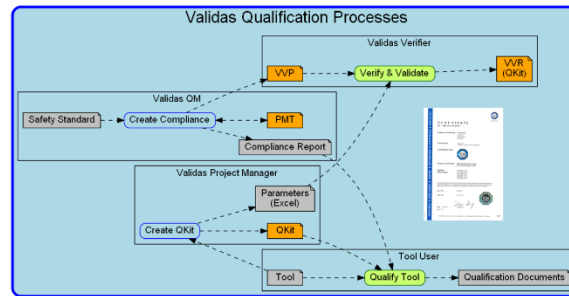


# Process Report for Perform Module Test



Version:	Template 0.6 / Document 1.0
Maximal Safety Level:	ASIL_D
Date:	2019-03-11
Status:	Generic / Adapted / Presented / Generated / Reviewed / <b>Final</b>
Author:	Dr. Oscar Slotosch
File:	PR.docm
Size:	34 Pages



## Template Revision History:

Version	Date	Status	Author	Change
0.1	2018-10-16	Generic	Slotosch	Created initial template

0.2	2018-10-29	Generic	Slotosch	Added Configuration Section 3
0.3	2018-11-11	Generic	Slotosch	Added Glossary
0.4	2018-11-11	Generic	Slotosch	Added Maximal Safety Level
0.5	2018-12-16	Generic	Slotosch	Added Tool Section
0.6	2019-01-18	Generic	Slotosch	Added Process Notation

## Document Revision History

Version	Date	Status	Author	Change
0.1	2019-03-11	Generic	Slotosch	Created from Generic Template 0.6
0.3	2019-03-11	Generated	oscar	Configured Process Parameters, Selected Requirements Claims and generated document using PMT
0.4	2019-03-11	Reviewed	Oscar Slotosch	Reviewed Process
1.0	2019-03-11	Final	Oscar Slotosch	Finalized document

## Contents

<b>1</b>	<b>Scope of this Document.....</b>	<b>5</b>
<b>2</b>	<b>Process Notation .....</b>	<b>6</b>
2.1	General.....	7
2.2	Process View.....	8
2.3	Artifact View.....	9
2.4	Stake Holder View .....	9
<b>3</b>	<b>Parameters of Perform Module Test.....</b>	<b>10</b>
3.1	Types in Perform Module Test .....	10
3.2	Project Parameter of Perform Module Test .....	11
3.3	Project Parameter of Perform Module Test .....	11
<b>4</b>	<b>Roles in Perform Module Test .....</b>	<b>11</b>
<b>5</b>	<b>Artifacts in Perform Module Test.....</b>	<b>13</b>
5.1	Artifact SUT.....	13
5.2	Artifact Safety Case .....	13
5.2.1	Artifact Code Coverage Report .....	14
5.2.2	Artifact Summary .....	15
5.2.3	Artifact Test Case.....	16
5.2.4	Artifact Test Report .....	17
5.3	Artifact Specification .....	18
5.4	Artifact Test Specification .....	19
<b>6</b>	<b>Process Perform Module Test .....</b>	<b>20</b>
6.1	VerificationModule Analyze Code Coverage.....	22
6.1.1	VerificationModule Analyze MCDC.....	23
6.2	VerificationModule Analyze Test Results.....	24
6.3	ProcessModule Execute Test .....	25
6.3.1	ProcessModule Measure MCDC.....	26
6.3.2	ProcessModule Run Test.....	26
6.4	ProcessModule Specify Test.....	27
6.4.1	ProcessModule Generate Tests .....	29
6.4.2	ProcessModule Model Test Behaviour .....	29
6.4.3	VerificationModule Validate Tests .....	30
<b>7</b>	<b>Tools in Perform Module Test.....</b>	<b>31</b>
<b>8</b>	<b>Glossary .....</b>	<b>32</b>
<b>9</b>	<b>References .....</b>	<b>34</b>

# 1 Scope of this Document

This document describes how the process Perform Module Test is performed. The process complies with its requirements, see [CR].

The document contains the following chapters:

- 1) This introduction
- 2) Process **Notation**: describing the used notation, especially explain the images
- 3) The **configuration** of the process, i.e. the types and process variables and their values
- 4) The **roles** / stakeholders in the process
- 5) The **artifacts** and their structure including models
- 6) The **process** steps ("Process Modules") modules (tasks)
- 7) The used **tools**.

The document concludes with a glossary in Section 8 and references in Section 9.

## 2 Process Notation

The used process notation is based on a model and consists of the following elements

- StakeHolder: A person responsible for Processes and Artifacts
- ProcessModule: A modular task/process with input & output artifacts
- Verification Module: a special module that verifies an artifact and checks that a requirement is satisfied by asking some review questions (called "Criteria")
- Model: a special form of an Artifact, allowing to define which modeling elements (e.g. Tool, Feature,.. in TCA Models) are mandatory (e.g. Tool-Name) and which are optional (e.g. Feature-Comment).
- Parameter: A parameter of the process indicating that the process has to be iterated for all values of the parameter. Parameters are used to tailor the process and to instantiate it. Parameters can be bound to values or list of values using Bindings.

There are the following parameters:

- Process Parameter: Describes a tailoring parameter, typically instantiated with concrete values. All elements that have variant terms with process parameters evaluating to false are removed.<sup>1</sup>
- Project Parameter: A Parameter indicating instances of the process, typically instantiated with lists of values indicating that the process modules have to be repeated for each value of the parameter
- Process Variable: Describing a condition in the process with alternative following processes. Process variables are not instantiated but determined when executing the process, e.g. "<REVIEW\_OK> -> YES -> Release"

All elements are contained in a container called Process.

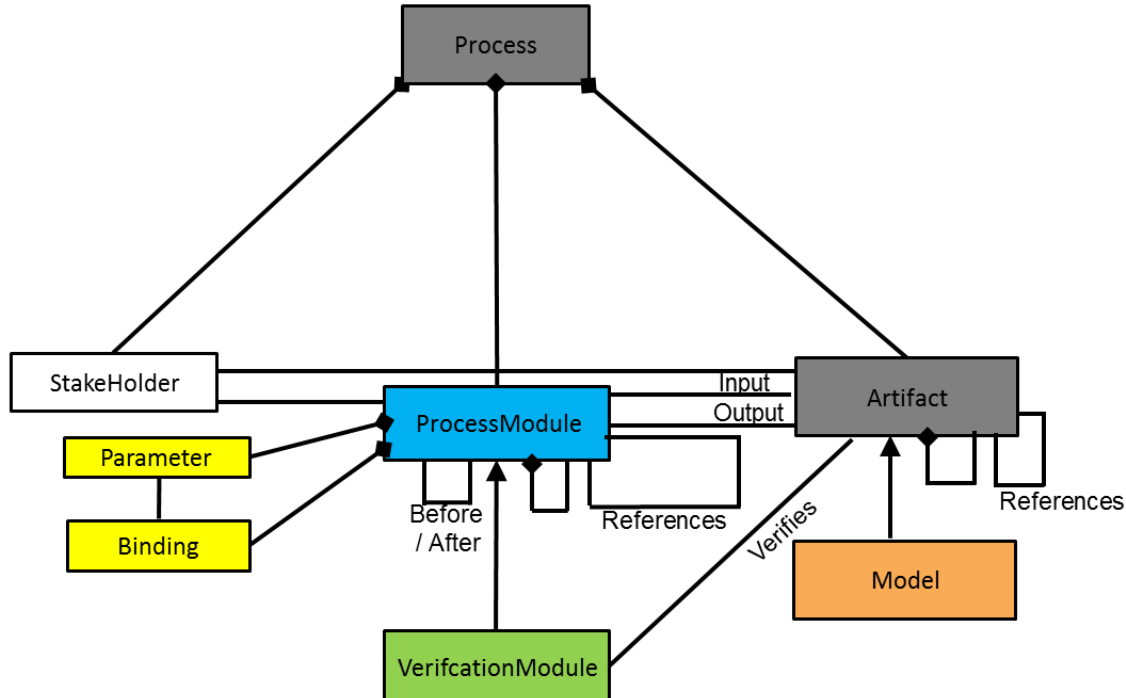


Figure 1: Main Process Description Elements and Relations

<sup>1</sup> This process has already been tailored, such that all process variables have values and all unused process elements are removed.

The graphical images are using a Business Process Modeling Notation BPMN like notation, see [BPMN] with “swim-boxes” instead of “swim-lines” to improve layout.

The following graphical views are supported:

- Process View: describes the process modules, see Section 2.1
- Artifact view: describes the structure of artifacts and their use, see Section 2.3
- Role View: describes the roles with their responsibilities, see Section 2.4

## 2.1 General

The graphical notation visualizes the following elements:

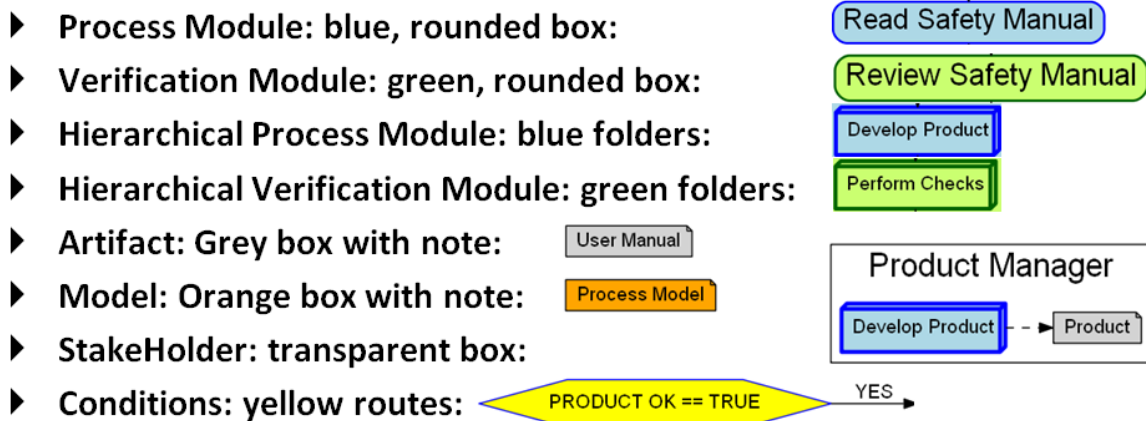


Figure 2: Visualization of Process Model Elements

The graphical notation visualizes the following relations between elements:

