SW 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](#_Toc496190317)

[1.1 Purpose 5](#_Toc496190318)

[1.2 Main Description 5](#_Toc496190319)

[1.3 Result of Process 5](#_Toc496190320)

[2. Major Roles Acting in this Process 5](#_Toc496190321)

[2.1 Refinement of the major roles 6](#_Toc496190322)

[3. Process 6](#_Toc496190323)

[3.1 Process Input – Output Definition 6](#_Toc496190324)

[3.2 Introducing Aspects 7](#_Toc496190325)

[3.3 Phase Model 7](#_Toc496190326)

[3.4 Process Flow Chart 9](#_Toc496190327)

[3.5 Test Planning Phase 10](#_Toc496190328)

[3.5.1 Task 1: Determine framework conditions 10](#_Toc496190329)

[3.5.2 Task 2: Determine content of test level 11](#_Toc496190330)

[3.5.3 Task 3: Develop test strategy 12](#_Toc496190331)

[3.5.4 Task 4: Organize test project 15](#_Toc496190332)

[3.5.5 Task 5: Review and release test plan 16](#_Toc496190333)

[3.6 Test Control Phase 16](#_Toc496190334)

[3.6.1 Task 6: Manage test team and project 16](#_Toc496190335)

[3.6.2 Task 7: Monitor and communicate test progress 17](#_Toc496190336)

[3.7 Test Infrastructure Phase 17](#_Toc496190337)

[3.7.1 Task 8: Specify infrastructure 17](#_Toc496190338)

[3.7.2 Task 9: Realize infrastructure 18](#_Toc496190339)

[3.7.3 Task 10: Putting infrastructure into operation 18](#_Toc496190340)

[3.8 Test Analysis and Design Phase 18](#_Toc496190341)

[3.8.1 Task 11: Analyze test basis 18](#_Toc496190342)

[3.8.2 Task 12: Specify abstract test cases 19](#_Toc496190343)

[3.8.3 Task 13: Specify concrete test cases 20](#_Toc496190344)

[3.8.4 Task 14: Ensure traceability and consistency 20](#_Toc496190345)

[3.8.5 Task 15: Review and release test specification 20](#_Toc496190346)

[3.9 Test Implementation and Execution Phase 20](#_Toc496190347)

[3.9.1 Task 16: Implement test automation 21](#_Toc496190348)

[3.9.2 Task 17: Set up test execution 21](#_Toc496190349)

[3.9.3 Task 18: Execute tests 21](#_Toc496190350)

[3.9.4 Decision 1: Completed test run? 22](#_Toc496190351)

[3.10 Test Evaluation and Report Phase 22](#_Toc496190352)

[3.10.1 Task 19: Analyze test results 22](#_Toc496190353)

[3.10.2 Task 20: Review test results 23](#_Toc496190354)

[3.10.3 Task 21: Assess and communicate maturity of test object 23](#_Toc496190355)

[3.11 Test Closure Phase 23](#_Toc496190356)

[3.11.1 Task 22: Analyze test experience 23](#_Toc496190357)

[3.11.2 Task 23: Archiving test environment 24](#_Toc496190358)

[4. Changes, References, Appendix, Terms 24](#_Toc496190359)

[4.1 References 24](#_Toc496190360)

[4.2 Template 24](#_Toc496190361)

# Purpose Description

## Purpose

The purpose of the software testing process is to confirm that the integrated software meets the defined software requirements.

## Main Description

The software test is the next test level after the software integration test level. Focus of the software test level is

* Functionality of the entire software product
* Non-functional requirements

The following problems can be found:

* Functional and non-functional failures
* Problems of the entire software product
* Conceptual failure

## Result of Process

As a result of successful implementation of this software testing process:

* A strategy is developed to test the integrated software according to the priorities and categorization of the software requirements;
* A test specification software test for the integrated software is developed that demonstrates compliance to the software requirements;
* The integrated software is verified;
* Test results are recorded;
* Consistency and bilateral traceability are established between software requirements and test specification software test including test cases; and
* A regression test strategy is developed and applied for re-testing the integrated software when a change in software items occurs.

