Document Title	Main Requirements
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	054
Document Classification	Auxiliary
Document Status	Final
Part of AUTOSAR Release	4.2.1

Document Change History		
Release	**Changed by**	**Change Description**
4.2.1	AUTOSAR Release Management	• New requirement for Secure Onboard Communication • New requirement for naming schemes and conventions
4.1.3	AUTOSAR Release Management	• Update of tracing information
4.1.2	AUTOSAR Release Management	• Editorial changes
4.1.1	AUTOSAR Administration	• Cleanup outdated glossary • Updated format of requirements according to the Standardization Template • Update tracing to Project Objectives
4.0.3	AUTOSAR Administration	• The following features are incorporated • Acceptance tests • multicore support • safety requirements
3.1.5	AUTOSAR Administration	• Changed [RS_Main_00270] •
3.1.4	AUTOSAR Administration	• Changed [Main90] and [Main 370] • Legal disclaimer revised
3.1.1	AUTOSAR Administration	• Legal disclaimer revised
3.0.1	AUTOSAR Administration	• Document meta information extended • Small layout adaptations made
2.1.16	AUTOSAR Administration	• “Advice for users” revised • “Revision Information” added
2.1.15	AUTOSAR Administration	• Legal disclaimer revised

Document Change History		
Release	**Changed by**	**Change Description**
2.0	AUTOSAR Administration	• Removed section “2.1 Recipients” • Update of section 3.1.1 • Changed [RS_Main_00011], [RS_Main_00060], [RS_Main_00300], [RS_Main_00310], [RS_Main_00320], [RS_Main_00330], [RS_Main_00340], [RS_Main_00350], [RS_Main_00360], [Main370], [Main380] • Use Case added • Changed [Main20] • Rationale extended • Changed [RS_Main_00010], [Main50], [RS_Main_00080], [Main90], [RS_Main_00130], [RS_Main_00230], [Main240] • Rationale added • Changed [RS_Main_00160], [RS_Main_00280] • Rationale and Use Case added • Changed [RS_Main_00300], [RS_Main_00310] • Headline and Short description written in active • Update of section 3.3 according to changes of 3.1.1
1.0	AUTOSAR Administration	• Initial release

Disclaimer

Advice for users

AUTOSAR specifications may contain exemplary items (exemplary reference models, "use cases", and/or references to exemplary technical solutions, devices, processes or software).

Any such exemplary items are contained in the specifications for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such specifications, nor any later documentation of AUTOSAR conformance of products actually implementing such exemplary items, imply that intellectual property rights covering such exemplary items are licensed under the same rules as applicable to the AUTOSAR Standard.

Table of Content

1 Scope of the document 5
2 How to read this document 6
2.1 Conventions used 6
2.2 Acronyms and Abbreviations 6
3 Requirements Tracing 7
4 Requirements Specification 8
4.1 Architecture 8
4.2 Basic Software and Runtime Environment 11
4.3 Methodology 15
4.4 Non-functional Requirements 20
4.5 Acceptance Testing 22
4.6 Processes 23
5 References 25

1 Scope of the document

Each partner has committed to the overall project objectives (PO) of AUTOSAR. The objectives are listed in the AUTOSAR Standard Info Pack V3.3 or in subsequent documents. AUTOSAR Standard Info Pack is an official communication paper of development partnership.

These objectives are not directly usable and have to be refined in order to generate the specific technical requirements. For this purpose, the AUTOSAR Main Requirements are established as a fundamental base to derive these specific requirements.

The goal of this document is to define the main requirements of AUTOSAR including its link to the AUTOSAR objectives.
The term AUTOSAR is used as a synonym of the development partnership and the technical product AUTomotive Open System ARchitecture.

2 How to read this document

Each requirement has its unique identifier starting with the prefix “RS_Main_” (for “Main Requirement”). For any review annotations, remarks or questions, please refer to this unique ID rather than chapter or page numbers!

2.1 Conventions used

- The representation of requirements in AUTOSAR documents follows the table specified in [TPS_STDT_00078]. In requirements, the following specific semantics are used (taken from Request for Comment RFC 2119 from the Internet Engineering Task Force IETF)
- The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119. Note that the requirement level of the document in which they are used modifies the force of these words.
- MUST: This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.
- MUST NOT: This phrase, or the phrase „SHALL NOT“, means that the definition is an absolute prohibition of the specification.
- SHOULD: This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
- SHOULD NOT: This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.
- MAY: This word, or the adjective „OPTIONAL“, means that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation, which does not include a particular option, MUST be prepared to interoperate with another implementation, which does include the option, though perhaps with reduced functionality. In the same vein an implementation, which does include a particular option, MUST be prepared to interoperate with another implementation, which does not include the option (except, of course, for the feature the option provides.)

