Document Title	Requirements on Platform Health Management
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	852

Document Status	published
Part of AUTOSAR Standard	Adaptive Platform
Part of Standard Release	R22-11

Document Change History

Date	Release	Changed by	Description
2022-11-24	R22-11	AUTOSAR Release Management	• Added RS_PHM_00114, RS_PHM_00115, RS_PHM_00116 and RS_PHM_00117 • Modified RS_PHM_00111 (Replaced Local Supervision with Elementary Supervision) • Cleanup of requirement trace
2021-11-25	R21-11	AUTOSAR Release Management	• Added RS_PHM_09255, RS_PHM_09257, RS_PHM_09240, RS_PHM_09241 (moved from FO) • Removed RS_PHM_00110 • Cleanup of requirement trace
2020-11-30	R20-11	AUTOSAR Release Management	• Marked Health Channel related items as obsolete • Added RS_PHM_00111 and RS_PHM_00112 for Mode Dependent Configuration • Modified description of Supervision Mode in RS_PHM_00104
2019-11-28	R19-11	AUTOSAR Release Management	• No content changes • Changed Document Status from Final to published
2019-03-29	19-03	AUTOSAR Release Management	• removed references to RS_Main_00330

2018-10-31	18-10	AUTOSAR Release Management	• minor corrections / clarifications / editorial changes
2018-03-29	18-03	AUTOSAR Release Management	• Initial release

Disclaimer

This work (specification and/or software implementation) and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only. For any other purpose, no part of the work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Contents

1 Scope of Document

This document specifies requirements on Platform Health Management. Platform Health Management implements the Platform Health Monitoring on the AUTOSAR Adaptive Platform.

2 Conventions to be used

The representation of requirements in AUTOSAR documents follows the table specified in [TPS_STDT_00078], see Standardization Template [1], chapter Support for Traceability.

The verbal forms for the expression of obligation specified in [TPS_STDT_00053] shall be used to indicate requirements, see Standardization Template [1], chapter Support for Traceability.

3 Acronyms and abbreviations

The glossary below includes acronyms and abbreviations relevant to the specification or implementation of Health Monitoring that are not included in the [2, AUTOSAR glossary].

Abbreviation:	Description:
CM	AUTOSAR Adaptive Communication Management
DM	AUTOSAR Adaptive Diagnostic Management
PHM	Platform Health Management
SE	Supervised Entity

Acronym:	Description:
Alive Supervision	Mechanism to check the timing constraints of cyclic Supervised Entityes to be within the configured min and max limits.
Application	see [2] AUTOSAR Glossary
ara::com	Communication middleware for the AUTOSAR Adaptive Platform
AUTOSAR Adaptive Platform	see [2] AUTOSAR Glossary
Checkpoint	A point in the control flow of a Supervised Entity where the activity is reported.
Daisy chaining	Chaining multiple instances of Health Monitoring
Deadline End Checkpoint	A Checkpoint for which Deadline Supervision is configured and which is a ending point for a particular Transition. It is possible that a Checkpoint is both a Deadline Start Checkpoint and Deadline End Checkpoint - if Deadline Supervision is chained.
Deadline Start Checkpoint	A Checkpoint for which Deadline Supervision is configured and which is a starting point for a particular Transition.
Deadline Supervision	Mechanism to check that the timing constraints for execution of the transition from a Deadline Start Checkpoint to a corresponding Deadline End Checkpoint are within the configured min and max limits.
Elementary Supervision Status	Status that represents the current state of an Alive Supervision, Deadline Supervision or Logical Supervision, based on the evaluation (correct/incorrect) of the supervision.
Executable	see [2] AUTOSAR Glossary
Execution Management	The element of the Adaptive Platform responsible for the ordered startup and shutdown of the Adaptive Platform and the Application.

