within a defined range.
18-52	Escalation path	<p>A sequence of organizational levels or responsible roles to which an unresolved issue, deviation, or risk shall be communicated in order to decision-making.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Trigger conditions • Stakeholders and roles • Escalation methods • Decision authority <p>Includes:</p> <ul style="list-style-type: none"> • Level specific conditions for an escalation. • Communication and documentation requirements <p>Examples and References:</p> <ul style="list-style-type: none"> • A task force group requested by a customer includes an escalation path
18-53	Configuration item selection criteria	<p>Rules or parameters to determine elements of a system to be included under configuration control.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Stakeholder needs • Regulatory needs • Configuration item identification requirements <p>Includes:</p> <ul style="list-style-type: none"> • Type of work products • Classification characteristic

ID	Name	Characteristics
		<p>Examples and References:</p> <ul style="list-style-type: none"> • Maintainability requirement of software evaluation work products. • Traceability requirements of supplier goods.
18-57	Change analysis criteria	<p>Definition of criteria to be included in the analysis of a change</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Resource requirements • Timing needs, for example to a project schedule • Risks and opportunities
18-58	Process performance objectives	<p>Description of performance goals and targets to be achieved by a process.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Evaluation criteria • Metrics, for example number of change requests • Timing definitions, for example duration until change requests are to be analyzed • Events, for example demonstration milestone <p>Includes:</p> <ul style="list-style-type: none"> • Assumptions and constraints <p>Examples and References:</p> <ul style="list-style-type: none"> • Effort or cost targets • Budget limitations • Frequency of tasks • Limits of parallel work items for an employee
18-59	Review and approval criteria for work products	<p>Specification of review and approval criteria for work products.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Review conditions • Timing needs, for example a specific milestone • Review methods, for example inspection or peer review <p>Includes:</p> <ul style="list-style-type: none"> • Approval definition <p>Examples and References:</p> <ul style="list-style-type: none"> • Definition when and how to perform a walkthrough meeting.
18-70	Business goals	<p>Explanation of business goals.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Requirements for the business needs • Associations to other goals • Reasons and rationales for the goals • Authorization level • Timeframe <p>Includes:</p> <ul style="list-style-type: none"> • Degree of the need • Effect on the business, for example in case of delay. • Assumptions and constraints <p>Examples and References:</p>

ID	Name	Characteristics
		<ul style="list-style-type: none"> • Enabler program of a new technical capability in production. • Demonstrations of a prototype to customer may be used to evaluate and gain feedback to business goals.
18-80	Process improvement opportunity	<p>Cause of an improvement need.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Improvement objectives, for example a specific business goal • Organizational scope • Process scope • Qualitative evaluation • Quantitative evaluation <p>Includes:</p> <ul style="list-style-type: none"> • Activities • Stakeholders • Priorities <p>Examples and References:</p> <ul style="list-style-type: none"> • Best practice analysis • State-of-the-art observations • Market studies
18-81	Improvement evaluation results	<p>Evaluation result of the effects of realized changes initiated for improvement.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Expected benefits • Conditions, for example evaluation cycle time • Constraints and assumptions <p>Includes:</p> <ul style="list-style-type: none"> • Tactical level impact • Strategical level impact • Operational level impact <p>Examples and References:</p> <ul style="list-style-type: none"> • Cycle time of a changed problem resolution process for optical hardware inspection. • ISO/IEC TR33014 Guide for process improvement
19-01	Process performance strategy	<p>Description of an approach to achieve performance objectives of a process.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Proceedings • Methodology • Scope of the strategy <p>Includes:</p> <ul style="list-style-type: none"> • Monitoring requirements, for example of performance • Domain specific references • Socio-cultural differences <p>Examples and References:</p> <ul style="list-style-type: none"> • Performance specific to development sites • Configuration management approaches for software

ID	Name	Characteristics
19-50	ML data quality approach	<p>Approach describing criteria and activities to satisfy ML data quality goals.</p> <p>Identifies:</p> <ul style="list-style-type: none"> • Quality criteria for example relevant data sources. • Analysis activities of the data <p>Includes:</p> <ul style="list-style-type: none"> • Reliability and consistency of labeling • Completeness against ML data requirements <p>Examples and References:</p> <ul style="list-style-type: none"> • Quality criteria to avoid data bias

Table B.2 — Information Item Characteristics

Annex C Key concepts

The following sections describe the key concepts that have been introduced in the Automotive SPICE® PRM resp. PAM 3.1. They relate to the terminology described in section 0.