2.2 Acronyms and Abbreviations

All acronyms and abbreviations used throughout this document are included in the official AUTOSAR glossary [Glossary]. For respective explanation please see there.

3 Requirements Tracing

The following table references the requirements specified in **[RS_ProjectObjectives]** and links to the fulfilments of these.

Requirement	Description	Satisfied by
RS_PO_00001	AUTOSAR shall support the transferability of software.	RS_Main_00060, RS_Main_00100, RS_Main_00130, RS_Main_00140, RS_Main_00150, RS_Main_00270, RS_Main_00310, RS_Main_00400, RS_Main_00410, RS_Main_00440, RS_Main_00450, RS_Main_00460, RS_Main_00480
RS_PO_00002	AUTOSAR shall support the scalability to different vehicle and platform variants.	RS_Main_00060, RS_Main_00130, RS_Main_00140, RS_Main_00150, RS_Main_00230, RS_Main_00310, RS_Main_00360, RS_Main_00400
RS_PO_00003	AUTOSAR shall support a broad variety of functional domains.	RS_Main_00190, RS_Main_00210, RS_Main_00280, RS_Main_00410
RS_PO_00004	AUTOSAR shall define an open architecture for automotive software.	RS_Main_00080, RS_Main_00170, RS_Main_00220, RS_Main_00260, RS_Main_00280, RS_Main_00320, RS_Main_00410, RS_Main_00430, RS_Main_00435, RS_Main_00440, RS_Main_00450, RS_Main_00460, RS_Main_00510
RS_PO_00005	AUTOSAR shall support the development of dependable systems.	RS_Main_00010, RS_Main_00011, RS_Main_00030, RS_Main_00170, RS_Main_00260, RS_Main_00340, RS_Main_00350, RS_Main_00480, RS_Main_00490, RS_Main_00510
RS_PO_00006	AUTOSAR shall support sustainable utilization of natural resources.	RS_Main_00200
RS_PO_00007	AUTOSAR shall support the collaboration between various partners.	RS_Main_00060, RS_Main_00080, RS_Main_00100, RS_Main_00160, RS_Main_00180, RS_Main_00250, RS_Main_00251, RS_Main_00300, RS_Main_00310, RS_Main_00320, RS_Main_00330, RS_Main_00420, RS_Main_00500
RS_PO_00008	AUTOSAR shall standardize basic software functionality of automotive ECUs.	RS_Main_00120, RS_Main_00280, RS_Main_00510
RS_PO_00009	AUTOSAR shall support applicable international automotive standards and state-of-the-art technologies.	RS_Main_00010, RS_Main_00011, RS_Main_00030, RS_Main_00170, RS_Main_00260, RS_Main_00290, RS_Main_00350, RS_Main_00420, RS_Main_00430, RS_Main_00490

4 Requirements Specification

4.1 Architecture

4.1.1 [RS_Main_00400] AUTOSAR shall provide a layered software architecture

⌈

Type:	valid
Description:	AUTOSAR shall provide a layered software architecture, which distinguishes between Software Components, Runtime Environment and Basic Software.
Rationale:	The Runtime Environment together with the Basic Software provides an abstraction layer for the Application Software. This enables the reallocation and reuse of Software Components.
Use Case:	Relocation of yaw rate control from one ECU to another.
Dependencies:	--
Supporting Material:	

⌋(RS_PO_00001, RS_PO_00002)

4.1.2 [RS_Main_00130] AUTOSAR shall provide an abstraction from hardware

⌈

Type:	valid
Description:	A MicroController Abstraction Layer (MCAL) shall provide an abstraction from hardware characteristics. This MCAL shall be limited to the software modules directly interacting with the HW. Other parts of the Basic Software shall be independent from hardware.
Rationale:	Reuse of the SW if the HW is changed (e.g. new microcontroller). Software sharing between multiple HW platforms.
Use Case:	Relocate SW application from one ECU with hardware A to ECU with hardware B without changes of the SW application.
Dependencies:	--
Supporting Material:	

⌋(RS_PO_00001, RS_PO_00002)

4.1.3 [RS_Main_00150] AUTOSAR shall support the reallocation of Software Components