Function Group	A Function Group is a set of coherent Processes, which need to be controlled consistently. Depending on the state of the Function Group, Processes are started or terminated.
Function Group State	The element of State Management that characterizes the current status of a set of (functionally coherent) user-level Applications. The set of Function Groups and their Function Group States is machine specific and are deployed as part of the Machine Manifest.
Functional Cluster	see [2] AUTOSAR Glossary
Global Supervision Status	Status that summarizes the Elementary Supervision Status of a set of supervisions within a Function Group.
Health Channel	Channel providing information about the health status of a (sub)system. This might be the Global Supervision Status of an application, the result any test routine or the status reported by a (sub)system (e.g. voltage monitoring, OS kernel, ECU status, ...).
Health Monitoring	Supervision of the software behaviour for correct timing and sequence.
Health Status	A set of states that are relevant to the supervised software (e.g. the Global Supervision Status of an application, a Voltage State, an application state, the result of a RAM monitoring algorithm).
Logical Supervision	Kind of online supervision of software that checks if the software (Supervised Entity or set of Supervised Entities) is executed in the sequence defined by the programmer (by the developed code).
Machine Manifest	Manifest file to configure a Machine.
Machine	see [2] AUTOSAR Glossary
Machine State	The element of the State Management which characterize the current status of the machine. It defines a set of active Applications for any certain situation. The set of Machine States is machine specific and it will be deployed in the Machine Manifest. Machine States are mainly used to control machine lifecycle (startup/shut-down/restart) and platform-level processes.
Manifest	see [2] AUTOSAR Glossary
Platform Health Management	Health Monitoring for the Adaptive Platform
Process	A process is a loaded instance of an Executable to be executed on a Machine.

State Management	The element of the Execution Management defining modes of operation for AUTOSAR Adaptive Platform. It allows flexible definition of functions which are active on the platform at any given time.
Supervised Entity	A whole or part of a software component type which is included in the supervision. A Supervised Entity denotes a collection of Checkpoints within the corresponding software component type. A software component type can include zero, one or more Supervised Entities. A Supervised Entity may be instantiated multiple times, in which case each instance is independently supervised. Remark: Safety critical applications and services are considered to be supervised entities, and therefore are expected to be treated as supervised entities within the AUTOSAR Methodology and Architectural Design.
Supervision Mode	State of a machine or Function Group in which Supervised Entity instances are to be monitored with a specific set of configuration parameters. Supervision parameters differ from one mode to other as the behavior (timing or sequence) of Supervised entity changes from one mode to other. Modes are mutually exclusive. A mode can be "Normal", "Degradation".

Table 3.1: Acronyms

4 Requirements Specification

This chapter describes all requirements driving the work to define the `Platform Health Management`.

4.1 Functional Overview

See RS Health Monitoring [3] for the overview of the functionality.

This document specifies the requirements regarding the realization of the `Health Monitoring` on Adaptive Platform. This includes:

- Standardized interfaces
- Mapping of abstract functionalities/concepts defined in Foundation to entities in Adaptive Platform.

EM, PHM and SM are the main safety relevant functional clusters of the AUTOSAR Adaptive Platform. Consequently, their development may require certain processes to be followed - as recommended in ISO26262, for instance [RS_SAF_21101] [4]. A safety argumentation for the AUTOSAR Adaptive Platform, describing functional safety measures and use-cases is provided through Explanation of Safety Overview [5].

4.2 Constraints and assumptions

4.2.1 Limitations

No known limitation.

4.2.2 Applicability to car domains

No restriction.

4.3 Functional Requirements

4.3.1 Supervision functions

[RS_PHM_00101]{DRAFT} **Platform Health Management shall provide a standardized C++ interface for the reporting of Checkpoints.** ⌈

Description:	Platform Health Management shall provide a standardized C++ interface for the reporting of Checkpoints.
Rationale:	Checkpoints are locations inside the code of Supervised Entitys. Platform Health Management checks that these locations are reached in correct time and order. Therefore Platform Health Management needs to be informed when a Checkpoint is reached.
Dependencies:	
Use Case:	Reporting of reached code locations for Alive Supervision, Deadline Supervision and Logical Supervision.
Supporting Material:	

