Document Title	AUTOSAR BSW & RTE Conformance Test Specification Part 2: Process Overview
Document Owner	AUTOSAR GbR
Document Responsibility	AUTOSAR GbR
Document Identification No	282
Document Classification	Auxiliary

Document Version	1.0.1
Document Status	Final
Part of Release	3.1
Revision	0001

Document Change History

Date	Version	Changed by	Change Description
23.06.2008	1.0.1	AUTOSAR Administration	Legal disclaimer revised
14.11.2007	1.0.0	AUTOSAR Administration	Initial Release

Page left intentionally blank

Advice to users of AUTOSAR Specification Documents:

AUTOSAR Specification Documents 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 Specification Documents for illustration purposes only, and they themselves are not part of the AUTOSAR Standard. Neither their presence in such Specification Documents, 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 Contents

1 Overview of the document

This document is a generic process description on the creation of conformance test specifications for AUTOSAR BSW modules: the “CTSpec creation process”.
It first introduces and describes the parts that play a role within AUTOSAR conformance testing and defines associated terminology. Furthermore, the specific circumstances with which the AUTOSAR conformance testing methodology has to deal with are worked out.
Based on this basic understanding, the CTSpec creation process is then defined by means of role and activity descriptions together with a list of artifacts that are input or output to the roles and activities. Note that these descriptions are not intended to give the details required for the realization of the activities and creation of the work artifacts. These details are specified in [2].

1.1 Focus and scope

The creation methodology for conformance test specifications described in this document is based on the results of the conformance test pilot project performed for ICC3. Therefore, the primary scope of this document is set on conformance testing BSW modules on ICC3 level.
However, the major concepts described here are generic enough to apply to conformance tests on ICC2 and ICC1 level as well but certain adaptations are probably necessary. They might be added in future versions of this document.

1.2 How to use the document

This document is used to gain an overall understanding and overview of the CTSpec creation process.
The basic background of AUTOSAR conformance testing [1] is helpful to understand the rationale behind certain decisions and definitions but is not mandatory to put the CTSpec creation process into realization.
Therefore, for realizing the CTSpec creation process, this document should be the first or second one (after [1]) to be read, i.e. before the documents [2] and [3] which assume understanding of the basics established in this document.

1.3 Abbreviations

Abbreviation	***Description***
API	Application Program Interface
BSW	Basic Software
CC	Conformance Class
CD	Coder/Decoder (TTCN-3 – see Part 4)
CH	Component Handling (TTCN-3 – see Part 4)
CTA	Conformance Test Agency
CTSpec	Conformance Test Specification
CTS	Conformance Test Suite

Abbreviation	Description
ECU	Electronic Control Unit
FCC	Functional Conformance Class
ICC	Implementation Cluster Conformance Class
ICS	Implementation Conformance Statement
IP	Intellectual Property
PA	Platform Adapter (TTCN-3 – see Part 4)
PS	Product Supplier
RTE	Run Time Environment
SA	System Adapter (TTCN-3 – see Part 4)
SUT	System Under Test
SW-C	Software Component
SWS	Software Specification
TE	TTCN-3 Executable (TTCN-3 – see Part 4)
TM	Test Manager (TTCN-3 – see Part 4)
TRI	TTCN-3 Runtime Interface (TTCN-3 – see Part 4)
TTCN-3	Testing and Test Control Notation, version 3

1.4 Bibliography

[1] AUTOSAR BSW & RTE Conformance Test Specification Part 1: Background
https:/svn2.autosar.org/repos2/22_Releases
AUTOSAR_CTSpec_Background

[2] AUTOSAR GbR. AUTOSAR BSW & RTE Conformance Test Specification Part 3: Creation & Validation
https:/svn2.autosar.org/repos2/22_Releases
AUTOSAR_CTSpec_Creation_Validation

[3] AUTOSAR GbR. AUTOSAR BSW & RTE Conformance Test Specification Part 4: Execution Constraints
https:/svn2.autosar.org/repos2/22_Releases
AUTOSAR_CTSpec_Execution_Constraint

[4] AUTOSAR Conformance Test Specification Template
https:/svn2.autosar.org/repos2/22_Releases
AUTOSAR_CTSpec_Template

[5] Specification of ECU Configuration Parameters
https:/svn2.autosar.org/repos2/22_Releases
AUTOSAR_EcucParamDef.xml

[6] AUTOSAR Layered Software Architecture
https:/svn2.autosar.org/repos2/22_Releases
AUTOSAR_LayeredSoftwareArchitecture

[7] Specification of Platform Types
https:/svn2.autosar.org/repos2/22_Releases

AUTOSAR_SWS_PlatformTypes

[8] “Specification of Standard Types
https:/svn2.autosar.org/repos2/22_Releases
AUTOSAR_SWS_StandardTypes.pdf

[9] Specification of Communication Stack Types
https:/svn2.autosar.org/repos2/22_Releases
AUTOSAR_SWS_ComStackTypes.pdf

[10] Specification of Memory Mapping
https:/svn2.autosar.org/repos2/22_Releases
AUTOSAR_SWS_MemoryMapping.pdf

[11] AUTOSAR GbR. Specification of C Implementation Rules,
https:/svn2.autosar.org/repos2/22_Releases
AUTOSAR_SWS_C_ImplementationRules

[12] Specification of ECU Configuration
https:/svn2.autosar.org/repos2/22_Releases
AUTOSAR_ECU_Configuration.pdf

[13] Methodology.
https:/svn2.autosar.org/repos2/22_Releases
AUTOSAR_Methodology.pdf

2 Introduction and Overview

2.1 Overall goal of the CTSpec creation process

The overall goal of the CTSpec creation process is to provide conformance test cases for all BSW-Modules. The scope of AUTOSAR conformance tests is given in [1].

2.2 Input and output of the CTSpec creation process

The CTSpec (Conformance Test Specification) creation process defined in this specification inputs the relevant AUTOSAR specifications (defined in Section 2.3) and outputs the "Conformance Test Specification" (defined in Section 2.4), also called "CTSpec" for a given BSW module. As a side-effect, the CTSpec Creation Process may lead to improvements in the relevant AUTOSAR specifications.