⌈

Type:	valid
Description:	AUTOSAR shall develop mechanisms, methods, and processes to enable reallocation of Software Components independent from communication infrastructure.
Rationale:	Enable the reallocation of Software Components to different ECUs.
Use Case:	Reallocation of yaw rate control from one ECU to another. Optimization of overall system architecture.
Dependencies:	RS_Main_00140, Main141
Supporting Material:	

⌋(RS_PO_00001, RS_PO_00002)

4.1.4 [RS_Main_00060] AUTOSAR shall provide a standardized software interface for communication between Software Components

⌈

Type:	valid
Description:	As the interface definition of Software Components is a prerequisite to reuse Software Components AUTOSAR shall provide such a standardized interface.
Rationale:	From a Software Components point of view the interface to another Software Component shall be identical, independent whether the Software Components are located • on the same ECU on the same core • on the same ECU on another core • on the same ECU on another microcontroller • on another ECU
Use Case:	Application SW is developed independently from the underlying communication system.
Dependencies:	--
Supporting Material:	

⌋(RS_PO_00001, RS_PO_00002, RS_PO_00007)

4.1.5 [RS_Main_00140] AUTOSAR shall provide network independent communication mechanisms for applications

⌈

Type:	valid
Description:	AUTOSAR shall support the development of Software Components independently from the underlying communication systems.
Rationale:	Independency from the underlying communication system is a prerequisite to support the reallocation of Software Components.
Use Case:	Relocation of SW component from ECU A with CAN-Bus to ECU B with FlexRay.
Dependencies:	--
Supporting Material:	

⌋(RS_PO_00001, RS_PO_00002)

4.1.6 [RS_Main_00410] AUTOSAR shall provide specifications for routines commonly used by Software Components to support sharing and optimization

⌈

Type:	valid
Description:	AUTOSAR shall support the development of Software Components by providing standardized libraries with commonly used functions.
Rationale:	Share routines between different SW-Component. Use of optimized routines by SW-Components integrated in different ECUs.
Use Case:	Relocation of SW component from ECU A to ECU B with a different microcontroller.
Dependencies:	--
Supporting Material:	

⌋(RS_PO_00001, RS_PO_00003, RS_PO_00004)

4.1.7 [RS_Main_00190] AUTOSAR shall support interoperability with non-AUTOSAR software on the same ECU

⌈

Type:	valid
Description:	Reuse of existing legacy software shall be supported by AUTOSAR. Integration of legacy software in an ECU complaint to AUTOSAR shall be supported. To ensure interoperability the interface definitions of the AUTOSAR architecture shall not be modified due to the integration of legacy software.
Rationale:	A smooth migration to AUTOSAR requires that existing software can be reused.
Use Case:	Reuse of existing driver software for a new ECU that is developed according to AUTOSAR. Communication with non-AUTOSAR software systems.
Dependencies:	--
Supporting Material:	

⌋(RS_PO_00003)

4.1.8 [RS_Main_00210] AUTOSAR shall support interoperability with non-AUTOSAR ECUs in a network

⌈

Type:	valid
Description:	AUTOSAR shall support the integration of AUTOSAR ECUs and non-AUTOSAR ECUs on the same network. This implies the support of legacy communication patterns.
Rationale:	This opens a migration path for the application of AUTOSAR into existing non-AUTOSAR architectures.
Use Case:	Integration of an AUTOSAR-ECU into a network with non-AUTOSAR ECUs. Integration of a non AUTOSAR-ECU on a network where most ECUs migrated to AUTOSAR.
Dependencies:	--
Supporting Material:	

⌋(RS_PO_00003)

4.1.9 [RS_Main_00330] AUTOSAR shall support the principle of information hiding

⌈

Type:	valid
Description:	AUTOSAR shall provide protection mechanisms for internal information of Software Components.
Rationale:	Well-defined interfaces and data flow between Software Components.
Use Case:	Reduce impact of modifications within a Software Component.
Dependencies:	--
Supporting Material:	

⌋(RS_PO_00007)

4.1.10 [RS_Main_00230] AUTOSAR shall support network topologies including gateways

⌈

Type:	valid
Description:	AUTOSAR shall support net topologies with different in-vehicle network technologies. Interconnection of these networks via gateways, bridges, or repeaters shall be supported.
Rationale:	ECUs communicate via different communication systems.
Use Case:	Support of today's network topologies of E/E-architectures in series production.
Dependencies:	--
Supporting Material:	

