

<b>Document Title</b>	Adaptive Platform Release Overview
<b>Document Owner</b>	AUTOSAR
<b>Document Responsibility</b>	AUTOSAR
<b>Document Identification No</b>	782

<b>Document Status</b>	Final
<b>Part of AUTOSAR Standard</b>	Adaptive Platform
<b>Part of Standard Release</b>	17-03
<b>Release Life Cycle Status</b>	Initial Release

<b>Document Change History</b>			
<b>Date</b>	<b>Release</b>	<b>Changed by</b>	<b>Change Description</b>
2017-03-31	17-03	AUTOSAR Release Management	Initial release

## Table of Contents

1	Introduction.....	3
1.1	Scope of this document.....	3
1.2	Dependencies to other standards.....	3
1.3	Content of chapters.....	3
2	Introduction to the Adaptive Platform.....	4
2.1	Release strategy.....	4
2.2	Parallel validation of specification via implementation.....	5
2.3	Specification depth.....	6
3	Related documentation.....	7
4	Summary of changes.....	8
5	Specification overview.....	9
6	Remarks to known technical deficiencies.....	11
7	Release history.....	13
7.1	Release 17-03.....	13
8	Appendix.....	14
8.1	Definitions.....	14
8.1.1	Release number.....	14
8.1.2	Specification item and requirement life cycle states.....	14

# 1 Introduction

## 1.1 Scope of this document

This document provides an overview on the AUTOSAR standard “Adaptive Platform” release “17-03”.

## 1.2 Dependencies to other standards

This release of the Adaptive Platform depends on the standard “Foundation” in Release 1.1.0, which

- defines protocols implemented by Adaptive Platform and
- contains the project objectives and the main requirements from which the features of the Adaptive Platform are derived.

These dependencies are refined in the trace information of the requirements in the respective specifications.

## 1.3 Content of chapters

This document is structured as follows:

- Chapter 2 provides an introduction to the Adaptive Platform and its standardization approach.
- Chapter 3 provides a list of documentation references.
- Chapter 4 provides a summary of changes since the previous release of the Adaptive Platform
- Chapter 5 contains the overview of specifications comprising the Release 17-03. This chapter is structured according to the clusters of AUTOSAR Release 17-03.
- Chapter 6 contains remarks about known technical deficiencies.
- Chapter 7 contains the detailed release history of all released specifications.
- Chapter 8 provides a set of definitions aimed to increase the understanding of the content of this document and the Release 17-03.

## 2 Introduction to the Adaptive Platform

The AUTOSAR Adaptive Platform is the standardized platform for microprocessor-based ECUs supporting use cases like highly automated and autonomous driving as well as high speed on-board and off-board communication.

The Adaptive Platform differs in a number of aspects from the standardization approach of the Classic Platform:

- Release strategy
- Parallel validation of specification via software implementation
- Specification of functional clusters instead of modules

### 2.1 Release strategy

The Adaptive Platform will be in “Development” mode until October 2018 according to AUTOSAR’s lifecycle model for its standards (see figures 1 and 2). During this time, AUTOSAR will release a new version of the Adaptive Platform in a 6-month cycle. The lifecycle state “Development” implies that users of the Adaptive Platform have no guarantee on backward compatibility. Consequently, all requirements have the lifecycle status *draft*. For items in specifications this attribute is not explicitly set but is the default state.

According to current planning the October 2018 release will end the first major development lifecycle. This release will no longer contain any draft specifications. Additionally this release will be fully synchronized with Classic Platform R4.4.0.

