

VDA QMC

German Association of the Automotive Industry
Quality Management Center

Joint Quality Management in the Supply Chain

Automotive SPICE®

Guidelines

**Process assessment using Automotive SPICE in the
development of software-based systems**

Draft version for 2nd edition, May 2023

Automotive SPICE®

Guidelines

**Process assessment using Automotive SPICE in the
development of software-based systems**

Draft version for 2nd edition, May 2023

Verband der Automobilindustrie e.V. (VDA)

German Association of the Automotive Industry (VDA)

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

Non-binding VDA standard recommendation

The Association of the German Automotive Industry (VDA) recommends its members to apply the following standard for the implementation and maintenance of quality management systems.

Exclusion of liability

This VDA volume is a recommendation available for general use. Anyone applying it is responsible for ensuring that it is used correctly in each case.

This VDA volume takes into account state of the art technology, current at the time of issue. Implementation of VDA recommendations relieves no one of responsibility for their own actions. In this respect everyone acts at their own risk.

The VDA and those involved in VDA recommendations shall bear no liability.

If during the use of VDA recommendations, errors or the possibility of misinterpretation are found, it is requested that these be notified to the VDA immediately so that any possible faults can be corrected.

Copyright

This publication is protected by copyright. Any use outside of the strict limits of copyright law is not permissible without the consent of VDA and subject to prosecution. This applies in particular to copying, translation, microfilming and storage or processing in electronic systems.

Translations

This publication will also be issued in other languages. The current status must be requested from VDA QMC.

Trademark

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

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

Copyright 2023 by

Verband der Automobilindustrie e. V. (VDA)
Qualitätsmanagement-Center (QMC)

10117 Berlin, Behrenstrasse 35
Germany
www.vda-qmc.de

Preface

Market demands require permanent innovations with increasing complexity within reliable time frames. It is essential to continually improve the development processes and methods for product creation and to ensure the stakeholders quality expectations.

The VDA reworked the existing Automotive SPICE Guidelines version 1.0 based on the Process Assessment Model Automotive SPICE 4.0 that is released in conjunction with this Blue-Gold-Book. This was made to take appropriate steps to improve the quality and comparability of assessment results.

Major improvements are made regarding addition of new domains like Hardware Development and Machine Learning. The rating guidelines are provided for all processes in the process assessment model and a new recommended scope for assessments has been determined.

The Blue-Gold-Book “Automotive SPICE for Cybersecurity” from 2021 remains a separate elaboration.

The “Automotive SPICE Process Assessment Model” is increasingly used within the global automotive industry for the objective evaluation of processes and the subsequent improvement of processes at project and organization level. It shall not be misinterpreted as a development methodology. The objective in drawing up this document was to support the interpretation and application of the model for the automotive industry and to provide guidance and recommendations to increase the comparability of assessments results.

This document is aimed to support a mature and sustainable development within the automotive industry.

Content

Terms and glossary	7
Introduction	11
Document scope	13
Relation to ISO/IEC 330xx series	14
Relation to Automotive SPICE	14
Part 1: Interpretation and rating guidelines	16
1 Application of interpretation and rating guidelines	16
1.1 Overview	16
1.2 Assessment purpose	17
1.3 Defining the assessment scope	17
2 Key concepts and overall guidelines	33
2.1 Specific terms used in base practices	33
2.1.1 No Production or Construction Processes	33
2.1.2 No Procurement Process	33
2.1.3 Technical Scope of the HWE processes	34
2.1.4 The scope of “system” in SYS.x	35
2.1.5 Requirements process oriented concepts	39
2.1.6 Base Practices on Consistency and traceability	43
2.1.7 Base Practice “Communicate”	48
2.1.8 Verification process oriented concepts	49
2.1.9 No explicit notion of “specification” and “strategy” at level 1	50
2.1.10 No extra BP on evaluating alternative architectures	51
2.2 Application in specific environments	60
2.2.1 Model based development	60
2.2.2 Agile environments	63
2.2.3 Development external to the assessed project (DEX)	67
2.2.4 Application parameters	75

3	Rating guidelines on process performance (level 1)	82
3.1	ACQ.4 Supplier Monitoring	82
3.2	SPL.2 Product Release	85
3.3	SYS.1 Requirements Elicitation	86
3.4	SYS.2 System Requirements Analysis	89
3.5	SYS.3 System Architectural Design	94
3.6	SYS.4 System Integration and Integration Verification	97
3.7	SYS.5 System Verification	100
3.8	SWE.1 Software Requirements Analysis	102
3.9	SWE.2 Software Architectural Design	107
3.10	SWE.3 Software Detailed Design and Unit Construction	109
3.11	SWE.4 Software Unit Verification and Integration Verification	116
3.12	SWE.5 Software Component Verification and Software Elements Integration Verification	121
3.13	SWE.6 Software Verification	123
3.14	VAL.1 Validation	124
3.15	MLE.1 Machine Learning Requirements Analysis	127
3.16	MLE.2 Machine Learning Architecture	129
3.17	MLE.3 Machine Learning Training	131
3.18	MLE.4 Machine Learning Model Testing	134
3.19	HWE.1 Hardware Requirements Analysis	136
3.20	HWE.2 Hardware Design	140
3.21	HWE.3 Verification against Hardware Design	143
3.22	HWE.4 Verification against Hardware Requirements	148
3.23	SUP.1 Quality Assurance	152
3.24	SUP.8 Configuration Management	157
3.25	SUP.9 Problem Resolution Management	162
3.26	SUP.10 Change Request Management	167
3.27	SUP.11 Machine Learning Data Management	172
3.28	MAN.3 Project Management	174
3.29	MAN.5 Risk Management	182
3.30	MAN.6 Measurement	186

3.31 PIM.3 Process improvement	188
3.32 REU.2 Management of products for reuse	190
4 Rating guidelines on process capability level 2	192
4.1 Process Performance Management (PA 2.1)	194
4.2 Work Product Management (PA 2.2)	207
5 Rating guidelines on process capability level 3	214
5.1 Process Definition (PA 3.1)	215
5.2 Process Deployment (PA 3.2)	222
5.3 Rating consistency	226
Part 2: Guidelines for performing the assessment	230
6 Documented assessment process	231
6.1 Introduction	231
6.2 Assessment input and output	232
6.3 Parties and roles involved in the assessment	235
6.4 Assessment activities	237
7 Improvement process	249
7.1 Introduction	249
7.2 Improvement activities	249
8 Recommendations for performing an assessment	254
8.1 Assessment results	254
8.2 Validity of assessments	255
8.3 Performing an assessment	257
8.4 Assessment Report	259
9 Requirements relating to assessor qualification	263
9.1 Requirements for assessors	263
9.2 Requirements for lead assessors	263
9.3 Requirements for non-lead assessors	264
9.4 Requirements for assessor license upgrade	264
9.5 Requirements for assessing additional domains	264
Bibliography	265

Terms and glossary

In the following, definitions of terms used in the present volume are provided. When applicable, a citation of the definition provided in the ISO/IEC 330xx process assessment series of standards is given in italic letters.

Please refer to ISO/IEC 33001:2015 for a full glossary of the terms used in the ISO/IEC 330xx series *[ISO33001]*.

Term	Definition
Assessing organization	The organization which performs the assessment.
Assessment log	The formal documentation of the execution of an assessment drawn up by the assessor. The assessment log is the evidence of the assessor's assessment activities and is provided to the certification body.
Assessment scope	<p><i>Definition of the boundaries of the assessment, provided as part of the assessment input, encompassing the boundaries of the organizational unit for the assessment, the processes to be included, the quality level for each process to be assessed, and the context within which the processes operate.</i></p> <p>→ <i>[ISO/IEC 33001:2015, 3.2.8]</i></p>
Assessment sponsor	<p><i>Individual or entity, internal or external to the organizational unit being assessed, who requires the assessment to be performed and provides financial or other resources to carry it out.</i></p> <p>→ <i>[ISO/IEC 33001:2015, 3.2.9]</i></p>
Assessment team	<p><i>One or more individuals who jointly perform a process assessment.</i></p> <p>→ <i>[ISO/IEC 33001:2015, 3.2.10]</i></p>
Assessor	<p><i>Individual who participates in the rating of process attributes.</i></p> <p>→ <i>[ISO/IEC 33001:2015, 3.2.11]</i></p>
Audit	<p><i>A systematic, independent and documented process for obtaining audit evidence [records, statements of fact or other information which are relevant and verifiable] and evaluating it objectively to determine the extent to which the audit criteria [set of policies, procedures or requirements] are fulfilled.</i></p>

Term	Definition
	→ [ISO 19011]
Automotive SPICE	A process assessment and reference model conformant to the requirements of ISO/IEC 33002:2015. It is primarily addressing the development of embedded software-based systems within the automotive domain. It can be downloaded free of charge on www.automotivespice.com .
AUTOSAR	AUT omotive O pen S ystem AR chitecture: an initiative by the automotive industry for standardization of software in electronic control units (www.autosar.org).
AUTOSAR domains	Categories used to classify electronic control units by their area of application, e.g. chassis, power-train, telematics, body.
Process capability	A characterization of the ability of a process to meet current or projected business goals.
Capability level	Point on a scale of achievement of the process capability derived from the process attribute ratings for an assessed process.
Certification body	A central body which administrates the certification information of the trained assessors and classifies the trained assessors by their qualifications and practical experience according to a certification scheme.
Certification scheme	A set of rules and procedures used by a certification body to certify assessors.
Evidence	Artefact or information reflecting practice performance. Evidences are used during assessment to understand process performance and can be documents, oral information, data or information from tools or other sources.
Evidence repository	Repository for storing evidences which have been obtained.
Feedback presentation	A process step at the end of the assessment, when the assessment team provides feedback on the results of the assessment. It usually covers the main strengths and potential improvements. The set of provisional process capability profiles is also presented if appropriate.
Findings	The evaluations documented by assessors regarding strengths and potential improvements of the organizational unit which was evaluated, based on verbal affirmations from interviews and work products presented (→ <i>Evidence</i>).

Term	Definition
Indicator	<p><i>Sources of objective evidence used to support the assessor's judgment in rating process attributes.</i></p> <p>→ [ISO/IEC 33001:2015, 3.3.1]</p>
Lead Assessor	<p><i>Assessor who has demonstrated the competencies to conduct an assessment and to monitor and verify the conformance of a process assessment.</i></p> <p>→ [ISO/IEC 33001:2015, 3.2.12]</p>
Process measurement framework	<p><i>Schema for use in characterizing a process quality characteristic of an implemented process</i></p> <p>→ [ISO/IEC 33001:2015, 3.4.6]</p>
NDA	Non-Disclosure Agreement
OEM	<p>“Original Equipment Manufacturer”. In the automotive industry this term is used to describe the vehicle manufacturers. (See also → <i>Tier 1...n</i>).</p>
Organization assessed	<p>The organizational unit which is assessed. This usually refers to projects in one or more departments in the assessed organization.</p>
Practice level	<p>Lowest level of granularity within the Automotive SPICE process assessment model, determined by the “base practices” and “generic practices” of the processes. Strengths and potential improvements should be traceable to this level and are derived from expectations regarding a state-of-the-art implementation of the practices.</p>
Process assessment model (PAM)	<p><i>Model suitable for the purpose of assessing a specified process quality characteristic, based on one or more process reference models</i></p> <p>→ [ISO/IEC 33001:2015, 3.3.9]</p>
Process reference model (PRM)	<p><i>Model comprising definitions of processes in a domain of application described in terms of process purpose and outcomes, together with an architecture describing the relationships between the processes.</i></p> <p>→ [ISO/IEC 33001:2015, 3.3.16]</p>
Process context	<p><i>Set of factors, documented in the assessment input, that influence the judgment, comprehension, and comparability of process attribute ratings</i></p> <p>→ [ISO/IEC 33001:2015, 3.2.16]</p>

Term	Definition
Process (capability) profile	<p><i>Set of process attribute ratings for an assessed process</i></p> <p>→ [ISO/IEC 33001:2015, 3.2.18]</p>
Process quality characteristic	<p><i>Measurable aspect of process quality; category of process attributes that are significant to process quality.</i></p> <p>→ [ISO/IEC 33001:2015, 3.2.10]</p>
SPICE	<p>Software Process Improvement and Capability determination</p> <p>Name of the starting project, elaborating the draft of ISO/IEC TR 15504. These days the term “SPICE” is used colloquially to refer to ISO/IEC 330xx.</p>
Tier 1...n	<p>The term “Tier 1...n” is used to refer to suppliers at various levels in the supply chain. Direct suppliers to the OEM are referred to as “Tier 1”, a supplier to a Tier 1 supplier is referred to as a “Tier 2”, etc.</p>
VDA	<p>“Verband der Automobilindustrie”, the German association of Automotive Industry</p>

Introduction

The intent of this publication is to revise the Blue-Gold Book Automotive SPICE Guidelines version 1 in order to improve the quality and reproducibility of assessment results.

The objective is to give necessary clarifications and recommendations for the application of Automotive SPICE in terms of performing assessments and monitoring of resulting process improvements in the development of software-based systems.

To fulfil this mandate, the following activities were performed:

Improving the Automotive SPICE Process Assessment and Reference Model regarding structure, inconsistencies, clarifications and additional concepts. This was done with the publication of the 4.0 version of Automotive SPICE PRM/PAM [AS40].

Improving and update of the guidelines on the interpretation of Automotive SPICE and on Assessment performance. This is provided by the current publication.

Setting requirements for the qualification of assessors and update existing procedures, training materials and examinations. This will be done in collaboration with the international assessor certification scheme (intacs®) to accompany the release and roll-out of this publication [intacs].

The current publication will replace the existing Blue-Gold Print Automotive SPICE Guidelines version 1 and can be applied with its official publication in the VDA QMC online shop.

The present publication addresses the mandate by providing two parts:

Part 1: Interpretation and rating guidelines

This part provides rules for the rating performed in an assessment.

Part 2: Guidelines for performing the assessment

By defining the requirements for the assessment process, it is intended to standardize the procedure, so that the companies involved in an assessment are able to follow a defined assessments approach. This

present volume specifies the requirements related to the assessment process, as well as the qualification of assessors carrying out assessments based on Automotive SPICE.

All rules for rating in assessments reflect best practices from assessors having extensive experience in assessments based on Automotive SPICE in various applications.

Besides the knowledge of the participating members and third party members involved, the present publication leverages other sources giving valuable input, which has been proven in many years of assessment practice and assessor trainings, in particular see bibliography:

Document scope

The scope of the current document is to support assessments using Automotive SPICE. It addresses the process of performing the assessment and in detail the rating performed in an assessment. It is based on the 4.0 version of the Automotive SPICE Process Assessment and Reference Model.

Automotive SPICE 4.0 is a full process assessment model (incl. reference model) complying to the requirements of ISO/IEC 33002. It can be used on its own to perform assessments. The intention of this publication is NOT to replace or extend the Automotive SPICE PAM or PRM.

The guidance given in Part 1 of this document is intended to support reproducible assessment results but cannot reflect all the variety in practicing engineering, management and supporting processes. Assessment teams need to understand the context of the assessed organization before they judge a rating rule from this document as applicable. Lists and enumerations that supplements the process related guidance need not to be interpreted as checklists for implementation and are not intended to be complete. References to rating rules must not be used as weakness statement to justify a rating of an indicator or a process attribute.

The aim of the Part 2 of this document is to set guidelines for the application of Automotive SPICE to assist the assessors while planning, executing and reporting the assessment. Beside this it specifically addresses the improvement process which should resolve issues found in an assessment.

The target audience is predominantly assessors which are active in the automotive domain, but can also be seen as an additional input for assessments in other domains. It also addresses other parties or roles involved or affected by an Automotive SPICE assessment like the assessing organization, the assessed organization or the assessment sponsor.

Furthermore, the document is intended to support the understanding of the assessment process and should be taken in case of dissent about the result of an assessment as a basis for clarification.

Relation to ISO/IEC 330xx series

The ISO/IEC 330xx series of international standards define the requirements and resources needed for process assessment. Several standards in the ISO/IEC 330xx family were intended to replace and extend parts of the former ISO/IEC 15504 series.

ISO/IEC 330xx process assessments are conducted based on three core elements:

- Process models that define processes, the entities that are the subject to assessment;
- Process measurement frameworks that provide scales for evaluating specified attributes; and
- A specification of the process to be followed in conducting assessments.

The intention of the Working Group 13 of the VDA QMC was to provide a domain specific set of documents covering these three elements for performing assessments conformant to ISO/IEC 33002. This has been achieved

- by providing the Automotive SPICE process reference and assessment model *[AS40]*
- by referencing ISO/IEC 33020:2015 *[ISO33020]* as the process measurement framework for assessment of process capability in the Automotive SPICE PAM and
- by providing a documented assessment process conformant to ISO/IEC 33002 in chapter 6 of this volume.

Relation to Automotive SPICE

At the beginning of the development of Automotive SPICE 4.0 different extensions to the previous version 3.1 have been evaluated by the working group 13 to provide an updated process set suitable for assessments in the automotive domain. Compared to the version 1.0 of the Guidelines for Automotive SPICE this documentation covers all processes in the process assessment model.

Since the scope of application has been enlarged to cover different engineering domains, the working group 13 decided to define a new recommended scope for performing assessments. This was done in

order not to increase the effort for performing an assessment and to provide reproducibility of assessment results.

It is a principle of process assessments according to the ISO/IEC 330xx series that the process scope (the selection of processes to be investigated in an assessment) might be adapted in accordance with the sponsor and with respect to the purpose of the assessment.

Part 1: Interpretation and rating guidelines

1 Application of interpretation and rating guidelines

1.1 Overview

The purpose of part one of the current publication is to support the assessors in interpreting the Automotive SPICE process reference and assessment models and rating the process attributes for the given target capability level.

These recommendations are based on the long-time experience of the assessor community. Most of the assessments in the automotive domain do not address capability levels higher than 3. Therefore, no guidelines are provided for level 4 or 5 due the reduced amount of experience in application of these levels.

Chapter 1, “Application of interpretation and rating guidelines” introduces a clearer definition of how-to set-up and consider the assessment purpose and scope input and provides an overall guideline on rating in an assessment.

An integral part of the interpretation and rating guidelines are rules addressing specific key concepts, application environments and the different capability levels.

In **chapter 2, “Key concepts and overall guidelines”** rules related to key concepts introduced or modified with the 4.0 version of Automotive SPICE are given. Further, rules for rating in specific application environments are provided.

Chapter 3, “Rating guidelines on process performance (level 1)” is related to the process specific outcomes, base practices and work products associated with the capability level 1. In this chapter, specific rating rules are given for each process of the VDA Scope.

In **chapter 4, “Rating guidelines on process capability level 2”** and **chapter 5, “Rating guidelines on process capability level 3”** specific rating rules for each process attribute of level 2 and 3 are given.

1.2 Assessment purpose

Automotive SPICE assessments are performed within a certain variety of use cases for a specific purpose. In general, the purpose of process assessment is to understand and assess the processes implemented by an organizational unit.

Specifically, as defined in ISO/IEC 33001, 3.2.6 the assessment purpose is a

“statement provided as part of the assessment input, which defines the reasons for performing the assessment”.

Note: The assessment purpose may not be confused with the process purpose.

Based on this definition, the assessment purpose needs to be documented when identifying the assessment input (See also 6.2.2 *Assessment inputs*).

Additionally, it is strongly recommended to include the assessment purpose in the assessment report (See chapter 8.4.2 *Formal information about the assessment*).

The assessment purpose may be documented by specifying the specific objectives of the assessment such as:

- Providing improvement potentials of specific processes of an organizational unit.
- Identifying and reducing process-related risks for a specific product delivery.

The assessment scope (see next chapter) shall be defined to cover the assessment purpose, accordingly.

1.3 Defining the assessment scope

As defined in ISO/IEC 33001, 3.2.8 the assessment scope shall provide

“The definition of the boundaries of the assessment, provided as part of the assessment input, encompassing

- *the boundaries of the organizational unit for the assessment,*
- *the processes to be included,*

- *the quality level for each process to be assessed and*
- *the context within which the processes operate (process context)”.*

Note: ISO/IEC 33001 uses the term “quality level”. Since Automotive SPICE applies only capability levels as a specific implementation of a quality level, the term “capability level” is used throughout the process assessment model and within this guideline.

1.3.1 Defining the boundaries of the organizational unit

As defined in ISO/IEC 33002, the boundaries of the assessed organizational unit according to the definition in ISO/IEC 33001 shall be given in the assessment scope. The definition of the organizational boundaries shall be given in terms of

- the localization of the involved organizational unit(s) and
- the responsibilities of the involved organizational unit(s).

These boundaries shall always be defined with respect to the defined processes (see chapter 1.3.2) and the defined process context (see chapter 1.3.3).

In summary, the boundaries shall identify which part of the organization is responsible for the performance of the given processes in the scope and provide information about the location of the development sites. This is a necessary input for the planning of the assessment.

1.3.2 Defining the processes to be included

It is a principle of process assessments according to the ISO/IEC 330xx series that the process scope (the selection of processes to be investigated in an assessment) might be adapted in accordance with the sponsor and with respect to the purpose of the assessment. Identifying the processes “under scope” is a significant step to tailor the content of the Automotive SPICE assessment model to cover the assessment purpose in terms of the specific development scope of the project assessed.

The following VDA process scope provides a standard selection of processes that are recommended to give a comprehensive overview of an assessed project. Depending on the purpose of the assessment, this may be tailored or extended.

It consists of a set base processes, including core supporting processes and the project management process. These processes need to be enhanced by at least one set of engineering processes addressing a specific development domain within the project (Plug-In). Depending on the disciplines involved in the development several plug-ins may be combined to enable a needs-based coverage of the assessment purpose.

According to the purpose of the assessment, the recommended VDA process scope for an assessment might be extended by other processes from Automotive SPICE (Flex scope).

In specific cases also other process reference models may be considered.

The recommended VDA Scope is based on the release 4.0 of the Automotive SPICE process reference and assessment model [Automotive SPICE].

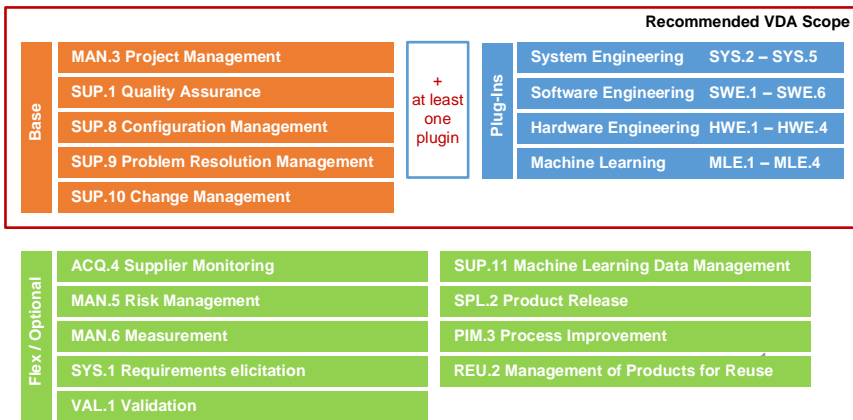


Figure 1-1: Recommended VDA Scope and optional processes

For certain use cases, the recommended VDA scope may also be further tailored. A typical example could be focusing on some specific aspects of the development to identify process specific improvement opportunities only.

The processes to be assessed shall be identified and a rationale shall be documented for choosing this specific set of processes with respect to the purpose of the assessment.

Each process in the defined assessment scope shall be assessed and the result shall be documented in the assessment report. To ensure a sufficient base for rating, each process defined in the scope shall be at least once performed.

Exceptionally there might be the need to exclude or add processes after agreement of the assessment scope, e.g. during the execution of the assessment. Any exclusion of processes in the rating shall be documented by a modified assessment scope and shall be approved by the sponsor of the assessment. An exclusion of a process must not be done based on a “not applicable” classification of the process.

1.3.3 Defining the target capability level for each process

Since the measurement framework used in Automotive SPICE is applicable for rating capability, the term “*capability level*” as a refinement for the “*quality level*” is used. There are five capability levels specified in the PAM for the assessment. As mentioned before, the rules given in this publication are only considering capability levels 1 to 3, due to the fact that this covers most of the Automotive SPICE assessments in the automotive domain.

Since the planning of the assessment is significantly influenced by the choice of the target capability level, the intended maximum capability level to be assessed shall be defined for each process as part of the assessment scope.

1.3.4 Defining the process context in the assessment scope

In ISO/IEC 33001, 3.2.16 the process context is defined as

“the set of factors, documented in the assessment input, that influence the judgment, comprehension, and comparability of process attribute ratings”.

When defining the process context, the boundaries within which the processes operate in terms of a set of aspects shall be identified.

Exemplary aspects for boundaries to be combined for defining the process context are:

- Functional / Content related
- Time / Release related

- Requirements / change related

Examples:

- All past releases, a specific or a selection of specific product releases
- Since a specific point in time (to address new processes, organizational changes)
- Including or excluding platforms/ standard components
- Including or excluding open source
- All requirements and changes valid for a specific product release
- All requirements and changes excluding particular product releases
- All requirements and changes related to a defined architectural element.
- All requirements and changes to be implemented between two defined project milestones.
- All changes and affected stakeholder requirements in a (delta) project developing additional functionalities based on an existing system or software.
- Complete system delivered by a supplier.
- A complete software platform delivered by an internal or external organization.

Note: The definition of the process context needs to be aligned with definition of the boundaries of the organizational unit (see chapter 1.3.2).

Each process attribute rating shall strictly remain within the boundaries of the process context in the assessment scope.

1.3.5 Defining instances when setting up the assessment scope

Depending on different constraints, the same process might be applied in different process instances within the same project e.g. for parts that are developed using model-based approaches in comparison to parts that are manual coded. Therefore, different process attribute ratings might be derived for different instances of the rated process. The corresponding rating methods are provided in the measurement framework of ISO/IEC 33020, 5.4.

There are different use cases, where a separation of a process into instances may be reasonable. Building instances may reflect the need of a higher granularity of the assessment findings due to the execution of the process with different approaches or in separate organizations or locations.

Setting up instances doesn't change the given scope and process context of an assessment. If instances are defined, they all shall be rated according to the given scope and the rules shall be applied on each process performance attribute rating of each single instance.

To provide a more detailed understanding of the term "process instance", the following exemplary use cases are given:

- A project has used standard process version 2 until March 2016, and standard process version 3 since then. If the assessor can clearly see that the usage of these two standard process versions actually do not overlap, a reasonable instantiation may be:
 - A rating of process instance "SWE.3 until March 2016"
 - A separate rating of Process instance "SWE.3 after March 2016"
- Parties responsible for different hierarchical levels in the architecture of a mechatronic product development project use different requirements engineering approaches, e.g.:
 - A rating of process instance "SYS.2 / Mechatronic level"
 - A separate rating of process instance "SYS.2 / ECU level"
 - A rating of process instance "SWE.2 / Application SW level"
 - A separate rating of process instance "SWE.2 / Basis SW level"
- Different reuse strategies used for different parts of the overall SW, e.g.
 - A rating of process instance "SWE.x / Platform code"
 - A rating of process instance "SWE.x / Project specific developed code"
- Different SW development paradigms are used for different parts of the overall SW, e.g.:

- A rating of process instance “SWE.3 / Model-based development”
- A rating of process instance “SWE.3 / Manual coding”
- Different sub-projects use different project management approaches, e.g.:
 - A rating of process instance “MAN.3 / SW level”
 - A separate rating of process instance “MAN.3 / Overall project”
- Different organizational units develop different parts of the software. These organizational units might even be located in different geographical locations and regions, with probably different social-cultural backgrounds, e.g.:
 - A rating of the process instances “SWE.x / Standard SW components in the reusable platform – Asia”
 - A rating of the process instances “SWE.x / Standard SW components in the reusable platform – Europe”
 - A rating of the process instances “SWE.x / All customer-/application-specific SW components – Germany”

Reasons for assessing different process instances separately can be meaningful e.g.

- in order to have company-internal benchmarking
- for a more accurate understanding of the various characteristics in the organization in order to better launch precise process improvement initiatives

The ratings of the process attributes for each process instance shall be documented in the assessment report.

In case instances are defined, a process is rated independently for each instance thus resulting in separate ratings of the process. This requires an aggregation of the results to a single process attribute rating considering the impact of the instance on the overall rating. The recommendations how to perform the aggregation can be found in chapter 1.3.3.

1.4 General rating practice

1.4.1 Rating outcomes and indicators

1.4.1.1 Rating of practices

According to the ISO/IEC 33002, which defines the requirements for performing process assessments, it is always mandatory to rate the process attributes [ISO/IEC 33002:2015].

Nevertheless, in terms of achieving a structured approach to determine the rating of a process attribute, ISO/IEC 33020 provides the following possibility:

Process outcomes and process attribute outcomes may be characterised as an intermediate step to providing a process attribute rating. [ISO/IEC 33002:2019, 5.4]

As mentioned in the overview chapter 1.5, this publication provides rating rules. These rules directly affect the process attribute rating or address the so-called “characterization of process (attribute) outcomes”.

In the recent years of application of Automotive SPICE, the terminology “rating a base practice” or “rating a generic practice” has been established as a synonym for performing this step of characterization. To avoid confusion in the community, the present publication continues using this terminology when defining rules which are not directly affecting process attribute ratings.

Formally, since base and generic practices are indicators and thereby only sources of objective evidence used to support the assessor’s judgment in rating process attributes, a rating of indicators is not a defined term in the ISO/IEC 330xx series.

In this context – and since Automotive SPICE has a defined relationship between process outcomes and base practices – the terminology “rating a ... practice” means:

“Characterizing the outcome based on the indicator to compile a consistent process attribute rating”

1.4.1.2 Consideration of information items

As described in the Automotive SPICE assessment model, information items (II) including their characteristics (IIC) serve as a second type of assessment indicators. They are provided as guidance for “what to look for” in the documentation available in the assessed organization.

The extent of implementation of an information item (inline with its defined characteristics) in a work product serves as objective evidence supporting the assessment of a particular process. Information item characteristics should be considered as indicators when considering whether, given the context, a documented information is contributing to the intended purpose of the process.

Please refer to the process assessment model for further understanding of information items and their relation to work products produced by the organization assessed.

1.4.2 Independent rating of processes

A process assessment model provides a two-dimensional view of a process quality characteristic. Each process within the scope (process dimension) shall be rated individually on the scale provide within the capability dimension.

This means that only weaknesses of that very process alone shall be the source of a potential downrating. This implies that only base practices explicitly referring to another process (such as the Consistency/Traceability BP’s) can be downrated, because these are the only “connection points” between processes.

[GEN.RL.1] A rating of PA 1.1 of P or N for a process X shall not be used to downrate PA 1.1 of the process Y.

1.4.3 Sampling of work products for rating

The selection of the work products has to be carried out carefully to ensure that work product samples are representative, comprehensive, and provide evidence of the implemented process.

1.4.3.1 Selection of work product samples

The following aspects apply for the selection of work products:

- Coverage of the most important functions, which are relevant for the assessment scope
- Coverage of new functionality, adapted functionality, reused software and platform software according to the assessment scope
- Coverage of the whole spectrum of ASIL levels applied within the assessment scope
- Coverage of manual coding (all programming languages used) and model based development (all modelling tools used), where applicable

Metrics (e.g. number of requirements, cyclomatic complexity, lines of code, number of change requests) can support the selection of work product samples. It can be useful to select units with different complexity to sample the corresponding detailed designs.

For the engineering processes the following approach is recommended: The assessor chooses stakeholder requirements based on above-mentioned aspects. The work products selected for evaluating the indicators of the processes should mark a clear path through the engineering life cycle. The same approach should be applied when evaluating supporting processes such as change management or problem management.

Although the assessed organization may propose certain work products, it remains the assessor's decision to which extent these work products are considered for the process attribute.

In all cases the number of work product samples selected shall be representative to cover the given assessment purpose and scope.

1.4.3.2 Plausibility checks of work product samples

All documents used as candidate for objective evidence have to be checked for consistency, in terms of plausibility of the last change time stamp and appropriateness of the change history. The latter can be easily checked by inspecting the history of the work product in the respective tool which is used for configuration or document management. If a document has been initially generated shortly before the assessment it should not be considered for the rating of the process attribute in question unless there is a plausible reason for the late documentation.

The history of the work product should show an appropriate life cycle and a number of versions which correlates with the update cycle of the respective work product.

For instance, it could be expected that if a schedule should be updated on a weekly basis there is at least one version per week (or some evidence that an update was not necessary). Technical documents tend to have more versions than plans. However, if the architecture is based on a platform, there may not be that many versions. It is up to the assessors to check whether the number of versions reflects appropriately the life cycle and status of the project and fulfills the purpose of the process attribute which is assessed.

1.4.3.3 Content-related examination

The content-related examination of the work products should always cover the whole scope of the assessment.

This means based on the criteria for the selection of the work products samples the whole scope shall be represented.

In the limited time it is not possible to cover all aspects of the project. Nevertheless, the samples shall also be checked regarding the right content. For the content of work products, the work product characteristics can be used as guidelines.

The system requirements for example are not only to be checked to determine whether there are linked stakeholder requirements but also if the system requirements reflect the intention of the stakeholder requirements. Another example would be to check the unit tests against the detailed design. The engineer should explain the detailed design. The unit tests are then checked against the detailed design. Inconsistencies found between the test cases and the explanation of the detailed design shall be considered when rating the process attributes.

Automotive SPICE shall not be mistaken for a checklist. The assessor has the duty to check appropriate instantiation of documentation to cover the different process attributes. Appropriateness is based on e.g. the scope, the size and complexity of the project team (e.g. distributed development), the size and complexity of the product, the timeline, and other influencing factors as defined in the process context.

1.4.4 Aggregation of process attribute ratings

It is recommended to use the rating method R2 from ISO/IEC 33020 for the rating of each process attribute.

This means,

- 1) firstly, to rate each process attribute for each process within the scope of the assessment for each process instance;
- 2) and secondly, aggregating the process attribute ratings of the process instances.

An aggregation of the process attribute ratings of all process instances is mandatory. This means, in the assessment report there will be one additional set of process attribute ratings for the aggregation.

The aggregation is done according to the following schema (“one dimensional aggregation using arithmetic mean” according to ISO/IEC 33020):

- 1) Firstly, in accordance with ISO/IEC 33020 NPLF rating values can be expressed as interval values as follows:

$$N \rightarrow 0; P \rightarrow 1; L \rightarrow 2; F \rightarrow 3$$

with rounding the result to the nearest integer (by rounding up or down), and converting the result back to the corresponding ordinal rating. Rounding rules are:

- rounding down to the nearest integer when the average value is less than the midpoint between consecutive integers;
 - rounding up if the average value is at or above the midpoint between consecutive integers.
- 2) Secondly, the aggregation can be done
 - a. by calculating an arithmetic mean, or
 - b. by assigning these internal values a percentage weighting first, and then converted back to the ordinal NPLF rating scale. Weightings and their rationale must be explained in the assessment report, and may depend on e.g.
 - size of personnel of organizational unit/ sub-project
 - strategic significance of the product, e.g. commodity vs. new innovative products

- contribution to the revenue in %
- criticality of product parts, e.g. a risk class according to ISO 26262

	Process instance A	Process instance B	Process instance C	Aggregated rating
2a. Arithmetic mean without any weighting of process instances	L (2)	L (2)	F (3)	$(2+2+3) / 3$ → L (2.33)
	P (1)	L (2)	F (3)	$(1+2+3) / 3$ → L (2)
	N (0)	P (1)	F (3)	$(0+1+3) / 3$ → P (1.33)
2b. Arithmetic mean with weighting	L (2) 70%	L (2) 15%	F (3) 15%	$(2*0.7+2*0.15+3*0.15)$ → L (2.15)
	P (1) 70%	L (2) 20%	F (3) 10%	$(1*0.7+2*0.2+3*0.1)$ → P (1.4)
	N (0) 30%	P (1) 20%	F (3) 50%	$(0*0.3+1*0.2+3*0.5)$ → L (1.7)

Each row represents a process as defined in the assessment scope.

1.5 Application of rating rules

1.5.1 Objective

Rating rules are intended to reduce variance in rating decisions across assessors because of different individual interpretation of Assessment Indicator and rating dependencies. This is seen as one of the key factors by the authors of this publication to improve the quality, reproducibility, and comparability of assessment results.

1.5.2 Rule Semantics

A Rating Rule (RL) in this Guideline is defined as a directive on how to rate indicators. These can be

- conditional (i.e. dependent on a specific context or domain such as software, firmware, hardware, mechanical engineering, machine learning etc.). A condition can refer to:
 - some context-specific scenario
 - the rating of an individual indicator
 - the rating of particular indicators
 - the ratings across particular indicators
- or unconditional (i.e. irrespective of any specific context or domain such as software, firmware, hardware, mechanical engineering, machine learning etc.).

In the case the assessor needs to deviate from an RL a compelling justification must be documented in the Assessment Report.

Examples of unconditional rules:

[CL2.RL.xx] SUP.1.BP3 shall not be rated higher than the ratings across GP 2.2.4 of all processes.

Examples of conditional rules:

[TPS.RL.xx] If 3rd Party Software is used but a valid license agreement is absent or not reflected, then SWE.2.BP1 shall be downrated

[AGE.RL.xx] If the software architecture is modified incrementally and impact analyses evidence that changes were discussed then SWE.2.BP1 shall not be downrated.

1.5.3 Rating terminology and patterns

	Wording	Explanation
1	If <condition> then X shall not be downrated.	<p>Condition can refer to:</p> <ul style="list-style-type: none"> • some context-specific scenario • the rating of an individual indicator B • the rating of particular indicators • the ratings across particular indicators <p>'X' can refer to a single indicator or to a set of indicators.</p>
2	If <condition> then X shall be downrated.	<p>Condition can refer to:</p> <ul style="list-style-type: none"> • some context-specific scenario • the rating of an individual indicator B • the rating of particular indicators • the ratings across particular indicators <p>'X' can refer to a single indicator or to a set of indicators.</p> <p>The indicator(s) shall be downrated for at least one step of the rating scale. It is the decision of the assessor, if a further downrating is necessary to reflect further identified weaknesses.</p>
3	X shall not be rated higher than <condition>.	<p>Condition can refer to:</p> <ul style="list-style-type: none"> • the rating of an individual indicator B • the rating of particular indicators • the ratings across particular indicators <p>'X' can refer to a single indicator or to a set of indicators.</p>

1.5.4 Further instructions for the application of rating rules

1.5.4.1 No “rating rule algebra”

There might be cases in which for rating a process attribute or Assessment Indicator different rules apply in parallel. However, the application of n different rules, each requiring a downrating, must not automatically lead to a downrating of this indicator exactly n times according to the NPLF scale. It remains the responsibility of the leading assessor to decide on the final rating value considering the actual context, gathered objective evidence, and identified process risk.

There can be rules that define for two indicators A and B in the same process “*If A is downrated then B shall be downrated*”. Still, how many NPLF steps B actually needs to be downrated also depends on the actual context, gathered objective evidence, and identified process risk.

1.5.4.2 Assessment Report and Record

A rating rule, generally, cannot replace comprehensive weakness statements in the Assessment Report: any downrating, or when detecting weaknesses within the percentage range of the Fully value, requires providing a comprehensive explanation of the associated process risk, substantiated by traceable objective evidence. Omitting, or neglecting, weakness statements in favor of only referring to rating rules is not a sufficient basis and therefore renders the Assessment Report invalid. A rating rule may only support a given weakness and a given rating. The leading assessor takes responsibility here.

2 Key concepts and overall guidelines

2.1 Specific terms used in base practices

Processes in Automotive SPICE are passed several times within the project lifecycle. This iterative work concept is considered in the description of the processes (except: ACQ.2; Automotive SPICE for Cybersecurity).

As a consequence there is no hierarchical or temporal dependency for Base Practices and processes. The Base Practices do not imply a certain sequence, hierarchical order or pattern. They are connected in a more logical order.

Therefore also continued re-evaluation of work products and work packages is in certain processes necessary (e.g. MAN.5, MAN.3).

2.1.1 No Production or Construction Processes

This PRM/PAM does not define a process or Assessment Indicators for production processes. To avoid redundancies and potential inconsistencies with other international standards having production in scope such as IATF 16949 or VDA 6.3, PRM and PAM counterparts of production processes are not included at all.

Correspondingly, there is no process for prototype and sample construction/workshops (German: 'Musterbau') either.