Annex C.1 The "Plug-in" concept

The following figure shows the basic principle of the "plug-in" concept. The top-level comprises all system engineering processes organized in a system "V". Depending on the product to be developed the corresponding engineering disciplines with their domain-specific processes (e.g., hardware engineering HWE or software engineering SWE) can be added to the assessment scope. All other processes such as management processes and supporting processes are domain-independent and are therefore designed in a way that they can be applied to both the system level and the domain levels.

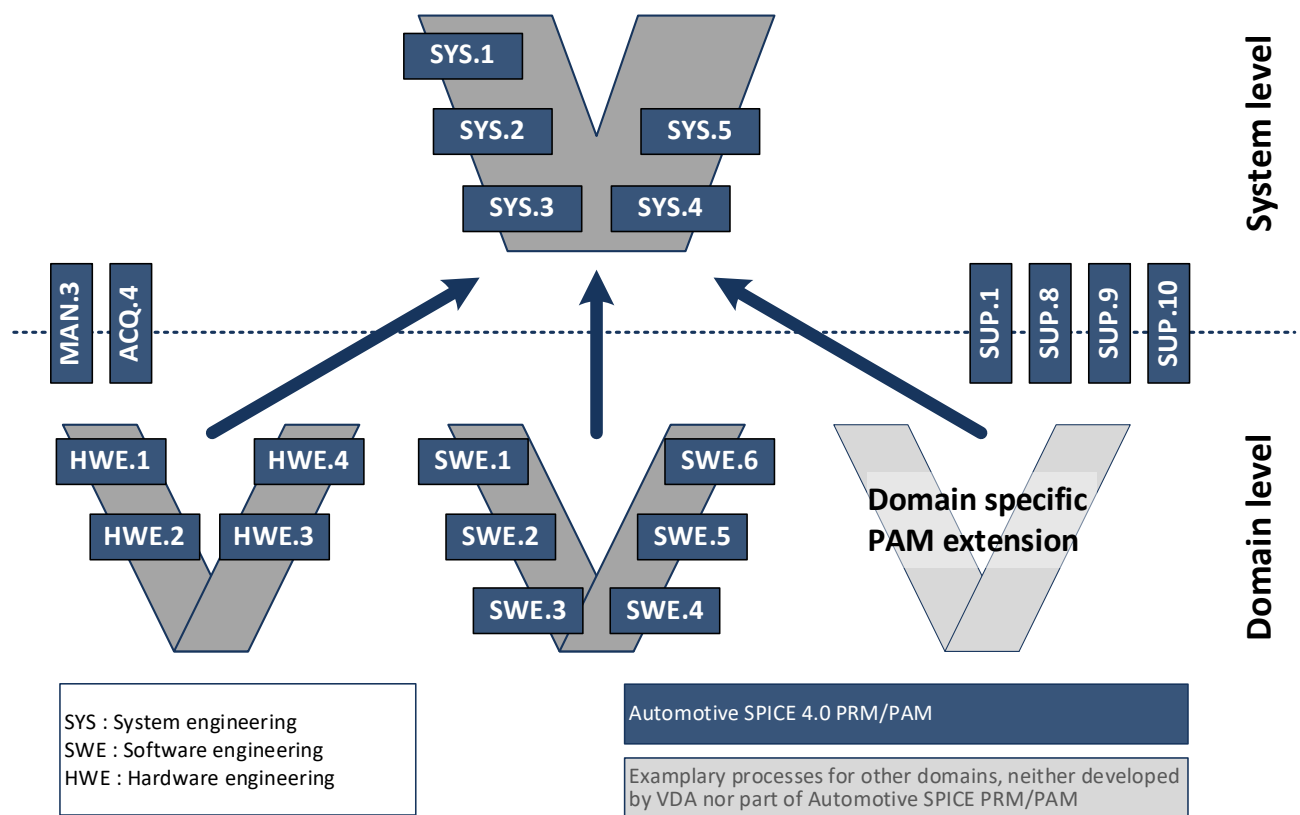


Figure 5 — Annex C.1: The "Plug-in" concept

Annex C.2 Integration of Machine Learning Engineering Processes

The following figure shows how the Machine Learning Engineering Processes are integrated within the engineering V-Cycle perspective. Usually, the MLE “Machine Learning Engineering” processes are used in a highly iterative way.