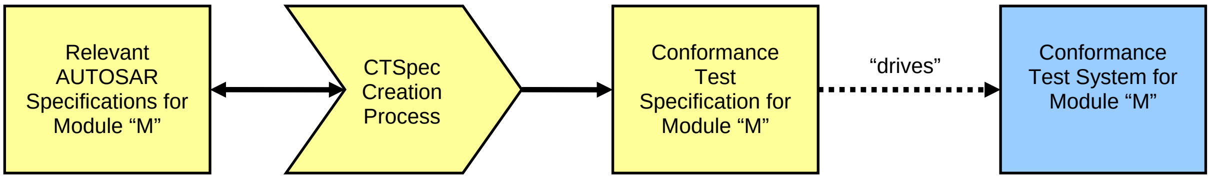

Figure 1 – The CTSpec Creation Process

2.3 Relevant AUTOSAR specifications

The requirements that must be covered by the conformance tests are distributed over a range of AUTOSAR specifications. This section gives an overview of the AUTOSAR specifications that must be considered when designing the CTSpec for a given module *M*.

1. The most important document to be considered is the Software Specification (SWS) of the module *M*. This document contains specification items on all aspects of the BSW module (such as source code structure, implementation approach, provided and required interfaces, functional behavior, configuration parameters). These specification items are mainly informal verbal descriptions but can also include more formal definitions such as finite state machines or sequence diagrams.
2. The SWS of neighbouring modules must also be considered. In case module *M* uses services provided by a neighbouring module, the requirements on the correct usage of the neighbouring module (which must be satisfied by *M* and are the object of the conformance tests of *M*) are typically only defined in the SWS of the neighbouring module and not in the SWS of *M* itself.

3. The conformance tests must consider the formal definition of the configuration parameters as specified in [5][1] and the associated formal XML-Schemas.
4. The conformance tests can make use of the models that underlie the aforementioned specifications, namely:
   a. The UML-model of the ECU configuration parameters
   b. The UML-model of the BSW modules
5. The conformance tests must take into account a substantial amount of background information which is scattered over the following documents (this list may not be exhaustive):
   a. The “Layered Software Architecture” [5], containing general concepts of the AUTOSAR BSW architecture
   b. The “Specification of Platform Types” [7], the “Specification of Standard Types” [8] and the “Specification of Communication Stack Types” [9] containing type definitions used in various SWSs of BSW modules
   c. Elements in the “Specification of Memory Mapping” [10] and the “Specification of C Implementation Rules” [11] define the details of the “C”-API’s of the BSW modules.
   d. The “Specification of ECU Configuration” [12] defining general concepts related to the configuration of BSW modules
   e. The “Methodology” [13] contains a detailed definition of the method used to configure and build BSW modules.

The conformance tests should on the other hand **not** concern themselves with the documents containing requirements that led to the definition of the documents mentioned above. This category of documents includes for example the Software Requirements Specifications (SRS) of the individual modules. In addition, many SWSs also contain requirements that are more concrete than the SRS of the module but that are refined and superseded by other, more concrete, requirements in the SWS. These requirements, which sit in between the SRS and the concrete requirements in the SWS, are irrelevant for conformance testing and are categorized as such within the CTSpec definition process.

2.4 Conformance Test System

When describing the parts that play a role in functional testing, it usually starts with the distinction between the object that is to be tested (i.e. the “system-under-test”) and the object that performs the test (i.e. the “tester”).
For AUTOSAR conformance testing, this distinction is not sufficient due to the importance and relevance of the whole configuration and build process for BSW modules.
Additionally to “system-under-test”, and “tester”, a third part has therefore to be introduced: the “configuration and build process”.
The “conformance test system” denotes the aggregation of the three aforementioned parts containing the whole conglomerate of source files, data files, configuration files, generation/transformation tools, executable files etc. that are part of and relevant to the conformance testing process by being input, output, intermediate or supporting artifacts.

[1] Note that the relevant contents of this document might move inside the SWS of the module in later versions of AUTOSAR

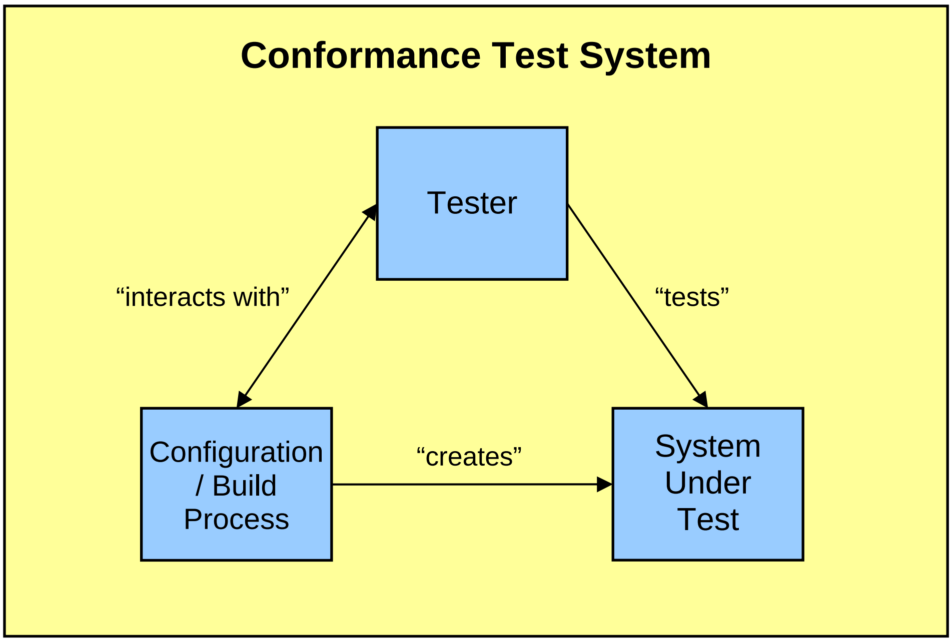

Figure 2- Overview of the "conformance test system"