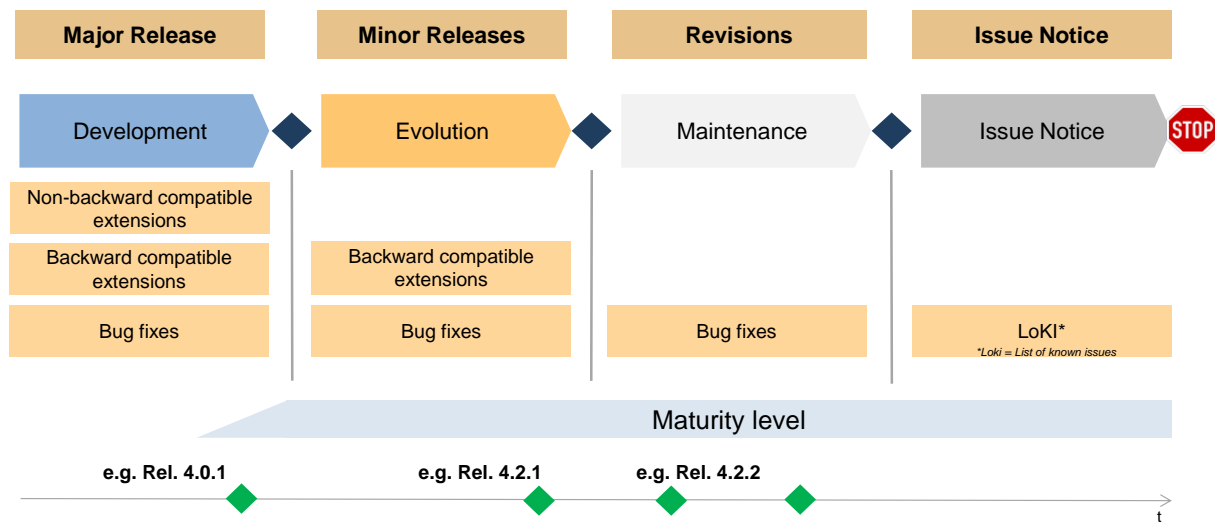
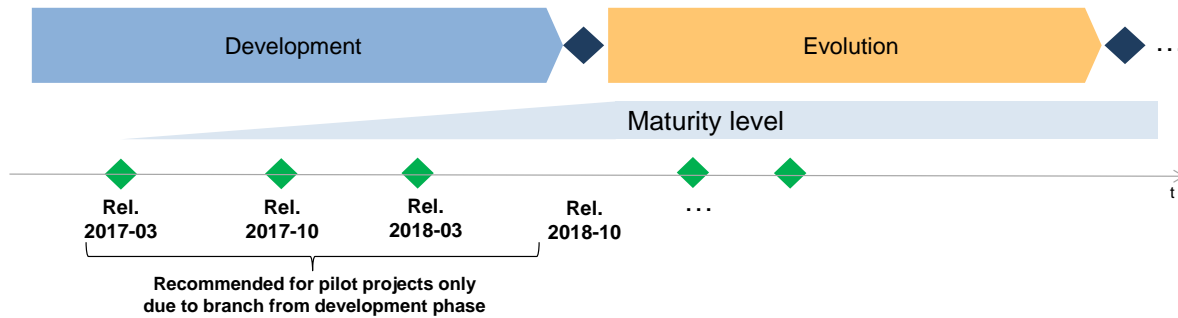


Figure 1: Lifecycle model of AUTOSAR standards and its application to Classic Platform



**Figure 2: Application of AUTOSAR lifecycle to Adaptive Platform**

Apart from the regular specifications that have been elaborated in intensive expert discussion, according to current planning, the releases until March 2018 may comprise draft specifications to indicate the intended scope and direction of discussion to the AUTOSAR development community.

The following must be considered for the draft specifications:

1. Minimal or less quality measures are being applied during development
2. No indication of quality / stability due to a lack of discussions between the AUTOSAR partners

As new features will be added with each release, AUTOSAR encourages the partners to contribute in the development of the features.

## 2.2 Parallel validation of specification via implementation

The Adaptive Platform is validated through an AUTOSAR-internal implementation: the Adaptive Platform Demonstrator. This Demonstrator is available to all the partners and can be a reference to understand the underlying concepts of the Adaptive Platform. All further development based on the Demonstrator will become the responsibility of the respective partner (for legal constraints see the dedicated paragraphs in the Development Agreement).

