SW Module Test

**Change History**

| **Version** | **Date** | **Change Description / Reason** | **Author** |
| --- | --- | --- | --- |
| 1.0 | 2017-12-08 | Initial version based on Hella Process (State Avenue)  Delete integration related tasks. | Hae-Min, Woo |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[1. Purpose Description 5](#_Toc496001250)

[1.1 Purpose 5](#_Toc496001251)

[1.2 Main Description 5](#_Toc496001252)

[1.3 Result of Process 5](#_Toc496001253)

[2. Major Roles Acting in this Process 5](#_Toc496001254)

[2.1 Refinement of the major roles 6](#_Toc496001255)

[3. Process 6](#_Toc496001256)

[3.1 Process Input – Output Definition 6](#_Toc496001257)

[3.2 Introducing Aspects 7](#_Toc496001258)

[3.3 Phase Model 7](#_Toc496001259)

[3.4 Process Flow Chart 8](#_Toc496001260)

[3.5 Test Planning Phase 10](#_Toc496001261)

[3.5.1 Task 1: Determine framework conditions 10](#_Toc496001262)

[3.5.2 Task 2: Determine content of test 10](#_Toc496001263)

[3.5.3 Task 3: Develop test strategy 11](#_Toc496001264)

[3.5.4 Task 4: Organize test project 14](#_Toc496001265)

[3.5.5 Task 5: Review and release test plan 15](#_Toc496001266)

[3.6 Test Control Phase 15](#_Toc496001267)

[3.6.1 Task 6: Manage test team and project 15](#_Toc496001268)

[3.6.2 Task 7: Monitor and communicate test progress 16](#_Toc496001269)

[3.7 Test Infrastructure Phase 16](#_Toc496001270)

[3.7.1 Task 8: Specify infrastructure 16](#_Toc496001271)

[3.7.2 Task 9: Realize infrastructure 17](#_Toc496001272)

[3.7.3 Task 10: Putting infrastructure into operation 17](#_Toc496001273)

[3.8 Test Analysis and Design Phase 17](#_Toc496001274)

[3.8.1 Task 11: Analyze test basis 17](#_Toc496001275)

[3.8.2 Task 12: Specify abstract test cases 18](#_Toc496001276)

[3.8.3 Task 13: Specify concrete test cases 19](#_Toc496001277)

[3.8.4 Task 14: Ensure traceability and consistency 19](#_Toc496001278)

[3.8.5 Task 15: Review and release test specification 19](#_Toc496001279)

[3.9 Test Implementation and Execution Phase 20](#_Toc496001280)

[3.9.1 Task 16: Implement test automation 20](#_Toc496001281)

[3.9.2 Task 17: Set up test execution 20](#_Toc496001282)

[3.9.3 Task 18: Execute tests 20](#_Toc496001283)

[3.9.4 Decision 1: Completed test run? 21](#_Toc496001284)

[3.10 Test Evaluation and Report Phase 21](#_Toc496001285)

[3.10.1 Task 19: Analyze test results 21](#_Toc496001286)

[3.10.2 Task 20: Review test results 22](#_Toc496001287)

[3.10.3 Task 21: Assess and communicate maturity of test object 22](#_Toc496001288)

[3.11 Test Closure Phase 22](#_Toc496001289)

[3.11.1 Task 22: Analyze test experience 23](#_Toc496001290)

[3.11.2 Task 23: Archiving test environment 23](#_Toc496001291)

[4. Changes, References, Appendix, Terms 23](#_Toc496001292)

[4.1 References 23](#_Toc496001293)

[4.2 Template 23](#_Toc496001294)

# Purpose Description

## Purpose

The purpose of the software module test process is to verify, that the software modules properly reflect the software specification and design.

Proof for a software module is provided independent of other software modules.

## Main Description

The software module test is the first level of dynamic testing after the software implementation. Focus of the software module test level is

* Functional test (functionality is implemented as specified)
* Robustness against unexpected values (invalid values, not specified values)
* Reproducible and deterministic behavior
* Parameter impact (functional impact of single parameter as well as parameter combination)
* Run time performance (of specific time critical modules)

The following issues can be detected:

* Functional failures within the software module
* Runtime failures
* (local) Performance problems
* Robustness problems
* Failures caused by parameter settings

## Result of Process

As a result of successful implementation of this software module test process:

* A software module verification strategy is defined;
* Consistency and bilateral traceability are established between detailed design and test specification module test including test cases
* Verification of the software modules is performed according to the module verification strategy and
* Results of the software module verification are recorded.