⌋(RS_PO_00002)

4.2 Basic Software and Runtime Environment

4.2.1 [RS_Main_00100] AUTOSAR shall provide standardized basic software

⌈

Type:	valid
Description:	AUTOSAR shall provide a complete specification of the Basic Software including interfaces and behavioral description.
Rationale:	The Basic Software provides the infrastructural functionalities of an ECU and is per definition not visible to the customer. To support reallocation of Software Components it is necessary that the Software Components can rely on identical services provided by the Basic Software. This basic software functionality shall be defined, implemented, and tested once but integrated in many ECUs of an E/E-architecture. Achieving this for many OEMs the suppliers and OEMs have stable foundations for implementing their application without bothering about details of the basic software. This leads to a higher quality and lower costs for automotive software.
Use Case:	Compatibility of the basic SW with defined quality and maintenance is achieved. Software Components shall be useable on multiple implementations of the Basic Software.
Dependencies:	--
Supporting Material:	

⌋(RS_PO_00001, RS_PO_00007)

4.2.2 [RS_Main_00420] AUTOSAR shall use established software standards and consolidate de-facto standards for basic software functionality

⌈

Type:	valid
Description:	The different solutions for basic software functionalities shall be consolidated to a single standard. Whenever possible AUTOSAR shall make use of existing standards provided that they meet the given requirements.
Rationale:	Historically, OEMs and the big Tier1 have created proprietary standard core solutions, with partly different functionality. To achieve a common standard, which is accepted and used by all of the participating partners these solution shall be consolidated by AUTOSAR. If an agreed common solution supported by OEMs and Tier 1 already exists, this solution shall be adopted by AUTOSAR in order to ease reuse of existing software.
Use Case:	Usage of OSEK Operating System in AUTOSAR ECUs. Partial Networking. Network Management.
Dependencies:	--

Supporting Material:	--

⌋(RS_PO_00007,RS_PO_00009)

4.2.3 [RS_Main_00430] AUTOSAR shall support established automotive communication standards

⌈

Type:	valid
Description:	AUTOSAR ECUs shall support common established communication systems. This includes at least but is not restricted to: CAN, LIN, FlexRay, Ethernet
Rationale:	Automotive ECUs communicate over different standardized communication systems. These shall be supported by AUTOSAR.
Use Case:	Implementation of distributed functionality e. g. driving assistance systems
Dependencies:	--
Supporting Material:	--

⌋(RS_PO_00004,RS_PO_00009)

4.2.4 [RS_Main_00435] AUTOSAR shall support automotive microcontrollers

⌈

Type:	valid
Description:	The AUTOSAR MCAL shall support hardware features of commonly used automotive microcontrollers.
Rationale:	Automotive ECUs use dedicated, highly integrated microcontrollers, which have to pass automotive qualification procedures. The AUTOSAR MCAL shall support the integrated features of these microcontrollers. These includes, but is not limited to: • Digital I/O • Analog/Digital converter • Pulse-width modulation • Bus controllers for CAN, LIN, FlexRay, Ethernet • Multicore • Memory protection units • Flash
Use Case:	Implementation of safety relevant systems Avoid complex drivers for commonly used features
Dependencies:	--
Supporting Material:	--

⌋(RS_PO_00004)

4.2.5 [RS_Main_00440] AUTOSAR shall standardize access to non-volatile memory

⌈

Type:	valid
Description:	An important functionality of Basic Software is the access non-volatile memory. The AUTOSAR Basic Software shall support access (read / write / management) to non-volatile memory.

Rationale:	Software Components need to store data, which is available after a restart.
Use Case:	Storage of error codes
Dependencies:	--
Supporting Material:	--

⌋(RS_PO_00001, RS_PO_00004)

4.2.6 [RS_Main_00450] AUTOSAR shall standardize access to general purpose I/O

⌈

Type:	valid
Description:	The AUTOSAR Basic Software shall support access to general purpose I/O.
Rationale:	Software Components need to access application specific hardware (sensor and actuators)
Use Case:	Temperature sensor for engine control.
Dependencies:	--
Supporting Material:	--

⌋(RS_PO_00001, RS_PO_00004)

4.2.7 [RS_Main_00460] AUTOSAR shall standardize methods to organize mode management on SWC, ECU and system level