⌋*(RS_Main_00011, RS_Main_00010, RS_Main_00030, RS_Main_00490, RS_Main_00340)*

[RS_PHM_00102]{OBSOLETE} **Platform Health Management shall provide a standardized C++ interface for the reporting of Health Channel.** ⌈

Description:	Platform Health Management shall provide a standardized C++ interface for the reporting of Health Channel.
Rationale:	A Health Channel is a channel for passing external supervision results (e.g. from RAM test, voltage monitoring, ...) to Platform Health Management. Therefore Platform Health Management needs to be informed the status of Health Channels.
Dependencies:	
Use Case:	Reporting of Global Supervision Status, results of test routines or status of (sub)systems (e.g. voltage monitoring, OS kernel, ECU status).
Supporting Material:	

⌋*(RS_Main_00011, RS_Main_00010, RS_Main_00030, RS_Main_00490)*

[RS_PHM_00103]{DRAFT}**Platform Health Management functionality shall be available within the same process and as a separate one.** ⌈

Description:	PHM functionality shall be able to be available, with respect to the monitored process, as: • library component executed in the context of the monitored process • a separate process in the same OS or in the same machine
Rationale:	Provide optimized functionallity for process local usage

▽

△

Dependencies:	
Use Case:	Local monitoring is necessary within the same process for efficiency reasons. Monitoring is also needed in another process for achieving independence. This means the reporting of checkpoints or reporting of health channel status do not cross the boundaries of the OS/VM.
Supporting Material:	

⌋*(RS_Main_00410, RS_Main_00010, RS_Main_00030, RS_Main_00490)*

[RS_PHM_09255]{OBSOLETE} **`Platform Health Management` shall provide an interface to receive Health Channel supervision status** ⌈

Description:	`Platform Health Management` shall check if the health indicators registered by the supervised software indicates a healthy or unhealthy state.
Rationale:	To detect errors like: over-temperature, high bus load, low memory.
Dependencies:	`Platform Health Management` relies on the assumption that the driver/software supervising the parameters such as temperature and memory, translate the numeric values to qualitative values (e.g. low or high), before sending it to the `Platform Health Management`
Use Case:	–
Supporting Material:	–

⌋*(RS_Main_00001, RS_Main_00010, RS_Main_00011, RS_Main_00340)*

[RS_PHM_09257]{OBSOLETE} **`Platform Health Management` shall provide an interface to Supervised Entities to report their health status.** ⌈

Description:	`Platform Health Management` shall provide an interface to `Supervised Entitys` to report their health.
Rationale:	Health Status information can provide useful information on the correct behavior of the system
Dependencies:	–
Use Case:	`Platform Health Management` can verify the Health Status of the `Supervised Entitys` and take the appropriate actions.
Supporting Material:	–

⌋*(RS_Main_00001, RS_Main_00010, RS_Main_00011, RS_Main_00340)*

[RS_PHM_09240]{DRAFT} **Platform Health Management shall support multiple occurrences of the same Supervised Entity.** ⌈

Description:	Platform Health Management shall support multiple occurrences of the same Supervised Entity.
Rationale:	An application or component can be instantiated multiple times
Dependencies:	–
Use Case:	Multiple occurrences of the same software component or application launched multiple times, as separate processes or threads.
Supporting Material:	–

⌋*(RS_Main_00001, RS_Main_00010, RS_Main_00011, RS_Main_00340)*

[RS_PHM_09241]{DRAFT} **Health Monitoring shall support multiple instances of Checkpoints in a Supervised Entity occurrence.** ⌈

Description:	Platform Health Management shall support multiple instances of Checkpoints in a Supervised Entity occurrence, where the number of Checkpoint instances at runtime may be variable.
Rationale:	An application or component containing a Checkpoint can be instantiated multiple times
Dependencies:	–
Use Case:	Parallel/concurrent execution of the same worker threads that execute the same code.
Supporting Material:	–

