Project number Data Science Workbench - DSW

Version: 0.1

|  |
| --- |
| The test plan describes the following topics that occur during IT project execution:   * Scope * Approach * Resources * Schedule of intended test activities   The test plan includes a complete documentation of all test activities, which are necessary before, during and after test execution in order to rate the quality of an IT solution. Thus, the test plan has a huge impact to deliver a high quality IT solution into productive use.  Tests have a big impact on project and product quality. It is strongly recommended to create the test plan simultaneously to other tasks in the project. The creation of the test plan is done in discussion with the project core team and optionally further necessary experts.  The creation of the test plan is done simultaneously to the creation of the Business Blueprint. |

**Authors** [to approve]

|  |
| --- |
| Please, note all persons involved in the creation of the test plan including their name and corresponding company. If applicable, you can group persons as follows   * Authors take responsibility for the whole document * Co-authors take responsibility for certain chapters * Co-worker give input to specific topics |
|  |

| **Name** | **Role** | **Department** | **Signature and Date** |
| --- | --- | --- | --- |
| Peter Rohner | Test Manager | Trivadis AG | Electronically signed |
|  |  |  |  |

**Review / Approval** [to approve]

| **Name** | **Role** | **Department** | **Signature and Date** |
| --- | --- | --- | --- |
|  | PM | E+H Flow | Electronically signed |
|  |  |  |  |

**History of Changes** [to approve]

|  |
| --- |
| The following name conventions are to administrate different versions of the document:   * Version 0.x: for the phase of creation. Further structuring behind 0.x is optional. * Version 1.0: for the first released test plan. * Version 1.x: for a changed status after release * Version n.0: for the n-the released test plan   The use of the history of changes is optional. It can also be done via versioning in Project Monitor. |

| **Version** | **Changes** | **Effective Date** |
| --- | --- | --- |
| 1 |  | Date of last signature |
|  |  |  |

**Table of Contents**

[1 Introduction [to approve] 5](#_Toc48054948)

[1.1 Purpose and Scope [to approve] 5](#_Toc48054949)

[1.2 Out of Scope [to approve] 5](#_Toc48054950)

[1.3 The 7 Principles of Software Testing 5](#_Toc48054951)

[2 Corresponding Documents [to approve] 5](#_Toc48054952)

[3 Test Phases and Responsibilities [to approve] 7](#_Toc48054953)

[4 Test Strategy [to approve] 8](#_Toc48054954)

[5 Test Scope [to approve] 8](#_Toc48054955)

[5.1 Testing Custom Code [to approve] 8](#_Toc48054956)

[5.2 Testing internal developed Code [to approve] 9](#_Toc48054957)

[5.3 Configuration Qualification (CQ) [to approve] 9](#_Toc48054958)

[5.4 Installation Qualification (IQ) [to approve] 9](#_Toc48054959)

[5.5 Sanity Test [to approve] 9](#_Toc48054960)

[5.6 Functional Testing (SAT) [to approve] 9](#_Toc48054961)

[5.6.1 System Functionality [to approve] 9](#_Toc48054962)

[5.6.2 Security Testing: Access Control and Authorization [to approve] 10](#_Toc48054963)

[5.6.3 System Performance Testing (Performance, Load and Stress Tests) [to approve] 10](#_Toc48054964)

[5.7 User Acceptance Test (UAT) [to approve] 10](#_Toc48054965)

[6 Test Planning and Organization [to approve] 10](#_Toc48054966)

[6.1 Pre-Requisites for testing [to approve] 10](#_Toc48054967)

[6.2 Test Schedule and Testers [to approve] 11](#_Toc48054968)

[6.3 Tester Training [to approve] 11](#_Toc48054969)

[6.4 Preparing of the System Under Test (SUT) [to approve] 11](#_Toc48054970)

[6.4.1 Environment: TEST [to approve] 11](#_Toc48054971)

[6.4.2 Environment: PRODUCTION [to approve] 11](#_Toc48054972)

[7 Test Documentation [to approve] 11](#_Toc48054973)

[7.1 Test Tools [to approve] 11](#_Toc48054974)

[7.2 Test Specifications [to approve] 12](#_Toc48054975)

[7.3 Test Results [to approve] 12](#_Toc48054976)

[7.4 Test Defects [to approve] 12](#_Toc48054977)

[7.4.1 Test Defect Log [to approve] 12](#_Toc48054978)