For these reasons, 'process interfaces' to the production domain are required. In this HWE PRM/PAM this is achieved by means of

- output Information Items characteristics for HWE.2:
 - 03-54 Hardware Production Data (including the bill of materials)
 - 17-57 Special Characteristics
- the BP 'Ensure use of compliant samples', including comprehensive Notes, for HWE.3 and HWE.4.

2.1.2 No Procurement Process

No procurement is introduced in this PRM/PAM for the following reasons:

- Hardware development is requirements-driven, too. Therefore, what matters is compliance to the requirements for the respective environment, irrespective of the source from which HW or mechanical parts are obtained. Verification (HWE.3, HWE.4, SYS.4, SYS.5) will demonstrate that the physical product or sample is compliant with the design and with the requirements, respectively.
- There is no predefined standard for procurement at the level of abstraction of a PRM/PAM beyond what is IATF 16949. Thus, defining a procurement process here would have required the cooperation with other parties competent in the procurement domain. The identification of, and collaboration with, such would have significantly delayed Automotive SPICE v4.0 PRM/PAM.

A 'process interface' to procurement can be considered existent by means of BP 'Develop hardware detailed design' in HWE.2, together with Note 7.

2.1.3 Technical Scope of the HWE processes

The technical scope of the HWE processes is electrical or electronic hardware engineering. This excludes:

- system level engineering, i.e. neither the mechatronic nor the ECU level. See also the definition of the term "hardware" in the glossary.
- procurement (see Section 2.1.2)
- mechanical or hardware sample manufacturing (see Section 2.1.1)
- production processes (see clause Section 2.1.1).

However, process interfaces are included to

- procurement in terms of receiving physical design-compliant hardware parts;
- production and prototype/sample workshops in terms of providing information such as production data and requirements, and receiving compliant samples, respectively.

Note that in this context, the definition of 'hardware part' and 'hardware component' can represent ISO 26262's notions of 'hardware subpart' and 'hardware elementary subpart'.

2.1.4 The scope of “system” in SYS.x

The scope of the SYS processes can be interpreted in a generic way, i.e. they are not tied to a particular system boundary. This also means that the Automotive SPICE PRM/PAM does not represent a product hierarchy. Rather, via different process instances the SYS.x processes may represent different levels of a product, e.g. mechatronic system, a drive (motor plus ECU), or an ECU).

The system boundary for:

1. a **mechatronic system supplier or drive (i.e. motor plus ECU) supplier** would be the mechatronic product. Both the mechatronic system boundary and the ECU system boundary would be reflected by separate process instances of the SYS processes in a decomposed manner. To the ECU system boundary within the mechatronic system the considerations in (2.) above apply. See also Figure 2-1.

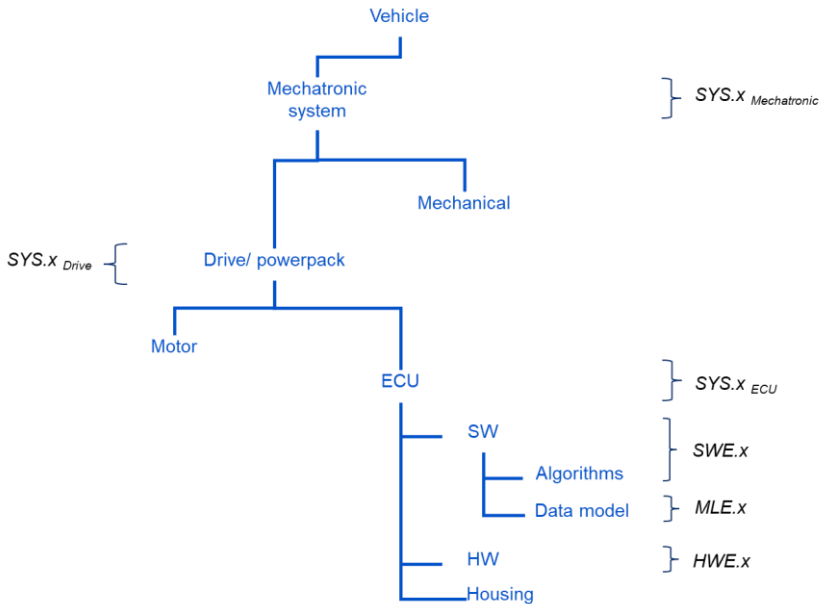


Figure 2-1 – Possible use of process instances to represent a mechatronic product composition

2. a **control device supplier** would be the ECU. This system boundary can also be reflected by the SYS processes because it typically comprises hardware, software, housing, connectors etc. In consistency with the scope of this document, the HWE processes should then be used to reflect development of the fully assembled PCB. In this respect, the definitions of ‘hardware part’ and ‘hardware component’ in this document apply. See also Figure 2-1.
3. a **semiconductor supplier** would be e.g. a microcontroller or a system-on-chip. This system boundary should be reflected in the Automotive SPICE® SYS processes because besides hardware it typically comprises a mechanical housing, firmware etc. The HWE processes should then be used to reflect the hardware-related interior of this system. Note that in this context, the definition of ‘hardware part’ and ‘hardware component’ can

represent ISO 26262's notions of 'hardware subpart' and 'hardware elementary subpart'. See also Figure 2-2.

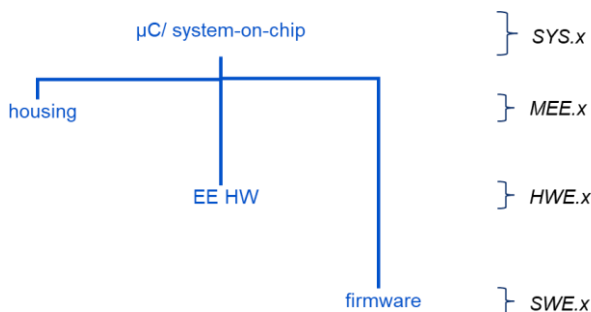


Figure 2-2 – Possible use of process instances to represent a microcontroller or system-on-chip

4. a “software system”

A further scenario is a coherent software comprising different pieces of software each of which running on a different node and/or target. Sometimes only the overall software behavior is in focus, therefore the nodes and targets being considered transparent. Some people refer to this as a “software system”.

A seemingly obvious approach could be not to address such a “software system” via SYS.x in favor of SWE.x because it is about software. Indeed, the

- overall software black-box behavior could be addressed via SWE.1
- logical and technical software interfaces between the different pieces of software in such a “software system” could be addressed via SWE.2.BP1; the technical interfaces behind memory-mapped IOs or microcontroller registers such as cables, connectors, or bus connections in between would not need to be considered in SWE.2.
- logical interactions could be addressed as SWE.2.BP1

- the “interior” of each piece of software could be addressed via SWE.3 and SWE.4

However, this view causes problems when it comes to SWE.5. The software requirements will (as demanded by SWE.1.BP1) include nonfunctional expectations such as response times or processing time limits. This is indeed serves as meaningful input for software integration verification. However, such timing requirements cannot be realized, and be verified, without considering the nodes and targets as these will, also, consume time budgets. Depending on the technical realization, these time budgets will even differ. Similar issues arise when discussing SWE.6.

Further example:

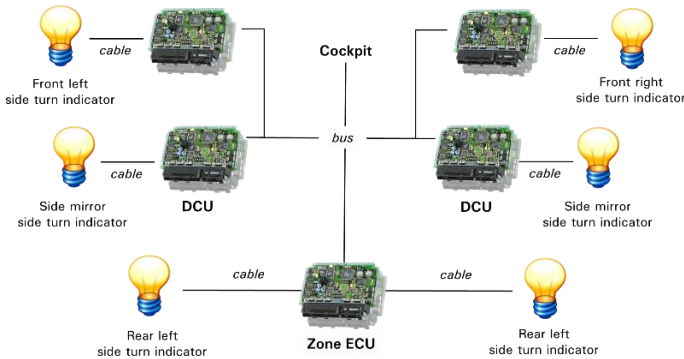


Figure 2-3: Example architecture

Consider the software-controlled synchronous blinking of all-round emergency flashers (Figure 2-3). This cannot be viewed as a sole “software system” while neglecting the hardware targets: during bus-off the different pieces of software cannot communicate. Therefore, they must have established a common pulse scheme before as otherwise during bus-off the synchronous flashing cannot be consistently maintained. The hardware, however, is subject to tolerances, and tolerances can change over time (e.g. because of thermal influence), potentially resulting in asynchronous flashing.

As a consequence, the notion of “software system” still requires using the SYS.x level as network nodes and hardware targets are, in fact, relevant. The SWE.x processes alone do not appear appropriate to address such scenarios.

2.1.5 Requirements process oriented concepts

2.1.5.1 Characteristics of requirements in SYS.2, SWE.1, HWE.1

The original motivation of having an extra Verification Criteria BP in Automotive SPICE was that, according to the requirements engineering state-of-the-art, a requirement shall be documented in a verifiable way, otherwise it does not represent a requirement. The former extra Verification Criteria BP was supposed to emphasize that. However, this has introduced PAM misunderstandings:

1. Consider ratings such as

- BP1 “Specify Reqs” = F
- BP4 “Ver Criteria” = N or P

It is difficult to argue how requirements as a whole (BP1), which have *not* been formulated in a verifiable way (BP4 = N/P), can be rated as F. Further, non-verifiable requirements even put in question how the entire process purpose can be regarded as being fulfilled.

2. The distinct verification criteria BP appears to suggest that isolated information documented separately from requirements would be necessary. However, verification criteria are actually inherent in a requirements statement:

Example 1:

#1 “The ECU shall be *able to receive 100 to 110 bus messages within 1 [s] with a tolerance of +0.2[s]*”

Example 2:

#1a “The ECU shall be *able to receive bus messages*”

#1b “When receiving bus messages, the ECU shall be able to receive *100 to 110 within 1 [s] with a tolerance of +0.2[s]*”

(The texts in italics are an example of information needed to make the requirement verifiable)

3. The Automotive SPICE Guidelines v1.0, clause 2.1.3, suggested that

*“There **may** be ‘explicit additional verification criteria’ on top of what a requirement already says, ... such as ‘Identification of a verification method or verification step (e.g. software test, system test) is necessary, ... special test methods, environments, ...”*

The word ‘may’ makes it clear that this is optional for requirements processes, i.e. not mandatory. Absence of such information therefore cannot be used for downrating.

Furthermore, the rules for SWE.6.BP1 and SYS.5.BP1 in the VDA Automotive SPICE Guidelines v1.0 expected the same information as quoted above in italics. Consequently, downrating this would mean “double punishment for both the requirements and the testing process which is not considered compliant with ISO/IEC 33004’s notion of disjoint processes in a PRM.

Further, ‘preconditions’, ‘verification methods’, ‘verification environment’ are testing or verification concerns, respectively, but not requirements concepts (“Separation of Concerns” principle) which are now correctly, and exclusively, addressed in SYS.4, SYS.5, SYS.4, SYS.5, SWE.4, SWE.5, SWE.6, HWE.3, and HWE.4.

4. Verifiability is only one out of many state-of-the-art requirements characteristics. Others are according to ISO/IEC IEEE 24765, ISO IEEE 29148, ISO 26262, INCOSE Guide for Writing Requirements, IREB CPRE e.g.

- design-free/implementation-free
- unambiguous/comprehensible
- consistent in itself, not contradicting any other requirement

- complete in itself
- no redundancy across requirements
- atomic/singular

In order to resolve all these misinterpretations the new BP1 in SYS-2, SWE.1, and HWE.1 was introduced, which integrates the notion of verification criteria.

Note that the decision of requiring characteristics for requirements at Capability Level 1 is not in conflict, or semantically overlapping, with GP 2.2.1. Reasons:

- As pointed out in [Metz2016], requiring quality characteristics for outcomes is not exclusive to Capability Level 2.
- GP 2.2.1 of SYS.2 may address different quality criteria such as structural requirements (e.g. by means of templates) or checklists

2.1.5.2 Terms ‘Functional Requirement’ and ‘Non-functional Requirement’

There is no clear internationally agreed definition of the terms ‘functional requirement’ and ‘non-functional requirement’, see discussion of references below. However, Automotive SPICE still uses the two term ‘functional requirement’ and ‘non-functional requirement’ in requirements-oriented processes in order to

- make practitioners not forget about the importance of equally reflecting on ‘non-functional’ characteristics
- enable assessors to downrate the absence of such information in requirements.

ISO/IEC IEEE 29148 defines in clause 5.2.8.3:

- *‘Functional/Performance. ... describe the system or system element functions or tasks to be performed by the system. ...’*
- *‘Quality (Non-Functional) Requirements. – Include a number of the ‘ilities’ in requirements to include, for example, transportability, survivability, flexibility, portability, reusability, reliability, maintainability and security.’*

The **IREB CPRE** says that

- *‘Non-functional requirements’ is an umbrella term and, thus, represents ‘quality requirements’ or ‘constraints’.*
- *Quality requirements are said to be e.g. performance, reliability, usability, portability.*

In **ISO/IEC IEEE 24765** the following can be found:

- There are two definitions for ‘functional requirement’:
 1. *‘A statement that identifies what a product or process must accomplish to produce required behaviour and/or results’*
 2. *‘A requirement that specifies a function that a system or system component must be able to perform’*

- The definition of ‘non-functional requirement’ is

‘A <software> requirement that describes not what the <software> will do but how the <software> will do it.’

- Non-functional requirements are further claimed to be synonymous to ‘design constraints’.

The systems engineering **INCOSE Guide for Writing Requirements** informs:

- *‘Types of requirement. Requirements that address capability and function may be expressed in a different manner to constraints and requirements specifying other system properties (often confusingly called ‘non-functional’ requirements – a term that will not be used again in this guide). The guide is intended to cover the whole range of requirement types.’*

2.1.5.3 “Functional” and “Nonfunctional” do not serve as requirements types

Base Practice 2 of both SYS.2 and SWE.1 require the structuring of requirements:

BP2: Structure system/software requirements. Structure and prioritize the system requirements.

NOTE 3: Examples for structuring criteria can be grouping (e.g. by functionality) or product variants identification.

NOTE 4: Prioritization can be done according to project or stakeholder needs via e.g. definition of release scopes. Refer to SPL.2.

In this context, the notions “functional” and “nonfunctional” are no relevant classification or categorization criteria for requirements.

Reasons:

- A particular requirement may, and on most cases will, contain both functional and non-functional information, and would therefore fall into both categories. See Section 2.1.5.1 for examples.
- Differentiating would not have any implication on how requirements are further processed, i.e. there is no difference in needs for traceability, verification/validation etc.

2.1.6 Base Practices on Consistency and traceability

In the Automotive SPICE consistency and traceability are addressed by a BP in the engineering processes and in the Change Request Management process. Furthermore, consistency is addressed in the Project Management process.

2.1.6.1 Purpose of consistency and traceability

The Information Item 13-51 ‘Consistency Evidence’ is explained as:

- Demonstrating bidirectional traceability between artifacts or information in artifacts, throughout all phases of the life cycle, by e.g.
 - tool links
 - hyperlinks
 - editorial references
 - naming conventions
- Evidence that the content of the referenced or mapped information coheres semantically along the traceability chain, e.g. by
 - performing pair working or group work
 - performing by peers, e.g. spot checks

- maintaining revision histories in documents
- providing change commenting (via e.g. meta-information) of database or repository entries

Experience has shown that it appeared unclear how to ensure consistency without being able to trace the two respective pieces of information (in whatever form). Therefore, these two BPs have been reintegrated into one, which does not invalidate the above-mentioned additional advantages of traceability.

The following figure shows the relationships respectively for traceability and consistency:

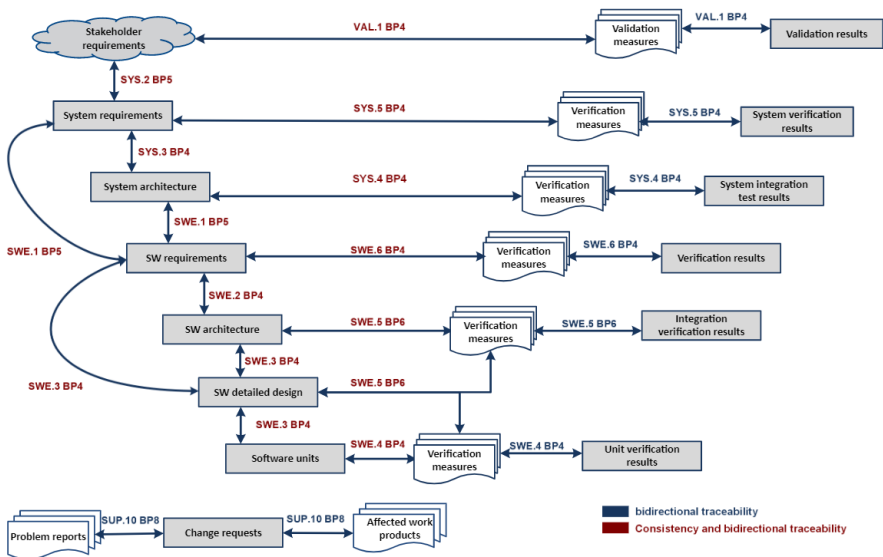


Figure 2-4: Traceability between system and software work products

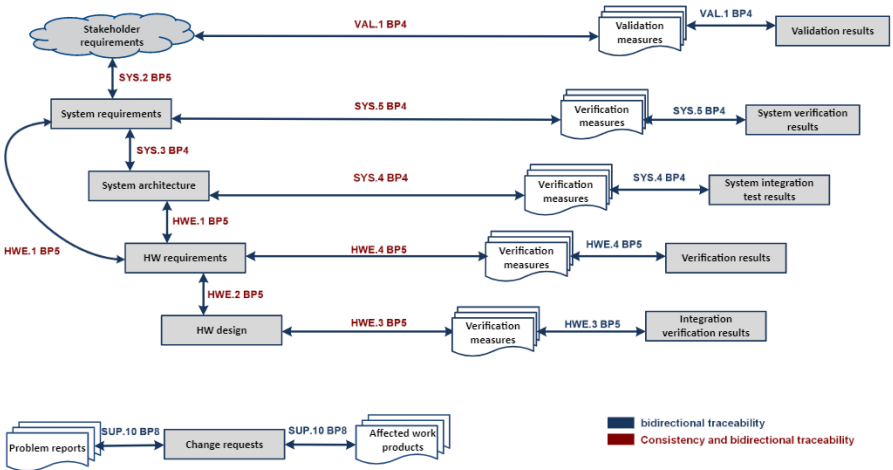


Figure 2-5: Traceability between system and hardware work products

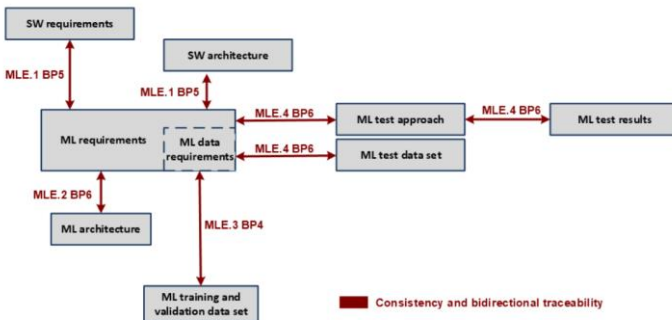


Figure 2-6: Traceability between ML work products

Further advantages of traceability

Traceability between within work products of the same, i.e. within, process is not addressed by the BPs at Capability Level 1. Instead, it may be considered in the context of GP 2.2.2 at Capability Level 2.

In addition, bidirectional traceability further supports:

- analysis of dependencies in both directions

- determination of requirements coverage
- determination of verification coverage
- status tracking of implementation of requirements and verification measures
- impact analysis and risk assessment of change requests on affected work products
- impact analysis and risk assessment for changing technology,
- impact analysis on cost, schedule, effort, and technical impact

Rating Rules

2.1.6.2 Granularity of traceability

The following list defines allowed levels of traceability granularity:

- requirements
 - single requirement
 - cluster of requirements¹
- architecture
 - single architectural element
 - cluster of architectural elements²
 - cluster of software components²
- software detailed design
 - single software Unit
 - cluster of software Units²
- hardware design
 - single HW Part
 - single HW component (i.e. a functionally coherent cluster of HW parts)
 - cluster of HW components
- verification/validation measures
 - single verification/validation measure
 - a cluster of verification measure
- verification results
 - single verification/validation result

- cluster of verification/validation results
- single change request
- single problem record

1) A requirement is to represent an atomic expectation. Therefore, a certain behavior can be represented by a set of requirements).

2) e.g. a UML/SysML sequence diagram depicting several architectural elements in combination

Rating Rules:

[TAC.RL.1] If traceability is distinctly established between clusters of information instead of individual atomic elements, then the 'Consistency and Traceability' BP shall not be downrated.

2.1.6.3 Methodology/approach for traceability

A PAM does not predefine any methodology/approach or tools. The same applies for the realization of traceability. The selected methodology/approach for traceability however need to be appropriate for handling the given complexity, e.g. tool support.

Rating Rules:

[TAC.RL.2] If for documenting traceability an automated tool-based approach is not in favor of manual maintenance of traceability with snapshot-based checks, the 'Consistency and Traceability' BP shall not be downrated.

2.1.6.4 Evidence for consistency

The Automotive SPICE PAM requires *ensuring* consistency but not *reviewing* or *documenting* which means that the exact way this is done cannot be predefined. See also the Information Item 13-51 'Consistency Evidence'

Further, the Automotive SPICE process SUP.8 does not predefine which work products/ artifacts/ documented Information Items are to

be part of baselines. Such decision are is subject to the HOW level (see Automotive SPICE PAM Section 3.3). Therefore, ensuring of consistency between pieces of information is orthogonal to the notion of baselines.

Rating Rules:

[TAC.RL.3] If there is no explicitly documented review record or analysis record proving consistency between related information in favor of approaches such as performing pair working or group work, peer spot checks, maintaining revision histories in documents, or providing change commenting (via e.g. meta-information) of database or repository entries, then the ‘Consistency and Traceability’ BP shall not be downrated.

[TAC.RL.4] If consistency and traceability is established and ensured between information that is not part of baselines, then the ‘Consistency and Traceability’ BP shall not be downrated.

2.1.7 Base Practice “Communicate”

At Capability Level 1 it is only required that agreement and communication is effective. A PAM cannot predefine a particular form. Therefore, the Information Item 13-52 ‘Communication Evidence’ is explained as:

- All forms of interpersonal communication such as
 - e-mails, also automatically generated ones
 - tool-supported workflows
 - podcast
 - blog
 - videos
 - forum
 - live chat
 - wikis
 - meeting, orally or via meeting minutes (e.g. daily standups)

This implies that communication does not need to be represented by baselining in terms of configuration management. Further, following

both a push or pull principle can be acceptable. Furthermore, this means that the sender and receiver do not necessarily need to communicate with each other directly.

Rating rules:

[COM.RL.1] If effective communication of agreed information at Capability Level 1 is not done based on information baselines or by explicitly documented communication or review records then BP “Communicate” shall not be downrated.

Moreover, note that there is no full semantical overlapping with GP 2.1.6 at Capability Level 2.

2.1.8 Verification process oriented concepts

2.1.8.1 “Verification” instead of “testing”

The respective SYS, SWE, and HWE processes have been advanced to address verification (being an umbrella term) instead of testing only.

Reasons:

- Especially at the system and hardware levels, testing is not the only verification approach. Rather, measurements (e.g. geometrical tolerances), calculations or analyses (e.g. strength/stress calculation using an FEM method), or simulations instead of using physical samples are other methods of verification. The same is true for mechanical or hardware development. Therefore, the umbrella term verification now forms the center of those processes' purposes.
- The process SWE.4 'Unit Verification' has already been an exception as a software unit can be verified coherently by means of a combination of static analysis, testing, and code reviews (a view that is also inherent in ISO 26262-6 clause 9).

2.1.8.2 No more use of term “item” in verification processes

In Automotive SPICE 3.1, the term “item” referring to an object-under-test was in conflict with other standards such as ISO 26262

‘Functional safety for Road Vehicles’. This automotive domain-specific Functional Safety standard rather refers to an ‘item’ rather as

- a term representing a technical product. or a distributed functionality. from a logical-functional perspective, irrespective of how many systems will help implementing it (e.g. a new vehicle function such as adaptive cruise control, or a mechatronic vehicle-level system such as an automatic side door access system)
- as the “thing” on which HARA is performed

In order to

- remove conflict with other standards
- to bridge the language of the protagonists of different standards,
- and to enable a better alignment of Automotive SPICE assessments and other types of assessments (e.g. ISO 26262 safety audits) in practice

Automotive SPICE 4.0 abandons the usage of the term ‘item’.

2.1.9 No explicit notion of “specification” and “strategy” at level 1

Today, requirements or verification/validation measures are not necessarily contained in a physical single document but objects or entries in e.g.

- a database
- repositories such as Application Lifecycle Management or Product Lifecycle Management tools.

These entries are usually allocated to releases and products variants, which is meta-information expressed via e.g. attributes. Further, requirements and verification measures for a particular product may come from various sources, e.g. standard product kits or platform documentation and new features for customers. Furthermore, selective baselining is possible for sets of entries in such repositories.

In this PAM this is emphasized by no longer talking about ‘specifications’ in the respective processes but about ‘requirements’ or ‘verification measures’ etc. This is further in line with ISO/IEC

330xx's new notion "Information Items" instead of "Work Product" Indicators.

In addition, this will prevent the assessor from downrating if such information is not represented in one physical document.

Similarly, the former strategy BPs at Capability Level 1 have been removed in favor of reallocating their content to other existing BPs. Further, the former process-specific Work Product Indicators 08-xx with their Work Product Characteristics have been removed in favor of reallocating their content to newly defined Information Items.

Reasons:

1. The extra strategy BP and Plan Work Product Indicators could be misinterpreted in a way that an explicitly written document would be required. In practice, this has resulted in downrating BP1 if such an explicit document is not available. In some contexts this led "over-engineered" processes.
2. In a context where an explicitly written strategy document is necessary, the existence of a "strategy" BP and the "Plan" Work Product Indicator, respectively, could be misinterpreted by requiring exactly one single document, and/or following the same structure as given in the Work Product Characteristics.
3. That 'strategy' BP is the "Plan" Work Product Indicator could be misinterpreted by requiring a more systematic and controlled approach at Capability Level 1 already. This makes the defined semantical distinction between, and the message behind, Capability Levels 1 and 2 become elusive.

As a consequence, assessment results on the same or on very similar contexts sometimes differed very significantly.

2.1.10 No extra BP on evaluating alternative architectures

The former Automotive SPICE v3.1 base practice 'Evaluate alternative architectures' has been revised, and integrated in SYS.2.BP3, SWE.2.BP.3, and HWE.2.BP4. It is now required to

document a rationale for the chosen architecture. Reason: it is considered of higher practical value to provide arguments why a given design was chosen rather than explaining which other particular approaches were not chosen. Further, it can be considered that the former implies the latter.

2.2 Software Unit Behavior and Unit Integration, Component Behavior, and software Component-level testing

Software Unit integration

In the case a software component comprises many software units it may, depending on the context and nature of the software component, be necessary to perform *intra-component* unit integration verification first before the component itself is verified from a black-box perspective. It may have seemed obvious to add to SWE.4 three more BPs on the explicit specification, selection, and performing of software unit integration.

However, there are, also, contexts in which such software unit integration is not applicable, or technically does not have added value. In such contexts, rating such additional BPs as 'F' is considered a falsification of the message behind the rating value 'F', which is the existence of objective evidence (found during the assessment) for an operational workflow with no significant systematic risks. Further, rating such BPs as 'N' would possibly, and unnecessarily, reduce the PA 1.1 rating. Furthermore, acc.to ISO/IEC 33020 assessment indicators (including BPs) generally cannot be considered, or rated as, "not applicable". An alternative might have been to introduce two different integration processes, one for the unit and one for the component levels. However, this would have unnecessarily increased the no. of processes and introduced replication of BPs (e.g. Select..., Communicate... etc.) redundant, both of which was not a goal for Automotive SPICE 4.0.

For these reasons, the decision for Automotive SPICE 4.0 was to express both levels of integration, namely software unit integration and software component integration into the full software, within SWE.5. This was done by SWE.5.BP1 and SWE.5.BP4, respectively, talking about

- “software elements”, which is an umbrella term for software units and software components (see the Automotive SPICE 4.0 glossary)
- “integrating the software elements hierarchically until the software is fully integrated“

This should provide freedom for the assesses to define and explain which elements in their context are to be integrated: software units alone, software components alone, or both.

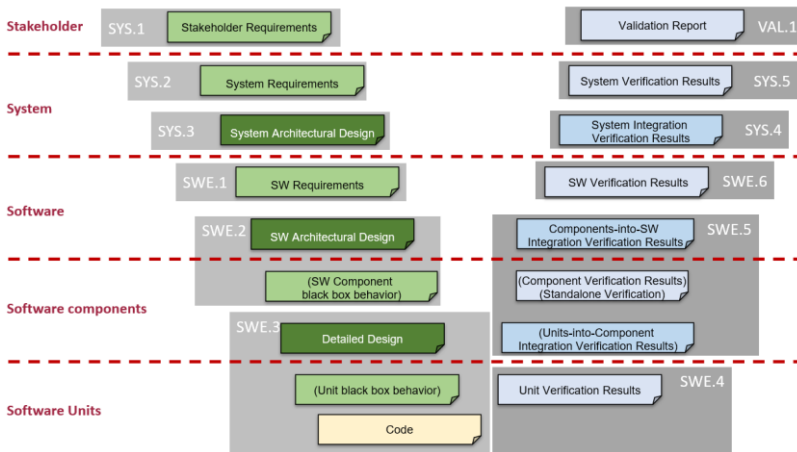
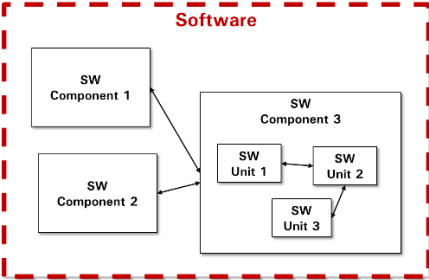


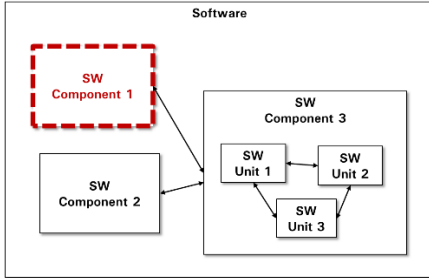
Figure 2-7: 2.2 Software Unit Behavior and Unit Integration, Component Behavior, and software Component-level testing

See also Table 1:

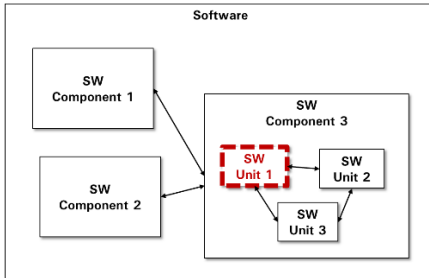
Table 1: Comparison of location of SWE.x concepts in Automotive SPICE versions 3.1 and 4.0

Aspect	As addressed in Automotive SPICE v3.1	As addressed in Automotive SPICE 4.0
<p>Software requirements</p> 	SWE.1	SWE.1
Definition of the behavior of a single software component	<i>not intuitively/adequately addressed</i>	SWE.2

(as opposed to interactions between components)



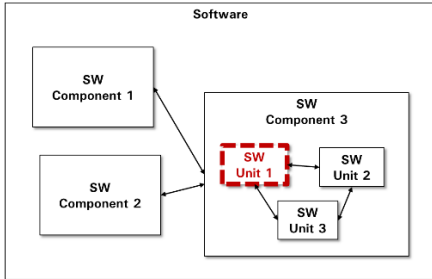
Definition of the behavior of a single software unit



not intuitively/adequately addressed

SWE.3

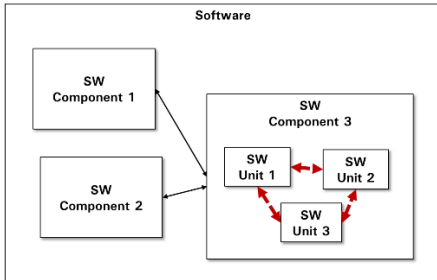
Verification of a single software unit



SWE.4

SWE.4

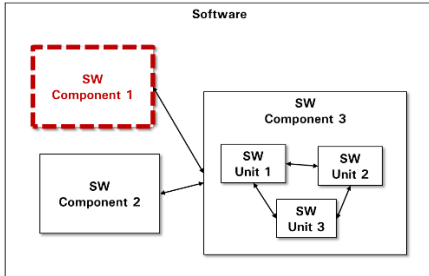
Integration, and integration testing, of software units into their component



not intuitively/adequately addressed

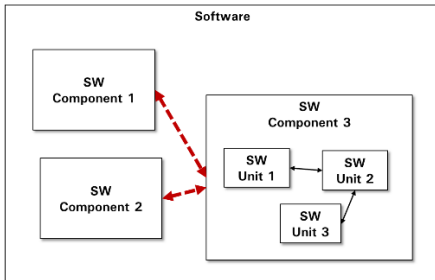
SWE.5

Testing of a single software component
(prior to integration with other
components)



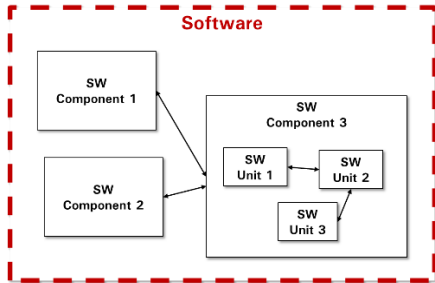
*not
intuitively/adequately
addressed*

Integration and testing of software
components



SWE.5

Testing of the integrated software



SWE.6

SWE.6

Software Component standalone black-box verification

The “next step above” software unit integration is verifying a software component alone from a black-box perspective. However, this was not sufficiently expressed in Automotive SPICE v3.1, see Table 1.

For Automotive SPICE 4.0 the decision was to embed this concept in SWE.5. It was not the decision to

- introduce a distinct process outside SWE.4, SWE.5, and SWE.6. Reason: this would also have unnecessarily increased the no. of processes and introduced replication of BPs (e.g. Select..., Communicate... etc.) redundant, both of which was not a goal.
- add it to SWE.6.

Reason: software component verification happens, from a lifecycle model perspective, after software unit integration but before the integration of all software components into the full software. Adding this concept to SWE.6 was considered to be less intuitive compared to the given solution of embedding it to SWE.5. Also, it would have made necessary traceability between SWE.6 and SWE.2 which, again, was not considered intuitive.

As a trade-off between avoiding mass of processes and BPs and maintaining best possible intuition, the SWE.5 process in Automotive SPICE 4.0 addresses all integration levels (see Figure 2-7). It combines the logical flow of integration of software units into their joint software component à software component standalone verification à integration of all software components into the full software.

2.3 Application in specific environments

2.3.1 Model based development

The approach of model-based development can be used for different purposes within the system and software development. For example, models can support the requirements elicitation process or the development of complex algorithms.

Refer to Section 2.1.6 for the generic concept of consistency and traceability.

2.3.1.1 Models need additional descriptions

Models can be used in different use cases within the development process (e.g. for requirements elicitation, architectural design, detailed design, code generation, verification). It has to be defined and documented what the use case of the model is, e.g. “the system architecture is documented using SysML”.

Modelling notations may be graphical, textual, or a mixture of both and may differ depending on the use case for the model. The syntax and semantics of the notations shall be defined in a formal, semi-formal, or informal way).

Aspects (e.g. design decisions) that the modelling notations cannot express require additional descriptions in natural language (e.g. via text annotations). The corresponding information output characteristics (see Annex in Automotive SPICE PAM) give guidance for the aspects of the additional descriptions.

The following rating rules must be interpreted in the respective context, process, and use case (e.g. if the model is used for software requirement elicitation, the corresponding indicator is SWE.1.BP1, if the model is used for software detailed design, the corresponding indicators are SWE.3.BP1, SWE.3.BP2, SWE.3.BP3).

Rating rules:

[MBD.RL.1] If the syntax and semantics of the model notation are not defined or not appropriate for the use case, then corresponding indicator shall be downrated.

[MBD.RL.2] If the additional description is missing or insufficient the corresponding indicator shall be downrated.

[MBD.RL.3] If the additional description is documented in extra documents but associated with the model, the corresponding indicator shall not be downrated.

2.3.1.2 Consistency of additional descriptions

Aspects that cannot be expressed by the modelling notation might be missing, if not documented in some other appropriate form.

If the model itself is part of a development artifact, e.g. for the use case of requirement elicitation the model is part of the requirement specification, it has to be ensured that this additional description in natural language of the model is considered in the following development process.

Rating rules:

[MBD.RL.4] If the additional description for the model is not considered in downstream processes then the corresponding indicator must be downrated.

2.3.1.3 Models for code generation

If automated code generation is used (a.k.a. graphical programming), then the basis for the code generation is

- inherent in the design or
- derived from the design (then traceability between model and design has to be established).

Commonly, in the software design there is information which is not usable for code generation but is important to convey an understanding of the software. An example is textual annotations to graphical elements.

Unit verification done performed at the model level shall provide evidence for consistency of the software units with the software detailed design and with the software requirements.

Traceability and consistency support the compliance of a model and code part. The consistency of additional descriptions with the model and/or the must be established, e.g. by reviews.

Rating rules:

[MBD.RL.5] If there is no or insufficient evidence for compliance of the auto-generated code generation with the detailed design then SWE.3.BP4 must not be rated higher than P.

NOTE: this will include consistency with the non-functional software requirements by means of consistency and traceability between the detailed design and the software requirements.

[MBD.RL.6] If for autocode generated from the verified model by using a qualified tool chain (and without any further modification after generation) static verification and unit testing is not performed, then SWE.4.BP3 shall not be downrated.

NOTE: Qualified tool chain for the code generation means that there is evidence that the generated code is correct and consistent with the model.

[MBD.RL.7] If autocode is modified after code generation but static verification or unit testing is not performed then SWE.4.BP3 shall be downrated.

2.3.2 Agile environments

Agile software development is based on principles of the Agile Manifesto with the objective to create lightweight development methods. Popular frameworks for agile software development are SCRUM, KANBAN, eXtreme Programming, and SAFe.

Automotive SPICE describes meaningful process principles but does not predefine any concrete lifecycle model, method, tool, templates, metrics, proceedings etc. (the WHAT level). This means the Automotive SPICE content resides at a higher level of abstraction than any process implementation (the HOW level) in order to allow for maximum freedom, and, also, for benchmarking. In contrast, agile methods rather reside at the HOW level. Therefore, Automotive SPICE and agile approaches cannot, by definition, contradict each other. The only valid question would be to ask whether concrete process implementations, following or including agile methods or not, actually satisfy the Automotive SPICE principles. Automotive SPICE does not predefine any type of lifecycle model like V- or Waterfall-model.

Agile Methods may support Automotive SPICE requirements and should be compliant to required rules and standards. For example, non-functional requirements, review and documentation criteria or coding guidelines are valid in an agile and non-agile life cycle.

The rating rules in this chapter are based on practical experience and have no pretention of completeness.

The documented practical experience within this chapter are partly not specific to agile development (e.g. missing software architecture) but have been detected often in Automotive SPICE Assessments of projects with agile development methods.

2.3.2.1 Planning in agile environment

Customer planning requirements are equal in agile and non-agile development. Projects have to ensure that the technical content of features is delivered and bugs are fixed as agreed and scheduled. The planning methods may differ.

Therefore, the agile project has to ensure that the project planning is in line with the customer release planning.

For example, an agile SCRUM project will ensure that the sequence of sprint cycles will deliver the needed functionality corresponding to the customer requirements. I.e. the planning has to ensure that the agreed features are developed and tested within the sprints before the planned release, and the planning has to be consistent across affected parties and agreed plans.

[AGE.RL.1] If evidence from project planning (e.g. backlog, burn down chart and/or sprint planning) show gaps regarding the release planning and this aspect is significant in the context of MAN.3.BP4, MAN.3.BP9, and SPL.2.BP1 then the indicators MAN.3.BP4, MAN.3.BP9, and SPL.2.BP1 shall be downrated.

Additionally, the remaining effort for function development until future deliveries and start of production shall be estimated and covered by available capacity to ensure that additional effort caused by underestimated tasks (e.g. user stories) is not summing up and impacts future project milestones.