NOTE: Software module verification will include software module testing and may include static analysis, code inspection/reviews, checks against coding standards and guidelines, and other techniques.

# Major Roles Acting in this Process

The following table gives an overview of the major roles involved with a short explanation. The definition of the roles can be found in the supplement A of MHE-PE-01.

|  |  |
| --- | --- |
| **Role** | **Contribution and Responsibilities** |
| Test Manager | Plans, coordinates, and controls all test activities across the test levels and creates the Master Test Plan (8316) and the SW Module Test Plans (8317). In large projects, this role should be executed by a team of “Domain” Test Managers, with one member acting as overall responsible. If there is no explicit Test Manager in a project (or in a domain in case of a test manager team), then the tasks of this role must be taken over by another member of the team and the corresponding project manager is responsible for decisions. |
| SW Module Tester | Analyzes the test basis, creates the test specification, the test scripts, and carries out the test including the corresponding documentation. |
| SW Project Manager | In this context relevant for all software development aspects like integration, SW module classification, SW schedule, etc. |
| Safety Manager | In case of safety related system responsible to support the test team, review / decide work products concerning their safety related content. |

Due to the generic character of this process description, the assignments of the generic roles to the test level specific roles are defined at cf. 5.2 Mapping of roles. Additionally, it is defined there, who may take over the role of the Test Manager at each respective test level. Should the Test Manager role be assumed by another role in personal union, then this must be documented in the (SW or HW) Project Manual (8314 or 8622).

## Refinement of the major roles

The following role refinement is made locally in this process description to improve the mapping into this process description of the specific work focuses and skills of the “SW Module Tester” role (MHE-PE-01,

Supplement A):

|  |  |
| --- | --- |
| **Role** | **Contribution and Responsibilities** |
| SW Module Test Designer | Tester with very good experiences with test design techniques and extensive knowledge of the test object responsible for designing the test cases (creates the test specification). |
| SW Module Tester | Carries out the tests as defined in the test specification and draws up the reports including measurements. |
| SW Module Test Automation Engineer | Developer with know-how to automate test activities; implementation of modular, maintainable test automation solutions, Tester dealing with the provision, version assignment, archiving, and administration of the necessary test environment. |

# Process

## Process Input – Output Definition

The following process input is required:

|  |  |
| --- | --- |
| **Process input** | **From supplier** |
| Master Test Plan (especially chapter 4 “Master test strategy”, if implemented) | Test Manager |
| Safety Plan (in case of safety related system) | Safety Manager |
| SW Architecture (e.g. Innovator SD model) | SW Architect |
| SW Design (Definition of all SW modules and interfaces) | SW Architect,  SW Developer |
| SW Module and Unit Design | SW Developer |
| Attributes for economic risk classification on software module level  (former Module Classes and Test Depth) | SW Architect |
| Release Plan | SW Project Manager,  Project Manager |
| Released SW Modules | SW Developer |

The following process output is produced:

|  |  |
| --- | --- |
| **Process output** | **To customer** |
| SW Module Test Plan | SW Project Manager  Safety Manager |
| Review Checklist SW Module Test Plan | SW Module Test Designer |
| SW Module Test Specification | SW Module Tester |
| Review Checklist SW Module Test Specification | SW Module Test Designer |
| SW Module Test Report | SW Project Manager  Safety Manager |
| Tested SW Modules | SW Integrator |
| Test Completion Report | SW Project Manager |

## Introducing Aspects

None

## Phase Model

Following the fundamental test process the process description distinguishes five phases: test planning and controlling, test analysis and design, test implementation and execution, test evaluation and report and test closure.



Figure 1 : Phase model of the test process

## Process Flow Chart





## Test Planning Phase

The objective of the first phase Test planning is to plan the test activities for a software module test level and document the planning decisions like how, by whom, what with and when what test activities are carried out in the SW Module Test Plan.