NOTE 1: The test specification for software testing includes the test design specification, test procedure specification and test case specifications.

NOTE 2: The verification is performed according to the test cases

NOTE 3: The test results for software testing include the test logs, test incident reports and test summary reports.

# 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 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 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 Tester” role (MHE-PE-01,

Supplement A):

|  |  |
| --- | --- |
| **Role** | **Contribution and Responsibilities** |
| SW 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 Tester | Carries out the tests as defined in the test specification and draws up the reports including measurements. |
| SW 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 |
| System Requirements Specification | System Analyst |
| Software Requirements Specification | SW Analyst |
| SW Architecture (e.g. Innovator SD model) | SW Architect |
| Attributes for economic risk classification on software level | SW Project Manager  SW Analyst  SW Architect |
| Release Plan | SW-Project Manager |
| Software product (test object – for HW integration) | SW Integrator |

The following process output is produced:

|  |  |
| --- | --- |
| **Process output** | **To customer** |
| Software Test Plan | SW Project Manager  Safety Manager |
| Review Checklist Software Test Plan | SW Test Designer |
| Software Test Specification | SW Tester |
| Review Checklist Software Test Specification | SW Test Designer |
| Software Test Report | SW Project Manager  Safety Manager |
| 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 specific test level and document the planning decisions like how, by whom, what with and when what test activities are carried out in the SW Test Plan.

### Task 1: Determine framework conditions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | |  | | | | | |
| Output | | SW 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 Test Plan, chapter 2: Introduction.

### Task 2: Determine content of test level

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | |  | | | | | |
| Output | | SW Test Plan | | | | | |
| D: | Test Manager | E: | SW 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:

* Software architectural design specification and the software safety requirements specification shall be determined and referenced as specification documents and shall be used as input for the creation of test cases
* Technical safety concept can be determined and referenced as project-specific particularities and can be used as input for the test strategy
* System design specification can be determined and referenced as specification documents and can be used as input for the creation of test cases
* Safety Plan shall be determined and referenced as project-specific particularities and should be used as input for the test strategy
* The embedded software shall be defined as the test object(s) of this test level

The software testing shall be applied to demonstrate that the embedded software fulfils the software safety requirements.

The testing of the implementation of the software safety requirements shall be executed on the target hardware, documented as project-specific particularities.

Furthermore the software testing shall be applied to demonstrate that the embedded software achieves:

* The specified functionality \*
* Robustness \* (e.g. the absence of inaccessible software, the effectiveness of error detection and error handling mechanisms.)
* Sufficient resources to support the functionality \* (as far as this has not been tested at software integration test level)

Remark: \* The marked topics can be tested partly in the software integration test level and partly in this software test level, which has the test of the complete embedded software in focus. The test strategies must ensure, that these topics are tested in an appropriate extend.

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

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

### Task 3: Develop test strategy

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

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

* The content of the Safety Plan determined for the software testing shall be taken over and integrated in this test strategy. This content shall contain the demanded test environments for conducting the software safety requirements verification.

To verify that the embedded software fulfils the software safety requirements, tests shall be conducted in the test environments listed in Table 1:

Note: Test cases that already exist, for example from software integration testing, can be re-used.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Methods | | ASIL | | | |
| A | B | C | D |
| 1a | Hardware-in-the-loop | + | + | ++ | ++ |
| 1b | Electronic control unit network environments a | ++ | ++ | ++ | ++ |
| 1c | Vehicles | ++ | + | ++ | ++ |
| a Examples include test benches partially or fully integrating the electrical systems of a vehicle, “lab-cars” or “mule” vehicles, and “rest of the bus” simulations. | | | | | |

Table 1: Test environments for conducting the software safety requirements verification

The following methods (Table 2 - Table 4) shall to be applied at the software integration test and partly at the software level. The assessment of the completeness of the application of these methods regarding the attainment of the test aim must be carried out comprehensive over these two test levels. (See Remark)