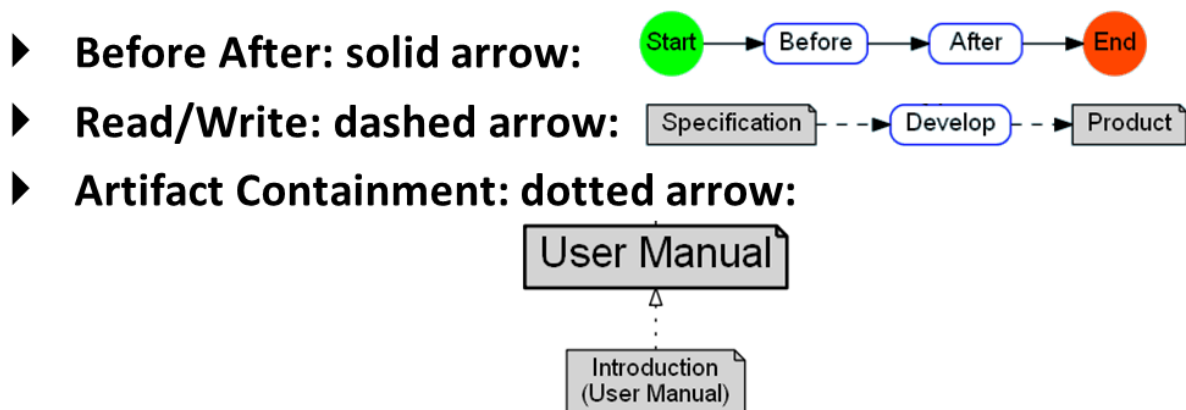


Figure 3: Relations between Model Elements

## 2.2 Process View

The process view describes a Process Module / Verification Module using the graphical notation of Section 2.1. The name of the selected module is written in double size. An example can be found in Figure 4.

It shows the process module “Develop Safe Product” and it’s sub-processes grouped into the swim-boxes of the involved stake holders.

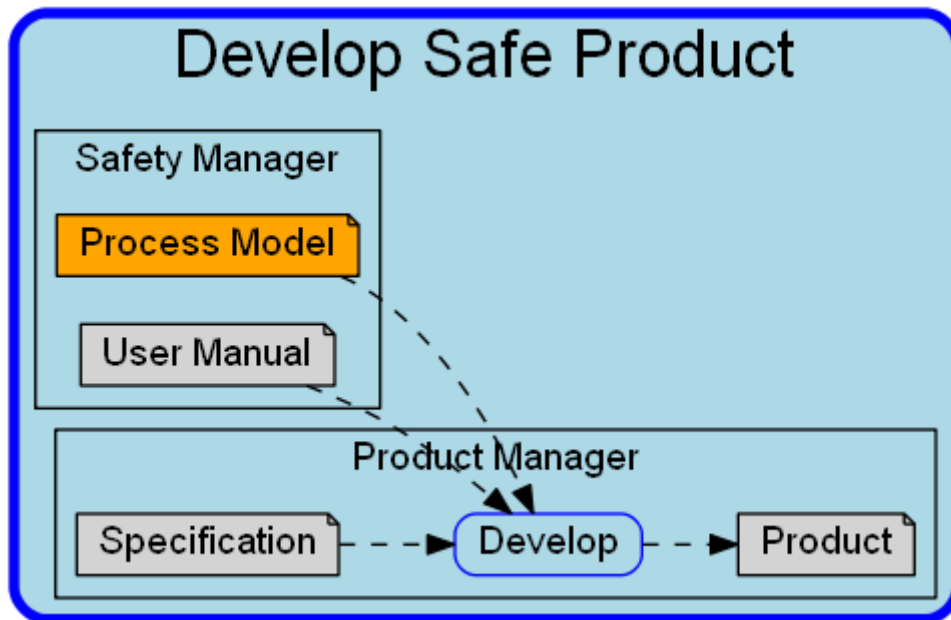


Figure 4: Process View for “Develop Safe Product”

In addition to this input output related modeling style, processes can also be modelled without artifacts using a sequential style by using the after/before relation as shown in the example in Figure 5. This shows a simple process “Sell Product” as interaction between a sales manager and a customer.



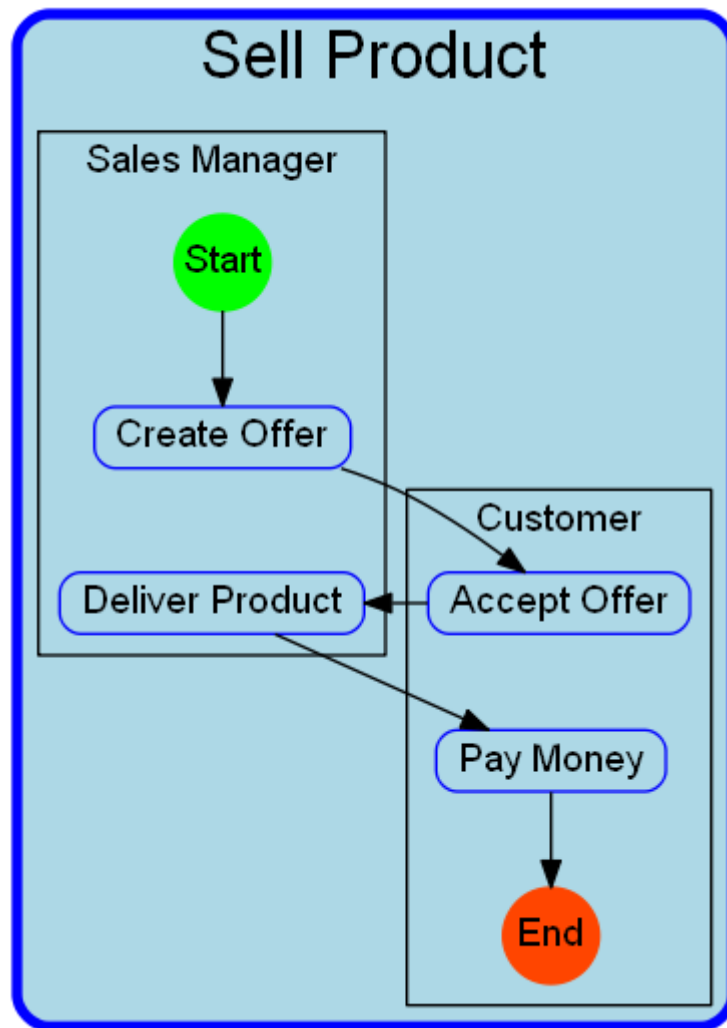


Figure 5: Sequential Process Module View for “Sell Product”

### 2.3 Artifact View

The artifact view describes an Artifact or Model using the graphical notation of Section 2.1. The name of the selected Artifact or Model is written in double size. An example can be found in Figure 6.

It shows the user manual and it’s content (here only the introduction) and that it is used from the product manager for development.

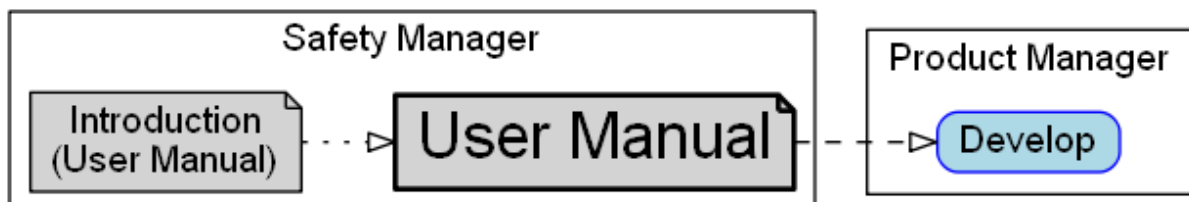


Figure 6: Artifact View for “User Manual”

### 2.4 Stake Holder View

The stake holder view describes a Stakeholder using the graphical notation of Section 2.1. The name of the selected Stakeholder is written in double size. An example can be found in

Figure 7. It shows the two hierarchic processes owned by the Product Manager to produce a product and his responsibilities for the specification.

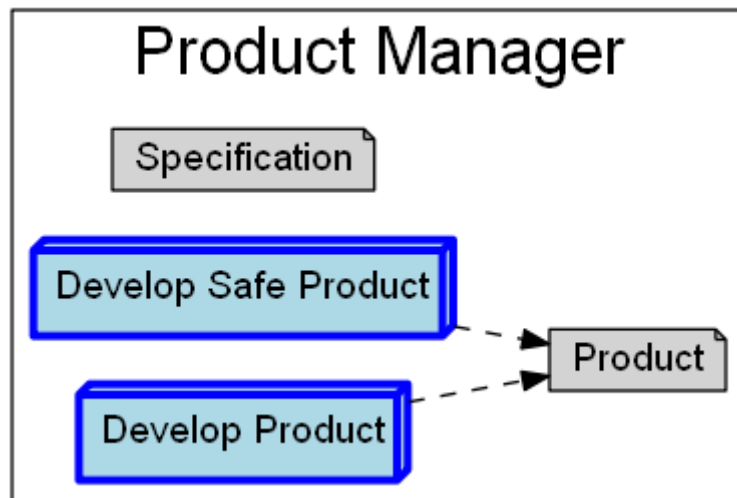


Figure 7: Stakeholder View for "Product Manager"

### 3 Parameters of Perform Module Test

This section describes the 2 parameters and 2 types for Perform Module Test. The parameters are typed and this section describes also the used 2 types and their values.

#### 3.1 Types in Perform Module Test

This section describes the 2 types for 'Perform Module Test' and their possible values.

EnumType ISO_ASIL	
<b>Name:</b>	ISO_ASIL
<b>Description:</b>	The type for all Automotive Safety Integrity Level (ASIL) as defined in ISO 26262.
<b>Enumeration Values:</b>	<ul style="list-style-type: none"> <li>ASIL_A: Automotive Safety Integrity Level A</li> <li>ASIL_B: Automotive Safety Integrity Level B</li> <li>ASIL_C: Automotive Safety Integrity Level C</li> <li>ASIL_D: Automotive Safety Integrity Level D</li> </ul>

Table 1 EnumType ISO\_ASIL

Type String	
<b>Name:</b>	String
<b>Description:</b>	The standard type of Strings with arbitrary values, usable for names, values,...

Table 2 Type String

### 3.2 Project Parameter of Perform Module Test

This section describes the 1 process variable for Perform Module Test.