### Task 1: Determine framework conditions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | Master Test Plan  Safety Plan | | | | | |
| Output | | SW Module Test Plan | | | | | |
| D: |  | E: | Test Manager | S: | SW PM,  Safety Manager | I: |  |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

The aim of this task is to define the framework conditions and influencing factors for the test project from the stakeholder point of view. This includes project-specific particulars like release milestones and objectives for releases as well as test object project-specific particulars. It also includes the scope of test content in demarcation with the tests executed by the customer.

For safety related systems, the Safety Plan as well as the Guidelines to apply test methods conform to ISO26262 must be considered to determine the specific framework conditions.

The result of this task must be documented in the SW Module Test Plan, chapter 2: Introduction.

### Task 2: Determine content of test

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | Input documents | | | | | |
| Output | | SW Module Test Plan | | | | | |
| D: | Test Manager | E: | SW Module Test  Designer | S: | SW PM,  Architect, Analyst or  Developer,  Safety Manager | I: |  |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

The aim of this task is to determine the test content of the test of this test (level) from the test object point of view. This includes analyzing the project-specific particulars, to determine the reference documents as well as the test object, and to define the scope of testing, in terms of “Features to be tested / Features not to be tested.

The test planning shall determine the content of the work products to be tested. This is done in this process task by

* Defining the test object of this test level
* Analyzing the project- and test object-specific particularities
* Determining and referencing the specification documents

The following information shall be available and referenced in detail:

* Hardware-software interface specification and software module design specification shall be determined and referenced as specification documents and shall be used as input for the creation of test cases
* Safety Plan shall be determined and referenced as project-specific particularities and shall be used as input for the test strategy
* Software module implementation shall be defined as the test object(s) of this test level (For model-based software development, the corresponding parts of the implementation model also represent objects for the test planning. Depending on the selected software development process the test objects can be the code derived from this model or the model itself.)

The software module testing shall be applied to demonstrate that the software modules achieve:

* Compliance with the software module design specification
* Compliance with the specification of the hardware-software interface
* The specified functionality
* Confidence in the absence of unintended functionality
* Robustness (e.g. the absence of inaccessible software, the effectiveness of error detection and error handling mechanisms.)
* Sufficient resources to support their functionality

These topics are defined as scope of testing of the software module test level within this task and considered in the following process steps.

For safety related systems the specific content of test must be determined

The result of this task must be documented in the SW Module Test Plan, chapter 3: Test objects.

### Task 3: Develop test strategy

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | Master Test Plan  Safety Plan | | | | | |
| Output | | SW Module Test Plan | | | | | |
| D: |  | E: | Test Manager | S: | SW Module Test  Designer,  Safety Manager | I: |  |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

The objective of the SW Module test strategy is to find errors as early as possible, with as little cost as possible, and to test the test object with an appropriate breadth and depth which achieves the desired quality and compliance to the requirements over the complete range of allowed application parameter combinations (for non integration test level) or which verify that only allowed application parameter combinations can influence functionality and that wrong parameter or parameter combinations are failsafe (for integration test level). The test strategy has to be consistent with the release strategy.

In case of an integration test level the test strategy has to identify test steps according to the order of integration defined in the integration strategy.

Based on a suitable granularity of the shares of the test object and on already defined attributes (“Extent of economic damage” caused by implemented defects, the “Implementation probability of defects”, and the “Frequency of use”, e.g. documented using form MHE 8325), the Test Manager has to determine the Economic risk classes of these shares before determining the test strategy.

The test strategy includes the definition of

* Suitable methods and techniques to be used and their coverage rates, based on the results of a risk assessment.
* Criteria for test exit, test interruption and resumption.
* A strategy concerning test automation
* A re-test and regression test /verification strategy

For safety related systems, the Guidelines to apply test methods conform to ISO26262 must be considered to determine the reasonable test strategy. The tests of the safety related functions are to prioritize appropriately high in the case of conflicts regarding a complete execution of the planned tests.

The test strategy should be developed according to priorities and categorization (based on the economic risk classification, safety classification as well as release relation) of the requirements.

The verification specification (spanned over the different verification level) shall select and specify the methods to be used for the verification, and shall include:

1. Review or analysis checklists; or
2. Simulation scenarios; or
3. Test cases, test data and test objects.

Although the test process is responsible for the planning and execution of the 3rd topic, the development of the test strategy considers the planned verification activities of the development phase (1st and 2nd topic) while determining the necessary breadth and depth of testing to ensure an appropriate overall verification depth.