[AGE.RL.2] If evidence from project planning is missing that remaining effort for features is not estimated which are to be delivered in future releases then MAN.3.BP5 shall be downrated.

2.3.2.2 Project life cycle

The chosen project life cycle should fit to the project scope, requirements, deliveries, complexity, etc. Therefore, it may be necessary to create a life cycle according to a standard process with tailoring to meet the project needs.

For example, the customer might continuously deliver requirements to the project and expect continuous integration by the project in order to monitor the progress of the product. An agile development process (e.g. SCRUM or Kanban) may support the customer requirements regarding progress monitoring and incremental requirements delivery.

[AGE.RL.3] If the defined project life cycle does not fit to project scope, requirements, deliveries, etc. then MAN.3.BP2 shall be downrated.

2.3.2.3 Management of requirements

In practice, some projects manage the requirements in a change management or tracking tool in which the requirements are managed within tasks or change requests only. These solutions may have the benefit to trace requirements to tasks and code easily but have the disadvantage that no overview of all requirements is established. Without an overview of requirements, the maintenance of requirements is very difficult in regard to impact analysis of changes and getting evidence that all requirements are implemented completely.

For example, a feature has different functions. In development, a first task is issued for development of the feature. During the development period, different change requests/tasks are assigned to the feature and implemented to add, change or delete functions of the feature. At project end the requirements of the feature can only be determined by assessing all tasks of the feature.

2.3.2.4 Risk management

Customers, company or project requirements often require integrating risk management for the development projects, and this risk management needs to be integrated into the agile project.

For example, if the customer requires managing of project and technical risks then the project has to identify, mitigate and manage project risks at project management level and technical risks on requirements and architecture level.

2.3.2.5 Architecture

An architecture has to be defined that identifies the components which are to be traced to the related requirements.

Agile projects have to ensure that an architecture is developed and maintained and that traceability between architecture and requirements, architecture and detailed design, and architecture and integration verification is established.

Example of a proceeding for creation of an architecture within an agile environment can be that basic architecture and architecture rules are defined at project start and the architecture is incrementally

completed within sprints (for SCRUM based projects). For all architectural modifications an impact analysis is performed.

[AGE.RL.4] If the system architecture is modified incrementally including impact analysis then SYS.3.BP1 shall not be downrated.

[AGE.RL.5] If the software architecture is modified incrementally including impact analysis then SWE.2.BP1 shall not be downrated.

2.3.2.6 Verification

Verification of system and software artifacts need to be established in development projects. Agile methods may combine verification levels. The agile project has then to ensure that the process purposes of all relevant verification processes are fulfilled by the defined activities. In such cases the related process areas should not be downrated.

2.3.2.7 Independent quality assurance

Agile development methodologies may define generic role descriptions which need to be derived for the roles and responsibilities in the development project. By defining the responsibilities, the project has to ensure that work product and process quality assurance are performed at project level independently and performed objectively without conflicts of interest.

For example, the agile project ensures the independency by an organization structure in which a quality assurance role is defined to ensure that work products and process quality assurance are checked independently and without conflicts of interest.

2.3.2.8 Pair programming

Agile methods may use pair programming in which two software developers work together at one computer. One writes code while the other reviews each line of code as the other developer types it in. The developers frequently switch roles.

[AGE.RL.6] If the pair programming method is not in conflict with code review requirements (e.g. inspection is required due to safety context) then SUP.1.BP3 and SWE.4.BP3 shall not be downrated.

2.3.3 Development external to the assessed project (DEX)

2.3.3.1 General information

Automotive software based systems are developed as a complex collaboration of system, hardware and software developers that are working in different organizations, organizational entities of these companies and in development sites that can be distributed over different countries. These organizations include the assessed organization (project) and external organizations.

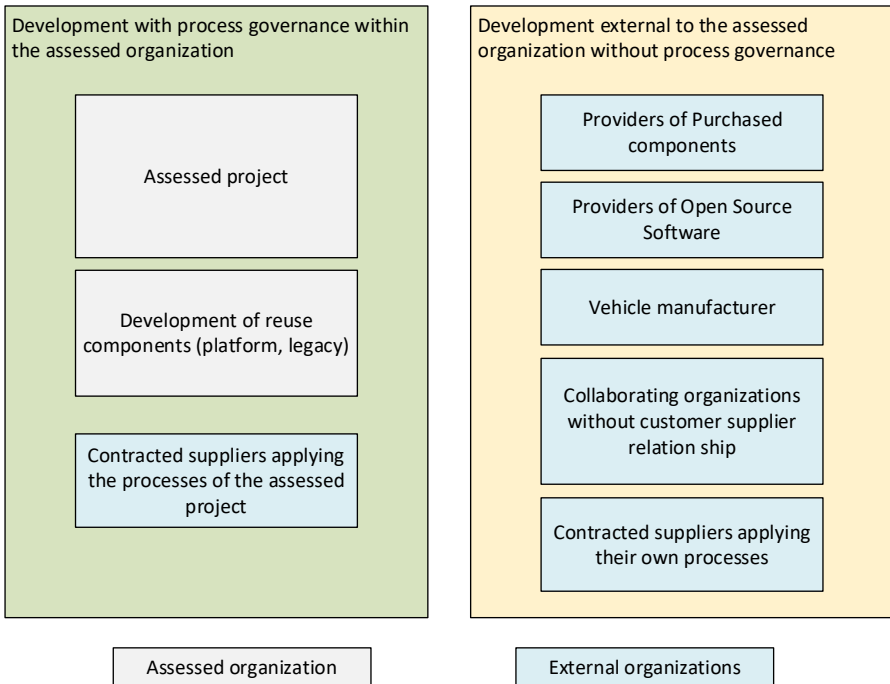


Figure 2-8: Collaborating Entities

External organizations contribute to the product development based on contracts or commitments. Their activities are typically not performed under supervision of the assessed organization.

External organizations include:

- the vehicle manufacturer and its subsidiaries,
- contracted suppliers,

- contracted sub-suppliers,
- contracted collaborating organizations that are not in a customer supplier relationship and
- third party organizations.

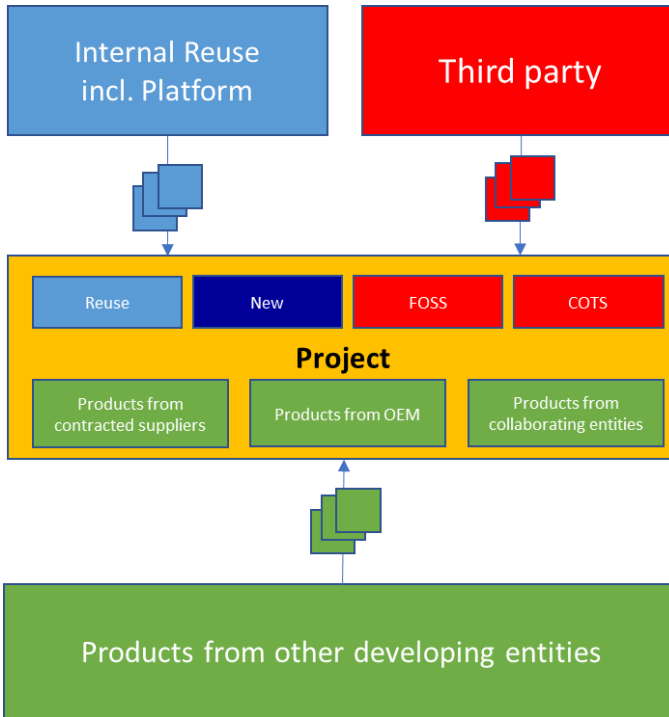


Figure 2-9: Interdependencies of a project integrating products from other parties

In an Automotive SPICE assessment the processes and practices shown in *Table 2* apply to evaluate the processing of software that is developed external to the assessed project. Software that is developed external to the project includes software from collaborating entities, from third party and reused software.

Table 2: Applicable processes for products that are developed external to the assessed project

Business case	Support from originator	To be evaluated in assessed project (if included in assessment scope)					
		MAN.3.BP 7/ GP 2.1.6	SUP.9/ SUP.1	SWE.1/ SWE.2	SWE.5/ SWE.6	REU.2	ACQ.4
Free and Open Source software			X	X	X		
Purchased products (COTS)	X	X		X	X		
Reused products	X	X	X	X	X	X	
Products from vehicle manufacturer or its subsidiaries	X	X			X		
Products from contracted supplier or sub-supplier	X	X		X	X		X
Products from (contracted) collaborating organizations that are not in a customer supplier relationship.	X	X			X		

The Engineering activities of automotive software-based systems within an organization are not necessarily performed at one location. In the context of a project for the development of a particular product the necessary engineering resources, supporting resources and management resources may be distributed across separate departments, locations, buildings, third party service providers etc.

In the planning phase of an assessment the sponsor and the lead assessor have to determine whether locations and departments within an organization will be covered with one assessment or with separate assessments.

If all locations and departments are performing their work based on a standard process, it may be optimal to include them all into the assessment scope. If one location is solely responsible for e.g. software testing the interviews for this process shall be performed with only that location.

When locations or departments have different processes, separate assessments could be performed, or a single assessment may be organized with defined process instances for the processes performed with the same purpose and outcomes (e.g.: project management, quality assurance, configuration management).

2.3.3.2 Maintain effective collaboration

Responsible roles within the project have to maintain an effective collaboration and communication including the definition of a consistent set of responsibilities to achieve the project goals.

Depending on the assessment scope the following aspects have to be evaluated for the interfaces regarding development activities and results that are performed external to assessed project:

- Scope of work for all collaborating entities
- Definition of responsibilities
- Interfaces between overall plans, sub-project plans and plans for support organizations
- Monitoring of agreed commitments
- Communication between all entities
- Compatibility of status models for work products
- Providing necessary work products to collaborating entities
- Preconditions to integrate work products from collaborating entities
- Escalation mechanisms when work product requirements are not met
- Verification and validation measures for the integration of system or software elements that were developed by different collaborating entities.

Based on vehicle manufacturer strategies, the vehicle manufacturer may deliver source or object code to the supplier's software project. I.e. the customer is part of the distributed development.

[DEX.RL.1] If the scope of work is not defined for all collaborating entities, the indicator MAN.3.BP1 must not be rated higher than L.

[DEX.RL.2] If the plans of the overall project and the collaborating entities show inconsistencies and this aspect is significant in the context of MAN.3.BP9, the indicator MAN.3.BP9 shall be downrated.

[DEX.RL.3] If the monitoring of the overall project does not recognize deviations in fulfillment of agreed commitments from the collaborating entities and this aspect is significant in the context of MAN.3.BP7, the indicator MAN.3.BP7 shall be downrated.

[DEX.RL.4] If the information about the properties used for the exchange of configuration items appears to be incompatible, the indicator SUP.8.BP2 shall be downrated.

[DEX.RL.5] If preconditions for work products from collaborating entities to be integrated are missing, the indicator SWE.5.BP4 or SYS.4.BP2 shall be downrated.

[DEX.RL.6] If the supplier project does not comply with the agreements and the agreed rules for the customer-supplied software and this aspect is significant in the context of MAN.3.BP7, the base practice MAN.3.BP7 should be downrated.

[DEX.RL.7] If the vehicle manufacturer does not comply with the agreements and the agreed rules for the supplied customer software, the base practice MAN.3.BP7 should not be downrated but the noncompliance of the customer should be documented in the assessment report.

[DEX.RL.8] If escalation mechanisms across the sub-projects are not defined and this aspect is significant in the context of MAN.3.BP7 or SUP.1.BP7, the indicator MAN.3.BP7 or SUP.1.BP7 shall be downrated.

2.3.3.3 Acceptance of software from collaborating entities

Evidence is needed that software from collaborating entities has been verified according to pass/fail criteria which are defined in validation measures. These acceptance criteria may contain for example the

review of the release documentation, fulfillment of coding guidelines and/or code coverage of manual and automated tests in compliance with the agreed requirements.

For software without any support from a third party provider (e.g. open source software) the project has to define acceptance criteria based on their integration and test strategy.

[DEX.RL.9] If the verification and validation measures for system or software integration do not include the verification and validation of elements that were developed at different collaborating entities and this aspect is significant in the context of SWE.5.BP2 or SYS.4.BP1, the indicators SWE.5.BP2 or SYS.4.BP1 shall be downrated.

[DEX.RL.10] If no pass/fail criteria are defined to check the compliance of third party software and this aspect is significant in the context of SWE.5.BP1 or SWE.5.BP2, the base practices SWE.5.BP3 or SWE.5.BP2 should be downrated.

[DEX.RL.11] If no acceptance tests are performed to check the compliance of third party software according the defined acceptance criteria and this aspect is significant in the context of SWE.5.BP5 and, the base practices SWE.5.BP5 shall be downrated.

2.3.3.4 Functional and non-functional software requirements

The specification or the contractual basis of third party software has to cover functional and non-functional software requirements.

The functional software requirements of the third party software have to be in line with software requirements of the project. In case of “software which is developed by a supplier on basis of project requirements” the project has to transfer these requirements to the supplier and should use the associated tests as acceptance tests.

For “commercial of the shelf software” the project has to ensure that the commercial of the shelf software complies with the requirements specified for the purchased software. The specified requirements should build the basis for acceptance tests of this kind of third party software.

The non-functional requirements include for example quality requirements (e.g. specific coding guidelines, metric targets), which are often used to support the validation process.

In case the third party software is software without any support (e.g. Free and open source software) the project has to ensure that non-functional requirements are met or whether the third party software (e.g. non-automotive commercial of the shelf software) is treated according legacy software rules (see chapter 2.2.5).

[DEX.RL.12] If the software properties of the software from collaborating entities are not in line with the requirements for the project and this aspect is significant in the context of SWE.1.BP5, the indicator SWE.1.BP5 should be downrated.

2.3.3.5 Software architecture

The software from collaborating entities and its interfaces (e.g. external API) have to be part of the software architecture.

For example, a purchased operating system has to be defined in the software architecture together with its interfaces and how the operating system is connected to the relevant software architecture elements.

[DEX.RL.13] If static aspects of software from collaborating entities are not part of the software architecture and this aspect is significant in the context of SWE.2.BP1, the base practice SWE.2.BP1 should be downrated.

[DEX.RL.14] If dynamic aspects of software from collaborating entities are not part of the software architecture and this aspect is significant in the context of SWE.2.BP2, the base practice SWE.2.BP2 should be downrated.

[DEX.RL.15] If the external interfaces of the third party software are not defined in the software architecture and this aspect is significant in the context of SWE.2.BP1, the base practice SWE.2.BP1 should be downrated.

2.3.3.6 Managing of free and open source software

Free Software is source code that allows users to use and modify the software for any purpose. In every case the open source license agreement has to be fulfilled by the project. Otherwise the project does not have the right to integrate and use the open source software

(e.g. open source licenses shall be transferred to customer; open source licenses require to disclose the complete source code of the developed system). Free Software normally has no support, the project has to define and check rules whether the free software elements fit to the project (non-functional) requirements.

Note: Open source software is source code under an open source software license agreement (e.g. GNU General Public License (GPL)).

Because open source software normally has no support, the project has to define and check rules whether the open source software elements and the license fit to the project (non-functional) requirements.

Note: The rules for managing open source software within a company are often called open source Policy.

[DEX.RL.16] If open source software is not managed according to rules, which ensure that the open source software license agreement is fulfilled and this aspect is significant in the context of MAN.3.BP3, the base practices MAN.3.BP3 should be downrated.

2.3.4 Application parameters

2.3.4.1 Interpretation of terms

In the following, the terms “calibration parameters” and “application parameters” are used synonymously.

Automotive SPICE 4.0 defines “application parameters” as follows:

“An application parameter is a solution for a requirements on the configurability of an aspect. As such, an application parameter contains data applied to the system or software functions, behavior or properties. The notion of application parameter is expressed in two ways: firstly, the logical specification (including name, description, unit, value domain or threshold values or characteristic curves, respectively), and, secondly, the actual quantitative data value it receives by means of data application.”

Application parameters can therefore generally be used for two scenarios:

Influencing the implemented system behavior

The software makes the system behave according to the stored application parameter data not containing any executable or interpretable code, e.g.

The range of the window glass in a door system within which antitrap protection shall be active

Values for low idle speed, motor characteristic diagrams etc.

Product vehicle impacting system behavior, e.g. such as country codes, left-hand/right-hand steering etc.

Code selection

Code variants can be determined at compile-time by e.g. preprocessor commands or preprocessor variable settings of e.g. the programming language C; as a result, the built program only contains code that is to be executed. In contrast, the expected executed code can also be determined later, i.e. at runtime, depending on application parameter values evaluated if-clauses.

In both scenarios, the actual data set can be flashed into the system by e.g. diagnosis jobs or end-of-line.

In this document, compile-time variants are not addressed.

2.3.4.2 Application parameters and requirements

In Automotive SPICE the processes SYS.2, SWE.1, and HWE.1 do not explicitly mention application parameters.

Reason: The SYS.2 process is about documenting requirements, i.e. expectations free from design & implementation decisions from a black-box perspective (see also Section 2.1.5). Therefore, SYS.2 will not know whether or not the system is actually going to have software in it. This is a decision made in the context of SYS.3 (see also Automotive SPICE PAM Section 3.4).

What SYS.2, SWE.1, and HWE.1 can require however is ‘configurability’ of a particular aspect.

Simplified example:

- Req #1: “The undervoltage boundary shall be configurable from 0[V] to 3.4[V].”
- Req #2: “When the system detects undervoltage then the system shall shut down in less or equal 500[ms] with a tolerance of +50[ms].”

In contrast, introducing application parameters (including the definition the parameters’ variable names, technical data types, default values etc.) is a software design decision for implementing such configurability requirements. Further, software application parameters are only one out of several possible solutions for implementing a configurability requirement. An alternative implementation solution in hardware for the same requirement would be e.g. e-Fuses.

Consequently, deciding on how many application parameters are to be implemented in the software in order to express this, and on specific logical information (i.e. the parameters’ variable names, technical data types, default values etc.) is a design decision.

Rating Rules:

[APA.RL.1] If with the implemented application parameters and their values in the detailed design are not consistent with configurability requirements, then SYS.3.BP1 and SYS.3.BP2 shall be downrated.

[APA.RL.2] If the detailed design or the implementation does not include checking for allowed value ranges of application parameters, then SWE.3.BP2 or SWE.2.BP3, respectively, shall be downrated.

2.3.4.3 Dependencies between parameters

Application parameters may have complex interdependencies, e.g. a particular parameter A may be exclusive to parameter B and C. Since application parameters are possible software solutions for configurability requirements, such interdependencies represent variants at the requirements level.

Examples:

- A navigational system for customer A additionally offers Points-Of-Interest while the variant for customer B does not;
- a fault diagnosis for a stuck relay is not required for a semiconductor solution of a powerstage, e.g. pulse-width based activation an actuator.

Depending on the complexity, the mastering of such variants at the requirements level can range from labelling requirements by e.g. meta-attributes in tools up to approaches as “feature trees”.

2.3.4.4 Application parameters for code selection at runtime may represent product variants

Application parameters may represent product variants. Therefore, the verification parties should use a product sample that correctly represents the desired variant. Otherwise, verification might fail. This further emphasizes why studying the requirements by the verification personnel is necessary.

There is no extra rating rule here as this is a regular BP rating proceeding.

2.3.4.5 Treating application parameter information as configuration items

For any application parameter representing software decisions at runtime, then the

- a) variable names
- b) the domain value range
- c) technical data types
- d) default values
- e) the corresponding memory maps

are part of configuration items, and subject to baselines.

Rating Rules:

[APA.RL.3] If application parameters including all aspects above are not treated as configuration items, then SUP.8.BP1 shall be downrated.

2.3.4.6 Quality assurance on parameter information

Quality assurance activities must not only include evaluating whether data ranges, default values, and final values are correct, but must also check for consistency of this information across all parameters. Quality assurance must also evaluate whether the chosen data values represent the desired product variants. This is particularly important if different parties are responsible for different application parameters (see chapter “Responsibility for application parameters”)

Example 1:

The customer wants Feature F_1 only. Therefore, it was decided to choose product variant V_2 . However, erroneously both parameters X and Y were activated which results in the product actually realizing F_1 and F_2 , i.e. Variant V_1 . This error should have been detected by e.g. design or code reviews against the table.

	Variant V ₁	Variant V ₂	Variant V ₃
Feature F ₁ , activated by parameter X	x	x	-
Feature F ₂ , activated by parameter Y	x	-	-

Example 2:

The customer wants features F₁ and F₂ only. Therefore, it was decided to choose variant V₁. Correspondingly, parameters X and Y were set. However, during requirements reviews, design reviews, and code reviews it remained unnoticed that parameter Y also activates feature F₃ which was never wanted.

	Variant V ₁	Variant V ₂	Variant V ₃
Feature F ₁ , activated by parameter X	x	x	-
Feature F ₂ , activated by parameter Y	x	-	-
Feature F ₃ , also activated by parameter Y	-	x	-

Rating Rules:

[APA.RL.4] If application parameters do not receive quality assurance with respect to technical correctness, product variant consistency, then BP2 of SUP.1 shall be downrated.

2.3.4.7 Change management related for application parameters

Furthermore, in the context of change request management (SUP.10) the impact of a change on application parameter information must explicitly be analyzed. For

- application parameters for code selection at runtime this means activating or deactivating features, and, thus, changing product variants;
- application parameters influencing the implemented system's behavior this means changing the product application.

Rating Rules:

None.

2.3.4.8 Application parameters and testing

Verification personnel will know about the configurability of undervoltage as the verification measures are to be consistently traced to the requirements (SYS.2, SWE.1).

Secondly, to prove configurability, the verification personnel will need to be able to modify application parameters. This is ensured by SYS.5.BP1 aspects a) to e) which require

- a) 'techniques': e.g. equivalence classes and boundary values for the undervoltage example
- c) 'entry criteria': the availability of e.g. extra parameter files to be provided by e.g. the software department
- d) 'Infrastructure/ environment setup': alternatively, the testing personnel may use a flash adapter or a calibration tool together with an e.g. *.a2l file (representing a parameter-address mapping) to be able to modify the parameters themselves.

The fact that, in practice, the verification personnel may of course be supported, or advised, by a requirements or software engineer here

does not change the fact that the above information is a verification concern (SYS.4, SYS.5) rather than a requirements concern (SYS.2). Recall here that a PRM and PAM do not represent lifecycle models (see Automotive SPICE 4.0 Section 3.4).

Rating rules:

[APA.RL.5] If samples that are used to perform verification measures on do not reflect the correct application parameter settings, then BPs on “Verify...” or “perform verification” in SWE.4, SWE.5, SWE.6, SYS.4, or SYS.5, respectively, shall be downrated.

2.3.4.9 Responsibility for application parameters

Application parameters for code selection at runtime

The responsibility of such application parameters for code selection at runtime (see above) is upon the supplier. Therefore, they must not be altered by the customer, so no application parameter information is exposed.

Parameters for influencing the implemented system behavior

Often the division of responsibility for application parameters does not follow the exact customer-supplier boundary.

Examples:

- A controller device supplier defines, and implements, all application parameters but the customer retains the right to alter some of them after the supplier’s delivery
- Owners of different reusable standard software components maintain their own local parameters

Some of the parameters shall not even be accessible to the customer. In such a situation, for e.g. product liability purposes, the responsibility for each of the application parameters should be explicitly defined. This may be done by e.g. an addendum to a development agreement interface.

Rating Rules:

[APA.RL.6] If application parameter values can be, or are, altered by a party at the product level by any other party than the developers of the product, but responsibilities are not clearly defined, then MAN.3.BP7 shall be downrated.

3 Rating guidelines on process performance (level 1)

3.1 ACQ.4 Supplier Monitoring

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

3.1.1 General Information

The customer or the supplier when acting as a customer for its own suppliers has to introduce a supplier monitoring process for the following relationships with external contract-based suppliers:

- Supplier develops a component on basis of the customer requirements
- Supplier delivers and maintains a component which is provided off the shelf to the customer (e.g. operating system, device drivers, system with hard- and software)
- Supplier delivers a component with off the shelf sub-components and development on basis of customer requirements

Interfaces between supplier and customer have to be established for exchanging, monitoring and tracking all relevant information between both parties. Even for a small number of deliveries (e.g. commercial off the shelf component) interfaces have to be set up and maintained for at least component deliveries and managing changes and problem reports.

3.1.1.1 Monitoring all contract-based suppliers

All project relevant contract-based suppliers have to be tracked and their performance against the agreed requirements has to be assessed. Based on the context of the project this may include suppliers for engineering service, commercial of the shelf products,

firmware, etc. Excluded are suppliers which deliver products without any support (e.g. open source software).

3.1.1.2 Incomplete agreements with supplier

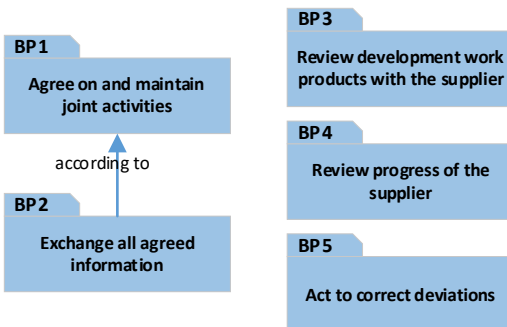
Agreements between supplier and customer have to be established and maintained, which cover:

- supplier’s project content and scope
- relevant requirements and standards from the customers customer
- exchanged information between customer and supplier
- joint activities and interfaces
- responsibilities and stakeholders
- joint problem and change management
- joint reporting and reviews
- escalation mechanism

Examples for such agreed documents are distributed interface agreements, statements of work, license agreements, etc.

3.1.2 Rating Rules within the process

The following figure shows the relationships between ACQ.4 base practices as well as their relationships to other processes:



These relationships are used as the basis for the rating rules defined in the following.

[ACQ.4.RL.1] If the indicator BP1 is downrated due to incomplete agreements about exchanged information between customer and supplier, the corresponding indicator BP2 shall be downrated.

3.1.3 Rating rules with other processes

None.

3.2 SPL.2 Product Release

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

3.2.1 General Information

In the course of a product development the functional content that is agreed with the customer or a development partner is usually implemented in an incremental way. The prioritization of the functions to be realized is done in the Requirements analysis processes SYS.2, SWE.1, HWE.1 and MLE.1.

3.2.2 Rating rules within the process

3.2.2.1 Release scope

The sequence of implementing these functionalities is substantiated in the release scope. The release scope is not necessarily a separate document. The relevant planning aspects can be part of the project's schedule.

[SPL.2.RL.1] If the scope of the current release is not identified in detail (features and/or functions per release), the indicators BP1 must not be rated higher than P.

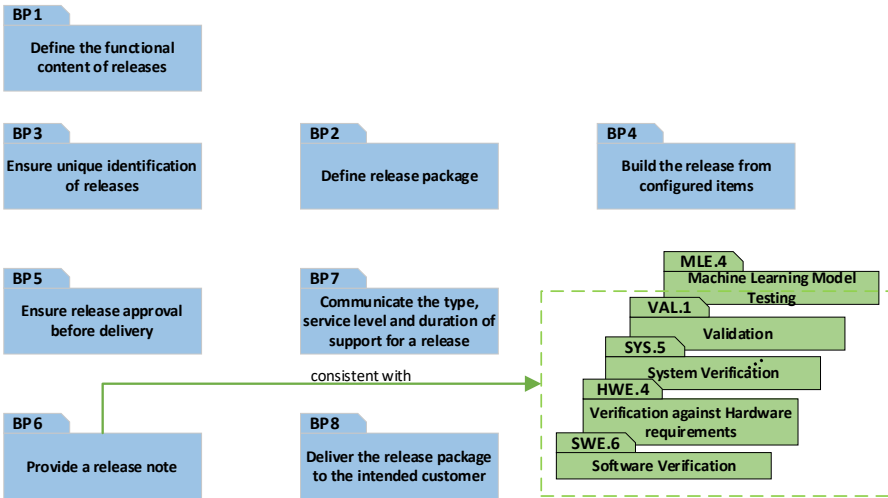
A release package consists usually of the released product, the information about the product and the release, and supporting tools as needed.

3.2.2.2 Release note

Changes and improvements that are made to the content of the delivered product compared to previous releases shall be documented in the release note.

[SPL.2.RL.2] If the release notes do not describe changes compared to previous releases, BP.6 shall be downrated.

3.2.3 Rating rules with other processes



The information regarding verification and validation results of the product has to be considered.

[SPL.2.RL.3] The release including the release notes shall be consistent with the results from VAL.1, SYS.4., SYS.5, SWE.4, SWE.5 and SWE.6. If there is any inconsistency, PA1.1 must not be rated higher than P.

3.3 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.*
-

3.3.1 General Information

The requirements elicited in the context of SYS.1 may span over a no. of different levels of abstraction (e.g. system, software, hardware). It may also contain design constraints and other general expectations.

This leads to two different conclusions:

Conclusion A: vertical tracing

Any requirement is to be verified or validated and, therefore, to be traced to verification or validation measures, respectively (“horizontal traceability”).

Some requirements do not address, or represent, direct properties or characteristics of the physical end product. These do not need to be traced to system requirements, software requirements, or hardware requirements, respectively (“vertical traceability”). Examples see Table 3.

Only those requirements which address, or represent, direct properties or characteristics of the physical end product are subject to vertical traceability. Their identification can be documented via e.g. chapter structuring or tool-based attributes. Examples see also Table 3.

Table 3 – Non-exhaustive examples for conclusion A

	Vertical tracing?
Requirements for work products/artifacts, e.g. <ul style="list-style-type: none"> • MISRA rules • Coding guidelines • Code metrics 	Not traced, but to be evidenced by verification results
Process requirements e.g. <ul style="list-style-type: none"> • Level 2 process capability according to Automotive SPICE 	Not traced, but to be evidenced by assessment reports
Direct functional end product characteristics, e.g. <ul style="list-style-type: none"> • CAN matrix • Behavior 	To be traced and evidenced by verification/validation results
Direct nonfunctional end product characteristics, e.g. <ul style="list-style-type: none"> • weight • response times 	To be traced and evidenced by verification/validation results

Production requirements such as <ul style="list-style-type: none"> • soldering process • Capability of inspection equipment 	Not traced, but to be evidenced by verification results
---	---

Conclusion B: direct or indirect tracing

The stakeholder requirements may include sub-domain requirements or design constraints (e.g. software hardware) which, clearly, do not affect the system requirements (SYS.2) or the system architecture (SYS.3). In such a case, the sub-domain requirement (SWE.1/HWE.1) may be traced directly to SYS.1. However, this must be agreed on by the sub-domain and system representatives.

3.3.2 Rating Rules within the process

Rules for rating consistency between the BPs in this process are not defined. This is due to the nature of BPs as describing separate concerns which shall be addressed individually. Further, a Process Attribute shall be rated based on the Process Performance Indicators, i.e. not based on a subset. If an assessment context-sensitive dependency is identified by the assessor, then he may rate correspondingly but shall provide comprehensive arguments for that in the Assessment Report.

3.3.3 Rating Rules with other processes at level 1

None.

3.4 SYS.2 System Requirements Analysis

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

3.4.1 General Information

Stakeholder requirements can be in contradiction to each other e.g., legal regulations with specific customer needs. System Requirements will be in such a case derived as a trade-off between such stakeholder requirements in dialog with the customer.

3.4.1.1 Iterative vs. incremental development

Normally the functional content in the product changes iteratively and incrementally evolves across releases. The term “increment” can be understood as adding a feature or element that did not exist before (analogy: building a house). The term “iteration” can be understood as refining, or adapting, an existing feature or element (analogy: a sculptor working on a sculpture).

Therefore, the complete set of requirements of the final end product does not necessarily have to be available at the project start. Rather, release scopes agreed with the customer will define increments and iterative rework. In this respect, requirements creation can be driven by release definitions over time.

3.4.1.2 Analysis of impact on the System Context

SYS.2.BP1 specifies the requirements for the system under consideration alone, i.e. the ones the system shall implement. In contrast, BP.4 asks for the impact and consequences the system has on its system context because of those requirements. In Requirements Engineering the term “system context” is a defined technical term. Its meaning denotes anything outside, i.e. beyond, the boundary of the system under consideration in SYS.2. Elements in the system context such as

- human users
- other mechatronic systems
- other controller devices

trigger the system's functionalities, are receivers and users of results of the system's functionalities, interact with the system, or have interfaces with the system. Note that "interface" here may not only refer to direct interaction interfaces but also to indirect ones. For example, another system installed in very close proximity of the system under consideration may suffer from its heat or radiation emission.

In alignment with Section 2.1.4 examples are:

- Vehicle
Noise, exhaust, leakage (e.g. fuel, oil, water, gas, refrigerants...)
- Infotainment
Stress, distraction, discomfort or fatigue as a result of poorly designed or over-designed HMIs.
- Mechatronic system
Vibration, acoustics, forces (e.g. tailgate, automatic door access system), leakage (oil, refrigerant...), stored energy (e.g. pre-loaded springs), moving or rotating elements, kinetic energy, electrostatic and electromagnetic phenomena, electrically live parts, debris of worn parts etc.
- ECU
Signal quality, emission of heat or radiation, size being in conflict with the designed mounting space, weight being in conflict with connection technology used in the system context

Such impact on the system context needs to be communicated back to the owners of the respective elements in the system context in order for them to make changes. Otherwise, this impact may be used to iterate the requirements of the system under consideration.

Note that for SWE.1 and HWE.1 the decision was to keep the term operating environment for two reasons:

1. The usage of the term "system" might not appear intuitive for processes that deal with software and hardware only
2. software runs on a target which is better expressed by using "operating environment"

3.4.1.3 Structuring of requirements

A possible approach to prioritizing requirements is the allocation of requirements to releases. The usage of such an approach will imply that the content of the next and future releases is supported.

3.4.1.4 Traceability and consistency

See also Section 3.3.1 of SYS.1 and Section 2.1.6 here.

3.4.2 Rating Rules within the process

3.4.2.1 System requirements

[SYS.2.RL.1] If different approaches of documenting requirements are used concurrently (e.g. Word processor file, Application Lifecycle Management tool, database) then SYS.2.BP1 shall not be downrated.

[SYS.2.RL.2] If not all system requirements are derived from, and traced to, the customer requirements but to internal standard requirements or to a product line/platform according to a reuse or application strategy, then SYS.2.BP1 and SYS.2.BP5 shall not be downrated.

[SYS.2.RL.3] If not all system requirements of the final product are available at a given point in time because of release-driven incremental development, then SYS.2.BP1 and SYS.2.BP2 shall not be downrated.

3.4.2.2 Structuring of requirements

To support the understanding in Section 2.1.5.3:

[SYS.2.RL.4] If the notions of “functional” and “non-functional” are the only requirements structuring, categorization, or classification criterion, then SYS.2.BP2 shall be rated as N.

[SYS.2.RL.5] If the notions “functional” and “non-functional” are not used as a structuring, categorization, or classification criterion, then SYS.2.BP2 shall not be downrated.

3.4.2.3 Requirements mapping to releases

A possible approach to prioritizing requirements is the allocation of requirements to releases. The usage of such an approach will imply that the content of the next and future releases is supported.

Rating Rules:

[SYS.1.RL.6] If there is no evidence for prioritization but a separate release plan consistently mapping software functionality to future releases then SYS.2.BP2 shall not be downrated.

3.4.2.4 Analysis of requirements

The indicator SYS.2.BP3 requires “*Analyze system requirements. ...and to support project management regarding project estimates*”. This means for example:

- A set of 100 requirements exists. An analysis was done together with the project manager during a project progress meeting. As a result, 20 out of the 100 requirements were decided not to be used, therefore being attributed as “rejected” with an accompanying comment providing expectations.
- A set of 10 requirements were planned for the next release. The development team reports to the project manager that this is no longer feasible due to resource constraints. The decision is to not change the status of those 10 requirements but to reallocate them to future releases. This can be evidenced by a comparison of the release plans (which is the process context of MAN.3 but not SYS.2).

Analysis of requirements can be done by means of using by e.g. tool-based attributes, or comments added to the requirements text.

The analysis of system requirements is the basis for a correct implementation. Even though requirements sometimes appear very simple, a well-founded analysis has to be conducted for those requirements. The scope and appropriateness of the analysis depends on the context of product (e.g., platform). The results of analysis can vary from a simple attribute to a complex simulation or the building of a demonstrator to evaluate the feasibility of software requirements.

Rating Rules:

[SYS.2.RL.7] If analysis results of requirements are not demonstrated by means of separate analysis reports or review records but by means of e.g. tool-supported attributes or tool-supported commenting, then SYS.2.BP3 shall not be downrated

[SYS.2.RL.8] If the analysis of requirements is not evidenced by separate review records, then SYS.2.BP3 shall not be downrated.

[SYS.2.RL.9] If requirements are prioritized by means of a separate project release plan assigning system requirements to releases, then SYS.2.BP3 shall not be downrated.

[SYS.2.RL.10] If the analysis of hardware requirements in regards to technical feasibility is covered by risk management then SYS.2.BP3 shall not be downrated.

[SYS.2.RL.11] If analysis results of hardware requirements in regard to impact on estimates is not consistently used by project management (MAN.3) then SYS.2.BP3 shall not be downrated

3.4.2.5 Traceability and consistency

System requirements are derived from stakeholder requirements. During the process of analysis of system requirements inconsistencies between stakeholder requirements and system requirements may occur as the customers does not always update their requirements.

[SYS.2.RL.12] If a system requirement is no longer consistent with a stakeholder requirement because of a meaningful adaptation, but the stakeholder do not adapt their respective requirement correspondingly and evidence of the alignment is available then SYS.2.BP5 shall not be downrated.

See also Section 3.3.1 of SYS.1 and Section 2.1.6 here.

3.4.3 Rating Rules with other processes at level 1

None.

3.5 SYS.3 System Architectural Design

The purpose is to establish an analyzed system architecture consistent with the system requirements.

3.5.1 General Information

The architecture is also nonfunctional requirements-driven, and system architectural design decisions may lead to iterative system requirements rework. This can be the case if e.g. two nonfunctional requirements cannot technically be realized as expected. Example: a signal is expected to be processed and the expected response time is 2[ms] while 1000 [bus message per minute] shall be able to be processed.

3.5.1.1 Specifying a system architecture

The system architectural design is the highest level of a design description of the system, potentially with different. These views are architecture visualizations that are required for communication, discussion, reviews, analysis, evaluation, planning, change request analysis, impact analysis, maintenance etc. of the system.

There is no common definition of which views are required and no criteria for the completeness such views. Essential views however are a static view providing an overview of the structure and a dynamic view describing the designated behavior behind system functionalities. In most cases the system architectural design is a graphical representation of the system supplemented by textual explanations.

Static system architecture views allow the recursive decomposition of the system into manageable elements with high cohesion and low coupling. This decomposition supports the assignment of requirements to these architecture elements and will help the organization to distribute the work. An architectural may need to include elements that are developed externally, e.g. platform, third-party parts, COTS etc.

At the stage of system architectural design, the allocation typically is done on the level of suitable requirement clusters (e.g. a chapter in

requirements specification) and not on the level of single requirements.

3.5.1.2 A single BP for designing an architecture

The former BPs 1,2 and 3 of SYS.3 and SWE.2 in Automotive SPICE v3.1 have been integrated into one.

Reasons:

The former BP1 required creating an “architecture”. The three pillars on which any architecture resides are

1. a structural view with interdependencies and behavioral descriptions of elements
2. interfaces
3. dynamic behavior and interactions

Having separate BPs for (2.) and (3.) renders a BP that talks about the entire on “architecture” questionable. For the same reasons, a rating of BP1 as F while rating BP3 (in case of inadequate or incomplete dynamic modelling) was rated lower does not support the understanding of having a “full” architectural design.

3.5.2 Rating Rules within the process

3.5.2.1 Analyzing the system architecture

The following BP has been introduced in SYS.3 (and similarly in SWE.2)

SYS.3.BP3: Analyze system architecture. 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 hardware elements. Document a rationale for the system architectural design decision.

in order to be able to reflect e.g.

- Cybersecurity, such as vulnerability analyses
- Functional Safety, such as Safety Analyses and Dependent Failure Analyses acc. to ISO 26262
- robustness needs for non-safety and non-cybersecurity products

Rating Rules:

[SYS.3.RL.1] If non-quantitative analysis approaches or techniques in favor of qualitative ones are used, then SYS.3.BP2 shall not be downrated.

3.5.2.2 Traceability and consistency

See also Section 3.3.1 of SYS.1 and Section 2.1.6 here.