[7.4.2 Re-test Process / Change Management Procedure [to approve] 12](#_Toc48054979)

[7.5 Acceptance Criteria [to approve] 12](#_Toc48054980)

[7.5.1 Test Case Result [to approve] 12](#_Toc48054981)

[7.5.2 Test Phase [to approve] 12](#_Toc48054982)

[7.6 Test Summary Reports [to approve] 12](#_Toc48054983)

[8 Test Scope 12](#_Toc48054984)

[8.1 System Scope 12](#_Toc48054985)

[8.2 Process Scope 13](#_Toc48054986)

[9 Use Cases und Tests 14](#_Toc48054987)

[9.1 Use Cases 14](#_Toc48054988)

[9.2 Performance, Load and Stress Tests 15](#_Toc48054989)

[9.3 Security Tests 15](#_Toc48054990)

[9.4 Acceptance Criteria 15](#_Toc48054991)

[10 Test activities 16](#_Toc48054992)

[10.1 Time Scheduling 16](#_Toc48054993)

[10.2 Analysis and Design 16](#_Toc48054994)

[10.3 Realization 17](#_Toc48054995)

[10.4 Execution 18](#_Toc48054996)

[11 Test tools 20](#_Toc48054997)

[12 Persons and roles 21](#_Toc48054998)

[12.1 Stakeholder 21](#_Toc48054999)

[12.2 Test Team 21](#_Toc48055000)

[13 Attachments 22](#_Toc48055001)

[13.1 Glossary 22](#_Toc48055002)

[13.2 Contact Persons for Test Topics 22](#_Toc48055003)

[14 Publishing [to approve] 22](#_Toc48055004)

# Introduction [to approve]

This document is the Test Plan for the Data Science Workbench (DSW) system. It describes all applicable test procedures including the handling of test specifications and results including how the traceability from requirements to test cases is done for this project.

Purpose and Scope [to approve]

All tests will be performed according to the procedures described in this document to ensure that they are properly planned, specified, executed, evaluated and approved.

Any procedures described in this document are applicable to all verification tests performed during system implementation.

It provides the framework for testing before deployment, based on any risk profile (system and/or functional risk assessments). It addresses the following testing topics:

* What must be tested
* How the testing will be performed
* Who is responsible for the testing
* Test defect handling and tracking method
* Criteria for system acceptance

Out of Scope [to approve]

Penetration Testing.

This document does not include details of specific tests cases. Test Cases are stored in a document [@Todo: document collecting all test cases].

The **7 Principles of Software Testing**

* Testing shows presence of defects
* Exhaustive testing is not possible
* Early testing
* Defect clustering (small number of modules contain most of the defects detected)
* Pesticide paradox (test cases need to be regularly reviewed & revised)
* Testing is context dependent
* Absence of errors fallacy (software which is 99% bug-free is still unusable)

# Corresponding Documents [to approve]

|  |
| --- |
| Please, list all documents the test plan refers to and its file storage, e.g.:   * Process documents from process workshops * Redbook * Business Blueprint * Project plan * Or all alternative file storages of business and technical requirements documentation * Storage for test cases * System landscape * Reports (test case status report, test status report, test progress report, test summary report, defect report and defect status report) |

| **Reference #** | **Management document[[1]](#footnote-1) / Version** | **File Storage** |
| --- | --- | --- |
| [M1] | Actual project plan | Project Monitor |
| [M2] | Released test plan | Project Monitor (active versioning)  HP ALM (in module Requirements) |

| **Reference #** | **Test documentation / Version** | **File Storage** |
| --- | --- | --- |
| [T1] | Test procedure description and  test cases | HP ALM „[DOMAIN]\[PROJECT]“ module „TestPlan“ |
| [T2] | Test execution report | HP ALM „[DOMAIN]\[PROJECT]“ module „TestLab“ |
| [T3] | Defect report | HP ALM „[DOMAIN]\[PROJECT]“ module „Defects“ |
| [T4] | Package of measures | HP ALM „[DOMAIN]\[PROJECT]“ module „TestLab“ As attachment in folder „package of measures “ to the project. |
| [T5] | Release documentation | HP ALM „[DOMAIN]\[PROJECT]“ module „TestLab“ As attachment in folder „release documentation“ to the project or in Project Monitor. |

| **Reference #** | **Report/Statistics / Version** | **File Storage** |
| --- | --- | --- |
| [R1] | Test case status report | HP ALM „[DOMAIN]\[PROJECT]“ 🡪 Test case status report |
| [R2] | Test status report | HP ALM „[DOMAIN]\[PROJECT]“ 🡪 Test status report |
| [R3] | Test progress report | HP ALM „[DOMAIN]\[PROJECT]“ 🡪 Test progress report |
| [R4] | Defect status report | HP ALM „[DOMAIN]\[PROJECT]“ 🡪 Defect status report |
| [R5] | Test summary report | HP ALM „[DOMAIN]\[PROJECT]“ 🡪 Test summary report |