Please note that in Figure 2, the relation between “tester” and “configuration/build process” is described very generally as “interacts with”. This is due to the amount of freedom left open by the AUTOSAR standards to be used by the vendor to design these processes according to their business and technology requirements and goals.
In the simplest case, the “tester” only inputs configuration data in standardized AUTOSAR XML formats into the configuration/build process and this configuration data can be directly used to build the configured BSW module (i.e. the system-under-test).
In more difficult cases, the configuration/build process might in turn pose requirements and/or constraints on the “tester”. For example, certain configuration values might not be allowed due to limitations of the BSW module implementation or the concept of memory configuration and allocation of the BSW module implementation needs to be taken into account in the “tester”.
Furthermore, some aspects of the configuration/build process are themselves subject to conformance testing as well (e.g. provision and correctness of header files that are output of the configuration/build process and to be used by other BSW modules). However, the CTSpecs do not specify automated test cases (in TTCN-3) for testing these aspects of the configuration/build process but describe verbally the checks that need to be performed manually or semi-automatically[2].

[2] The reason for describing “test cases” on the configuration/build process only verbally is the great diversity of possible realizations of configuration/build processes and the relatively simple checks to be performed within these “test cases”.

2.4.1 Salient aspects of AUTOSAR with respect to the “system-under-test”

The definition of the “system-under-test” within the AUTOSAR conformance test system is highly complicated by the following characteristics of the AUTOSAR standard:

- AUTOSAR BSW modules are highly configurable.
- AUTOSAR does not only define how a module should behave for a specific configuration but also how a module can be configured using standardized configuration formats.
- Therefore, a vendor supplying an AUTOSAR module generally ships a tool that is used in the configuration and build process to generate or configure the module from an AUTOSAR-compliant definition of the module’s configuration.
- In certain cases, this configuration is possible at different steps in the build process of the module. In the case of so-called “pre-compile configuration”, the module’s implementation can be generated out of its configuration. In the case of “post-build configuration” certain aspects of the module’s behavior can be changed after the module has been built.
- A vendor has a lot of flexibility in shipping the BSW module as a mixture of source-code and object-code. As mentioned before, the module’s source can be configuration-specific and can be generated out of the configuration by a vendor-provided tool.
- In addition to the configuration parameters standardized by AUTOSAR, a module can have additional “vendor-specific” configuration parameters.
- Conformance should be ascertained against an “Implementation Conformance Statement”: the ICS represents the product supplier’s statement on the implemented parameter ranges of a BSW module and consequently determines what can be tested for the concrete implementation[3].
- In certain cases, AUTOSAR does not include a complete definition of the behavior of the module. This is especially the case for all BSW modules that interface directly with hardware components (i.e. hardware drivers): large aspects of the behavior of such modules are hardware-specific and are not defined in the module’s SWS. As a consequence, the module will not work (and cannot be tested for conformance) without the presence of the specific hardware for which it has been designed.

2.4.2 Overview of the conformance test system

When creating and executing AUTOSAR BSW conformance tests, there are strong interdependencies between the parts of the conformance test system (i.e. tester, system-under-test and its configuration/build process) that must be taken into account. While the CTSpec creation process specifies the tester (i.e. the test cases and the test execution environment) and the configuration of the system-under-test, the module vendor has to implement the configuration/build process for the BSW module under test.
The following is provided by the vendor as part of the configuration/build process:

[3] The flexibility of the ICS is not entirely fixed at the time of writing and is still under discussion. The details of the ICS might have a substantial impact on the CTSpec execution.

1. An Implementation Conformance Statement (ICS): this is a formal description (in XML) of the extent to which the module is supposed to cover the AUTOSAR specifications.
2. A BSW Module Description: this is a formal description (in XML) of the BSW module to be tested: it is needed as it contains information that is left open in the AUTOSAR specifications themselves (such as the definition of data types left open by the AUTOSAR specification or the extension of the AUTOSAR standardized configuration parameters with vendor-specific configuration parameters).
3. Values for vendor-specific configuration parameters: As the vendor-specific configuration parameters are not known to the standardized conformance tests, they must be given specific values as input to the conformance testing; otherwise the module cannot be generated, built and tested.
4. Various files containing "C" source-code, object-code, header-files: these files will typically contain aspects of the vendor's implementation of the module which are not influenced by the values of the configuration parameters
5. A means of generating the module given the configuration parameters: the vendor usually implements a tool which inputs an "ECU configuration description" and outputs either an error (when the configuration is invalid) or additional artifacts (header files, configuration files etc.) which must be used during the build-process or at execution time of the module.