ProcessVariable ISO_ASIL	
<b>Name:</b>	
ISO_ASIL	
<b>Description:</b>	
The current ASIL of the project.	
<b>Type:</b>	
ISO_ASIL	
<b>Value:</b>	
ASIL_D	

Table 3 ProcessVariable ISO\_ASIL

### 3.3 Project Parameter of Perform Module Test

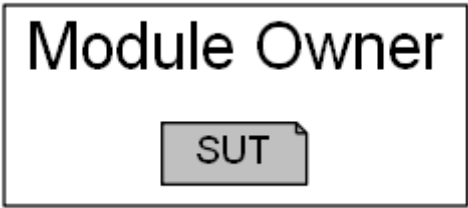
This section describes the 1 project parameter for Perform Module Test.

ProjectParameter MODULE	
<b>Name:</b>	
MODULE	
<b>Description:</b>	
The name of module that is currently tested.	
<b>Comment:</b>	
The name can vary for different modules.	
<b>Type:</b>	
String	
<b>Value:</b>	
To be defined in project	

Table 4 ProjectParameter MODULE


## 4 Roles in Perform Module Test

This section describes the involved stakeholders and their roles in Perform Module Test.

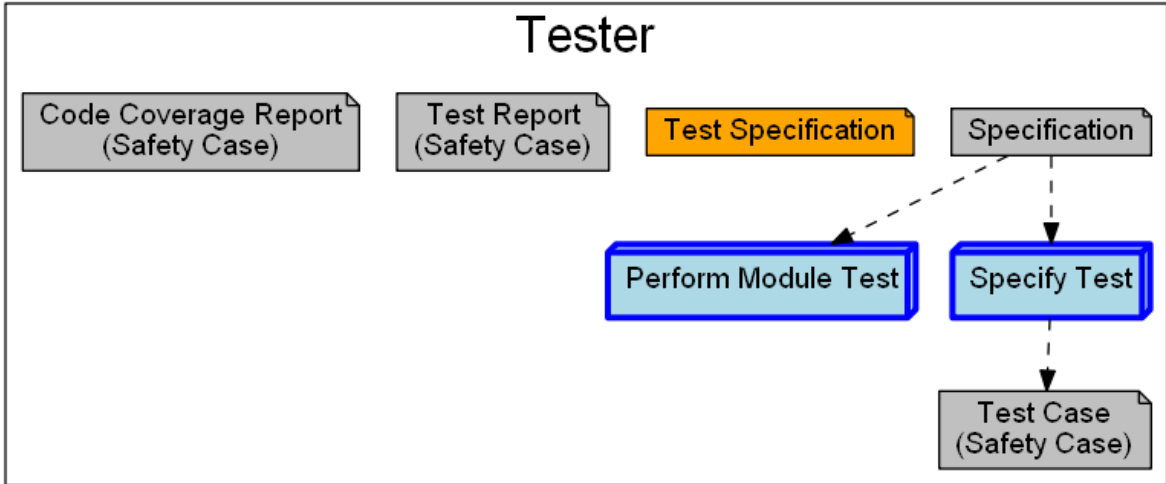
Stake Holder Module Owner	
<b>View of Module Owner</b>	
	
<b>Name:</b>	
Module Owner	

<b>Description:</b>
Is responsible for the specification and implementation of the module.

**Table 5 Stake Holder Module Owner**

Stake Holder Safety Manager	
View of Safety Manager	
	
<b>Name:</b>	
Safety Manager	
<b>Description:</b>	
The Functional Safety Manager with all tasks assigned in the ISO 26262 (or other safety standards).	

**Table 6 Stake Holder Safety Manager**

Stake Holder Tester	
View of Tester	
	
<b>Name:</b>	
Tester	
<b>Description:</b>	
Responsible for testing the module including creation of test specification, implementation analysis.	
<b>Owned Process Modules:</b>	
<ul style="list-style-type: none"> <li>• Perform Module Test, see Table 16</li> <li>• Specify Test, see Table 23</li> </ul>	

**Table 7 Stake Holder Tester**

## 5 Artifacts in Perform Module Test

This section describes the 8 artifacts used by the Perform Module Test.

### 5.1 Artifact SUT

This section describes artifact SUT.

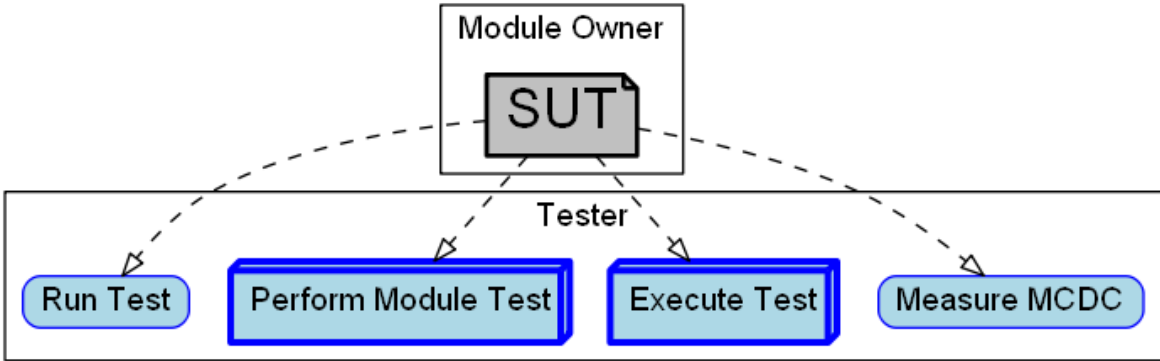
Artifact SUT	
View of SUT	
	
<b>Name:</b>	SUT
<b>Description:</b>	Subject Under Test, typically SW modules.
<b>Path:</b>	ModuleTest/Artifacts/Modules
<b>Main StakeHolder (Owner):</b>	Module Owner
<b>Is Project Input:</b>	True
<b>Used By:</b>	<ul style="list-style-type: none"><li>• Execute Test, see Table 20</li><li>• Measure MCDC, see Table 21</li><li>• Perform Module Test, see Table 16</li><li>• Run Test, see Table 22</li></ul>

Table 8 Artifact SUT

### 5.2 Artifact Safety Case

This section describes artifact Safety Case.

