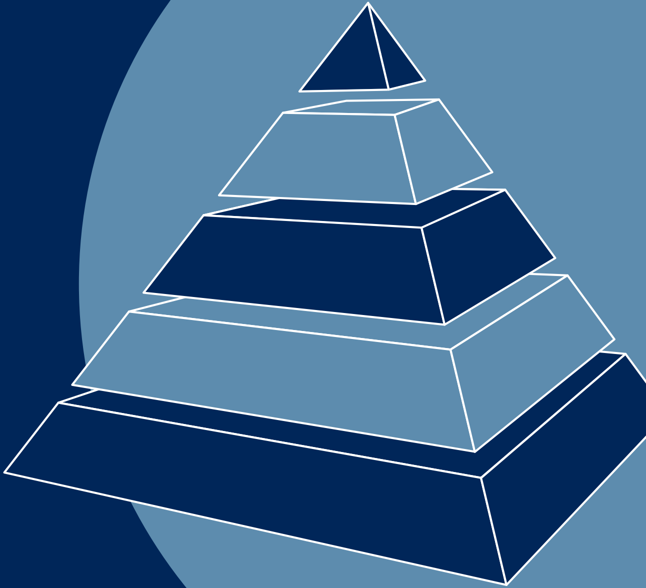
Automotive SPICE® Pocket Guide

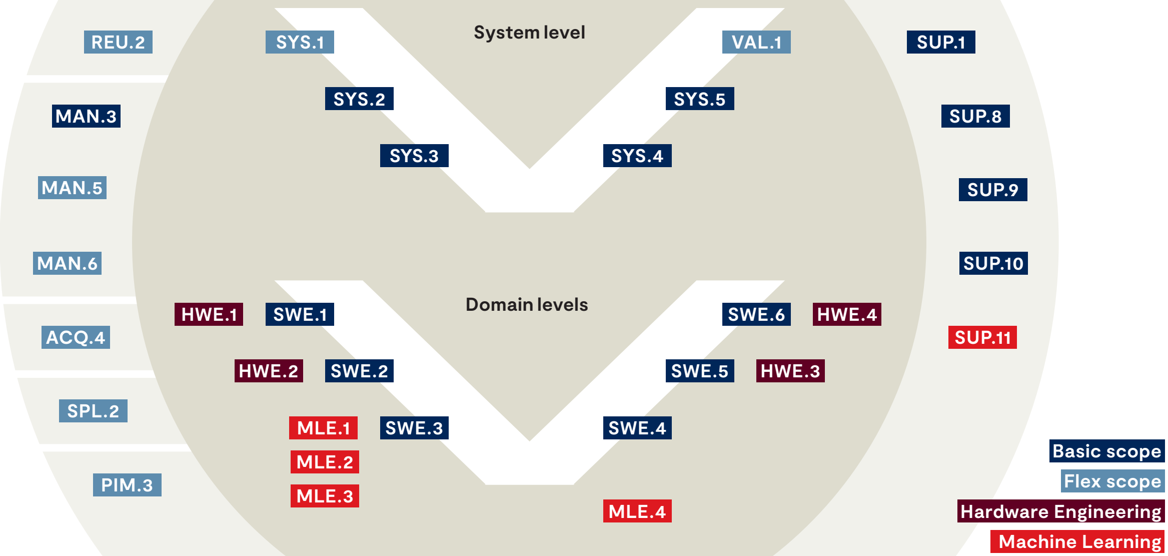
Process assessment model
Version 4.0



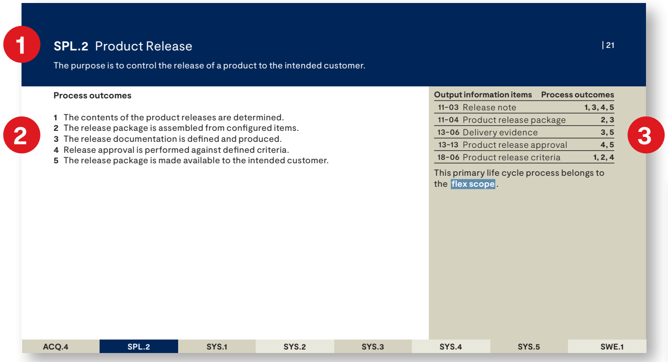
UL Solutions

KUGLER MAAG CIE by UL Solutions
kVA by UL Solutions
methodpark by UL Solutions

Automotive SPICE v4.0 Process overview



Automotive SPICE v4.0 Process structure


1 **SPL.2** Product Release | 21

The purpose is to control the release of a product to the intended customer.

Process outcomes

2
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. Release approval is performed against defined criteria.
5. The release package is made available to the intended customer.

Output information items		Process outcomes
11-03	Release note	1, 3, 4, 5
11-04	Product release package	2, 3
13-06	Delivery evidence	3, 5
13-13	Product release approval	4, 5
18-06	Product release criteria	1, 2, 4

3

This primary life cycle process belongs to the flex scope.

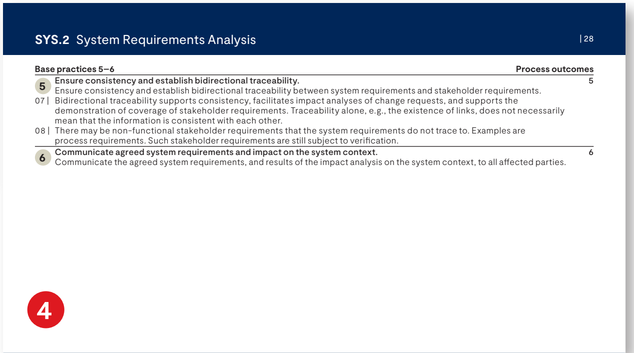
ACQ.4 SPL.2 SYS.1 SYS.2 SYS.3 SYS.4 SYS.5 SWE.1

1
What is the process needed for?

2
Which work products could potentially be delivered to prove that the process purpose has been achieved?

3
Which results should be achieved by the process?

4
Based on experience, what could be done for generating the process outcomes and fulfilling the process purpose?


SYS.2 System Requirements Analysis | 28

Base practices 5–6 — **Process outcomes**

5 **Ensure consistency and establish bidirectional traceability.** — 5
Ensure consistency and establish bidirectional traceability between system requirements and stakeholder requirements.
07 | Bidirectional traceability supports consistency, facilitates impact analyses of change requests, and supports the demonstration of coverage of stakeholder requirements. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent with each other.
08 | 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.

6 **Communicate agreed system requirements and impact on the system context.** — 6
Communicate the agreed system requirements, and results of the impact analysis on the system context, to all affected parties.

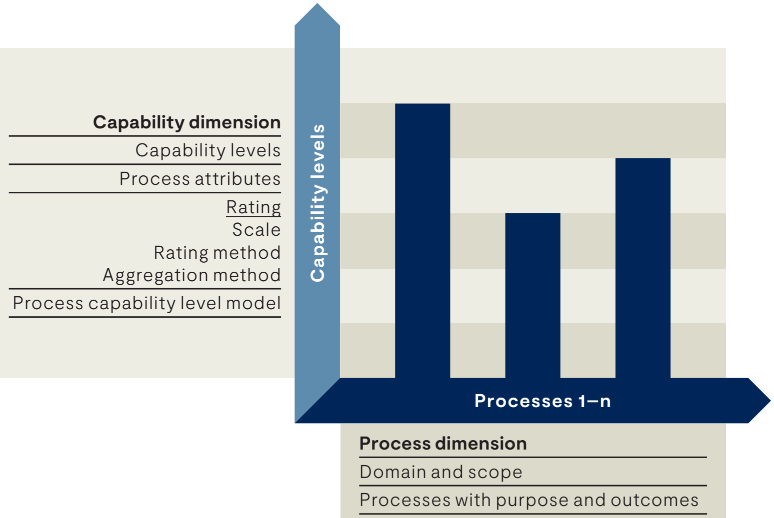
4

Automotive SPICE®
for rookies

Are you not familiar with Automotive SPICE®?
Before you carefully study the following pages, watch our video. The featured protagonist, Lisa, will tell you from scratch how she implemented Automotive SPICE® in her company.

Automotive SPICE® Model dimensions

The concept of process capability determination by using the Automotive SPICE® assessment model is based on a two-dimensional framework. The framework consists of a process dimension and a capability dimension.



Process dimension Process overview

For a practical overview, please refer to page **130**. The illustration on this page maps the processes to the lifecycle phases, a logic that originates from the former ISO/IEC 15504.

Primary life cycle processes (LCP)

Acquisition processes

ACQ.4	Supplier Monitoring

Validation processes

VAL.1	Validation

Hardware Engineering processes

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

Machine Learning Engineering processes

MLE.1	Machine Learning Requirements Analysis
MLE.2	Machine Learning Architecture
MLE.3	Machine Learning Training
MLE.4	Machine Learning Model Testing

Supply processes

SPL.2	Product Release

Systems Engineering processes

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

Software Engineering processes

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
SWE.6	Software Verification

Supporting LCP

Supporting processes

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

Organizational LCP

Management processes

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

Processes Improvement processes

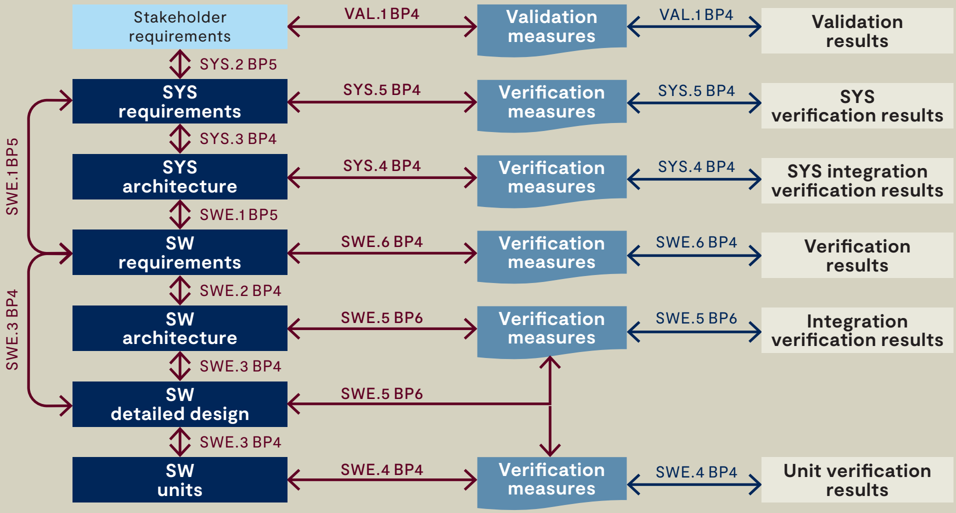
PIM.3	Process Improvement

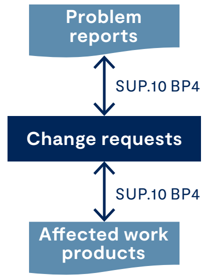
Reuse processes

REU.2	Management of Products for Reuse

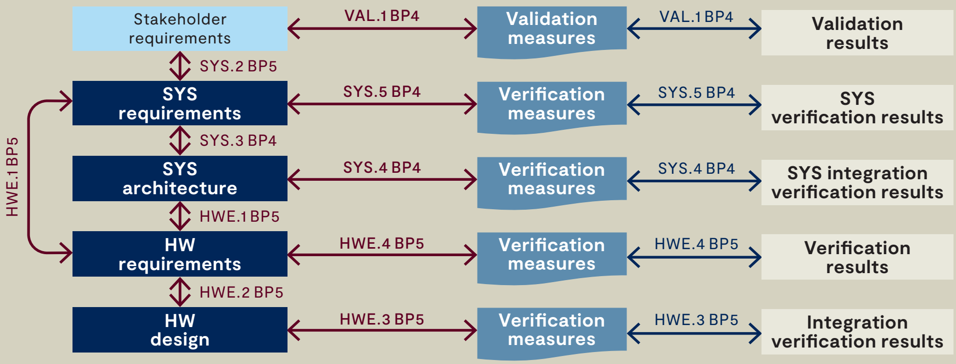
Automotive SPICE® Traceability and consistency concept

Traceability and consistency are both addressed in Automotive SPICE®. Traceability refers to the existence of meaningful references or links between work products. Consistency on the other hand addresses content and semantics.