The test methods listed in Table 2 shall be applied to demonstrate that the embedded software achieves the scope of test content defined:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Methods | | ASIL | | | |
| A | B | C | D |
| 1a | Requirements-based test a | ++ | ++ | ++ | ++ |
| 1b | Interface test | ++ | ++ | ++ | ++ |
| 1c | Fault injection test b | + | + | ++ | ++ |
| 1d | Resource usage test c, d | + | + | + | ++ |
| 1e | Back-to-back comparison test between model and code, if applicable e | + | + | ++ | ++ |
| a The software requirements at the architectural level are the basis for this requirements-based test.  b This includes injection of arbitrary faults in order to test safety mechanisms (e.g. by corrupting software or hardware components).  c To ensure the fulfilment of requirements influenced by the hardware architectural design with sufficient tolerance, properties such as average and maximum processor performance, minimum or maximum execution times, storage usage (e.g. RAM for stack and heap, ROM for program and data) and the bandwidth of communication links (e.g. data buses) have to be determined.  d Some aspects of the resource usage test can only be evaluated properly when the software tests are executed on the target hardware or if the emulator for the target processor supports resource usage tests.  e This method requires a model that can simulate the functionality of the software components. Here, the model and code are stimulated in the same way and results compared with each other. | | | | | |

Table 2: Methods for software testing

The selection of the method for software testing or the criteria to define the selection of these methods for specific types of software interfaces 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 test methods selected in accordance with Table 2, test cases shall be derived using the methods listed in Table 3.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 parameters or variables, 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. | | | | | |

Table 3: Methods for deriving test cases for software testing

The selection of the methods for deriving test cases for software testing or the criteria to define the selection of these methods for specific types of software 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 tests and to obtain confidence that there is no unintended functionality the coverage of requirements at the software architectural level by test cases shall be determined. If necessary, additional test cases shall be specified or a rationale shall be provided.

To evaluate the completeness of test cases and to obtain confidence that there is no unintended functionality the structural coverage shall be measured in accordance with the metrics listed in Table 4.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Methods | | ASIL | | | |
| A | B | C | D |
| 1a | Function coverage a | + | + | ++ | ++ |
| 1b | Call coverage b | + | + | ++ | ++ |
| a Method 1a refers to the percentage of executed software functions. This evidence can be achieved by an appropriate software integration strategy.  b Method 1b refers to the percentage of executed software function calls. | | | | | |

Table 4: Structural coverage metrics at the software architectural level

Note 1 The structural coverage can be determined using appropriate software tools.

Note2 In the case of model-based development, software architecture testing can be performed at the model level using analogous structural coverage metrics for models.

It shall be verified that the embedded software that is to be included as part of a production release contains all the specified functions, and only contains other unspecified functions if these functions do not impair the compliance with the software safety requirements.

(In this context unspecified functions include code used for debugging or instrumentation. If deactivation of these unspecified functions can be ensured, this is an acceptable means of compliance with this requirement. Otherwise the removal of such code is a change).

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 Test Plan, chapter 4: Test strategy.

### Task 4: Organize test project

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | |  | | | | | |
| Output | | SW Test Plan | | | | | |
| D: |  | E: | Test Manager | S: | SW PM, SW Test Designer, SW Tester, SW 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 and considers demanded test environments for conducting the software safety requirements verification defined and taken over in the test strategy.

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

### Task 5: Review and release test plan

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

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

The result is documented in the Review Checklist SW 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 | |  | | | | | |
| D: |  | E: | Test Manager | S: | SW Tester,  SW Test Designer,  SW 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 | |  | | | | | |
| D: |  | E: | Test Manager | S: | SW Tester,  SW Test Designer, SW 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 Test Plan and the results logged in the SW 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 Test Plan; temporary changes in the Test Report.

## 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 | |  | | | | | |
| Output | |  | | | | | |
| D: |  | E: | SW Test Automation Engineer | S: | SW 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 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 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.

The test planning shall determine the demands on the test environment, if applicable. The activities of Task 7: Organize test project are continued by specifying in this process task and in the following tasks.