# Test Phases and Responsibilities [to approve]

The table below gives an overview of the applicable test phases including on which system environment those tests is be performed as well as the responsibilities for the different test phases.

Each test phase is executed in a specific project phase and test environment.

A person with its assigned role in a specific test phase is responsible for the correct execution of the test phase. If the test type is formal, the required documentation must be created in this phase.

The documentation to use will help while execution of a test phase.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Project Phase** | **Test Phase** | **Test Environments** | **Responsibility / Role** | **Test Type** | **Documentation to use** |
| Requirements and Architecture | Review BRS, SRS, URS | |  |  |  | | --- | --- | --- | | ~~DEV~~ | ~~TEST~~ | ~~PROD~~ | | Requirements Engineering Team Leader, Product Owner, Business | Informal | Use: BRS, SRS, URS  Update: BRS, SRS, URS |
| Detailed Design | Review FS | |  |  |  | | --- | --- | --- | | ~~DEV~~ | ~~TEST~~ | ~~PROD~~ | | Requirements Engineering Team Leader, Product Owner, Business | Informal | Use: FS  Update: FS |
| Implementation | Unit Test | |  |  |  | | --- | --- | --- | | **DEV** | ~~TEST~~ | ~~PROD~~ | | Development Team Leader | Informal | *Optional: Use if any exists.* |
| Implementation | Component Test | |  |  |  | | --- | --- | --- | | **DEV** | ~~TEST~~ | ~~PROD~~ | | Development Team Leader | Informal | *Optional: Use if any exists.* |
| Implementation | Code Review | |  |  |  | | --- | --- | --- | | **DEV** | ~~TEST~~ | ~~PROD~~ | | Development Team Leader | Informal | *Optional: Use if any exists.* |
| Pre-Installation  And / Or  Post-Installation | Configuration Qualification (CQ) | |  |  |  | | --- | --- | --- | | ~~DEV~~ | **TEST** | **PROD** | | Development Team Leader, Configuration Manager | Formal | Use: FS, Configuration Settings, Installation Scripts  Create: Test Summary Report |
| Installation | Installation Qualification (IQ) | |  |  |  | | --- | --- | --- | | ~~DEV~~ | **TEST** | **PROD** | | Development Team Leader | Formal | Use: Installation Scripts, Check Scripts  Create: Installation Report |
| System Integration Test | Sanity Test | |  |  |  | | --- | --- | --- | | ~~DEV~~ | **TEST** | ~~PROD~~ | | Test Manager | Formal | *Optional: Informal test to see if system is somehow working and the next phase of testing can be started.* |
| System Integration Test | Functional Testing (SAT) | |  |  |  | | --- | --- | --- | | ~~DEV~~ | **TEST** | ~~PROD~~ | | Test Manager | Formal | Use: Test Plan, Test Cases  Create/Update: Defect Report, Test Summary Report |
| Acceptance Test | User Acceptance Test (UAT) | |  |  |  | | --- | --- | --- | | ~~DEV~~ | **TEST** | ~~PROD~~ | | Business | Formal | Use: BRS, SRS, URS  Create/Update: Defect Report, Test Summary Report, Acceptance Report |

Table Test Phases

# Test Strategy [to approve]

At the moment there exists no Risk Assessment.

Testing will be performed according to E+H CSV SOP xyz [PRO: do you have any CSV SOP or guide lines?].

The purpose of testing is to provide documented evidence that the functionality of the DSW application operates as expected according to the user requirements.

Application tests will be performed according to the procedures described in this section to ensure that testing is properly planned, specified, executed, evaluated and approved.

Project scope includes the following activities: [PRO: does following step match E+H requirements?]