The test activities shall prove, that the safety requirements are fulfilled, while the software is executed on the target processor, compiled with the target compiler and the final compiler configuration. In the context of the master test strategy must be determined to which degree tests on the target processor (including target compiler, final compiler configuration) are necessary on module test level and to which degree the target related verification is sufficient on the other test levels, which uses this execution setting in general. As a result the master test strategy has to define targets for the module test strategy, which have to be taken into account correspondingly in the module test strategy.

The following information shall be used for the determination of the test strategy:

* The content of the Safety Plan determined for the software module testing shall integrated in this test strategy. This content shall contain
* Methods for software module testing
* Methods for deriving test cases for software module testing
* Structural coverage metrics at the software module level

The software module testing methods listed in Table 1 shall be applied to demonstrate that the software modules achieve the scope of test content defined in:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Methods | | ASIL | | | |
| A | B | C | D |
| 1a | Requirements-based test a | ++ | ++ | ++ | ++ |
| 1b | Interface test | ++ | ++ | ++ | ++ |
| 1c | Fault injection test b | + | + | + | ++ |
| 1d | Resource usage test c | + | + | + | ++ |
| 1e | Back-to-back comparison test between model and code, if applicable d | + | + | ++ | ++ |
| a The software requirements at the module level are the basis for this requirements-based test.  b This includes injection of arbitrary faults (e.g. by corrupting values of variables, by introducing code mutations, or by corrupting values of CPU registers)  c Some aspects of the resource usage test can only be evaluated properly when the software module tests are executed on the target hardware or if the emulator for the target processor supports resource usage tests.  d This method requires a model that can simulate the functionality of the software modules. Here, the model and code are stimulated in the same way and results compared with each other. | | | | | |

Table 1: Methods for software module testing

The selection of the method for software module testing or the criteria to define the selection of these methods for specific types of software modules must be defined and justified in the test strategy. Furthermore the omission of methods also must be justified and documented.

To enable the specification of appropriate test cases for the software module testing in accordance with Table 1: Methods for software module testing test cases shall be derived using the methods listed in Table 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Methods | | ASIL | | | |
| A | B | C | D |
| 1a | Analysis of requirements | ++ | ++ | ++ | ++ |
| 1b | Generation and analysis of equivalence classes a | + | ++ | ++ | ++ |
| 1c | Analysis of boundary values b | + | ++ | ++ | ++ |
| 1d | Error guessing (or Explorative testing) c | + | + | + | + |
| a Equivalence classes can be identified based on the division of inputs and outputs, such that a representative test value can be selected for each class.  b This method applies to interfaces, values approaching and crossing the boundaries and out of range values.  c Error guessing tests can be based on data collected through a “lessons learned” process and expert judgment. MHE remark: Alternative to Error guessing the method Explorative testing can be applied. | | | | | |

Table 2: Methods for deriving test cases for software module testing

The selection of the methods for deriving test cases for software module testing or the criteria to define the selection of these methods for specific types of software modules must be defined and justified in the test strategy. Furthermore the omission of methods also must be justified and documented.

To evaluate the completeness of test cases and to demonstrate that there is no unintended functionality, the coverage of requirements at the software module level shall be determined and the structural coverage shall be measured in accordance with the metrics listed in Table 3.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Methods | | ASIL | | | |
| A | B | C | D |
| 1a | Statement coverage | ++ | ++ | + | + |
| 1b | Branch coverage | + | ++ | ++ | ++ |
| 1c | MC/DC (Modified Condition/Decision Coverage) | + | + | + | ++ |

Table 3: Structural coverage metrics at the software module level

Note 1 The structural coverage can be determined by the use of appropriate software tools.

Note 2 In the case of model-based development, the analysis of structural coverage can be performed at the model level using analogous structural coverage metrics for models.

Note 3 If instrumented code is used to determine the degree of coverage, it can be necessary to show that the instrumentation has no effect on the test results. This can be done by repeating the tests with noninstrumented code.