Artifact Safety Case	
View of Safety Case	

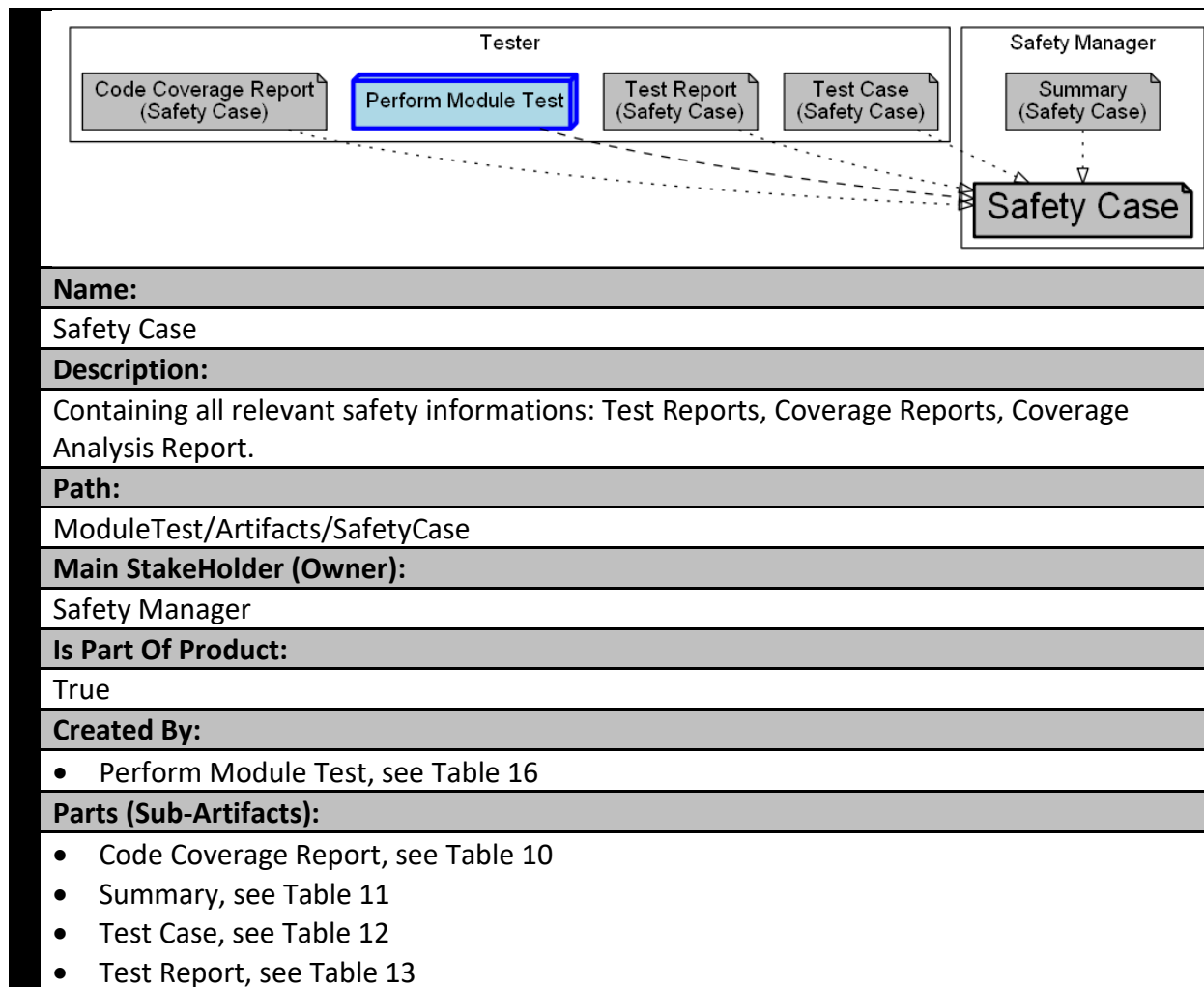


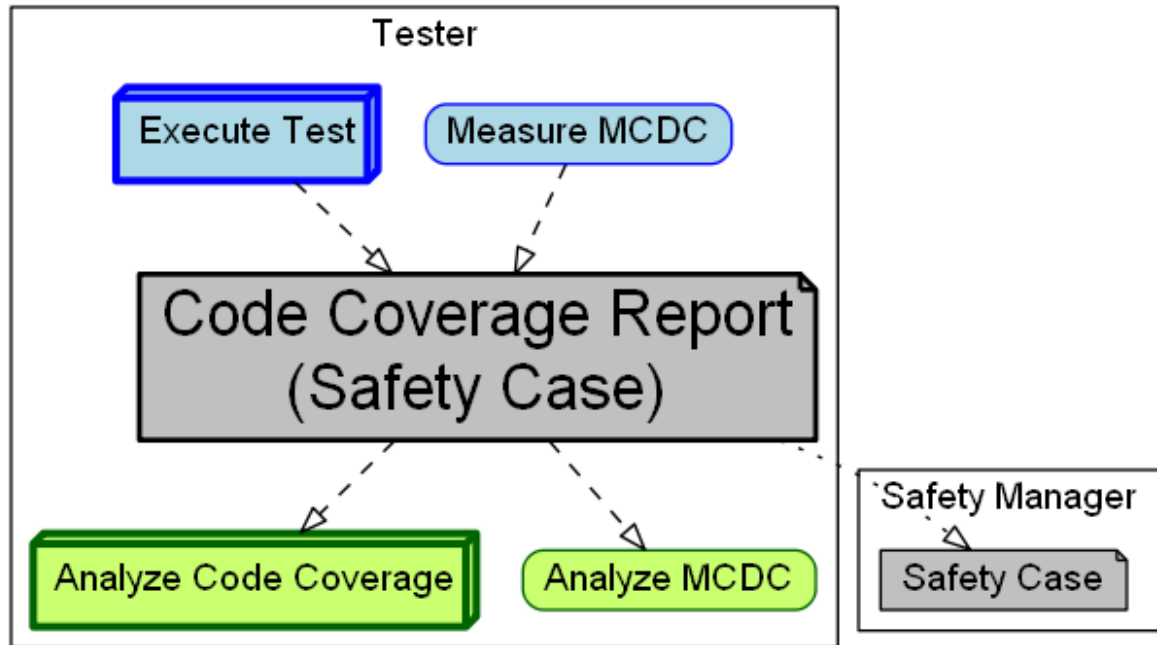
Table 9 Artifact Safety Case

## 5.2.1 Artifact Code Coverage Report

This section describes artifact Code Coverage Report.

### Artifact Code Coverage Report

#### View of Code Coverage Report



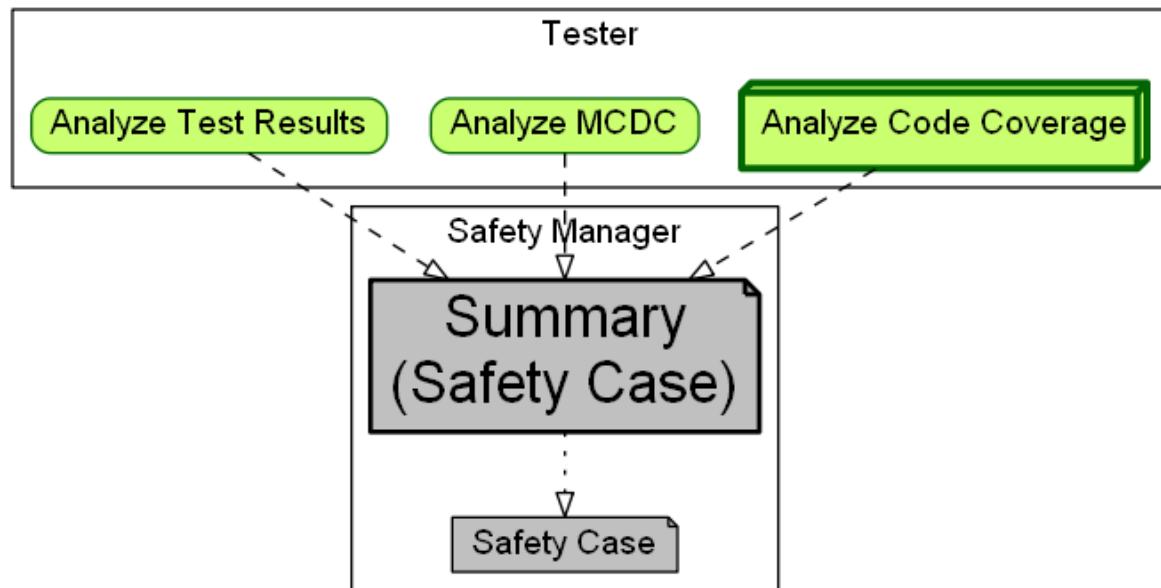
<b>Name:</b>
Code Coverage Report
<b>Description:</b>
The report with the selected coverage (Statement, Branch, MCDC).
<b>Artifact Hierarchy:</b>
Code Coverage Report (in Safety Case)
<b>Path:</b>
ModuleTest/Artifacts/SafetyCase/CoverageReport
<b>Main StakeHolder (Owner):</b>
Tester
<b>Created By:</b>
<ul style="list-style-type: none"> <li>Execute Test, see Table 20</li> <li>Measure MCDC, see Table 21</li> </ul>
<b>Used By:</b>
<ul style="list-style-type: none"> <li>Analyze Code Coverage, see Table 17</li> <li>Analyze MCDC, see Table 18</li> </ul>

Table 10 Artifact Code Coverage Report

## 5.2.2 Artifact Summary

This section describes artifact Summary.

<b>Artifact Summary</b>
<b>View of Summary</b>



<b>Name:</b>
Summary
<b>Description:</b>
ModuleTest/Artifacts/SafetyCase/Summary.
<b>Comment:</b>
Note: The summary can also contain limitations that restrict the usage of some modules, e.g. do not use this module with inputs X and Y
<b>Artifact Hierarchy:</b>
Summary (in Safety Case)
<b>Path:</b>
Summarizes the safety of the modules, either successfully or NOT successfully (with some argumnets).
<b>Is Part Of Product:</b>
True
<b>Created By:</b>
<ul style="list-style-type: none"> <li>Analyze Code Coverage, see Table 17</li> <li>Analyze MCDL, see Table 18</li> <li>Analyze Test Results, see Table 19</li> </ul>

Table 11 Artifact Summary

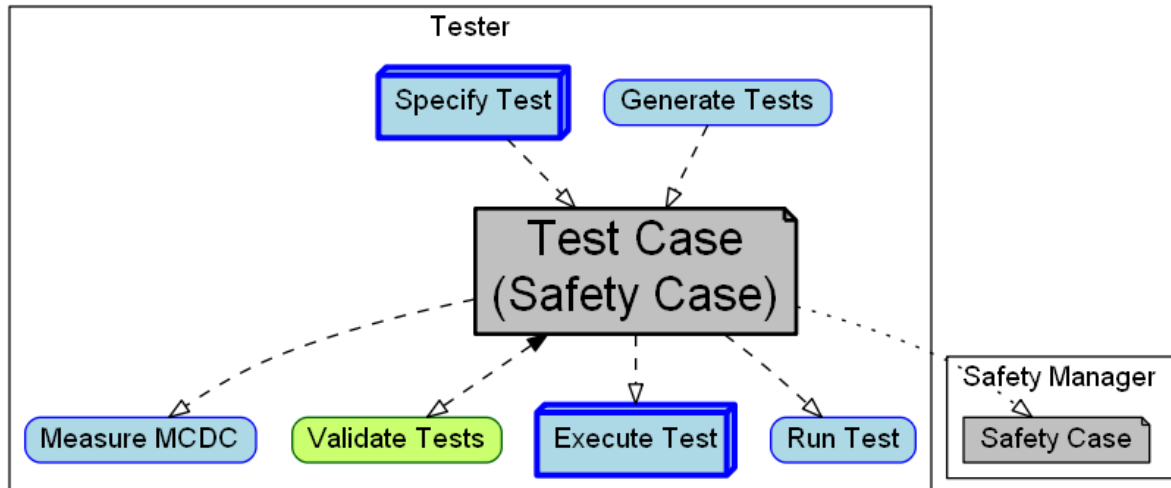
### 5.2.3 Artifact Test Case

This section describes artifact Test Case.

#### Artifact Test Case

##### View of Test Case





Name:
Test Case
Description:
A module test case (implementation).
Artifact Hierarchy:
Test Case (in Safety Case)
Path:
ModuleTest/Artifacts/SafetyCase/Tests
Main StakeHolder (Owner):
Tester
Created By:
<ul style="list-style-type: none"> <li>• Generate Tests, see Table 24</li> <li>• Specify Test, see Table 23</li> <li>• Validate Tests, see Table 26</li> </ul>
Used By:
<ul style="list-style-type: none"> <li>• Execute Test, see Table 20</li> <li>• Measure MCDC, see Table 21</li> <li>• Run Test, see Table 22</li> <li>• Validate Tests, see Table 26</li> </ul>

Table 12 Artifact Test Case

## 5.2.4 Artifact Test Report

This section describes artifact Test Report.