⌋*(RS_Main_00001, RS_Main_00010, RS_Main_00011, RS_Main_00340)*

[RS_PHM_00111]{DRAFT} **Platform Health Management shall determine Supervision status** ⌈

Description:	Platform Health Management shall determine the Supervision status of Supervisions and Function Groups. i.e. it shall determine the following. • Elementary Supervision Status of Alive, Deadline and Logical Supervisions • Global Supervision Status of whole/part of a Function Group
Rationale:	Global Supervision Status is needed by State Management to trigger recovery action. Global Supervision Status will be an aggregation of Elementary Supervision Status of Supervisions corresponding to processes of a Function Group.
Dependencies:	
Use Case:	Notification based on Global Supervision status to State Management.

▽

△

Supporting Material:	

⌋*(RS_Main_00010)*

[RS_PHM_00112]{DRAFT} **`Platform Health Management` shall provide configurable delays of error reactions.** ⌈

Description:	`Platform Health Management` shall provide configurable delays of error reactions.
Rationale:	Giving the time to the whole software to prepare properly to the upcoming recovery actions, e.g. to the reset.
Dependencies:	
Use Case:	
Supporting Material:	

⌋*(RS_Main_00001, RS_Main_00010, RS_Main_00011, RS_Main_00340)*

[RS_PHM_00115]{DRAFT} **If supervision of State Management fails then Platform Health Management shall trigger a watchdog reset.** ⌈

Description:	
Rationale:	State Management is a fundamental functional cluster of the Adaptive AUTOSAR, if it fails then Platform Health Management (which controls the watchdog) shall trigger a reset which is the only reasonable safety measure
Use Case:	SM is managing a safety critical application. Supervision of SM fails and is detected by PHM. PHM shall trigger a watchdog reset.
Dependencies:	SM
Supporting Material:	

⌋*(RS_SAF_10006, RS_SAF_10030, RS_SAF_10005)*

[RS_PHM_00116]{DRAFT} **If supervision of Execution Management fails then Platform Health Management shall trigger a watchdog reset.** ⌈

Description:	
Rationale:	Execution Management is a fundamental functional cluster of the Adaptive AUTOSAR, if it fails then Platform Health Management (which controls the watchdog) shall trigger a reset which is the only reasonable safety measure
Use Case:	EM is managing safety critical applications and supervision of EM fails and is detected by PHM. PHM shall trigger a watchdog reset.
Dependencies:	EM
Supporting Material:	

⌋*(RS_SAF_10006, RS_SAF_10030, RS_SAF_10005)*

[RS_PHM_00117]{DRAFT} **Platform Health Management shall notify State Management in case an AUTOSAR Adaptive Platform functional cluster, application or service other than Execution Management and State Management fails.** ⌈

Description:	Platform Health Management shall notify State Management in case an AUTOSAR Adaptive Platform functional cluster, application or service other than Execution Management and State Management fails.
Rationale:	Recovery actions are coordinated in SM, the failures shall be reported to SM except if SM or EM themselves fail.
Use Case:	PHM supervises a safety critical application. This application fails. PHM detects the issue and reports to SM.
Dependencies:	–
Supporting Material:	

⌋*(RS_SAF_10005, RS_SAF_10006)*

4.3.2 Mapping of Supervised Entitys to threads and processes

[RS_PHM_00104]{DRAFT} **Platform Health Management shall derive the Supervision Mode from Function Group State(s).** ⌈

Description:	Platform Health Management shall derive the Supervision Mode from Function Group State(s).
Rationale:	Depending on Function Group State, the behavior of process can differ (e.g. other execution path, other timing). Hence, it should be possible to change Supervision configuration based on Function Group State.
Dependencies:	RS_HM_09253
Use Case:	The program flow of a Sensor driver could differ between "Normal mode" and "Sensor Learning mode". Logical Supervision configuration will have to be changed between the corresponding Function Group States.
Supporting Material:	