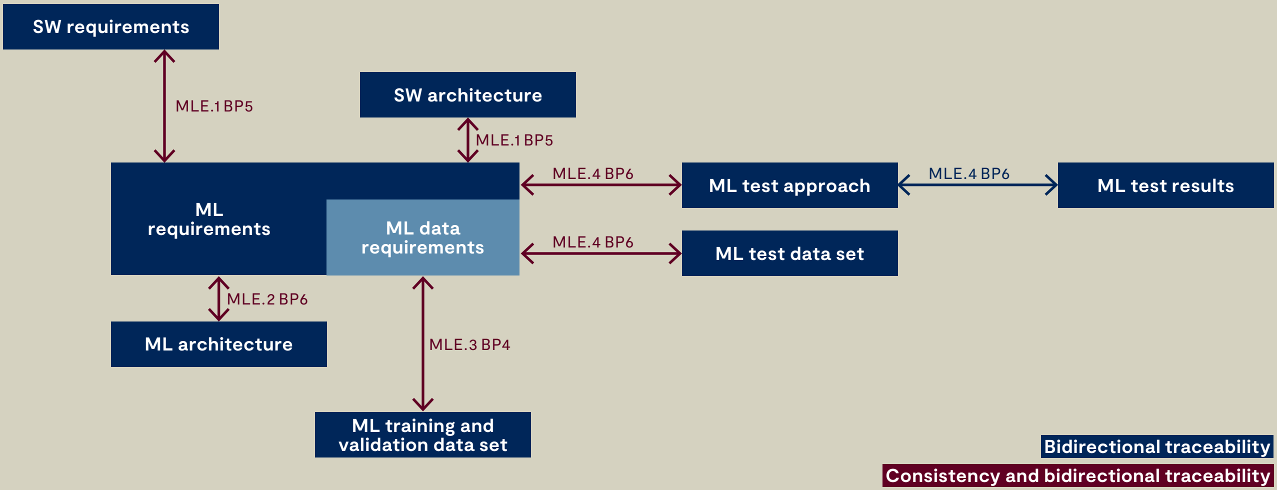
Consistency and traceability between system and software work products



Bidirectional traceability
Consistency and bidirectional traceability

Consistency and traceability between system and hardware work products

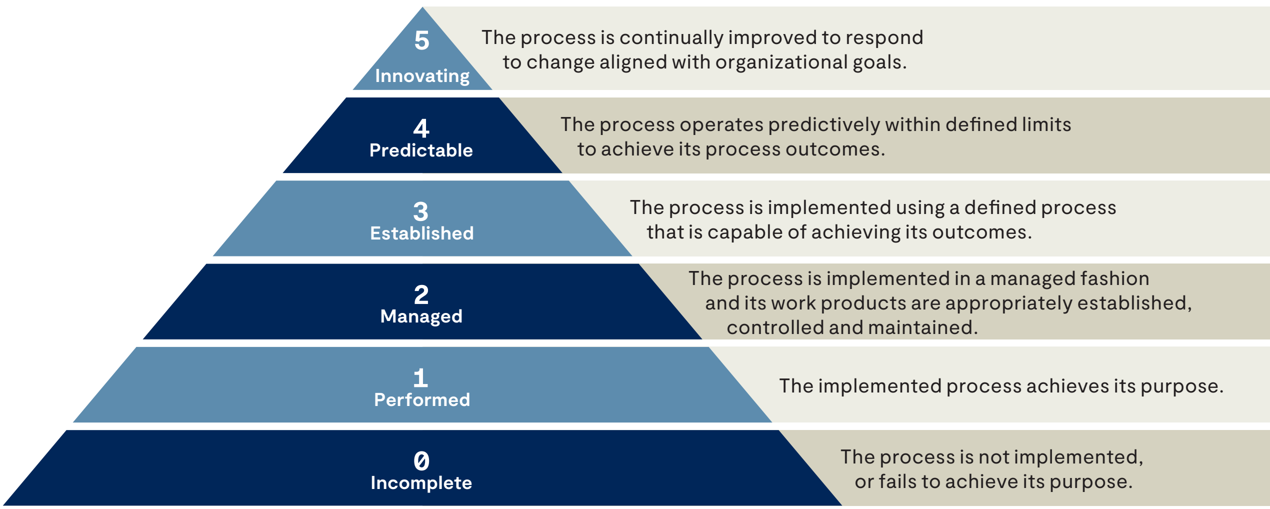
Automotive SPICE® Traceability and consistency concept



Consistency and traceability between ML work products

Capability dimension Overview

The capability dimension consists of capability levels (CL) which are further subdivided into process attributes (PA). These PAs provide the measurable characteristics to determine the process capability.



Process capability Determination

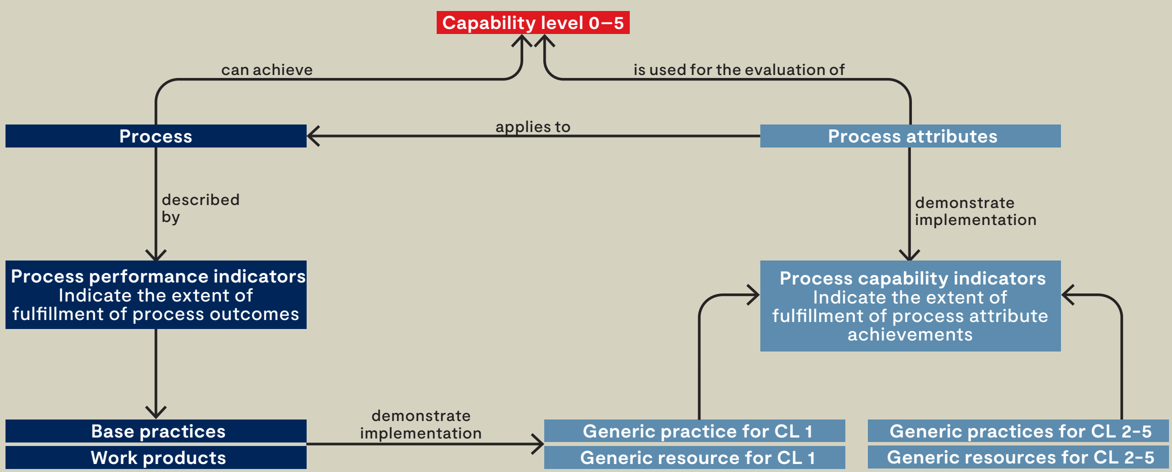
The capability levels are determined by rating the process attribute for each capability level.
The rating is made using the NPLF rating scale. A capability level is achieved when its process attributes are rated with an L or F and all process attributes of lower capability levels are rated with an F.

	PA1.1	PA2.1	PA2.2	PA3.1	PA3.2
Capability level 3	F	F	F	L / F	L / F
Capability level 2	F	L / F	L / F		
Capability level 1	L / F				

	The NPLF rating scale	
N	Process attribute **not** achieved	**0 to ≤ 15% achievement**
P	Process attribute **partially** achieved	**> 15% to ≤ 50% achievement**
L	Process attribute **largely** achieved	**> 50% to ≤ 85% achievement**
F	Process attribute **fully** achieved	**> 85% to ≤ 100% achievement**

Process capability Determination indicators

Automotive SPICE® offers indicators that are used by the assessors to determine whether a certain capability level is achieved.
The indicators should not be considered as a mandatory set of checklists to be followed.



Automotive SPICE® and functional safety

By implementing Automotive SPICE®, a large part of the ISO 26262 (Road vehicles – Functional safety) requirements can also be fulfilled. The table below displays the Automotive SPICE® support for an ISO 26262 implementation.

Automotive SPICE®			ISO 26262 (Road vehicles – Functional safety)
SYS.1	Requirements Elicitation	strong support	Item definition (detailed level)
SYS.2	System Requirements Analysis	weak support	Functional safety concept
		weak support	Specification of the technical safety requirements
		medium support	Specification and management of safety requirements
SYS.3	System Architectural Design	medium support	System design
SWE.1	Software Requirements Analysis	medium support	Specification of software safety requirements
SWE.2	Software Architectural Design	medium support	Software architectural design
SWE.3	Software Detailed Design and Unit Construction	medium support	Software unit design & implementation
SWE.4	Software Unit Verification	medium support	Software unit testing
SWE.5	Software Component and Integration Verification	medium support	Software integration & testing
SWE.6	Software Verification	medium support	Verification of software safety requirements
SYS.4	System Integration Verification and Integration Verification	medium support	Item integration and testing
SYS.5	System Verification	medium support	–
ACQ.4	Supplier Monitoring	medium support	Interfaces within distributed developments
SPL.2	Product Release	strong support	Release for production
MAN.3	Project Management	medium support	Safety management during the concept phase and the product development
		strong support	Item definition (top level)
		medium support	Initiation of the safety lifecycle
		medium support	Initiation of product development at the system level
		medium support	Initiation of product development at the hardware level
		medium support	Initiation of product development at the software level
MAN.5	Risk Management		–
SUP.1	Quality Assurance	medium support	Safety management during the concept phase and the product development
			Functional safety assessment
SUP.8	Configuration Management	medium support	Configuration management
SUP.9	Problem Resolution Management		–
SUP.10	Change Request Management	medium support	Change management
SUP.11	Machine Learning Data Management		–
REU.2	Management of Products for Reuse		–
VAL.1	Validation		–

● strong support ● medium support ● weak support

Automotive SPICE®

Processes for Systems, Software, Hardware and Machine Learning Engineering incl. Acquisition, Supply, Support and Management processes

ACQ.4 Supplier Monitoring

The purpose is to track and assess the performance of an external contract-based supplier company against agreed commitments.

Process outcomes

1 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 Changes to the agreement, if needed, are negotiated between the customer and the supplier and documented in the agreement.

Output information items		Process outcomes
02-01	Commitment/Agreement	1, 2, 3, 4
13-52	Communication evidence	1, 2, 3
13-09	Meeting support evidence	1, 2
13-14	Progress status	2, 3
13-16	Change Request	4
13-19	Review evidence	2
14-02	Corrective action	4
15-51	Analysis results	3

This primary life cycle process belongs to the **basic scope**.

We have published a video tutorial on this process for better understanding.

ACQ.4 Supplier Monitoring

Base practices 1–5		Process outcomes
1	**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.	1, 2, 4
2	**Exchange all agreed information.** Use the defined joint interfaces between customer and supplier for the exchange of all agreed information.	1, 2, 3
3	**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 open measures. 01 | See SUP.9 for management of problems	1, 3, 4
4	**Review progress of the supplier.** Review progress of the supplier regarding schedule, quality, and cost on the agreed regular basis. Track open measures to closure and perform risk mitigation activities. 02 | See MAN.5 for management of risks	1, 3, 4
5	**Act to correct deviations.** Take action when agreed objectives are not achieved. Negotiate changes to objectives and document them in the agreements.	3, 4

01 | This numbering indicates a note: Non-obligatory, but helpful aid to application

SPL.2 Product Release

The purpose is to control the release of a product to the intended customer.

Process outcomes

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 Release approval is performed against defined criteria.
5 The release package is made available to the intended customer.

Output information items		Process outcomes
11-03	Release note	1, 3, 4, 5
11-04	Product release package	2, 3
13-06	Delivery evidence	3, 5
13-13	Product release approval	4, 5
18-06	Product release criteria	1, 2, 4

This primary life cycle process belongs to the **flex scope**.

Base practices 1–6		Process outcomes
1	**Define the functional content of releases.** Define the functionality to be included and the release criteria for each release.	1
01 |	This may include the hardware elements, software elements, and extra application parameter files (influencing the identified system functionality) that are needed for the release.	
2	**Define release package.** Define the release as well as supporting tools and information.	1
02 |	The release package may include also programming tools.	
3	**Ensure unique identification of releases.** Ensure a unique identification of the release based upon the intended purpose and expectations of the release.	3
03 |	Unique identification may be realized by a classification and numbering scheme for product releases.	
4	**Build the release from items under configuration control.** Build the release from items under configuration control to ensure integrity.	2
04 |	This practice may be supported by the SUP.8 Configuration Management Process.	
5	**Ensure release approval before delivery.** Criteria for the release are satisfied before delivery takes place.	4
6	**Provide a release note.** A release is accompanied by information detailing key characteristics of the release.	3, 5
05 |	The release note may include information about legal aspects like relevant target markets, legislation that is considered etc. See also VAL.1 Validation.	