### Artifact Test Report

#### View of Test Report

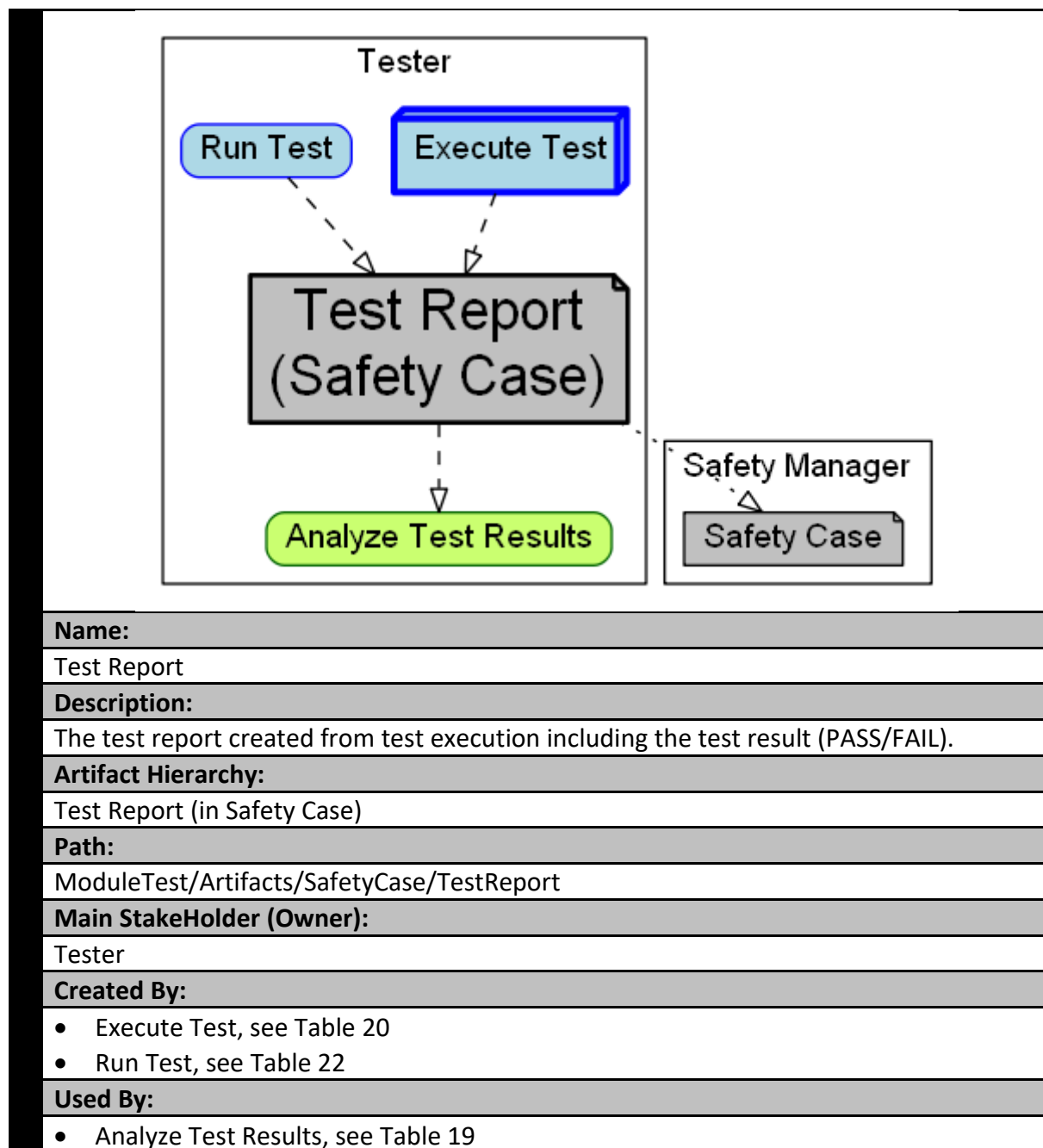


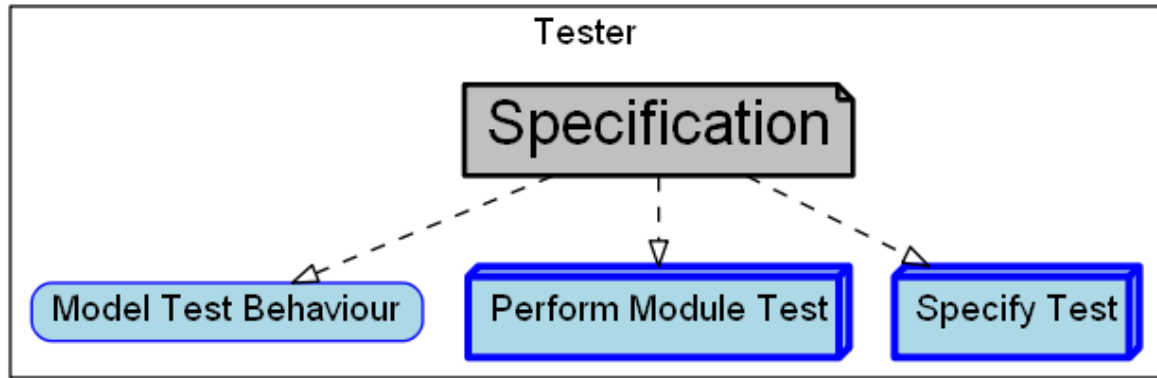
Table 13 Artifact Test Report

### 5.3 Artifact Specification

This section describes artifact Specification.

#### Artifact Specification

##### View of Specification



**Name:**

Specification

**Description:**

Functional Specification.

**Path:**

ModuleTest/Artifacts/Specification

**Main StakeHolder (Owner):**

Tester

**Is Project Input:**

True

**Used By:**

- Model Test Behaviour, see Table 25
- Perform Module Test, see Table 16
- Specify Test, see Table 23

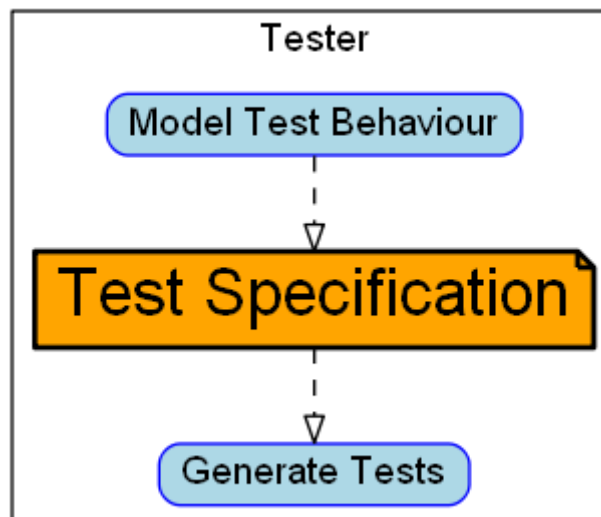
**Table 14 Artifact Specification**

## 5.4 Artifact Test Specification

This section describes artifact Test Specification, which is a model.

### Model Test Specification

#### View of Test Specification



<b>Name:</b>
Test Specification
<b>Description:</b>
Model based test specification, formalized e.g. using LTG or ForeC++.
<b>Path:</b>
ModuleTest/Artifacts/TestSpecification
<b>Main StakeHolder (Owner):</b>
Tester
<b>Created By:</b>
<ul style="list-style-type: none"> <li>Model Test Behaviour, see Table 25</li> </ul>
<b>Used By:</b>
<ul style="list-style-type: none"> <li>Generate Tests, see Table 24</li> </ul>

Table 15 Model Test Specification

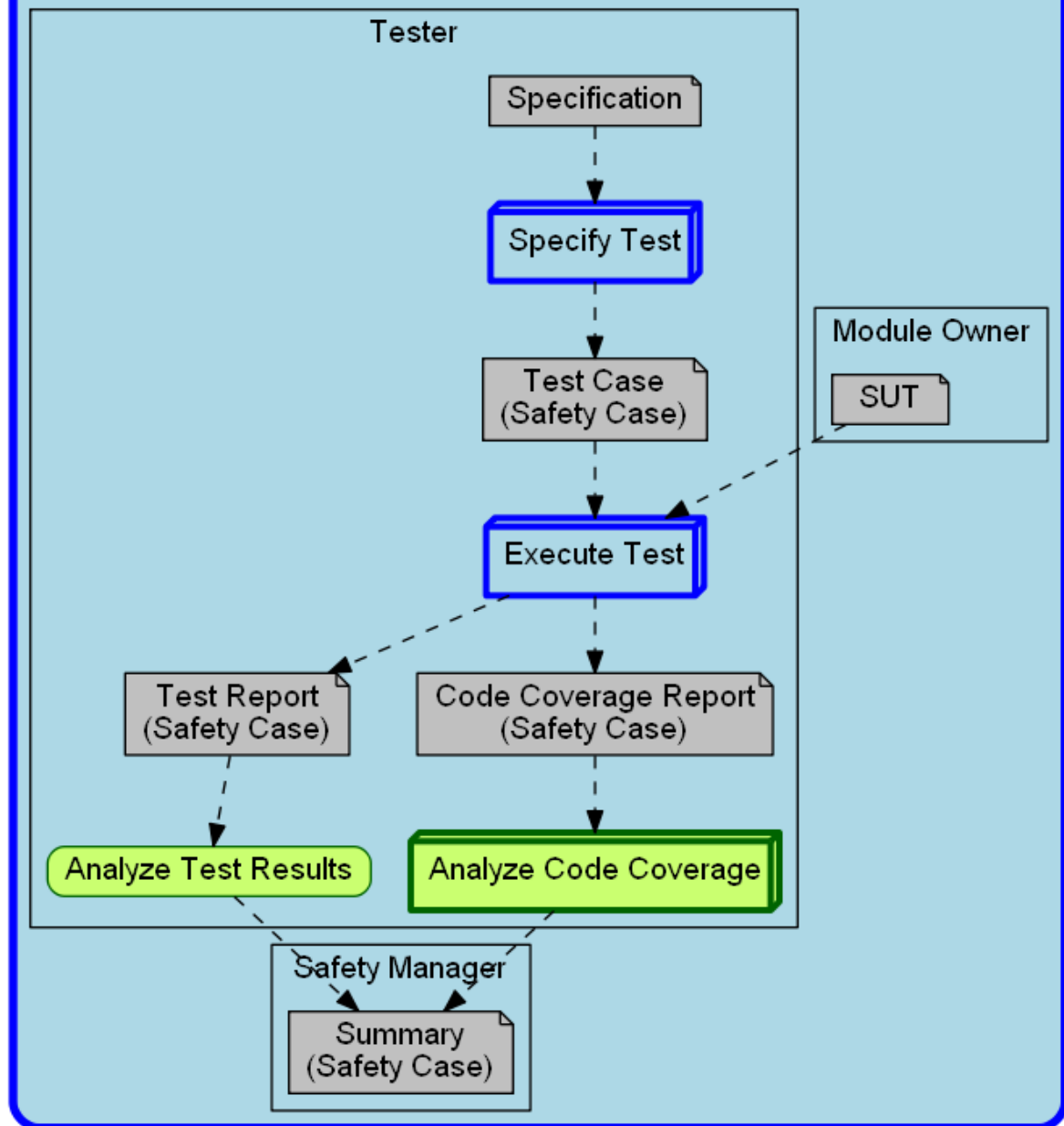
## 6 Process Perform Module Test

This section describes the 11 process modules for Perform Module Test.

### ProcessModule: Perform Module Test

#### View of Perform Module Test

## Perform Module Test



### Name:

Perform Module Test

### Description:

All activities for performing the module tests (currently only code coverage). In this example only the coverage measurement is considered (most complex & interesting).