⌋*(RS_Main_00049, RS_Main_00460)*

[RS_PHM_00105]{DRAFT} **Platform Health Management shall support different allocations/distributions of a Supervised Entity through threads and processes.** ⌈

Description:	Platform Health Management shall support the following Supervised Entities: • A Supervised Entity belonging to one thread • A Supervised Entity spread across several threads of the same process

▽

△

Rationale:	Algorithms can be executed in one thread, multiple threads or processes. It must be possible to supervise a whole algorithm.
Dependencies:	
Use Case:	Supervision of the global flow of algorithms distributed to multiple threads or processes.
Supporting Material:	

⌋*(RS_Main_00410, RS_Main_00460, RS_Main_00010, RS_Main_00030, RS_Main_00490)*

[RS_PHM_00106]{DRAFT} **`Platform Health Management` shall support allocating of multiple `Supervised Entitys` to the same process or thread.** ⌈

Description:	`Platform Health Management` shall support allocating of multiple `Supervised Entitys` to the same process or thread
Rationale:	It shall be possible to define separate `Supervised Entitys` for different supervision functionalities or for subfunctions within the same process or thread
Dependencies:	
Use Case:	Separate `Supervised Entitys` for `Alive Supervision` and `Logical Supervision` of the same thread.
Supporting Material:	

⌋*(RS_Main_00501, RS_Main_00460, RS_Main_00010, RS_Main_00030, RS_Main_00490)*

[RS_PHM_00107]{DRAFT} **`Platform Health Management` shall support multiple instantiation.** ⌈

Description:	`Platform Health Management` shall support: • multiple instantiation of the same executable (resulting with several processes) • multiple instantiation of threads (performing the same action) in an executable • static and dynamic libraries executed in different context • services/servers that can be concurrently invoked by different clients.
Rationale:	The `Health Status` shall be collected and passed between multiple instances by daisy chaining.
Dependencies:	
Use Case:	Collect and validate the `Health Status` reported by the instance(s) on one or multiple microcontroller(s)/cores by another instance running on a separate controller for safety supervisions.
Supporting Material:	

⌋*(RS_Main_00460, RS_Main_00010, RS_Main_00030, RS_Main_00490)*

4.3.3 Daisy chaining

[RS_PHM_00108]{DRAFT} **`Platform Health Management` shall provide a standardized interface between `Platform Health Management` components used in a daisy chain.** ⌈

Description:	`Platform Health Management` shall provide a standardized interface between `Platform Health Management` components used in a daisy chain.
Rationale:	Provide the possibility to use the output of one PHM instance as input to another PHM instance
Dependencies:	
Use Case:	The components are possibly provided by different vendors, working on different microcontrollers or virtual machines. On each controller or (virtual) machine a separate instance of `Platform Health Management` might be used and it should be possible to operate these instances in a daisy chain.
Supporting Material:	

⌋*(RS_Main_00511, RS_Main_00190, RS_Main_00010, RS_Main_00030, RS_Main_00490)*

[RS_PHM_00109]{DRAFT} **`Platform Health Management` shall provide the `Daisy chaining` interface over `ara::com`.** ⌈

Description:	`Platform Health Management` shall provide the `Daisy chaining` interface over at least `ara::com`.
Rationale:	PHM instance shall be able to communicate across microcontrollers or virtual machines
Dependencies:	
Use Case:	The `Platform Health Management` is possibly provided by different vendors, working on different microcontrollers or virtual machines. On each controller or (virtual) machine a separate instance of `Platform Health Management` might be used and it should be possible to operate these instances in a daisy chain. Note: Providing the `ara::com` is mandatory for each implementation `Platform Health Management`, but it is also possible to add more efficient implementations locally.
Supporting Material:	

⌋*(RS_Main_00511, RS_Main_00190, RS_Main_00010, RS_Main_00030, RS_Main_00490)*

4.4 Non-Functional Requirements (Qualities)

[RS_PHM_00001]{DRAFT} **The Platform Health Management shall provide a standardized header file structure for each service.** ⌈

