Document Title	Requirements on Methodology
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	362

Document Status	Final
Part of AUTOSAR Standard	Foundation
Part of Standard Release	1.1.0

Document Change History			
Date	**Release**	**Changed by**	**Description**
2017-03-31	1.1.0	AUTOSAR Release Management	• – Migration of document to standard “Foundation” – • Only those requirements from Classic Platform incorporated which apply to Adaptive Platform as well • New requirements for Adaptive Platform added

Disclaimer

Table of Contents

Bibliography

[1] Requirements on Methodology
AUTOSAR_RS_Methodology

[2] Standardization Template
AUTOSAR_TPS_StandardizationTemplate

[3] Main Requirements
AUTOSAR_RS_Main

1 Introduction

This document defines the requirements needed to specify the AUTOSAR methodology.
The document is structured into several sections with general requirements for the AUTOSAR methodology, see section 2.1, as well as dedicated requirements for the Adaptive Platform in section 2.2.

1.1 Limitations

The general requirements in section 2.1, which are valid for the Classic Platform as well as for the Adaptive Platform, were taken over from the Classic Platform document [1]. This implies that currently some requirements are redundant to those in the respective Classic Platform document [1]. Those requirements have not been been adapted in their content and they remain valid for the Classic Platform as well. For the next Classic Platform release all requirements will be merged into this document.

1.2 Document Conventions

The representation of requirements in AUTOSAR documents follows the table specified in [TPS_STDT_00078], see Standardization Template, chapter Support for Traceability ([2]).

The verbal forms for the expression of obligation specified in [TPS_STDT_00053] shall be used to indicate requirements, see Standardization Template, chapter Support for Traceability ([2]).

1.3 Abbreviations

Abbreviation	Description
AP	Adaptive Platform
AUTOSAR	Automotive Open System Architecture
CP	Classic Platform
ECU	Electronic Control Unit
OEM	Original Equipment Manufacture
RTE	Runtime Environment
SIL	Safety Integrity Level (IEC61508 definition)
SWC	Software Component
VFB	Virtual Functional Bus

1.4 Requirements Tracing

The following table references the requirements specified in [3] and links to the fulfillments of these.

Requirement	Description	Satisfied by
[RS_Main_00002]	AUTOSAR shall provide a software platform for high performance computing platforms	[RS_METH_00204] [RS_METH_00207]
[RS_Main_00030]	AUTOSAR shall support development processes for safety related systems	[RS_METH_00018]
[RS_Main_00060]	AUTOSAR shall provide a standardized software interface for communication between Applications	[RS_METH_00201]
[RS_Main_00080]	AUTOSAR shall provide means to describe a component model for Application Software	[RS_METH_00202]
[RS_Main_00130]	AUTOSAR shall provide an abstraction from hardware	[RS_METH_00032]
[RS_Main_00140]	AUTOSAR shall provide network independent communication mechanisms for applications	[RS_METH_00032]
[RS_Main_00150]	AUTOSAR shall support the deployment and reallocation of AUTOSAR Application Software	[RS_METH_00078] [RS_METH_00079] [RS_METH_00201] [RS_METH_00202] [RS_METH_00205]
[RS_Main_00190]	AUTOSAR shall support interoperability with non-AUTOSAR software on the same ECU	[RS_METH_00018]
[RS_Main_00220]	The functional interfaces of AUTOSAR shall be specified in standard C	[RS_METH_00015]
[RS_Main_00250]	AUTOSAR methodology shall provide a predefinition of typical roles and activities in work-share model	[RS_METH_00042] [RS_METH_00066]
[RS_Main_00300]	AUTOSAR shall provide data exchange formats to support work-share in large inter and intra company development groups	[RS_METH_00006] [RS_METH_00018] [RS_METH_00020] [RS_METH_00077] [RS_METH_00078] [RS_METH_00079]
[RS_Main_00301]	AUTOSAR shall specify profiles for data exchange to support work-share in large inter- and intra-company development groups	[RS_METH_00084]
[RS_Main_00310]	AUTOSAR shall support hierarchical Application Software design methods	[RS_METH_00041]
[RS_Main_00320]	AUTOSAR shall provide formats to specify all aspects necessary to integrate Application Software on an ECU	[RS_METH_00206]
[RS_Main_00330]	AUTOSAR shall support the principle of information hiding	[RS_METH_00032]
[RS_Main_00350]	AUTOSAR specifications shall be analyzable and support according methods to demonstrate the achievement of safety related properties.	[RS_METH_00041]
[RS_Main_00400]	AUTOSAR shall provide a layered software architecture	[RS_METH_00032]
[RS_Main_00503]	AUTOSAR shall provide a Software Platform that supports adaptation of communication topology after production	[RS_METH_00203] [RS_METH_00204] [RS_METH_00205]