### Qualified Name:

### Relevant Parameter:

- MODULE = <to be defined>, see Table 4

### Owner:

Tester

### Sub-Processes:

- Analyze Code Coverage, see Table 17
- Analyze Test Results, see Table 19
- Execute Test, see Table 20
- Specify Test, see Table 23

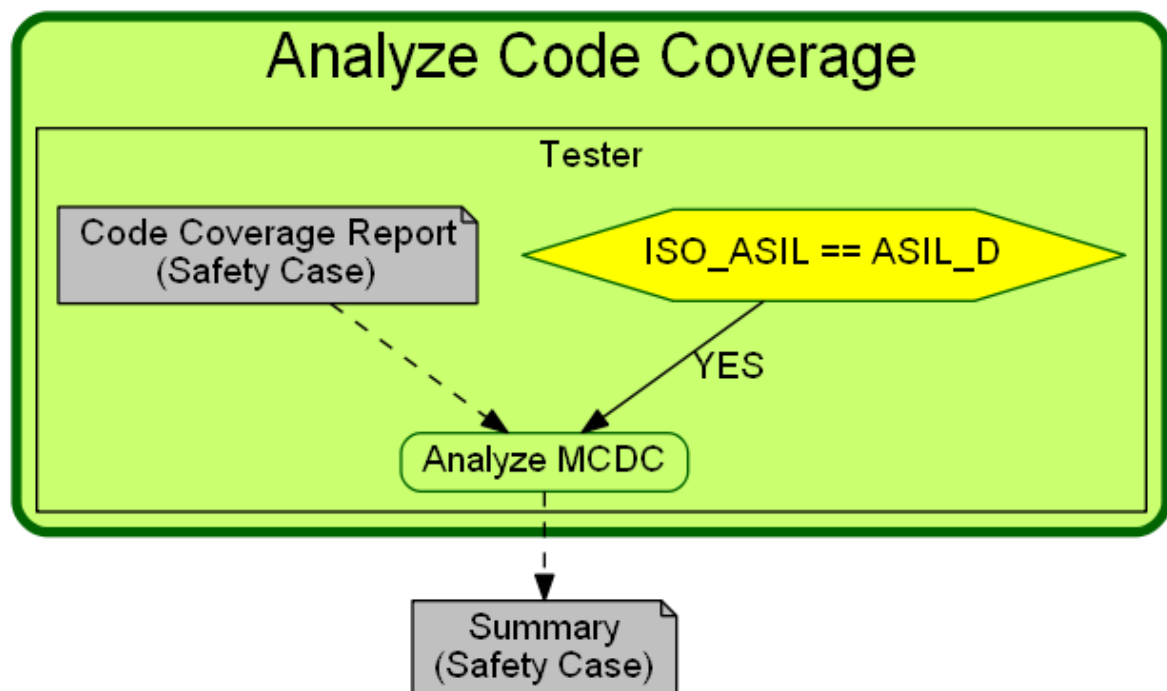
**Table 16 ProcessModule: Perform Module Test**

## 6.1 VerificationModule Analyze Code Coverage

This section describes the process module for Analyze Code Coverage

### VerificationModule: Analyze Code Coverage

#### View of Analyze Code Coverage



#### Name:

Analyze Code Coverage

#### Description:

The code coverage is checked for 100%. The coverage metrics are determined by the ASIL in the corresponding sub-processes.

#### Qualified Name:

Analyze Code Coverage

#### Relevant Parameter:

- MODULE = <to be defined>, see Table 4

#### Implicit/Automatically:

True

#### Owner (Inherited):

Tester

#### Verifies:

- Artifact: Code Coverage Report, see Table 10

#### Sub-Process:

- Analyze MCDC, see Table 18

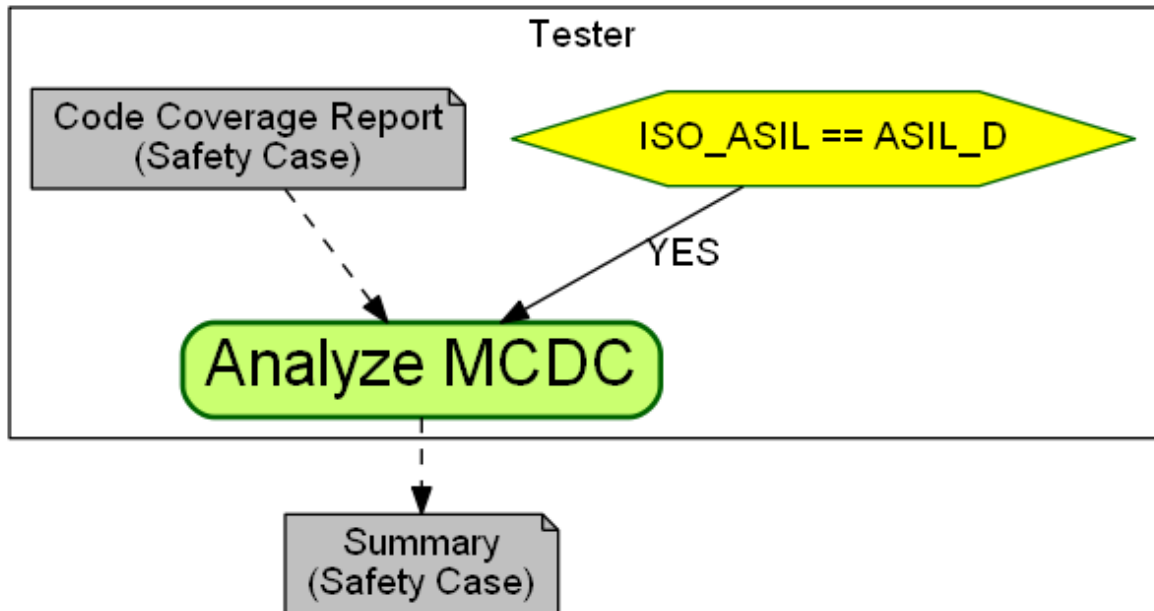
Table 17 VerificationModule: Analyze Code Coverage

### 6.1.1 VerificationModule Analyze MCDC

This section describes the process module for Analyze MCDC

#### VerificationModule: Analyze MCDC

##### View of Analyze MCDC



##### Name:

Analyze MCDC

##### Description:

Analyze if MCDC is 100% or if there are explained reasons why the coverage is <100%.

##### Qualified Name:

Analyze Code Coverage.Analyze MCDC

##### Relevant Parameter:

- MODULE = <to be defined>, see Table 4

##### Variant Condition:

- (ISO\_ASIL == ASIL\_D)

##### Criteria:

- MCDCE OK [AnaCCR-MC-C1]: Is the MCDC 100% (or are reasonable explanations given in case it is less than 100%)?
- All Conditions Considered [AnaCCR-MC-C2]: Have all conditions of the module been considered / instrumented (or are some files not instrumented that belong to the module)?

##### Compliance:

- 6.9.4.4-9.1c Compliance with MCDC

##### Owner (Inherited):

Tester

##### Verifies:

- Artifact: Code Coverage Report, see Table 10

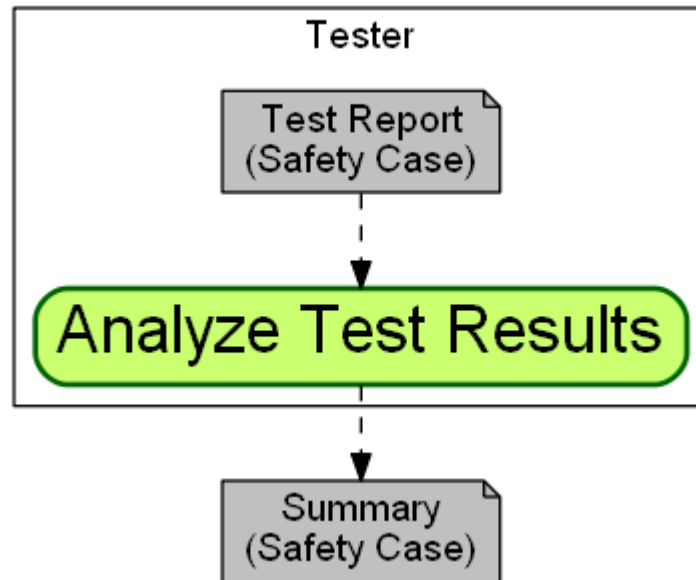
Table 18 VerificationModule: Analyze MCDC

## 6.2 VerificationModule Analyze Test Results

This section describes the process module for Analyze Test Results

### VerificationModule: Analyze Test Results

#### View of Analyze Test Results

**Name:**

Analyze Test Results

**Description:**

Checks the test results (There should not be surprises and all expected tests should be OK). It also contributes to the coveral sumary statement on the safety of the tested module(s).

**Comment:**

Analysis of test results is done as part of the process (and it is required for unit tests), but it is not required for coverage (6.9.4.4)

**Qualified Name:**

Analyze Test Results

**Relevant Parameter:**

- MODULE = <to be defined>, see Table 4

**Criteria:**

- Test Results Complete [AnaTR-C1]:Are the test results complete, i.e. have all required tests been executed?
- Test Results OK [AnaTR-C2]:Are the tests results OK, i.e. PASS or only failing/error for known bugs?

**Owner (Inherited):**

Tester

**Verifies:**

- VerificationModule: Analyze Test Results, see Table 19
- Artifact: Test Report, see Table 13

**VerifiedBy:**

- Analyze Test Results, see Table 19



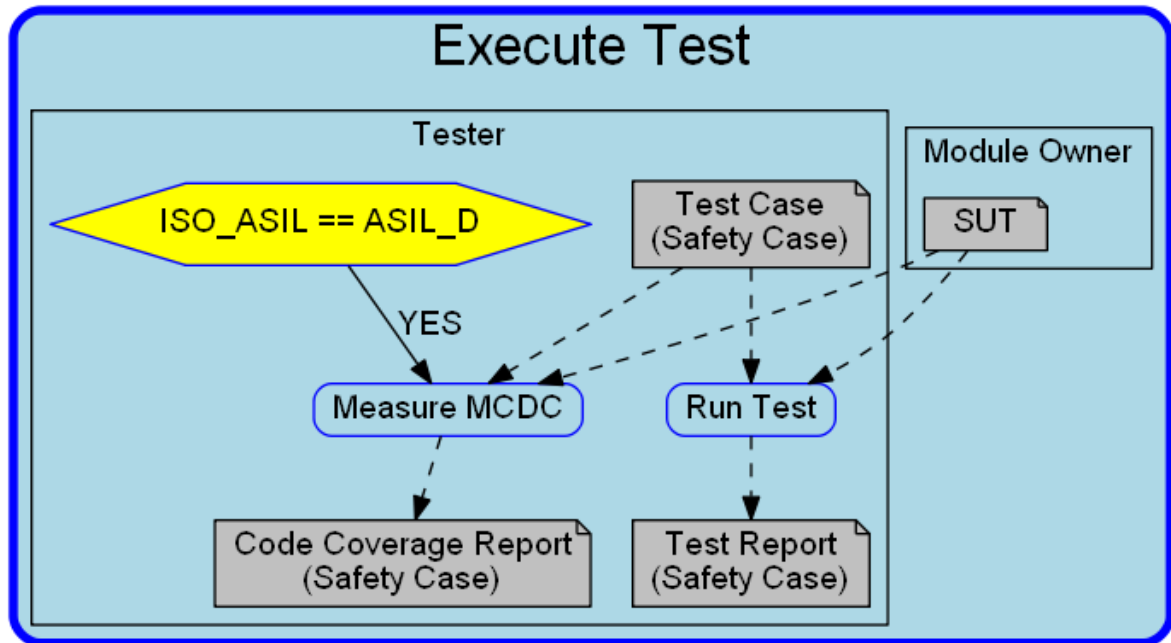
Table 19 VerificationModule: Analyze Test Results

### 6.3 ProcessModule Execute Test

This section describes the process module for Execute Test

#### ProcessModule: Execute Test

##### View of Execute Test



##### Name:

Execute Test

##### Description:

Run the test cases and create a report with test results.