Description:	The Platform Health Management shall provide a standardized header file structure for each service. The application uses the standardized header files which are independent of the underlying implementation.
Rationale:	The application code shall be reusable for different AUTOSAR Adaptive platform implementations.
Dependencies:	–
Use Case:	The application developers implement their code against the standardized header files.
Supporting Material:	–

⌋*(RS_Main_00060, RS_Main_00010, RS_Main_00030, RS_Main_00490)*

[RS_PHM_00002]{DRAFT} **The service header files shall define the namespace for the respective service.** ⌈

Description:	The service header files shall define the namespace for the respective service to uniquely identify each service instance.
Rationale:	The application code shall be reusable for different AUTOSAR Adaptive platform implementations and for different vehicle lines.
Dependencies:	–
Use Case:	To avoid conflicts with other applications and other services, each service shall have its own namespace.
Supporting Material:	–

⌋*(RS_Main_00060, RS_Main_00010, RS_Main_00030, RS_Main_00490)*

[RS_PHM_00003]{DRAFT} **The Platform Health Management shall define how language specific data types are derived from modeled data types.** ⌈

Description:	The Platform Health Management shall define how language specific data types, e.g. C++ data types, are derived from modeled data types.
Rationale:	The Platform Health Management shall support different language bindings.
Dependencies:	–
Use Case:	The Health Management supports C++ language binding and therefore has to define the modeled data types in C++.
Supporting Material:	–

⌋*(RS_Main_00060, RS_Main_00010, RS_Main_00030, RS_Main_00490)*

[RS_PHM_00114]{DRAFT} **Platform Health Management shall be implemented at least according the highest safety integrity level from any process that is supported on the platform.** ⌈

Description:	
Rationale:	Platform Health Management is responsible for ensuring part of the safe execution of safety relevant processes/applications, it should at least be developed with the highest ASIL as the process/application that is being executed.
Use Case:	An ASIL C, B and QM Application is running on the adaptive Platform. PHM shall supervise the ASIL C and B application, therefore PHM shall be implemented with an ASIL C
Dependencies:	–
Supporting Material:	

⌋*(RS_HM_09249)*

5 Requirements Tracing

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

Feature	Description	Satisfied by
[RS_HM_09249]	Health Monitoring shall support building safety-related systems.	[RS_PHM_00114]
[RS_Main_00001]	Real-Time System Software Platform	[RS_PHM_00112] [RS_PHM_09240] [RS_PHM_09241] [RS_PHM_09255] [RS_PHM_09257]
[RS_Main_00002]	AUTOSAR shall provide a software platform for high performance computing platforms	[RS_PHM_NA]
[RS_Main_00010]	Safety Mechanisms	[RS_PHM_00001] [RS_PHM_00002] [RS_PHM_00003] [RS_PHM_00101] [RS_PHM_00102] [RS_PHM_00103] [RS_PHM_00105] [RS_PHM_00106] [RS_PHM_00107] [RS_PHM_00108] [RS_PHM_00109] [RS_PHM_00111] [RS_PHM_00112] [RS_PHM_09240] [RS_PHM_09241] [RS_PHM_09255] [RS_PHM_09257]
[RS_Main_00011]	Mechanisms for Reliable Systems	[RS_PHM_00101] [RS_PHM_00102] [RS_PHM_00112] [RS_PHM_09240] [RS_PHM_09241] [RS_PHM_09255] [RS_PHM_09257]
[RS_Main_00012]	Highly Available Systems Support	[RS_PHM_NA]
[RS_Main_00026]	AUTOSAR shall support high speed and high bandwidth communication between executed SW	[RS_PHM_NA]
[RS_Main_00030]	Safety Related Process Support	[RS_PHM_00001] [RS_PHM_00002] [RS_PHM_00003] [RS_PHM_00101] [RS_PHM_00102] [RS_PHM_00103] [RS_PHM_00105] [RS_PHM_00106] [RS_PHM_00107] [RS_PHM_00108] [RS_PHM_00109]
[RS_Main_00049]	AUTOSAR shall provide an Execution Management for running multiple applications	[RS_PHM_00104]