3.5.3 Rating Rules with other processes at level 1

None.

3.6 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.

3.6.1 General Information

3.6.1.1 Why no “production data compliant sample” BP in SYS.4/ SYS.5

The processes HWE.3 and HWE.4 include such a BP because hardware production data compliance does not automatically imply design compliance. In contrast, SYS.4 and SYS.5 do not have such a base practice as there is no notion of ‘system production data’ as opposed to system design. As, further, the process purposes of SYS.4 and SYS.5 are about evidencing that a physical sample is compliant with the design and requirements, respectively, such a single BP would be entirely redundant with these purposes – it would render all other BPs in SYS.4 and SYS.5 useless.

3.6.1.2 Specify verification measures for system integration

SYS.4.BP1 requires the identification of the necessary verification infrastructure and environment setup. This may be supported using simulation of the environment such as hardware-in-the-Loop simulation, vehicle network simulations, digital mock-up).

Verification results can support the update of simulation models.

3.6.1.3 Selecting verification measures

According to BP2, during the selection of verification measures is supposed the is to be considered. Still, this could be achieved by e.g.

- a document depicting the release context further states which verification measures are to be done
- via a meeting with the verification personnel and e.g. a person responsible for the system design

3.6.2 Rating Rules within the process

3.6.2.1 Verification measure definition

Rating Rules:

[SYS.4.RL.1] If entry/exit criteria are reasonably specified for a set of verification measures instead of each individual verification measure, then SYS.4.BP1 shall not be downrated.

3.6.2.2 Automation of verification measures

Rating Rules:

[SYS.4.RL.2] If a verification measure is automated and the correctness, completeness, and consistency of the corresponding scripts and programs are not addressed in the verification measure definition, then SYS.4.BP1 must be downrated.

3.6.2.3 Explorative testing vs. traceability/consistency

The testing state-of-the-art not only comprises testing derived from requirements but also explorative testing based on experience, such as “error guessing based on knowledge”. This is valuable as it adds to the quality of the product. Therefore, explorative tests that are based on experience cannot, by definition, be traced or consistent with the software requirements.

Still, traceability is needed between explorative test cases and their results.

Rating Rules:

[SYS.4.RL.3] If explorative tests are defined as verification measures, then SYS.4.BP4 shall not be downrated.

3.6.2.4 Traceability and consistency

See also Section 3.3.1 of SYS.1 and Section 2.1.6 here.

3.6.3 Rating Rules with other processes at level 1

3.6.3.1 Verification measures selection vs. release plans

Rating Rules:

[SYS.4.RL.4] If selection of verification measures is properly done but based on an inadequate or incomplete release plan, then SYS.4.BP2 shall not be downrated.

3.7 SYS.5 System Verification

- *The purpose is to ensure that the system is verified to provide evidence for compliance with the system requirements using verification measures consistent with the system requirements.*
-

3.7.1 General Information

3.7.1.1 Why no “production data compliant sample” BP in SYS.4/ SYS.5

See Section 3.6.1.

3.7.1.2 Specify verification measures for system integration

SYS.5.BP1 requires the identification of the necessary verification infrastructure and environment setup. This may be supported using simulation of the environment such as hardware-in-the-Loop simulation, vehicle network simulations, digital mock-up).

Verification results can support the update of simulation models.

3.7.2 Rating Rules within the process

3.7.2.1 Verification measure definition

[SYS.5.RL.1] If entry/exit criteria are reasonably specified for a set of verification measures instead of each individual verification measure, then SYS.5.BP1 shall not be downrated.

3.7.2.2 Automation of verification measures

Rating Rules:

[SYS.5.RL.2] If a verification measure is automated and the correctness, complete-ness, and consistency of the corresponding scripts and programs are not addressed in the verification measure definition, then SYS.5.BP1 must be downrated.

3.7.2.3 Explorative testing vs. traceability/consistency

The testing state-of-the-art not only comprises testing derived from requirements but also explorative testing based on experience, such

as “error guessing based on knowledge”. This is valuable as it adds to the quality of the product. Therefore, explorative tests that are based on experience cannot, by definition, be traced or consistent with the software requirements.

Still, traceability is needed between explorative test cases and their results.

Rating Rules:

[SYS.5.RL.3] If explorative tests are defined as verification measures, then SYS.5.BP4 shall not be downrated.

3.7.2.4 Traceability and consistency

See also Section 3.3.1 of SYS.1 and Section 2.1.6 here.

3.7.3 Rating Rules with other processes at level 1

3.7.3.1 Verification measures selection vs. release plans

Rating Rules:

[SYS.5.RL.4] If selection of verification measures is properly done but based on an inadequate or incomplete release plan, then SYS.5.BP2 shall not be downrated.

3.8 SWE.1 Software Requirements Analysis

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

3.8.1 General Information

3.8.1.1 Iterative vs. incremental development

Normally the functional content in the product changes iteratively and incrementally evolves across releases. The term “increment” can be understood as adding a feature or element that did not exist before (analogy: building a house). The term “iteration” can be understood as refining, or adapting, an existing feature or element (analogy: a sculptor working on a sculpture).

Therefore, the complete set of requirements of the final end product does not necessarily have to be available at the project start. Rather, release scopes agreed with the customer will define increments and iterative rework. In this respect, requirements creation can be driven by release definitions over time.

3.8.1.2 Impact on the operating environment

SWE.1.BP1 specifies the requirements for the software under consideration alone, i.e. the ones the software shall implement. In contrast, BP4 asks for the impact and consequences the system has on its operating environment because of those requirements. Its meaning denotes anything outside, i.e. beyond, the boundary of the software under consideration in SWE.1. Elements in the operating environment such as

- human users e.g. in case of infotainment systems
- the target on which the software is running
- stress, distraction, discomfort or fatigue as a result of poorly designed or over-designed HMIs.

Such impact on the operating environment needs to be communicated back in order to be able to make changes. Otherwise, this impact may be used to iterate the requirements of the software under consideration.

3.8.2 Rating Rules within the process

3.8.2.1 Software development without system requirements

In case of software development only, the software requirements may refer directly to the stakeholder requirements. Consequently, consistency and bidirectional traceability have to be ensured between stakeholder requirements and software requirements.

Rating Rules:

[SWE.1.RL.1] In the case of software development only, if the traceability and consistency from software requirements to stakeholder requirements is established then SWE.1.BP5 shall not be downrated.

[SWE.1.RL.2] If software requirements are not derived from system requirements but from platform requirements according to a reuse strategy, then SWE.1.BP1 shall not be downrated.

3.8.2.2 Structuring of requirements

Software requirements can be grouped or categorized to support an overview and prioritization. See also Section 2.1.5.2 here.

Rating Rules:

[SWE.1.RL.3] If “functional” and “non-functional” are the only requirements categorization or classification criterion, then SWE.2.BP2 shall be rated as N.

[SWE.1.RL.4] If there is no evidence for prioritization other than a release planning mapping the functionality to future releases, then SWE.1.BP2 shall not be downrated.

3.8.2.3 Analysis of requirements

See also Section 3.3.1 of SYS.1.

The indicator SWE.1.BP3 requires

“Analyze software requirements. ...and to support project management regarding project estimates”. This means for example:

- A set of 100 requirements exists. An analysis was done together with the project manager during a project progress meeting. As a result, 20 out of the 100 requirements were decided not to be

used, therefore being attributed as “rejected” with an accompanying comment providing expectations.

- A set of 10 requirements were planned for the next release. The development team reports to the project manager that this is no longer feasible due to resource constraints. The decision is to not change the status of those 10 requirements but to reallocate them to future releases. This can be evidenced by a comparison of the release plans (which is the process context of MAN.3 but not SWE.1).

Analysis of requirements can be done by means of using by e.g. tool-based attributes, or comments added to the requirements text.

The analysis of software requirements is the basis for a correct implementation. Even though requirements sometimes appear very simple, a well-founded analysis has to be conducted for those requirements. The scope and appropriateness of the analysis depends on the context of product (e.g., platform). The result of analysis can vary from a simple attribute to a complex simulation or the building of a demonstrator to evaluate the feasibility of software requirements.

Rating Rules:

[SWE.1.RL.5] If analysis results of requirements are not demonstrated by means of separate analysis reports or review records but by means of e.g. tool-supported attributes or tool-supported commenting, then SWE.1.BP3 shall not be downrated.

[SWE.1.RL.6] If the analysis of requirements is not evidenced by separate review records then SWE.1.BP3 shall not be downrated.

[SWE.1.RL.7] If requirements are prioritized by means of a separate project release plan assigning system requirements to releases, then SWE.1.BP3 shall not be downrated.

[SWE.1.RL.8] If the analysis of hardware requirements in regards to technical feasibility is covered by risk management then SWE.1BP3 shall not be downrated.

[SWE.1.RL.9] If analysis results of hardware requirements in regards to impact on estimates is not consistently used by project management then SWE.1.BP3 shall not be downrated.

3.8.2.4 No traceability redundancy

SWE.1.BP5 offers the possibility of having two paths for traceability:

- a) between software requirements and the system architecture (SYS.3)
- b) between software requirements and system requirements (SYS.2)

However, redundancy, i.e. using the two traceability paths for the very same software requirement at the same time, is neither intended by this BP nor meaningful. Further, it is not intended to express that all, or the majority of the, software requirements should be traced to system requirements directly as a default. Which path appears more appropriate must depend on the actual content of the software requirement itself.

Example 1: system requirements ↔ system architecture ↔ software requirements

Consider the system requirements demanding a particular and coherent system service. As an architectural solution, different parts of software run on different microcontrollers or (maybe including dual core microcontrollers) on e.g. different PCBs. Traceability between software requirements and system architecture would be needed here.

- in order to allocate different software behavior to the different microcontrollers
- as there are different communication mechanisms in between the different pieces of software.

Example 2: system requirements ↔ software requirements

The system interface requirements define a particular CAN matrix to be used. Such requirements can be traced to corresponding software requirements directly.

Rating Rules:

[SWE.1.RL.10] If traceability is established for one path only but not for the other redundant path, SWE.1.BP5 shall not be downrated.

3.8.2.5 Requirements mapping to releases

A possible approach to prioritizing requirements is the allocation of requirements to releases. The usage of such an approach will imply that the content of the next and future releases is supported.

Rating Rules:

[SWE.1.RL.11] If there is no evidence for prioritization but a separate release plan consistently mapping software functionality to future releases then SWE.1.BP2 shall not be downrated.

3.8.2.6 Traceability and consistency

See also Section 3.3.1 of SYS.1 and Section 2.1.6 here.

3.8.3 Rating Rules with other processes at level 1

3.8.3.1 PA 1.1 of SWE.1 vs. other processes

None.

3.9 SWE.2 Software Architectural Design

-
- *The purpose is to establish an analyzed software architecture consistent with the software requirements.*
-

3.9.1 General Information

3.9.1.1 Software architectural design

Beyond a static and a dynamic view, there is no common definition which views are required and no criteria for the completeness of the sum of views. There are some approaches in the industry that specify the kind of information that is required for the view (“viewpoints” which are collections of patterns, templates, and conventions for constructing one type of view) and the integration of the views in a thoroughly architectural design description.

In most cases the software architectural design is a graphical representation of the software supplemented by textual explanations.

Static software architecture views allow the decomposition of the software into manageable elements with high cohesion and low coupling. Is decomposition supports the assignment of requirements to these architecture elements and will help the organization to distribute the work to the developers. Architecture elements of the software that are developed external to the assessment scope (e. g. open-source software, platform software, third-party software, etc.) will also be included as dedicated elements in the software architectural design and have to be considered as well for interface analysis, dynamic behavior, resource consumption objectives etc.

As appropriate the architecture elements are detailed further in the architectural design down to the components as the lowest level elements. The components consist of one or more units and are subject of the software detailed design process (SWE.3) (See “Annex C Terminology” of the PAM for definition of the terms element and component).

Although, according to the state-of-the-art, interrupt service routines shall not include any domain logic behavior or complex algorithms, interrupt handling still represents parallel control or even data flows. Especially high interrupt loads may cause interferences in the

application. Automotive SPICE v4.0 therefore considers it important to treat relevant interrupt routines as software units in order for them to be reflected in the software design, the dynamic design in particular.

Although, according to the state-of-the-art, interrupt service routines shall not include any domain logic behavior or complex algorithms, interrupt handling still represents parallel control or even data flows. Especially high interrupt loads may cause interferences in the application.

Automotive SPICE v4.0 therefore considers it important to treat relevant interrupt routines as software units in order for them to be reflected in the software design, the dynamic design in particular.

3.9.1.2 Talking about interrupts

Although, according to the state-of-the-art, interrupt service routines shall not include any domain logic behavior or complex algorithms, interrupt handling still represents parallel control or even data flows. Especially high interrupt loads may cause interferences in the application.

Automotive SPICE v4.0 therefore considers it important to treat relevant interrupt routines as software units in order for them to be reflected in the software design, the dynamic design in particular.

3.9.2 Rating Rules within the process

Rules for rating consistency between the BPs in this process are not defined. This is due to the nature of BPs as describing separate concerns which shall be addressed individually. Further, a Process Attribute shall be rated based on the Process Performance Indicators, i.e. not based on a subset. If an assessment context-sensitive dependency is identified by the assessor, then he may rate correspondingly but shall provide comprehensive arguments for that in the Assessment Report.

3.9.3 Rating Rules with other processes at level 1

None.

3.10 SWE.3 Software Detailed Design and Unit Construction

- *The purpose is to establish a software detailed design consistent with the software requirements and the software architecture, and to construct software units consistent with the software detailed design.*
-

3.10.1 General Information

3.10.1.1 Detailing out software components

The software detailed design refines the components specified in the Software Architecture Design process into software units and their interfaces. These software units that are not further refined on the design level and their interfaces are the basis for generating or developing the source code for the derived software units.

The detailed design for a component shall describe the approach to satisfy the mapped software requirements by describing how code will be organized both statically and dynamically. It shall also describe how different units will interact.

In assessment practice it was observed software units often lack in a description of their own intended technical or domain knowledge-oriented behavior. Apparently, the Automotive SPICE v3.1 texts

SWE.3.BP1 “Develop a detailed design for each software component ... that specifies all software units...”

...was often interpreted as the mere identification of software units.

SWE.3.BP2 “Define interfaces of software units”

...was often interpreted as the mere signature of software units.

SWE.3.BP3: “... Evaluate ... the interaction between ...software units.” ...was often interpreted as mere call structures of software unit interfaces.

For these reasons, Automotive SPICE 4.0 now uses more explicit verbs and terms in BPs:

- SWE.3.BP1: ... Specify the *static structure* of the software units, their *relationships*, and their *interfaces* including...
- SWE.3.BP2: ... Specify the *behavior of each* software unit ... Specify the *interactions* between relevant software units to *fulfill the component’s dynamic behavior*.

3.10.1.2 Views on software detailed design

Beyond a static and a dynamic view, there is no common definition which views are required and no criteria for the completeness of the sum of views. There are some approaches in the industry that specify the kind of information that is required for the view (“viewpoints” which are collections of patterns, templates, and conventions for constructing one type of view) and the integration of the views in a thoroughly detailed design description.

In most cases the software detailed design is a mix of graphical representation and/or textual explanations.

3.10.1.3 Traceability in SWE.3

It is necessary to understand which software requirement is, finally, represented in in the detailed design. Reasons are e.g. comprehension of the logic of the software and efficient impact analysis in the context of changes.

During software requirements analysis the two following traceability options can be considered, depending on the content of the requirement:

Option A:

Traceability via software architecture.

Realizing a requirement that requires dynamic behavior and interactions between software components and, subsequently, their software units.

Example:

Consider the software requirement:

“The software shall process the bus message frame “start motor” within 500 [ms] with a tolerance of +20[ms]”.

Option B:

Traceability between a particular software requirement and a software detailed design element.

Example: CAN matrix

Software interface requirements may demand using a defined CAN matrix. Since there will be e.g. a set of software units decoding such messages, direct traceability may be intuitive.

This is why in Automotive SPICE 4.0 SWE.3.BP talks about traceability between software requirements and “detailed design” instead of “software units” only. One more reason is: the term “software unit” may be interpreted as the implementation of the unit in source code. This interpretation is not intended because the source code is the implementation solution of a unit specified in the detailed design.

3.10.1.4 Strengthening of ‘SWE.3.BP2 Develop Software Units’

In SWE.3, BP2 emphasizes principles according to which the code is to be developed, i.e. reflecting such principles at coding time already. In fact, there are coding principles that can be expected at CL1. Note 7 in SWE.3 suggests that such coding principles are e.g. “no implicit type conversions”, “one entry and one exit point in subroutines”, and “range checks (design-by-contract)”. Further CL1-level coding principles relate to the robust, error-free and technically correct behavior of the final software product. Consequently, in SWE.4 software unit static verification and code reviews, respectively, can then also verify whether those coding principles have been adhered to.

Examples for coding principles that can be expected at CL1:

- no implicit type conversions
(to avoid value range under-/overflows)
- one entry and one exit point in subroutines
(to avoid systematic faults with respect to the application domain logic)
- encapsulation at the code level as opposed to e.g. global visibility of variables
(to avoid systematic faults)
- defensive programming to avoid systematic faults, e.g.
 - range checks (design-by-contract)
 - an 'Enum' in C with explicit initialization and distinct values with a certain Hamming distance instead of a single bit to increase robustness against memory corruption.

This supports the consideration of coding principles at an earlier point in time from a development lifecycle perspective.

Note that this strengthening is not to introduce redundancy, or is overlapping, with CL2. Other coding principles that are to be considered in regards to GP 2.2.1 are ones that are not generally necessary because they depend on the specific assessed context. The following examples for coding principles depending on the product business strategy such as platform development could be expected at CL2:

- maintainability and comprehensibility by means of e.g. naming conventions and commenting templates
- portability
- scalability
- reusability

as opposed to a context which is about developing and maintaining a very customer-specific legacy product for only one particular application; none of the above-mentioned principles would necessarily apply.

A further advantage of strengthening SWE.3.BP2 is that it should now receive a higher attention by assessors. Previously, during assessments this BP was often rated as F based on the mere existence of code.

3.10.2 Rating Rules within the process

3.10.2.1 What a “software unit” is

In the first place, “software unit” is not an implementation-level term but a logical modeling-level term (see SWE.3.BP1). The logical modeling level represents the detailed software design, and a detailed software design is always a semantical abstraction from the source code but not identical with the source code itself. Further, the software architectural and detailed design are created using the application domain language and entities. Consequently, the view of a software unit being an “inseparable coherent piece of behavior” that is also “verifiable standalone” makes it an application domain knowledge perspective. This is independent of the question

- of how many C functions will realize the software unit
- in which *.h and *.c files the software units finally are “physically” represented.

Examples:

- a. A motor driver *.c file with several C (sub-)functions transforming logical motor commands into IO signals (for the direction of rotation and for PWM / duty cycles for setting the motor speed) can be considered a software unit. A “motor driver” is an application domain entity name with exactly that coherent expected behavior.

A single C function implements a UML state machine (that is defined for a software Unit in the design model) by means of several `switch-case` statements (see the Example 2 in subsection “**The purpose of code coverage**” in Section “

- b. **SWE.4 Software Unit Verification and Integration Verification**). This single C function can be considered a software unit as, still, it includes the considered application domain behavior.

In both examples, at the code level these software units may of course be further divided up into many smaller C functions or even *.c files. In example (b.) specifically all behavior in a `case`-block might be factored out into their own C subfunctions which makes sense. However, doing so does change the fact that, from the application domain knowledge perspective, the software unit is still the sum of all those subfunctions.

Carrying out such refactoring too far, however, may even introduce code review inefficiency. The reason is that a software unit shall be verified against its specification (and not against the course code itself), which would result in having to switch between many files.

As a result, it can be concluded that a software unit can be both, a single subroutine or a no. of subroutines, e.g. a single C function but also an entire *.c file containing several C functions, and that the decision of a software Unit boundary must, also, be application domain-driven.

Rating Rules:

[SWE.3.RL.1] If software units in the detailed design are mapped to a cluster of programming language routines but not to single atomic routines then SWE.3.BP1 and SWE.3.BP3 shall not be downrated.

3.10.2.2 Code metrics vs. software unit boundaries

A further consequence from the above is that code complexity metrics alone are a reason to determine a software unit boundary. Furthermore, considering one single code metric alone for such purposes should be avoided. A combination of selective code metrics may provide meaningful hints on where refactoring should be discussed in order to achieve “clean”, comprehensible, and maintainable code. However, should there be no didactical or conceptual advantage in regard to the application domain knowledge then the software unit boundary should not be reconsidered.

[SWE.3.RL.2] If code metric targets for software units are formally violated but there are arguments why the size and boundary of a software unit are acceptable then SWE.3.BP1 shall not be downrated.

3.10.2.3 Dynamic behavior

For the description of the internal behavior of the software units graphical representations (e.g. UML) and/or textual explanations abstracting from the implemented source code are to be used.

Rating Rules:

[SWE.3.RL.3] If a software unit is of such a low complexity from the technical application domain knowledge perspective so that its dynamic behavioral description does not require graphical notation in favor of comprehensible narrative explanations, then SWE.3.BP2 shall not be downrated.

3.10.2.4 Traceability and consistency

See also Section 3.3.1 of SYS.1 and Section 2.1.6 here.

3.10.3 Rating Rules with other processes at level 1

None.

3.11 SWE.4 Software Unit Verification and Integration Verification

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

3.11.1 General Information

The software unit verification part of this process covers not only software unit testing aspects but also unit verification aspects e.g. static verification of units.

3.11.1.1 The purpose of code coverage

As is clear from SWE.3 and SWE.4, a unit at the source code level shall be verified against the unit specification in the detailed design. A unit at the source code level shall not be verified against the code itself as this does not prove if the unit works correctly according to the application domain logic. This would just prove that the code works as programmed.

A recurring question is whether during SWE.4 a 100% code coverage of the unit shall be achieved.

Answer:

The purpose is not, generally, to achieve a 100% coverage of all unit code as a verification objective on its own. The purpose rather is to check if a particular test case did cover exactly those parts of the code it was supposed to, based on the test case definition. In other words, code coverage represents accompanying information that addresses completeness of the selected test cases. This means that code coverage alone, in itself, is not a verification objective. See also ISO 26262-6 clause 9.4.4 here.

In example 1 below, when testing the unit `stateTransition1()` with the goal of checking whether the state change is performed entirely and correctly, a code coverage of 100% is expected. The reason is that each single state transition has its own method.

In example 2 below, however, when testing the unit `stateChange()` with the same goal of checking whether the state change for `MY_EVENT1` is performed entirely and correctly, a code coverage of

<100% is expected. The reason is there is only one method including all state changes represented by the various switch-case branches; the code for MY_EVENT1 is only a subset of the entire unit code. Test cases for MY_EVENT2 might not be necessary because e.g.

- already tested in a previous project and not changed ever since
- not relevant for the current release scope

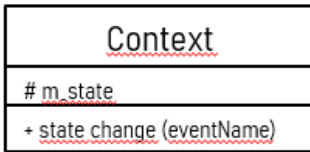
This is one of the reasons why SWE.4.BP2 requires a selection of unit test cases, supported by SWE.3.BP1 Note 1 stating that “a software unit in the detailed design may be, at the code level, represented by a single subroutine (e.g. Example 2) or a set of subroutines (e.g. Example 1)”.

Example 1: possible state machine implementation for a class

<u>Context</u>
<u># m_state</u>
+ stateTransition1() + stateTransition2()

```
void stateTransition1()          void stateTransition2()
{
    If (STATE_A == m_state)     {
        {
            exit_CORRECT_STATE();
            state = NEXT_STATE;
            entry_NEXT_STATE();
            do_NEXT_STATE();
        }
    }
};                               };
```

Example 2: alternative statemachine implementation



```
void state_change (possibleEventsEnum event) {
switch (m_state)
{
    case <stateName1>:
        if (MY_EVENT1 == event)
        {
            exit_ <stateName1>();
            m_state = <stateName4>;
            entry_ <stateName4>();
            do_ <stateName4>();
        }
        break;
    case <stateName1>:
        if (MY_EVENT2 == event)
        {
            exit_ <stateName1>();
            m_state = <stateName6>;
            entry_ <stateName6>();
            do_ <stateName6>();
        }
        break;
    default: // invalid state
};
```

3.11.2 Rating Rules within the process

3.11.2.1 Define software unit verification measures

Rating Rules:

[SWE.4.RL.1] If the complexity of a software unit is below a defined threshold related to a set of combined code metrics so that unit testing is not considered necessary, then SWE.4.BP1 shall not be downrated.

3.11.2.2 Automation of verification measures

Rating Rules:

[SWE.4.RL.2] If a verification measure is automated and the correctness, complete-ness, and consistency of the corresponding scripts and programs are not addressed in the verification measure definition then SWE.4.BP1 must be downrated.

3.11.2.3 Explorative testing vs. traceability/consistency

The testing state-of-the-art not only comprises testing derived from requirements but also explorative testing based on experience, such as “error guessing based on knowledge”. This is valuable as it adds to the quality of the product. Therefore, explorative tests that are based on experience cannot, by definition, be traced or consistent with the software requirements.

Still, traceability is needed between explorative test cases and their results.

Rating Rules:

[SWE.4.RL.3] If verification measures represent explorative tests, which, by definition, cannot be traced to the detailed design, then SWE.4.BP4 shall not be downrated.

3.11.2.4 Traceability and consistency

See also Section 3.3.1 of SYS.1 and Section 2.1.6 here.

3.11.3 Rating Rules with other processes at level 1

3.11.3.1 Verification measures selection vs. release plans

Rating Rules:

[SWE.4.RL.4] If selection of verification measures is properly done but based on an inadequate or incomplete release plan, then SWE.4.BP2 shall not be downrated.

3.12 SWE.5 Software Component Verification and Software Elements 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.*
-

3.12.1 General Information

3.12.1.1 The Scope of SWE.5

For understanding the concepts of software unit integration and the standalone verification of software components see Section 2.2.

The term "integrated software" as used in the context SWE.5 refers to the sole technical software product, or sample, on which verification is performed. This term alone therefore

- does not address documentation,
- nor does it imply that SWE.4 must have been done prior to SWE.5 as a PAM does not represent a lifecycle model.

3.12.2 Rating Rules within the process

3.12.2.1 Verification measure definition

Rating Rules:

[SWE.5.RL.1] If entry/exit criteria are reasonably specified for a set of verification measures, SWE.5.BP1 and SWE.5.BP2 shall not be downrated.

3.12.2.2 Automation of verification measures

Rating Rules:

[SWE.5.RL.2] If a verification measure is automated and the correctness, complete-ness, and consistency of the corresponding scripts and programs are not addressed in the

verification measure definition, then SWE.5.BP1 or SWE.5.BP2, respectively, must be downrated.

3.12.2.3 Explorative testing

The testing state-of-the-art not only comprises testing derived from requirements but also explorative testing based on experience, such as “error guessing based on knowledge”. This is valuable as it adds to the quality of the product. Therefore, explorative tests that are based on experience cannot, by definition, be traced or consistent with the software requirements.

Still, traceability is needed between explorative test cases and their results.

Rating Rules:

[SWE.5.RL.3] If verification measures represent explorative tests, which, by definition, cannot be traced to the detailed design, then SWE.5.BP6 shall not be downrated.

3.12.2.4 Traceability and consistency

See also Section 3.3.1 of SYS.1 and Section 2.1.6 here.

3.12.3 Rating Rules with other processes at level 1

3.12.3.1 Verification measures selection vs. release plans

3.12.3.2 Rating Rules:

[SWE.5.RL.4] If selection of verification measures is properly done but based on an inadequate or incomplete release plan, then SWE.5.BP3 shall not be downrated.

3.13 SWE.6 Software Verification

- *The purpose is to ensure that the integrated software is verified to provide evidence for compliance with the software requirements using verification measures consistent with the software requirements.*
-

3.13.1 General Information

The aim of software verification is to verify that the integrated software is consistent with the software requirements, which means taking a black-box view on the software. The object-under-verification is the integrated software, not the verification environment. This implies that any verification environment can be applicable.

3.13.2 Rating Rules within the process

3.13.2.1 Verification measure definition

Rating Rules:

[SWE.6.RL.1] If entry/exit criteria are reasonably specified for a set of verification measures instead of each verification measure, then SWE.6.BP1 shall not be downrated.

3.13.2.2 Automation of verification measures

Rating Rules:

[SWE.6.RL.2] If a verification measure is automated and the correctness, completeness, and consistency of the corresponding scripts and programs are not addressed in the verification measure definition, then SWE.6.BP1 must be downrated.

3.13.2.3 Explorative verification vs. traceability/consistency

The testing state-of-the-art not only comprises testing derived from requirements but also explorative testing based on experience, such as “error guessing based on knowledge”. This is valuable as it adds to the quality of the product. Therefore, explorative tests that are

based on experience cannot, by definition, be traced or consistent with the software requirements.

Still, traceability is needed between explorative test cases and their results.

Rating Rules:

[SWE.6.RL.3] If verification measures defined explorative tests, which by definition, cannot be traced to the detailed design, then SWE.6.BP4 shall not be downrated.

3.13.2.4 Traceability and consistency

See also Section 3.3.1 of SYS.1 and Section 2.1.6 here.

3.13.3 Rating Rules with other processes at level 1

3.13.3.1 Verification measures selection vs. release plans

Rating Rules:

[SWE.6.RL.4] If selection of verification measures is properly done but based on an inadequate or incomplete release plan, then SWE.6.BP2 shall not be downrated.

3.14 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.

3.14.1 General Information

3.14.1.1 Motivation behind the Process Purpose

The process VAL.1 Validation centers around “intended use”, thereby addressing the product’s end users. It therefore excludes looking at pure embedded software products, an ECU, or a drive (comprising a motor and an ECU), none of which providing a direct end user interface.

In absence of legal requirements (e.g. a maximum closing force of 100N for window regulators, or homologation requirements), the target expectations behind Validation may be of an explorative, or even subjective, nature.

Example 1: automatic transmission being a mechatronic system

Meeting defined gear shifting time constraints is considered Verification as these can be measured objectively. In contrast, providing an adequate gear-shifting “feeling” rather is a Validation concern requiring feedback from end users or end users representatives.

Example 2: automatic side door access systems

There are no legal closing force requirements. Therefore, how much closing force represents intolerable user harm considering the concrete inertia, kinematics, spring rates, and thickness of rubber seals etc. is a matter of validation, e.g. by means of accident simulations. In contrast, the angle at which the automatic door movement support is to be triggered is a matter of decision which can objectively measured against, thus representing Verification.

Note that the possibility of being able to write up a requirement in the first place does not serve as a distinction criterion for differentiating between Verification and Validation. This is to say, it is not possible to argue it is about Verification whenever one is able to specify a requirement. Related to the two examples above, a requirement could still be about

- 1) defining certain max. acoustics and vibration to express a gear-shifting “feeling”,
- 2) or a maximum closing force, respectively.

Still, determining whether or not these requirements are “adequate” would be a matter of Validation because they must be approximated. This is because of limitations, and the nature, of requirements engineering in terms of dealing with potentially unidentified needs, or identification of appropriate requirements only in an iterative manner.

3.14.2 Rating Rules within the process

3.14.2.1 Verification measure definition

[VAL.1.RL.1] If entry/exit criteria are reasonably specified for a set of validation measures instead of each individual validation measure, then VAL.1.BP1 shall not be downrated.

3.14.2.2 Explorative validation vs. traceability/consistency

Explorative validation measures that are based on experience cannot, by definition, be traced or consistent with stakeholder requirements.

Still, traceability is needed between explorative test cases and their results.

Rating Rules:

[VAL.1.RL.2] If explorative tests are defined as validation measures, then VAL.1.BP4 shall not be downrated.

3.14.2.3 Traceability and consistency

See also Section 3.3.1 of SYS.1 and Section 2.1.6 here.

3.14.3 Rating Rules with other processes at level 1

3.14.3.1 Verification measures selection vs. release plans

Rating Rules:

[VAL.1.RL.3] If selection of verification measures is properly done but based on an inadequate or incomplete release plan, then VAL.1.BP2 shall not be downrated.

3.15 MLE.1 Machine Learning Requirements Analysis

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

3.15.1 General Information

The Machine Learning Requirements Analysis process uses the software requirements that were processed in the Software Requirements Analysis process and the elements of the software architecture as an input.

Results of this analysis are specified functional and non-functional Machine Learning requirements (ML requirements) and specified Machine Learning data requirements (ML data requirements).

3.15.2 Rating Rules within the process

Since the ML requirements belong to the group of Software requirements the rating recommendations from SWE.1 “Software Requirements Analysis process” are also valid here (see 3.8).

ML requirements are derived from the Software requirements that are categorized to be implemented in an ML based software element. ML requirements consist of ML data requirements which are the main input for the SUP.11 Machine Learning Data Management and other ML requirements which are input for the other MLE processes.

ML data requirements shall address:

- Data characteristics to be covered and their expected distributions
- Non-functional requirements (e.g., regarding labeling quality, integrity of data)
- Structure and format of ML data

Other ML requirements should address:

- Functional parts to be implemented for training and testing the ML Model

- Hardware related ML functions
- Receiving signals from electronic sensors
- Non-functional requirements (e.g., performance, quality requirements)

ML requirements have to be granular, understandable, and verifiable. Unclear or generic requirements have to be clarified with the system or software requirement owner.

[MLE.1.RL.1] If aspects a) and b) of the ML data requirements are not addressed then BP.1 shall not be rated higher than P.

3.16 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.

3.16.1 General Information

The goal of this process is to establish an ML architecture. The ML architecture requires consideration of the problem which should be addressed with the ML model. ML models are very good at identifying patterns, but some are better suited for specific problems than others. As an example, often convolutional neural networks are used for object detection.

The ML architecture must contain all necessary ML architectural elements like hyperparameter ranges and initial values, details of the ML model, and possible other software parts which are necessary for MLE.3 “Machine Learning Training”.

For the ML architecture the resource consumption objectives are required to be derived from ML requirements for all resource-critical elements and may differ between the trained ML model and the deployed ML model.

The training is often done in a specific training environment defined in the ML training and validation approach (see MLE.3). Also, for this environment resource consumption objectives should be defined to ensure feasibility of the ML architecture.

3.16.2 Rating Rules within the process

The ML architecture has to consider not only the ML model itself but also any potential additional software which is required to train, deploy, and test the ML model.

Typical examples of necessary ML architectural elements are pre- and postprocessing components, e.g., data augmentation and ground

truth evaluation. It should also be considered that some of these ML architectural elements are required for training but will not be available once the ML model is deployed. Such (classical) software components should be developed according to SWE.3 “Software Detailed Design & Unit Construction” and SWE.4 “Software Unit Verification”. Evaluation of these ML architectural elements (e.g., pre- and postprocessing) should be documented.

Often different ML models are considered and trained. Hyperparameters like learning rate, loss function, model depth, regularization constants will allow the configuration of the ML model. The rationale for different hyperparameters and initial values should be provided, and the decisions taken should be documented.

The ML architecture also has to consider the interfaces between the different ML architectural elements. Typically, interfaces are documented in terms of name, type, range, default value, unit, resolution, and direction.

[MLE.2.RL.1] If the ML architecture does not consider elements necessary to train, deploy, and test the ML model then BP1 shall not be rated higher than P.

3.17 MLE.3 Machine Learning Training

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

3.17.1 General Information

Machine Learning uses an ML model capable of performing a functional mapping of an input to an output tensor of data. The quality of the mapping is optimized by adjusting internal parameters of the ML model until the deviation of output tensors from expected values is better than a predefined threshold measured by the loss function. Usually, these parameters are the weights of a weighted sum or average as input to the activation function of a neuron and the hyperparameters as defined by the ML architecture (see MLE.2).

Even simple tasks lead quickly to a high-dimensional optimization problem because the number of weights depends on the sizes of input and output tensors, the number of layers, and other aspects. Therefore, the training process of a ML model consumes high amounts of memory and computing power. Even more if floating point operations are needed.

ML validation as part of the training process supports the optimization of the hyperparameters during Machine Learning Training (MLE.3). The term “validation” has a different meaning than VAL.1.

Due to the complexity of the task, the training process is usually an iterative process which can require changes of the ML architecture (MLE.2), the training and validation approach, or the training and validation data set. Even with experience, it cannot be ensured from the beginning that a defined ML architecture achieves the required quality immediately with the first training. Therefore, iterative changes are not an indication of failure with MLE.2 or MLE.3 but an inherent part of the process to establish an ML model which eventually satisfies all ML requirements.

The data set for training and validation has to be created from the ML data collection provided by SUP.11 according to the ML training and validation approach. Deviating leads to training results which are not ensured to meet the ML requirements.

3.17.2 Rating Rules within the process

Machine Learning Training requires already for achievement of PA 1.1 a careful preparation of the training environment, especially the required HW resources, and optimization approaches to be used to achieve the wanted optimum in a reasonable time but prevent problems like overfitting.

The data set for ML training and validation of the achieved capability of the ML model in the training cycle must be carefully selected based on predefined criteria to support the training goal and prevent common problems (e.g., bias). Be aware, that a separated data set for training and validation is not necessarily required at training start for some validation approaches (e.g., k-fold cross validation). If validation is required for the training process dedicated validation data must be available.

The expectations for the ML training and validation approach cover these aspects:

- entry and exit criteria of the training including comparison of achieved capability of the ML model with the ML requirements;
- approaches for hyperparameter tuning / optimization to be used in the training;
- approach for data set creation and modification for the ML training and validation;
- training environment, including:
 - required training hardware (e.g., GPU, or supercomputer to be used);
 - interface adapter for provision of input data and storage of output data;
- if required, actions to organize the data set and training environment.

[MLE.3.RL.1] If the ML training and validation approach does not cover one of the aspects b, c, or d then BP1 shall not be rated higher than P.

[MLE.3.RL.2] If the ML training and validation approach uses validation techniques which do not require separated ML training

and validation data sets at ML training start then BP1 shall not be downrated.

3.18 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.

3.18.1 General Information

The Machine Learning Model Testing process focuses on testing the agreed trained ML model to ensure compliance with the ML requirements. Therefore, an ML test approach is specified, and an ML test dataset is created from the ML data collection provided by SUP.11 based on ML data requirements. After successfully testing the trained ML model, a deployed ML model is derived and tested as well.

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.

Testing an ML model is done by comparing results of test data computed using the trained or deployed ML model with expected results and non-functional ML requirements (e.g., KPIs) with defined pass/fail criteria defined in the ML test approach.

Test results supplying a meaningful summary of the computed results for the used test data are required evidence for test execution.

The test data set has to be created from the ML data collection provided by SUP.11 according to the ML test approach.

The ML test data set shall be used for final testing of the trained ML model and the deployed ML model and must not be used for training. This means that no major changes / optimization are performed based on the ML test data set. Because with every optimization some information over the data set leaks into the model quickly resulting in overfitting to the used data set.

If the test fails and optimization of the ML model is needed it must be ensured that the ML test data set is still reliable to ensure compliance with the ML requirements, therefore a change of the ML test data set may be needed.

3.18.2 Rating Rules within the process

3.18.2.1 ML test approach

In general, all ML testing activities should be in line with the ML test approach.

The ML test approach should cover these aspects:

- ML test scenarios with distribution of data characteristics defined by ML data requirements. Therefore, a data characteristic is defined as one property of the data that may have different expressions in the Operating Design Domain (ODD). E.g., weather condition can be a data characteristic that may contain expressions like sunny, foggy or rainy. A ML test scenario is then defined as a combination of expressions of all defined data characteristics e.g., *weather conditions = sunny, street conditions = gravel road*;
- Quantity of each ML test scenario inside the ML test data set. This may be oriented on the frequency of each ML test scenario in the ODD or on the expected criticality of the ML test scenario.
- Expected test result per test datum;
- The required ML testing infrastructure and environment configuration.
- Pass/fail criteria for the ML testing;
- Entry and exit criteria for the ML testing;

[MLE.4.RL.1] If the ML test approach does not cover one of the aspects a-d then BP1 shall not be rated higher than P.

[MLE.4.RL.2] If the ML test data set is used to perform major changes / optimization of the ML model then BP1 shall not be rated higher than P.

3.19 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.*
-

3.19.1 General Information

3.19.1.1 Scope of the HWE processes

See Section 2.1.3.