[RS_Main_00505]	AUTOSAR support the interaction of onboard application software with offboard systems.	[RS_METH_00206]
[RS_Main_00507]	AUTOSAR shall reflect the stages of a software system development in a formal model description	[RS_METH_00056]

2 Methodology Requirements

This chapter provides the definition of the requirements.

2.1 General Requirements

This sections specifies the general requirements, which are valid for both platforms. These requirements were taken over from the CP document [1] and were not changed in their content. Therefore, often the rationale or the use cases are related to the Classic Platform only. This will be adapted for the next Classic Platform release.

2.1.1 Main Requirements

[RS_METH_00006] Methodology shall explain how Autosar system is built ⌈

Type:	valid
Description:	Methodology shall explain how Autosar system is built using the templates and activities supported by guidance. It should be like a user manual to help an organization efficiently apply Autosar.
Rationale:	A strong methodology is necessary to effectively manage building a large system.
Dependencies:	–
Use Case:	Engineer would like to complete an activity and would like to know what inputs are needed, Guidance should be used, etc. Typical use cases involved to build an Autosar system include: • SWC implementation • ECU integration • System integration
Applies to:	AP
Supporting Material:	–

⌋(*RS_Main_00300*)

[RS_METH_00041] Methodology shall support Bottom/Up Approach ⌈

Type:	valid
Description:	Methodology shall support the Bottom/Up (B/U) Approach. In this approach, all constraints coming from the Hardware in the B/U (ECUs/Sensors/Actuators) should be taken in account
Rationale:	To improve the integration phases, and to master the complexity in embedded RT distributed systems
Dependencies:	–

Use Case:	If in a given vehicle architecture, a new ECU is added or an existing ECU is replaced with a new one, all the new or modified resources from the ECU need to be included into the system configuration during integration.
Applies to:	AP
Supporting Material:	–

⌋(*RS_Main_00310*, *RS_Main_00350*)

[RS_METH_00018] Methodology shall be modular ⌈

Type:	valid
Description:	Utilize process components. Sub processes shall be complete and testable on their own to allow usage of certain portions of the Methodology while still integrating with legacy tools and processes.
Rationale:	Easier to understand and verify all portions of the Methodology. Easier to manage modifications, encapsulates ripple effect due to changes to allow migration of current processes. Easier to utilize both legacy and Autosar activities. It should be possible to start from an intermediate activity and not necessarily from the beginning of the methodology. A modular Methodology facilitates organizations to migrate from or merge with their current processes. A modular Methodology allows organizations to insert intermediate activities such as quality gates, or other inspections, as well as collect metrics necessary to comply with CMMI processes and/or SIL-3.
Dependencies:	–
Use Case:	An organization is planning to introduce an Autosar ECU into their existing architecture, but is not planning to use the System Template and their respective activities and work products. Rather they plan to begin directly at the ECU level.
Applies to:	AP
Supporting Material:	–

⌋(*RS_Main_00190*, *RS_Main_00300*, *RS_Main_00030*)

[RS_METH_00032] The methodology shall respect the different levels of abstractions ⌈