⌈

Type:	valid
Description:	The AUTOSAR Basic Software shall provide mode management mechanisms for SW-C to control or react on modes of the ECU / vehicle
Rationale:	The behavior of Application Software Component highly depends on the overall mode of the ECU and/or the System. Therefore the overall mode management has to be standardized to achieve the same behavior if Software Components are transferred from one ECU to another ECU or from one System to another System
Use Case:	Degradation of application functionality in certain power modes.
Dependencies:	--
Supporting Material:	--

⌋(RS_PO_00001, RS_PO_00004)

4.2.8 [RS_Main_00170] AUTOSAR shall provide secure access to ECU

⌈

Type:	valid
Description:	AUTOSAR shall provide secure access to ECU, (e.g. by user authentication), including standardized up- and download of data and software. For this mechanisms and methods shall be defined.
Rationale:	The update and upgrade feasibility provided by AUTOSAR includes technical challenges (e.g. standardized up-/download protocol, partly update of the software) and mechanisms (e.g. how to authorize the user).
Use Case:	Download of dedicated Software Components in ECU.
Dependencies:	To fulfill this requirement it is also necessary that the environment that is not standardized by AUTOSAR (e.g. bootloader) needs to match the same security requirements.
Supporting Material:	--

⌋(RS_PO_00004, RS_PO_00005, RS_PO_00009)

4.2.9 [RS_Main_00510] AUTOSAR shall support secure onboard communication

⌈

Type:	valid
Description:	AUTOSAR shall provide means to check data authenticity, data integrity and data freshness in inter ECU communication.
Rationale:	Dependable systems rely on authentic and trustworthy exchange of information between ECUs. Protecting and assuring data authenticity, data integrity and data freshness in inter ECU communication allows for the development of secure and safe systems by using the AUTOSAR platform.
Use Case:	Protection of on-board communication against manipulation.
Dependencies:	--
Supporting Material:	--

⌋(RS_PO_00004, RS_PO_00005, RS_PO_00008)

4.2.10 [RS_Main_00260] AUTOSAR shall provide diagnostics means during runtime, for production and services purposes

⌈

Type:	valid
Description:	AUTOSAR shall at least support diagnostics standards like ISO14229 or OBD. Specifications of error handling must be developed. Furthermore, AUTOSAR shall reflect the invention of encapsulated SW in its diagnostics specifications.
Rationale:	Standardized diagnostics is necessary for field service and admission.
Use Case:	Diagnosis of an encapsulated SW function or a ECU.
Dependencies:	--
Supporting Material:	

⌋(RS_PO_00004, RS_PO_00005, RS_PO_00009)

4.2.11 [RS_Main_00280] AUTOSAR shall provide a communication interface to the infotainment systems

⌈

Type:	valid
Description:	AUTOSAR shall provide a communication interface to infotainment systems, enabling the data exchange with AUTOSAR systems.
Rationale:	Dynamic communication pattern are well established in the infotainment domain. Infotainment systems (Multimedia, Human Machine Interface) differ in several aspects from systems of other automotive domains (body, powertrain, chassis, driving assistance). They use dynamic schedulers, dynamic communication patterns, etc.
Use Case:	Display of engine related data in an head up display. Configuration of different chassis modes via human machine interface.
Dependencies:	--
Supporting Material:	--

⌋(RS_PO_00003, RS_PO_00004, RS_PO_00008)

Dependability

4.2.12 [RS_Main_00010] AUTOSAR shall provide a software platform to support the development of safety related systems.

⌈

Type:	valid
Description:	A safe AUTOSAR platform supports commonly used safety mechanisms of current applications. This includes but is not limited to: • Mechanisms to ensure freedom of interference for Software Components and Basic Software Modules • Safe inter ECU communication However, it cannot ensure safety for systems. This is to be ensured on system level.
Rationale:	Facilitate the development of safety related systems by using the AUTOSAR platform.
Use Case:	Safety related software platforms are needed for safety related ECUs like digital engine control units and electronic power steering systems.
Dependencies:	--
Supporting Material:	ISO 26262

⌋(RS_PO_00005, RS_PO_00009)

4.2.13 [RS_Main_00011] AUTOSAR shall support the development of reliable systems

⌈

Type:	valid
Description:	AUTOSAR shall provide mechanisms for • error avoidance • error detection • error handling in order to support the development of reliable systems, which are properly functioning over time.
Rationale:	Reliability is one possibility to achieve safety without decreasing availability.
Use Case:	Reduction of repair time of a vehicle in field service.
Dependencies:	--
Supporting Material:	

⌋(RS_PO_00005, RS_PO_00009)

4.3 Methodology

4.3.1 [RS_Main_00160] AUTOSAR shall provide means to describe interfaces of the entire system.