3.19.1.2 Iterative vs. incremental development

Normally the functional content in the product changes iteratively and incrementally evolves across releases. The term “increment” can be understood as adding a feature or element that did not exist before (analogy: building a house). The term “iteration” can be understood as refining, or adapting, an existing feature or element (analogy: a sculptor working on a sculpture).

Therefore, the complete set of requirements of the final end product does not necessarily have to be available at the project start. Rather, release scopes agreed with the customer will define increments and iterative rework. In this respect, requirements creation can be driven by release definitions over time.

3.19.2 Rating Rules within the process

3.19.2.1 Hardware development without system requirements

In case of software development only, the software requirements may refer directly to the stakeholder requirements. Consequently, consistency and bidirectional traceability have to be ensured between stakeholder requirements and software requirements.

Rating Rules:

[HWE.1.RL.1] In the case of hardware development only, if the traceability and consistency from hardware requirements to stakeholder requirements is established then HWE.1.BP5 shall not be downrated.

[HWE.1.RL.2] If hardware requirements are not derived from system requirements but from platform requirements according to a reuse strategy, then HWE.1.BP1 shall not be downrated.

3.19.2.2 Structuring of requirements

Hardware requirements can be grouped or categorized to support an overview and prioritization. See also Section 2.1.5.2 here.

Rating Rules:

[HWE.1.RL.3] If “functional” and “non-functional” are the only requirements categorization or classification criterion, then HWE.2.BP2 shall be rated as N.

[HWE.1.RL.4] If there is no evidence for prioritization other than a release planning mapping the functionality to future releases, then HWE.1.BP2 shall not be downrated.

3.19.2.3 Requirements mapping to releases

A possible approach to prioritizing requirements is the allocation of requirements to releases. The usage of such an approach will imply that the content of the next and future releases is supported.

Rating Rules:

[HWE.1.RL.5] If there is no evidence for prioritization but a separate release plan consistently mapping hardware functionality to future releases then HWE.1.BP2 shall not be downrated.

3.19.2.4 Analysis of requirements

The indicator HWE.1.BP3 requires “*Analyze hardware requirements. ...and to support project management regarding project estimates*”. This means for example:

- A set of 100 requirements exists. An analysis was done together with the project manager during a project progress meeting. As a result, 20 out of the 100 requirements were decided not to be used, therefore being attributed as “rejected” with an accompanying comment providing expectations.
-

- A set of 10 requirements were planned for the next release. The development team reports to the project manager that this is no longer feasible due to resource constraints. The decision is to not change the status of those 10 requirements but to reallocate them to future releases. This can be evidenced by a comparison of the release plans (which is the process context of MAN.3 but not HWE.1).

Analysis of requirements can be done by means of using by e.g. tool-based attributes, or comments added to the requirements text.

The analysis of system requirements is the basis for a correct implementation. Even though requirements sometimes appear very simple, a well-founded analysis has to be conducted for those requirements. The scope and appropriateness of the analysis depends on the context of product (e.g., platform). The results of analysis can vary from a simple attribute to a complex simulation or the building of a demonstrator to evaluate the feasibility of software requirements.

Rating Rules:

[HWE.1.RL.6] If analysis results of requirements are not demonstrated by means of separate analysis reports or review records but by means of e.g. tool-supported attributes or tool-supported commenting, then HWE.1.BP3 shall not be downrated

[HWE.1.RL.7] If the analysis of requirements is not evidenced by separate review records then HWE.1.BP3 shall not be downrated.

[HWE.1.RL.8] If requirements are prioritized by means of a separate project release plan assigning system requirements to releases, HWE.1.BP3 shall not be downrated.

[HWE.1.RL.9] If the analysis of hardware requirements in regards to technical feasibility is covered by risk management then then HWE.1BP3 shall not be downrated.

[HWE.1.RL.10] If analysis results of hardware requirements in regards to impact on estimates is not consistently used by project management then HWE.1.BP3 shall not be downrated

3.19.2.5 Traceability and consistency

HWE.1.BP5 offers the possibility of having two paths for traceability:

- a) between hardware requirements and hardware design (SYS.3)
- b) between hardware requirements and system requirements (SYS.2)

However, redundancy, i.e. using the two traceability paths for the very same hardware requirement at the same time, is neither intended by this BP nor meaningful. Further, it is not intended to express that all, or the majority of the, hardware requirements should be traced to system requirements directly as a default. Which path appears more appropriate must depend on the actual content of the hardware requirement itself.

See also Section 3.3.1 of SYS.1 and Section 2.1.6 here.

3.19.3 Rating Rules with other processes at level 1

None.

3.20 HWE.2 Hardware Design

- *The purpose is to provide an analyzed design, that is suitable for manufacturing, and to derive production-relevant data.*
-

3.20.1 General Information

3.20.1.1 Scope of the HWE processes

See Section 2.1.3.

3.20.1.2 Why no extra processes for HW Architectural Design and HW Detailed Design?

In hardware engineering practice

- HW architectural design begins at the block diagram level, being the starting point for the detailed design. Detailed hardware design is the level of information from which physical HW instances can be created, i.e. initial block diagrams do not reveal that level of detail
- The entire HW designing process is performed iteratively. Technical details that originate from lower design levels such as schematics or layout (detailed design) might be added to block diagram models (architectural design) in order to provide further information for distinct verification and testing that are aimed to be done at the architectural level.

Further, note that the following assumptions would not serve as a motivation for separating HW architectural and detailed design at the level of a PRM:

1. *“In their development processes companies may have extra activities for architectural and detailed design, mostly done iteratively with HW detailed design.”*
- A PRM/PAM does not represent a lifecycle model, see Automotive SPICE 4.0 Section 3.4
 - A PRM/PAM is at the process-WHAT-level, while processes in companies are at the process-HOW-level. Therefore, it is the assessor’s responsibility to map Assessment Indicators in a PAM

need to the assessed context. See Automotive SPICE 4.0 Section 3.3.

2. *“Two processes would provide a better overview, i.e. a more orderly partitioning of topics”*

- A PRM/PAM, by definition, does not represent a lifecycle model. Therefore, it is the assessor’s responsibility to map Assessment Indicators in a PAM to information presented by projects and organisational units, see Automotive SPICE 4.0 Section 3.3.
- HWE.2 has 10 BPs which is not extensive (other processes have a similar no. of BPs, e.g. MAN.3)

Also note that this HWE PRM/PAM does not represent an ECU level (see Section 2.1.4).

For these reasons, at the level of a PRM, there is no necessity to separate HW Architectural Design and HW Detailed Design into two processes. The BPs needed to assess architectural and detailed design remain within HWE.2.

This is also consistent with the following models:

- ISO 26262-5
- Swedish Standard SS 7740:2018¹
- PISA²
- AIDA³

¹ The choice of the Swedish Standard SS 7740:2018 (being a PRM/PAM aiming for integrating elements from Automotive SPICE® PRM v 4.5 and PAM v2.5, and particular process-related clauses in ISO 26262:2011 1st Ed) was to also have a single hardware design process only (SE.ENG.5). This single process comprises BPs for both hardware architectural design and hardware detailed design.

² In the PISA model (Process Improvement Scheme for Automotive, as proposed by the System & Software Evaluation Centre, National Research Council of Italy), the “hardware segment” consists of four processes. Only one of them “...*pertains to the definition of electronics design, including the preparation of the physical layout*”, namely HW1. There is no separation into HW architectural design and hardware detailed design at the process level; a distinction between HW architectural and detailed design is internal to HW.1.

³ Similarly, to the Swedish standard SS7740, the Italian AIDA model explains itself both as a reference for reaching compliance with ISO 26262 and as a PAM for processes assessment.

3.20.1.3 ISO 26262 “Evaluation of HW Elements” is not an alternative for HWE.3 and HWE.4

The processes HWE.3 and HWE.4

- do not take a single HW element perspective. This is because the term ‘hardware element’ can denote a HW part, a HW component, or the complete hardware (see glossary).
- are not restricted to safety-related products or contexts, so for hardware development the HWE processes represent what ISO 26262 calls ‘evidence of compliance with standards that support quality management’ [ISO 26262-8:2018 clause 5.3.2 example 2].

Clause 13 in ISO 26262-8:2018 addresses how to proceed with a procured individual HW element that is supposed to be used in a safety-related product. Therefore, hardware part evaluation according to ISO 26262-8:2018 clause 13 is complementary to HWE.3 and HWE.4 and does not contradict them.

3.20.2 Rating Rules within the process

3.20.2.1 Traceability and consistency

See also Section 3.3.1 of SYS.1 and Section 2.1.6 here.

3.20.3 Rating Rules with other processes at level 1

None.

It defines a single PRM process “hardware design”, i.e. no separation of a HW architectural and detailed design; the process “hardware architectural metrics” only covers the ISO 26262 clauses on HW architectural metrics and evaluation of safety goal violations due to random hardware failures.

3.21 HWE.3 Verification against Hardware Design

- *The purpose is to ensure that the production data compliant hardware is verified to provide evidence for its compliance with the hardware design.*
-

3.21.1 General Information

3.21.1.1 Scope of the HWE processes

See Section 2.1.3.

3.21.1.2 General explanation

Integration in terms of software or mechanical lifecycle processes is understood as a stepwise assembly of a product, and performing tests along, or in between, the assembly steps. This notion is not always applicable per se to hardware development. Rather, a HW often is fully assembled first, and then HW testing is performed on the fully assembled hardware by e.g. using measuring points inside the HW to test the inputs and outputs with variations.

Further, testing of a single HW element always includes the testing of the interfaces as such tests need electrical input signals and output load. This means, there is no conceptual distinction between ‘testing a single HW element in isolation’ and ‘testing interfaces between HW elements’.

The notion of ‘reusing HW components’ might be understood as integrating a physical, and already verified, HW component. However, reusing HW component is not a “physical activity” in terms of that a HW component would be something “taken off the shelf and soldered onto a PCB”. Rather, this refers to reusing parts of HW design drawings, or models, during the creation of the HW design.

Examples:

- a voltage measurement solution is taken from an earlier schematic design, or from a model database, and placed into another schematic.
- a model library contains components for re-use in the chip design.

This will also require verification of the “re-used” HW component as the influences of the rest of the (new) surrounding hardware must be considered.

Thus, in order to avoid confusion and speculation the term ‘integration’ is not used in the context of HWE.3.

The processes HWE.3 and HWE.4 can be mapped to ISO 26262-5 clause 10.

3.21.1.3 BP “Ensure use of compliant samples” – why

What if HWE.3 was not having this Base Practice?

- Upon not-Ok verification results one would not know whether this is due to design flaws or production (or sample construction workshops, respectively) errors. The later exactly is not in the scope of the HWE PRM/PAM (see section 2.1.1). SPICE models remain PRMs/PAMs for development.
- It would be economically disadvantageous to spend effort on HWE.3 just find out later that this was waste because the verification was performed on an incorrect sample in the first place (Note that this is not an argument at the abstraction level of a PRM/PRM, however still a reasonable one).

In fact, reality shows that

- Sample construction workshops (German: “Musterbau”) sometimes deliver samples that are not compliant (e.g. the exact soldering paste might not have been available, or because of the manual activities).
- Electronics production generally has varying manufacturing quality, or even deviations, even in presence of state-of-the-art production quality plans, production traceability etc.

For these reasons, HWE.3 needs an “interfacing base practice” to make sure that the delivered sample is like what was ordered (means: hardware production data compliant). In order to satisfy HWE.3.BP3, one out of many possibilities certainly is e.g. EOL testing. However, as mentioned above, such processes or aspects are not in the scope of HWE PRM/PAM.

Comparison with Configuration Management:

This BP could be viewed as some sort of “HW baseline integrity check”, and therefore be replaced by an editorial pointer to the Automotive SPICE® process SUP.8. However, SUP.8 encompasses more than just a single baseline audit Base Practice. Further, the definition of a “HW baseline” (which is another Base Practice in SUP.8), actually happens in the context of HWE.2. For better usability and intuition of the HWE PRM/ PAM, the respective BPs in HWE.3 and HWE.4 are introduced, and kept, instead of pointers to SUP.8.

3.21.1.4 Hardware samples are not required for all verification measures

The need for physical hardware samples depends on the actual content of the verification measure. There are verification measures that do not require, or even cannot be performed on, physical hardware samples, i.e. calculations, simulations, reviews, and analyses; still, simulation models can be improved based on measurements with real physical samples.

This is why HWE.3.BP2 “Ensure use of compliant samples” is distinct from HWE.3.BP4 “Verify hardware design”. As per BP text, the latter does not demand using physical samples. Correspondingly, the former only implies that if physical samples will be needed, then they shall be production data compliant.

3.21.1.5 Specify verification measures

HWE.3.BP1 requires the identification of the necessary verification infrastructure and environment setup. This may be supported using simulation of the environment. This environment can be hardware-in-the-Loop simulation, vehicle network simulations, or digital mockups.

Verification results can support the updating of simulation models.

3.21.1.6 Verification logs as evidence for verification results

When verifying the software units, large amount of logged data may be generated, which will be available via e.g. verification logs. This is especially true for automated tests and static verification. Also, if verification is performed manually the results may be provided in different levels of detail.

Verification results need to be meaningfully abstracted, or derived from, such log data. Still, for the purpose of BP5 “Communicate”, the verification results will further be summarized.

3.21.1.7 Communicate agreed hardware architecture and hardware detailed design

Apart from verification personnel, further important stakeholders can be manufacturing. Integrating this party in the information supports ensuring that Special Characteristics and relevant production data are properly verified and controlled in production, and through decommissioning of development.

3.21.2 Rating Rules within the process

3.21.2.1 Verification measure definition

[HWE.3.RL.1] If entry/exit criteria are reasonably specified for a set of verification measures instead of each individual validation measure, then HWE.1.BP1 shall not be downrated.

3.21.2.2 Automation of verification measures

Rating Rules:

[HWE.3.RL.2] If a verification measure is automated and the correctness, completeness, and consistency of the corresponding scripts and programs are not addressed in the verification measure definition then HWE.3.BP1 must be downrated.

3.21.2.3 Explorative verification vs. traceability/consistency

The testing state-of-the-art not only comprises testing derived from requirements but also explorative testing based on experience, such as “error guessing based on knowledge”. This is valuable as it adds to the quality of the product. Therefore, explorative tests that are based on experience cannot, by definition, be traced or consistent with the software requirements.

Still, traceability is needed between explorative test cases and their results.

Rating Rules:

[HWE.3.RL.3] If explorative tests are defined as verification measures, then HWE.3.BP5 shall not be downrated.

3.21.2.4 Traceability and consistency

See also Section 3.3.1 of SYS.1 and Section 2.1.6 here.

3.21.3 Rating Rules with other processes at level 1

3.21.3.1 Verification measures selection vs. release plans

Rating Rules:

[HWE.3.RL.4] If selection of verification measures is properly done but based on an inadequate or incomplete release plan, then HWE.3.BP3 shall not be downrated.

3.22 HWE.4 Verification against Hardware Requirements

- *The purpose is to ensure that the complete hardware is verified to provide evidence for compliance with the hardware requirements.*
-

3.22.1 General Information

3.22.1.1 Scope of the HWE processes

See Section 2.1.3.

3.22.1.2 BP “Ensure use of compliant samples”

Why is this Base Practices also needed in HWE.4? Will the samples used for HWE.3 not be the same as the ones used in HWE.4? This might be, but is not generally, the case. From a HW development lifecycle perspective, reasons are e.g.

- In early development phases breadboards are used which are typically not used anymore in the context of HWE.4.
- The samples for HWE.3 and HWE.4 do not always have the same assembly placement/mounting options (German: “Bestückung”) due to different verification goals, verification environments, and HW delivery purposes in HWE.3 and HWE.4.

3.22.1.3 Why HWE.4 does not require HW design-compliant samples

It is not necessary to use the same samples for both HWE.3 and HWE.4. Reason:

- a) The BP “Ensure use of compliant samples” guarantees that varying manufacturing quality, or even manufacturing deviations, are excluded (see HWE.3).
- b) Therefore, two different samples can be used in HWE.3 and HWE.4, respectively. This means, the same samples do not need to undergo both HWE.3 and HWE.4. If HWE.3 proves for a sample X to be design-compliant, then it can be concluded that sample Y, which was used for HWE.4 only is design-compliant, too, and vice versa.

For this reason, HWE.4 only requires production data-compliant samples as HWE.3 does. Also remember that a PAM is not a lifecycle model and therefore cannot predefine any order of processes or sample processing (see Automotive SPICE PRM and PAM Section 3.4). However, both design compliance and hardware requirements compliance must still be proved by means of physical samples.

3.22.1.4 Hardware samples are not required for all verification measures

The need for physical hardware samples depends on the actual content of the verification measure. There are verification measures that do not require, or even cannot be performed on, physical hardware samples, i.e. calculations, simulations, reviews, and analyses; still, simulation models can be improved based on measurements with real physical samples.

This is why HWE.4.BP2 “Ensure use of compliant samples” is distinct from HWE.3.BP4 “Verify hardware design”. As per BP text, the latter does not demand using physical samples. Correspondingly, the former only implies that if physical samples will be needed, then they shall be production data compliant.

3.22.1.5 Specify verification measures

HWE.3.BP1 requires the identification of the necessary verification infrastructure and environment setup. This may be supported using simulation of the environment. This environment can be hardware-in-the-Loop simulation, vehicle network simulations, or digital mockups.

Verification results can support the update of simulation models.

3.22.1.6 Verification logs as evidence for verification results

When verifying the software units, large amount of logged data may be generated, which will be available via e.g. verification logs. This is especially true for automated tests and static verification. Also, if verification is performed manually the results may be provided in different levels of detail.

Verification results need to be meaningfully abstracted, or derived from, such log data. Still, for the purpose of BP5 “Communicate”, the verification results will further be summarized.

3.22.2 Rating Rules within the process

3.22.2.1 Verification measure definition

[HWE.4.RL.1] If entry/exit criteria are reasonably specified for a set of verification measures instead of each individual validation measure, then HWE.4.BP1 shall not be downrated.

3.22.2.2 Automation of verification measures

Rating Rules:

[HWE.4.RL.2] If a verification measure is automated and the correctness, completeness, and consistency of the corresponding scripts and programs are not addressed in the verification measure definition, then HWE.4.BP1 must be downrated.

3.22.2.3 Explorative verification vs. traceability/consistency

The testing state-of-the-art not only comprises testing derived from requirements but also explorative testing based on experience, such as “error guessing based on knowledge”. This is valuable as it adds to the quality of the product. Therefore, explorative tests that are based on experience cannot, by definition, be traced or consistent with the software requirements.

Still, traceability is needed between explorative test cases and their results.

Rating Rules:

[HWE.4.RL.3] If explorative tests are defined as verification measures, then HWE.4.BP5 shall not be downrated.

3.22.2.4 Traceability and consistency

See also Section 3.3.1 of SYS.1 and Section 2.1.6 here.

3.22.3 Rating Rules with other processes at level 1

3.22.3.1 Verification measures selection vs. release plans

Rating Rules:

[HWE.4.RL.4] If selection of verification measures is properly done but based on an inadequate or incomplete release plan, then HWE.4.BP3 shall not be downrated.

3.23 SUP.1 Quality Assurance

The purpose 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.

3.23.1 General Information

The Quality assurance process covers all independent and objective activities for work products and processes based on defined project specific criteria.

From the identified project-specific criteria, methods are derived which ensure the quality of all work products (i.e. not just software source code) and processes for the project.

Agile methods in development are also compatible with suitable quality assurance measures, e.g. "early and objective evaluation", DOD (definition of done), integrated learning cycles etc.

Measures should cover review methods, audits, assessments, lessons learned workshops, frequency, review coverage, and review participants for all relevant work products and processes.

Based on the established independence an appropriate level of management and other relevant stakeholder for escalate non-conformances has to be identified.

These cover all relevant stakeholders (e.g. technical and quality management, management, customer, suppliers). After escalations, these stakeholders shall drive corrective actions.

If quality assurance non-conformances are to be treated as problems according to the problem resolution process, this may have impact in quality assurance process too.

If the indicator about configured items in configuration management process may have impact to quality assurance because the scope for QA is not well defined.

The quality assurance of work products and process activities is the essential task in QA. The absence of even one of these two activities has a significant impact on the process and needs to be reflected within PA 1.1.

3.23.2 Rating rules within the process

3.23.2.1 Ensure independence of quality assurance

Independently and objectively quality assurance without conflicts of interest can be reached using different approaches.

Internal persons from a different project or team, department, or business area.

In small organizations with people who have close relationships with each other, organizational independence can sometimes not be sufficiently effective. The more independent the individual is in terms of organization, the less competent he or she is likely to be in the subject matter.

External contracted persons.

External persons may have less knowledge of the facts to be examined. In addition, external contractors are not necessarily completely independent, as they may seek repeat business.

Internal heterogeneous team.

A mix of internal representatives of different teams, departments or business areas.

External heterogeneous team.

A mix of external parties and internal representatives of different teams or departments.

Quality assurance must also extend to the quality of supplier deliveries, if necessary and negotiated. Supplier-related activities must be clearly identified and may also include assessments.

[SUP.1.RL.1] If the quality assurance is neither reporting nor escalating independently, the indicator BP1 must not be rated higher than P.

3.23.2.2 Define criteria for quality assurance

Identify and define project specific criteria based on the project situation. This may include timing, budget, customer quality criteria, complexity and all known constraints.

[SUP.1.RL.2] If there are no quality criteria defined, the indicator BP2 must not be rated higher than P.

3.23.2.3 Assure quality of work products

To assure the quality of work products, reviews as a universal effective tool have to be performed. These reviews based on predefined review methods and review criteria. Review coverage and all relevant review participants needs to be known. All review participants have to be identified which have an interest in the work product (e.g. testers have to review the requirements).

[SUP.1.RL.3] If appropriate activities to evaluate the work products do not contain review methods, the indicator BP3 shall be downrated.

[SUP.1.RL.4] If the quality assurance of work products is done based on checking for pure existence only, the indicator BP3 must not be rated higher than P.

[SUP.1.RL.5] If the quality assurance of work products (BP3) is rated N or P, PA 1.1 must not be rated higher than L.

3.23.2.4 Assure quality of process activities.

Process quality assurance may include process assessments and spot checks, problem analysis, regular check of methods, tools, documents and the adherence to defined processes, reports and lessons learned that improve processes for future projects.

[SUP.1.RL.6] If the quality assurance of process activities is based on performing process assessments (either by a customer or internally) only, the indicator BP4 must not be rated higher than P.

[SUP.1.RL.7] If the quality assurance of process activities (BP4) is rated N or P, PA 1.1 must not be rated higher than L.

3.23.2.5 Ensure resolution of non-conformances

Non-conformances identified in any kind of quality activities, such as reviews have to be resolved.

Often there is a lack of understanding that the initiator of the problem cannot determine when something needs to be improved and therefore joint coordination is necessary for these. This coordination is therefore inevitably necessary in a timely manner.

Non conformances must have a priority, a defined time span and a due date for resolution. The solutions must be agreed with those responsible for the solution and the associated stakeholders.

[SUP.1.RL.8] If non-conformances related to work products neither identified nor documented the indicator BP3 shall be downrated.

[SUP.1.RL.9] If non-conformances related to process activities neither identified nor documented the indicator BP4 shall be downrated.

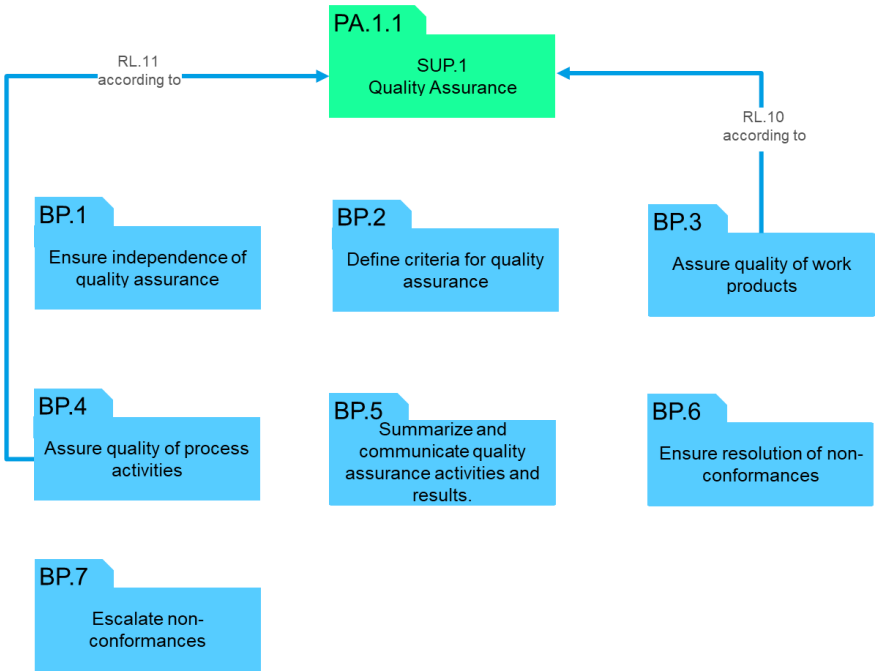
[SUP.1.RL.10] If non-conformances are not tracked or not resolved in a timely manner the indicator BP6 shall be downrated.

3.23.2.6 Escalate non-conformances

In practice, escalations are often delayed, not adequately addressed and not systematically tracked until they are resolved. It can happen that management tends to demand more and more information in order to delay decisions.

Established criteria of urgency and impact can help address the appropriate level of management, escalation principles can be helpful in the preparation of decisions and provide the necessary information. A defined status model for escalations can help to track the escalations to completion.

[SUP.1.RL.11] If escalations are not followed up by corrective actions, the indicator BP7 must not be rated higher than P.



3.24 SUP.8 Configuration Management

The purpose is to establish and maintain the integrity of relevant work products and make them available to affected parties.

3.24.1 General Information

The configuration management covers the identification of configuration items, i.e. inputs and work products of relevant stakeholders of processes and to control their modifications for all conditions.

Furthermore, configuration management covers the management of baselines resulting from different properties like disciplines, sites, processes, etc.

The configuration management varies between domains (like hardware engineering, system or software development). They can have very different management approaches and “de facto” standard tools, but have the same purpose.

Configuration management importance and complexity rises with the size of the organization, number of interfaces and the number of work products that need to be maintained.

Configuration Management supports Level 2 objectives related to other processes. This characteristic may challenge the process review for its own relationship to PA 2.2. For example the actual versioning of configuration items refers to GP 2.2.2 / GP 2.2.3.

3.24.2 Rating rules within the process

3.24.2.1 Identification of configuration items

The identification of configuration items needs to consider the organization, domains and respective stakeholders. As there can be a number of reasons to include or exclude configuration items, criteria shall be defined. Such criteria can be derived for example from formal requirements (e.g. in safety or security), policies, application parameters, categories such as documents, requirements, source code, deliveries etc. As configuration items identification can support

Quality assurance (SUP.1) and vice versa, also quality driven criteria may be given or derived for it.

Configuration items have very different characteristics for their creation, change and maintenance, resulting in high effort and time to be spend for **change on hardware physical items** (e.g. printed circuit boards, ICs, power supplies, sensors, enclosures), while software related configuration items in relation can be changed quickly with low effort. An evaluation of selection criteria should consider the implications of this characteristic.

For the Configuration Management it is crucial to establish control of changes to the relevant product it is intended to support. Therefore the selection criteria should be verifiable on the outcome of the configuration item identification. It is not an obligation to include development tools themselves as configuration items, unless they may become a part of the product to be controlled.

[SUP.8.RL.1] If the configuration items identification fails to include the organizational and stakeholder needs related to the product(s) to be controlled, the indicator BP1 must not be rated higher than P.

[SUP.8.RL.2] If the identification of configuration items is not sufficient to control the changes to the related product(s) (BP1), the indicators BP4, BP5 and BP6 shall be downrated, respectively.

3.24.2.2 Configuration Management mechanisms

In order to support organizations for the availability of configuration items, the configuration management enables and supports the parallel working of configuration item owners for which every domain has developed different practices and tooling's.

Configuration mechanisms need to scale for size and complexity and the sheer amount of configuration items to be managed. With low numbers of configuration items, this may be managed in simplified form and can even be predefined by Project Management (MAN.3) or product release plans (SPL.2).

For domains with a few “hard to change” **physical configuration items** an individual change may be managed sufficiently without a

dedicated tooling, for example within the respective integrated design tool. The complexity however can become challenging with the rising number of parallel configuration items and series of changes to subcomponents.

Software driven domains benefit from the low effort and time to change their mostly nonphysical configuration items, while ensuring their owners to work in parallel as much as possible without conflicts. As this results in the highest complexity to be managed for its configuration management, different and additional practices can be necessary.

For example **branching and merging** is a practice to create different versions of a codebase (branch), allowing to make changes to it and then merging these changes back or even forward to a higher level of a codebase. Branches can satisfy different purpose like single release creation and maintenance, stabilization and troubleshooting of versions, or preparation of versions that can be supported for long time bug fixing only (frozen branch). With branching and merging control not only the availability is managed, it may also support quality related objectives for driving specific software metrics specific for each branch (e.g. different definitions of code test and review coverage).

[SUP.8.RL.3] If there is no dedicated configuration management tool in place, but the established procedure is adequate for the complexity of the product to be developed, this must not be used to downrate the indicator BP3.

[SUP.8.RL.4] If the established mechanisms for configuration management are not able to support the complexity related to the product, the indicator BP3 shall be downrated.

3.24.2.3 Baselines

Configuration baselines are snapshots of configuration items at a specific point in time. They can act as reference point for future changes and support roll back when necessary. Baselines can be driven by purpose and time, based on the configuration items properties.

The expectations for establishing baselines cover these aspects:

- Internal and external baselines are created for all events as required, according to the configuration items properties. For example, external baselines may be reflected with SPL.2.BP8 for the release package delivered to the intended customer.
- Overall baselines are created over different disciplines, sites, processes etc.
- The baselines contain complete and consistent sets of items required to reproduce the progress taken between the baselines.

[SUP.8.RL.5] If baseline are not sufficiently established to control configuration items, the indicator BP5 shall be downrated.

[SUP.8.RL.6] If baselines are based on obsolete or inaccurate properties of configuration items, the BP.5 shall not be rated F.

3.24.2.4 Completeness and consistency

Completeness and consistency of configuration items and baselines is important to verify the overall quality and reliability of the configuration management. It requires an appropriate set of measures for ensuring completeness and consistency.

Such can be supported by using traceability information, results of verifications, data of version control systems and change control systems for verifying changes are properly documented.

Dedicated configuration audits may verify the suitability for the respective domain and organization in addition to the physical integrity of the configuration management.

[SUP.8.RL.7] If baselines for different disciplines or processes are not consistent, or if overall baselines do not exist, the indicator BP7 shall be downrated.

[SUP.8.RL.8] If establishing baselines (BP5) is downrated, the indicator BP7 shall be downrated.

3.24.2.5 Verify backup and recovery mechanisms' availability

Backups are an aspect to ensure the availability, integrity and security of the configuration management for all its stakeholders. Factors that influence the robustness and performance of backups are the

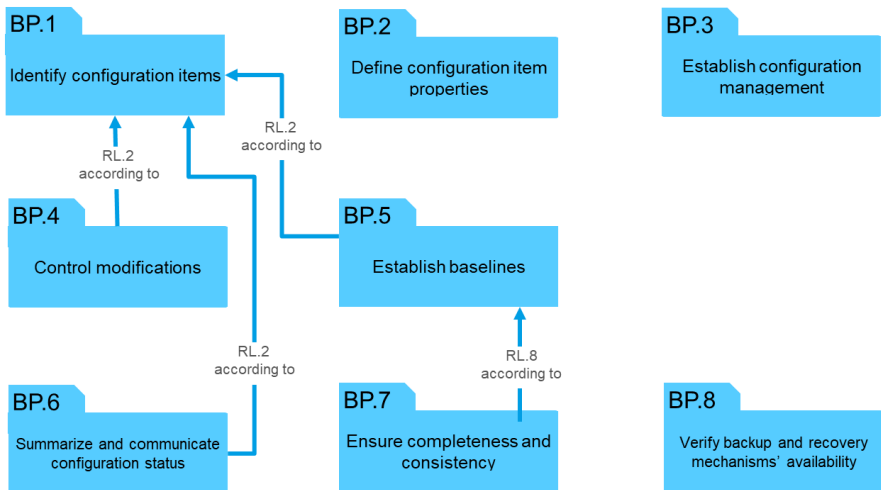
frequency and location they are created at and the evaluation of retention period and recovery process.

Even as this is recognized as a major foundation for configuration management, it is often only a centralized IT organization that can demonstrate the evidence data, while it is seen outside the scope of the configuration management personnel. This should not be noted in a negative evaluation since it can be based for example on IT policies or strategic decisions (e.g. outsourced IT for higher professional level)

IT services may also be demonstrated by certifications.

Backup and recovery mechanisms must not be confused with archiving.

[SUP.8.RL.9] If the frequency or the location is insufficient for the product configuration managements backup and recovery services, the BP.7 shall be downrated.



3.25 SUP.9 Problem Resolution Management

The purpose is to ensure that problems are identified, recorded, analyzed, and their resolution is managed and controlled.

3.25.1 General Information

The Problem Resolution Management Process covers the management of all issues where e.g. more than one stakeholder is involved, or which are not resolved immediately.

Problem management may include multiple interfaces or instances to observe a problem for all its specific interactions between the initiator and the relevant organization. For example, an initiator may have a customer support organization in front of a development group that may require to review each interface individually.

3.25.1.1 Authorize urgent resolution action

When the timeframe for the creation of a permanent problem resolution is not given, an urgent resolution can be required.

These may require for example, the release of recommendation, guidance information or even a workaround. Urgent resolution actions may result out of the need to prevent further damage and/or harm, therefore may also include unconventional approaches such as disabling functionality or setting systems out of order.

Such short term and workaround actions need to be synchronized with further permanent problem solution(s) including their authorization.

Urgent resolution(s) and permanent solution(s) may need to be managed with parallel problem records. Therefore, the authorization needs to include all related interactions of such parallel management also in any later problem status. (see also **3.25.2.4**)

3.25.2 Rating rules within the process

3.25.2.1 Problem identification

The identification of problems shall include these aspects:

- Project life cycle phase in which problem is recorded and needed (e.g., during prototype construction, series development)
- Initiating interfaces of project-specific disciplines, affected domains and subprojects (e.g., software platform, AI build, hardware sample).
- Initial status or mapped workflow for problem records.
- Supporting information required for example for reproducibility, frequency of the problem occurrence or other related observed effects and patterns.

[SUP.9.RL.1] If the problem identification does not include interfaces between multisite organizations/projects, subprojects, and/or groups in case of correspondingly complex projects, the indicator BP1 must not be rated higher than P.

[SUP.9.RL.2] If the identification and recording of problems is rated P or N due to insufficient content, the indicator BP2 shall be downrated respectively.

3.25.2.2 Determination of cause and impact

The expectations for an adequate cause and impact determination of problem records cover these aspects:

- The systematic evaluation of potential effects of detected problems on systems (e.g., use of base software components in different software projects).
- Identification of work products which are affected by the problem.
- The systematic consideration of similar problems in the same application (e.g., in software clones, variants).

[SUP.9.RL.3] If the determination of impact is incomplete due to missing consideration of similar problems in the same application or potential effects on related systems, the indicator BP2 must not be rated F.

[SUP.9.RL.4] If affected work products are not identified by the determination of impact, the indicator BP2 must not be rated F.

[SUP.9.RL.5] If the determination of the cause and impact of the problem (BP.2) is rated P or N, the indicator BP3 must not be rated higher.

[SUP.9.RL.6] If the determination of the problem cause and impact of the problem (BP2) is rated P or N, the indicator BP7 shall be downrated.

3.25.2.3 Alert Notification

Preparing an alert notification for problem resolution involves communicating problems to relevant customers and stakeholders, independent of the initiator or reporter of the problem.

This process step identifies issues in connected or distributed projects, systems and related variants or similar clones. Proactive alert notification may be needed to inform their direct customers, receiving platforms, connected systems or even authorities about problems of critical or urgent character.

Such notifications may be triggered based on criticality, type, or source of a problem. In highly automated environments this may include also lower risk related criteria to inform affected parties on less critical and less urgent items.

For all alert notifications it should be foreseen to include timely suitable descriptions with clear and concise language, reflecting the receiver's level of understanding to become effective.

[SUP.9.RL.7] If there is no evidence for required alert notifications due to missing consideration of potential effects on clones, variants or related systems, the indicator BP4 shall be downrated.

3.25.2.4 Parallelism of problem resolution

Problem resolution often creates parallel work activities, for example problems relating to other problems or change requests. This parallelism may be reflected in handling, linking of parallel or parent/child relationships of problem reports to each other or even to

tasks and change requests. As stated in 3.25.1.1, the urgent resolution action can be one example for such parallelism. Parallelism may quickly become complex, challenging the management of all related disciplines and processes throughout the different states or workflow of them until closure.

[SUP.9.RL.8] If authorization is insufficient in a relevant status or workflow, the indicator BP3, BP4, BP5 and BP6 shall be downrated respectively.

[SUP.9.RL.9] If parallel management of problems with other problems, actions and tasks is insufficiently controlled, the indicator BP.3, BP.4, BP.5 and BP.6 shall be downrated respectively.

3.25.2.5 Track problems to closure

Any problem resolution action is to be tracked to closure of which there may be more than one final state. Final closure may depend on feedback from the initiator of the problem itself which should therefore be searched for in early and objective evaluation, for example in system demos or inspection meetings.

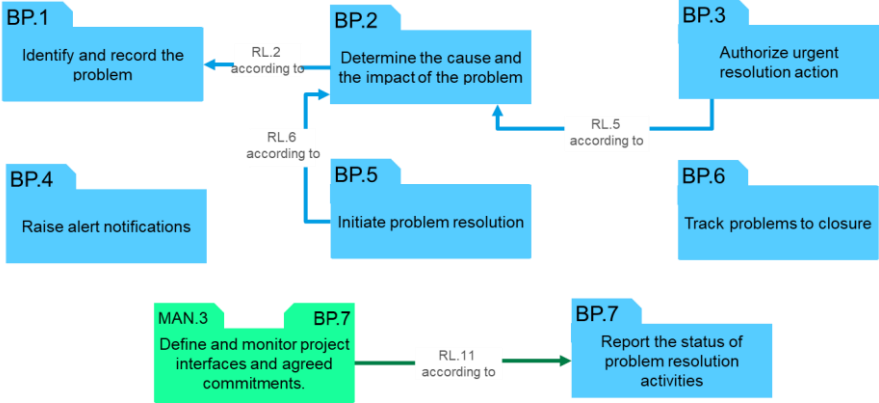
[SUP.9.RL.10] If the initiator of the problem is not sufficiently authorizing the closure of the problem and this is substantial in regard to the project, the indicator BP6 and BP.7 shall be downrated respectively.

3.25.3 Rating rules with other processes on Level 1

3.25.3.1 Track problems to closure

Large amounts of problems may have been driven to closure and not received a final feedback of their initiator. Such condition may create a risk for late failure, resulting into unplanned efforts like additional release cycles and therefore may relate to project management:

[SUP.9.RL.11] If BP7 is downrated due to insufficient authorization of closure and this is insufficiently monitored to the timeline of the project, MAN.3 BP.7 shall be downrated respectively.



3.26 SUP.10 Change Request Management

The purpose is to ensure that change requests are managed, tracked and implemented.

3.26.1 General Information

Change request management may be using the same workflow approach as Problem Resolution Management (SUP.9) or even an independent, fully separated one. In both cases, the decision authority and interaction of their stakeholders is of high importance according to the organizational and/or project-specific aspects like affected disciplines (e.g., system, software, electronics), affected domains (e.g., platform, COTS-Software), internal and external stakeholders or affected sites.

3.26.2 Rating rules within the process

3.26.2.1 Identification and recording of change requests

The identification and recording of change requests shall include these aspects:

- Project life cycle phase in which change is recorded and needed (e.g., during prototype construction, series development)
- Initiating interfaces of project-specific disciplines, affected domains and subprojects (e.g., software platform, AI build, hardware sample).
- Initial status or mapped workflow for change request records.
- Supporting information required, for example alternatives and variable content for a change, references or demonstrators.

Traceability between change requests, problems, affected work products and corresponding baselines has to be ensured over all affected disciplines and all affected domains considering the project-specific complexity.

[SUP.10.RL.1] If the initial recording of change requests is missing information about initiator or reason, BP1 shall be rated not higher than P.