Type:	valid
Description:	The methodology shall respect the Software Component, the System, and the ECU levels of Abstractions.
Rationale:	To improve the integration phases and to master the complexity in embedded RT distributed systems.
Dependencies:	–
Use Case:	AUTOSAR is using several abstractions levels to describe the information exchanged between the different players. In an early phase the "Virtual Functional Bus" is used in other phases we are working with the implementation of the SWC in several ECUs. The exchange between the "real" world and the Virtual world should be described and supported by the methodology.
Applies to:	AP

Supporting Material:	–

⌋(*RS_Main_00130*, *RS_Main_00140*, *RS_Main_00330*, *RS_Main_00400*)

[RS_METH_00020] Methodology shall support iterations ⌈

Type:	valid
Description:	Need to support both small and large iteration loops.
Rationale:	Meet Autosar Quality requirements. Redoing work is error-prone. Support round trip engineering.
Dependencies:	[RS_METH_00062] < Methodology shall support configuration of parameters with different binding time > in [1]
Use Case:	Small iteration loops inside ECU Configuration Activity. Large iteration loops from System Design impacting ECU Configuration. Automotive systems are typically developed in several sample phases (A, B, C, etc). A Single Software Component is updated in a AUTOSAR System. The updated ECU Extract still matches the existing ECU Configuration (as long as no contradicting changes are made in the iteration).
Applies to:	AP
Supporting Material:	–

⌋(*RS_Main_00300*)

[RS_METH_00077] Methodology shall explain the typical interaction between OEMs and suppliers ⌈

Type:	valid
Description:	Methodology shall support use cases of interaction between OEM and supplier, where the OEM and the supplier have different views on the SW-C structure.
Rationale:	Methodology consistency using the System Templates
Dependencies:	–
Use Case:	The OEM hands over the initial System Extract to the supplier as a formal requirements specification. The supplier extends and refactors this System Extract. In the next development cycle the OEM hands over an updated System Extract to the supplier. Thereafter the supplier has to update his System Extract structure based on the updates made by the OEM. The amount of changes on the supplier side shall be minimized to the changes caused by OEM updates.
Applies to:	AP
Supporting Material:	–

⌋(*RS_Main_00300*)

[RS_METH_00078] Methodology shall explain the typical usage of different views on the system of the OEM ⌈

Type:	valid

Description:	Methodology shall support use cases of the OEM, where the OEM has different views on the system.
Rationale:	Methodology consistency using the System Templates
Dependencies:	–
Use Case:	An OEM might structure the AUTOSAR software components from a functional point of view. However, for the concrete vehicle development project a topological view of structure of SW-Cs is needed. For better handling during the lifecycle, the SW-Cs from the functional decomposition are mapped to the topological view using appropriate mappings.
Applies to:	AP
Supporting Material:	–

⌋(*RS_Main_00300*, *RS_Main_00150*)

[RS_METH_00079] Methodology shall explain the typical usage of different views on the system of the Supplier ⌈

Type:	valid
Description:	Methodology shall support use cases of the supplier where the supplier has different views on the system.
Rationale:	Methodology consistency using the System Templates
Dependencies:	–
Use Case:	The supplier needs to map different views of the system, e.g. a) the supplier already has an existing software architecture. Via software sharing some of the components are substituted by the ones delivered by the OEM. b) The supplier needs to formally describe changes between system descriptions representing different releases. c) The supplier develops one ECU for different OEMs and therefore needs to map the requirement-views of the OEMs to his solution view. d) The supplier realizes the OEMs definition for one ECU by 2 ECUs and therefore needs to map the system descriptions.
Applies to:	AP
Supporting Material:	–

⌋(*RS_Main_00300*, *RS_Main_00150*)

[RS_METH_00084] AUTOSAR methodology shall relate templates to a distributed development process ⌈