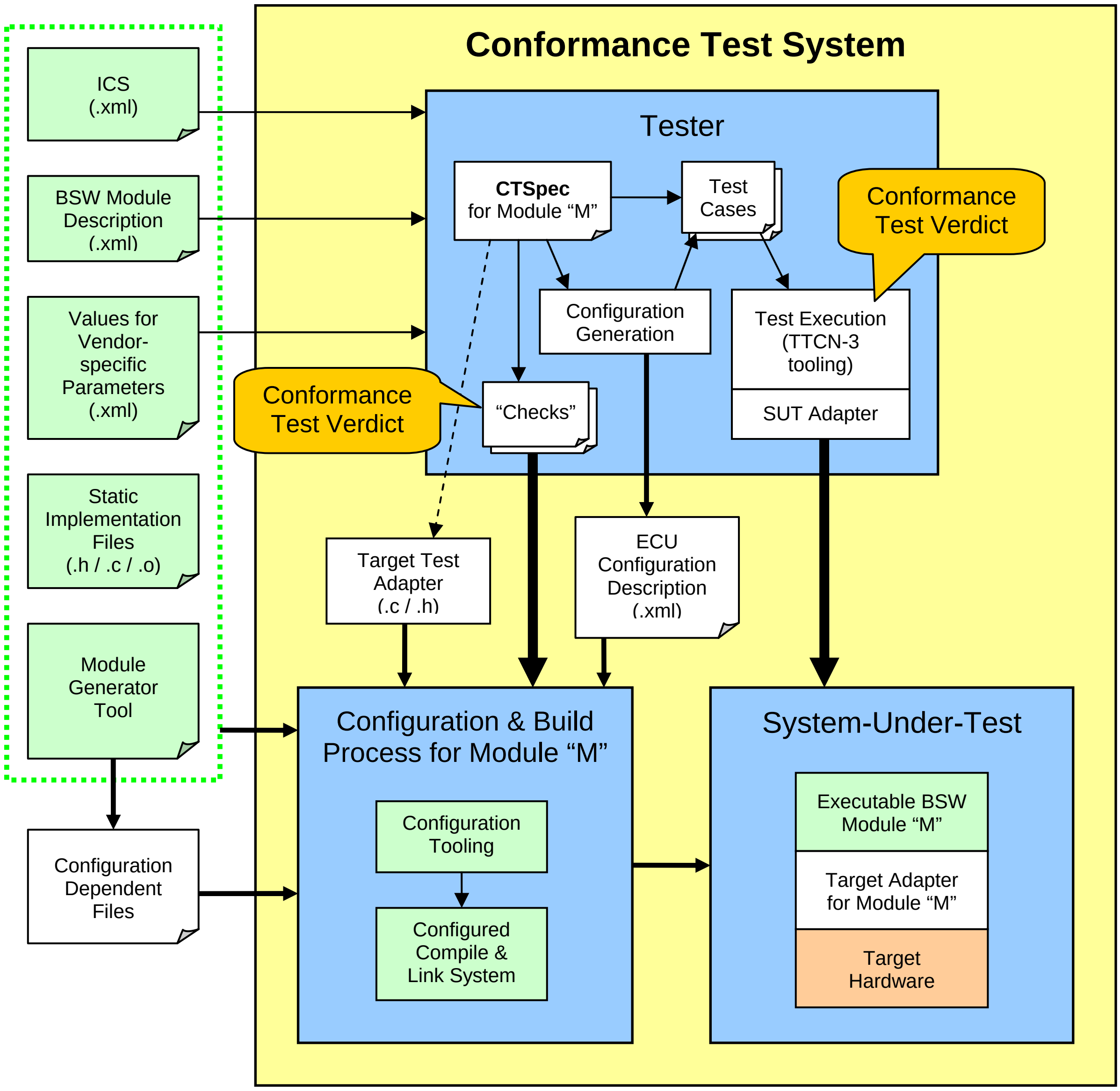

Figure 3- Detailed view on the "Conformance Test System"

The "conformance test system" uses the tools and documents provided by the module vendor to ascertain whether this set of artifacts conforms to the AUTOSAR specifications. The output is a test verdict.

In addition, the "conformance test system" relies on:

1. A specific build suite: typically consisting of a specific compiler and linker and a specific configuration for these tools. Although this tooling is typically not provided by the vendor of the BSW module to be tested, conformance is only ascertained for the specific build tools and for their specific configuration (for example: modifying a compiler configuration parameter might break conformance to the AUTOSAR specifications)
2. A specific target hardware: Especially for the hardware-dependent modules (drivers for specific hardware), conformance can only be ascertained on

specific target hardware (using the module on different hardware might break conformance to the AUTOSAR specifications)

2.4.3 Details of the conformance test system

Based on the previous discussions and the technical decisions taken by AUTOSAR, the conformance test system typically consists of the following main elements:

1. The “tester”:
    a. Tooling[4] that takes the ICS, the BSW module description, the values for the vendor-specific parameters and the configuration values defined by the CTSpec and which generates multiple configuration sets to be used during further steps.
    b. A TTCN-3 based system used to test the dynamic behavior of the built module. This system typically consists of:
        1. An implementation of the test cases and test control in TTCN-3.
        2. Standard TTCN-3 execution environment.
        3. A test adapter on the tool-side (i.e. “SUT adapter”) which (among other things) ensures the communication with the target on which the component runs.
2. The “configuration/build process”:
    a. Tooling that takes AUTOSAR configuration files and transforms them into configuration files usable in the vendor-specific build process.
    b. A vendor-specific build process during which the compile and link system and the vendor-provided tooling is used to build the module for the given configuration. Typically, the module will be linked with test-specific code that also runs on the target.
3. The “system-under-test”
    a. The executable BSW module implementation appropriately configured and linked with the target-side test adapter (i.e. “target adapter”).
    b. The hardware platform required for running the BSW module implementation.

The eventual test verdict is influenced by the success or failure of these different steps:

1. The provided ICS, BSW module description and vendor-specific parameters might not conform to AUTOSAR specifications.
2. The provided tool might not read input data and/or create output data that conforms to AUTOSAR specifications.
3. The build process might fail, or it might produce artifacts that do not conform to AUTOSAR.
4. The dynamic behavior of the module in a specific configuration might not conform to AUTOSAR specifications.

2.5 Details of the CTSpec

The expected output of the CTSpec creation process now is:

[4] In principle, these steps could be done by hand. In practice, the complexity of many AUTOSAR modules is so high, that tooling is absolutely necessary.