[SUP.10.RL.2] If the identification of change requests does not address interfaces between distributed organizations, subprojects, and/or distributed groups in case of correspondingly complex projects, the indicator BP1 must not be rated higher than P.

3.26.2.2 Analysis and assessment of change requests

The expectations for an adequate analysis of change requests cover these aspects:

- The input from all relevant stakeholders (internal and external) is considered including technical aspects and potential side effects, for example degraded functionality or compatibility problems.
- Feasibility, risks, complexity and impact regarding the potential changes are systematically evaluated and documented.
- Modification and potential alternatives are documented.
- Acceptance Criteria for confirming implementation are established (e.g., selection of existing regression test case(s), newly developed test case, review of all modified work products).
- Change request meets the compliance of agreed regulations and policies.

The analysis of change requests should be capable to identify affected work products. The process performance indicator PA 1.1 of this process therefore should be reflecting the importance of the analysis result.

[SUP.10.RL.3] If the analysis misses to address potential side effects, the indicator BP2 must not be rated F.

[SUP.10.RL.4] If the identification and recording of changes (BP1) is rated P or N due to insufficient content, the indicator BP2 shall be downrated respectively.

3.26.2.3 Decision authority

Due to often more sensitive information like actual efforts, timelines, delegation, subcontracting or different stakeholders participating, a formation of decision authority may become mandatory.

This decision authority, which is for simplification considered as *change control board* (CCB) is expected to cover these aspects:

- All affected disciplines are appropriately represented
- All required stakeholders are represented (e.g. project manager, tester, customer sales manager, Product Owner)
- The participants have the necessary authority to take decisions
- CCB takes decisions in time, delegates issues if necessary..
- Agreement and approval in suitable timely manner, for supporting the alignment of changes into planned releases. (see SPL.2 BP1)
- Dependent on the organizational/project structure and/or constraints (e.g., platform responsibility, budget, effort), there may be multiple, for example hierarchical or organizational CCBs which may have to be represented as well.

A decision authority depends on analysis results for its approval and can be expected to provide its share, but not to provide a verification or repetition of such analysis results.

[SUP.10.RL.5] If not all relevant disciplines or stakeholders are represented in the approval authority the indicator BP3 must not be rated F.

[SUP.10.RL.6] If it is apparent that approval decisions are not taken or not taken in time without justification, the indicator BP3 shall be downrated.

[SUP.10.RL.7] If the analysis of the change request (BP2) is rated P or N, the indicator BP3 must not be rated higher.

3.26.2.4 Parallelism and Traceability of change requests

Change request management often creates parallel work activities, for example changes relating to other change requests, work products, problems, etc. This parallelism may be reflected in handling, linking of parallel or parent/child relationships of change requests to each other or even to tasks and problems.

Such parallelism may quickly become complex, challenging the management of all related disciplines and processes throughout the different states or workflow of them until closure.

Traceability of change requests should support the parallelism in all means. The process performance indicator PA 1.1 of this process should reflect the importance of the traceability.

[SUP.10.RL.8] If the rating of establishing bidirectional traceability (BP4) is downrated due to missing dependencies between change requests and affected work products, the indicator BP2 shall be downrated.

3.26.2.5 Confirmation of Implementation

When confirming change request after implementation following aspects may need to be considered:

- A review of the implemented change requests ensures that all relevant processes (e.g., SYS, SWE, MLE, MAN, and SUP) are applied and corresponding work products are updated accordingly.
- Following activities, actions and tasks are reflected, such as trainings, inspect and adapt meetings, reporting, etc.

[SUP.10.RL.9] If the confirmation of implemented changes misses that relevant processes are not applied, the indicator BP5 shall be downrated.

[SUP.10.RL.10] If the confirmation of an implemented change request is not including agreed acceptance criteria or policies, the indicator BP5 shall be downrated.

[SUP.10.RL.11] If the analysis of change requests (BP2) is rated P or N due to missing information regarding their implementation confirmation, the indicator BP5 shall be downrated.

3.26.2.6 Track change requests to closure

Any change request action is to be tracked to closure of which there may be more than one final state. Final closure may depend on feedback from the initiator of the change itself which should therefore be searched for in early and objective evaluation, for example in system demos or inspection meetings.

[SUP.10.RL.12] If the initiator of the change request is not sufficiently authorizing the closure of the change and this is substantial in regards to the project, the indicator BP6 shall be downrated respectively.

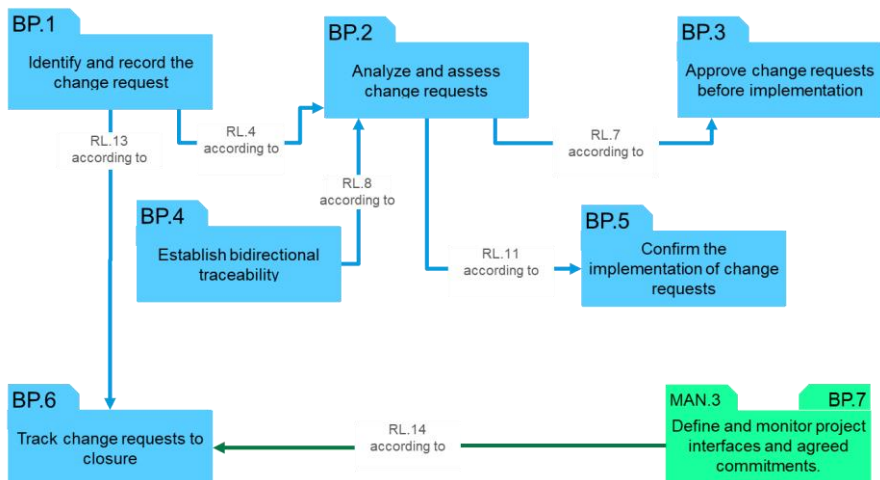
[SUP.10.RL.13] If the initial recording of change requests (BP1) is rated P or N due to missing information about initiator or reason, the indicator BP6 shall be downrated.

3.26.3 Rating rules with other processes on level 1

3.26.3.1 Track change requests to closure

Large amounts of change requests may have been driven to closure and not received a final feedback of their initiator. Such condition may create a risk for late failure, resulting into unplanned efforts like additional release cycles and therefore may relate to project management.

[SUP.10.RL.14] If BP6 is downrated due to insufficient authorization of closure and this is insufficiently monitored to the timeline of the project, MAN.3 BP.7 shall be downrated respectively.



3.27 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 all ML data, and to make them available to affected parties.

3.27.1 General Information

Machine Learning needs data for the training process of MLE.3 and the testing activities of MLE.4. To ensure success of the training process, data of a controlled quality are required which are aligned with the ML data requirements of MLE.1.

Because of cost and effort, ML data are often not only collected and categorized for usage in a single project. Instead, data collection and categorization might be performed continuously by a dedicated organizational entity. A project would then make use of the ML data pool provided by the organization.

SUP.11 Machine Learning Data Management is related only to the data management activities which are required by the MLE process group of the assessed project.

3.27.2 Rating Rules within the process

3.27.2.1 Establish an ML data management system.

The management of ML data requires an ML data management system which includes a suited lifecycle management. This system could be in simple cases reduced to a configuration management system and metadata maintained on the data itself, provided the file type supports metadata.

The assessor needs to judge the suitability of this ML data management system based on the ML data requirements and the amount of data collected for the ML training and test.

For the rating, the assessor needs to understand especially the interfaces to provide and categorize data.

E.g., object detection in video might require a way to label video sequences by company external workers for supervised training. Reinforcement learning might require an interactive approach to judge the correctness of output created by the model which in this case could be part of SUP.11 depending on the setup even if the judgement is part of the ML training process MLE.3. In both cases, the ML data management system must generally provide an opportunity to import all data and store it.

[SUP.11.RL.1] If required ML data management activities are not supported by the ML data management system then BP.1 shall be downrated.

3.27.2.2 Develop an ML data quality approach.

ML data with known quality is an important factor for the success of the MLE group. This requires an approach which defines the ML data quality criteria to be met and how to check that the ML data satisfies the ML data quality criteria. One important aspect of ML data quality criteria is the avoidance of biased data. Biases to avoid may include sampling bias (e.g., gender, age) and feedback loop bias.

Usually, the amount of data required for the ML training and test is so high, that the analysis of the quality of the ML data uses statistical methods.

ML data quality criteria shall be defined before application to the data. Examples of ML data quality criteria are e.g., relevant data sources, reliability and consistency of labelling, completeness against ML data requirements.

3.28 MAN.3 Project Management

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

3.28.1 General Information

The purpose of the process Project Management is to cover all aspects of planning, monitoring and tracking.

In Automotive SPICE 4.0 all planning activities are covered in MAN.3 Project Management and PA 2.1 Performance management process attribute of the respective processes.

Release planning and the management of release baselines represent the determining of functional content to be implemented and are addressed in SPL.2 Product release and SUP.8 Configuration management.

3.28.1.1 Changed concept in Automotive SPICE 4.0 (define and monitor)

The formulation “Define and Monitor” is used for the base practices BP4 (work packages), BP5 (estimates and resources), BP6 (skill, knowledge and experience), BP7 (interfaces and commitments), and BP8 (schedule).

The term “define” addresses the setup of artifacts or documented information where the term “monitor” covers the continued disjoint re-evaluation of artifacts and documented information.

To ensure consistency, adjustment is done based on issues found in monitoring of all of the above mentioned aspects. Consistency here means that all planning aspects demonstrate feasibility of the project.

3.28.2 Rating rules within the process

3.28.2.1 Scope of Work

The scope of work has to cover the motivation (goals), the boundaries including project and product scope, and the constraints of the project,. Describing only the product to be developed is not sufficient.

[MAN.3.RL.1] If the scope of work (BP1) is a product description only, the indicator BP1 must not be rated higher than L.

[MAN.3.RL.2] If the scope of work (BP1) is not appropriately documented and updated during project life cycle, the indicator BP9 must not be rated higher than L.

[MAN.3.RL.3] If the required content of the scope of work (BP1) is distributed over several work products, the indicator BP1 must not be downrated.

3.28.2.2 Defining Project planning artifacts

Based on the scope of work the project life cycle (BP2) is defined that is appropriate to the size, complexity and the context of the project. The life cycle defines major milestones of the project like project start, sample deliveries or start of production and it defines the development phases. The life cycle may be standardized on organizational level and adapted to project specific conditions or developed solely for one particular product development.

It is not always recommendable to set up a detailed work package planning for the entire project life cycle, as there are many changes to the scope of work during the conduct of a project. Therefore as a thumb rule an appropriate detailed planning of work packages encompasses the next two releases.

New concept: IIC – work package (instead of work breakdown structure) work packages also can be:

- Tickets in a tracking system
- Entries in a planning tool
- Cards on a Kanban Board

Dependencies of work packages have to be documented in the work package definition. Usually this includes a precondition that must be present before starting the work package and/or a certain output that serves as precondition for another work package.

The planned effort for work packages shall be determined based on a reproducible estimation.

Not acceptable are e.g. estimates by a single person only without any further review, or without involvement of affected parties.

Another aspect which has to be considered is an adequate size of the work packages. The size of the work packages should not exceed the time of one, max. two monitoring cycles to ensure proper monitoring of the work packages. As an alternative the monitoring of work packages refers to a defined status information to evaluate the degree of completion.

An important aspect which has to be considered is an adequate size of the work packages. Work packages should not exceed the time of one, max. two monitoring cycles.

The necessary resources should include e.g. people, development tools, hardware samples, infrastructure & test equipment.

The way how effort for work packages is estimated shall be reproducible and comprehensible. A simple “best guess” by project manager is not acceptable.

Skills are specific characteristics of people like the ability to communicate, to learn new things, leadership etc. Knowledge include also process, project and product specific training. Experience is a result of long-term practicing certain activities.

Interfaces (BP7) to the project can be

- Development partner (e.g.
 - o other development parties contracted by customer, working on the same system
 - o other development parties contracted by the assessed organization, working on the same system, see ACQ.4)
 - o the customer, working on the same system
- Internal departments (sales, purchasing, quality management etc.)
- Service provider (for e.g. infrastructure, cloud services)
- Platform development
- other development sites.

For all the interfaces that have an impact on the results of the assessed project, the commitments have to be documented and monitored. In case of any deviation an escalation mechanism shall be effective.

[MAN.3.RL.4] If the dependencies between work packages are not identified, the indicator BP4 shall not be rated higher than L.

[MAN.3.RL.5] If any of the following:

- start and end date,
- planned effort and actual effort,
- correction of effort or end date if work package is not completed on time

is missing for work packages, the indicator BP4 must not be rated higher than P.

[MAN.3.RL.6] If the estimation approach used and the origin of the estimates are not reasonable, the indicator BP5 shall not be rated higher than P.

[MAN.3.RL.7] If the size of work packages is larger than two monitoring cycles of the project and the progress of work packages cannot be measured, the indicator BP4 should be downrated.

[MAN.3.RL.8] If critical dependencies in the schedule are not determined, the indicator BP8 shall be downrated.

[MAN.3.RL.9] If training for process, project and product specific topic is not provided to project participants the indicator BP6 shall be downrated.

[MAN.3.RL.10] If more than two development partners are involved and agreements and commitments are not documented and signed the indicator BP7 shall be downrated.

3.28.2.3 Monitoring

A proper monitoring cycle ensures a timely detection of deviations regarding work packages (BP4), Estimates (BP5), Skill, knowledge and experience (BP6), agreed interfaces and commitments (BP7), and schedule (BP8). As a thumb rule a weekly monitoring is in most the cases appropriate. In the context of the project a more frequent

monitoring may be necessary (e.g. when project is in task force mode). The monitoring of skills, knowledge and experience may be decoupled from the monitoring of the other planning aspects.

Tracking of corrective actions may also be linked to SUP.9 Problem Resolution Management.

[MAN.3.RL.11] If the monitoring cycle is not appropriate to detect deviations of planned versus actual planning items, the respective indicators for monitoring (BP4, BP5, BP6, BP7, BP8) must not be rated higher than P.

3.28.2.4 Actual project progress

In practice a project manager cannot resolve all issues that arise during project monitoring. Resource issues frequently are decided by higher level management, schedule deviations may be discussed with the customer. It is essential for the success of a project that mechanisms for communication and escalation with all involved stakeholders are effective.

3.28.2.5 Release management

Releases and their management are not dealt within a single process only but represent a topic distributed across several processes:

- Generally, the project has to define which information, work products, and products have to be delivered to, or received from all relevant stakeholders (MAN.3.BP7).
- The planning of releases is based on the work packages (BP4), and the schedule (BP8) in MAN.3 and on the release plan in SPL.2 Product Release
- The release must be built from configured items (SPL.2.BP4) which relates to configuration management that ensures integrity (SUP.8). Deadline information of product releases will be part of schedules (MAN.3.BP8).
- Release planning is also covered in the requirements processes (SYS.2.BP2, SWE.1.BP2, HWE.1.BP2 and MLE.1.BP2) which expect a mapping of requirements to specific releases (see Note of those BPs).

[MAN.3.RL.12] If product release deadlines or milestones are not consistent with the release scope, the indicators BP8 and BP9 must be downrated .

[MAN.3.RL.13] If for the current and next release the expected activities are defined and monitored appropriately, the indicators BP4 and BP8 shall not be downrated.

[MAN.3.RL.14] If links between different types of planning information are not supported by tools, this must not be used to downrate the indicator BP9.

3.28.2.6 Consistency of planning information

For the rating of the project management process it is important that the definition and monitoring of project attributes like work packages, estimates and resources, project interfaces and dependencies will be evaluated disjoint.

All these attributes have strong dependencies that require to maintain consistency between them. Therefore the adjustment of project management work products is combined into one Base Practice in order to ensure consistency.

Activities of the master project and subprojects have to be aligned and consistent, e.g. project plans for the different engineering domains. Dependencies between these plans have to be easily identified and mapped. Adjustments to activities have to be considered in all relevant planning artifacts.

For project management, explicit links between e.g. plans and schedules are not required. Consistency can be reached by comparing planned versus actual and if needed adjusting planning information.

[MAN.3.RL.15] If planning information of sub-projects is not consistent with the overall planning the indicator BP9 must be downrated.

3.28.2.7 Estimation of change requests and problem resolution

In the course of an automotive development change requests, risk treatment activities, problems, quality issues and defect removals can

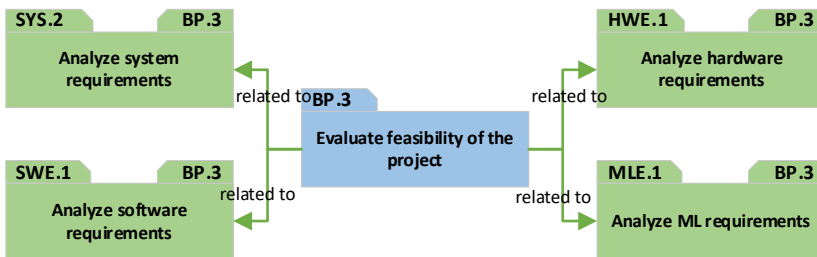
be anticipated. This needs to be reflected in the project planning information.

[MAN.3.RL.16] If the definition of work packages, effort and resources, and the definition of schedule(s) do not sufficiently reflect change requests, risk treatment activities, problems, quality issues and defect removals, the indicators BP2, BP8 and BP5 shall be downrated.

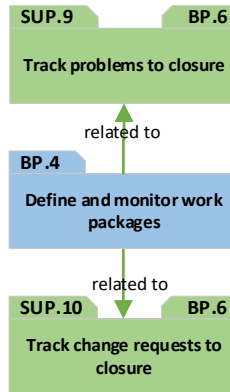
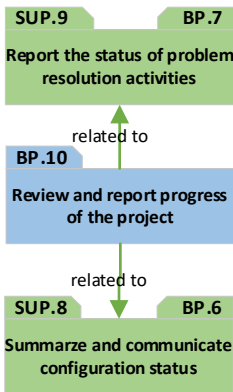
3.28.3 Rating rules with other processes

[MAN.3.RL.17] If the definition of Risk treatment activities (MAN.5.BP5) is downrated due to incomplete definition of risk treatment activities then the indicator BP4 (Define and monitor work packages) shall be downrated.

[MAN.3.RL.18] If the definition of Risk treatment activities (MAN.5.BP5) is downrated due to insufficient identification of required project resources for risk treatment activities, then the indicator BP5 (control of estimates and resources) should not be rated higher than L.



[MAN.3.RL.19] If one of the related BPs in the requirement processes for system (SYS.2.BP3), hardware (HWE.1.BP3), software (SWE.1.BP3) or machine learning (MLE.1.BP3) is downrated due to a missing or incomplete analysis regarding technical feasibility, this should be in line with the rating of the indicator BP3.



[MAN.3.RL.20] If one of the related BPs regarding status of configuration items (SUP.8.BP6) or regarding the status of problems (SUP.9.BP7) is downrated due to a missing or incomplete report, this should be in line with the rating of the indicator BP10.

[MAN.3.RL.21] If one of the related BPs regarding tracking of problems (SUP.9.BP6) or regarding tracking of change requests (SUP.10.BP6) is downrated due to a missing or incomplete status tracking, this should be in line with the rating of the indicator BP4.

3.29 MAN.5 Risk Management

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

3.29.1 General information

It is in the nature of development projects to deal with events or incidents that potentially have a negative impact on achieving its goals in terms of schedule, cost, quality and functional content. In Automotive SPICE these events are called “undesirable events”. To perform an effective risk management for the development project it is necessary to determine the risk management scope. This includes potential incidents that can occur during the project life cycle, regarding the activities performed under responsibility of the project, regarding relevant work products or regarding resources of the project. The risk scope is strongly dependent on the context of the project.

3.29.2 Rating rules within the process

3.29.2.1 Sources of risks

Risk management should consider at least the risk sources of process related and product related undesirable events for which the risk has to be evaluated.

Examples for process related undesirable events are:

- schedule deviations,
- Project progress not according to the plan
- Unavailability of human resources
- commitments from development partners are not fulfilled

Examples for product related undesirable events are:

- defects delivered to customer
- chosen platform is not capable for customer application
- requirements not understood
- inappropriate branch and merge activities
- baselining not consistent
- impact of changes to system behavior

[MAN.5.RL.1] If risk management does not consider process related undesirable events, the indicator BP1 should be downrated.

[MAN.5.RL.2] If risk management does not consider product related undesirable events, the indicator BP1 should be downrated.

[MAN.5.RL.3] If the sources of risks are not updated on a regular basis the indicators BP1 and BP6 shall not be rated higher than P.

[MAN.5.RL.4] If the risk management sources are not updated regularly, the indicator BP.1 shall be downrated.

3.29.2.2 Identify potential undesirable events and determine risks

For the risk identification knowledge and experiences of the product, project and operating environment has to be considered. Also by reuse of components from former projects risks may occur.

To evaluate the risks of an undesirable event all possible influencing effects have to be considered. The risk is characterized by probability of occurrence and the severity of impact that is related to potential undesirable events. The analysis of the risks supports the prioritization of risks and their treatment. Usually the probability and the severity are rated by a discrete scale (e.g. “high”, “medium”, “low”) that is easy to handle and to reproduce. These ratings are then combined to a risk value.

[MAN.5.RL.5] If aspects of reused development results are not considered in the identification of undesirable events but reused components are foreseen within the scope of the project, MAN.5.BP2 shall be downrated.

[MAN.5.RL.6] If impact and probability of undesired events are not evaluated in a reproducible way the indicator BP3 shall be downrated.

3.29.2.3 Risk treatment

The risk value is the basis for prioritization of the risks and for the application of risk treatment. For potential incidents with low risk value risk treatment can be limited to monitoring of the risk.

Concepts for risk treatment may include:

Experiences from problem resolution to avoid re-occurrence

Experiences from previous projects

Accepting the risk

Transferring the risk

Threshold values for risk treatment

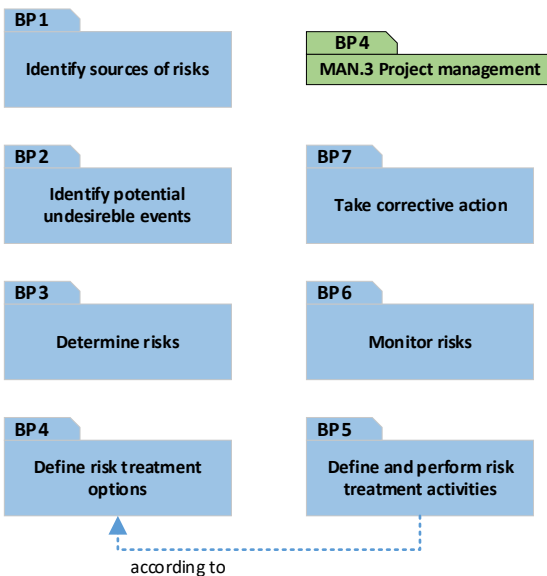
When risk mitigation failed and the incident occurs the Problem resolution process is typically applied to solve the problem.

[MAN.5.RL.7] If the definition of risk treatment activities is not suitable to evaluate the progress and effectiveness of risk treatment activities, the indicator BP5 must not be rated higher than P.

3.29.2.4 Monitor risks

The risk monitoring has to be performed regularly and synchronized to the project milestones and the release plan.

[MAN.5.RL.8] If monitoring of risk and progress of the mitigation activities is not performed regularly (e.g. synchronized with project monitoring cycle), the indicator BP6 must not be rated higher than L.



[MAN.5.RL.9] If BP4 is downrated due to missing criteria for selection of risk treatment options, this shall be in line with rating for BP5 .

3.29.3 Rating rules with other processes

None.

3.30 MAN.6 Measurement

The purpose is to collect and analyze data relating to the work products developed and processes implemented within the organization and its projects, to support effective management of the processes

3.30.1 General Information

As MAN.6 Measurement has been not involved in the former VDA scope the experiences about typical pitfalls in applying this process in an assessment are very limited. Therefore, in this guideline the number of rating rules for this process is rather low.

3.30.2 Rating rules within the process

3.30.2.1 Key Metrics

To understand the behavior of processes, their characteristics and limitations it is necessary to measure certain key metrics. These metrics shall reflect information needs of the management. Examples for process related metrics are:

Lead time

Resource consumption

Defects

Test coverage

Requirements by status

For each metric has to be documented:

- the algorithm (if applicable),
- input data,
- frequency of reporting and
- the control limits or threshold values.

[MAN.6.RL.1] If the metric specification is not completely documented in terms of the topics listed above, BP.2 shall be not rated higher than P.

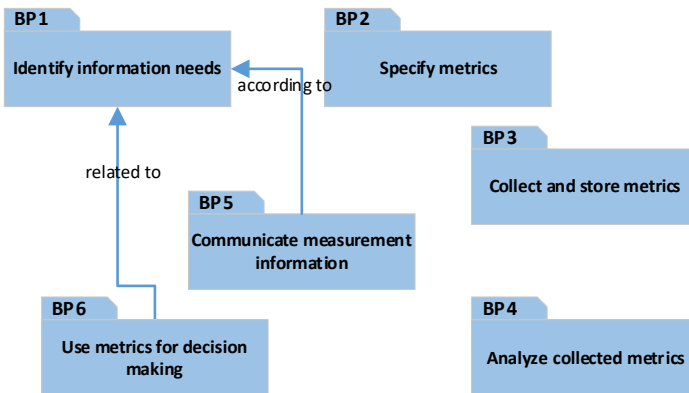
3.30.2.2 Information needs

The information needs should be aligned with the stakeholders and decision makers.

For decision making a review of the analysis of the collected metric is needed.

[MAN.6.RL.2] IF BP.1 is downrated because there is no involvement of the responsible decision makers, BP.5 and BP.6 should be downrated, too.

[MAN.6.RL.3] If the analysis is not reviewed before decision making, BP.4 and BP.6 shall be downrated.



3.30.3 Rating rules with other processes

None.

3.31 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

3.31.1 General Information

As PIM.3 Process Improvement has been not involved in the former VDA scope the experiences about typical pitfalls in applying this process in an assessment are very limited. Therefore, in this guideline the number of rating rules for this process is rather low.

3.31.2 Rating rules within the process

3.31.2.1 Process improvement application

Process improvement is established in the automotive industries and known e.g. as “lessons learned processes” or “continuous improvement”. In the context of Automotive SPICE it is important to act in a structured way.

The Process improvement process is applicable e.g. to:

Achieving a target capability profile

Internal process optimization

Remove particular weaknesses in process performance

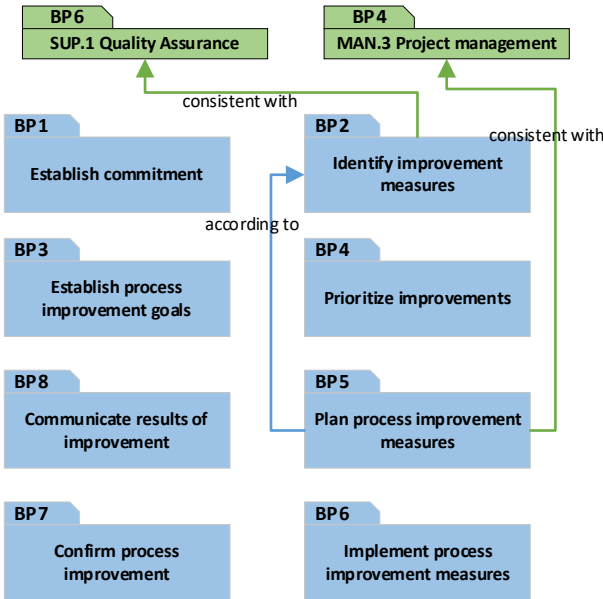
Activities to achieve target performance determined from MAN.6 Measurement

3.31.2.2 Improvement goals

A clear improvement goal shall be committed and defined. Improvement goals shall be communicated to the organization.

[PIM.3.RL.1] If there is no commitment regarding the improvements, BP1 shall not be rated higher than P.

[PIM.3.RL.2] The rating of BP.5 shall be in line to the rating of BP2.



3.31.3 Rating rules with other processes

[PIM.3.RL.3] If the planning and performing of process improvement activities is in scope of a project the rating of the indicator BP5 shall be in line with MAN.3.BP4.

[PIM.3.RL.4] If the identification of improvement measures does not consider aspects of improvements identified in SUP.1.BP6, PIM.3.BP2 should be downrated.

3.32 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

3.32.1 General Information

As REU.2 Management of products for reuse has been not involved in the former VDA scope the experiences about typical pitfalls in applying this process in an assessment are very limited. Therefore, in this guideline the number of rating rules for this process is rather low.

The reuse of components is in the automotive business a common method to establish sustainable development. It is important to reflect this aspect according to the context of a project.

For aspects of reuse of components please also check chapter 2.2.3.

3.32.1.1 Analysis of reused products

The analysis of reused products shall respect the current functional and non-functional requirements, environment, and stakeholder expectations.

As result of the analysis constraints and needed qualification for the product shall be defined.

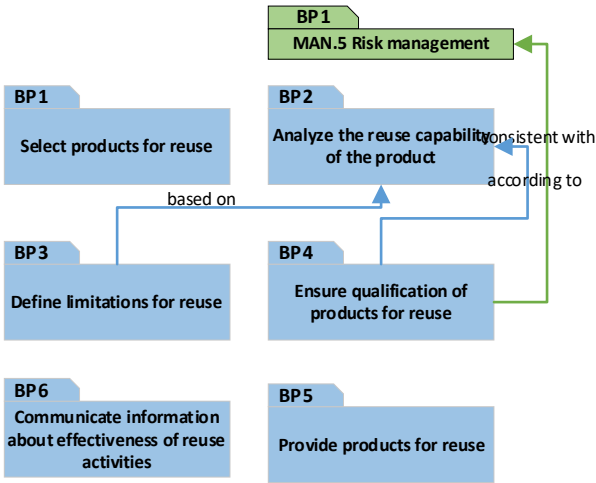
3.32.1.2 Ensure qualification of products for reuse

Unless the qualifying is not finished, the reuse products have to be evaluated if they are relevant for the risk management scope.

3.32.2 Rating rules within the process

[REU.2.RL.1] If the analysis of the reuse capability of the product does not consider architectural constrains, BP.2 shall not be rated higher than P.

[REU.2.RL.2] If the constraints and defined qualification for the reused product is not based on the analysis (BP.2), the rating of BP.3 and BP4 shall be downrated.



3.32.3 Rating rules with other processes

[REU.2.RL.3] If MAN.5.BP1 is downrated due to not considering reuse products in defining risk sources this has to be in line with the rating of REU.2.BP4.

4 Rating guidelines on process capability level 2

The previously described performed process is now implemented in a managed fashion (planned, monitored and adjusted) and its work products are appropriately established, controlled and maintained.

On capability level one process-specific indicators are used to evaluate the extent to which the outcomes of the process are achieved. Assessors regularly use the base practices to assess a project's capability. These are activity-based indicators. In addition, there are information items which are result-oriented indicators. Guidance on possible content of the output work products is documented in Annex B of Automotive SPICE.

At higher capability levels, generic practices and related information items are available as indicators. As the names imply these indicators are not process-specific and have to be used for all processes. Hence, they must be interpreted for each single process individually.

On capability Level 1 the intent is to achieve the purpose of the process. Therefore, the assessor judges whether the result of the process is appropriate with respect to the context of the project including achievement of all outcomes.

On capability level 2 all activities which lead to the purpose of the process and capability level 2 itself (like e.g., reviews) have to be planned and controlled and all resulting work products have to be considered regarding configuration management and quality assurance.

Additionally, on capability level 2, strategies including objectives (e.g., planning goals) for the activities which must be planned for the assessed process have to be defined and documented. Also, requirements for all relevant work products of each process must be defined. These requirements include such information as content and structure (e.g., as table of contents), history, layout, etc. Very often, the requirements for a work product are documented as a work product template including instructions for the usage of the template. If tools are used it should be documented how the tools have to be used, e.g., which fields are mandatory and which optional.

There is a strong dependency between project management (MAN.3) and process attribute 2.1 Process Performance Management. Regarding the process attribute 2.2 Work Product Management there is a strong dependency to quality assurance (SUP.1) and configuration management (SUP.8). For details refer to the chapters on PA 2.1 and PA 2.2 below.

4.1 Process Performance Management (PA 2.1)

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

4.1.1 General information

4.1.1.1 Usage of the term strategy

In the current version of the process assessment model the term “strategy” is used as an information item “process performance strategy” for PA 2.1. It is a key element in terms of having a managed process.

In terms of this assessment model, having a strategy means that all parties involved in achieving the process outcomes have agreed on the methodological approach to achieve the process purpose, and on how to deal with constraints, in order to achieve these process outcomes.

This includes the need for process performance objectives and criteria (e.g. deadlines for activities, maximum effort to consume, see GP 2.1.1) as the basis and “starting point” for a detailed planning. Further, the team will use agreed proceedings in terms of e.g., methodologies as otherwise effective team collaboration, in general, would be unlikely. The latter is expressed by introducing the term “strategy” in in GP 2.1.1.

A strategy may include certain planning aspects, such as bringing activities in the right order and considering certain constraints. This term may not be mistaken for explicit planning documents such as a work breakdown structure, schedule, resource allocation or work packages.

4.1.1.2 Documentation of the strategy

A managed process (represented by Automotive SPICE Level 2) supports the process performance to be predictable, repeatable, or sustainable.

This requires that the strategy to perform the process is not only known by the people acting (as a “virtual” strategy) but needs a representation usable and accessible for others. Therefore, such a strategy needs to be available as documented information (see chapter regarding the understanding of information items, documented information and work products).

It is however not the intent here to promote over-engineering or unnecessary bureaucracy that is not of operational added value. Thus, a strategy does not need to be a specific text document.

For example, a strategy can be evident in, or indicated by

- a) Presentation slides of an organizational unit describing the purpose and objectives of their individual processes and providing sufficient explanation of corresponding proceedings
- b) Existence of tools that enforce certain workflows (e.g., including GUIs with mandatory or restricted edit fields, attributes in a document management, configuration or change request management system)
- c) Automated, or partially automated, workflows implemented by tools and scripts, e.g.
 - automatically generated test result report frame with traceability links to the test case specification
 - build tools including a static software verification step
 - continuous integration approaches
 - continuous delivery approaches
- d) An appropriate media source such as a photo of a whiteboard drawing, video or podcast explaining key elements of the process performance.

A process performance strategy may not necessarily be documented specifically for each process. It’s a common and useful practice to document elements applicable for multiple processes in common documents. For example,

- a joint test strategy for the system testing-oriented processes
- the change request, and configuration management, strategies, as change requests are to be placed against concrete versions of artifacts, or entire product baselines.

- communication strategy aspects of several processes in a project handbook

It remains essential that the strategy must be adhered to and be effective; just documenting a strategy does not necessarily ensure that it is followed and effective. Therefore,

- the necessary comprehensiveness, and detail of information indicated by the examples above is always context-dependent
- further, people interviewed must independently confirm the strategy, which is available as documented information.

The task and responsibility of the assessor is to check whether a strategy exists which is effective in regards of fulfillment of the specific process outcomes and in the concrete context of the assessment of PA 2.1.

4.1.1.3 Planning, monitoring and adjustment of the process performance

For achieving the process attribute PA 2.1 there is more to achieve beyond a strategy. This includes systematic planning, tracking, and adjustment of schedule, effort and resources. Process performance management also includes the planning, monitoring, and adjusting of all activities related to work product management process according to process attribute PA 2.2 (e.g., work product reviews).

An explicit process description is not necessarily required for fulfilling the process performance management attribute PA 2.1 if the outcomes are achieved by accomplishing the generic practices.

Organizations do not need to structure the activities to be planned and monitored in the same way as it is done in the process assessment model (e.g., use of own process naming conventions). Process assessors are responsible for the mapping of process performance related evidence to the right processes and practices of the model.

It is up to the project to define its own structure and use this structure for planning, monitoring, and adjusting of activities. It might even not be reasonable to plan all activities in detail (e.g., planning of all check-in / check-out tasks in the configuration management process).

The generic practices of the process performance management are used to evaluate the capability of a project to plan and monitor activities related to a certain process. The degree to which planning, and monitoring of particular processes are consistent regarding the overall project is the main focus of the Project Management Process (MAN.3), but there is a relationship between the rating of the process performance management attribute PA 2.1 and MAN.3. Therefore, the guidelines defined for Project Management (MAN.3) have to be considered correspondingly for all generic practices of the process performance management attribute PA 2.1 (e.g., granularity of activities, frequency of monitoring activities).

4.1.2 Rating rules within the process attribute

4.1.2.1 Identify the objectives and define a strategy for the performance of the process (GP 2.1.1)

As a basis for a systematic, repeatable, or sustainable process performance management, first the objectives and criteria for the performance of the process need to be identified. Based on this, a strategy for the performance of the process including required activities, tasks, responsibilities, resources, and involved stakeholders must be defined. The defined strategy ensures the proper planning, monitoring, and adjusting of the activities of the corresponding process.

During the identification of objectives and process performance criteria, and for the definition of the strategy the following characteristics shall be considered:

- a) Process scope (including e.g. related objects, issues, disciplines, domains, and sites to be considered)
- b) Needs, objectives, to be satisfied, including criteria to evaluate the achievement of the process performance goals
- c) Process performance criteria (e.g., entry/exit, lifecycle related process achievement goals, frequency of activities)
- d) Options, approach, and methods, tools, and environment to perform the process activities and appropriate to handle the level of product and organizational complexity (e.g., multi-site development, technical system complexity)

- e) Assumptions and constraints (given implicitly by e.g., budget, resources, efforts, milestones, and due dates)
- f) References to relevant regulatory requirements, standards, and customer requirements
- g) Deliverables including completeness criteria (e.g., definition of done) and approach to handle internal and external interfaces (relevant input to / outputs of affected parties, supplier, and customer)
- h) approach for the monitoring of the process performance (e.g., by metrics)
- i) approach for the handling of deviations (e.g., in case of problems and failures during process performance)

Process performance objectives can either be quantitative (e.g., requirements to be implemented for specific releases, maximum/minimum efforts to be spent) or qualitative (e.g., adherence to Automotive SPICE capability level).

If process performance objectives and aspects of the strategy are derived from an existing standard process, the suitability of the standard process, objectives and strategy for the specific project context needs to be considered for the rating.

The definition of objectives, criteria and strategies on an organizational level is not required but may support the achievement of the process performance management outcomes.

The strategy must consider the relevant process outcomes and enable the achievement of the process purpose. The strategy must neither be described in a specific document, nor for each process. Any aggregation of information regarding strategy in common documents (e.g., Master Test Plan, Requirement Engineering Plan, Problem and Change Management Plan, Project Management Plan) shall be considered and rated as a suitable implementation approach of GP2.1.1.

The definition and existence of documented information related to a strategy is not relevant for the rating of PA 1.1 of a certain process.

This leads to the following rating rules and recommendations for the indicator GP 2.1.1:

[PA2.1.RL.1] If a standard process providing a generic strategy approach does not exist, this shall not be used to downrate GP 2.1.1.

[PA2.1.RL.2] If a strategy is not documented as a specific text document, but there is evidence of a strategy known by all relevant parties, this shall not be used to downrate GP 2.1.1.

[PA2.1.RL.3] If the strategy is not described in a single document for each process, but strategies of different processes are combined in common documents, this shall not be used to downrate GP 2.1.1.

[PA2.1.RL.4] If a documented process performance strategy does not exist, this shall not be used to downrate PA 1.1 of the process.

4.1.2.2 Plan the performance of the process (GP 2.1.2)

To ensure a proper planning for the process performance, the following aspects can be covered while considering the identified process performance objectives, criteria, and strategy adequately:

- a) all required activities to fulfill the process outcomes and process purpose are defined
- b) the estimates for the defined process performance attributes are done (e.g., effort, duration, size of work products), estimates are reasonable and reproducible
- c) the sequence of required activities is defined
- d) a schedule including key milestones and required activities is defined and in line with the stakeholder requirements
- e) the schedule includes buffer time (e.g., for bug fixing, vacation)
- f) the schedule of the defined activities shall consider the context related definition of objectives and can include
 - due date, effort, assigned resources, and responsibility for each required activity (typically e.g., for engineering activities), or
 - as percentage or absolute number of a full-time-equivalent's available time for a certain period of time (typically e.g., for project management / supporting and quality related activities)

- g) work product management activities like work product reviews are considered and part of the planning
- h) Communication and meetings are considered and part of the planning
- i) evidence of the planning must be available, e.g.:
 - as part of the project plan,
 - as process-specific document (e.g., meeting plan, audit plan),
 - as backlog, task board, Kanban board, ticket / tracking system, etc.
 - as part of an open-item list

Even though GP 2.1.1 and GP 2.1.2 require the definition of activities to be performed to satisfy the objectives and performance criteria of the process, for PA 2.1 rating it is not mandatory to have a process description in place, if the information regarding process objectives and performance criteria is available elsewhere.