⌈

Type:	valid
Description:	Well-defined interfaces are the key for exchangeability, reusability and the basis for the freedom of interference. Decomposition on interface level is essential for an appropriate clustering and partitioning in the Software Components.
Rationale:	Principle: “divide and conquer” which is a key success factor in the development of large systems.
Use Case:	Development of large interconnected software systems with a high degree of reuse, such as driving assistance systems.

Dependencies:	--
Supporting Material:	

⌋(RS_PO_00007)

4.3.2 [RS_Main_00180] AUTOSAR shall provide mechanisms to protect intellectual property in a shared development process

⌈

Type:	valid
Description:	Integration of software of different suppliers requires exchange of software (especially source code) between the different parties involved. Thus, AUTOSAR shall provide mechanisms to safeguard software. AUTOSAR shall ensure a smooth integration process that at the same time protects intellectual property of the companies involved.
Rationale:	Integration of third party solutions requires dealing with intellectual property issues.
Use Case:	1) SW sale of split-screen software for navigation. 2) Integration of BSW modules of different suppliers.
Dependencies:	--
Supporting Material:	

⌋(RS_PO_00007)

4.3.3 [RS_Main_00300] AUTOSAR shall provide data exchange formats to support work-share in large inter and intra company development groups

⌈

Type:	valid
Description:	AUTOSAR shall support the work-share in large development projects via well-defined exchange formats.
Rationale:	A typical AUTOSAR system is expected to carry a huge number of signals per vehicle. To develop vehicle descriptions in this size a good organization of work-share is needed. To support such organizations, well defined concepts for information exchange are required.
Use Case:	Data sharing between OEM and 1st Tier supplier.
Dependencies:	--
Supporting Material:	

⌋(RS_PO_00007)

4.3.4 [RS_Main_00080] AUTOSAR shall provide means to describe a component model for application software

⌈

Type:	valid
Description:	AUTOSAR shall provide a formal description of components of application software named Software Component Description (SWCD). This description together with source code and/or object code forms a Software Component. AUTOSAR shall provide a partitioning of the functionality into reusable Software Components.
Rationale:	Software reuse is one of the major aims of AUTOSAR.
Use Case:	Momentum control in different ECUs.
Dependencies:	--
Supporting Material:	

⌋(RS_PO_00004, RS_PO_00007)

4.3.5 [RS_Main_00310] AUTOSAR shall support hierarchical design methods

⌈

Type:	valid
Description:	It must be possible to structure Software components in a hierarchical way, so that only links to outside Software Components need to be treated / adapted / changed in the next hierarchical level.
Rationale:	Objective is to allow each actor in the development chain to focus on the required level and tasks.
Use Case:	Software development of an engine management system can only be achieved by using hierarchical strategies.
Dependencies:	--
Supporting Material:	

⌋(RS_PO_00001, RS_PO_00002, RS_PO_00007)

4.3.6 [RS_Main_00320] AUTOSAR shall provide formats to specify all aspects necessary to integrate a Software Component on an ECU

⌈

Type:	valid
Description:	In AUTOSAR it shall be possible to describe all requirements of Software Components to their runtime environment. This enables the integrator to provide the Software Component such a environment on an ECU.
Rationale:	Software Components need a well defined environment to ensure correct and reliable functionality.
Use Case:	OEM designs Software Component and a 1st Tier Supplier shall integrate this SW component into an ECU. In this case exhaustive and formal description of the API is necessary
Dependencies:	--
Supporting Material:	

⌋(RS_PO_00004, RS_PO_00007)

4.3.7 [RS_Main_00340] AUTOSAR shall support the observance of timing requirements

⌈

Type:	valid
Description:	AUTOSAR shall support the description and correct implementation of timing requirements. This includes dataflow and control flow related requirements.
Rationale:	E.g. Software components have specific timing requirements, which are not to be overwritten by architectural changes.
Use Case:	Real time control of today's gasoline injection systems.
Dependencies:	--
Supporting Material:	

⌋(RS_PO_00005)