##### Comment:

Test execution of modules is done using any test execution environment allowing to execute the tests on the target.  
 In addition the test coverage in the source code is measured, which is modeled as sub-processes.  
 There are the following three sub-processes available

- Statement coverage (for ASIL A)
- Branch coverage (for ASIL B and C)
- MCDC (for ASIL D)

Note that not all sub-processes are executed at once, but only the relevant depending on the ASIL.  
 This is modelled using variant terms (as in the corresponding requirements)

##### Qualified Name:

Execute Test

##### Relevant Parameter:

- MODULE = <to be defined>, see Table 4

##### Owner (Inherited):

Tester

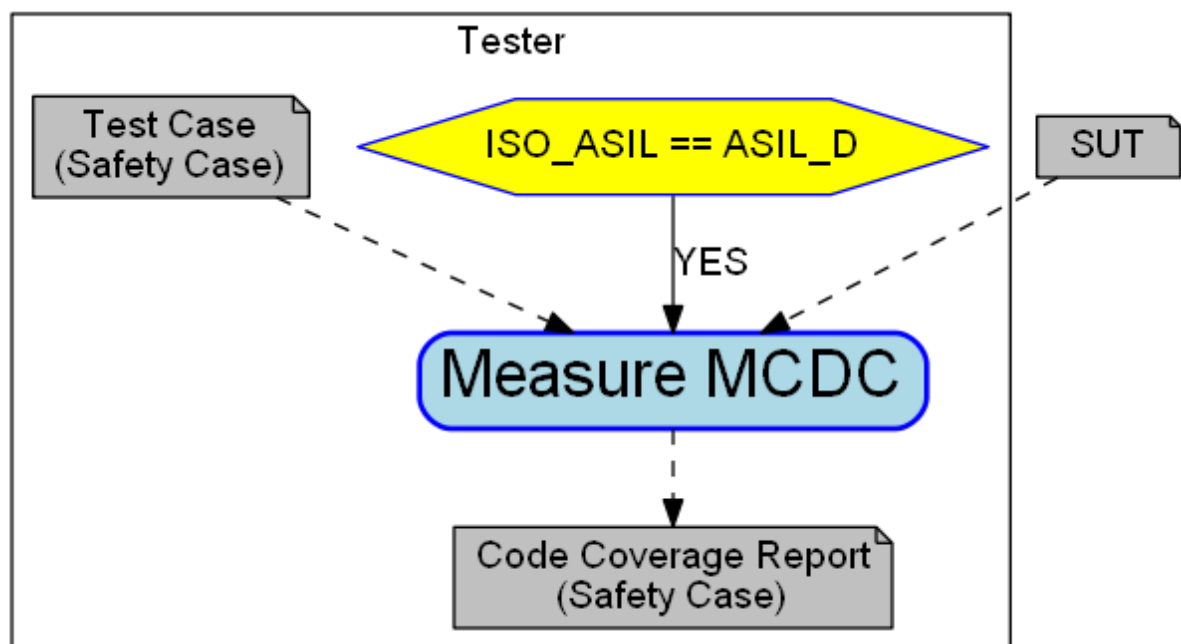
**Sub-Processes:**

- Measure MCDC, see Table 21
- Run Test, see Table 22

Table 20 ProcessModule: Execute Test

**6.3.1 ProcessModule Measure MCDC**

This section describes the process module for Measure MCDC

**ProcessModule: Measure MCDC****View of Measure MCDC****Name:**

Measure MCDC

**Description:**

Perform the tests and measure modified condition/decision coverage.

**Qualified Name:**

Execute Test.Measure MCDC

**Relevant Parameter:**

- MODULE = <to be defined>, see Table 4

**Variant Condition:**

- (ISO\_ASIL == ASIL\_D)

**Owner (Inherited):**

Tester

Table 21 ProcessModule: Measure MCDC

**6.3.2 ProcessModule Run Test**

This section describes the process module for Run Test

**ProcessModule: Run Test**

View of Run Test	
<pre> graph TD     subgraph Tester         TC[Test Case&lt;br/&gt;(Safety Case)]         RT[Run Test]         TR[Test Report&lt;br/&gt;(Safety Case)]         TC -.-&gt; RT         RT -.-&gt; TR     end     SUT[SUT] -.-&gt; RT </pre>	
<b>Name:</b>	
	Run Test
<b>Description:</b>	
	Executes test case and produces test result (contained in test report).
<b>Qualified Name:</b>	
	Execute Test.Run Test
<b>Relevant Parameter:</b>	
	<ul style="list-style-type: none"> <li>MODULE = &lt;to be defined&gt;, see Table 4</li> </ul>
<b>Compliances:</b>	
	<ul style="list-style-type: none"> <li>6.9.4.4-9.1b Compliance with Branch Coverage</li> <li>6.9.4.4-9.1c Compliance with MCDC</li> </ul>
<b>Owner (Inherited):</b>	
	Tester

Table 22 ProcessModule: Run Test

## 6.4 ProcessModule Specify Test

This section describes the process module for Specify Test

ProcessModule: Specify Test
View of Specify Test

<div data-bbox="442 215 1189 1200"> <h2 style="text-align: center;">Specify Test</h2> <pre> graph TD     subgraph Tester         direction TB         Spec[Specification] -.-&gt; MTB(Model Test Behaviour)         MTB -.-&gt; TS[Test Specification]         TS -.-&gt; GT[Generate Tests]         GT -.-&gt; TC[Test Case&lt;br/&gt;(Safety Case)]         TC -.-&gt; VT[Validate Tests]         VT -.-&gt; GT     end </pre> </div>	
<b>Name:</b>	Specify Test
<b>Description:</b>	Create a model based test specification and generate tests.
<b>Qualified Name:</b>	Specify Test
<b>Relevant Parameter:</b>	<ul style="list-style-type: none"> <li>MODULE = &lt;to be defined&gt;, see Table 4</li> </ul>
<b>Compliances:</b>	<ul style="list-style-type: none"> <li>6.9.4.4-9.1b Compliance with Branch Coverage</li> <li>6.9.4.4-9.1c Compliance with MCDC</li> </ul>
<b>Owner:</b>	Tester
<b>Sub-Processes:</b>	<ul style="list-style-type: none"> <li>Generate Tests, see Table 24</li> <li>Model Test Behaviour, see Table 25</li> <li>Validate Tests, see Table 26</li> </ul>

Table 23 ProcessModule: Specify Test

### 6.4.1 ProcessModule Generate Tests

This section describes the process module for Generate Tests

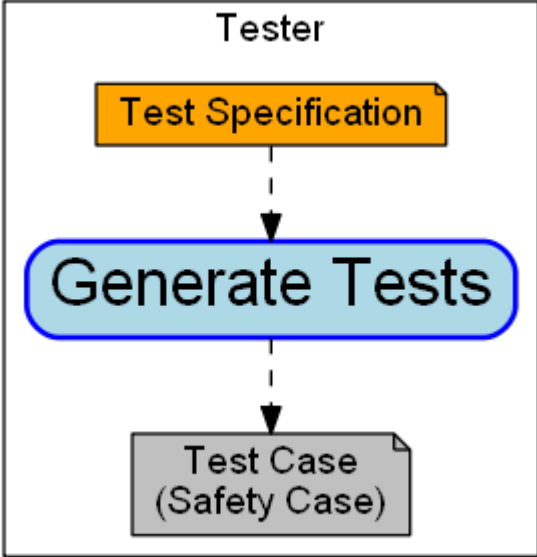
ProcessModule: Generate Tests	
View of Generate Tests	
	
<b>Name:</b>	Generate Tests
<b>Description:</b>	Generate tests cases from the model based test specification using test genertaors, e.g. LTTG or ForeC++.
<b>Qualified Name:</b>	Specify Test.Generate Tests
<b>Relevant Parameter:</b>	<ul style="list-style-type: none"><li>• MODULE = &lt;to be defined&gt;, see Table 4</li></ul>
<b>Owner (Inherited):</b>	Tester