This leads to the following rating rules and recommendations for the indicator GP 2.1.2:

[PA2.1.RL.5] If the determination of critical dependencies of activities and work packages is not considered in the process performance planning, then GP 2.1.2 shall be downrated.

[PA2.1.RL.6] If supporting activities are not planned as explicit activities but are planned as percentage or absolute number of hours over a certain period of time, then GP 2.1.2 shall not be downrated.

[PA2.1.RL.7] If no process description including required activities and tasks is available, but all aspects above are covered, then GP 2.1.2 shall not be downrated.

[PA2.1.RL.8] If the indicator for identifying the objectives and defining the strategy for the performance of the process (GP 2.1.1) is downrated due to missing suitability to achieve the process outcomes, then GP 2.1.2 shall downrated.

4.1.2.3 Determine resource needs (GP 2.1.3)

The following aspects need to be covered in the project and adequately documented:

- a) The need of human, as well as physical and material resources to perform dedicated work packages is determined based on the planned activities. The needs have to include activities related to process performance management and work product management
- b) Responsibilities (e.g., RACI-Definition for activities), commitments and authorities (e.g., access rights, budget release, release of work products) to perform the process activities of the project need to be defined, assigned, communicated, and agreed.
- c) Responsibilities and authorities to verify process work products need to be defined, assigned, communicated, and agreed. E.g., for verification measures of work products, the responsibility and authority for the verification must be defined (e.g., senior engineer, independent quality assurance, management).
- d) Needs for process performance experience, knowledge and skills are defined. Needs can either be process-specific, product-specific or project-specific (e.g., methods / tool skills, needed algorithms, customer flash tool).

In distinction to GP 3.1.2 all definitions for responsibilities and authorities can be made specifically for the project without considering a standard process and roles.

This leads to the following rating rules and recommendations for the indicator GP 2.1.3:

[PA2.1.RL.9] If the determination of resource needs only relates to human resources (while also physical or material resources need to be considered), then GP 2.1.3 shall not be rated higher than P.

[PA2.1.RL.10] If the aspects above are adequately covered without considering role definitions of a standard process, then GP 2.1.3 shall not be downrated.

[PA2.1.RL.11] If the indicator for planning the performance of the process (GP 2.1.2) is downrated due missing activities related to process performance management and work product management, then GP 2.1.3 shall be downrated.

4.1.2.4 Identify and make available resources (GP 2.1.4)

The identification and provision of resources shall cover the following aspects:

- a) The individuals / people required for the process performance, process performance management and work product management are identified, made available, allocated and used. Resource planning is comprehensible (e.g., rate of utilization is transparent, vacation and trainings are considered, procedures for planning in matrix organization or distributed development are defined). A comparison of needed human resources versus allocated resources should be available and be maintained during project life cycle (see also GP 2.1.5)
- b) The physical and material resources required for the process performance, process performance management and work product management (e.g., tool licenses, samples, test equipment) are made available and used. A comparison of physical and material resources versus allocated resources should be available and be maintained during project life cycle (see also GP 2.1.5)
- c) The individuals performing and managing the process and work products are qualified by training, mentoring, or coaching to execute their responsibilities. A qualification fit/gap analysis should be performed. Necessary qualification measures are planned and performed in time, according to the needs of the project.
- d) The information necessary to perform the process is made available for all individuals performing and managing the processes and work products (e.g., manuals, project wiki).

This leads to the following rating rules and recommendations for the indicator GP 2.1.4:

[PA2.1.RL.12] If the identification and provision of resources only relate to individuals (while also physical or material resources need to be considered), then GP 2.1.4 shall not be rated higher than P.

[PA2.1.RL.13] If the indicator for determination of resource needs (GP 2.1.3) is downrated due to inadequately defined resource needs, then GP 2.1.4 shall be downrated.

4.1.2.5 Monitor and adjust the performance of the process (GP 2.1.5)

In order to monitor the performance of the process against the plans and to adjust the performance of the process, the following aspects have to be considered:

- a) the process is performed as planned and according to the strategy
- b) Estimates for human, material and physical resources needs as well as needs for skills, knowledge and experience are still matching to the projects needs
- c) data regarding the defined process performance and related process performance quality criteria is continuously collected
- d) actual data is continuously compared with planned values (this means also that the granularity of planned and actual data is similar), e.g.
 - by comparing actual results in given time/duration/effort
 - by comparing booked effort per cost center to planned values
- e) the comparison between planned and actual data should:
 - show the current state of progress,
 - ensure that planned results are achieved, or
 - identify deviations from the plan,
 - be performed in an adequate frequency (e.g., in case of delivery every eight weeks and monitoring and comparison every four weeks, a higher frequency would be adequate)
- f) documentation of monitoring activities, e.g., as:
 - status report
 - status meeting minutes
- g) process performance issues must be identified on the basis of deviations
- h) in case of identified deviations regarding the defined process performance quality criteria (e.g., due dates, effort estimations, resource usage)
 - deviations are analyzed and root causes determined, and
 - either corrective measures to align performance with plans must be taken or
 - plans must be adapted in such way that plan changes are still in line with the stakeholder requirements

This leads to the following rating rules and recommendations for the indicator GP 2.1.5:

[PA2.1.RL.14] If the levels of granularity of planning and monitoring do not match in absence of a consistent mapping in between, then GP 2.1.5 shall be downrated.

[PA2.1.RL.15] If the chosen frequency of monitoring activities is not capable to identify deviations versus plan in time, then GP 2.1.5 shall be downrated.

[PA2.1.RL.16] If the indicator for planning the performance of the process (GP 2.1.2) is downrated due to incomplete planning of activities and work packages or missing determination of resource needs, then GP 2.1.5 shall be downrated.

4.1.2.6 Manage the interfaces between involved parties (GP 2.1.6)

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

Managing the interfaces should cover the exchange of information and work products and should include the following aspects:

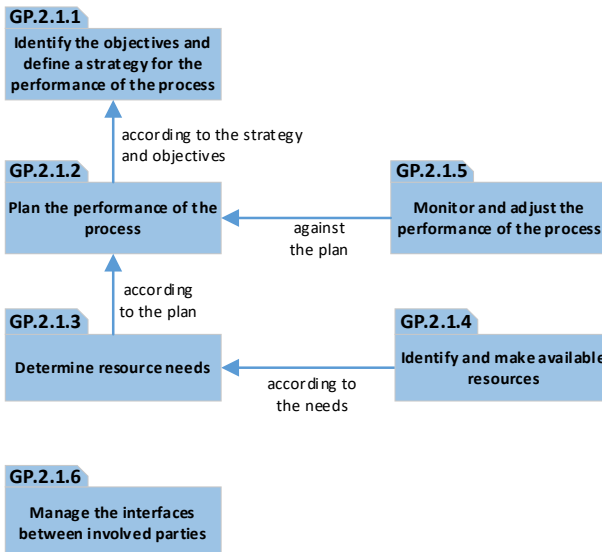
- a) Responsibilities of the involved parties are assigned. It should be defined e.g.,
 - who delivers or communicates
 - what is the subject of delivery and communication (e.g., work products, documents, information, escalation)
 - who is the receiver
- b) Interfaces between the involved parties are managed. Evidence could be e.g.,
 - communication and meetings are planned / set up on a regular basis
 - participants for the meetings are defined (depending on responsibilities, tasks, or processes)
 - the communication path is defined (e.g., protocol, link to a baseline)
 - the trigger is defined (push/pull)
 - distribution lists are established (e.g., for meeting minutes)

- c) Communication between the involved parties is established, managed, maintained and effective in a repeatable fashion. Evidence could be e.g.,
- regular or planned meetings take place as planned
 - interfaces are used as defined
 - type, mechanism, and media of communication is defined, e.g., active (mail or status transition in a tool / database) and/or passive (information just made available)
 - communication is documented (e.g., agenda, meeting minutes, open item lists)
 - follow-up on open items identified is assured

4.1.3 Rating consistency

4.1.3.1 Rating consistency within PA 2.1

The following figure shows relationships among GP 2.1.x generic practices:



4.1.3.2 Rating consistency with other processes and practices

During the assessment and in terms of the consistency of the assessment ratings, it can be useful to consider the dependencies between Process Performance Management attribute PA 2.1 and practices of other processes.

Assessment ratings on related practices and process attributes have to be performed on the same insight. Accordingly, evidence obtained during the assessment needs to be analyzed regarding potential relevance for other dependent practices and process attributes. Similar weaknesses identified in generic practices of PA 2.1 of several processes might be considered also in the rating of the corresponding practices of PA 1.1 and vice versa. Significant rating differences across dependent processes and process attributes (e.g., with more than one rating scale) might be an indicator of an inconsistent rating.

Dependencies of the Process Performance Management attribute PA 2.1 and its generic practices to other processes are:

For the rating of PA 1.1 of MAN.3 the assessor should also consider the ratings of PA 2.1 of the other processes. Several down ratings in PA 2.1 might be an indicator of a weak implementation of the project management process.

Similar consistency evaluations might be useful between the ratings of generic practices of PA 2.1 of several processes with the rating of the corresponding base practices of MAN.3 (e.g., BP.4, BP.5, BP.6, BP.7 and BP.8).

For the rating of GP2.1.6 (Manage the interfaces between involved parties) the assessor should also consider the ratings on corresponding base practices of other processes than MAN.3 related to communication (e.g., SYS.2.BP6, SWE.2.BP5, SYS.4.BP5, SWE.6.BP5, MLE.1.BP6, HWE.1.BP6, etc.).

4.2 Work Product Management (PA 2.2)

The work product management process attribute is a measure of the extent to which the work products produced by the process are appropriately managed.

4.2.1 General information

Relevant work products of the process are those that are required to fully achieve capability level 1, and additionally, evidence (work products) to prove successful implementation of the process attributes 2.1 and 2.2. A work product may not only be a document but could also be a record or database entry in a tool (e.g., change requests or problem reports implemented in a workflow tool are also work products).

Not included in the term “work product” are all process-related documents like e.g., process descriptions, procedures, method descriptions, or role descriptions. Any weaknesses in handling these process assets that are not related to the content (e.g., improper versioning) must not be reflected in the process attribute 2.2 of the process under investigation. However, if organizational process documents are available, they can support the implementation of process attribute 2.2.

Instead of work products, output information items are defined to describe the required content of output work products in the Automotive SPICE PAM 4.0. Each of the output information items is associated with one or more outcomes of the process and further detailed by information item characteristics in Annex B of the PAM. These information items and their characteristics can be used as a starting point for considering whether, given the context, the observed work products are contributing to the intended purpose of the process, and are thus relevant for rating the work product management attribute.

4.2.2 Rating rules within the process attribute

4.2.2.1 Define the requirements for the work products (GP 2.2.1)

Work product requirements include:

- a) Requirements defining content and structure, e.g.:
 - Information regarding the structure such as layout, history, table of contents
 - Technical content (e.g., requirement specifications, architectural descriptions)
 - Project content (e.g., plans, minutes, open point lists)
 - Guidelines (e.g., programming or modelling guidelines)
 - Standards
- b) Appropriate review and approval criteria, e.g.:
 - Definition whether the work product needs to be explicitly reviewed or only implicitly reviewed by distributing them and accepting them in case of no feedback (e.g., minutes, open-point-lists, reports etc.).
 - Definition regarding review method, review coverage (including justification), review frequency (including justification), and review participants
- c) Quality criteria (based on aspects a) or b)).

Very often, the requirements for a work product are documented as a work product template or checklist. However, defining templates or checklists is not necessarily required by the work product management attribute as long as all aspects above are adequately documented.

This leads to the following rating rules and recommendations for the indicator GP 2.2.1:

[PA2.2.RL.1] If no template or checklist exists for the work product, but the aspects of content and structure, review and approval, and quality criteria are adequately documented, the indicator GP 2.2.1 must not be downrated.

[PA2.2.RL.2] If standard work product templates provided by a standard process are available, but the project has defined a

project-specific solution that is effective, the indicator GP 2.2.1 must not be downrated.

[PA2.2.RL.3] If standard work product templates provided by a standard process are available and used by the project, but do not fit for the purpose of the project, the indicator GP 2.2.1 shall be downrated.

4.2.2.2 Define the requirements for storage and control of the work products (GP 2.2.2)

Certain requirements regarding storage and control have to be defined for all relevant work products. These requirements have to be set-up for each identified work product (see also SUP.8.BP2 and chapter 7.2.3.2).

The requirements for storage and control should cover at least these minimal required aspects:

- a) Identification of work products
- b) Naming convention
- c) Ownership
- d) Access rights (at least read and write permission)
- e) Work product status model, including states and transitions, workflow, approval, and release procedure
- f) Versioning rules (including baselining mechanisms depending on the work product type)
- g) Storage media (e.g., project drive, configuration management tool)
- h) Distribution channels (communication mechanisms for releases and changes)

The expectations for a status model and workflow (see aspect e) above) definition cover these aspects:

- For work products which require a status attribute, the status model and workflow are defined, including criteria for status changes, and relevant stakeholder together with their responsibility and authorization, etc.
- The work product status transitions follow the workflow to a final status, and it is tracked accordingly. There might be more than one final status (e.g., closed, rejected, cancelled), but it has to be ensured that one of them is

always reached (e.g., there is a status “solved” but the status model defines an additional step “closed” that will usually not be reached).

- Work products with a very simply status definition (e.g., meeting minutes) do not require a complete status model with workflow, approval, and release procedure.

This leads to the following rating rules for the indicator GP 2.2.2:

[PA2.2.RL.4] If the requirements for storage and control do not cover the minimal required aspects, the indicator GP 2.2.2 shall be downrated.

[PA2.2.RL.5] If the requirements for storing and controlling work products do not cover versioning and storage requirements, the indicator GP 2.2.2 must not be rated higher than P.

[PA2.2.RL.6] If the definition of a status model for a relevant work product with a non-trivial status definition lack definitions of workflow, criteria for status changes, stakeholder and their authorization, the indicator GP 2.2.2 shall be downrated.

4.2.2.3 Identify, store and control the work products (GP 2.2.3)

All identified work products must be stored and controlled (indicator GP 2.2.3) according to their requirements (indicator GP 2.2.2). Because of this dependency, the corresponding rules are defined.

[PA2.2.RL.7] If the indicator for defining requirements for storage and control of the work products (GP 2.2.2) is downrated, the indicator GP 2.2.3 must not be rated higher.

4.2.2.4 Review and adjust work products (GP 2.2.4)

Work product reviews have to be performed against defined work product review criteria (see GP 2.2.1) in accordance with the planning (see PA 2.1). The execution of work product reviews including results has to be demonstrable. This does not necessarily require a formal review including dedicated review record but can also be a less formal approach like walk-through, or pair-programming according to the quality assurance strategy (see PA 2.1 for SUP.1). However, it is required that the following aspects must be demonstrable:

- a) Review information:
 - Work product under review (including name and version to ensure proper identification)
 - Date of the review
 - Name(s) of reviewer(s)
 - Review findings, if they are not immediately solved in the review (e.g., in pair programming)
 - Review result (e.g., “Passed”, “Conditionally Passed”, “Failed / Re-review required”)
 - Used review and approval criteria
- b) Handling of review findings:
 - A procedure for handling of review findings has to be defined
 - Review findings have to be monitored and tracked until resolution

For the rules 9 and 10 below, the most relevant work products are defined as those that are required to fully achieve capability level 1.

This leads to the following rating rules and recommendations for the indicator GP 2.2.4:

[PA2.2.RL.8] If the proof of work product reviews does not cover all of the required demonstrations of aspects, the indicator GP 2.2.4 shall be downrated.

[PA2.2.RL.9] If the proof of work product reviews for the most relevant work products does not cover the name and version of the work product under review, review findings (unless immediately solved), and the used review and approval criteria, the indicator GP 2.2.4 must not be rated higher than P.

[PA2.2.RL.10] If work product review findings are not resolved for the most relevant work products, the indicator GP 2.2.4 must not be rated higher than P.

[PA2.2.RL.11] If work product reviews are demonstrable according to all aspects above but are not explicitly documented in a formal review record, the indicator GP 2.2.4 must not be downrated.

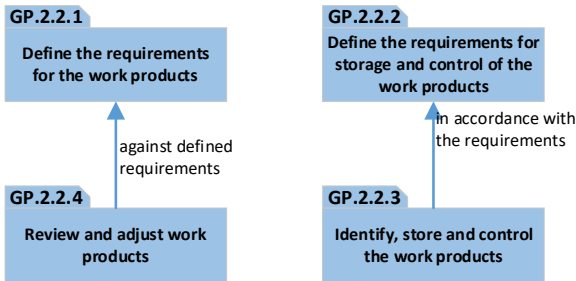
[PA2.2.RL.12] If the indicator for defining requirements for the work products (GP 2.2.1) is downrated due to non-appropriate

review and approval criteria, the indicator GP 2.2.4 shall be downrated.

4.2.3 Rating consistency

4.2.3.1 Rating consistency within PA 2.2

The following figure shows relationships among GP 2.2.x generic practices:



The generic practices of capability level 2 can be grouped into two main topics. The first one covers requirements, quality criteria, review, and adjustment of all relevant work products of the corresponding process (GP 2.2.1 & GP 2.2.4), whereas the second one covers the storage and control of those work products (GP 2.2.2 & GP 2.2.3).

4.2.3.2 Rating consistency with other processes and practices

There is a strong dependency between quality assurance (SUP.1) respectively configuration management (SUP.8) and process attribute PA 2.2 “Work product Management”. Thus, if PA 2.2 is downrated for several processes, this should be in line with the rating of SUP.1 and SUP.8.

If the indicator for defining requirements for the work products (GP 2.2.1) is downrated for several processes due to non-appropriate review and approval criteria, the indicator SUP.1.BP2 should reflect that.

If the indicator for defining requirements for the storage and control of work products (GP 2.2.2) is downrated for several processes for reasons other than the lack of status model and workflow definitions, the indicator SUP.8.BP2 should reflect that.

The rating of the indicator GP 2.2.3 of all processes should be in line with the ratings of the indicators SUP.8.BP3, SUP.8.BP4, and SUP.8.BP5, respectively.

The rating of the indicator GP 2.2.3 of all processes should be in line with the ratings of the indicators SUP.10.BP1, SUP.10.BP3, and SUP.10.BP6, respectively.

The rating of the indicator GP 2.2.4 of all processes should be in line with the rating of the indicator SUP.1.BP3.

The rating of the indicator GP 2.2.4 should be in line with the rating of the indicator of the corresponding process for ensuring consistency of work products, if review is there used as a primary means for establishing consistency (SYS.2. BP5, SYS.3. BP3, SYS.4. BP5, SYS.5. BP4, SWE.1. BP5, SWE.2.BP3, SWE.3.BP3, SWE.4.BP4, SWE.5.BP6, SWE.6.BP4, SUP.8.BP7).

5 Rating guidelines on process capability level 3

The previously described Managed process is now implemented using a defined process that is capable of achieving its process outcomes.

On capability level 2 all projects may use “their“ own process as long as the requirements of Automotive SPICE are fulfilled.

On capability level 3 the projects have to use a standard process. A possibility to cover variations between projects is to describe tailoring guidelines. This derived process is the so-called “defined” process. The defined process has to cover all activities and work products of capability level 1 and 2 for the assessed project.

Large organizations would have problems with only one standard process. The organization may define several different standard processes (e.g., one standard process for each development site, or one standard for each business unit). The other possibility to cover variations between projects is the afore-mentioned description of tailoring guidelines. Based on predefined criteria the process may be tailored to the needs of the project.

Exceptionally waivers for the standard process may be used (which should not be the rule), assessors should check whether these exceptions have a rationale and are approved by appropriate organizational roles.

It has to be kept in mind that the advantage of organizational processes is to standardize the approach to e.g.:

- establish processes known by the stakeholders
- establish interfaces to facilitate cooperation (also between different locations)
- facilitate introduction of new personnel or exchange personnel between projects
- facilitate reuse of assets and work products
- establish benchmarking

The aim of establishing processes might get missed if there are too many variations of the processes. This should be reflected by the assessment result.

5.1 Process Definition (PA 3.1)

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.

5.1.1 General information

The process defined is organization wide and no longer project specific. This includes at least

- a developed, established and maintained standard process including tailoring guidelines (GP 3.1.1)
- required competencies, skills, and experience for the identified roles performing the standard process (GP 3.1.2)
- required physical and material resources and process infrastructure needs for performing the standard process GP 3.1.3)
- suitable methods and required activities for monitoring the standard process (GP 3.1.4)

Each of these aspects has to be rated only in the respective GP.

5.1.2 Rating rules within the process attribute

5.1.2.1 Establish and maintain the standard process (GP 3.1.1)

GP 3.1.1 covers the definition and maintenance of the standard process including tailoring guidelines.

Definition of the standard process

In order to define the standard process, its scope, purpose and intended use must be identified and correspondingly documented. The fundamental process elements such as process activities including detailed descriptions, required inputs and expected outputs including corresponding entry and exit criteria for the process as well as for the process activities, must be incorporated into the defined process.

[PA3.1.RL.1] If scope, purpose and intended use are missing in the standard process definition, the indicator GP 3.1.1 must not be rated F.

[PA3.1.RL.2] If process activities including descriptions are missing in the standard process definition, the indicator GP 3.1.1 must not be rated higher than P.

[PA3.1.RL.3] If required inputs or expected outputs are missing in the standard process definition, the indicator GP 3.1.1 must not be rated higher than P.

[PA3.1.RL.4] If entry and exit criteria are missing in the standard process definition, the indicator GP 3.1.1 must not be rated F.

Additionally, the sequence and interaction of process activities inside one process as well as the sequence and interaction of the process to other processes must be identifiable. This might also include parallel or iterative sequencing of activities, which are synchronized by e.g., work product completion.

[PA3.1.RL.5] If the sequence and interactions of process activities within the process or to other processes is not identifiable, the indicator GP 3.1.1 must not be rated higher than P.

[PA3.1.RL.6] If the sequence and interactions of process activities within the process or to other processes is not explicitly documented as such, but is identifiable (e.g., by work product status and entry/exit criteria), the indicator GP 3.1.1 must not be downrated.

In order to support the execution of the process, guidance, procedures, method descriptions, and/or templates should be provided as needed.

[PA3.1.RL.7] If required guidance in terms of procedures, method descriptions, or templates for the process is not provided, the indicator GP 3.1.1 must be downrated.

[PA3.1.RL.8] If templates for the expected outputs are not explicitly provided, but corresponding detailed requirements regarding the expected content are given, the indicator GP 3.1.1 must not be downrated.

Process performance roles must be identified and assigned to the standard process activities including their level of involvement, responsibilities, and authorities (e.g., by RACI-matrix). Here in GP

3.1.1, only the identification of the roles including their involvement is covered, whereas the detailed role descriptions are handled in GP 3.1.2 (see next section 5.1.2.2).

[PA3.1.RL.9] If process performance roles are not identified and assigned to standard process activities, the indicator GP 3.1.1 must not be rated higher than P.

[PA3.1.RL.10] If the kind of involvement of the process roles in the process activities is not defined, the indicator GP 3.1.1 must be downrated.

[PA3.1.RL.11] If role definition details like competencies, skills, experience, or qualification methods are missing, the indicator GP 3.1.1 must not be downrated.

The responsibilities for process development and maintenance (e.g., process owner, process developer) need to be defined.

[PA3.1.RL.12] If the process owner for the standard process is not defined, the indicator GP 3.1.1 shall be downrated.

Maintenance of the standard process

The defined standard process needs to be continuously maintained according to corresponding feedback from monitoring the deployed process (see also GP 3.2.4 for corresponding input), and adapted process requirements (standards, regulations, laws, changed/new infrastructure, etc.). The maintenance should be documented in change requests and corresponding process version numbering. The validity of process versions needs to be defined, which includes:

- Date for obligatory use of the latest version of the standard process for all upcoming projects
- Handling of usage of new version of the standard processes or new process elements for running projects (e.g., not applicable for projects at a stage later than x)
- Mechanism to ensure availability of previous process versions

This leads to the following rating rules:

[PA3.1.RL.13] If the standard process does not have a unique version number, the indicator GP 3.1.1 must not be rated F.

[PA3.1.RL.14] If changes from one version to another version of the standard process are not documented and identifiable, the indicator GP 3.1.1 must be downrated.

[PA3.1.RL.15] If it is unclear when a new version of the standard process will be mandatory for new projects, or by when ongoing projects will have to switch to the new standard process, the indicator GP 3.1.1 must be downrated.

[PA3.1.RL.16] If older versions of the standard process are not available, but still to be used in running projects, the indicator GP 3.1.1 must be downrated.

Tailoring of the standard process

Deployment can be done with or without tailoring of the standard process, which is supported by corresponding tailoring guidelines (see also GP 3.2.1). Tailoring can be performed through different proceedings such as deleting, adding or selection between different elements of the process based on predefined criteria. Additionally, the responsibility for tailoring and corresponding approval must be defined.

[PA3.1.RL.17] If the tailoring guidelines do not include predefined criteria for tailoring, the indicator GP 3.1.1 shall be downrated.

[PA3.1.RL.18] If the tailoring guidelines do not include the proceeding for tailoring, the indicator GP 3.1.1 shall be downrated.

[PA3.1.RL.19] If the tailoring guidelines do not include the responsibility for tailoring and corresponding approval, the indicator GP 3.1.1 shall be downrated.

If there is no tailoring defined, and the standard process is used as the defined process, the following rules must be considered:

[PA3.1.RL.20] If the standard process is either not suitable for the project or cannot be effectively applied by the project, the indicator GP 3.1.1 must not be rated F.

[PA3.1.RL.21] If deviations from the standard process are approved as an exceptional case by the corresponding process

roles (e.g., by a waiver), the indicator GP 3.1.1 must not be downrated.

In case of several approved similar deviations from the standard process, the process either needs to be reworked, or an additional, corresponding tailoring guideline including criteria must be added.

[PA3.1.RL.22] If similar deviations from the standard process are regularly approved (e.g., by waivers) without updating the standard process and/or tailoring guideline, the indicator GP 3.1.1 must be downrated.

5.1.2.2 Determine the required competencies (GP 3.1.2)

The standard process identifies and assigns required process performance roles to process activities including their level of involvement, responsibilities, and authorities (see GP 3.1.1, including corresponding rating).

[PA3.1.RL.23] If the involvement of the process roles regarding responsibilities, or authorities in standard process activities is not defined, the indicator GP 3.1.2 must not be downrated.

But the process roles need to be described in more detail including required competencies, skills, and experience, which is covered by the indicator GP 3.1.2. Furthermore, appropriate qualification methods need to be determined, maintained, and made available.

[PA3.1.RL.24] If process roles do not have a textual description, the indicator GP 3.1.2 must be downrated.

[PA3.1.RL.25] If required competencies or required skills are missing for the defined process roles, the indicator GP 3.1.2 must not be rated higher than P.

[PA3.1.RL.26] If required experience is missing for the defined process roles, the indicator GP 3.1.2 must not be rated F.

[PA3.1.RL.27] If qualification methods are either not determined, or not maintained, or not available for the defined roles, the indicator GP 3.1.2 must not be rated F.

5.1.2.3 Determine the required resources (GP 3.1.3)

Requirements for human resources are covered by GP 3.1.2.

[PA3.1.RL.28] If requirements for human resources are not determined, the indicator GP 3.1.3 must not be downrated.

But other non-human resources like physical and material resources as well as process infrastructure needs must be determined, which is covered by GP 3.1.3. This includes the definition and description of the used tools (including qualification, if relevant, e.g., for safety critical use) and infrastructure, methods and responsibilities to ensure that the needed work environment is available for the projects (e.g., licenses), or also samples.

[PA3.1.RL.29] If required tools are not defined, the indicator GP 3.1.3 must not be rated higher than P.

[PA3.1.RL.30] If tool qualification is not evident in a safety critical context, the indicator GP 3.1.3 must not be rated F.

[PA3.1.RL.31] If required samples are not defined, the indicator GP 3.1.3 must be downrated.

5.1.2.4 Determine suitable methods to monitor the standard process (GP 3.1.4)

In order to monitor effectiveness, suitability and adequacy of the standard process in a systematic and defined way, corresponding methods and required activities need to be determined. Successful process compliance checks or internal audits/assessments can be evidence for the suitability of a process, whereas a lack in process compliance for a majority of projects can be evidence that the process is not suitable. Lessons learned or retrospective meetings can be used to get process feedback. Feedback should be documented in a defined way, analyzed, and taking in account for process development. In addition, there should be a defined and well-known way for project staff to give process feedback to the responsible process development organization.

Furthermore, metrics could be defined to monitor key figures of the standard process (e.g., number of review findings, failures found after a dedicated test step, ideal discovery to actual discovery of failures, effort variance). They could be observed in relation to industrial standards, other standard processes of the company or as a trend for a single process.

[PA3.1.RL.32] If the determined standard process monitoring methods do neither address effectiveness, nor suitability, nor adequacy, the indicator GP 3.1.4 must not be rated higher than P.

[PA3.1.RL.33] If the determined standard process monitoring methods include only project staff feedback and lessons learned, the indicator GP 3.1.4 must not be rated F.

[PA3.1.RL.34] If the determined standard process monitoring methods are only of qualitative nature, but still appropriate regarding effectiveness, suitability, and adequacy, the indicator GP 3.1.4 must not be downrated.

5.1.3 Rating consistency within the process attribute

No explicit consistency dependencies were identified within this process attribute, and therefore, no corresponding rating rules were defined.

5.2 Process Deployment (PA 3.2)

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.

5.2.1 General information

The rating of process attribute 3.2 should reflect the degree to which the process is using the standard process under consideration of the tailoring guidelines.

5.2.2 Rating rules within the process attribute

5.2.2.1 Deploy a defined process that satisfies the context specific requirements of the use of the standard process (GP 3.2.1)

The deployment of a defined process should include

- a) the project specific selection and/or tailoring from the standard process using the defined tailoring guideline and criteria. The decisions made and the rationale for the decisions need to be documented.
- b) the verification that the defined process is conformant with standard process requirements and accordingly applied in the project. This has to be done by an authorized role, e.g., process owner, process group, quality management or quality assurance. Evidences of the verification or a final release of the defined process need to be documented.

[PA3.2.RL.1] If the defined process is not selected, documented and verified according to the tailoring guideline and corresponding criteria, the indicator GP 3.2.1 shall be downrated.

5.2.2.2 Ensure required competencies for the defined roles (GP 3.2.2)

Roles, responsibilities and authorities for performing the defined process are assigned and communicated.

Necessary skills and competencies can either be process-specific (e.g., role or standard tool trainings) or project-specific (e.g., customer flash tool).

Ensuring required competencies includes:

- a) The assurance of appropriate skills and competencies for assigned personnel. Evidence that the assigned persons have the required qualifications (e.g. qualification records) should be available. The qualification has to be in line with the skills and competencies defined in GP 3.1.2 for performing the standard process.
- b) If gaps in skills and competencies shown, adequate qualification measures should be defined and monitored.
- c) The availability of suitable qualification for those performing the defined process. Availability ensures that project members are qualified in time, to perform the defined processes in the project.
[PA3.2.RL.2] If the roles, responsibilities and authorities are not assigned, the indicator GP 3.2.2 must not be rated F.

[PA3.2.RL.3] If no evidence that the assigned persons have the required qualification is available, the indicator GP 3.2.2 must not be rated higher than P.

[PA3.2.RL.4] If the necessary skills and competencies are not available in time, the indicator GP 3.2.2 must not be rated F.

Rationale: If a qualification measure is planned for the future, but the qualification is required today, the qualification is still missing.

Ensuring the availability, allocation and usage of the project stuff and related information includes that

- a) the required human resources within the project are made available, allocated and used.
 - a) In addition to GP 2.1.4:
 - the resources need to be available according the roles and qualification defined in the standard process considering the project specific definitions.
 - The availability of the resources needs to be ensured, taking into account that resources may be also used by other projects of the organization.

b) Related information about human resources should be available and include

- Expert knowledge from previous projects and training materials or
- Models for resource estimation based on the recording of needed resources of former projects

[PA3.2.RL.5] If related information about human resources are not available, the indicator GP 3.2.2 shall be downrated.

[PA3.2.RL.6] If the availability and usage of the human resources is not measured and monitored, the indicator GP 3.2.3 must not be rated F.

[PA3.2.RL.7] If the availability and usage of the process improvement related resources is not measured and monitored, the indicator GP 3.2.3 must not be downrated.

5.2.2.3 Ensure required resources to support the performance of the defined process (GP 3.2.3)

Ensuring the required physical and material resources, process infrastructure and work environment includes that

- a) The required resources, infrastructure and work environment, according to standard process and the project specific definition is available.
- b) Organizational support to effectively manage and maintain the resources, infrastructure and work environment is available and known by the project members.
 - Resources for the support are planned by the organization.
 - Availability of licenses is checked regularly.
 - Information about anticipated or planned process infrastructure changes, e.g. new tool chains, shall be made available to the projects.
- c) Infrastructure and work environment is used and maintained. If updates or new versions of the work environment are available, the handling has to be planned in coordination with the project.

[PA3.2.RL.8] If the organizational support is not adequate to effectively manage and maintain the resources, the indicator GP 3.2.3 must not be rated F.

The availability and usage of the resources are measured and monitored.

[PA3.2.RL.9] If the availability and usage of the resources is not measured and monitored, the indicator GP 3.2.3 must not be rated F.

5.2.2.4 Monitor the performance of the defined process (GP 3.2.4)

The defined process should ensure that

- a) Information required to understand the behavior and evaluate the suitability, effectiveness and adequacy of the defined process are identified based on the definitions of GP 3.1.4. Information about process performance may be qualitative or quantitative.
- b) Information is collected and analyzed to understand the behavior of the process, and to evaluate the suitability, effectiveness and adequacy of the defined process. Frequency and approach for collecting and analyzing is defined on project and process level.
- c) Results of the analysis and evaluation are used to identify where continual improvement of the standard and/or defined process can be made. Results should be documented and made available to all affected parties.

[PA3.2.RL.10] If the collected information is not analyzed to understand the behavior and evaluate the suitability, effectiveness and adequacy of the defined process, the indicator GP 3.2.4 must not be rated higher than P.

[PA3.2.RL.11] If the analyzed information for the suitability, effectiveness and adequacy of the defined process is not made available to all affected parties, the indicator GP 3.2.4 must not be rated F.

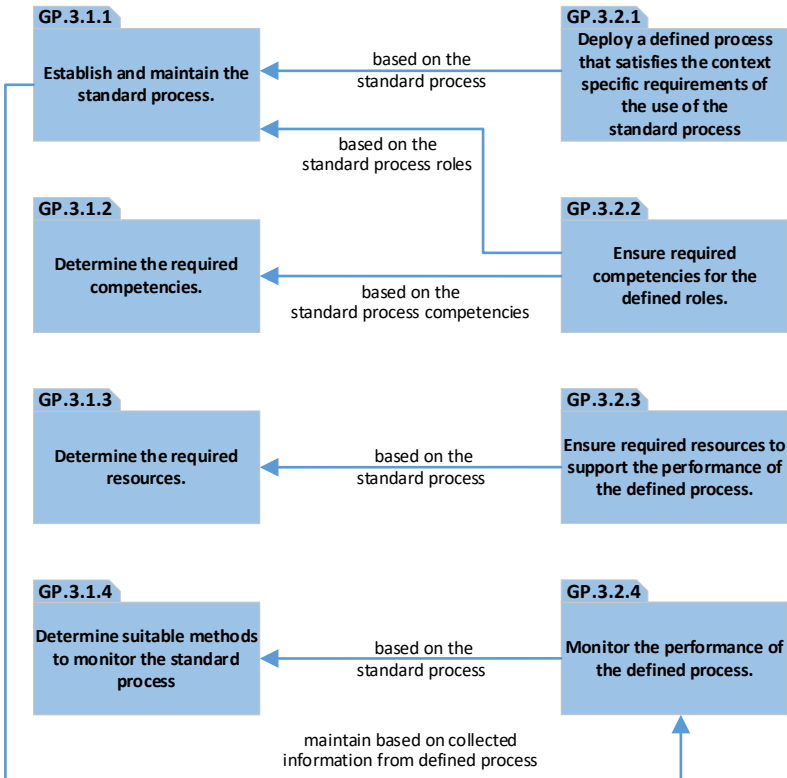
5.2.3 Rating consistency within the process attribute

No explicit consistency dependencies were identified within this process attribute, and therefore, no corresponding rating rules were defined.

5.3 Rating consistency

5.3.1 Rating rules within capability level 3

The following figure shows relationships among generic level 3 practices:



GP 3.1.1 Establish and maintain the standard process

[PA3.1.RL.12] If the indicator 3.2.4 is downrated due to missing or inadequate information from monitoring the performance of the process, the indicator GP 3.1.1 must not be rated F.

GP 3.2.1 Deploy a defined process that satisfies the context specific requirements of the use of the standard process.

[PA.3.2.RL.13] If the indicator GP 3.1.1 is downrated due to missing or inadequate definition of the standard process, the indicator GP 3.2.1 shall be downrated.

GP 3.2.2 Ensure required competencies for the defined roles.

[PA3.2.RL.14] If the indicator GP 3.1.1 is downrated due to missing or inadequate definitions of roles, responsibilities and authorities, the indicator GP 3.2.2 shall be downrated.

[PA3.2.RL.15] If the indicator GP 3.1.2 is downrated due to missing or inadequate definitions of competencies, skills or experiences, the indicator GP 3.2.2 shall be downrated.

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

[PA3.2.RL.16] If the indicator GP 3.1.3 is downrated due to missing or inadequate definitions of resources, the indicator GP 3.2.3 shall be downrated.

GP 3.2.4 Monitor the performance of the defined process.

[PA3.2.RL.17] If collecting and analyzing the required information is not performed according to the defined methods and activities (GP 3.1.4), the indicator GP 3.2.4 shall be downrated.

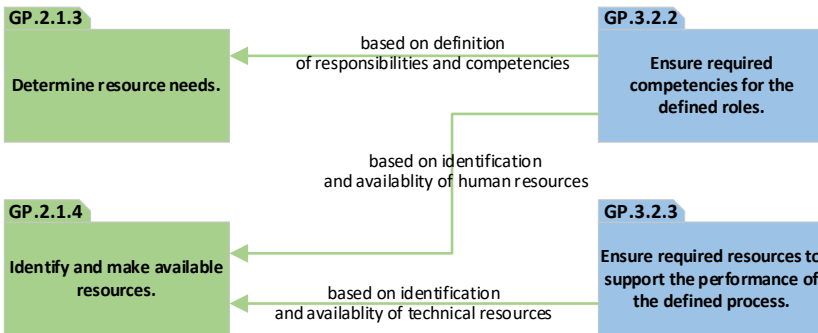
5.3.2 Rating rules between capability level 2 and 3

Process attribute 3.1 Process definition is one of the few process attributes which does not have a dependency on the lower process attributes.

The rationale is that whether the lower process attributes are performed well or badly may or may not affect the definition of the standard process.

However, for a capability level 3 the standard process has to cover all aspects of capability level 1 and 2 and a feedback mechanism to regularly check and improve the standard process itself. Therefore, the rating of the process attribute PA 3.1 is relatively independent of the project.

If this standard process is used the following picture shows the dependencies to of PA 3.2 to level 2:



GP 3.2.2 Ensure required competencies for the defined roles

[PA3.2.RL.18] If the indicator GP 2.1.3 is downrated due to missing or inadequate determination of responsibilities, authorities, knowledge or skills the indicator GP 3.2.2 shall be downrated.

Rationale: If there is a weakness on GP 2.1.3 regarding definition of the responsibilities and authorities this weakness could be evident in two possible scenarios on level 3:

- *The weakness is also found in the GP 3.1.1, the identification of roles, competencies etc. which in turn would lead to a weakness in the project which is using this standard process. (GP 3.2.2.)*
- *The standard process is F regarding GP 3.1.1 but the project does not use the process properly, otherwise the GP 2.1.3 would not be downrated*

[PA3.2.RL.19] If the identification, allocation and availability of resources (GP 2.1.4) is downrated due to human resources issues, the indicator GP 3.2.2 shall be downrated.