1. The parameterizable test cases in TTCN-3 (which can be executed directly on standards-conformant TTCN-3 tooling)
2. Constraints and rules on configuration parameter values that need to be considered when the configuration sets are generated for CTSpec execution and additional configuration files used for the validation of the CTSpecs.
3. Verbal specifications of all other elements needed within the conformance test system
4. Results of the analysis of the relevant AUTOSAR specifications including traces between test cases and AUTOSAR requirements and a definition of the coverage of AUTOSAR requirements by the defined test system

2.6 Characteristics of AUTOSAR influencing the CTSpec Creation process

The conformance test specification process is intended to generate the CTSpec, as defined in Section 2.5, for an arbitrary BSW module.
Thus, it has to consider the following facts:

1. As the input to the CTSpec creation process is a large set of heterogeneous documents, these need to be consolidated into a good test-base
2. During the CTSpec creation process, mistakes or ambiguities in the AUTOSAR specifications might be uncovered and need to be resolved before the creation process can be completed
3. The remaining relevant requirements need to be categorized according to their testability. Some requirements need to be tested during the configuration or build process (and will lead to a verbal description of the test in the CTSpec); others are related to the dynamic behavior of the module (and will lead to executable tests implemented in TTCN-3). Certain requirements will not be testable because of technical limitations.
4. As the AUTOSAR specifications are documents that are further developed and improved, the process must prepare for a maintenance phase, during which changes in the AUTOSAR specifications must be traced to the artifacts in the CTSpec that depend on them.
5. The sheer size and complexity is high: a large number of modules, API's, requirements and configuration parameters must be handled both during the CTSpec creation process and during the execution of the conformance tests. It is necessary to use automation through tools and formal descriptions of inputs and outputs wherever possible.
6. The correctness of the CTSpecs with respect to the AUTOSAR specifications must be ensured. The validation of the implemented test cases of the CTSpecs is very important in order to obtain sufficient confidence that the CTSpecs are suitable for attesting conformance. Because of the aforementioned complexity, this validation should be automated as much as possible.

3 CTSpec Creation Process

This chapter defines the CTSpec creation process, then introduces the roles and how they collaborate and finally describes each activity in more detail.

3.1 Basic approach

Due to the large number of AUTOSAR BSW modules and the complexity of work that is required to create conformance test specifications, the CTSpec creation process has been designed to fulfill the following key requirements:

Division of work	It must be possible to separate work activities between different organizations. Linking separated work activities together requires a good definition of the work artifacts that are exchanged between these work activities.
Division of know-how	The people involved in the creation process cannot not be required to know all details of the creation process and to be proficient with the BSW modules and all employed methods and technologies. The creation process must rather enable the assignment of “specialists” to the work activities. This leads to the definition of various roles with each of them having different know-how requirements.
Quality assurance	The quality of intermediate work artifacts as well as the quality of the final work artifact (i.e. the CTSpec) must be assured. Therefore, reviews of intermediate work artifacts and the CTSpec validation activities have been made inherent parts of the CTSpec creation process.
Work efficiency	Some of the work activities involve recurring work steps that is suitable for automation (e.g. generation of configuration parameter values). Potential for automation shall be exploited as much as possible.
Traceability	The creation process must enable the use of tools that realize the linking between the specification items of the original SWS documents and the test cases of the resulting CTSpec.
Scalability	The creation process must be applicable to both simple BSW modules and to very complex and large BSW modules in the same basic way.

The basic approach on specifying the CTSpec creation process is to divide it into a sequence of “phases” with associated “activities” and then to define the “roles” that perform the activities and the work artifacts that are output of and/or input to the different activities.
The following sections give an overview on the CTSpec creation process from a “management view”, i.e. with an emphasis on roles and their activities in order to enable planning of CTSpec creation projects.
Details on the CTSpec creation process with an emphasis of its realization (i.e. an “employee’s view”) are given in [2].