Base practices 7–8		Process outcomes
7	**Communicate the type, service level and duration of support for a release.** Identify and communicate the type, service level and duration of support for a release.	3, 5
8	**Deliver the release package to the intended customer.** Deliver the release package to the intended customer.	5
01 |	The intended customer may be an internal organizational unit or an external organization.	

SYS.1 Requirements Elicitation

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

1 Continuing communication with the stakeholder is established.
2 Stakeholder expectations are understood, and requirements are defined and agreed.
3 Stakeholder requirements changes arising from stakeholder needs are analyzed to enable associated risk assessment and impact management.
4 Determination of stakeholder requirements status is ensured for all affected parties.

Output information items		Process outcomes
15-51	Analysis results	3
13-52	Communication evidence	1, 2
17-00	Requirement	2
17-54	Requirement attribute	2, 3, 4

This primary life cycle process belongs to the flex scope.

Base practices 1–5		Process outcomes
1	**Obtain stakeholder expectations and requests.** Obtain and define 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 stakeholder requirements, and consideration of the target operating and hardware environment.	1
01 |	Documenting the stakeholder, or the source of a stakeholder requirement, supports stakeholder requirements agreement and change analysis (see BP2 and BP3).	
2	**Agree on requirements.** Formalize the stakeholder's expectations and requests into requirements. Reach a common understanding of the set of stakeholder requirements among affected parties by obtaining an explicit agreement from all affected parties.	2
02 |	Examples of affected parties are customers, suppliers, design partners, joint venture partners, or outsourcing parties.	
03 |	The agreed stakeholder requirements may be based on feasibility studies and/or cost and schedule impact analysis.	
3	**Analyze stakeholder requirements changes.** Analyze all changes made to the stakeholder requirements against the agreed stakeholder requirements. Assess the impact and risks, and initiate appropriate change control and mitigation actions.	3
04 |	Requirements changes may arise from different sources as for instance changing technology, stakeholder needs, or legal constraints.	
05 |	Refer to SUP.10 Change Request Management, if required.	
4	**Communicate requirements status.** Ensure all affected parties can be aware of the status and disposition of their requirements including changes and can communicate necessary information and data.	1, 4

SYS.2 System Requirements Analysis

The purpose is to establish a structured and analyzed set of system requirements consistent with the stakeholder requirements.

Process outcomes

1 System requirements are specified.
2 System requirements are structured and prioritized.
3 System requirements are analyzed for correctness and technical feasibility.
4 The impact of system requirements on the operating environment is analyzed.
5 Consistency and bidirectional traceability are established between system requirements and stakeholder requirements.
6 The system requirements are agreed and communicated to all affected parties.

Output information items		Process outcomes
17-00	Requirement	1, 2
17-54	Requirement attribute	2, 3
15-51	Analysis results	3, 4
13-51	Consistency evidence	5
13-52	Communication evidence	6

This primary life cycle process belongs to the **basic scope**.

We have published a video tutorial on this process for better understanding.

Base practices 1–4	Process outcomes
1 Specify system requirements. Use the stakeholder requirements to identify and document the functional and non-functional requirements for the system according to defined characteristics for requirements. 01 | Characteristics of requirements are defined in standards such as ISO IEEE 29148, ISO 26262-8:2018, or the INCOSE Guide For Writing Requirements. 02 | 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.)	1
2 Structure system requirements. Structure and prioritize the system requirements. 03 | Examples for structuring criteria can be grouping (e.g. by functionality) or product variants identification. 04 | Prioritization can be done according to project or stakeholder needs via e.g. definition of release scopes. Please refer to SPL.2.BP1.	2
3 Analyze system requirements. Analyze the specified system requirements including their interdependencies to ensure correctness, technical feasibility, and to support project management regarding project estimates. 05 | See MAN.3.BP3 for project feasibility and MAN.3.BP5 for project estimates. 06 | Technical feasibility can be evaluated based on e.g. platform or product line, or by means of prototype development or product demonstrators.	3
4 Analyze the impact on the system context. Analyze the impact that the system requirements will have on elements in the relevant system context.	4

Base practices 5–6		Process outcomes
5	**Ensure consistency and establish bidirectional traceability.** Ensure consistency and establish bidirectional traceability between system requirements and stakeholder requirements.	5
07 |	Bidirectional traceability supports consistency, facilitates impact analyses of change requests, and supports the demonstration of coverage of stakeholder requirements. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent with each other.	
08 |	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.	
6	**Communicate agreed system requirements and impact on the system context.** Communicate the agreed system requirements, and results of the impact analysis on the system context, to all affected parties.	6

SYS.3 System Architectural Design

The purpose is to establish an analyzed system architecture, comprising static and dynamic aspects, consistent with the system requirements.

Process outcomes

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 defined criteria, and special characteristics are identified.
3. Consistency and bidirectional traceability are established between system architecture and system requirements.
4. The agreed system architecture and the special characteristics are communicated to all affected parties.

Output information items		Process outcomes
04-06	System architecture	1
13-51	Consistency evidence	3
13-52	Communication evidence	4
15-51	Analysis results	2
17-57	Special characteristics	2

This primary life cycle process belongs to the **basic scope**.

We have published a video tutorial on this process for better understanding.

Base practices 1–3 — **Process outcomes**

1 **Specify static aspects of the system architecture.** — 1
Specify and document the static aspects of the system architecture with respect to the functional and non-functional system requirements, including external interfaces and a defined set of system elements with their interfaces and relationships.

2 **Specify dynamic aspects of the system architecture.** — 1
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.
01 | 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.

3 **Analyze system architecture.** — 2
Analyze the system architecture regarding relevant technical design aspects related to the product lifecycle, and to support project management regarding project estimates, and derive special characteristics for non-software elements. Document a rationale for the system architectural design decisions.
02 | See MAN.3.BP3 for project feasibility and MAN.3.BP5 for project estimates.
03 | Examples for product lifecycle phases are production, maintenance & repair, decommissioning.
04 | Examples for technical aspects are manufacturability for production, suitability of pre-existing system elements to be reused, or availability of system elements.
05 | Examples for methods being suitable for analyzing technical aspects are prototypes, simulations, and qualitative analyses (e.g. FMEA approaches)
06 | 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).

Base practices 4–5 — **Process outcomes**

4 **Ensure consistency and establish bidirectional traceability.** — 3
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.
07 | Bidirectional traceability further supports consistency, and facilitates impact analysis of change requests, and demonstration of verification coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent with each other.
08 | There may be non-functional requirements that the system architectural design does not trace to. Examples are do not address, or represent, direct properties or characteristics of the physical end product. Such requirements are still subject to verification.

5 **Communicate agreed system architecture.** — 4
Communicate the agreed system architecture, including the special characteristics, to all affected parties.

SYS.4 System Integration and Integration Verification

The purpose is to integrate systems elements and verify that the integrated system elements are consistent with the system architecture.

Process outcomes

1. Verification measures are specified for system integration verification of the integrated system elements based on the system architecture, including the interfaces of, and interactions between, system elements.
2. 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. 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 verification results and verification measures is established.
7. Results of the system integration and integration verification are summarized and communicated to all affected parties.

Output information items	Process outcomes
08-60 Verification measure	1
06-50 Integration sequence instruction	2
03-50 Verification measure data	4
08-58 Verification measure selection set	3
15-52 Verification results	4
13-51 Consistency evidence	5, 6
13-52 Communication evidence	7
11-06 Integrated system	2

This primary life cycle process belongs to the **basic scope**.

We have published a video tutorial on this process for better understanding.

Base practices 1–2	**Process outcomes**
1 Specify verification measures for system integration. Specify the verification measures, based on a defined sequence and preconditions for the integration of system elements against the system static and dynamic aspects of the system architecture, including → techniques for the verification measures; → pass/fail criteria for verification measures; → a definition of entry and exit criteria for the verification measures, and → the required verification infrastructure and environment setup. 01 | Examples on what a verification measure may focus 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.	1
2 Select verification measures. Document the selection of verification measures for each integration step considering selection criteria including criteria for regression verification. The documented selection of verification measures shall have sufficient coverage according to the release scope. 02 | Examples for 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.)	3

Base practices 3–5	Process outcomes
3 Integrate system elements and perform 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 defined preconditions. Perform the selected system integration verification measures. Record the verification measure data including pass/fail status and corresponding verification measure data. 03 | Examples for preconditions for starting system integration can be successful system element verification or qualification of pre-existing system elements. 04 | See SUP.9 for handling verification results that deviate from expected results	2, 4
4 Ensure consistency and establish bidirectional traceability. Ensure consistency and establish bidirectional traceability between verification measures and the system architecture. Establish bidirectional traceability between verification results and verification measures. 05 | Bidirectional traceability supports consistency, and facilitates impact analysis of change requests, and demonstration of verification coverage.Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent with each other.	5, 6
5 Summarize and communicate results. Summarize the system integration and integration verification results and communicate them to all affected parties. 06 | Providing all necessary information from the test case execution in a summary enables other parties to judge the consequences.	7

SYS.5 System Verification

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

Process outcomes

1 Verification measures are specified for system 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 system verification are recorded.
4 Consistency and bidirectional traceability are established between verification measures and system requirements.
5 Bidirectional traceability is established between verification results and verification measures.
6 Verification results are summarized and communicated to all affected parties.

Output information items	Process outcomes
08-60 Verification measure	1
08-58 Verification measure selection set	2
03-50 Verification measure data	3
15-52 Verification results	3
13-51 Consistency evidence	4, 5
13-52 Communication evidence	6

This primary life cycle process belongs to the **basic scope**.

We have published a video tutorial on this process for better understanding.

Base practices 1–3	Process outcomes
1 **Specify verification measures for system verification.** Specify the verification measures for system verification suitable to provide evidence for compliance with the functional and non-functional information in the system requirements, including → techniques for the verification measures; → pass/fail criteria for verification measures; → a definition of entry and exit criteria for the verification measures; → necessary sequence of verification measures, and → the required verification infrastructure and environment setup. 01 | The system verification measures may cover aspects such as thermal, environmental, robustness/lifetime, and EMC.	1
2 **Select verification measures.** Document the selection of verification measures considering selection criteria including criteria for regression verification. The selection of verification measures shall have sufficient coverage according to the release scope. 02 | 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.)	2
3 **Perform 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 corresponding verification measure data. 03 | See SUP.9 for handling verification results that deviate from expected results	3

Base practices 4–5	Process outcomes
4 **Ensure consistency and establish bidirectional traceability.** Ensure consistency and establish bidirectional traceability between verification measures and system requirements. Establish bidirectional traceability between verification results and verification measures. 05 | Bidirectional traceability supports consistency, and facilitates impact analysis of change requests, and demonstration of verification coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent with each other.	4, 5
5 **Summarize and communicate results.** Summarize the system verification results and communicate them to all affected parties. 06 | Providing all necessary information from the test case execution in a summary enables other parties to judge the consequences.	6

SWE.1 Software Requirements Analysis

The purpose is to establish a structured and analyzed set of software requirements consistent with the system requirements, and the system architecture.

Process outcomes

1 Software requirements are specified.
2 Software requirements are structured and prioritized.
3 Software requirements are analyzed for correctness and technical feasibility.
4 The impact of software requirements on the operating environment is analyzed.
5 Consistency and bidirectional traceability are established between software requirements and system requirements.
6 Consistency and bidirectional traceability are established between software requirements and system architecture.
7 The software requirements are agreed and communicated to all affected parties.

Output information items		Process outcomes
17-00	Requirement	1, 2
17-54	Requirement attribute	2
15-51	Analysis results	3, 4
13-51	Consistency evidence	5, 6
13-52	Communication evidence	7

This primary life cycle process belongs to the **basic scope**.

We have published a video tutorial on this process for better understanding.

Base practices 1–3		Process outcomes
1	**Specify 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 defined characteristics for requirements.	1
01 |	Characteristics of requirements are defined in standards such as ISO IEEE 29148, ISO 26262-8:2018, or the INCOSE Guide for Writing Requirements.	
02 |	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).	
03 |	In case of software-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 software.	
04 |	The hardware-software-interface (HSI) definition puts in context hardware and therefore iits an interface decision at the system design level. If such a HSI exists, then it may provide input to software requirements.	
2	**Structure software requirements.** Structure and prioritize the software requirements.	2
05 |	Examples for structuring criteria can be grouping (e.g. by functionality) or expressing product variants.	
06 |	Prioritization can be done according to project or stakeholder needs via e.g. definition of release scopes. Refer to SPL.2.BP1.	
3	**Analyze software requirements.** Analyze the specified software requirements including their interdependencies to ensure correctness, technical feasibility, and to support project management regarding project estimates.	3
07 |	See MAN.3.BP3 for project feasibility and MAN.3.BP5 for project estimates.	
08 |	Technical feasibility can be evaluated based on e.g. platform or product line, or by prototyping.	