For the first release 17-03, the Demonstrator software implementation has undergone only informal reviews with no strict quality assurance. AUTOSAR plans to increase the quality assurance significantly to ensure the maintainability and extensibility of the Demonstrator software implementation.

The Demonstrator comes with traceability back to the specifications and explanatory documents or the so-called Functional Cluster Design specifications.

The Adaptive Platform source code is frozen but still needs clearance regarding obeying the licenses of the Open Source Software being part of the Adaptive Platform source code. We anticipate the Adaptive Platform software implementation being released latest end of April 2017. The frozen source code will undergo an Open Source Scanning and may be subject to change due to potential findings of the Open Source Scanning.

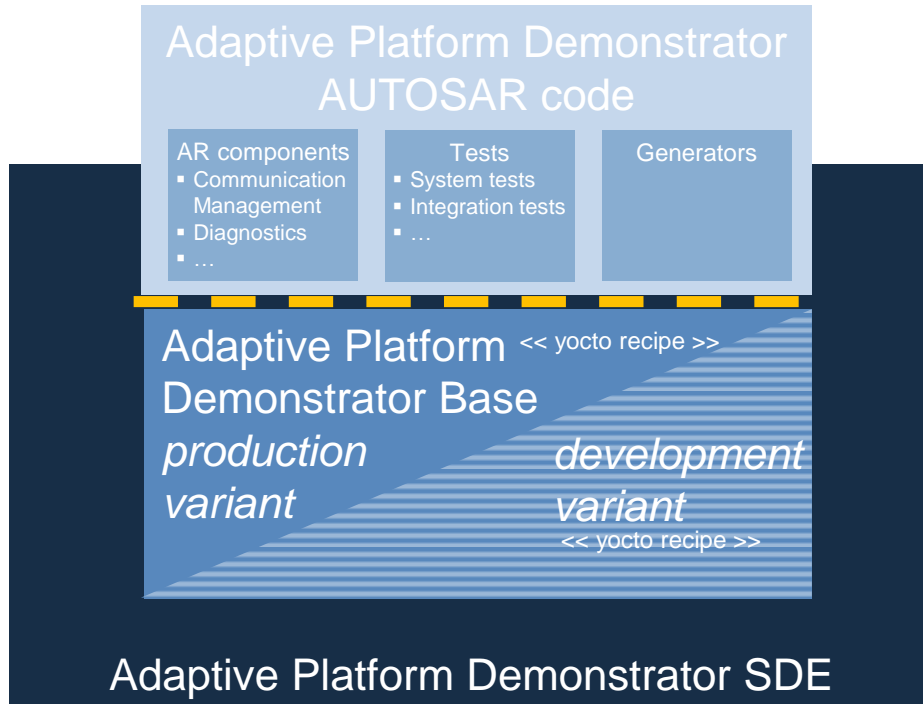


Figure 3: Overview of the AUTOSAR Adaptive Platform Demonstrator

### 2.3 Specification depth

Based on the development history of the Classic Platform, AUTOSAR has decided to specify functional clusters instead of a specific software architecture to provide the implementers with options to find efficient solutions for the standardized features.

### 3 Related documentation

- [1] AUTOSAR Specifications in general
- [2] Change Documentation
- [3] Glossary

## 4 Summary of changes

This chapter contains a summary of changes which were implemented since the previous Release.

*This is the introductory release of all documents in the Adaptive Platform standard.*



## 5 Specification overview

The published specifications are divided into the following clusters:

- Release Documentation
- General
- Methodology and Manifests
- Adaptive Foundation
- Adaptive Services

These clusters are further structured into subcategories to provide better guidance to the specification users. The assignment of specifications to clusters is shown below.