Type:	valid
Description:	The AUTOSAR templates specify the language for describing an AUTOSAR-based software or system. The methodology shall support the specification of a subset of the templates, which is used for a specific work product in a distributed development process.
Rationale:	Exchange of AUTOSAR artifacts in distributed development
Dependencies:	[RS_METH_00083]
Use Case:	A (VFB) system description shall only contain relevant information for development of SW-Cs without deployment to an ECU network yet.
Applies to:	AP
Supporting Material:	

⌋(*RS_Main_00301*)

2.1.2 Programming Language

[RS_METH_00015] Methodology shall be independent of programming language ⌈

Type:	valid
Description:	The methodology shall be independent of programming language by providing generic solutions. For portions that are necessarily dependent on the programming language, these sections shall be explicitly noted and modular such that the overall methodology can be tailored to accommodate other programming languages.
Rationale:	By appropriately structuring the methodology to support existing and emerging programming languages, the Methodology can be consistently and successfully applied across an entire vehicle.
Dependencies:	–
Use Case:	An ECU is built on a microcontroller optimized for programming language ABC. The Methodology explains when and how to specify and select the implementation of the software components deployed to that node that are compatible with that programming language.
Applies to:	AP
Supporting Material:	–

⌋(*RS_Main_00220*)

2.1.3 Activities

[RS_METH_00066] Methodology shall support activities that reference tools ⌈

Type:	valid
Description:	Activities may reference tools that help to complete the activity. The methodology shall describe these types of tools and when they are used.
Rationale:	By defining which tools are needed, the performers of the activity can ensure all the tools have been sourced and installed prior to beginning the activity. As well, the implementers of tools that are Autosar specific, have a clear understanding of what activities their tool should support and know what the input and output work products are available. This will help ensure interoperability of Autosar Tools.
Dependencies:	RS_METH_00021 in [1]
Use Case:	The activity "Generate RTE" requires an RTE generator Tool and a compiler.
Applies to:	AP
Supporting Material:	–

⌋(*RS_Main_00250*)

[RS_METH_00042] Methodology shall incorporate the usage of industry standard tools ⌈

Type:	valid
Description:	Where industry standard tools, such as compilers and linkers exist, the Methodology shall incorporate them.
Rationale:	AUTOSAR should not require the use of tools when industry standard tools already exist.
Dependencies:	–
Use Case:	Compilers are industry standard tools.
Applies to:	AP
Supporting Material:	–

⌋(*RS_Main_00250*)

2.1.4 Process Requirements

[RS_METH_00056] AUTOSAR methodology shall not be bound to a particular lifecycle model ⌈

Type:	valid
Description:	AUTOSAR methodology shall not be bound to a particular lifecycle model. Activities must be independent with respect to the time and phase of the development process they are executed.
Rationale:	Connection to company specific lifecycle model: The methodology shall enable the use of different lifecycle models such as e.g. V-Model, Rational Unified Process.
Dependencies:	–
Use Case:	If e.g. extreme programming is used, the test cases are created prior to the implementation. For most other development processes, the implementation is generated prior to the creation of test cases.
Applies to:	AP
Supporting Material:	–

⌋(*RS_Main_00507*)

2.2 Requirements for the Adaptive Platform

This section specifies requirements, which are valid for the Adaptive Platform only.

2.2.1 Main Requirements

[RS_METH_00201] Methodology shall explain how to design the services of a system ⌈

Type:	valid
Description:	The methodology shall explain how to describe services for service-oriented communication used in an Adaptive AUTOSAR system. The service interfaces consist of methods, events and fields, which need to be specified.
Rationale:	Consistent description of the information that is exchanged between applications.
Dependencies:	–
Use Case:	Specify a service interface, which consists of three events and one method.
Applies to:	AP
Supporting Material:	–

⌋*(RS_Main_00150, RS_Main_00060)*

[RS_METH_00206] Methodology shall explain how to configure the instances of services of a system ⌈

Type:	valid
Description:	The methodology shall explain the necessary steps for the deployment of services. This starts with the configuration of the deployment of service interfaces for the chosen network binding. The methodology shall further describe how service instances are defined and configured for a specific machine.
Rationale:	Complete description of service instances within a system.
Dependencies:	–
Use Case:	Define if service instances are required or provided as well as their search or offer criteria for service-oriented communication.
Applies to:	AP
Supporting Material:	–