3.2 Overview of the activities of the CTSpec Creation Process

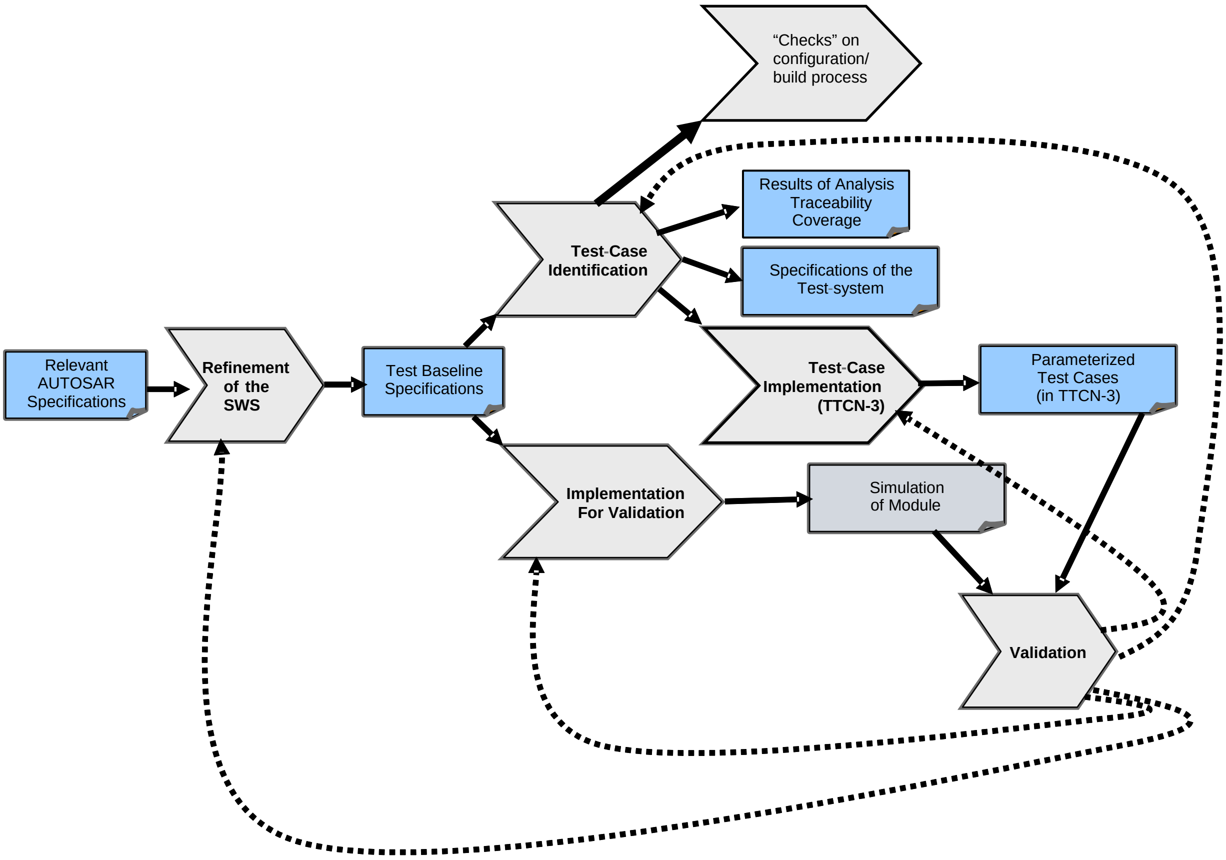

Figure 4 – Overview of the conformance test specification process

Figure 4 depicts the overall activities during the creation of CTSpecs for a BSW module.
Input to the overall creation process are the various AUTOSAR specifications (see Section 2.3 for details) out of which the Software Specification (SWS) document of the BSW module is the most important one.
In a refinement activity, these documents are transformed into a suitable base line for the creation of the conformance test specification.
When a suitable base line is available, the test case identification can start: the output of this activity are a description of the results of the analysis phase, including traceability to requirements and coverage of the tests, a design specification of the test setup and CTSpec architecture, a verbal description of conformance test cases which can not be implemented in TTCN-3 and, finally, a high-level description of the test cases that is input for the "Test Case Implementation in TTCN-3".
In parallel to the "Test Case Identification" and the "Test Case Implementation", an independent implementation of a BSW module simulation for the purpose of validation is prepared.
In the validation activity, the executable test cases (in TTCN-3) are validated against a simulation of the module. During this validation, errors might be uncovered that will lead to iteration of any of the previous activities.

Please note that the implementation of a test execution environment and the actual execution of the test cases is an inherent part of the CTSpec creation process to serve the aforementioned purpose of validation of the test cases against the simulation of the BSW module.

3.3 CTSpec creation process with roles and artifacts

The following figure gives a static view (a time-line is not shown) on the entire CTSpec creation process, the various roles within the process and the artifacts they use to collaborate.

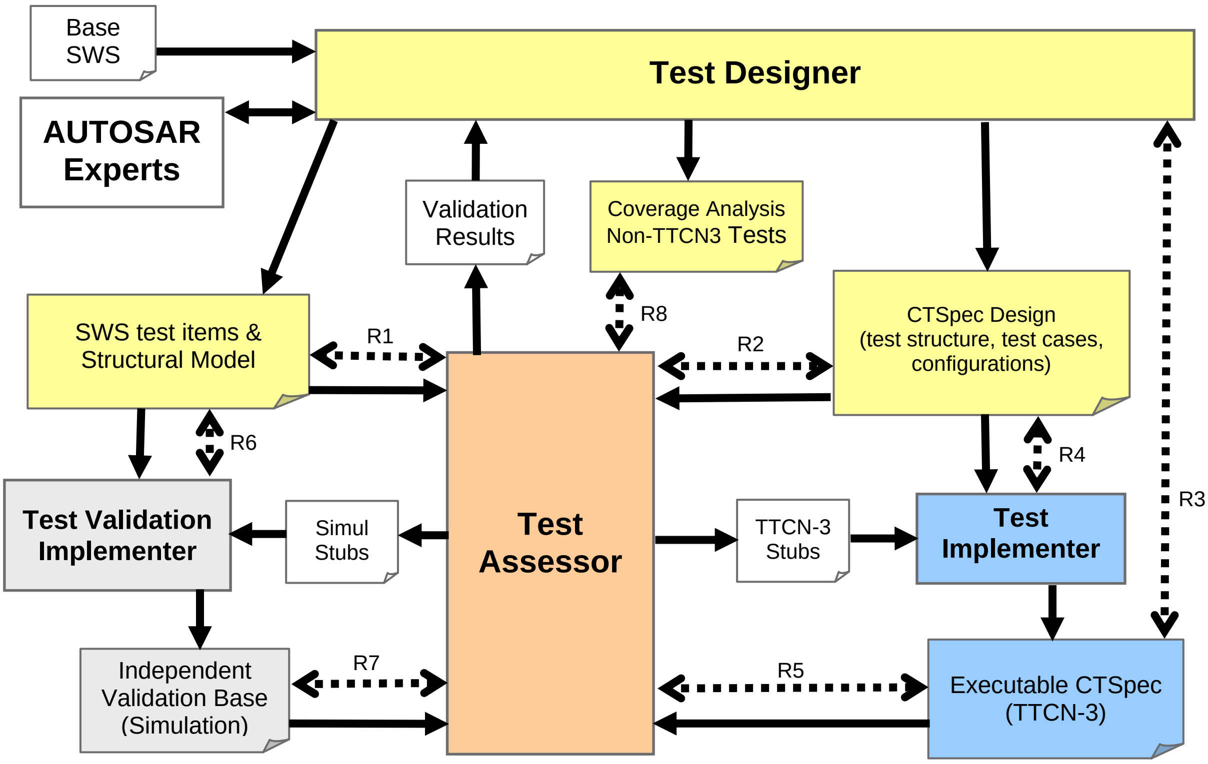

Figure 5 – Detailed structure of the CTSpec creation process with focus on artifacts and roles

Four roles are defined within the CTSpec creation process for a specific module:

- Test Designer
- Test Implementer
- Test Validation Implementer
- Test Assessor

Each “role” is responsible for certain artifacts (which are shown in the same color as the role itself).
In addition, each role is responsible for certain reviews that ascertain ahead of time that input artifacts are suitable or that improve confidence in the quality of an activity.
These reviews are:

R1 Test Assessor verifies that the SWS test items are correct from a formal point of view (contain for example enough information to define API’s and to generate the associated stubs).