Base practices 4–6		Process outcomes
4	**Analyze the impact on the operating environment.** Analyze the impact that the software requirements will have on elements in the operating environment.	4
5	**Ensure consistency and establish bidirectional traceability.** Ensure consistency and establish bidirectional traceability between software requirements and system architecture. Ensure consistency and establish bidirectional traceability between software requirements and system requirements.	5, 6
09 |	Redundant traceability is not intended.	
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.	
11 |	Bidirectional traceability supports consistency, and facilitates impact analysis of change requests, and demonstration of verification coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent with each other.	
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.	
6	**Communicate agreed software requirements and 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.	7

SWE.2 Software Architectural Design

The purpose is to establish an analyzed software architecture, comprising static and dynamic aspects, consistent with the software requirements.

Process outcomes

1. A software architecture is designed including static and dynamic aspects.
2. The software architecture is analyzed against defined criteria.
3. Consistency and bidirectional traceability are established between software architecture and software requirements.
4. The software architecture is agreed and communicated to all affected parties.

Output information items	Process outcomes
04-04 Software architecture	1
13-51 Consistency evidence	3
13-52 Communication evidence	4
15-51 Analysis results	2

This primary life cycle process belongs to the **basic scope**.

We have published a video tutorial on this process for better understanding.

Base practices 1–3 **Process outcomes**

1 Specify static aspects of the software architecture. 1
Specify and document the static aspects of the software architecture with respect to the functional and non-functional software requirements, including external interfaces and a defined set of software components with their interfaces and relationships.
01 | The hardware-software-interface (HSI) definition puts in context the hardware design and therefore is an aspect of system design (SYS.3).

2 Specify dynamic aspects of the software architecture. 1
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.
02 | Examples for concurrency aspects are application-relevant interrupt handling, preemptive processing, multi-threading.
03 | Examples for behavioral descriptions are natural language or semi-formal notation (e.g. SysML, UML).

3 Analyze software architecture. 2
Analyze the software architecture regarding relevant technical design aspects and to support project management regarding project estimates. Document a rationale for the software architectural design decision.
04 | See MAN.3.BP3 for project feasibility and MAN.3.BP5 for project estimates.
05 | The analysis may include the suitability of pre-existing software components for the current application.
06 | Examples of methods suitable for analyzing technical aspects are prototypes, simulations, qualitative analyses.
07 | Examples of technical aspects are functionality, timings, and resource consumption (e.g. ROM, RAM, external / internal EEPROM or Data Flash or CPU load).
08 | 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).

Base practices 4–5 **Process outcomes**

4 Ensure consistency and establish bidirectional traceability. 3
Ensure consistency and establish bidirectional traceability between the software architecture and the software requirements.
09 | 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.
10 | Bidirectional traceability supports consistency, and facilitates impact analysis of change requests, and demonstration of verification coverage. Traceability alone, e.g, the existence of links, does not necessarily mean that the information is consistent with each other.

5 Communicate agreed software architecture. 4
Communicate the agreed software architecture to all affected parties.

SWE.3 Software Detailed Design and Unit Construction

The purpose is to establish a software detailed design, comprising static and dynamic aspects, consistent with the software architecture, and to construct software units consistent with the software detailed design.

Process outcomes

1. A detailed design is specified including static and dynamic aspects.
2. Software units as specified in the software detailed design are produced.
3. Consistency and bidirectional traceability are established between software detailed design and software architecture; and consistency and bidirectional traceability are established between source code and software detailed design; and consistency and bidirectional traceability are established between the software detailed design and the software requirements.
4. The source code and the agreed software detailed design are communicated to all affected parties.

Output information items		Process outcomes
04-05	Software detailed design	1
11-05	Software unit	1, 2
13-51	Consistency evidence	3
13-52	Communication evidence	4

This primary life cycle process belongs to the **basic scope**.

We have published a video tutorial on this process for better understanding.

Base practices 1–3	Process outcomes
1 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 → valid data value ranges for inputs and outputs (from the application domain perspective), and → physical or measurement units applicable to inputs and outputs (from the application domain perspective). 01 | 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. 02 | 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. 03 | Examples of a measurement unit are '%' or '‰'. 04 | A counter is an example of a parameter, or a return value, to which neither a physical nor a measurement unit is applicable. 05 | The hardware-software-interface (HSI) definition puts in context the hardware design and therefore is an aspect of system design (SYS.3).	1
2 Specify 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. 06 | Examples for behavioral descriptions are natural language or semi-formal notation (e.g. SysML, UML).	1
3 Develop software units. Develop and document the software units consistent with the detailed design, and according to coding principles. 07 | 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.	2

Base practices 4–5	Process outcomes
4 **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. 08 | Redundancy should be avoided by establishing a combination of these approaches. 09 | 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. 10 | Bidirectional traceability supports consistency, and facilitates impact analysis of change requests, and demonstration of verification coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent with each other.	3
5 **Communicate agreed software detailed design and developed software units.** Communicate the agreed software detailed design and developed software units to all affected parties.	4

SWE.4 Software Unit Verification

The purpose is to verify that software units are consistent with the software detailed design.

Process outcomes

1 Verification measures for software unit verification are specified.
2 Software unit verification measures are selected according to the release scope, including criteria for regression verification.
3 Software units are verified using the selected verification measures, and results are recorded.
4 Consistency and bidirectional traceability are established between verification measures and software units; and bidirectional traceability are established between verification results and verification measures.
5 Results of the software unit verification are summarized and communicated to all affected parties.

Output information items		Process outcomes
08-60	Verification measure	1
03-50	Verification measure data	3
08-58	Verification measure selection set	2
15-52	Verification results	3
13-51	Consistency evidence	4
13-52	Communication evidence	5

This primary life cycle process belongs to the **basic scope**.

We have published a video tutorial on this process for better understanding.

SWE.4 Software Unit Verification

Base practices 1–5 — **Process outcomes**

1 Specify software unit verification measures. — 1

Specify verification measures for each software unit defined in the software detailed design, including
- → pass/fail criteria for verification measures,
- → entry and exit criteria for verification measures, and
- → the required verification infrastructure.

01 | Examples for unit verification measures are static analysis, code reviews, and unit testing.
02 | Static analysis can be done based on MISRA rulesets and other coding standards.

2 Select software unit verification measures. — 2

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

3 Verify software units. — 3

Perform software unit verification using the selected verification measures. Record the verification results including pass/fail status and corresponding verification measure data.

03 | See SUP.9 for handling of verification results that deviate from expected results.

4 Ensure consistency and establish bidirectional traceability. — 4

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

04 | Bidirectional traceability supports consistency, and facilitates impact analysis of change requests, and demonstration of verification coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent with each other.

5 Summarize and communicate results. — 5

Summarize the results of software unit verification and communicate them to all affected parties.

05 | Providing all necessary information from the test case execution in a summary enables other parties to judge the consequences.

SWE.5 Software Component Verification and Integration Verification

The purpose is to verify that software components are consistent with the software architectural design, and to integrate software elements and verify that the integrated software elements are consistent with the software architecture and software detailed design.

Process outcomes

1 Verification measures are specified for software integration verification of the integrated software elements based on the software architecture and detailed design, including the interfaces of, and interactions between, the software components.
2 Verification measures for software components are specified to provide evidence for compliance of the software components with the software components' behavior and interfaces.
3 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 Software components are verified using the selected verification measures, and the results of the integration verification are recorded.
6 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 verification measures and the software architecture and detailed design; and consistency and bidirectional traceability are established between verification results and verification measures.
8 The results of software component verification and software elements integration verification are summarized and communicated to all affected parties.

Output information items		Process outcomes
08-60	Verification measure	1, 2
06-50	Integration sequence instruction	3
03-50	Verification measure data	5
08-58	Verification measure selection set	4
15-52	Verification results	5, 6
13-51	Consistency evidence	7
13-52	Communication evidence	8
01-03	Software component	3
01-50	Integrated software	3

This primary life cycle process belongs to the **basic scope**.

We have published a video tutorial on this process for better understanding.

Base practices 1–2 — **Process outcomes**

1 Specify software integration verification measures. — 1

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

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

01 | 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.

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

---

2 Specify verification measures for verifying software component behavior. — 2

Specify verification measures for 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.

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

Base practices 3–5 — **Process outcomes**

3 Select verification measures. — 4

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

04 | 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.).

---

4 Integrate software elements and perform integration verification. — 3, 6

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 defined preconditions. Perform the selected integration verification measures. Record the verification measure data including pass/fail status and corresponding verification measure data.

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

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

07 | See SUP.9 for handling deviations of verification results deviate expected results.

---

5 Perform software component verification. — 5

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

08 | See SUP.9 for handling verification results that deviate from expected results.

Base practices 6–7		Process outcomes
6	**Ensure consistency and establish bidirectional traceability.** Ensure consistency and establish bidirectional traceability between verification measures and the static and dynamic aspects of the software architecture and detailed design. Establish bidirectional traceability between verification results and verification measures.	7
09 |	Bidirectional traceability supports consistency, and facilitates impact analysis of change requests, and demonstration of verification coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent with each other.	
7	**Summarize and communicate results.** Summarize the software component verification and the software integration verification results and communicate them to all affected parties.	8
10 |	Providing all necessary information from the test case execution in a summary enables other parties to judge the consequences.	

SWE.6 Software Verification

The purpose of the Software Verification process is to ensure that the integrated software is verified to be consistent with the software requirements.

Process outcomes

1. Verification measures are specified for software 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 software verification are recorded.
4. Consistency and bidirectional traceability are established between verification measures and software requirements; and bidirectional traceability are established between verification results and verification measures.
5. Results of the software verification are summarized and communicated to all affected parties.

Output information items	Process outcomes
08-60 Verification measure	1
03-50 Verification measure data	3
08-58 Verification measure selection set	2
15-52 Verification results	3
13-51 Consistency evidence	4
13-52 Communication evidence	5

This primary life cycle process belongs to the **basic scope**.

We have published a video tutorial on this process for better understanding.

Base practices 1–3 | **Process outcomes**

1 Specify verification measures for software verification. — 1

Specify the verification measures for software verification suitable to provide evidence for compliance of the integrated software with the functional and non-functional information in the software requirements, including

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

01 | 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”.

2 Select verification measures. — 2

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

02 | 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.)

3 Verify the integrated software. — 3

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

03 | See SUP.9 for handling verification results that deviate from expected results.

Base practices 4–5 | **Process outcomes**

4 Ensure consistency and establish bidirectional traceability. — 4

Ensure consistency and establish bidirectional traceability between verification measures and software requirements. Establish bidirectional traceability between verification results and verification measures.

04 | Bidirectional traceability supports consistency, and facilitates impact analysis of change requests, and demonstration of verification coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent with each other.

5 Summarize and communicate results. — 5

Summarize the software verification results and communicate them to all affected parties.

05 | Providing all necessary information from the test case execution in a summary enables other parties to judge the consequences.

VAL.1 Validation

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

1. Validation measures are selected considering criteria for regression verification.
2. The product is validated using the selected validation measures and the results of validation are recorded.
3. Consistency and unidirectional traceability are established between validation measures and stakeholder requirements; and consistency and bidirectional traceability is established between validation results and validation measures.
4. Results of the validation are summarized and communicated to all affected parties.

Output information items		Process outcomes
08-59	Validation measure	**1**
08-57	Validation measure selection set	**2**
13-24	Validation results	**3**
13-51	Consistency evidence	**4, 5**
13-52	Communication evidence	**6**

This primary life cycle process belongs to the **flex scope**.