Within the Software Architecture software elements shall be identified which need to be developed with machine learning. For these ML-based software elements the MLE processes apply, for other software components SWE.3 “Software Detailed Design & Unit Construction” and SWE.4 “Software Unit Verification” applies. After the successful testing of the ML-based software elements they need to be integrated with the other software components by applying SWE.5 “Software Integration & Integration Test”.

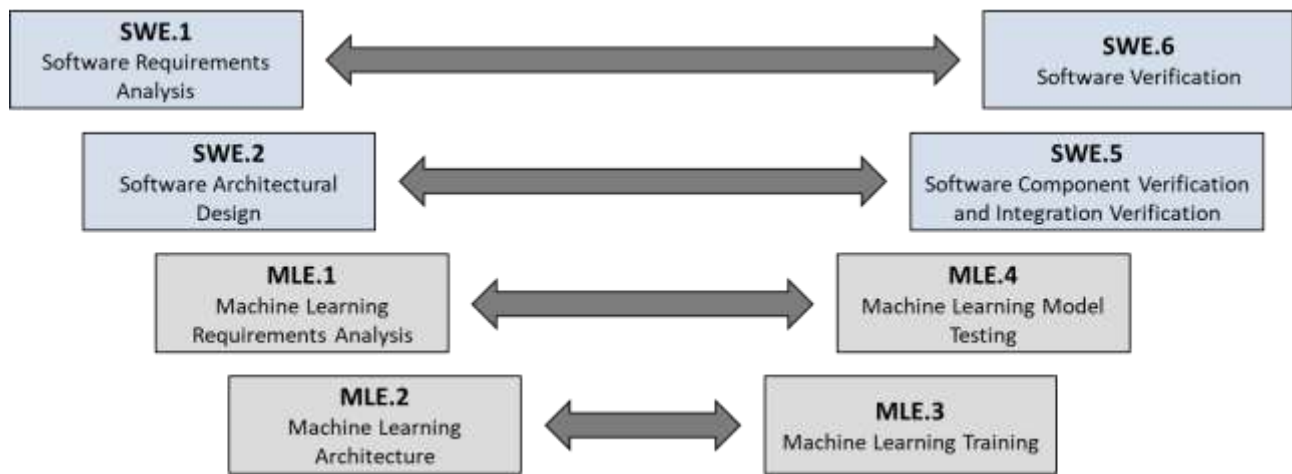


Figure 6 — Annex C.3: Integration of MLE Processes

The second figure shows the interdependencies within the MLE “Machine Learning Engineering” process group and to the SUP.11 “Machine Learning Data Management”.

With MLE.1 “Machine Learning Requirements Analysis”, machine learning related Software requirements allocated to the ML-based software elements need to be refined into a set of ML requirements. These requirements must contain ML data requirements which are input for the SUP.11 “Machine Learning Data Management” and other ML requirements which are input for the other MLE “Machine Learning Engineering” processes.

By applying the SUP.11 “Machine Learning Data Management” process ML data with assured quality and integrity, which fulfill the ML data requirements, are collected, processed, and made available to all affected parties.

The other ML requirements should be used within the MLE.2 “Machine Learning Architecture” process to develop an ML architecture supporting training and deployment. Therefore, the ML architecture must contain all necessary ML architectural elements like hyperparameter ranges and initial values, details of the ML model, and possible other software parts which are necessary for MLE.3 “Machine Learning Training”. These other software parts should be developed according to SWE.3 “Software Detailed Design & Unit Construction” and SWE.4 “Software Unit Verification”.