### Task 9: Realize infrastructure

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | |  | | | | | |
| Output | |  | | | | | |
| D: |  | E: | SW 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 Tester | S: | SW 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 Test Specification (for the corresponding templates see 5.1 References – Forms).

### Task 11: Analyze test basis

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | |  | | | | | |
| Output | | SW Test Specification | | | | | |
| D: |  | E: | SW 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 Test Specification, chapter 3: Examination of the System Test Basis.

### Task 12: Specify abstract test cases

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | |  | | | | | |
| Output | | SW Test Specification | | | | | |
| D: |  | E: | SW 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 Test Specification, chapter 4-6: Refinement of the Test Approach, Test Environment, and Specification of the Test Cases.

The specification of each test case shall include the following content:

* A unique identification,
* The reference to the version of the associated work product to be verified,
* The preconditions and configurations, (If a complete test of the possible configurations of a work product (e.g. variants of a system) is not feasible, a reasonable subset is selected (e.g. minimum or maximum functionality configurations of a system)).
* The environmental conditions, if appropriate, (Environmental conditions relate to the physical properties (e.g. temperature) of the surroundings in which the test is conducted or is simulated as part of the test.)
* The input data, their time sequence and their values, and
* The expected behavior which includes output data, acceptable ranges of output values, time behavior and tolerance behavior (When specifying the expected behavior, it might be necessary to specify the initial output data in order to detect changes. To avoid the redundant specification and storage of preconditions, configurations and environmental conditions used for various test cases, the use of an unambiguous reference to such data is recommended.)

Test cases shall be grouped according to the test methods to be applied. For each test method, in addition to the test cases, the following shall be specified:

* The test environment,
* The logical and temporal dependencies, and
* The resources.

Together with the test cases or (at least) the test case groups it must be documented

* Which methods for software testing are actually used according to the recommended methods
* Which methods for deriving test cases are actually used according to the recommended methods and the definition how to map these methods to software defined in the SW Test Plan

### Task 13: Specify concrete test cases

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | |  | | | | | |
| Output | | SW Test Specification | | | | | |
| D: |  | E: | SW 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 Test Specification, chapter 6: Specification of the Test Cases.

### Task 14: Ensure traceability and consistency

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | |  | | | | | |
| Output | | SW Test Specification | | | | | |
| D: |  | E: | SW 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 Test Specification | | | | | |
| Output | | Review Checklist SW Test Specification | | | | | |
| D: | Test Manager,  Safety Manager | E: | SW 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 Test Specification.

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

## Test Implementation and Execution Phase

The test cases as defined in the SW 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 Test Report (for the corresponding templates see 5.1 References – Forms).

### Task 16: Implement test automation

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | | SW Test Specification | | | | | |
| Output | | Test Scripts | | | | | |
| D: |  | E: | SW 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 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 SW Module Test Report and the SW Integration (Test) 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 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 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 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 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 Test Plan and specified in accordance with the test specification documented in the SW 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: |  | 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 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 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 | |  | | | | | |
| Output | | SW Test Report | | | | | |
| D: |  | E: | SW 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 Test Report.

The results of the verification of the software safety requirements shall be evaluated with regard to:

* Compliance with the expected results
* Coverage of the software safety requirements
* Pass or fail criteria

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.

### Task 20: Review test results

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

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

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

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

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | |  | | | | | |
| Output | | SW 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 Test Report and have to be communicated to the customer.

The result is added in the SW 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 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 test level.

The result is the SW Test Completion Report.

### Task 23: Archiving test environment

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Input | |  | | | | | |
| Output | |  | | | | | |
| D: |  | E: | SW Tester,  SW 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 Test Plan | 8517 |
| Template | SW Test Specification | 8518 |
| Template | SW Test Report | 8519 |
| Template | Review-Checklist SW Test Plan | 8392 |
| Template | Review-Checklist SW Test Specification | 9394 |
| Template | Review-Checklist SW Test Report | 8832 |
| Template | SW Test Completion Report | 8391 |