The verification planning shall define the pass and fail criteria for testing. This is defined in this process task by

* Defining the test exit criteria

The verification planning shall define the actions to be taken if anomalies are detected. This is defined in this process task by

* Defining particularities of the incident handling

The verification planning shall define the actions to be taken after changes of the system. This is done in this process task by

* Defining the re- and regression test strategy

(A regression strategy specifies how verification is repeated after changes have been made to the item or element. Verification can be repeated fully or partially and can include other items or elements that might affect the results of the verification.)

The developed test strategy and regression test strategy are documented in the SW Module Test Plan, chapter 4: Test strategy.

### Task 4: Organize test project

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | |  | | | | | |
| Output | | SW Module Test Plan | | | | | |
| D: |  | E: | Test Manager | S: | SW PM, SW Module  Test Designer,  SW Module Tester,  SW Module Test  Automation Engineer,  Safety Manager | I: |  |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

The aim of this task is to define and to set up the internal project structure for the test activities. This includes:

* To define and establish test organization, including role assignments, trainings needs, team meetings and team communication
* To estimate the test effort and setup the temporal planning (incl. the resource assignment to work packages)
* To define approach to ensure bilateral traceability between the requirements and the corresponding test cases.
* Determining project specific procedures as “How to’s” like:  
  - assignment of requirements to test level  
  - to define particularities of the incident handling
* To determine the document planning like test deliverables,
* To define the necessary test environment and the demands on this environment
* To define the necessary tools
* To define how to manage, to supervise, and to report the test progress. This includes determining whether additional people, in addition to those already defined by default, must be informed about the test status. Furthermore metrics should be applied as control measure.

For safety related systems, the Safety Plan, the Guidelines to apply test methods conform to ISO26262, and specific time schedules must be considered to determine the organization of the test project.

Additionally specific monitoring and communication needs must be determined.

The test planning shall determine the test environment and the tools used for testing, if applicable. This is done in this process task by

* Defining the demands on test environment

The test environment for software module testing shall correspond as closely as possible to the target environment.

The gathering and reporting procedure concerning the structural coverage metrics at software module test level according to the recommended methods of Table 3 must be defined in this task.

The results are documented in the SW Module Test Plan, chapter 5: Test organization and chapter 6: Test environment / infrastructure.

### Task 5: Review and release test plan

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | Input documents | | | | | |
| Output | | SW Module Test Plan | | | | | |
| D: | Test Manager,  Safety Manager | E: | Test Manager | S: | SW PM,  SW Quality Planner,  SW Module Test  Designer , SW Module Tester,  Safety Manager | I: |  |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

The aim is to review and to release the SW Module Test Plan.

The result is documented in the Review Checklist SW Module Test Plan.

## Test Control Phase

As shown in Figure 1: Phase model of the test process the test control activities are running in parallel to the other phases of the test process. Control comprises the continuous monitoring of the test activities in comparison with the planning, the reporting of deviations or abortion, and the definition of counter measures in the case of deviations from the planned targets.

### Task 6: Manage test team and project

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | |  | | | | | |
| Output | | Schedule  Meeting Minutes  Review Master  PTC-IM Tasks | | | | | |
| D: |  | E: | Test Manager | S: | SW Module Tester,  SW Module Test  Designer,  SW Module Test  Automation Engineer | I: |  |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

The aim of this task is to manage the test team and the test project in general and to determine counter measures in the case of deviations.

The results are documented in dependence of the managed aspects or the counter measures.

### Task 7: Monitor and communicate test progress

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | |  | | | | | |
| Output | | Test Metrics | | | | | |
| D: |  | E: | Test Manager | S: | SW Module Tester,  SW Module Test  Designer, SW Module  Test Automation  Engineer | I: | SW PM,  Safety Manager |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

In this task, the test progress is assessed and test exit criteria are analyzed with regard to the actual test results. In contrast to “Task 19: Analyze test results”, the analysis of the test exit criteria is here characterized more by the Test Manager’s point of view like timing aspects and test budget.

The input of this task is the project schedule; test exit criteria defined in the SW Module Test Plan and the results logged in the SW Module Test Report.

The states of the test progress are documented in the Project Cockpit Chart (8386).

Permanent changes to the test exit criteria must be documented in the SW Module Test Plan; temporary changes in the Module Test Report.

The structural coverage metrics at software module test level according to the recommended methods and the definition how to map these methods to software modules from the SW Module Test Plan, must be gathered and reported in this task.