Base practices 1–2	**Process outcomes**
1 **Specify validation measures for 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 → techniques for the validation measures; → pass/fail criteria for validation measures; → a definition of entry and exit criteria for the validation measures; → necessary sequence of validation measures, and → the required validation infrastructure and environment setup. 01 | 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). 02 | 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. 03 | 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.	1
2 **Select validation measures.** Document the selection of validation measures considering selection criteria including criteria for regression validation. The documented selection of validation measures shall have sufficient coverage according to the release scope. 04 | 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.	1

Base practices 3–5	Process outcomes
3 Perform validation and evaluate 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. 05 | 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. 06 | See SUP.9 for handling verification results that deviate from expected results.	2
4 Ensure consistency and establish bidirectional traceability. Ensure consistency and establish bidirectional traceability from validation measures to the stakeholder requirements from which they are derived. Establish bidirectional traceability between validation results and validation measures. 07 | 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). 08 | 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. 09 | Bidirectional traceability supports consistency, and facilitates impact analyses of change requests, and demonstration of verification coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent with each other.	3
5 Summarize and communicate results. Summarize the validation results and communicate them all to affected parties. 10 | Providing all necessary information from the test case execution in a summary enables other parties to judge the consequences.	4

MLE.1 Machine Learning Requirements Analysis

The purpose is to refine the machine learning-related software requirements into a set of ML requirements.

Process outcomes

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. ML requirements are structured and prioritized.
3. ML requirements are analyzed for correctness and verifiability.
4. The impact of ML requirements on the ML operating environment is analyzed.
5. Consistency and bidirectional traceability are established between ML requirements and software requirements, and between ML requirements and software architecture.
6. The ML requirements are agreed and communicated to all affected parties.

Output information items	Process outcomes
17-00 Requirement	1, 2
17-54 Requirement attribute	2, 3
13-52 Communication evidence	6
13-51 Consistency evidence	5
15-51 Analysis results	3, 4

This primary life cycle process belongs to the **flex scope**.

We have published a video tutorial on the machine learning process model for better understanding.

Base practices 1–4 | **Process outcomes**

1 Specify ML requirements. — 1
Use the software requirements and the software architecture to identify and specify functional and non-functional ML requirements, as well as ML data requirements specifying data characteristics (e.g., gender, weather conditions, street conditions within the ODD) and their expected distributions.

01 | Non-functional requirements may include relevant characteristics of the ODD and KPIs as robustness, performance and level of trustworthiness.

02 | The ML data requirements are input for SUP.11 Machine Learning Data Management but also for other MLE processes.

03 | In case of ML development only, stakeholder requirements represent the software requirements.

2 Structure ML requirements. — 2
Structure and prioritize the ML requirements.

04 | Examples for structuring criteria can be grouping (e.g. by functionality) or variants identification.

05 | Prioritization can be done according to project or stakeholder needs via e.g. definition of release scopes. Refer to SPL.2.BP1.

3 Analyze ML requirements. — 3
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.

06 | See MAN.3.BP3 for project feasibility and MAN.3.BP5 for project estimates.

4 Analyze the impact on the ML operating environment. — 4
Analyze the impact that the ML requirements will have on interfaces of software components and the ML operating environment.

07 | 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.

Base practices 5–6 | **Process outcomes**

5 Ensure consistency and establish bidirectional traceability. — 5
Ensure consistency and establish bidirectional traceability between ML requirements and software requirements and between ML requirements and the software architecture.

08 | 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 with each other.

09 | Redundant traceability is not intended, but at least one out of the given traceability paths.

6 Communicate agreed ML requirements and impact on the operating environment. — 6
Communicate the agreed ML requirements, and the results of the impact analysis on the ML operating environment to all affected parties.

MLE.2 Machine Learning Architecture

The purpose is to establish an ML architecture supporting training and deployment, consistent with the ML requirements, and to evaluate the ML architecture against defined criteria.

Process outcomes

1 A ML architecture is developed.
2 Hyperparameter ranges and initial values are determined as a basis for the training.
3 Evaluation of ML architectural elements is conducted.
4 Interfaces of the ML architectural elements are defined.
5 Resource consumption objectives for the ML architectural elements are defined.
6 Consistency and bidirectional traceability are established between the ML architectural elements and the ML requirements.
7 The ML architecture is agreed and communicated to all affected parties.

Output information items		Process outcomes
04-51	ML architecture	1, 2, 3, 4, 5
13-52	Communication evidence	7
13-51	Consistency evidence	6
01-54	Hyperparameter	1, 2
15-51	Analysis results	1, 3

This primary life cycle process belongs to the **flex scope**.

Base practices 1–7		Process outcomes
1	**Develop ML architecture.** Develop and document the ML architecture that specifies ML architectural elements including details of the ML model, pre- and postprocessing, and hyperparameters which are required to create, train, test, and deploy the ML model. 01 | 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. 02 | The details of the ML model may differ between the ML model used during training and the deployed ML model.	1
2	**Determine hyperparameter ranges and initial values.** Determine and document the hyperparameter ranges and the initial values as a basis for the training.	2
3	**Analyze ML architectural elements.** Define criteria for for analysis of the ML architectural elements. Analyze ML architectural elements according to the defined criteria. 03 | Trustworthiness and explainability might be criteria for the analysis of the ML architectural elements.	3
4	**Define interfaces of the ML architectural elements.** Determine and document the internal and external interfaces of each ML architectural element including its interfaces to related software components.	4
5	**Define resource consumption objectives for the ML architectural elements.** Determine and document the resource consumption objectives for all relevant ML architectural elements during training and deployment.	5
6	**Ensure consistency and establish bidirectional traceability.** Ensure consistency and establish bidirectional traceability between the ML architectural elements and the ML requirements. 04 | 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 with each other. 05 | The bidirectional traceability should be established on a reasonable level of abstraction to the ML architectural elements.	6
7	**Communicate agreed ML architecture.** Inform all affected parties about the agreed ML architecture including the details of the ML model and the initial hyperparameter values.	7

MLE.3 Machine Learning Training

The purpose is to optimize the ML model to meet the defined ML requirements.

Process outcomes

1 A ML training and validation approach is specified.
2 The data set for ML training and ML validation is created.
3 The ML model, including 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 Results of optimization are summarized, and the trained ML model is agreed and communicated to all affected parties.

Output information items	Process outcomes
08-65 ML training and validation approach	1
03-51 ML data set	2
01-53 Trained ML model	3
01-54 Hyperparameter	3
13-51 Consistency evidence	4
13-52 Communication evidence	5

This primary life cycle process belongs to the **flex scope**.

Base practices 1–3	Process outcomes

1 Specify ML training and validation approach. — 1

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

→ entry and exit criteria of the training and validation
→ approaches for hyperparameter tuning / optimization
→ approach for data set creation and modification, and
→ training and validation environment

01 | The ML training and validation approach may include random dropout and other robustification methods.
02 | ML validation is the optimization of the hyperparameters during Machine Learning Training (MLE.3). The term “validation” has a different meaning than VAL.1.
03 | The training environment should reflect the environment of the deployed model.

2 Create ML training and validation data set. — 2

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.

04 | The ML training and validation data set may include corner cases, unexpected cases, and normal cases depending on the ML requirements.
05 | 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).

3 Create and optimize ML model. — 3

Create the ML model according to the ML architecture and train it, using the identified ML training and validation data set according to the ML training and validation approach to meet the defined ML requirements, and training and validation exit criteria.

Base practices 4–5	Process outcomes
4 **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. 06 | 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 with each other.	4
5 **Summarize and communicate agreed trained ML model.** Summarize the results of the optimization and inform all affected parties about the agreed trained ML model.	5

MLE.4 Machine Learning Model Testing

The purpose is to ensure compliance of the trained ML model and the deployed ML model with the ML requirements.

Process outcomes

1. A ML test approach is defined.
2. A 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 ML test results.
6. Results of the ML model testing are summarized and communicated with the deployed ML model to all affected parties.

Output information items		Process outcomes
08-64	ML test approach	1
03-51	ML data set	2
13-50	ML test result	3, 4
11-50	Deployed ML model	4
13-51	Consistency evidence	5
13-52	Communication evidence	6

This primary life cycle process belongs to the **flex scope**.

Base practices 1–3 | **Process outcomes**

1 Specify an ML test approach. 1

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
→ distribution and frequency of each ML test scenario inside the ML test data set
→ expected test result per test datum
→ entry and exit criteria of the testing
→ approach for data set creation and modification, and
→ the required testing infrastructure and environment setup

01 | Expected test result per test datum might require labeling of test data to support comparison of output of the ML model with the expected output.

02 | Test datum is the smallest amount of 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.

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

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

2 Create ML test data set. 2

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.

05 | The ML test data set for the trained ML model might differ from the test data set of the deployed ML model.

06 | Additional data sets might be used for special purposes like assurance of safety, fairness, robustness.

Base practices 4–7 | **Process outcomes**

3 Test trained ML model. 3

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

07 | Evaluation of test logs might include pattern analysis of failed test data to support e.g., trustworthiness.

4 Derive deployed ML model. 4

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 integration8

08 | 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.

5 Test deployed ML model. 4

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.

6 Ensure consistency and establish bidirectional traceability. 5

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 ML test results.

09 | 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 with each other.

7 Summarize and communicate results. 6

Summarize the test results of the ML model. Inform all affected parties about the agreed results and the deployed ML model.

HWE.1 Hardware Requirements Analysis

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

1 Hardware requirements are specified.
2 Hardware requirements are structured and prioritized.
3 Hardware requirements are analyzed for correctness and technical feasibility.
4 The impact of hardware requirements on the operating environment is analyzed.
5 Consistency and bidirectional traceability are established between hardware requirements and system requirements.
6 Consistency and bidirectional traceability are established between hardware requirements and system architectural design.
7 The hardware requirements are agreed and communicated to all affected parties.

Output information items		Process outcomes
13-52	Communication evidence	7
13-51	Consistency evidence	5, 6
17-00	Hardware requirement	1, 2, 3
17-54	Requirement attribute	2
15-51	Analysis results	3, 4

This primary life cycle process belongs to the **flex scope**.

Base practices 1–3 **Process outcomes**

1 Specify hardware requirements. 1
Use the system requirements, and the system architecture including interface definitions, to identify and document the functional and non-functional requirements of the hardware according to defined characteristics for requirements.
01 | 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.
02 | 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.
03 | 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.
04 | The hardware-software-interface (HSI) definition puts in context software and therefore is an interface decision at the system design level (see SYS.3). If such a HSI exists, then it may provide input to hardware requirements.

2 Structure hardware requirements. 2
Structure and prioritize the hardware requirements.
05 | Examples for structuring criteria can be grouping (e.g. by functionality) or variants identification.
06 | Prioritization can be done according to project or stakeholder needs via e.g. definition of release scopes. Refer to SPL.2.BP1.

3 Analyze hardware requirements. 3
Analyze the specified hardware requirements including their interdependencies to ensure correctness, technical feasibility, and to support project management regarding project estimates.
07 | See MAN.3.BP3 for project feasibility and MAN.3.BP5 for project estimates.
08 | The analysis of technical feasibility can be done based on a given hardware design (e.g., platform) or by prototype development.

Base practices 4–6	Process outcomes
4 **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.	4
5 **Ensure consistency and establish bidirectional traceability.** Ensure consistency and establish traceability between hardware requirements and the system architecture. Ensure consistency and establish traceability between hardware requirements and system requirements. 09 | Redundant traceability is not intended. 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. 11 | Bidirectional traceability supports consistency, and facilitates impact analysis of change requests, and demonstration of verification coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent with each other. 12 | In case of hardware 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 hardware requirements.	5, 6
6 **Communicate agreed hardware requirements and impact on the operating environment.** Communicate the agreed hardware requirements and results of the analysis of impact on the operating environment to all relevant parties.	7

HWE.2 Hardware Design

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

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 special characteristics are identified.
3. Consistency and bidirectional traceability are established between hardware requirements and hardware design.
4. Hardware production data is derived from the HW detailed design and communicated to all affected parties.
5. Information for production test is derived from the HW detailed design and communicated to all affected parties.
6. The hardware architecture and hardware detailed design and the special characteristics are agreed and communicated to all affected parties.

Output information items	Process outcomes
04-52 Hardware architecture	1
04-53 Hardware detailed design	1
15-51 Analysis results	2
13-51 Consistency evidence	3
17-57 Special characteristics	2
13-52 Communication evidence	6
04-54 Hardware schematics	1, 4, 5
14-54 Hardware bill of materials	1, 4, 5
04-55 Hardware layout	1, 4, 5
03-54 Hardware production data	1, 4, 5
04-56 Hardware element interface	1

This primary life cycle process belongs to the **flex scope**.