R2 Test Assessor verifies that the CTSpec is correct from a formal point of view; that the stubs needed for implementation can be generated.
R3 The Test Designer verifies that the Test Implementer has correctly interpreted the high-level description of the test cases that were implemented in TTCN-3.
R4 The Test Implementer verifies that the high-level definition of the test cases is sufficiently clear to allow an implementation in TTCN-3.
R5 The Test Assessor verifies that the Executable TTCN-3 specs can be executed in the validation environment.
R6 The Test Validation Implementer reviews the SWS to make sure an unambiguous simulation can be built.
R7 The Test Assessor verifies that the module simulation can run in the validation environment.
R8 The Test Assessor verifies that the coverage analysis is complete from a formal point of view (complete traceability, all requirements covered).

The separation between the four roles is based on the following criteria:

- Allow separation of design and implementation from validation: In order to ensure a meaningful validation, the “Test Designer” must work independently from the “Test Validation Implementer” and from the “Test Implementer”
- Split required know-how across different roles (in this creation process, it is unlikely that a single person/company will have expert knowledge in all required aspects)
- Allow for optimized outsourcing between repetitive low-cost activities and expert high-cost activities

The following table gives an overview on the level of required knowledge for the four roles in the relevant areas:

	Test Designer	**Test Implementer**	**Test Validation Implementer**	**Test Assessor**
Module structure & behavior	Expert	Partly	Expert	Partly
AUTOSAR BSW concepts	Expert	Partly	Partly	Expert
CTSpec Methodology & Tools	Partly[5]	Good	Partly	Expert
Test case development	Expert	None	None	Partly
TTCN-3	Passive[6]	Expert	None	Expert

Table 1 - Overview of the know-how required in the different roles

[5] coaching available in project

[6] for review; some pre-existing know-how desired

3.4 Detailed description of roles and activities

3.4.1 The “Test Designer” role

The following table summarizes the role of the “Test Designer”:

Role of the “Test Designer”	
INPUT: • Base SWS (and associated documents) • Implementation of test cases in TTCN-3 (for Review) • Results of the validation	**OUTPUT:** • SWS-items for test • Structural model of the module (UML) • Structural model of the test architecture (UML) • High-level description of all test cases • Description of non-TTCN-3 test cases • Coverage analysis
TASKS / RESPONSIBILITIES: • Responsible for all “OUTPUT” documents • Responsible for the quality of the test cases for the module (in terms of contents, coverage of and consistency with the SWS test items) taking into account the results of the validation • R3: review of the TTCN-3 implementation (scope: make sure the patterns used in the high-level description are understood by the implementer) • Rp: review of the AUTOSAR CTSpec Creation Process and Methodology and participation in related coaching	**KNOW-HOW:** • In-depth understanding of AUTOSAR (BSW architecture, configuration, UML model, …) • In-depth understanding of the relevant BSW modules (expert-level) • Experience in identifying test cases out of requirements • Basic TTCN-3 knowledge is desirable • Knowledge of AUTOSAR CTSpec Creation Process and Methodology (gained during project)

Table 2 - Detailed description of the role of the "Test Designer"

The Test Designer is in particular responsible for the activities “Refinement of the SWS” and “Test Case Identification” which are described in the following sections.

3.4.1.1 The “Testable Items Extraction” activity

Inputs to this activity are the “Input Specifications” in a given version. This set of documents includes:

- The Software Specification (SWS) document of the BSW module under consideration
- Additional SWS documents of components that have interdependencies with the BSW module and additional relevant AUTOSAR specifications

The objective of the activity is to extract all testable items from the set of SWS-items in the SWS for the BSW module under consideration.
The output of this activity is the set of all testable items extracted from the respective SWS – the ‘SWS test items’.

3.4.1.2 The "Test Case Identification" activity

The "Test Case Identification" is the most critical and challenging activity within the CTSpec creation process, which makes "Test Designer" into the most demanding role.
The Test Designer works on three axes:

1. Overall Structure and Interfaces: The Test Designer is responsible for defining the overall structure of the test system and the interfaces between the components within the test system: this test system includes the BSW module under test, the test cases and a number of additional "test-components" which facilitate the development and execution of test cases.
2. Detailed Requirements: The Test Designer is responsible for analyzing each requirement and categorizing the requirement according to its testability. For test cases not related to the dynamic behavior of the module, a simple verbal description of the test case suffices. For test cases that are related to the dynamic behavior of the module under test, the Test Designer must identify one or more test cases to cover the requirement. These test cases must be described in sufficient detail so that an implementation in TTCN-3 is possible in a later activity. The test cases are parameterized by configuration parameters: these can be a mix of standardized AUTOSAR configuration parameters for the module and additional test case specific parameters.
3. Configuration Parameter Values: These parameters are a mix of standardized AUTOSAR configuration parameters for the module and additional parameters introduced as part of the test case definition. To generate the sets of configuration parameter values, the Test Designer must analyze interdependencies between the parameters, must define the values to be used for parameters with a large value range and must define constraints on valid combinations of parameter values. The Test Designer must specify the test cases in a way that later, during test execution, they can be parameterized with configuration values that satisfy the constraints of the ICS which therefore may reduce the value ranges of the configuration parameters according to what the object under test actually implements. Furthermore, during the test execution phase the configuration parameters and value ranges have to be checked against the BSWMD ranges to ensure compatibility since the ICS cannot specify anything that exceeds limits set in the BSWMD. All this information must be formalized so that it is usable by tooling.

3.4.2 The "Test Implementer" role

The following table summarizes the role of the "Test Implementer":

Role of the "Test Implementer"	
INPUT: • High-level description of all test cases to be implemented • Structural model of the module (UML) • Structural model of the test architecture (UML) • Stubs and interfaces in TTCN-3 • A "no-op" simulation of the BSW module	**OUTPUT:** • Executable implementation of the test cases in TTCN-3 • TTCN-3 implementation guidelines
TASKS / RESPONSIBILITIES:	**KNOW-HOW:**

• Responsible for the implementation in TTCN-3 • R4: ensure that high-level description of the test cases allows implementation • Responsible for the quality of the implementation (correct TTCN-3, adherence to coding guidelines, documentation in TTCN-3, correct execution against “no-op” simulation) • Participation in validation workshop • Rp: review of the AUTOSAR CTSpec Creation Process and Methodology and participation in related coaching	• Deep experience with TTCN-3 implementation and tooling • In-depth understanding of the relevant aspects of the AUTOSAR CTSpec Methodology (including, for example, coding guidelines, configuration concepts) • Good overview of the AUTOSAR BSW architecture • Rough understanding of the relevant module(s)

Table 3 - Detailed description of the role of the "Test Implementor"

The Test Implementer is in particular responsible for the activity “Test Case Implementation”.