* Prior testing, application must be successfully installed.
* Installation Qualification (IQ) / Configuration Qualification (CQ) shall be performed for Test and Production environments.
* Upon successful completion of CQs for Test environment and successfully execution of sanity testing, Functional Testing (SAT) shall be performed in the Test environment.
* Only upon successful completion of SAT in Test environment, User Acceptance Test (UAT) may be carried out in Test environment.
* Only upon successful completion of UAT in Test environment, IQ/CQ shall be performed in Production environment.
* Test Summary Reports will be written to report activities stated in the test plan.

Testing will be performed by using *<testing tool for IQ/CQ/SAT/UAT> [Rxxx]*:

* Any tool specific information ....

The test approach for DSW is considering Unit, Component, Configuration, Functional and User Acceptance Testing, as shown in Table 1 Test Phases in this document.

The validation approach adopted by the DSW is risk-based. This means that the project will conduct directed testing around the system functionality (SAT) and business processes (UAT) to provide a high degree of assurance that the system is performing according to its requirements.

Similarly, end-to-end testing of processes and task flow is performed during User Acceptance Test (UAT), which is described in this document in section User Acceptance Test (UAT). The risk-based approach applied to SAT and UAT uses a CSV Risk Assessment [PRO: will this be done by E+H] to rate the risk factors for the processes respectively the testing effort. Then test design for SAT and UAT Testing is focused on providing evidence of the successful mitigation of those risks.

The test phases with their corresponding test environments and responsible units are shown in Table 1 Test Phases.

# Test Scope [to approve]

The following chapters will specify the performed tests type to ensure the readiness of the DSW system for the final usage. The test approach will is based on the performance of the risk assessment [PRO: reference to DSW risk assessment document].

A test scope can have an entry criteria to be passed before it can be executed.

Testing Custom Code [to approve]

A custom code can be a software product used in the DSW architecture or a library used by the E+H internal IT development for DSW project.

All custom software solution used in the DSW project are considerate as working as expected without any issues.

Therefore all used custom software solution in this project will not be tested.

Testing internal developed Code [to approve]

Testing internal developed E+H Code is performed by software engineers in the development environment. This is achieved by doing pair programming or pull request code reviews. There is also the requirement to create unit tests for developed APIs, functions, services. For some of the URS there may also exist unit tests to prove them.

Aim of these unit tests is to challenge software components to ensure that they function properly prior to assembling the system for formal testing.

Formal documentation of development testing is not required, but a statement shall be given by the development team stating that the final code is ready for testing: this can be done after a successful source code review and merge process.

Configuration Qualification (CQ) [to approve]

The Aim of the CQ is to verify the correct configuration of the application and all its depending components.

CQ will be executed on the TEST/Qualification system and again on the PRODUCTION system.

Installation Qualification (IQ) [to approve]

IQ covers the successful installation of the application software.

It could also covers the installation and configuration of components like system software (e.g. Operating System, libraries, etc.), supporting software (e.g. all required testing tools to test), hardware (e.g. hard disk space, memory, CPU count, terminal, etc.).

If any software or hardware suppliers IQs are available then those IQ may also be used. Those suppliers IQ may be referenced in the Test Summary Report.

IQ will be executed on the TEST/Qualification system and again on the PRODUCTION system.

Sanity Test [to approve]

Sanity testing is a kind of Software Testing performed after receiving a software build to ascertain that no major issues or blockers are introduced. The goal is to determine that the proposed functionality works roughly as expected. If sanity test fails, the build is rejected to save the time and costs involved in a more rigorous testing.

Functional Testing (SAT) [to approve]

### System Functionality [to approve]

The purpose of SAT is to verify in an operational environment that the complete system satisfies the system functionality and configuration as outlined in Functional Specifications. This includes testing the master data this system depends on in any case.

Also the Integration Testing of the different subsystem combined to one final system will be tested if the work as expected via their interfaces.

If a new version of the SUT is created, the Regression Testing must be executed on unchanged part of the system to guarantee those systems are not affected by the new deployed version.

**Outcome**

Defect Report: Any found issues or failed test cases must be reported in the Defect Report.

Test Summary Report: The result of this test is be reported in the Test Summary Report.

### Security Testing: Access Control and Authorization [to approve]

Access Control & Authorization testing ensures that access to the system and data is granted based on defined user profiles.

**Outcome**

Defect Report: Any found issues or failed test cases must be reported in the Defect Report.

Test Summary Report: The result of this test is be reported in the Test Summary Report.