Performing MLE.3 “Machine Learning Training” should start with specifying an ML training and validation approach. Based on this approach a training and validation dataset need to be created from the ML data pool provided by SUP.11 “Machine Learning Data Management” which is then used iteratively to optimize the ML model weights and hyperparameter values. When training exit criteria are reached the trained model should be agreed and communicated to all affected parties.

MLE.4 “Machine Learning Model Testing” focuses on testing the agreed trained model to ensure compliance with the ML requirements. Therefore, an ML test approach needs to be specified and an ML test dataset must be created from the provided ML data pool.

The ML test approach defines besides other details the distribution of data characteristics (e.g., gender, weather conditions, street conditions within the ODD) defined by ML requirements. The test dataset contains different test scenarios applying the required distribution of data characteristics, e.g., driving during rain on a gravel road.

After successful testing the trained model, a deployed model is derived and tested as well. The deployed model will be integrated into the target system and may differ from the trained model which often requires powerful hardware and uses interpretative languages. Finally, the agreed test results and the deployed model must be communicated to all affected parties, so that the deployed model can be integrated with the other software units by applying SWE.5 “Software Integration and Integration Test”.

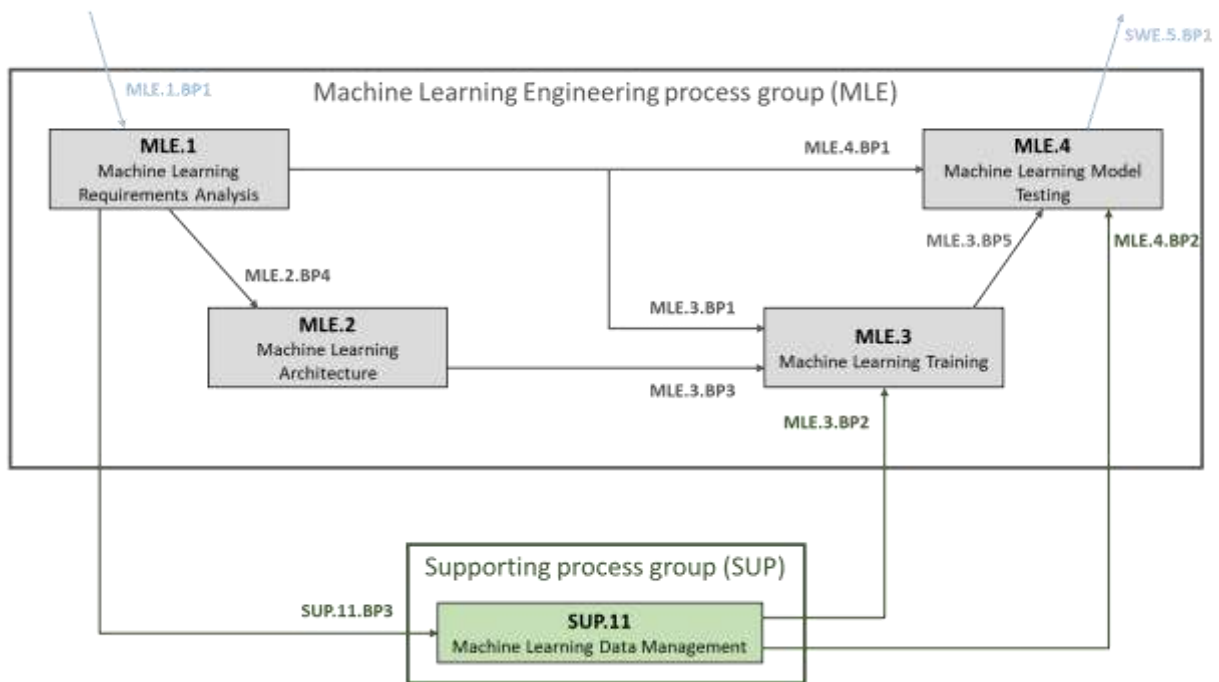


Figure 7 — Annex C.3: Interdependencies within MLE and SUP.11

Annex C.3 Example of an ML Architecture

The following figure shows an example of an ML architecture, describing the overall structure of the ML-based software element and the interfaces within the ML-based software element and to other software elements. The ML architecture typically consists of an ML model and other ML architectural elements, which are other (classical) software components developed according to SWE.3 “Software Detailed Design & Unit Construction” and SWE.4 “Software Unit Verification” and provided to train, deploy and test the ML model. Furthermore, the ML architecture describes details of the ML model like used layers, activation functions, loss function, and backpropagation.