3.4.2.1 Details for Activity “Test Case Implementation”

The implementation activity comprises the actual coding of the test cases.
In [2], further recommendations on the design and the implementation strategy for the CTSpec are given.

3.4.3 The “Test Validation Implementer” role

The following table summarizes the role of the “Test Validation Implementer”:

Role of the “Test Validation Implementer”	
INPUT: • SWS test items and structural model • Stubs needed for implementation of the validation (e.g. in Java)	**OUTPUT:** • Normally: executable simulation of the module in e.g. Java • A “no-op” simulation of the BSW module • Exceptions: for some modules, a different validation approach might be necessary • Implementation guidelines and process description for e.g. Java simulation implementation (or implementation in other technology)
TASKS / RESPONSIBILITIES: • Responsible for the preparation of an executable validation of the module (normally: a complete simulation of the module in e.g. Java) • R6: review the SWS test items and the structural model (must be complete and unambiguous) • Participation in “validation workshops” • Rp: review of the AUTOSAR CTSpec Creation Process and Methodology and participation in related coaching	**KNOW-HOW:** • Good understanding of AUTOSAR concepts (BSW architecture, configuration, UML model, …) • In-depth understanding of the dynamic behavior of the relevant BSW module(s) • (For most modules): very good programming capabilities (e.g. in Java)

Table 4 - Detailed description of the role of the "Test Validation Implementer"

The Test Validation Implementer is in particular responsible for the activity “Test Case Validation Implementation”.

3.4.3.1 The “Test Case Validation Implementation” activity

The goal is to come to an independent implementation against which the TTCN-3 implementation can be verified during the “test case validation” activity.
This implementation is a simulation of the BSW module and needs to cover 2 aspects:

1. It needs to simulate the functional behaviour of a correctly implemented BSW module (only the dynamic behaviour relevant for conformance testing must be simulated)
2. It needs to be able to simulate an erroneously implemented BSW module[7]

3.4.4 The “Test Assessor” role

The following table summarizes the role of the “Test Assessor”:

Role of the “Test Assessor”	
INPUT: • SWS test items and structural model • CTSpec design • Executable tests • Validation base (typically a simulation in e.g. Java)	**OUTPUT:** • Stubs for TTCN-3 implementation • Stubs for validation implementation (e.g. Java) • Results of validation • CTSpec Creation Process & Methodology • Validation reports
TASKS / RESPONSIBILITIES: • Coaching related to the output documents • R1: review of SWS test items and structural model (focus: tools can be used to generate relevant stubs) • R2: review of CTSpec design (focus: formally correct so that tools can be used to generate relevant stubs and configurations) • R5/R7: review of CTSpec and simulation implementations (as part of validation of TTCN-3 against simulation) • R8: review of analysis and non-TTCN-3 specs (for formal completeness and consistency with validation) • Ri and Rj: review of coding guidelines and process used by the Implementer and the Validation Implementer • Executing the validation and coordinating associated validation activities	**KNOW-HOW:** • Deep know-how of TTCN-3 language and methodology • Deep knowledge of overall AUTOSAR concepts (especially in the area of BSW configuration) • Deep knowledge of the AUTOSAR methodology to develop CTSpecs • Rough knowledge of the module (mainly structurally)

Table 5 - Detailed description of the role of the "Test Assessor"

[7] This aspect of the validation approach is not entirely clear and technically not entirely solved. Therefore, this 2nd aspect of the validation implementation might be modified in future versions of this document.

The CTSpec creation process needs to be supported by several tools, which handle the complexity of the API's, the configuration process and which make sure that the API's of the BSW module, the TTCN-3 tests and the simulation fit together.
The Test Assessor has the responsibility to maintain and run these tools and to make sure that all artifacts are complete and fit together from a formal point of view.
Furthermore, the Test Assessor is also in charge of the activity "Test Case Validation".

3.4.4.1 Details for Activity "Test Case Validation"

The activity Test Case Validation is run by the "Test Assessor" but requires active participation of all other roles.
The goal of "test case validation" is to increase confidence in the correctness of the executable TTCN-3 code by executing it against an independently developed simulation.
To be concrete, the Test Assessor collects the executable TTCN-3 code, the configurations to be used during conformance testing and the executable module simulation and runs them against each other in a validation framework maintained by the Test Assessor.
This activity requires active participation of the other roles. In case the validation does not return the expected results (for example, a test case reports "failed" against a correct simulation of the BSW module), the owners of the relevant documents need to work together to find the cause of the problem.

- Failing validation might be caused by ambiguities in AUTOSAR specifications (where the designer and the validation implementer interpreted the same SWS in a different way).
- Validation might show mistakes in the test case identification (where the test case does not conform to the SWS).
- Validation might fail because of bugs in the implementation in TTCN-3 (where the TTCN-3 code is not a correct translation of the test design).
- Validation might fail because of errors in the simulation (where the simulation does not conform to the SWS).
- Validation might fail because of problems in the validation framework or the TTCN-3 tooling itself.
- Validation might fail because the test case results "passed" although a specified misbehavior with respect to the functionality under test has been switched on in the simulation.

This will work best in a "validation workshop" during which all roles are physically together.
The result of the validation is a "validation report"[8] showing, for all test cases and run configurations, whether the validation passed or failed and what the identified cause (in case of a "fail") is.
Depending on the observed quality and the analysis of the causes, previous activities might need to be repeated to improve the quality of the CTSpecs.

---

[8] In addition, a "coverage report" might be generated, which shows the code-coverage of the simulation and/or the TTCN-3 implementation during these validation runs. Whether and how this will be done is not entirely clear yet at the time of writing.