### System Performance Testing (Performance, Load and Stress Tests) [to approve]

|  |
| --- |
| Hints for performance and load tests:  At Endress+Hauser all applications are included to the network and thus influence the performance of other applications. For that reasons, it is important to consider the user performance already in Redbook phase and to keep it during the whole IT project.  Please keep in mind:   * Do not test with empty databases. You can simulate user behavior by using slow network connections. Test all relevant modules as developer to recognize bottlenecks. * Developer can optimize the implementation in using small test programs for important components. The InfoServe performance experts support you creating such programs. * Performance tests are complex. Usually, they are run for more than one week to come to a respectable statement about the application performance. |

Aim of performance testing is to determine how a system performs in terms of responsiveness and stability under a particular workload. For the DSW project the good performance and a stable system has a high importance for the acceptance of the system.

**Outcome**

Defect Report: Any found issues or failed test cases must be reported in the Defect Report.

Test Summary Report: The result of this test is be reported in the Test Summary Report.

User Acceptance Test (UAT) [to approve]

The UAT must be executed by key users of the business to verify that the business requirements (BSR, USR, SRS) are met.

The business must define the acceptance criteria that must be met in the UAT that the system can be installed to the production environment.

**Outcome**

Defect Report: Any found issues or failed test cases must be reported in the Defect Report.

Test Summary Report: The result of this test is be reported in the Test Summary Report.

# Test Planning and Organization [to approve]

The Test Manager is responsible for test coordination.

Details on the tests executed in the specific phase can be found in the Test Summary Report as defined in chapter Test Specifications [to approve]

Test Results [to approve]

Test Defects [to approve]

### Test Defect Log [to approve]

### Re-test Process / Change Management Procedure [to approve]

Acceptance Criteria [to approve]

### Test Case Result [to approve]

Acceptance Criteria for a Test Result are:

Acceptance Criteria for a Test Step is defined as part of the definition of the “Expected Result”. Based on the initial defect assessment, the Test Result can be “accepted”, “accepted with defects” or “rejected” by the approvers. This decision is documented as part of the Test Result.

### Test Phase [to approve]

Acceptance Criteria for a Test Phase are:

* All tests planned for / in scope of this phase were executed and the results were approved.
* Re-Tests applicable to the test phase (for example as defined in the Defect Log) were executed and the results were approved.
* Defects documented in Defect Log.
* Open defects must be specifically assessed for their impact on the release. Plans for closure / mitigation actions for open defects must be identified, recorded and accepted by all approvers of the Test Summary Reports [to approve].

Test Summary Reports [to approve].

Any found issues while test execution will be reported in the Defect Log Report as defined in chapter Test Defect Log [to approve] in this document.

Pre-Requisites for testing [to approve]

The following pre-requisites are required for the testing:

* Test System Ready
* All system code has passed the development phase (unit test, component test, test coverage, code reviews)
* A successful build of the system
* The role based trainings of all testers are completed
* Any required privileges are granted to the testers
* Test cases are approved and signed

Test Schedule and Testers [to approve]

Details on general test responsibilities and test environments per test phase is listed in chapter Test Phases and Responsibilities [to approve] in this document.

Tester Training [to approve]

A tester is trained according the E+H Flow internal tester training guidelines, knows how to use the testing tool used in this project and how to document an executed test case.

It would be nice if tester do have experience in CSV.

Preparing of the System Under Test (SUT) [to approve]

The configuration of the SUT should be done in CQ phase; after the CQ is successfully passed, the Test Manager will coordinate with the testing team and the installation team about the creation of the data that is required for the execution of the test cases.

The creation of the test data is a prerequisite for the SAT and UAT.

### Environment: TEST [to approve]

Test environment will be considered fit for testing purpose after the successful execution of IQ / CQ and approval of IQ / CQ results.

All Test scripts of the SAT and UAT must are executed on the Test Environment.

### Environment: PRODUCTION [to approve]

The Production environment will be considered fit for testing purpose after the successful execution of CQ / IQ in Production environment and approval of CQ / IQ results.

# Test Documentation [to approve]

The following chapters describes the documents, which are needed for the test execution and all documents, which will be created or maintained while executing testes respectively at the end of a test phase.

Test Tools [to approve]

@Todo: what tools do we have at E+H for manual test execution?