If the achieved structural coverage is considered insufficient, either additional test cases shall be specified or a rationale shall be provided.

Example 1 Analysis of structural coverage can reveal shortcomings in requirement-based test cases, inadequacies in requirements, dead code, deactivated code or unintended functionality.

Example 2 A rationale can be given for the level of coverage achieved based on accepted dead code (e.g. code for debugging) or code segments depending on different software configurations; or code not covered can be verified using complementary methods (e.g. inspections).

## Test Infrastructure Phase

In every project, requirements for the test infrastructure arise which cannot be covered with the standard infrastructure. The activities which must be executed within the test project are taken into account in this phase. This includes small customizations of the existing solutions done by the test team themselves as well as the initiation of parallel projects (like e.g. development of a HiL system) which must be initiated and tracked by the project.

### Task 8: Specify infrastructure

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | SW Module Test Plan | | | | | |
| Output | |  | | | | | |
| D: |  | E: | SW Module Test  Automation Engineer | S: | SW Module Test  Designer,  Safety Manager | I: |  |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

During the test planning phase, “Task 4: Organize test project” the requirements on the test environment were defined in the SW Module Test Plan. The aim of this task is to specify the required test environment with the necessary degree of detail.

The test environment is documented in the SW Module Test Specification or in a separate document, depending on the content of the necessary infrastructure.

For safety related systems, specific demands on the test infrastructure must be determined.

### Task 9: Realize infrastructure

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | |  | | | | | |
| Output | |  | | | | | |
| D: |  | E: | SW Module Test  Automation Engineer | S: |  | I: |  |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

Aim of this task is to implement the required infrastructure by project - if the implementation will be done by the project team - or to initiate and control the realization outside of the project.

### Task 10: Putting infrastructure into operation

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | |  | | | | | |
| Output | |  | | | | | |
| D: |  | E: | SW Module Tester | S: | SW Module Test  Automation Engineer | I: | Test Manager |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

The aim of this task is to put the test infrastructure into operation in accordance with the specification, as defined in Task 11: Specify infrastructure.

The result is communicated to the Test Manager.

## Test Analysis and Design Phase

In the test analysis and design phase, the test basis is examined with regard to testability first and then the test cases are specified. The test cases are documented in the SW Module Test Specification.

### Task 11: Analyze test basis

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | Input documents | | | | | |
| Output | | SW Module Test Specification | | | | | |
| D: |  | E: | SW Module Test  Designer | S: | Architect, Analyst or  Developer | I: | SW PM,  Test Manager |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

The aim of this task is to analyze the test basis to see whether test cases can be derived from the documents.

If the test basis is not sufficient for the construction of test cases and for test execution, suitable measures must be initiated by the test manager.

The inputs to this task are the necessary documents defined in the test planning phase, “Task 5: Determine content of test (level)”. These reference documents compose the test basis.

The results shall be assigned with the documents, describing the test basis and summarized in the SW Module Test Specification, chapter 3: Examination of the System Test Basis.

The following information shall be used for the creation of test cases:

* hardware-software interface specification and software module design specification determined and referenced as specification documents

If the software module testing is not carried out in the target environment, the differences in the source and object code, and the differences between the test environment and the target environment, shall be analyzed in order to specify additional tests in the target environment during the subsequent test phases.

Note 1 Differences between the test environment and the target environment can arise in the source code or object code, for example, due to different bit widths of data words and address words of the processors.

Note 2 Depending on the scope of the tests, the appropriate test environment for the execution of the software module is used (e.g. the target processor, a processor emulator or a development system).

Note 3 Software module testing can be executed in different environments, for example:

* Model-in-the-loop tests;
* Software-in-the-loop tests;
* Processor-in-the-loop tests; and
* Hardware-in-the-loop tests.

Note 4 For model-based development, software module testing can be carried out at the model level followed by back-to back comparison tests between the model and the object code. The back-to-back comparison tests are used to ensure that the behavior of the models with regard to the test objectives is equivalent to the automatically-generated code.

### Task 12: Specify abstract test cases

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | Input documents | | | | | |
| Output | | SW Module Test Specification | | | | | |
| D: |  | E: | SW Module Test  Designer | S: |  | I: |  |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

The aim of this task is to specify the abstract test cases and to define their priorities. Additionally the used test design techniques according to the regulation of the test strategy must be documented.