[RS_Main_00050]	AUTOSAR shall provide an Execution Framework towards applications to implement concurrent application internal control flows	[RS_PHM_NA]
[RS_Main_00060]	Standardized Application Communication Interface	[RS_PHM_00001] [RS_PHM_00002] [RS_PHM_00003]
[RS_Main_00080]	Formal Description Language	[RS_PHM_NA]
[RS_Main_00106]	AUTOSAR shall provide the possibility to extend the software with new SWCs without recompiling the platform foundation	[RS_PHM_NA]
[RS_Main_00150]	AUTOSAR shall support the deployment and reallocation of AUTOSAR Application Software	[RS_PHM_NA]
[RS_Main_00170]	AUTOSAR shall provide secure access to ECU data and services	[RS_PHM_NA]
[RS_Main_00180]	Intellectual Property Protection	[RS_PHM_NA]
[RS_Main_00190]	Non-AUTOSAR Software Integration	[RS_PHM_00108] [RS_PHM_00109]
[RS_Main_00230]	Network Technology Support	[RS_PHM_NA]
[RS_Main_00250]	AUTOSAR methodology shall provide a predefinition of typical roles and activities	[RS_PHM_NA]
[RS_Main_00260]	Runtime Diagnostics Means	[RS_PHM_NA]
[RS_Main_00261]	AUTOSAR shall provide means for calibration	[RS_PHM_NA]
[RS_Main_00270]	Backward Compatibility	[RS_PHM_NA]
[RS_Main_00280]	Standardized Automotive Communication Protocols	[RS_PHM_NA]
[RS_Main_00300]	AUTOSAR shall provide data exchange formats to support work-share in large inter and intra company development groups	[RS_PHM_NA]
[RS_Main_00301]	AUTOSAR shall specify profiles for data exchange to support work-share in large inter- and intra-company development groups	[RS_PHM_NA]
[RS_Main_00310]	AUTOSAR shall support hierarchical Application Software design methods	[RS_PHM_NA]
[RS_Main_00320]	AUTOSAR shall provide formats to specify system development	[RS_PHM_NA]
[RS_Main_00340]	AUTOSAR shall support the continuous timing requirement analysis	[RS_PHM_00101] [RS_PHM_00112] [RS_PHM_09240] [RS_PHM_09241] [RS_PHM_09255] [RS_PHM_09257]
[RS_Main_00350]	Documented Software Architecture	[RS_PHM_NA]
[RS_Main_00360]	Variant Management Support	[RS_PHM_NA]
[RS_Main_00410]	AUTOSAR shall provide specifications for routines commonly used by Application Software to support sharing and optimization	[RS_PHM_00103] [RS_PHM_00105]
[RS_Main_00420]	AUTOSAR shall use established software standards and consolidate de-facto standards for basic software functionality	[RS_PHM_NA]
[RS_Main_00440]	AUTOSAR shall standardize access to non-volatile memory	[RS_PHM_NA]
[RS_Main_00445]	AUTOSAR shall standardize access to crypto-specific HW and SW	[RS_PHM_NA]