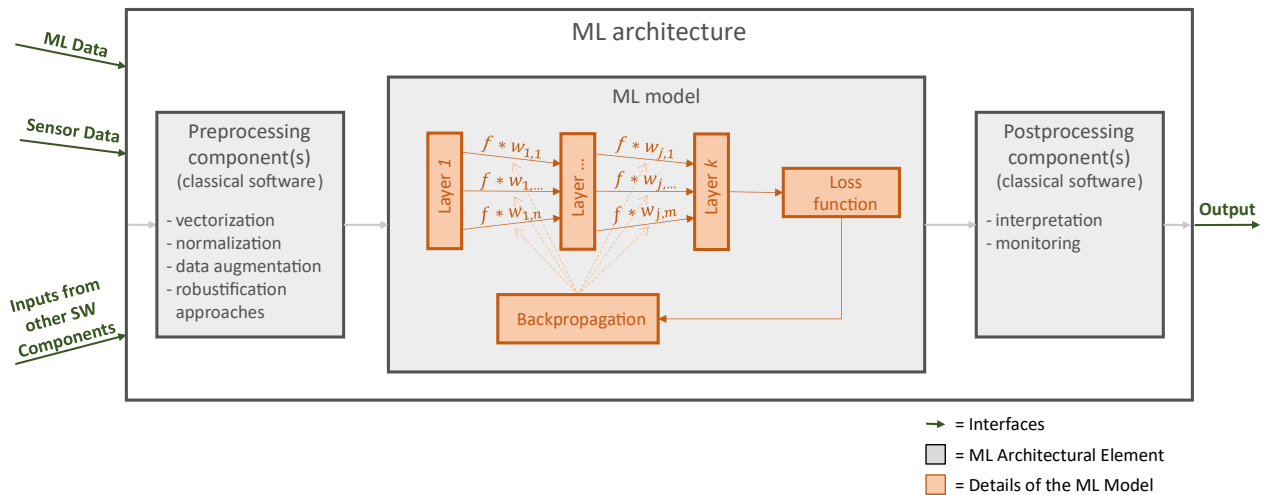


Figure 8 — Annex C.4: Example of an ML Architecture

During training, hyperparameters defining the ML model will be tuned therefore it is recommended to define ranges of hyperparameter values and initial values for training start are defined. Because developing an ML-based software element is highly iterative, changes in the ML architecture might come up.

Furthermore, the ML architecture used for training can differ from the architecture of the deployed model, which will be integrated with the other software elements, these differences are part of the ML architecture as well.