Base practices 1–3 — **Process outcomes**

1 Specify the hardware architecture. — **1, 4, 5**
Develop the hardware architecture that identifies the hardware components. Document the rationale for the defined hardware architecture.

01 | Examples for aspects reflected in the hardware architecture are ground concept, supply concept, EMC concept.

02 | 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.

2 Specify the hardware detailed design. — **1, 4, 5**
Based on components identified in the hardware architecture, develop 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.

03 | The identification of 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.).

04 | 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.

3 Specify dynamic aspects. — **1**
Evaluate and document the dynamic behaviour of the relevant hardware elements and the interaction between them.

05 | Not all hardware elements have dynamic behaviour that needs to be described.

Base practices 4–6 — **Process outcomes**

4 Analyze the hardware architecture and the hardware detailed design. — **2**
Analyze the hardware architecture and hardware detailed design regarding relevant technical aspects, and support project management regarding project estimates. Identify special characteristics.

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

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

5 Ensure consistency and establish bidirectional traceability. — **3**
Ensure consistency and establish traceability between hardware elements and hardware requirements. Ensure consistency and establish traceability between the hardware detailed design and components of the hardware architecture.

08 | 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.

09 | Bidirectional traceability further supports consistency, and facilitates impact analysis of change requests, and demonstration of verification coverage. Traceability alone, e.g, the existence of links, does not necessarily mean that the information is consistent with each other.

6 Communicate agreed hardware architecture and hardware detailed design. — **4, 5, 6**
Communicate the agreed hardware architecture and the hardware detailed design, including the special characteristics and relevant production data, to all affected parties.

HWE.3 Verification against Hardware Design

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

Output information items	Process outcomes
08-60 Verification measure	**1**
03-50 Verification measure data	**3**
08-58 Verification measure selection set	**2**
15-52 Verification results	**3**
13-51 Consistency evidence	**4, 5**
13-52 Communication evidence	**6**

This primary life cycle process belongs to the **flex scope**.

Base practices 1–4	Process outcomes
1 **Specify verification measures for the verification against 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 verification measures → a definition of entry and exit criteria for the verification measures → necessary sequence of verification measures → the required verification infrastructure and environment setup 01 | Examples on what a verification measure may focus on are the timeliness and timing dependencies of the correct signal flow between interfacing hardware elements, interactions between hardware components. 02 | Measuring points can be used for stepwise testing of hardware elements.	1
2 **Ensure use of compliant samples.** Ensure that the samples used for verification against hardware design are compliant with the corresponding production data, including special characteristics. Ensure that deviations are documented and that they do not alter verification results. 03 | Examples of compliance are sample reports, record of visual inspection, ICT report.	3
3 **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. 04 | 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.)	2
4 **Verify hardware design.** Verify the hardware design using the selected verification measures. Record the verification results including pass/fail status and corresponding verification measure output data. 05 | See SUP.9 for handling of non-conformances.	3

Base practices 5–6	Process outcomes
5 **Ensure consistency and establish bidirectional traceability.** Ensure consistency and establish bidirectional traceability between hardware elements and the verification measures. Establish bidirectional traceability between the verification measures and verification results. 06 | Bidirectional traceability supports consistency, and facilitates impact analysis of change requests, and demonstration of verification coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent with each other.	4, 5
6 **Summarize and communicate results.** Summarise the verification results and communicate them to all affected parties. 07 | Providing all necessary information from the test case execution in a summary enables other parties to judge the consequences.	6

HWE.4 Verification against Hardware Requirements

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

Output information items	Process outcomes
08-60 Verification measure	1
03-50 Verification measure data	3
08-58 Verification measure selection set	2
15-52 Verification results	3
13-51 Consistency evidence	4, 5
13-52 Communication evidence	6

This primary life cycle process belongs to the **flex scope**.

Base practices 1–4	Process outcomes
1 Specify verification measures for the verification against 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 verification measures → a definition of entry and exit criteria for the verification measures → necessary sequence of verification measures → the required verification infrastructure and environment setup 01 | The verification measures may cover aspects such as thermal, environmental, robustness/lifetime, and EMC.	1
2 Ensure use of compliant samples. Ensure that the samples used for the verification against hardware requirements are compliant with the corresponding production data, including special characteristics, provided by hardware design. 02 | Examples of compliance are sample reports, record of visual inspection, ICT report.	3
3 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. 03 | 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.).	2
4 Verify the compliant hardware samples. Verify the compliant hardware samples using the selected verification measures. Record the verification results including pass/fail status and corresponding verification measure output data. 04 | See SUP.9 for handling of non-conformances.	3

Base practices 5–6	Process outcomes
5 Ensure consistency and establish bidirectional traceability. Ensure consistency between hardware requirements and verification measures. Establish bidirectional traceability between hardware requirements and verification measures. Establish bidirectional traceability between verification measures and verification results. 05 | Bidirectional traceability supports consistency, and facilitates impact analysis of change requests, and demonstration of verification coverage. Traceability alone, e.g., the existence of links, does not necessarily mean that the information is consistent with each other.	4, 5
6 Summarize and communicate results. Summarize the verification results and communicate them to all affected parties. 06 | Providing all necessary information from the test case execution in a summary enables other parties to judge the consequences.	6

SUP.1 Quality Assurance

The purpose of the Quality Assurance process is to provide independent and objective assurance that work products and processes comply with defined criteria and that non-conformances are resolved and further prevented.

Process outcomes

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 the relevant parties.
4. Non-conformances are tracked, resolved, and further prevented.
5. Non-conformances are escalated to appropriate levels of management.
6. Management ensures that escalated non-conformances are resolved.

Output information items		Process outcomes
16-50	Organization structure	1, 5
18-52	Escalation path	5, 6
18-07	Quality criteria	2, 3, 4
13-52	Communication evidence	3, 4, 5
13-18	Quality conformance evidence	3, 4
13-19	Review evidence	3, 4
14-02	Corrective action	4, 6

This primary life cycle process belongs to the **basic scope**.

We have published a video tutorial on this process for better understanding.

Base practices 1–5		Process outcomes
1	**Ensure independence of quality assurance.** Ensure that quality assurance is performed independently and objectively without conflicts of interest. 01 | 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).	1
2	**Define criteria for quality assurance.** Define quality criteria for work products as well as for process tasks and their performance. 02 | Quality criteria may consider internal and external inputs such as customer requirements, standards, milestones, etc.	2
3	**Assure quality of work products.** Identify 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. 03 | Quality assurance activities may include reviews, problem analysis and lessons learned that improve the work products for further use.	3, 4
4	**Assure quality of process activities.** Identify 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. 04 | 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.	3, 4
5	**Summarize and communicate quality assurance activities and results.** Regularly report performance, non-conformances, and trends of quality assurance activities to all affected parties.	3, 4, 5

Base practices 6–7	Process outcomes
6 **Ensure resolution of non-conformances.** Analyze, track, correct, resolve and further prevent non-conformances found in quality assurance activities. 05 | Non-conformances detected in work products may be entered into the problem resolution management process (SUP.9). 06 | Non-conformances detected in the process definition or implementation may be entered into a process improvement process (PIM.3).	4, 6
7 **Escalate non-conformances.** Escalate relevant non-conformances to appropriate levels of management and other relevant stakeholders to facilitate their resolution. 07 | The decision whether to escalate non-conformances may be based on criteria such as delay of resolution, urgency, and risk.	5, 6

SUP.8 Configuration Management

The purpose of the Configuration Management process is to establish and maintain the integrity of relevant configuration items and baselines, and make them available to affected parties.

Process outcomes

1. Selection criteria for configuration items are defined and applied.
2. Configuration item properties are defined.
3. Configuration management is established.
4. 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.

Output information items	Process outcomes
18-53 Configuration item selection criteria	1
01-52 Configuration item list	1, 2, 7
16-03 Configuration management system	3, 4, 5
13-08 Baseline	5, 7
14-01 Change history	3, 4, 6
15-56 Configuration status	6
13-51 Consistency Evidence	7
06-52 Backup and recovery mechanism information	8

This primary life cycle process belongs to the **basic scope**.

We have published a video tutorial on this process for better understanding.

Base practices 1–4		Process outcomes
1	**Identify configuration items.** Define selection criteria for identifying relevant work products to be subject to configuration management. Identify and document configuration items according to the defined selection criteria. 01 | Configuration items are representing work products or group of work products which are subject to configuration management as a single entity. 02 | 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. 03 | The selection criteria may be applied to single work products or a group of work products.	1
2	**Define configuration item properties.** Define the necessary properties needed for the modification and control of configuration items. 04 | The configuration item properties may be defined for single configuration items or a group of items. 05 | Configuration item properties may include a status model (e.g. under work, checked in, tested, released, etc.), storage location, access rights, etc. 06 | The application of properties may be implemented by attributes of the configuration items.	2
3	**Establish configuration management.** Establish configuration management mechanisms for control of identified configuration items including the configuration item properties, including mechanisms to control parallel modifications of configuration items. 07 | This may include specific mechanisms for different configuration item types, such as branch and merge management or checkout control.	3, 4
4	**Control modifications.** Control modifications using the configuration management mechanisms. 08 | This may include the application of a defined status model for configuration items.	4

Base practices 5–8		Process outcomes
5	**Establish baselines.** Define and establish baselines for internal purposes and for external product delivery for all relevant configuration items.	5
6	**Summarize and communicate configuration status.** Record, summarize and communicate the status of configuration items and established baselines to affected parties in order to support the monitoring of progress and status. 09 | Regular communication of the configuration status, e.g. based on a defined status model supports project management, quality activities and dedicated project phases like software integration.	6
7	**Ensure completeness and consistency.** Ensure that the information about configuration items is correct and complete including configuration item properties. Ensure the completeness and consistency of baselines. 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.	7
8	**Verify backup and recovery mechanisms' availability.** Verify the availability of appropriate backup and recovery mechanism for the configuration management including the controlled configuration items. Initiate measures in case of insufficient backup and recovery mechanisms. 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.	8

SUP.9 Problem Resolution Management

The purpose of the Problem Resolution Management process is to ensure that problems are identified, recorded, analyzed, and their resolution is managed and controlled.

Process outcomes

1 Problems are uniquely identified, recorded and classified.
2 Problems are analyzed and assessed to determine an appropriate solution.
3 Problem resolution is initiated.
4 Problems are tracked to closure.
5 The status of problems including trends identified are reported to stakeholders.

Output information items		Process outcomes
13-07	Problem	1, 2, 3, 4
15-55	Problem analysis evidence	2
15-12	Problem status	5

This primary life cycle process belongs to the **basic scope**.

We have published a video tutorial on this process for better understanding.

Base practices 1–5		Process outcomes
1	**Identify and record the problem.** 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 problem. 01 | Problems may relate to e.g., product, resources, or methods. 02 | Example values for the problem status are “new”, “solved”, “closed”, etc. 03 | Supporting information may include, e.g. the origin of the problem, how it can be reproduced, environmental information, by whom it has been detected. 04 | Unique identification supports traceability to changes made as needed by the change request management process (SUP.10).	1, 4
2	**Determine the cause and the impact of the problem.** Analyze the problem, determine its cause, including common causes if existing, and impact. Involve relevant parties. Categorize the problem. 05 | Problem categorization (e.g., light, medium, severe) may be based on severity, criticality, urgency, etc.	1, 2
3	**Authorize urgent resolution action.** Obtain authorization for immediate action if a problem requires an urgent resolution according to the categorization.	3
4	**Raise alert notifications.** If according to the categorization the problem has a high impact on other systems or other affected parties, an alert notification needs to be raised accordingly.	3
5	**Initiate problem resolution.** Initiate appropriate actions according to the categorization to resolve the problem long-term, including review of those actions or initiate a change request. This includes synchronization and consistency with short-term urgent resolution actions, if applicable.	3

Base practices 6–7		Process outcomes
6	**Track problems to closure.** Track the status of problems to closure including all related change requests. The closure of problems is accepted by relevant stakeholders.	4, 5
7	**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 relevant stakeholders. 06 | Collected data may contain information about where the problems occurred, how and when they were found, what were their impacts, etc.	5

SUP.10 Change Request Management

The purpose of the Change Request Management process is to ensure that change requests are recorded, analyzed, tracked, approved, and implemented.