[RS_Main_00460]	AUTOSAR shall standardize methods to organize mode management on Application, ECU and System level	[RS_PHM_00104] [RS_PHM_00105] [RS_PHM_00106] [RS_PHM_00107]
[RS_Main_00490]	AUTOSAR processes shall be compliant to ISO26262	[RS_PHM_00001] [RS_PHM_00002] [RS_PHM_00003] [RS_PHM_00101] [RS_PHM_00102] [RS_PHM_00103] [RS_PHM_00105] [RS_PHM_00106] [RS_PHM_00107] [RS_PHM_00108] [RS_PHM_00109]
[RS_Main_00491]	Function Monitoring	[RS_PHM_NA]
[RS_Main_00500]	AUTOSAR shall provide naming conventions	[RS_PHM_NA]
[RS_Main_00501]	AUTOSAR shall support redundancy concepts	[RS_PHM_00106]
[RS_Main_00503]	AUTOSAR shall support change of communication and application software at runtime.	[RS_PHM_NA]
[RS_Main_00507]	Development Collaboration Support	[RS_PHM_NA]
[RS_Main_00510]	Secure Onboard Communication	[RS_PHM_NA]
[RS_Main_00511]	AUTOSAR shall support virtualization	[RS_PHM_00108] [RS_PHM_00109]
[RS_Main_00512]	AUTOSAR shall support time synchronization	[RS_PHM_NA]
[RS_Main_00514]	System Security Support	[RS_PHM_NA]
[RS_Main_00650]	AUTOSAR shall support up - and download of data and software	[RS_PHM_NA]
[RS_Main_00653]	Means for Functional Modeling	[RS_PHM_NA]
[RS_Main_01001]	Intra ECU Communication Support	[RS_PHM_NA]
[RS_Main_01002]	AUTOSAR shall support service-oriented communication	[RS_PHM_NA]
[RS_Main_01003]	AUTOSAR shall support data-oriented communication	[RS_PHM_NA]
[RS_Main_01004]	AUTOSAR shall support standards for wireless off-board communication	[RS_PHM_NA]
[RS_Main_01005]	AUTOSAR shall establish communication paths dynamically	[RS_PHM_NA]
[RS_Main_01007]	AUTOSAR communication shall assure quality of service on communication	[RS_PHM_NA]
[RS_Main_01008]	AUTOSAR shall provide secure communication with off-board entities	[RS_PHM_NA]
[RS_Main_01025]	AUTOSAR shall support debugging of software on the target and onboard	[RS_PHM_NA]
[RS_SAF_10005]	AUTOSAR shall provide mechanisms to support safe shutdown and termination of applications, software components, basic software modules and services.	[RS_PHM_00115] [RS_PHM_00116] [RS_PHM_00117]
[RS_SAF_10006]	AUTOSAR shall provide mechanisms to support safe transition of states in a basic software module, software component, application or service life cycle.	[RS_PHM_00115] [RS_PHM_00116] [RS_PHM_00117]
[RS_SAF_10030]	AUTOSAR shall provide mechanisms to support safe program execution.	[RS_PHM_00115] [RS_PHM_00116]

5.1 Not applicable requirements

[RS_PHM_NA]{DRAFT} ⌈These requirements are not applicable as they are not within the scope of this release.⌋*(RS_Main_00002, RS_Main_00012, RS_Main_00026, RS_Main_00050, RS_Main_00080, RS_Main_00106, RS_Main_00150, RS_Main_00170, RS_Main_00180, RS_Main_00230, RS_Main_00250, RS_Main_00260, RS_Main_00261, RS_Main_00270, RS_Main_00280, RS_Main_00300, RS_Main_00301, RS_Main_00310, RS_Main_00320, RS_Main_00350, RS_Main_00360, RS_Main_00420, RS_Main_00440, RS_Main_00445, RS_Main_00491, RS_Main_00500, RS_Main_00503, RS_Main_00507, RS_Main_00510, RS_Main_00512, RS_Main_00514, RS_Main_00650, RS_Main_00653, RS_Main_01001, RS_Main_01002, RS_Main_01003, RS_Main_01004, RS_Main_01005, RS_Main_01007, RS_Main_01008, RS_Main_01025)*

6 References

[1] Standardization Template
AUTOSAR_TPS_StandardizationTemplate

[2] Glossary
AUTOSAR_TR_Glossary

[3] Requirements on Health Monitoring
AUTOSAR_RS_HealthMonitoring

[4] Safety Requirements for AUTOSAR Adaptive Platform and AUTOSAR Classic Platform
AUTOSAR_RS_Safety

[5] Explanation of Safety Overview
AUTOSAR_EXP_SafetyOverview