The inputs to this task are the necessary documents defined in the test planning phase, “Task 5: Determine content of test (level)”.

The test case descriptions and the assignment of priorities of the test cases are documented in the SW Module Test Specification, chapter 4-6: Refinement of the Test Approach, Test Environment, and Specification of the Test Cases.

### Task 13: Specify concrete test cases

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | Input documents | | | | | |
| Output | | SW Module Test Specification | | | | | |
| D: |  | E: | SW Module Test  Designer | S: |  | I: |  |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

The aim of this task is to detail the abstract test cases to concrete test cases and to define the order in which the test cases are processed (according to the order of integration defined in the integration strategy in case of integration tests).

The inputs to this task are the necessary documents defined in the test planning phase, “Task 5: Determine content of test (level)” and the already defined abstract test cases.

The detailed concrete test case descriptions are documented in the SW Module Test Specification, chapter 6: Specification of the Test Cases.

### Task 14: Ensure traceability and consistency

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | Input documents | | | | | |
| Output | | SW Module Test Specification | | | | | |
| D: |  | E: | SW Module Test  Designer | S: |  | I: |  |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

The aim of this task is to ensure consistency and bilateral traceability of test specifications to the corresponding requirements, specifications and design, relevant for this test level. Consistency has to be ensured by establishing and maintaining the bilateral traceability.

### Task 15: Review and release test specification

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | SW Module Test Specification | | | | | |
| Output | | Review Checklist SW Module Test Specification | | | | | |
| D: | Test Manager,  Safety Manager | E: | SW Module Test  Designer | S: | Architect, Analyst or  Developer,  Safety Manager | I: |  |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

The aim is to review and to release the SW Module Test Specification.

The result is documented in the Review Checklist SW Module Test Specification.

## Test Implementation and Execution Phase

The test cases as defined in the SW Module Test Specification are executed and the test results are documented. The Tester is responsible for the implementation of all necessary preconditions before executing the test cases. The trial run must be carried out in the test environment and recorded in the SW Module Test Report.

As mentioned above the integration aspects of the Software Integration as well as the System Integration Test Process are incorporated in the process flow charts.

### Task 16: Implement test automation

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | SW Module Test Specification | | | | | |
| Output | | Test Scripts | | | | | |
| D: |  | E: | SW Module Tester | S: |  | I: |  |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

The aim of this task is to implement the test scripts for the corresponding test cases which shall be executed automated, fitting to the test system defined for the project.

The resulting test scripts must be stored in the tool specific format.

### Task 17: Set up test execution

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | |  | | | | | |
| Output | |  | | | | | |
| D: |  | E: | SW Module Test  Designer | S: |  | I: | SW PM,  Test Manager |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

The aim of this task is to intake the test object, to check whether the input conditions are met, and to set up the test object and test environment for the start of testing. If the test object is not sufficient for test execution, suitable measures must be initiated by the test manager.

No documentation is required.

The Software Verification Report shall be used to determine the verification progress during the development phase before this test level and assess the maturity of the test object.

### Task 18: Execute tests

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | |  | | | | | |
| Output | |  | | | | | |
| D: |  | E: | SW Module Tester | S: |  | I: |  |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

The aim of this task is to execute the test cases as specified in the SW Module Test Specification according to the test strategy and to obtain the test results. This includes the execution of the re- and regression tests, following the re- and regression test strategy.

The necessary input of this task is the test object and the SW Module Test Specification including the prioritized test cases or the test scripts in case of automated tests.

The test execution is documented in log files. Deviations from the planned test sequence and abortion of tests are documented in the SW Module Test Report. Deviations of the test results from the desired behavior are documented and propagated by PTC error tickets.

The testing shall be executed as planned in accordance with test planning documented in SW Module Test

Plan and specified in accordance with the test specification documented in the SW Module Test Specification.

The test planning shall determine the test environment and the tools used for testing, if applicable. During the test execution the test environment incl. the tools actually used must be documented, particularly deviations compared with the original planning.

### Decision 1: Completed test run?

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | |  | | | | | |
| Output | |  | | | | | |
| D: | SW Module Tester | E: |  | S: |  | I: |  |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

If this test level can’t be completed before the test execution of the next level an early passing on procedure can be applied. In this case the test process for this level must split in two parallel paths.