Process outcomes

1 Requests for changes are recorded and identified.
2 Change requests are analyzed, dependencies and relationships to other change requests are identified, and the impact is estimated.
3 Change requests are approved before implementation and prioritized accordingly.
4 Bidirectional traceability is established between change requests and affected work products.
5 Implementation of change requests is confirmed.
6 Change requests are tracked to closure and status of change requests is communicated to affected parties.

Output information items		Process outcomes
18-57	Change analysis criteria	2
13-16	Change request	1, 2, 3, 5, 6
13-51	Consistency evidence	4

This primary life cycle process belongs to the **basic scope**.

We have published a video tutorial on this process for better understanding.

Base practices 1–6		Process outcomes
1	**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.	1
01 |	Change requests may be used for changes related to e.g., product, process, methods.	
02 |	Example values for the change request status are "open", "under investigation", "implemented", etc.	
03 |	The change request handling may differ across the product life cycle e.g., during prototype construction and series development	
2	**Analyze and assess change requests.** Change requests are analyzed by relevant parties according to analysis criteria. Work products affected by the change request and dependencies to other change requests are determined. The impact of the change requests is assessed.	2
04 |	Examples for analysis criteria are: resource requirements, scheduling issues, risks, benefits, etc.	
3	**Approve change requests before implementation.** Change requests are prioritized and approved for implementation based on analysis results and availability of resources.	3
05 |	A Change Control Board (CCB) is an example mechanism used to approve change requests.	
06 |	Prioritization of change requests may be done by allocation to releases.	
4	**Establish bidirectional traceability.** Establish bidirectional traceability between change requests and work products affected by the change requests. In case that the change request is initiated by a problem, establish bidirectional traceability between change requests and the corresponding problem reports.	4
5	**Confirm the implementation of change requests.** The implementation of change requests is confirmed before closure by relevant stakeholders.	5
6	**Track change requests to closure.** Change requests are tracked to closure. The status of change requests is communicated to all affected parties.	6
07 |	Examples for informing affected parties can be daily standup meetings or tool-supported workflows.	

SUP.11 Machine Learning Data Management

The purpose is to define and align ML data with ML data requirements, maintain the integrity and quality of the ML data, and make them available to affected parties.

Process outcomes

1 A ML data management system including a ML data lifecycle is established.
2 A ML data quality approach is developed including ML data quality criteria.
3 Collected ML data are processed for consistency with ML data requirements.
4 ML data are verified against defined ML data quality criteria and updated as needed.
5 ML data are agreed and communicated to all affected parties.

Output information items	Process outcomes
16-52 ML data management system	1
19-50 ML data quality approach	2
03-53 ML data	3, 4
13-52 Communication evidence	5

This primary life cycle process belongs to the **flex scope**.

Base practices 1–3		Process outcomes
1	**Establish an ML data management system.** Establish an ML data management system which supports → ML data management activities; → relevant sources of ML data; → ML data life cycle including a status model; → interfaces to affected parties.	1
01 |	Supported ML data management activities may include data collection, labeling/annotation, and structuring.	
2	**Develop an ML data quality approach.** Develop an approach to ensure that the quality of ML data is analyzed based on defined ML data quality criteria and activities are performed to support avoidance of biases of data.	2
02 |	Examples of ML data quality criteria are relevant data sources, reliability and consistency of labelling, completeness against ML data requirements.	
03 |	The ML data management system should support the quality criteria and activities of the ML data quality approach.	
04 |	Biases to avoid may include sampling bias (e.g., gender, age) and feedback loop bias.	
05 |	For creation of ML data sets see MLE.3.BP2 and MLE.4.BP2.	
3	**Collect ML data.** Relevant sources for raw data are identified and continuously monitored for changes. The raw data is collected according to the ML data requirements.	3
06 |	The identification and collection of ML data might be an organizational responsibility.	
07 |	Continuous monitoring should include the ODD and may lead to changes of the ML requirements.	

Base practices 4–6		Process outcomes
4	**Process ML data.** The raw data are processed (annotated, analyzed, and structured) according to the ML data requirements.	3
5	**Assure quality of ML data.** Perform the activities according to the ML data quality approach to ensure that the ML data meets the defined ML data quality criteria.	4
08 |	These activities may include sample-based reviews or statistical methods.	
6	**Communicate agreed processed ML data.** Inform all affected parties about the agreed processed ML data and provide them to the affected parties.	5

MAN.3 Project Management

The purpose is to identify and control the activities, and establish resources necessary for a project to develop a product, in the context of the project's requirements and constraints.

Process outcomes

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.

Output information items		Process outcomes
08-53	Scope of work	1
08-54	Feasibility analysis	2, 4
14-10	Work package	3, 4, 5
13-52	Communication evidence	2, 3
13-16	Change request	7
13-51	Consistency evidence	2, 7
14-02	Corrective action	6, 7
18-52	Escalation path	4, 6, 7
08-56	Schedule	3, 5, 7
14-50	Stakeholder groups list	4
15-06	Project status	4, 6

We have published a video tutorial on this process for better understanding.

This primary life cycle process belongs to the **basic scope**.

Base practices 1–6		Process outcomes
1	**Define the scope of work.** Identify the project's goals, motivation and boundaries.	1
2	**Define project life cycle.** Define the life cycle for the project, which is appropriate to the scope, context, and complexity of the project. Define a release scope for relevant milestones.	1, 2
01 |	This may include the alignment of the project life cycle with the customer's development process.	
3	**Evaluate feasibility of the project.** Evaluate the feasibility of achieving the goals of the project with respect to time, project estimates, and available resources.	2
02 |	The evaluation of feasibility may consider technical constraints of the project.	
4	**Define and monitor work packages.** Define and monitor work packages and their dependencies according to defined project life cycle and estimations.	3, 4, 5, 7
03 |	The structure and the size of the work packages support an adequate progress monitoring.	
04 |	Work packages may be organized in a work breakdown structure.	
5	**Define and monitor project estimates and resources.** Define and monitor project estimates of effort and resources based on project's goals, project risks, motivation and boundaries.	2, 3, 7
05 |	Examples of necessary resources are budget, people or infrastructure.	
06 |	Project risks (using MAN.5) may be considered.	
07 |	Estimations and resources may include engineering, management and supporting processes.	
6	**Define and monitor 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.	3, 7
08 |	Training, mentoring or coaching of individuals may be applied to resolve deviations from required skills and knowledge.	

Base practices 7–10		Process outcomes
7	**Define and monitor project interfaces and agreed commitments.** Identify and agree interfaces of the project with affected stakeholders and monitor agreed commitments. Define an escalation mechanism for commitments that are not fulfilled.	3, 5, 7
09 |	Affected stakeholders may include other projects, organizational units, sub-contractors, and service providers.	
8	**Define and monitor project schedule.** Allocate resources to work packages and schedule each activity of the project. Monitor the performance of activities against schedule.	6, 7
9	**Ensure consistency.** Regularly adjust estimates, resources, skills, work packages and their dependencies, schedules, plans, interfaces, and commitments for the project to ensure consistency with the scope of work.	3, 4, 5, 7
10 |	This may include the consideration of critical dependencies, that are an input for risk management.	
10	**Review and report progress of the project.** Regularly review and report the status of the project and the fulfillment of work packages against estimated effort and duration to all affected parties. Prevent recurrence of identified problems.	6, 7
11 |	Project reviews may be executed at regular intervals by the management. Project reviews may contribute to identify best practices and lessons learned.	
12 |	Refer to SUP.9 for resolution of problems	

MAN.5 Risk Management

The purpose is to regularly identify, analyze, treat and monitor process related risks and product related risks.

Process outcomes

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. Risks are analyzed and the priority in which to apply resources to treatment of these risks is determined.
4. Risk measures are defined, applied, and assessed to determine changes in the status of risk 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.

Output information items		Process outcomes
15-51	Analysis results	1-3, 5
08-55	Risk measure	4, 5
14-02	Corrective action	4, 5
15-09	Risk status	1, 3, 4, 5

This primary life cycle process belongs to the **flex scope**.

Base practices 1–7		Process outcomes
1	**Identify sources of risks.** Identify and regularly update the sources of risks with affected parties.	1
01 |	Risks may include technical, economical, and schedule risks.	
02 |	Risks may include the suppliers' deliverables and services.	
03 |	The risk sources may vary across the entire project life cycle.	
2	**Identify potential undesirable events.** Identify potential undesirable events within the scope of the risk management for the project.	2
3	**Determine risks.** Determine the probability and severity of the undesirable events to support priorities for mitigating the risks.	3
04 |	Different methods may be used to analyze technical risks of a system, for example, functional analysis, simulation, FMEA, FTA etc.	
4	**Define risk treatment options.** For each risk select a treatment option to accept, mitigate, avoid or share (transfer) the risk.	4, 5
5	**Define and perform risk treatment activities.** Define and perform risk activities for risk treatment options.	4, 5
6	**Monitor risks.** Regularly re-evaluate the risk related to the identified potential undesirable events to determine changes in the status of a risk and to evaluate the progress of the risk treatment activities.	4
05 |	Risks of high priority may need to be communicated to and monitored by higher levels of management.	
7	**Take corrective action.** When risk treatment activities are not effective, take appropriate corrective action.	5
06 |	Corrective actions may involve reevaluation of risks, developing and implementing new mitigation concepts or adjusting the existing concepts.	

MAN.6 Measurement

The purpose is to collect and analyze data relating to the development results and processes implemented within the organization and its projects, to support effective management of the processes.

Process outcomes

1 The measurement information needs that are necessary to evaluate the achievement of process objectives and the achievement of desired documented information 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.

Output information items		Process outcomes
03-03	Benchmarking data	4, 5
03-04	Customer satisfaction data	4, 5
03-06	Process performance information	4, 5
07-51	Measurement result	2, 3, 4, 5
15-51	Analysis results	1,4, 5

This primary life cycle process belongs to the **flex scope**.

MAN.6 Measurement

Base practices 1–6		Process outcomes
1	**Identify information needs.** Identify the measurement information needs that are necessary to evaluate the achievement of process objectives and work products. 01 | Information needs may change over time. Therefore, the measurement process may be used in an iterative way.	1
2	**Specify metrics.** Identify and develop an appropriate set of metrics based on measurement information needs. 02 | Metrics may be related to processes or development results.	2, 3
3	**Collect and store metrics.** Collect and store both base and derived metrics, including any context information necessary to verify and understand the metrics. 03 | 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”.	3, 4
4	**Analyze collected metrics.** Analyze, interpret and review measured values to support decision-making.	4, 5
5	**Communicate analysis results.** Communicate analysis results to all affected parties.	5
6	**Use metrics for decision-making.** Make accessible and use information from collected metrics and analysis results for any decision-making process for which it is relevant.	5

PIM.3 Process Improvement

The purpose is to continually improve the organization’s effectiveness and efficiency through the processes used and ensure alignment of the processes with the business needs.

Process outcomes

1 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 Analysis of the current status of the existing process is performed.
4 Improvement goals are identified and prioritized, and consequent changes to the process are defined, documented and implemented.
5 The effects of process implementation are monitored, measured and confirmed against the identified improvement goals.
6 Knowledge gained from the improvement is communicated within the organization.

Output information items		Process outcomes
16-50	Organizational structure	1, 5
18-52	Escalation path	5, 6
18-07	Quality criteria	2, 3, 4
13-52	Communication evidence	3, 4, 5
13-18	Quality conformance evidence	3, 4
13-19	Review evidence	3, 4
14-02	Corrective action	4, 6

This primary life cycle process belongs to the **flex scope**.

Base practices 1–5	Process outcomes
1 Establish commitment. Establish commitment to support the process improvement staff, to provide resources and further enablers to sustain improvement actions. 01 | 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. 02 | Commitment at all levels of management may support process improvement. 03 | Enablers for improvement measures may include trainings, methods, infrastructure, etc.	1
2 Identify improvement measures. Identify issues from the analysis of process performance and derive improvement opportunities with justified reasons for change. 04 | Analysis may include problem report trend analysis (see SUP.9), analysis from Quality Assurance and Verification results and records (see SUP.1), validation results and records, and product quality metrics like defect rate. 05 | Issues and improvement suggestions may be addressed by the customer. 06 | Sources for identification of issues may include: process assessment results, audits, customer's satisfaction reports, measurements of organizational effectiveness/efficiency, costs of quality.	2, 3
3 Establish process improvement goals. Analyze the current status of the existing processes and establish improvement goals. 07 | The current status of processes may be determined by process assessment.	4
4 Prioritize improvements. Prioritize the improvement goals and improvement measures.	4
5 Define process improvement measures. Process improvement measures are defined. 08 | Improvements may be documented in incremental steps.	4

Base practices 6–8	Process outcomes
6 Implement process improvement measures. Implement and apply the improvements to the processes. Update the Process documentation and train people as needed. 09 | Process application can be supported by establishing policies, adequate process infrastructure, process training, process coaching and tailoring processes to local needs. 10 | Improvements may be piloted before roll out within the organization.	4
7 Confirm process improvement. The effects of process implementation are monitored and measured, and the achievement of defined improvement goals is confirmed.	5
8 Communicate results of improvement. Knowledge gained from the improvements and progress of the improvement implementation is communicated to affected parties.	6

REU.2 Management of Products for Reuse

The purpose is to ensure that reused work products are analyzed, verified, and approved for their target context.