Long Name	File Name	Life cycle changes	Draft Specification
<b>Cluster: Release Documentation</b>			
Adaptive Platform Release Overview	AUTOSAR_TR_AdaptivePlatformReleaseOverview	Initial release	
AUTOSAR Adaptive Platform Specification Hashes	AUTOSAR_TR_AdaptivePlatformSpecificationHashes	Initial release	
<b>Cluster: General</b>			
Platform Design for Adaptive Platform	AUTOSAR_EXP_PlatformDesign	Initial release	
General Requirements specific to Adaptive Platform	AUTOSAR_RS_General	Initial release	
General Specification of Adaptive Platform	AUTOSAR_SWS_General	Initial release	
Guidelines for the use of the C++14 language in critical and safety-related systems	AUTOSAR_RS_CPP14Guidelines	Initial release	x
Functional Cluster Shortnames	AUTOSAR_TR_FunctionalClusterShortnames	Initial release	
<b>Cluster: Methodology and Manifests</b>			
Requirements on Manifest Specifications	AUTOSAR_RS_ManifestSpecification	Initial release	
Requirements on Manifest Specifications	AUTOSAR_TPS_ManifestSpecification	Initial release	
Methodology for Adaptive Platform	AUTOSAR_TR_AdaptiveMethodology	Initial release	x
Meta Model	AUTOSAR_MMOD_MetaModel	Initial release	
Meta Model-generated XML Schema	AUTOSAR_MMOD_XMLSchema	Initial release	
Supplementary material of the AUTOSAR XML Schema	AUTOSAR_TR_XMLSchemaSupplement	Initial release	
<b>Cluster: Adaptive Foundation</b>			
Requirements on Communication Management	AUTOSAR_RS_CommunicationManagement	Initial release	
Specification of Communication Management	AUTOSAR_SWS_CommunicationManagement	Initial release	
Requirements on Operating System Interface	AUTOSAR_RS_OperatingSystemInterface	Initial release	
Specification of Operating System Interface	AUTOSAR_SWS_OperatingSystemInterface	Initial release	
Requirements on Execution	AUTOSAR_RS_ExecutionManagement	Initial release	

Long Name	File Name	Life cycle changes	Draft Specification
Manager	ement		
Specification of Execution Management	AUTOSAR_SWS_ExecutionManagement	Initial release	
Explanation of ara::com API	AUTOSAR_EXP_ARAComAPI	Initial release	
Requirements on E2E for Adaptive Platform	AUTOSAR_RS_AdaptiveE2E	Initial release	x
Requirements on Watchdog for Adaptive Platform	AUTOSAR_RS_AdaptiveWatchdog	Initial release	x
Specification of LogAndTrace for Adaptive Platform	AUTOSAR_SWS_AdaptiveLogAndTrace	Initial release	x
Requirements on Persistency	AUTOSAR_RS_Persistency	Initial release	x
Specification of Persistency	AUTOSAR_SWS_Persistency	Initial release	x
Requirements on Log and Trace for Adaptive Platform	AUTOSAR_RS_AdaptiveLogAndTrace	Initial release	x
<b>Cluster: Adaptive Services</b>			
Specification of Diagnostics for Adaptive Platform	AUTOSAR_SWS_AdaptiveDiagnostics	Initial release	x

All specifications contain requirements which are identified by special braces:

[ = Beginning of the requirement content

] = End of the requirement content

In addition, all XML files and schemas are considered as requirements.

## 6 Remarks to known technical deficiencies

The technical deficiencies per specification are – if applicable – mentioned inside the respective specification in a chapter called “Known Limitations” which is located after the table of contents.