* On the one side the test execution of the remaining test cases shall be continued.
* On the other side the current test progress shall be analyzed as described in the next tasks which end with the “Task 21: Assess and communicate maturity of test object”. In this case the report will be done as an SW Module Interims Test Status Report in the standard test report template.

## Test Evaluation and Report Phase

In this phase of the test process, the results of the current test period are summarized and assessed in the SW Module Test Report. In the case of an early passing on of the test object, the maturity of the test object is preliminary assessed.

### Task 19: Analyze test results

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | SW Module Test Logs  Deviations  Metrics | | | | | |
| Output | | SW Module Test Report | | | | | |
| D: |  | E: | SW Module Tester | S: | SW Developer | I: |  |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

In this task, the test results and the fulfillment of the test exit criteria are analyzed and summarized.

The inputs of this task are the test logs, the derivations, and the metrics.

The result is documented in the SW Module Test Report.

The evaluation of the test results shall contain the following information:

* The unique identification of the tested work product,
* The reference to the test plan and test specification,
* The configuration of the test environment and test tools used, and the calibration data used during the evaluation, if applicable,
* The level of compliance of the test results with the expected results,
* An unambiguous statement of whether the test passed or failed; if the test failed the statement shall include the rationale for failure and suggestions for changes in the tested work product, (The test is evaluated according to the criteria for completion and termination of the verification and to the expected test results.)
* The reasons for any test steps not executed.

These contents mentioned above are part of the template 8519.

### Task 20: Review test results

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | SW Module Test Report | | | | | |
| Output | | Review Checklist SW Module Test Report | | | | | |
| D: | Test Manager | E: | SW Module Tester | S: |  | I: |  |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

The aim is to hand over of the SW Module Test Report to the test manager in form of a review.

The result is documented in the Review Checklist SW Module Test Report.

### Task 21: Assess and communicate maturity of test object

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | |  | | | | | |
| Output | | SW Module Test Report | | | | | |
| D: | Test Manager,  Safety Manager | E: | Test Manager | S: | SW Quality Planner | I: | SW PM |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

The aim of this task is to provide an assessment of the maturity of the test object and to communicate the test results to all relevant parties as defined in Task 7: Organize test project.

Not performed tests regarding safety related functions have to be documented in the SW Module Test Report and have to be communicated to the customer.

The result is added in the SW Module Test Report.

## Test Closure Phase

In the test closure phase at project closure, the experiences are consolidated and for each test level a Test Completion Report is drawn up. Another important issue is the archiving of the test environment for reproduction and possible reuse.

### Task 22: Analyze test experience

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | |  | | | | | |
| Output | | SW Module Test Completion Report | | | | | |
| D: |  | E: | Test Manager | S: | SW PM,  SW Quality Planner  Test Team,  Representatives of  SW Development | I: |  |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

The aim of this task is to analyze the experiences gained in the SW Module test level.

The result is the SW Module Test Completion Report.

### Task 23: Archiving test environment

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | |  | | | | | |
| Output | |  | | | | | |
| D: |  | E: | SW Module Tester,  SW Module Test  Automation Engineer | S: |  | I: |  |
| D = Decision | | E = Execution | | S = Support | | I = Information | |

The aim of this task is to archive the test environment, test specification, test cases and other aids for subsequent tests (e.g. reuse).

The relevant test basis, documents, and test environment are stored in the PTC Source Integrity.

# Changes, References, Appendix, Terms

## References

|  |  |  |
| --- | --- | --- |
| **Category** | **Document Name** | **Document Number** |
| Process | Test of Products Containing Software | AD-PE1-1-03 |

## Template

|  |  |  |
| --- | --- | --- |
| **Category** | **Document Name** | **Document Number** |
| Template | SW Module Test Plan | 8517 |
| Template | SW Module Test Specification | 8518 |
| Template | SW Module Test Report | 8519 |
| Template | Review-Checklist SW Module Test Plan | 8340 |
| Template | Review-Checklist SW Module Test Specification | 8353 |
| Template | Review-Checklist SW Module Test Report | 8832 |
| Template | SW Module Test Completion Report | 8391 |