4.3.8 [RS_Main_00360] AUTOSAR shall support management of vehicle diversity

⌜

Type:	valid
Description:	Diversity management is introduced on vehicle level. This enables formal check of compatibility of Software Components, management of availability of partner Software Components in vehicle versions their release state etc. Also the number of required software versions per ECU integration can be evaluated and tracked with reasonable effort.
Rationale:	Diversity of e.g. a wiring harness is reaching the amount of 10 000 to 1.000.000 different versions for the same vehicle platform. Same diversity requirements multiplied with the version management per Software Component apply for the entity of the software on the vehicle level. Unmanaged this effect can lead to deadlock situations in the logistic of vehicle software. Potential implementations are e.g. "existence" property matrix per class, group and instance of SW components and connections.
Use Case:	Integration of Software Components in different ECUs and/or E/E-architectures.
Dependencies:	--
Supporting Material:	

⌟(RS_PO_00002)

4.3.9 [RS_Main_00250] AUTOSAR process shall provide a predefinition of typical roles and activities in work-share model

⌜

Type:	valid
Description:	Work-share requires visibility and common understanding of specific roles and activities in the design process. Roles could be e.g. vehicle architect, domain architect, ECU integrator, function designer. Roles could also be found related to different disciplines like static architecture, dynamic architecture, communication architecture, etc. Typical activities could be e.g. SW/SW partitioning, mapping of SW to ECU, configuration of Basic Software.
Rationale:	This definition serves the understanding of workflow for the design of the tools. Individual roles and activities might be combined to one person in the later application, or distributed differently. This also does not assign tasks to OEM or Tier1 suppliers. Application of later work-share-distribution is matter of individual contracts between the parties.
Use Case:	System architect or SW component developer are such roles which shall be predefined.
Dependencies:	--
Supporting Material:	

⌟(RS_PO_00007)

4.3.10 [RS_Main_00251] AUTOSAR process shall support roles and rights in a work-share model

⌜

Type:	valid
Description:	Distributed Development needs definable responsibilities. AUTOSAR shall support a general approach to express roles and their rights to create/access/modify AUTOSAR descriptions.
Rationale:	The definition of roles and the corresponding rights guide the user while

	editing AUTOSAR descriptions and reduce the risk to introduce errors.
Use Case:	It is forbidden e.g. for SW component developers to change the configuration of basic software modules.
Dependencies:	--
Supporting Material:	

⌋(RS_PO_00007)

4.4 Non-functional Requirements

4.4.1 [RS_Main_00200] AUTOSAR specifications shall allow resource efficient implementations

⌈

Type:	valid
Description:	AUTOSAR specifications shall allow efficient implementations with respect to • RAM • ROM, Flash • Computing Power
Rationale:	Limited resources like flash, RAM, computing power characterize automotive microcontrollers.
Use Case:	Integration of the AUTOSAR platform and a single application in a typical 16-bit automotive microcontroller.
Dependencies:	--
Supporting Material:	--

⌋(RS_PO_00006)

4.4.2 [RS_Main_00220] The functional interfaces of AUTOSAR shall be specified in C90

⌈

Type:	valid
Description:	In today's embedded systems only a small range of programming languages are used: C and C++. The specification of functional interfaces of AUTOSAR shall be specified in C90 according to ISO/IEC 9899:1990. This implies that languages, which can interface to C90 can be used for application programming.
Rationale:	A useful reduction of programming languages to current programming languages reduces the impacts on AUTOSAR definitions and specifications due to logical and/or technical differences of different programming languages.
Use Case:	AUTOSAR implementation in C, C++.
Dependencies:	--
Supporting Material:	ISO/IEC 9899:1990 (C90)

⌋(RS_PO_00004)

4.4.3 [RS_Main_00270] AUTOSAR shall provide mitigation strategies towards new releases

⌈

Type:	valid
Description:	Migration from AUTOSAR release n to release n+1 shall be supported well. AUTOSAR shall provide migration strategies, how Software Components and ECUs of different release have to be adapted to interoperate.
Rationale:	Compatibility ensures a long time usage of the AUTOSAR standard.
Use Case:	Integration of ECU's using infrastructure software of the latest AUTOSAR release in a network built from ECU's using a former release.
Dependencies:	--
Supporting Material:	--

⌋(RS_PO_00001)

4.4.4 [RS_Main_00500] AUTOSAR shall provide naming conventions

⌈

Type:	valid
Description:	AUTOSAR shall define naming conventions for internal and external symbols created and used by the standard.
Rationale:	Naming conventions shall be defined in specification documents to achieve a standardized and consistent documentation. This is good documentary practice, helps for better understanding, reduces ambiguities and improves cooperation
Use Case:	Work-share models between OEM and supplier. Development of AUTOSAR specifications.
Dependencies:	--
Supporting Material:	--

⌋(RS_PO_00007)

4.5 Acceptance Testing

4.5.1 [RS_Main_00120] AUTOSAR shall provide means to assure interoperability of AUTOSAR implementations (ICC1 level) on application level (RTE) and bus level.