[PA3.2.RL.20] If the indicator GP 2.1.4 is downrated due to missing or inadequate qualification of individuals, the indicator GP 3.2.2 shall be downrated.

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

[PA3.2.RL.21] If the identification, allocation and availability of resources (GP 2.1.4) is downrated due to physical or material resource issues, the indicator GP 3.2.3 shall be downrated

Rationale: If there is a weakness on 2.1.4 regarding identification and availability of the resources, especially technical resources this weakness could be evident in two possible scenarios on level 3:

- *The weakness is also found in the GP 3.1.3, the determination of resources (human, material, process infrastructure), which in turn would lead to a weakness in the project which is using this standard process (GP 3.2.3).*
- *The standard process is F regarding GP 3.1.3 but the project does not use the process properly, otherwise the GP 2.1.4 would not be downrated*

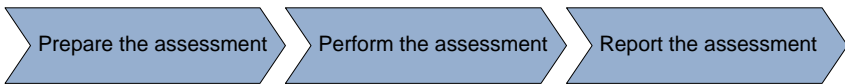
5.3.3 Rating rules to other processes

Dependencies to PIM.3 “Process improvement process” and ORG.1 “Life cycle model management process” (ISO/IEC TS 33060:2020, ISO/IEC TS 33061:2021) are obvious but not described, because both are not part of the recommended VDA Scope.

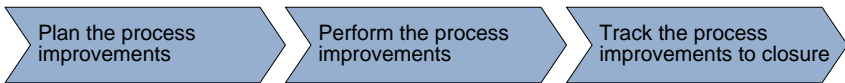
Part 2: Guidelines for performing the assessment

The purpose of the part two of the current publication is to support the assessors in performing an assessment based on the Automotive SPICE process reference and assessment model, considering the requirements of ISO/IEC 33002.

Chapter 6, “Documented assessment process” provides a necessary input for performing the assessment defined in ISO/IEC 33002. It provides the tasks and activities of the so-called evaluation phase, in which the assessment is planned, prepared, performed and documented.



In chapter 7, “*Improvement process*” an overview of the tasks and activities are given, in the case that the assessment results are to serve as an input for subsequent improvement measures. In this so-called improvement phase the assessment results of the evaluation phase are used to plan, execute and track the process improvement actions.



Chapter 8, “*Recommendations for performing an assessment*” provides additional requirements when applying the documented assessment process.

In chapter 9, “*Requirements relating to assessor qualification*” the requirements for assessors to demonstrate the competencies to conduct an assessment and to monitor and verify the conformance of a process assessment are given.

6 Documented assessment process

6.1 Introduction

This chapter provides a documented assessment process (DAP) according to ISO/IEC 33002, clause 4.1:

The assessment shall be conducted according to a documented assessment process. The documented assessment process shall be capable of meeting the assessment purpose and shall be structured in a manner that ensures that the purpose for performing the assessment is satisfied, in terms of the rigor and independence of the assessment and its suitability for the intended use.

The documented assessment process provided was set up to serve most assessments within the automotive domain. It fulfils the requirements of ISO/IEC 33002 under the following preconditions:

- The assessment is using the PRM and PAM Automotive SPICE 4.0 and subsequent versions.
- The assessment is using the process measurement framework defined in Automotive SPICE 4.0 which is an adaptation of ISO/IEC 33020:2019 “*Process measurement framework for assessment of process capability*”.
- A defined rating and aggregation method according to ISO/IEC 33020:2019 is used.
- The assessment is classified as “Class 3” according to ISO/IEC 33002 clause 4.6.
- The category of independence of the body performing the assessment, the lead assessor and the other members of the assessment team is A, B or C according to ISO/IEC 33020:2019, Annex A.
- The assessment is not intended to evaluate organizational maturity.

It is the responsibility of the lead assessor to evaluate whether the assessment provides the given preconditions. In case of deviations, the lead assessor shall take appropriate steps to modify this given DAP or select another suitable one. In this case the lead assessor takes responsibility for the conformity of the DAP to ISO/IEC 33002.

6.2 Assessment input and output

6.2.1 Assessment plan

According to ISO/IEC 33002 an assessment plan shall be setup. Within this DAP the assessment plan shall contain the following elements:

- Required inputs specified in this standard → 6.2.2
- Definition of the class of assessment and the category of independence of the body performing the assessment, the lead assessor and the other members of the assessment team → 6.1
- Communications to the personnel involved in the assessment → 6.3
- Identification of the documented assessment process including:
 - The strategy and techniques for the selection, identification, collection and analysis of objective evidence and data, to satisfy any requirements for coverage of the process scope of the assessment as defined for class 3 assessments → 6.4.1
 - The approach to derive an agreed process attribute rating, where relevant → 6.4.1 and Part 1
 - Activities to be performed in the assessment → 6.4
 - Resources and schedule assigned to these activities → 6.4
 - Identification and definition of roles and responsibilities of the participants in the assessment → 6.3
 - Criteria to verify that the requirements of ISO/IEC 33002 are met → 6.1
 - Description of the planned assessment outputs → 8.4

6.2.2 Assessment inputs

According to ISO/IEC 33002 the necessary assessment input shall be identified. Within this DAP the necessary input shall contain as a minimum the following elements:

- Identity of the sponsor and the sponsor's relationship to the organizational unit(s) being assessed;
- Business context including the organization business's goals and circumstances of the assessment;

- Purpose of the assessment, e.g., process improvement or evaluation of the process capability assigned to a specific product delivery;
- Assessment scope as it applies to the business comprising a defined and declared organization scope, including:
 - The processes to be investigated within the assessment;
 - The process quality characteristic to be investigated, including the highest process quality level for each individual process within the assessment scope;
 - The organizational unit(s) that deploy the process;
 - The boundaries of the assessed organization, including:
 - the size of each organizational unit, e.g., number of personnel;
 - the application domain (e.g., system development, software, development, hardware development) of the products or services of each organizational unit; and
 - key characteristics (e.g., size, criticality, quality) of the products or services of each organizational unit.
 - The process context including the set of stakeholder requirements and changes which are under investigation.
 - The process instances, which have been selected, if applicable
- Identity of the model(s) and process measurement framework used:
 - Automotive SPICE 4.0 or higher
 - Automotive SPICE 4.0 process measurement framework
- Assessment requirements, including:
 - reference to this documented assessment process
 - definition of the class of assessment and the category of independence of the body performing the assessment the lead assessor and the other members of the assessment team
 - rating method(s) to be employed;
 - aggregation method(s) to be employed.
 - assessment constraints considering, at minimum:
 - availability of key resources;

- maximum duration of the assessment;
- specific processes or organizational units to be excluded from the assessment;
- Ownership of the assessment outputs and any restrictions on their use;
- Controls for handling confidential information and non-disclosure.
- Participants and their roles, the assessment team and assessment support staff with specific responsibilities for the assessment;
- Criteria for competence of the lead assessor.

6.2.3 Assessment report

The requirements and recommendations for the assessment report are defined in detail in chapter 8.4.

6.2.4 Objective evidence gathered

For evaluating the processes within the assessment scope objective evidence and additional information shall be collected. Each evidence shall be traceable to associated assessment indicators (base practices, WP, generic practices, etc..).

6.3 Parties and roles involved in the assessment

The main parties involved in the assessment are the sponsor, the assessing organization, and the assessed organization. The following roles shall be identified:

LAC: Local Assessment Coordinator

Individual or entity, who takes responsibility for the organization of the assessment within the organizational unit assessed.

SP: Sponsor

Individual or entity, internal or external to the organizational unit to be assessed, who requires the assessment to be performed, and provides financial or other resources to carry it out (see ISO/IEC 33001 clause 3.2.9).

AS: Co-Assessor

Individual who participates in the rating of process attributes (see ISO/IEC 33001 clause 3.2.11). Assessors have appropriate education, training and both capability assessment experience and domain experience to perform the required class of assessment and make professional judgments (see ISO/IEC 33001 clause 3.2.11).

LA: Lead Assessor or Assessment team leader

Assessor who has demonstrated the competencies to conduct an assessment and to monitor and verify the conformity of a process assessment (see ISO/IEC 33001 clause 3.2.12).

PP: Participant

Individual from the organizational unit to be assessed, who takes part in the assessment.

Note: While the role definitions provided above are considered to represent the standard approach to responsibility distribution, it is possible that individual assessments may extend or reduce these role definitions as is appropriate for a given assessment. For example, the SP may be knowledgeable of process assessment and may therefore participate in the detailed aspects of the assessment. The LAC may also be capable of performing a greater role in the process

assessment depending on their knowledge and training with respect to process assessment.

For the description of the responsibilities the following abbreviations are used:

R: Responsible

Those who do the work to achieve the task. There is at least one role with a participation type of responsible, although others can be delegated to assist in the work required (see also RACI below for separately identifying those who participate in a supporting role).

A: Accountable (also approver or final approving authority)

The one ultimately answerable for the correct and thorough completion of the deliverable or task, and the one who delegates the work to those responsible [7]. In other words, an accountable must sign off (approve) work that responsible provides. There must be only one accountable specified for each task or deliverable.

C: Consulted (sometimes counsel)

Those whose opinions are sought, typically subject matter experts; and with whom there is two-way communication.

I: Informed

Those who are kept up to date on progress, often only on completion of the task or deliverable; and with whom there is just one-way communication.

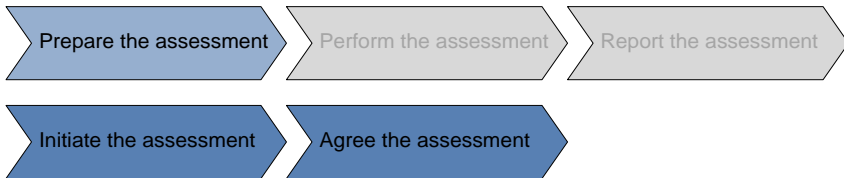
6.4 Assessment activities

The assessment process consists of three tasks:

- Prepare the assessment
- Perform the assessment
- Report the assessment

6.4.1 Prepare the assessment

The preparation for an assessment is split into two sub-tasks:



6.4.1.1 Initiate the assessment

In the initialization phase the assessing organization determines the need for an assessment and determines the framework conditions (scope, time period, team, etc.). All necessary information on the assessed organization is collected.

Brief description	The need for an assessment is determined and the framework conditions for its execution are established.
Process inputs	<ul style="list-style-type: none">• Formal or informal assessment enquiry• Information about the organization assessed• Previous audit reports and assessment reports
Process outputs	<ul style="list-style-type: none">• Assessment purpose• Assessment agreement• Assessment scope• Time frame• Contact persons in both organizations• Assessment team list• Assessment plan

Activities / Responsibilities	LA	AS	SP	LAC	PP
Determine the need for an assessment	-	-	A,R	-	-
Establish the assessment agreement	C	C	A,R	C	-
Define the assessment scope	C, R	I	A	C	-
Collecting and evaluating information on the organization assessed	A,R	C	-	C	-
Define the assessment team	A,R	C	-	C	-

Determine the need for an assessment

The need for an assessment must be determined by the sponsor. This may be derived based on different use cases and defines the purpose of the assessment. Examples for use cases are given in chapter 1.2.1. The purpose of the assessment is the base input for setting up the assessment scope.

Establish the assessment agreement

The assessment agreement is established based on the assessment purpose by

- defining the main focus of the assessment. This may be, for example, project management, engineering aspects or other areas of risk. If appropriate, a pre-selection should be made of the processes to be checked.
- By determining the assessing organization, which is responsible for performing the assessment,
- selecting the lead assessor and the assessment team members,
- defining the timeframe, within which the assessment should be carried out, and
- identifying the business divisions or departments and personnel in the organization assessed that are to be involved.

Define the assessment scope

The boundaries of the assessment, provided as part of the assessment input, encompassing

- the boundaries of the organizational unit for the assessment,
- the processes to be included,
- the capability level for each process to be assessed, and
- the context within which the processes operate

are defined.

Collecting and evaluating information on the organization assessed

The information on the organization assessed which is relevant to the assessment must be collected and evaluated. This may include:

- Organizational structure of all those involved in the project, such as
 - Sponsor,
 - Project team,
 - Core/platform development,
 - (independent) quality assurance department,
 - (independent) test department or
 - Sub-suppliers.
- Standard software components/of the shelf items.
- If appropriate, the department responsible for the selection, release and maintenance of tools or the IT department, for example for configuration management.
- Results of other audits and assessments.

Note: Results from previous audits and assessments can be used for determining the assessment scope. Here, the time has to be considered that has passed since the audit or assessment and whether the results are applicable for the project (assessment method, assessed department, personnel involved).

Define the assessment team

The assessment team is determined and appointed.

6.4.1.2 Agree the assessment

The exact terms of the assessment are agreed between the involved parties.

Brief description	The assessment and its framework conditions are agreed.				
Process inputs	<ul style="list-style-type: none"> • Assessment scope • Time frame • Assessment team list • Assessment plan 				
Process outputs	<ul style="list-style-type: none"> • Non-disclosure agreement (NDA) • Assessment time schedule • List of documents to be exchanged in advance • Requirements for the evidence repository • Distribution list for the report • Optional: minutes of the pre-assessment meeting 				
Activities \ Responsibilities	LA	AS	SP	LAC	PP
Agree the details of how the assessment shall be performed	R	I	A	C	-
Perform pre-assessment meeting (optional)	A,R	C	-	C	I

Agree the details of how the assessment shall be performed

With the assessment agreement, a consensus regarding the assessment should be achieved by defining details of how the assessment shall be performed and agree them between the parties.

It is essential that the sponsor, the assessing organization, and the organization assessed agree on the modalities of the assessment. The agreement can be reached formally by means of a contract and acknowledgement, or in an informal manner. Furthermore, the assessment agreement must consider and specify the following points:

- A non-disclosure agreement (NDA) should be agreed by all parties involved (assessing organization, organization assessed and assessors) and signed (if not already done in the project).
- The final schedule is agreed.
- Contact persons are appointed on both sides for coordination.

- The distribution list for the report is established.
- Requirements relating to the evidence repository for the assessment are established.
- Requirements relating to the infrastructure, e.g., meeting rooms, beamers, printers, flipcharts etc. are established.
- Constraints for the scheduling, e.g., availability due to bank holidays, breaks, local conventions etc. are identified.

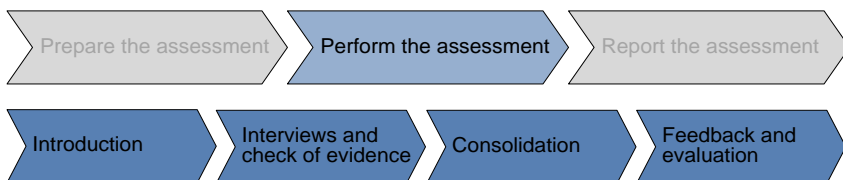
Perform pre-assessment meeting (optional)

If necessary, a pre-assessment meeting can be carried out (on-site, by email or by a telecommunications conference). The purpose is to

- explain the framework and process of the assessment to the personnel involved;
- specify the set of documents to be handed out to the assessment team in advance for study;
- to understand and confirm the assessment context; and
- to perform preliminary document analysis.

6.4.2 Perform the assessment

The execution of the assessment is split into four tasks:



In the introduction task the assessment scope, the project to be assessed and the assessment method are presented. This is followed by the interviews and document reviews, where the actual collection of evidence is done which is the crucial part of the assessment. Once the collection of evidence has been completed, the consolidation task starts, and the first evaluation of the results (findings) takes place. Finally, in the feedback and evaluation task, the collected results are stored in the evidence repository, the preliminary process attribute rating results are presented, and possible immediate actions are recommended.

6.4.2.1 Introduction

Brief description	The organization to be assessed, the project, the evaluation methodology and the activities of the assessment are presented.				
Process inputs	<ul style="list-style-type: none"> Information on the organization assessed and the project Assessment scope Assessment time schedule Assessment plan 				
Process outputs	<ul style="list-style-type: none"> Information of the organization assessed and project Information on Automotive SPICE, the assessment scope, and the assessment time schedule 				
Activities \ Responsibilities	LA	AS	SP	LAC	PP
Present the organization assessed and the project	I	I	-	A, R	C
Present the assessment activities	A, R	C	I	I	I

The introduction should give all those involved an overview of the organization assessed, the project, the assessment methodology and sequence.

Present the organization assessed and the project

The organization presents itself and the project in the scope to be assessed to the assessment team. The purpose of this activity is to provide the assessment team with an introduction to the project-specific conditions and circumstances.

Present the assessment activities

The assessment team presents the concrete activities of the Automotive SPICE assessment. The purpose of this activity is to inform the organization assessed and the interviewees about the detailed procedure which will be followed during the assessment (for example, the evidence repository).

6.4.2.2 Interviews and checks of evidence

Brief description	The project-related information regarding the selected processes is collected and documented in accordance with the assessment model.				
Process inputs	<ul style="list-style-type: none"> • Assessment time schedule • Project-related work products 				
Process outputs	<ul style="list-style-type: none"> • Assessment notes regarding results of interviews, documents which have been examined and results of the inspection of the work environment • List of documents which have been examined 				
Activities \ Responsibilities	LA	AS	SP	LAC	PP
Perform interviews, document checks and inspections of the work environment, if appropriate	A, R	C	-	C	C
Collect evidence for rating the processes	A, R	C	-	C	C

Evidence which is relevant to the project in terms of the selected processes is collected and documented.

Perform interviews, document checks and inspections of the work environment, if appropriate

Based on the assessment time schedule, interviews on the individual processes with the key personnel of the organization assessed are carried out and the associated documents/evidence are examined. If necessary, the conditions under which the process is performed can be checked at the workplace.

The results of the interviews are documented in the assessment notes.

Collect evidence for rating the processes

The assessment team collects the evidence to justify and document the findings for the individual processes (for example, with regard to process compliance, the tools used in the project and the quality of existing documents).

6.4.2.3 Consolidation

Brief description	The selected processes are rated by the assessors based on the available evidence.				
Process inputs	<ul style="list-style-type: none"> Assessment notes 				
Process outputs	<ul style="list-style-type: none"> Consolidated assessment notes Provisional process capability profiles 				
Activities \ Responsibilities	LA	AS	SP	LAC	PP
Evaluate the collected evidence	A, R	C	-	-	-
Provide a provisional rating	A, R	C	-	-	-
Document strengths and potential improvements	A, R	C	-	I	I
Establish the traceability of process attribute rating to evidence	A, R	C	-	-	-
Document the deviation of rating rules	A, R	C	I	-	-

The evidence collected from interviews and document reviews is consolidated by the assessors.

Note: The consolidation might also be done incrementally after each interview session, see chapter 6.4.2.2.

Evaluate the collected evidence

Following the interviews and the document reviews the assessment team consolidates and documents the analysis results and reaches consensus on the identified strengths and potential improvements of the processes which have been assessed.

Provide a provisional rating

Based on the findings the process attributes are rated and a provisional set of process capability profiles is determined for the assessed processes. The rating is evaluated whether the rating is consistent with the rules and recommendations given in part one of this publication. The rating shall consider the rating rules and recommendations given in Part 1 of this document.

Document strengths and potential improvements

The findings are evaluated in terms of strengths and potential improvements.

Establish the traceability of process attribute rating to evidence

For each process attribute rating the traceability to the collected evidence used in determining that rating is established. The relationship between the assessment indicators for each process attribute rated and the objective evidence is documented.

Document the deviation of rating rules

The rules not obeyed by the lead assessor are identified. A justification, why the rule is not applicable or why it has no significant impact on the process attribute rating, is provided.

Note: The purpose of the justification is to briefly document the lead assessor's decision not following a specific rule. It is the clear intention of the authors of this publication not to generate additional effort due to extensive documentation of rule deviations. The provision of a list of all rules, no matter whether they are obeyed or not might make sense for unexperienced assessors and might give an overview but is not required or intended by the authors of this publication.

6.4.2.4 Feedback and evaluation

Brief description	A provisional evaluation of the organization assessed is presented and immediate actions are identified.				
Process inputs	<ul style="list-style-type: none"> • Provisional process capability profiles • List of documents which have been examined • Consolidated assessment notes 				
Process outputs	<ul style="list-style-type: none"> • Provisional process attribute ratings and the process capability profiles • List of the most important findings (strengths and potential improvements) • Document archive related to the assessment • List of immediate actions, if applicable 				
Activities \ Responsibilities	LA	AS	SP	LAC	PP
Present the results	A, R	C	I	I	I

Identify immediate actions (optional)	C	C	-	A, R	C
Store the evidence in the repository	I	-	-	A, R	I

The purpose of feedback is to provide information on the assessment results and to reach a common understanding of the rating.

The feedback shall contain the following as a minimum:

- The provisional process attribute ratings
- The provisional process capability profiles
- The major strengths and potential improvements (for each process assessed).

The feedback should be provided directly following the conclusion of all interviews. The contents of the feedback should be documented in writing as a feedback presentation and afterwards made available as a copy to the assessed party.

Present the results

The provisional process attribute ratings and the capability profiles are prepared and presented to the organization assessed. The most important findings (strengths and potential improvements) are presented.

Identify immediate actions (optional)

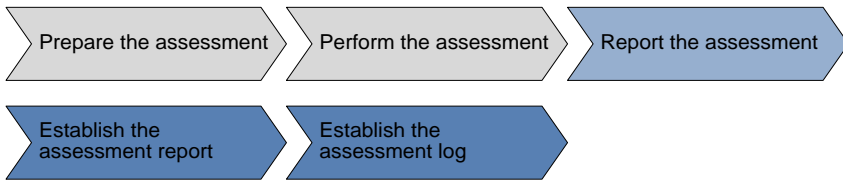
Based on the presented identified potential improvements, immediate actions are recommended to eliminate critical weaknesses.

Store the evidence in the repository

The organization assessed stores the evidence repository including references to the documents which have been analyzed.

6.4.3 Report the assessment

The elaboration and distribution of the report following an assessment is split into two tasks:



The detailed assessment report is drawn up in order to document the results of the assessment. The assessment log is drawn up for submission to the certification body.

6.4.3.1 Establish the assessment report

Brief description	The assessment team compiles the assessment report to be distributed within four calendar weeks in the assessed organization.				
Process inputs	<ul style="list-style-type: none"> • Consolidated assessment notes • Provisional process capability profiles • List of the most important findings. 				
Process outputs	<ul style="list-style-type: none"> • Assessment report with the process attribute ratings and the final process capability profiles • An explanation of deviations at the practice level 				
Activities \ Responsibilities	LA	AS	SP	LAC	PP
Consolidate the final process attribute ratings and the final process capability profiles	A, R	C	-	-	-
Compile the assessment report	A, R	C	I	I	I
Distribute the assessment report	-	-	A, R	C	I

Consolidate the final process attribute ratings and the final process capability profiles

The set of final process capability profiles is drawn up. The consolidated findings and observations are documented in detail based on the assessment notes.

Compile the assessment report

The assessment report must be compiled, checked and released by the assessment team. The lead assessor is responsible for drawing up and releasing the assessment report. Deviations from rating rules given in Part 1 of this publication shall be documented in the assessment report. The assessment report is provided within normally four calendar weeks to the assessment sponsor for distribution in assessed organization. Please refer to chapter 8.4 for detailed requirements on the assessment report.

Distribute the assessment report

The released version is distributed within the assessed organization.

6.4.3.2 Establish the assessment log

Brief description	The assessment team draws up the assessment log.				
Process inputs	• Template for the assessment log				
Process outputs	• The assessment log				
Activities \ Responsibilities	LA	AS	SP	LAC	PP
Issue the assessment log	R	C	A	-	-

Issue the assessment log

The assessment log represents the confirmation of the sponsor, the LAC and the assessment team about the performance of the assessment according to the defined assessment process.

The assessment log shall be signed by the lead assessor and the assessment team members. The log shall be approved by the sponsor.

The assessment log shall be drawn up on the basis of the template provided by the certification scheme (see chapter 9, *“Requirements relating to assessor qualification”*).

7 Improvement process

7.1 Introduction

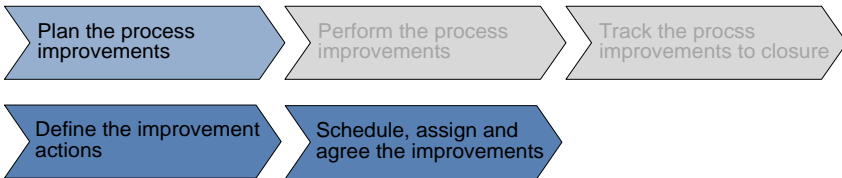
The process improvement phase may follow the evaluation phase and is split into the planning on the process improvement actions, into performing and into tracking these actions.

Since the improvement actions will be in general not be assigned to the roles involved in the evaluation phase, no assignment of responsibilities is given in this chapter.

7.2 Improvement activities

7.2.1 Plan the process improvements

The process improvement actions are established, together with the monitoring criteria, responsibilities and the time schedule.



7.2.1.1 Define the improvement actions

Brief description	The process improvement actions to be carried out are selected and prioritized.
Process inputs	<ul style="list-style-type: none">• Assessment report• List of immediate actions, if applicable
Process outputs	<ul style="list-style-type: none">• List of process improvement actions• Monitoring criteria for process improvement actions
Activities	
Specify the process improvement actions	
Prioritize the process improvement actions	
Define the monitoring criteria	

Specify the process improvement actions

A list of process improvement actions is established including the desired improvement result based on the assessment report. A traceability to the identified assessment findings is provided, if applicable.

Prioritize the process improvement actions

Prioritization is performed based on an evaluation of the effectiveness of the improvement actions.

Define the monitoring criteria

Based on the list of process improvement actions monitoring criteria are defined which allow to check whether the implementation of the actions have the desired effects.

7.2.1.2 Schedule, assign and agree the improvements

Brief description	The improvements are scheduled, assigned and a commitment on the improvements is achieved.
Process inputs	<ul style="list-style-type: none">List of process improvement actions
Process outputs	<ul style="list-style-type: none">Responsibilities for process improvement actionsTime schedule for process improvement actions
Activities	
Define the responsibilities	
Define the time schedule for implementation	
Agree on the improvement actions	

Define the responsibilities

The improvement actions are assigned to persons who are responsible for their implementation.

Define the time schedule for implementation

Dates and priorities are assigned to the individual process improvement actions. Based on a risk assessment, the actions from

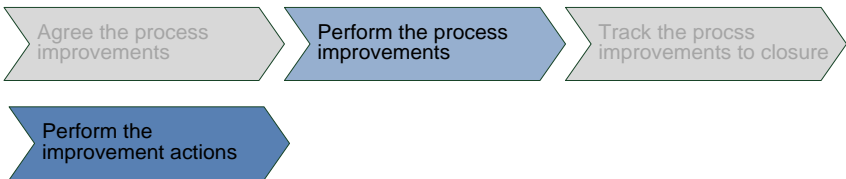
the list are identified which are to be implemented in the project and/or in the organization which has been assessed.

Agree on the improvement actions

An agreement on the improvements is achieved from all affected parties.

7.2.2 Perform the process improvements

Immediate actions should be carried out directly after the assessment. Other process improvement actions are implemented according to the defined schedule.



7.2.2.1 Performing process improvement actions

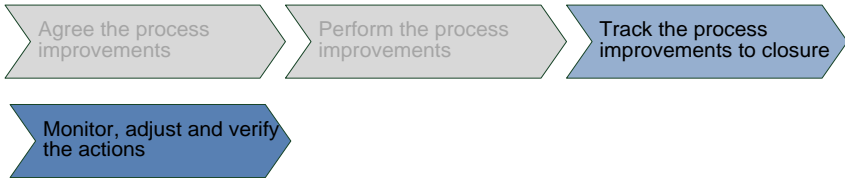
Brief description	The process improvement actions are carried out
Process inputs	<ul style="list-style-type: none"> List of process improvement actions Responsibilities for process improvement actions Time schedule for process improvement actions.
Process outputs	<ul style="list-style-type: none"> Documentation of the improvements which have been carried out
Activities	
Execute the process improvement actions	

Execute the process improvement actions

The process improvement actions should be carried out in due time by those responsible and according to priority.

7.2.3 Track the process improvement to closure

Tracking the process improvement actions represents the completion of the improvement process:



The process improvement actions are monitored and any necessary adjustments are made, taking risks into account.

7.2.3.1 Monitor, adjust and verify the actions

Brief description	The actions are monitored and adjusted if necessary
Process inputs	<ul style="list-style-type: none"> List of process improvement actions Monitoring criteria for process improvement actions Documentation of the improvements which have been carried out
Process outputs	<ul style="list-style-type: none"> Status report of the process improvement actions Road map for long term actions exceeding the project scope
Activities	
Monitor the process improvement actions	
Modify improvement actions if deficiencies are detected	
Verify and close improvement actions	
Plan long term actions exceeding the project scope	

Monitor the process improvement actions

Based on the defined monitoring criteria the process improvement actions are checked regularly regarding their implementation and effectiveness.

Modify improvement actions if deficiencies are detected

If the actions do not achieve the desired effect, modified or new actions are specified.

Verify and close improvement actions

The improvement actions are closed, if they achieved their purpose.

Plan long term actions exceeding the project scope

Long term actions exceeding the project scope should be addressed within a road map.

8 Recommendations for performing an assessment

In the current chapter recommendations are provided, which should be considered when following the documented assessment process specified in chapter 6.

8.1 Assessment results

8.1.1 Confidentiality of information

As a fundamental rule, assessment results and the information obtained during an assessment must be treated as confidential by all persons and organizations involved.

8.1.2 Handling the assessment results

The ownership of the assessment results is defined in the initial assessment agreement (see 6.4.1.1); by default, the Sponsor is the owner of the results.

If the assessment results are issued to third parties, an additional non-disclosure agreement should be signed where appropriate.

The assessment results and any relevant part of them should be made available within normally four calendar weeks after the assessment to all individuals involved in the assessed project and individuals involved in the performance and monitoring of the improvement actions. The criterion here is their involvement in the project or process development.

The assessment results should be documented and archived by the assessing organization.

8.2 Validity of assessments

8.2.1 Area of validity of the assessment results

Automotive SPICE is predominantly used to assess single projects based on a given scope. In these assessments the focus is always on one particular project. Neither the complete set of all projects in an organization nor a statistically significant selection is investigated. It follows therefore that assessment results are a representative sample of the process capability within the scope of the assessment, but not applicable in general to the assessed organization as whole, the development location or the entire company.

The assessment results may be considered to reflect potential capability of another project with identical characteristics. Here the following criteria should be considered:

- Development locations: As a general rule, assessment results are not transferable from one location to another.
- ECU domains: If at a large development location ECUs are developed for various domains, such as powertrain, chassis or body, assessment results are transferable only to a limited degree, given the different development environments.
- Distributed development: Where the development work on ECUs is distributed over several departments or several locations, the assessment results apply only to those locations or departments which have been assessed.

The degree to which assessment results may be transferred will depend on various factors, including the process capability level and must be examined in each individual case.

8.2.2 Period of validity of assessment results

Assessment results have only a limited validity in terms of time. Experience has shown that they allow reliable conclusions to be drawn for 12 months regarding the project which has been assessed.

Changes within the project, such as, for example

- the transfer of the development work to a different location,
- a re-organization in the organization which has been assessed or

- changes to the development processes

can, however, significantly affect the relevance of the assessment results to individual processes even within 12 months. Such changes may cause the actual capability of the development process to be better or worse than indicated by the last assessment result.

On the other hand, where there is a high degree of project stability, the assessment results may permit reliable conclusions regarding the project to be drawn for longer than 12 months. For these reasons, the period of validity must always be considered relative to the specific project circumstances.

8.3 Performing an assessment

The following recommendations should be observed when performing assessments:

8.3.1 General

The assessment team leader has the authority, and the responsibility, to take any necessary precautions and actions to ensure that the assessment is conducted in compliance with the relevant ISO/IEC 330xx parts, the Automotive SPICE 4.0 measurement framework and this document. This includes the right to dismiss individuals (assessment team or interviewees), or to cancel interviews.

8.3.2 Assessment scheduling

When planning the assessment, at a minimum the following conditions should be considered:

- The scope of the assessment, specifically the number of assessed processes, the number of process instances and the highest assessed level
- The process context as defined in chapter 1.2.3.
- The complexity of the assessed project, e.g., in terms of distributed developments, size of the assessment scope, complexity of the developed product
- Results and experiences from previous assessments
- Assessment experience of the assessed party
- Problems associated with different cultures and languages

Based on this sufficient interview and consolidation time frames should be planned.

There should be at least four weeks between agreement on an assessment and its execution.

In some cases it is appropriate to perform interviews for data collection only using phone and/or video conferences.

8.3.3 Individuals involved in the assessment

The assessing organization performing the assessment decides on the composition of the assessment team in agreement with the sponsor.

Participation by observers or other guests in interviews:

- In principle, observers can be present at an interview – e.g., observers from the process development department.
- The number of people taking part in the interview should be kept as small as possible.
- The interviews must not be impaired by observers, whether active or passive.
- The assessment team leader decides whether observers may be present at the interviews and can exclude observers (in general or particular individuals) even during the course of the assessment.

8.3.4 Composition of the assessment team

Editors note: This chapter will be updated after alignment with intacs (Certification scheme).

The interviews in the assessment should be carried out by at least two assessors.

Independence of the assessors, depending on the assessment type, should be ensured in order to avoid any conflict of interest.

The assessment team leader has the final authority for the selection of the assessor(s) and to exclude participants from the assessment.

8.4 Assessment Report

In the assessment report the organization which has been assessed is given more detailed feedback of the strengths and potential improvements detected in the assessment. The assessment report should document in particular those points which led to a downrating of the process attribute by referencing to the individual base or generic practices.

The assessment report should contain the following information:

8.4.1 General information

This chapter contains general information on the assessment report.

Item	Required information
Unique identifier	<ul style="list-style-type: none">• Document/Version number or equal
Date of issue	<ul style="list-style-type: none">• Issue date of the report
Version	<ul style="list-style-type: none">• Version identification of the report
Issuer	<ul style="list-style-type: none">• Issuer of the report
Change history	<ul style="list-style-type: none">• Document change history

8.4.2 Formal information about the assessment

This chapter contains formal information about the assessment.

Item	Required information
Assessment model	<ul style="list-style-type: none">• Assessment model and version that has been used (e.g., Automotive SPICE PAM V4.x)
Assessment period	<ul style="list-style-type: none">• The period during which the assessment was carried out
Sponsor	<ul style="list-style-type: none">• Name of the assessment sponsor
Local assessment coordinator	<ul style="list-style-type: none">• Name of the responsible coordinator of the assessed organization

Evidence	<ul style="list-style-type: none"> • The work products examined for each process.
Distribution list	<ul style="list-style-type: none"> • Distribution list of the report
Assessment class	<ul style="list-style-type: none"> • Class of the assessment according to ISO/IEC 33002
Assessment type	<ul style="list-style-type: none"> • Type A, B, C, or D according to ISO/IEC 33002 Annex A

8.4.3 Scope of the assessment

This chapter contains information about the assessment scope. Refer also to chapter 1.2.2, “*Defining the assessment scope*”.

Item	Required information
Process scope	<ul style="list-style-type: none"> • Selection of processes in the assessment • In case of derivation of the recommended VDA scope: A rationale for the selection of the processes
Capability level	<ul style="list-style-type: none"> • Target capability level for each process assessed
Assessed project	<ul style="list-style-type: none"> • Project Name / description
Organization	<ul style="list-style-type: none"> • Company name • Organizational / Business unit • Assessed sites • Assessed Departments
Process context	<ul style="list-style-type: none"> • Identification of the set of stakeholder requirements considered for the assessment • Identification of the set of changes considered for the assessment <p><i>Note: It is sufficient to identify the sets by suitable criteria, please refer to chapter 1.2.3, “Defining the process context in the assessment scope”</i></p>

8.4.4 Participants of the assessment

This chapter contains information about the assessment team, the interview persons and other participants of the assessment.

Item	Required information
Assessment team leader	<ul style="list-style-type: none"> • Name of the assessment team leader • Assessor grade (e.g., Competent, Principal) • License number of the assessment team leader • Expiry date of the assessor license
Co-Assessor(s)	<ul style="list-style-type: none"> • Name of the Co-Assessor(s) • Assessor(s) grade (e.g., Provisional, Competent, Principal) • License number of Assessor(s) license(s) • Expiry date of the assessor(s) license(s)
Local assessment coordinator	<ul style="list-style-type: none"> • Name of local assessment coordinator
Interviewed persons	<ul style="list-style-type: none"> • Names of interviewed individuals incl. • their role in the project or organizational unit • mapping to the processes for that they were interviewed for (project manager e.g., could be interviewed for more than one process)
Guests (optional)	<ul style="list-style-type: none"> • Names of persons passively attending the interviews without any rights, e.g., observers, assessor candidates... <p><i>Note: To gather experience assessor candidates may participate in the process attribute rating but should not be involved in the rating decision.</i></p>

8.4.5 Constraints

This chapter contains information about constraints that have to be considered to understand the assessment results.

Item	Required information
Constraints (if applicable)	e.g. <ul style="list-style-type: none"> • Somebody was not available (e.g., off, sick) • Separated development areas have been included via Video/WebEx (no on-site assessment) • Disclaimer (e.g., that the assessment results does not allow conclusions to the complete organization or other departments of the organization that has been not assessed) • Confidentiality constraints, e.g., access to evidence or to infrastructure and sites may be subject to legal access rights.

9 Requirements relating to assessor qualification

It is essential that Automotive SPICE assessments are conducted by appropriate and trained specialists. The lead assessor entrusted with the leadership of the assessment, who also accepts responsibility for the result of the evaluation, plays a special role.

The training of assessors shall be carried out by registered training organizations based on a published certification scheme.

The personal certification of assessors shall be carried out by a certification body on the basis of a published certification scheme. The certification scheme shall cover the guidance, the rules and the recommendations given within this publication.

Acceptance of valid qualification schemes for assessors is carried out by the quality management board of the VDA QMC. Currently, the intacs scheme is a valid and accepted qualification scheme.

9.1 Requirements for assessors

According to the definitions provided in ISO/IEC 33001, clause 3.2.11, the term “assessor” is defined as:

individual who participates in the rating of process attributes

A valid personal Automotive SPICE Provisional, Competent or Principal SPICE Assessor license issued by the VDA QMC is required as evidence for the qualification and experience of any assessor who is member of the assessment team.

9.2 Requirements for lead assessors

According to the definitions provided in ISO/IEC 33001, clause 3.2.12, the term “lead assessor” is defined as:

Assessor who has demonstrated the competencies to conduct an assessment and to monitor and verify the conformance of a process assessment.

A valid personal Automotive SPICE competent or principal assessor license or a valid instructor license issued by the VDA QMC is

required as evidence for the qualification and experience of the lead assessor.

9.3 Requirements for non-lead assessors

According to the definitions provided in ISO/IEC 33001, clause 3.2.11, the term “assessor” is defined as:

individual who participates in the rating of process attributes

A valid personal Automotive SPICE provisional, competent or principal assessor license or a valid instructor license issued by the VDA QMC is required as evidence for the qualification and experience of any other assessor who is member of the assessment team.

9.4 Requirements for assessor license upgrade

Editors note: This chapter will be updated after alignment with intacs (Certification scheme).

9.5 Requirements for assessing additional domains

Editors note: This chapter will be updated after alignment with intacs (Certification scheme).

Bibliography

[ISO33001] ISO/IEC 33001:2015, Information technology — Process assessment — Concepts and terminology, 2015-03-01

[ISO33020] ISO/IEC 33020:2015, Information technology — Process assessment — Process measurement framework for assessment of process capability, 2015-03-01

[ISO24765] ISO/IEC/IEEE 24765:2017, Systems and software engineering — Vocabulary, 2017-12

[ISO19011] ISO 19011:2018, Guidelines for auditing management systems

[Metz2016] Dr. Pierre Metz, Automotive SPICE - Capability level 2 und 3 in der Praxis, August 2016, dpunkt.verlag, ISBN 978-3-86490-360-1

[IntAgile] Frank Besemer, Dr. Pierre Metz, Joachim Pfeffer, Intacs white paper, Clarifying Myths with Process Maturity Models vs. Agile, Aug 6th 2014, www.intacs.info

[intacs] International Assessor Certification Scheme, www.intacs.info

[AS40] Automotive SPICE® Process Reference Model, Process Assessment Model, Version 4.0 Draft, 2023-06-06,