Test Specifications [to approve]

Test Results [to approve]

Test Defects [to approve]

### Test Defect Log [to approve]

### Re-test Process / Change Management Procedure [to approve]

Acceptance Criteria [to approve]

### Test Case Result [to approve]

Acceptance Criteria for a Test Result are:

Acceptance Criteria for a Test Step is defined as part of the definition of the “Expected Result”. Based on the initial defect assessment, the Test Result can be “accepted”, “accepted with defects” or “rejected” by the approvers. This decision is documented as part of the Test Result.

### Test Phase [to approve]

Acceptance Criteria for a Test Phase are:

* All tests planned for / in scope of this phase were executed and the results were approved.
* Re-Tests applicable to the test phase (for example as defined in the Defect Log) were executed and the results were approved.
* Defects documented in Defect Log.
* Open defects must be specifically assessed for their impact on the release. Plans for closure / mitigation actions for open defects must be identified, recorded and accepted by all approvers of the Test Summary Reports [to approve].

Test Summary Reports [to approve]

\*\*\* Old Part \*\*\*

# Test Scope

This chapter describes the boundaries of the project concerning the considered systems as well as the considered processes.

System tests will be performed according to the procedures described in this document to ensure that they are properly planned, specified, executed, evaluated and approved.

Procedures described in this document are applicable to all verification tests performed during system implementation.

System Scope

|  |
| --- |
| In this chapter describe all necessary interfaces to pre and post systems that have to be considered in the test plan. Thus, you will get a clear definition of test boundaries (where does the test start and where does it end from the IT systems point of view).  Give an overview about the system landscape (graphical view).  List all system configuration and applications that are part of the test, e.g.   * Web application using Internet Explorer 11, other browsers are excluded * SAP Q-System * Office 2010, Windows 7 |

Process Scope

|  |
| --- |
| In this chapter describe all necessary interfaces to pre and post processes that have to be considered in the test plan. Thus, you will get a clear definition of test boundaries (where does the test start and where does it end from the process point of view).  Use the kind of VAD presentation from the process modeling standard 080. If this is already defined in Redbook or Blueprint in detail and if there are no changes to the processes, you can refer to these documents. |

# Use Cases und Tests

|  |
| --- |
| Describe all processes (including pre and post processes) and/ or use cases that have to be tested, even though they were not explicitly changed by the project. You can refer to existing process documentation.  Please list for every use case:   * Which role has to execute the test? * What is the expected test result? * Which test data has to be used? * In which test the use case has to be executed?   The structure of this chapter can be taken from Blueprint document, chapter 3 and 4. Please keep in mind when creating the use case list that all affected processes and variants have to be considered completely.  The listed tables include examples. They can be deleted in a specific project. |

Use Cases

Use case: 01\_Create\_Request

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case shell** | **Role** | **Expected result** | **Test level** | **Test data** |
| 01\_Create\_Request\_PMO | PMO | Request is created | Acceptance test  System test | In ressource file |
| 01\_Create\_Request\_GUEST | GUEST | Request cannot be created | Acceptance test | In ressource file |

Use Case: 02\_Store\_Allowed\_Equipment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test case shell** | **Role** | **Expected result** | **Test level** | **Test data** |
| 02\_Store\_Allowed\_Equipment \_IBT | IBT | Equipment ist stored | Acceptance test | in test case |
| 02\_Store\_Allowed\_Equipment\_XYZ | XYZ | XYZ | Acceptance test | free |

Performance, Load and Stress Tests

|  |
| --- |
| Hints for performance and load tests:  At Endress+Hauser all applications are included to the network and thus influence the performance of other applications. For that reasons, it is important to consider the user performance already in Redbook phase and to keep it during the whole IT project.  Please keep in mind:   * Do not test with empty databases. You can simulate user behavior by using slow network connections. Test all relevant modules as developer to recognize bottlenecks. * Developer can optimize the implementation in using small test programs for important components. The InfoServe performance experts support you creating such programs. * Performance tests are complex. Usually, they are run for more than one week to come to a respectable statement about the application performance. |

Security Tests

No specific Test Specifications for Access Control & Authorization must be created, as there is no possible access from outside of Endress + Hauser Flowtec AG network possible.