Table 24 ProcessModule: Generate Tests

### 6.4.2 ProcessModule Model Test Behaviour

This section describes the process module for Model Test Behaviour

ProcessModule: Model Test Behaviour	
View of Model Test Behaviour	

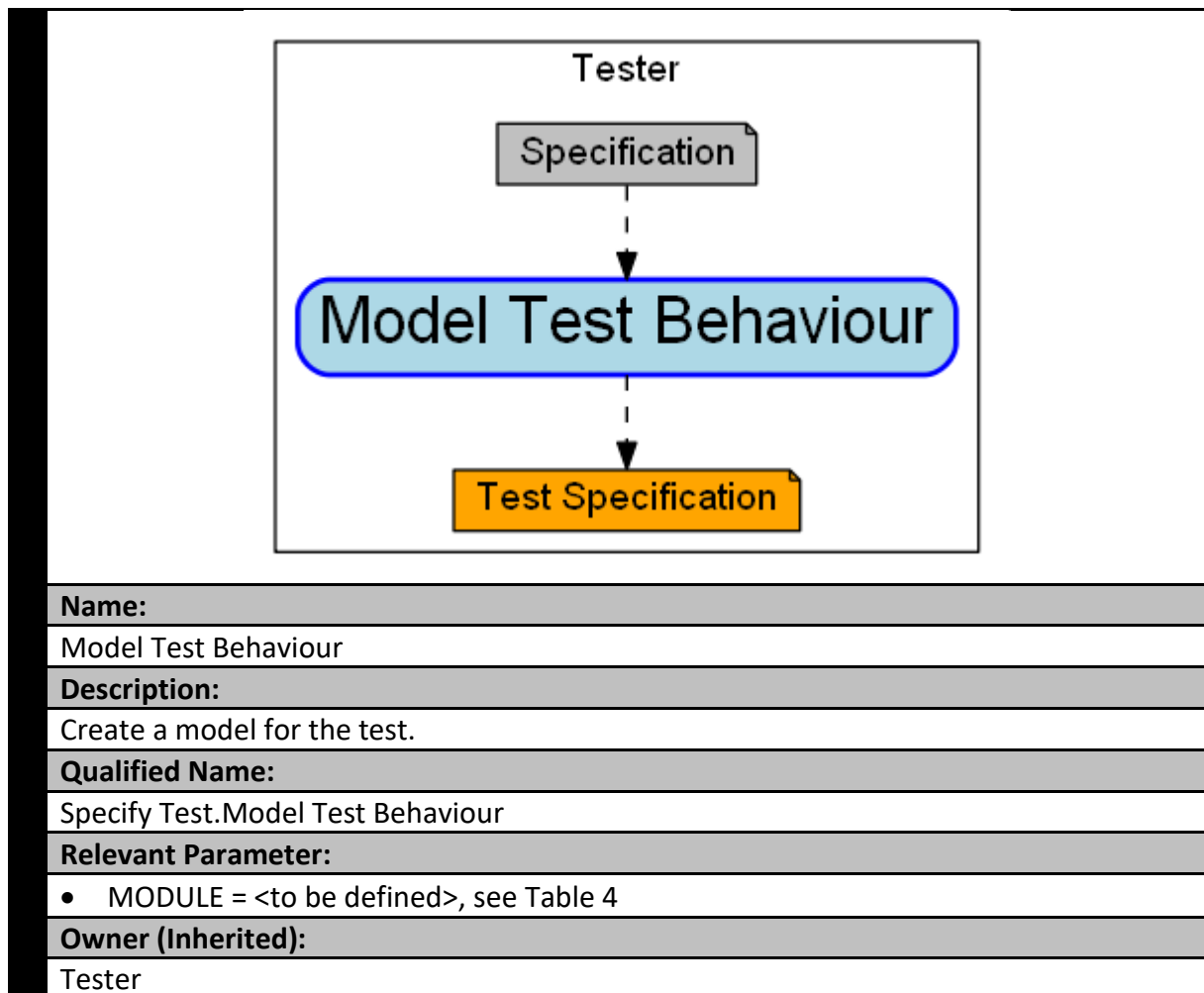
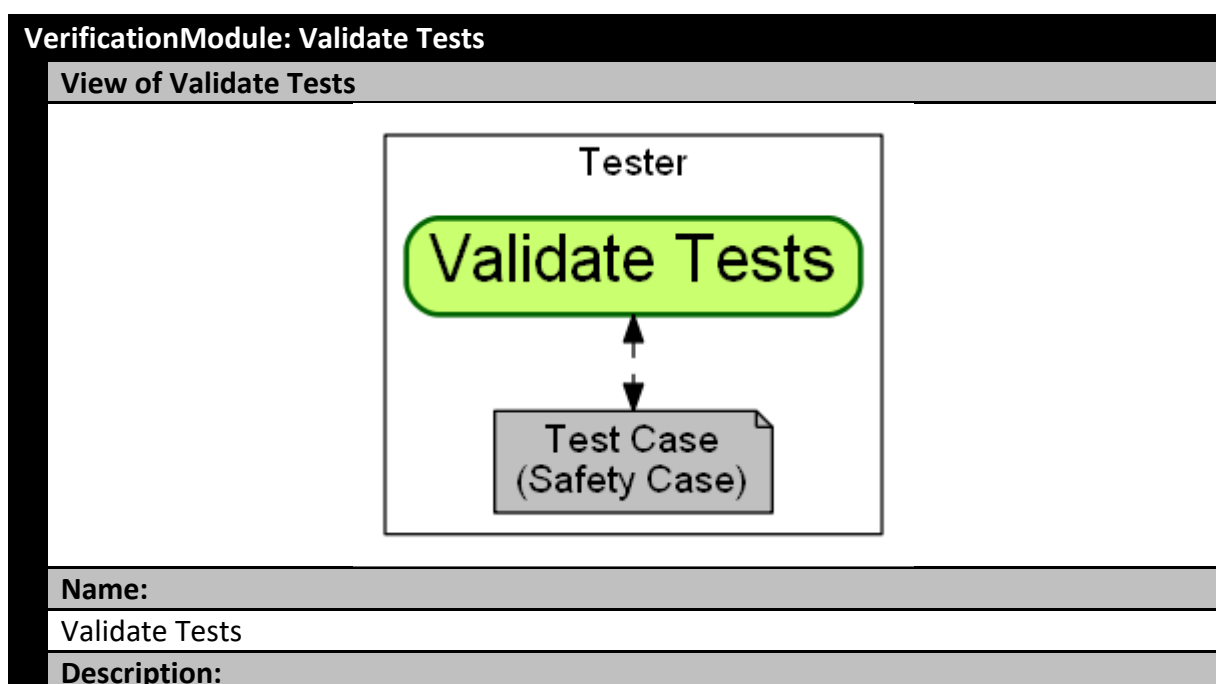


Table 25 ProcessModule: Model Test Behaviour

### 6.4.3 VerificationModule Validate Tests

This section describes the process module for Validate Tests



Tests can be validated in many ways, typically by compiling them, eventually by testing against a standard implementations from open source or reviewing them. Validas.
<b>Qualified Name:</b>
Specify Test.Validate Tests
<b>Relevant Parameter:</b>
<ul style="list-style-type: none"> <li>MODULE = &lt;to be defined&gt;, see Table 4</li> </ul>
<b>Criteria:</b>
<ul style="list-style-type: none"> <li>Test Complete [VT-C1]:Does the test cover the requirement completly?</li> <li>Test Effective [VT-C2]:Can the test detect dviations (failures/errors) in case they would occur, e.g. by using some statements like "assert"?</li> </ul>
<b>Compliances:</b>
<ul style="list-style-type: none"> <li>6.9.4.4-9.1b Compliance with Branch Coverage</li> <li>6.9.4.4-9.1c Compliance with MCDC</li> </ul>
<b>Owner (Inherited):</b>
Tester
<b>Verifies:</b>
<ul style="list-style-type: none"> <li>Artifact: Test Case, see Table 12</li> </ul>

**Table 26 VerificationModule: Validate Tests**

## 7 Tools in Perform Module Test

This section describes the involved tools and their supported processes and preliminary classification in Perform Module Test.

## 8 Glossary

The following abbreviations are used in the document. More information on the concepts & processes can be found in [QMeth].

- AOC: Anomalous Operating Condition
- Artifact: Element exchanged between processes
- CR: Compliance Report<sup>2</sup>
- CT: Construction Task (during QKit creation)
- KB: Known Bug
- LCR: Library Classification Report
- LQP: Library Qualification Plan
- LQR: Library Qualification Report
- LSM: Library Safety Manual
- LTG: Library Test Generator
- PCCP: (Development) Process Compliance Check Plan
- PCCR: (Development) Process Compliance Check Report
- PMT: Process Modeling Tool
- Process Module: modular tasks in the process
- PT: Preparation Task (before QKit creation)
- Role: see Stakeholder
- QKit: Qualification Kit
- QP: Qualification Plan (general), can be LQP or TQP
- QR: Qualification Report (general), can be LQR or TQR
- QST: Qualification Support Tool
- SEOOO: Safety Element Out Of Context according to [ISO26262]
- SM: Safety Manual (general), can be LSM or TSM
- Stakeholder: abstract person taking over responsibilities in the process
- SWC: Software Component, e.g. a library<sup>3</sup>
- TAU: Test Automation Unit
- TCA: Tool Chain Analyzer
- TD: Tool Detection (part of TCL computation according to [ISO26262])
- TCL: Tool Confidence Level (according to [ISO26262])
- TCR: Tool Classification Report
- TI: Tool Impact (part of TCL computation according to [ISO26262])
- TQL: Tool Qualification Level (according to [DO330])
- TQP: Tool Qualification Plan
- TQR: Tool Qualification Report
- TSM: Tool Safety Manual
- V&V: Verification and Validation
- Verification Module: special form of Process module used to verify an artifact in the process
- VVP: Verification and Validation Plan
- VVR: Verification and Validation Report

---

<sup>2</sup> Do not confuse with Classification Reports LCR and TCR.

<sup>3</sup> Note that libraries can be both changes and unchanged software components.



- VVT: Verification and Validation Tool
- VT: Verification task (after QKit creation)

## 9 References

[BPMN] Business Process Modeling Notation, see <https://www.omg.org/spec/BPMN/2.0/>

[CR] Compliance Report for Perform Module Test, generated by PMT

[DO330] RTCA. DO-330: Software Tool Qualification Considerations 1st Edition 2011-12-13.

[DO178C] RTCA. DO-178C: Software Considerations in Airborne Systems and Equipment Certification, 2011-12-13.

[EN50128] BS EN 50128:2011, Railway applications — Communication, signaling and processing systems — Software for railway control and protection systems, BSI Standards Publication

[FDA2002] General Principles of Software Validation; Final Guidance for Industry and FDA Staff, Jan 2002, from <http://www.fda.gov/downloads/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/ucm085371.pdf>

[FDA\_OTS] Guidance for Industry, FDA Reviewers and Compliance on Off-The-Shelf Software Use in Medical Devices, Center for Devices and Radiological Health (CDRH), from <http://www.fda.gov/downloads/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/ucm073779.pdf>

[IEC61508] International Electrotechnical Commission, IEC 61508, Functional safety of electrical/electronic/programmable electronic safety-related systems, Edition 2.0, Apr 2010.

[IEC62304] International Electrotechnical Commission, IEC 62304, Medical device software – Software life cycle processes

[ISO26262] International Organization for Standardization, ISO 26262 Road Vehicles – Functional safety– 2st Edition, 2018-12.

[PR] Process Report for Perform Module Test, generated by PMT

[QMeth] Validas Qualification Method, White Paper, Version 1.7, see <Documentation>/QualificationMethodology.pdf.

[VVP] Verification and Validation Plan (Model) for Perform Module Test, generated by PMT

[VVR] Project specific Verification and Validation Report for <Product\_Name>, to be created by performing and documenting [VVP]