Annex C.4 Traceability and consistency

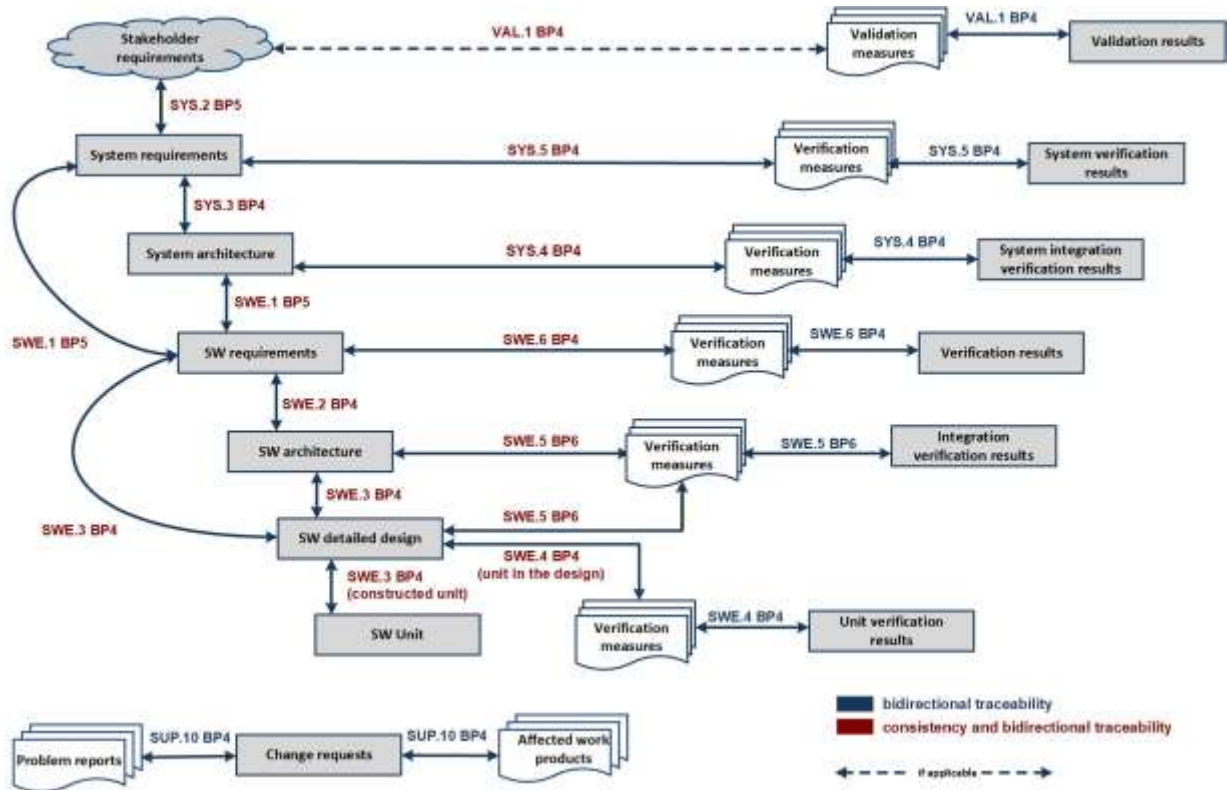


Figure 9 — Annex C.4: Consistency and traceability between system and software information items