There are the following technical deficiencies which are not related to a particular specification:

Document Long Name	Technical Deficiencies
Guidelines for the use of the C++14 language in critical and safety-related systems	<ul style="list-style-type: none"> <li>• The rule set for parallel computing is not provided</li> <li>• The rule set for C++ standard libraries is only initial (incomplete)</li> <li>• The rule set for security (as long as it is not common to critical software or safety-related software) is not provided</li> <li>• The traceability to JSF, ISO CPP contains some non-analyzed rules</li> <li>• The traceability to ISO 26262 is not provided</li> </ul>
Specification of Communication Management	<p>The following features are not yet supported:</p> <ul style="list-style-type: none"> <li>• Attributes (SOME/IP Fields)</li> <li>• ApplicationErrors of methods</li> <li>• ExtendedApplicationErrors of methods as exceptions</li> <li>• LocalBufferOverrun</li> <li>• SubscriptionState</li> </ul>
Specification of Execution Management	<ul style="list-style-type: none"> <li>• Application authentication and integrity validation (RS_EM_00003, RS_EM_00004)</li> <li>• Resource management, e.g. budgets (RS_EM_00005, RS_EM_00006, RS_EM_00007)</li> <li>• Application core binding is only minimally described and will need further elaboration (RS_EM_00008)</li> <li>• Interaction with the system as a whole (system wide coordination, external triggers of applications and support – wrappers - for configuring cyclic events) are not fully considered. There is some support, e.g. machine state management and POSIX support for cyclic events but not such that RS_EM_00050/RS_EM_00051/RS_EM_00052 can be fully considered fulfilled.</li> </ul>
Specification of Diagnostics for Adaptive Platform	<ul style="list-style-type: none"> <li>• Diagnostics Manager Model lacks specification depth</li> <li>• Some Diagnostic Services, DoIP messages and common fault memory features not supported</li> </ul>
Specification of LogAndTrace for Adaptive Platform	<ul style="list-style-type: none"> <li>• The provided C++ API is designed to be independent from the underlying Logging protocol back-end implementation</li> </ul>
Specification of Operating System Interface	<ul style="list-style-type: none"> <li>• Budgets (resource management) is not described (RS_OSI_00201, RS_OSI_00202)</li> <li>• Core binding is not supported (RS_OSI_00203)</li> <li>• Access control (authenticated access to OS objects) is not supported (RS_OSI_00204)</li> </ul>

Document Long Name	Technical Deficiencies
Specification of Persistency	<ul style="list-style-type: none"> <li>• Specification is currently very close to a particular implementation, users can expect that the level of abstraction will increase</li> <li>• APIs are not modeled and are expected to be formulated in a more abstract form in upcoming releases</li> <li>• The persistency API is not able to handle concurrent access to one persistent storage location. Data can only be shared between multiple applications in read only mode where no applications writes to that persistent storage location.</li> </ul>
Methodology for Adaptive Platform	<ul style="list-style-type: none"> <li>• No Diagnostics use case is described in TR_AdaptiveMethodology</li> <li>• No roles and detailed task modelling is included in TR_AdaptiveMethodology</li> <li>• RS_Methodology in FO only refers to TR_AdaptiveMethodology (in AP), not to TR_Methodology (in CP). References to CP will be fixed with next CP release.</li> </ul>

Adaptive Platform R17-03 does not cover Network Management. Two approaches (UDP- / SOME/IP-based) are under discussion. These will be evaluated for the next release 17-10.

## 7 Release history

### 7.1 Release 17-03

Release 17-03 was originally released on the 31<sup>th</sup> of March 2017.

*This is the introductory release of all documents in the Adaptive Platform standard.*

## 8 Appendix

### 8.1 Definitions

See [3] for AUTOSAR definitions.

#### 8.1.1 Release number

AUTOSAR applies a four-digit numbering scheme Ryy-mm to identify Releases.

- yy = year
- mm = month

#### 8.1.2 Specification item and requirement life cycle states

**Specification items:** The life cycle state is after the specification item ID surrounded by curly brackets. The states are:

- **Valid:** This indicates that the related entity is a valid part of the document. This is the default.
- **Draft:** This indicates that the related entity is newly introduced but still experimental. This information is published but is subject to change without backward compatibility guarantee.
- **Obsolete:** This indicates that the related entity is obsolete and will be removed in the next release.

If there is no life cycle state information stated then the state is Valid.

**Requirements:** The requirement attribute „type“ indicates the life cycle state of the requirement. The states are the same as the specification item states.