Acceptance Criteria

|  |
| --- |
| The exit criteria that a component or system must satisfy in order to be accepted by a user, customer, or other authorized entity. [IEEE 610]  The acceptance criteria determine the conditions which must be fulfilled for the acceptance of a software or a system. The definition of the acceptance criteria takes place in the test plan. General acceptance criteria are:   * each requirement should be linked with a test case * at least one test case should be executed successfully per requirement * none critical defect has to be open (defect status: new, open and fixed)   The definition of additional, more sophisticated acceptance criteria (component- or module-related) are possible and take place in the test concept as well.  e.g.  All relevant Tests must be executed at least one time.  All critical Defects must be handled before launching the application |

# Test activities

|  |
| --- |
| “Testing” includes all activities starting with the analysis of requirements, design of test cases, creation of test cases (realization), linkage of test cases with requirements, the execution of test cases as well as re-testing, monitoring of test execution and – last, but not lease – time schedule of all intended test activities.  The activity “adaption” includes adjustments concerning analysis, design, planning, realization and execution.  All relevant activities are planned in the test plan. Thus, this chapter has as strong impact on the project plan. The results of test planning have to be included in the overall project plan. Test planning can be included completely into the general project planning. In this case, the corresponding document has to be linked here.  The aim of test planning is to define time period and responsibilities for all test activities.  Please, take care of the fact that all defined steps in this chapter have a strong impact on the project planning. |





Time Scheduling

| **Activitiy** | **Start date** | **End date** | **Responsible person** | **Comment** |
| --- | --- | --- | --- | --- |
| Analysis, design and planning | 01.02.2015 | 31.12.2015 |  |  |
| Realization |  |  |  |  |
| Execution |  |  |  |  |
| Test evaluation |  |  |  |  |

Analysis and Design

|  |
| --- |
| In this chapter, the test manager verifies if the requirement/s is/are testable. The test manager defines test level/s and test type/s for each requirement.  Furthermore, the test manager verifies if there is already an existing test procedure for a requirement available. |

| **Activity** | **Start date** | **End date** | **Responsible person** | **Comment** |
| --- | --- | --- | --- | --- |
| Verify testability of requirements |  |  |  |  |
| Define test levels and test types for each requirement |  |  |  |  |
| Verify test procedure availability |  |  |  |  |

Realization

|  |
| --- |
| After analysis and design, the realization starts, which means, the creation of test cases has to be done.  All necessary activities are listed in the table. |

| **Activity** | **Start date** | **End date** | **Responsible person** | **Comment** |
| --- | --- | --- | --- | --- |
| Describe test procedure |  |  |  |  |
| Deliver test data |  |  |  |  |
| Assign test data |  |  |  |  |
| Link requirement with test case |  |  |  |  |
| Approve test case |  |  |  |  |
| Create test suite |  |  |  |  |
|  |  |  |  |  |
| Training of testers in the application under test |  |  |  |  |
| Training of testers in the test tool |  |  |  |  |
| Create specific authorization for testers |  |  |  |  |
| Setup infrastructure of test environment |  |  |  |  |

Execution

|  |
| --- |
| This chapter includes all activities for the first execution of tests as well as re-tests after handling of defects.  For both kinds of test execution, several test levels have to be considered. |

| **Activity** | **Start date** | **End date** | **Responsible person** | **Comment** |
| --- | --- | --- | --- | --- |
| Execute tests and document results |  |  |  |  |
| Execute re-tests and document results |  |  |  |  |
| Supervise test execution |  |  |  |  |

Test execution in detail

| **Test level**[[2]](#footnote-2) | **Activity** | **System[[3]](#footnote-3)** | **Time period** | **Resonsible person** | **Comment** |
| --- | --- | --- | --- | --- | --- |
| Component test | Functional tests | D |  |  |  |
| Security test: Static Code-Analysis | D |  |  |  |
| Supply system for test execution | Q |  |  | After release of system by developers and after preparation of all test necessary tasks/ master data |
|  |  |  |  |  |  |
| Integration test | Functional tests | Q |  |  |  |
| Performance test | Q |  |  |  |
| Load test | Q |  |  | To find out the productive system parameters |
| Training for testers | Q |  |  |  |
|  |  |  |  |  |  |
| System test | Functional Tests | Q |  |  | System test, ends with functional release (milestone MS2) |
| Move to P-System | Q->P |  |  |  |
| Performance test | P |  |  |  |
| Load test | P |  |  | To find out the productive system parameters |
| Dynamic security tests | D |  |  | As late as possible and with as less firewall protection as possible |
|  |  |  |  |  |  |
| Acceptancetest | Functional tests | P |  |  | Final acceptance test, ends with release for productive use (milestone MS3) |
| --- | GoLive | P |  |  |  |