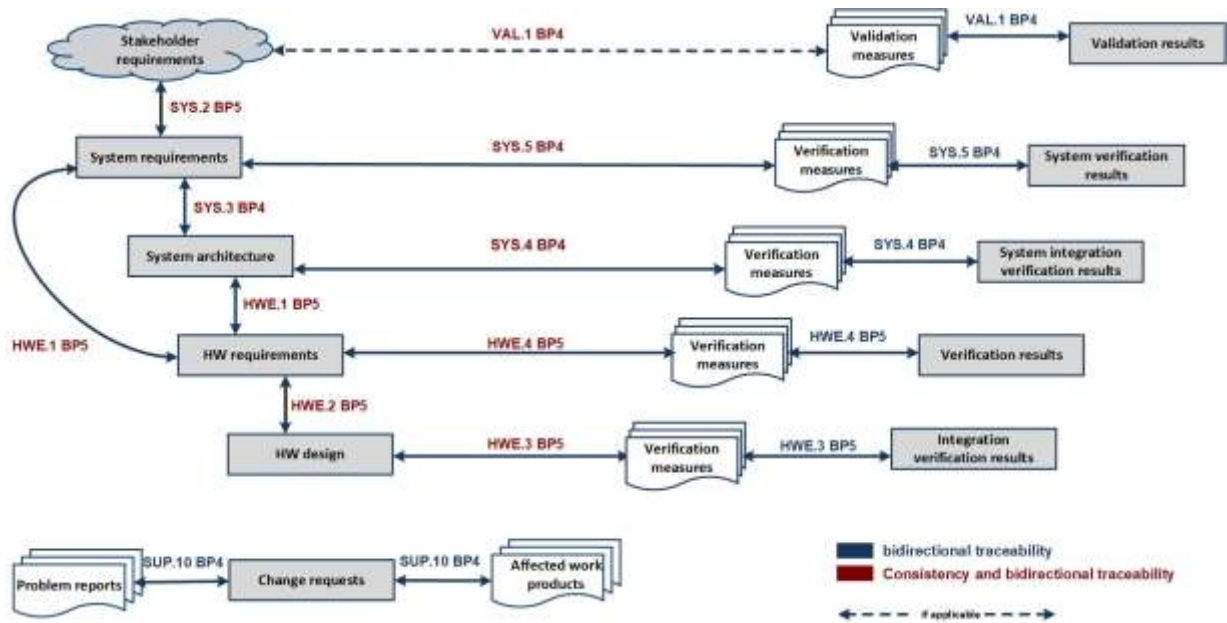


Figure 10 — Annex C.4: Consistency and traceability between system and hardware information items

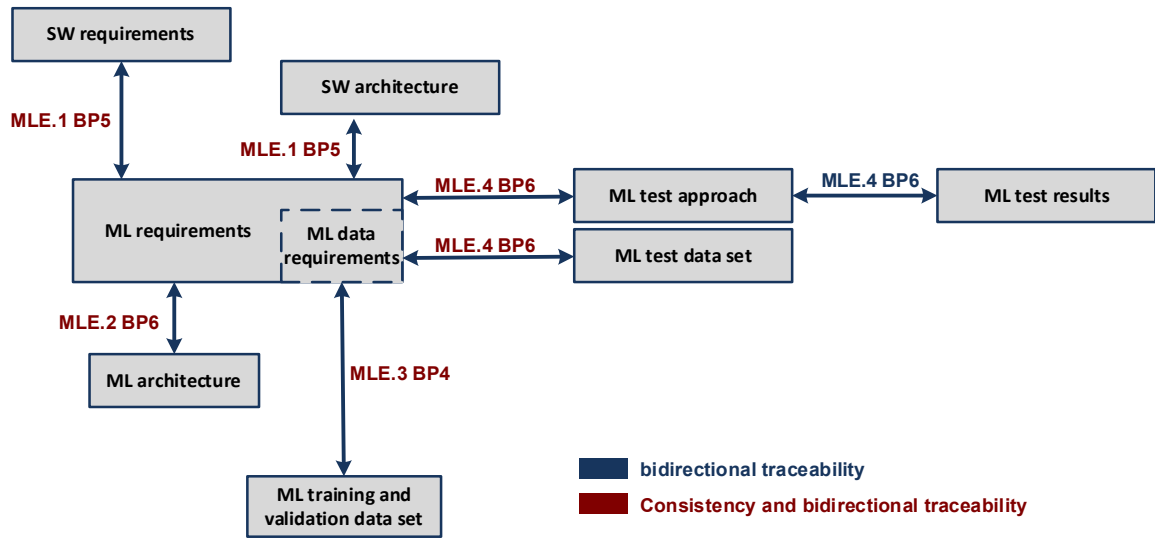


Figure 11 — Annex C.4: Consistency and traceability between ML work products

Annex D Reference standards

Annex D provides a list of reference standards and guidelines that support implementation of the Automotive SPICE® PAM / PRM.

ISO/IEC 33001:2015	Information technology -- Process assessment – Concepts and terminology
ISO/IEC 33002:2015	Information technology -- Process assessment – Requirements for performing process assessment
ISO/IEC 33003:2015	Information technology -- Process assessment – Requirements for process measurement frameworks
ISO/IEC 33004:2019	Information technology -- Process assessment – Requirements for process reference, process assessment and maturity models
ISO/IEC 33020:2019	Information technology -- Process assessment – Process measurement framework for assessment of process capability
ISO/IEC/IEEE 24765:2017	Systems and software engineering -- Vocabulary
ISO/IEC/IEEE 29148:2018	Systems and software engineering – Life cycle processes – Requirements engineering
INCOSE Guide for Writing Requirements	https://www.incose.org/
PAS 1883:2020	Operational design domain (ODD) taxonomy for an automated driving system (ADS)
ISO 26262:2018	Road vehicles — Functional safety, Second edition 2018-12
ISO/SAE 21434:2021	Road vehicles — Cybersecurity Engineering, First edition 2021-08

Table D.1 — Referenced standards