⌜

Type:	valid
Description:	AUTOSAR shall provide specified test cases and the essential test methodology to ensure interoperability on application (RTE side) and bus level for BSW on ICC1 level (Black Box Test). These specified test cases and its related methodology shall be developed to test implementations of AUTOSAR basic software.
Rationale:	Acceptance tests are strongly needed to provide evidence that a product complies with the AUTOSAR specification i.e. to ensure a certain behavior of the regarded elements at the interfaces to application and communication busses.
Use Case:	Integration of the infrastructure SW into a specific ECU, bring it into the E/E-architecture without backlashes on the system. Example from real world: Integration of BSW stack (ICC1 level) to applications and the ECU infrastructure without difficulties.
Dependencies:	--
Supporting Material:	--

⌟(RS_PO_00008)

4.6 Processes

4.6.1 [RS_Main_00030] AUTOSAR shall support development processes for safety related systems

⌈

Type:	valid
Description:	To develop safety related automotive systems all processes applied need to follow the corresponding requirements given in ISO26262. AUTOSAR shall support development processes for safety related systems by providing according exchange formats (e.g. for requirements tracing) and concepts.
Rationale:	Automotive software is in many cases safety related. Therefore dedicated development processes have to be followed. AUTOSAR shall support the users to apply these standards
Use Case:	Development of brake assist
Dependencies:	--
Supporting Material:	ISO26262

⌋(RS_PO_00005, RS_PO_00009)

4.6.2 [RS_Main_00490] AUTOSAR processes shall be compliant to ISO26262

⌈

Type:	valid
Description:	To develop safety related automotive systems all processes applied need to follow the corresponding requirements given in ISO26262. Accordingly the applicable process related requirements of ISO26262 have to be fulfilled by AUTOSAR processes.
Rationale:	AUTOSAR shall support the development of systems according to the highest ASIL.
Use Case:	Development of safety related automotive systems.
Dependencies:	--
Supporting Material:	ISO26262

⌋(RS_PO_00005, RS_PO_00009)

4.6.3 [RS_Main_00290] AUTOSAR shall support the verification of its specifications

⌈

Type:	Valid
Description:	To develop safety related automotive systems all development artifacts need to be verified. Accordingly the development artifacts of AUTOSAR shall be verified.
Rationale:	AUTOSAR shall support the development of systems according to the highest ASIL.
Use Case:	Development of safety related automotive systems.
Dependencies:	--
Supporting Material:	ISO26262

⌋(RS_PO_00009)

4.6.4 [RS_Main_00350] AUTOSAR specifications shall be analyzable and support according methods to demonstrate the achievement of safety related properties.

⌜

Type:	Valid
Description:	To achieve safety-related properties an adequate software architectural design and implementation matching the safety requirements is required and has to be demonstrated. Such demonstration can be done by safety analyses, therefore. • AUTOSAR specifications shall be analyzable accordingly. • Corresponding analysis methods shall be applicable to the development artifacts specified by AUTOSAR.
Rationale:	In the context of the safety-related developments a confirmation that design and implementation are adequately safe is required.
Use Case:	--
Dependencies:	--
Supporting Material:	ISO26262

⌟(RS_PO_00005, RS_PO_00009)

4.6.5 [RS_Main_00480] AUTOSAR shall support the test of implementations

⌜

Type:	Valid
Description:	AUTOSAR shall support the testability of compliant products.
Rationale:	Testing of implementations is required by software development, software maturity and software safety standards.
Use Case:	Software unit and integration testing of the implementation of AUTOSAR Basic Software Modules.
Dependencies:	--
Supporting Material:	ISO26262

⌟(RS_PO_00001, RS_PO_00005)

5 References

[Glossary] Glossary,
AUTOSAR_TR_Glossary.pdf

[ISO 26262] Road vehicles — Functional safety — Part 1 to 9

[TPS_STDT] Standardization Template,
AUTOSAR_TPS_StandardizationTemplate.pdf

[RS_ProjectObjectives] Project Objectives
AUTOSAR_RS_ProjectObjectives.pdf