# Test tools

|  |
| --- |
| Which tools are used in test phase for the different test types?  Please define in this chapter:   * Which test types will be executed with which test tools? * The reason when a test type is not included   The results of the different test types are reported during test phase in the test status report. With the end of the test phase, a final test report is created. |

|  |  |  |
| --- | --- | --- |
| **Test type** | **Tool** | **Comment** |
| Manual functional test  Automated functional test | HP Application Lifecycle Management (ALM)  HP Unified Functional Testing (UFT) |  |
| Security test: Static Code Analysis | HP Fortify SCA  Sonar |  |
| Performance test | Compuware APM aaS  Compuware synth. Monitoring  HP Performance Center |  |
| Load test / stress test | HP Performance Center |  |
| Dynamic security test | HP Fortify WebInspect  External Security Check |  |

# Persons and roles

Stakeholder

|  |
| --- |
| Which persons in the project are involved in which role and persons who are generally involved, but not for testing. |

Following table describes persons, which are involved in their role for this project and persons who are generally involved, but not for testing.

|  |  |
| --- | --- |
| **Role** | **Person/s – name and company** |
| Requester |  |
| Subject Matter Expert |  |
| Project leader |  |
| Escalation manager[[4]](#footnote-4) |  |
| Escalation team4 |  |
| IS-Operation |  |
| IS-Application |  |

Test Team

|  |
| --- |
| Which persons in the project are involved for testing in general?  Which persons are involved with which effort for the single test activities?  The following rules have to be defined to prepare, organize, execute and monitor the tests:   * Test manager (coordination, reporting) * Test designer * Test reviewer * IT Tester * Business Tester * Defect manager   Furthermore, test persons are planned for different tasks and at different times during test phase. They have to be listed here with name and field of duty. |

Following table describes persons and their roles in this project in their function as a member of the testing team.

|  |  |
| --- | --- |
| **Role** | **Person/s – name and company** |
| Test manager |  |
| Test designer |  |
| Test reviewer |  |
| IT Tester |  |
| Business Tester |  |
| Defect manager |  |
| Performance, load and stress test responsible |  |
| Security test responsible |  |

# Attachments

As attachment, you find the glossary and a list of all contact persons of this project.

Glossary

General terms around testing can be found in the actual glossary of the process „Test software and systems“.

|  |  |
| --- | --- |
| **Abbreviation** | **Meaning** |
| APC | Associated Production Center |
| AWS | Amazon Web Services |
| BRS | Business Requirements Specification |
| CIA | Architecture & Data Management |
| CII | Infrastructure & Network Management |
| DSW | Data Science Workspace |
| EH FL | Endress+Hauser Flow – Endress+Hauser Flowtec AG |
| EH LP | Endress+Hauser Level and Pressure – Endress+Hauser SE+Co. KG |
| FS | Functional Requirements Specification |
| PC | Production Center |
| SAT | System Acceptance Test |
| SMint | Smart Machine Interface |
| SRS | System Requirements Specification |
| SUT | System Under Test |
| TDB | Technical Database |
| TDB EP | Technical Database Electronic Production |
| UAT | User Acceptance Test |
| URS | User Requirements Specification |
|  |  |
|  |  |
|  |  |

Contact Persons for Test Topics

Following list contains persons involved in testing his project.

|  |  |  |
| --- | --- | --- |
| **Name** | **Company** | **Topics** |
| Isabelle Farina | InfoServe | General questions concerning testing  Functional tests  Performance, load and stress tests |
| Stéphane Le Caignec  Damian Schlager | InfoServe | Dynamic security tests |
| Lars Brößler  Ricardo Yanes-Yanes | InfoServe | Static security tests |

# Publishing [to approve]

This document is published in E+H Flow file share.

1. Standard 050: Alternatively to these two tables, it is possible to link the specified file storage, e.g. for project order (if general file storage for the project is described there in detail). [↑](#footnote-ref-1)
2. Test levels are described in detail in Standard 050 [↑](#footnote-ref-2)
3. The mentioned system environments represent the ideal situation. It has to be checked in every project if a test in the corresponding environment is possible or if an adjustment is needed. In case of adjustment, please give a reason for the change in the comment column. [↑](#footnote-ref-3)
4. Standard 050: Steering team [↑](#footnote-ref-4)