⌋*(RS_Main_00505, RS_Main_00320)*

[RS_METH_00202] Methodology shall explain how to develop an Adaptive Application ⌈

Type:	valid
Description:	An Adaptive Application is developed based on the service interfaces. The methodology shall describe the necessary activities for first designing and then implementing the Adaptive Application.
Rationale:	Clear navigation with a description of possible development approaches for the application developer.
Dependencies:	–
Use Case:	Design a model of the software component with all necessary ports in order to use the service interfaces.
Applies to:	AP

Supporting Material:	–

⌋(*RS_Main_00080*, *RS_Main_00150*)

[RS_METH_00203] Methodology shall explain the high-level usage of the Manifest Specification ⌈

Type:	valid
Description:	The manifest contains all necessary information that is needed in order to deploy applications onto the Adaptive Platform. The methodology shall explain how this information will be collected, for the machine, the service instances as well as for the application itself, and later on how the manifest will be used for configuration purposes.
Rationale:	Methodology consistency using the Manifest Specification
Dependencies:	–
Use Case:	The Application Manifest is used for describing all deployment and execution related aspects of an executable instance.
Applies to:	AP
Supporting Material:	–

⌋(*RS_Main_00503*)

[RS_METH_00207] Methodology shall explain how to develop Platform Software for the Adaptive Platform ⌈

Type:	valid
Description:	The methodology shall explain how to develop the functional clusters for an Adaptive Platform.
Rationale:	Efficient development of Adaptive Platform.
Dependencies:	–
Use Case:	Development of the Execution and Communication Management of an Adaptive Platform.
Applies to:	AP
Supporting Material:	–

⌋(*RS_Main_00002*)

[RS_METH_00204] Methodology shall describe how to configure a machine for the Adaptive Platform ⌈

Type:	valid
Description:	The methodology shall describe the different steps for defining and configuring the machine so that software can be deployed on it. This step shall be independent of other development steps in order to ensure that software can be easily uploaded later without a new configuration of the machine.
Rationale:	Deployment or updating of software without adapting machine configuration.
Dependencies:	–
Applies to:	AP

Use Case:	Configuration of all ports and IP addresses on the machine for service-oriented communication.
Supporting Material:	–

⌋(*RS_Main_00503*, *RS_Main_00002*)

[RS_METH_00205] Methodology shall describe how to deploy software on the Adaptive Platform ⌈

Type:	valid
Description:	A SW package is the smallest unit for deployment onto an Adaptive AUTOSAR Platform instance. The methodology shall describe the content of a SW package and how it is deployed on the Adaptive Platform.
Rationale:	Complete description of application development workflow until software is deployed.
Dependencies:	–
Applies to:	AP
Use Case:	Downloading and deploying a software update.
Supporting Material:	–

⌋(*RS_Main_00503*, *RS_Main_00150*)

3 Change History

3.1 Change History FO 1.1.0

3.1.1 Added Requirements in FO 1.1.0

Id	Heading
[RS_METH_00201]	Methodology shall explain how to design the services of a system
[RS_METH_00202]	Methodology shall explain how to develop an Adaptive Application
[RS_METH_00203]	Methodology shall explain the high-level usage of the Manifest Specification
[RS_METH_00204]	Methodology shall describe how to configure a machine for the Adaptive Platform
[RS_METH_00205]	Methodology shall describe how to deploy software on the Adaptive Platform
[RS_METH_00206]	Methodology shall explain how to configure the instances of services of a system
[RS_METH_00207]	Methodology shall explain how to develop Platform Software for the Adaptive Platform

Table 3.1: Added Requirements in FO 1.1.0

3.1.2 Changed Requirements in FO 1.1.0

N/A

3.1.3 Deleted Requirements in FO 1.1.0

N/A