Process outcomes

1 Products for reuse are selected using defined criteria.
2 Products for reuse are analyzed for portability and interoperability.
3 Limitations for reuse are defined and communicated.
4 Products for reuse are verified.
5 Products for reuse are provided to affected parties.
6 Communication mechanism is established with the reuse product provider.

Output information items		Process outcomes
04-02	Domain architecture	2, 3
12-03	Reuse candidate	1, 5
13-52	Communication evidence	6
15-07	Reuse analysis evidence	2, 3
13-53	Qualification evidence	4

This primary life cycle process belongs to the flex scope.

Base practices 1–6		Process outcomes
1	**Select products for reuse.** Select the products to be reused using defined criteria.	1
01 |	Products for reuse may be systems, hardware or software components, third party components or legacy components.	
2	**Analyze the reuse capability of the product.** Analyze the designated target architecture and the product to be reused to determine its applicability in the target architecture according to relevant criteria.	2
02 |	Examples for criteria can be requirements compliance, verifiability of the product to be reused in the target architecture, or portability/interoperability.	
3	**Define limitations for reuse.** Define and communicate limitations for the products to be reused.	3
03 |	Limitations may address parameters of operational environment.	
4	**Ensure qualification of products for reuse.** Provide evidence that the product for reuse is qualified for the intended use of the deliverable.	4
04 |	Qualification may be demonstrated by verification evidence.	
05 |	Verification may include the appropriateness of documentation.	
5	**Provide products for reuse.** Make available the product to be reused to affected parties.	5
06 |	Refer to HWE.3, SWE.5 or SYS.4 for more information on integration of hardware, software, or system components.	
6	**Communicate information about effectiveness of reuse activities.** Establish communication and notification mechanism about experiences and technical outcomes to the provider of reused products.	6
07 |	The communication with the provider of a reused product may depend on whether the product is under development or not.	

Part 2

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.

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

Process capability level	Process attribute ID	Each process attribute is identified with a unique identifier and name. A process attribute scope statement is provided, and process achievements are defined.
	Process attribute name	
	Process attribute scope	
	Process achievements	

Process attribute achievement indicators	Generic practices	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. The generic practice headers are summarized at the end of a process to demonstrate their relationship to the process attribute achievements.
	Output information items	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. 02 | Refer to Annex B for the characteristics of each information item.

Template for the process description

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.

Process capability Level 1 Performed process

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

PA 1.1 Process performance process attribute

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.

Information items	Achievement
Process specific information items, as described in chapter 4	**a**

Generic practices 1	Achievement
1 **Achieve the process outcomes** Achieve the intent of the base practices. Produce work products that evidence the process outcomes.	a

Process capability Level 2 Managed process

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

PA 2.1 Process performance management process attribute

The performance management process attribute is a measure of the extent to which the performance of the process is managed.

Process attribute achievements

1. Strategy for the performance of the process is defined based on identified objectives.
2. Performance of the process is planned.
3. Performance of the process is monitored and adjusted to meet the planning.
4. Needs for human resources including responsibilities and authorities for performing the process are determined.
5. Needs for physical and material resources are determined.
6. Persons performing the process are prepared for executing their responsibilities.
7. Physical and material resources for performing the process are identified, made available, allocated and used.
8. Interfaces between the involved parties are managed to ensure both effective communication and the assignment of responsibilities.

Output information items		**Achievement**
19-01	Process performance strategy	**1**
18-58	Process performance objectives	**1**
14-10	Work package	**2**
08-56	Schedule	**2, 3**
13-14	Progress status	**3**
17-55	Resource needs	**4, 5**
08-61	Resource allocation	**6, 7**
08-62	Communication matrix	**8**
13-52	Communication evidence	**8**

Generic practices 1–2 — **Achievement**

1 **Identify the objectives and define a strategy for the performance of the process.** — Achievement: 1

The scope of the process activities including the management of process performance and the management of work products are determined.
Corresponding results to be achieved are determined.
Process performance objectives and associated criteria are identified.

01 | Budget targets and delivery dates to the customer, targets for test coverage and process lead time are examples for process performance objectives.

02 | Performance objectives are the basis for planning and monitoring.

Assumptions and constraints are considered when identifying the performance objectives. Approach and methodology for the process performance is determined.

03 | A process performance strategy may not necessarily be document-ed specifically for each process. Elements applicable for multiple processes may be documented jointly, e.g. as part of a common project handbook or in a joint test strategy.

2 **Plan the performance of the process.** — Achievement: 2

The planning for the performance of the process is established according to the defined objectives, criteria, and strategy.
Process activities and work packages are defined.
Estimates for work packages are identified using appropriate methods.

04 | Schedule and milestones are defined.

Generic practices 3–6 — **Achievement**

3 **Determine resource needs.** — Achievement: 4, 5

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

05 | Physical and material resources may include equipment, laboratories, materials, tools, licenses etc.

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

06 | The definition of responsibilities and authorities does not necessarily require formal role descriptions.

4 **Identify and make available resources.** — Achievement: 6, 7

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.

07 | 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.

5 **Monitor and adjust the performance of the process.** — Achievement: 3

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.

6 **Manage the interfaces between involved parties.** — Achievement: 8

The individuals and groups including required external parties involved in the process performance are determined.
Responsibilities are assigned to the relevant individuals or parties.
Communication mechanisms between the involved parties are determined.
Effective communication between the involved parties is established and maintained.

PA 2.2 Work product management process attribute

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

1. Requirements for the work products of the process are defined.
2. 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.

Output information items		Achievement
17-05	Requirements for work products	1, 2
18-59	Review and approval criteria for work products	1
18-07	Quality criteria	1
13-19	Review evidence	4
13-08	Baseline	3
16-00	Repository	3

Generic practices 1	Achievement
1 **Define the requirements for the work products.** 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. 01 | Possible sources of documentation requirements may be best practices or lessons learned from other projects, standards, organization requirements, customer requirements, etc. 02 | There may be types of work products where no review or approval is required, thus then there would be no need to define the corresponding criteria.	1

Generic practices 2–4	Achievement
2 **Define the requirements for storage and control of the work products.** Requirements for the storage and control of the work products are defined, including their identification and distribution. 03 | Possible sources for the identification of requirements for storage and control may be: → Legal requirements, data policies → Best practices from other projects → Tool related requirements, etc. 04 | Examples for work product storage are files in a file system, ticket in a tool, Wiki entry, paper documents etc. 05 | Where status of a work product is required in base practices, this should be managed via a defined status model.	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. 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.	3
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.	4

Process capability Level 3 Established process

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

PA 3.1 Process definition process attribute

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

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. Roles, responsibilities, authorities, and required competences for performing the standard process are defined.
4. Tailoring guidelines for deriving the defined process from the standard process are defined.
5. 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.

Output information items		Achievement
06-51	Tailoring guideline	4
08-63	Process monitoring method	6
10-00	Process description	1, 2
10-50	Role description	3
10-51	Qualification method description	3
10-52	Process resource and infrastructure description	5

PA 3.1 Process definition process attribute

Generic practices 1–2 — **Achievement**

1 **Establish and maintain the standard process.** — 1, 2, 3, 4

A standard suitable process is developed including required activities and their interactions.
Inputs and outputs of the standard process are defined including the corresponding entry and exit criteria to determine the interactions and sequence with other processes.
Process performance roles are identified and assigned to the standard process activities including their type of involvement, responsibilities, and authorities.

01 | An example for describing the involvement of the process roles in the activities is a RASI/RASIC representation.

Suitable guidance, procedures, and templates are provided to support execution of the process as needed.

02 | Procedures may also include description of specific methods to be used.

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

The standard process is maintained according to corresponding feedback from monitoring of the deployed processes.
04 | For guidance on how to perform process improvements see the Process Improvement process (PIM.3).

2 **Determine the required competencies.** — 3

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.
05 | Qualification methods are e.g., trainings, mentoring, self-study.
06 | Preparation includes e.g., identification or definition of trainings, mentoring concepts, self-learning material.

Generic practices 3–4 — **Achievement**

3 **Determine the required resources.** — 5

Required physical and material resources and process infrastructure needs for performing the standard determined.
07 | This may include e.g., facilities, tools, licenses, networks, services, and samples supporting the establishment of the required work environment.

4 **Determine suitable methods to monitor the standard process.** — 6

Methods and required activities for monitoring the effectiveness and adequacy of the standard process are determined.
08 | 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.

09 | Information about process performance may be of qualitative or quantitative nature.

PA 3.2 Process deployment process attribute

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

1. A defined process is deployed based upon an appropriately selected and/or tailored standard process.
2. Assignment of persons necessary for performing the defined process to roles is performed and communicated.
3. Required education, training and experience is ensured and monitored for the person(s) assigned to the roles.
4. 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 behaviour of the process.

Output information items		Achievement
10-00	Process description	1
15-54	Tailoring documentation	1
14-53	Role assignment	2, 3
13-55	Process resource and infrastructure documentation	4
03-06	Process performance information	5

Generic practices 1–4		Achievement
1	**Deploy a defined process that satisfies the context specific requirements of the use of the standard process.** The defined process is appropriately selected and/or tailored from the standard process. Conformance of defined process with standard process requirements and tailoring criteria is verified. The defined process is used as managed process to achieve the process outcomes. 01 | Changes in the standard process may require updates of the defined process.	1
2	**Ensure required competencies for the defined roles.** Human resources are allocated to the defined roles according to the required competencies and skills. Assignment of persons to roles and corresponding responsibilities and authorities for performing the defined process are communicated. Gaps in competencies and skills are identified, and corresponding qualification measures are initiated and monitored. Availability and usage of the project staff are measured and monitored.	2, 3
3	**Ensure required resources to support the performance of the defined process.** Required information to perform the process is made available, allocated and used. Required physical and material resources, process infrastructure and work environment are made available, allocated and used. Availability and usage of resources are measured and monitored.	4
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. Results of the analysis are made available to all effected parties and used to identify where continual improvement of the standard and/or defined process can be made. 02 | For guidance on how to perform process improvements see the Process Improvement process (PIM.3).	5

Within UL Solution, we provide a broad portfolio of offerings for the automotive industry. This includes testing, inspection, assessment, certification and consulting services.
In order to protect and prevent any conflict of interest, perception of conflict of interest and protection of both our brand and our customers' brands, UL Solutions has processes in place to identify and manage any potential conflicts of interest and maintain impartiality.

Pocket Guide Automotive SPICE 4.0
ISBN-978-3-945547-47-2

UL Solutions

Kugler Maag Cie by UL Solutions
Leibnizstraße 11
70806 Kornwestheim
Germany

kuglermaag.com
kvausa.com
methodpark.com
ul.com

Acquisition processes

ACQ.4	Supplier Monitoring

Supply processes

SPL.2	Product Release

Validation processes

VAL.1	Validation

System Engineering processes

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

Software Engineering processes

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

Machine Learning Engineering processes

MLE.1	Machine Learning Requirements Analysis
MLE.2	Machine Learning Architecture
MLE.3	Machine Learning Training
MLE.4	Machine Learning Model Testing

Hardware Engineering processes

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

Supporting processes

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

Management processes

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

Process Improvement processes

PIM.3	Process Improvement

Reuse processes

REU.2	Management of Products for Reuse

Process capability levels

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

- Basic scope processes
- Flex scope processes
- Hardware Engineering processes
- Machine Learning Engineering processes

UL Solutions

KUGLER MAAG CIE
by UL Solutions

kVA
by UL Solutions

methodpark
by UL Solutions

ul.com