

Work-Package 7: "Toolchain"

## Toolchain Test Plan

Amaia Martija and Izaskun de la Torre

August 2014



Funded by:


 Federal Ministry  
 of Education  
 and Research

 Région de  
 Bruxelles-  
 Capitale

 GOBIERNO  
 DE ESPAÑA  
 MINISTERIO  
 DE INDUSTRIA, ENERGÍA  
 Y TURISMO

This page is intentionally left blank

**Work-Package 7: “Toolchain”****OETCS/WP7/D7.3  
August 2014**

# Toolchain Test Plan

**Document approbation**

Lead author:	Technical assessor:	Quality assessor:	Project lead:
location / date	location / date	location / date	location / date
signature	signature	signature	signature
Amaia Martija (SQS)	Michael Jastram (Formal Mind)	Marielle Petit-Doche (Systerel)	Klaus-Rüdiger Hase (DB Netz)

Amaia Martija and Izaskun de la Torre  
SQS

Tool chain Test Plan

Prepared for openETCS@ITEA2 Project

**Abstract:** This document describes the way in which the Open ETCS tool chain will be tested in the validation stage, the test strategy covers, the OpenETCS interface and the plugins it works with.

**Disclaimer:** This work is licensed under the "openETCS Open License Terms" (oOLT) dual Licensing: European Union Public Licence (EURL v.1.1+) AND Creative Commons Attribution-ShareAlike 3.0 – (cc by-sa 3.0)

THE WORK IS PROVIDED UNDER openETCS OPEN LICENSE TERMS (oOLT) WHICH IS A DUAL LICENSE AGREEMENT INCLUDING THE TERMS OF THE EUROPEAN UNION PUBLIC LICENSE (VERSION 1.1 OR ANY LATER VERSION) AND THE TERMS OF THE CREATIVE COMMONS PUBLIC LICENSE ("CCPL"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS OLT LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

<http://creativecommons.org/licenses/by-sa/3.0/>  
<http://joinup.ec.europa.eu/software/page/eupl/licence-eupl>

# Table of Contents

<b>Document Information .....</b>	<b>iv</b>
<b>1 Introduction to Toolchain Test Plan .....</b>	<b>1</b>
1.1 Introduction.....	1
1.2 Executive Summary .....	1
1.3 Intended Audience .....	1
1.4 Evolution .....	1
1.5 References, Guidelines and Standards .....	2
1.6 Definitions and Abbreviations .....	2
<b>2 OpenETCS Toolchain Test Plan .....</b>	<b>3</b>
2.1 Test Approach .....	3
2.1.1 Tests Execution Model .....	3
2.1.2 Tests Types .....	4
2.2 Test Items .....	7
2.3 Features to be tested .....	8
2.4 Item Pass / Fail Criteria .....	8
2.5 Test Environment .....	8
2.6 Test Deliverables.....	11
2.7 Schedule.....	12
2.8 Responsibilities .....	13
<b>Appendices.....</b>	<b>15</b>
.1 Test Plan Template .....	16
.1.1 Introduction .....	16
.1.2 Toolchain/Plugin Test Plan .....	16
.2 Test Specification Template .....	18
.3 Test Results Template .....	19
.3.1 Introduction .....	19
.3.2 Toolchain or plugins Test Summary.....	19

## Document Information

Document information	
Work Package	WP7
Deliverable ID or doc. ref.	xxx
Document title	Toolchain Test Plan
Document version	00.16
Document authors (org.)	Amaia Martija (SQS) Izaskun de la Torre (SQS)

Review information	
Last version reviewed	00.13
Main reviewers	Michael Jastram, Marielle Petit-Doche

Approbation			
	Name	Role	Date
Written by	Amaia Martija	WP7 participant	14.08.2014
Approved by			

Document evolution			
Version	Date	Author(s)	Justification
00.10	14.08.2014	Amaia Martija	Document creation
00.11	28.08.2014	Izaskun de la Torre	Creation of Sections 2.8 and Appendices Updated sections 2.6, 2.7
00.12	05.09.2014	Amaia Martija	Updated sections 2.1, 2.2, 2.7
00.13	19.09.2014	Izaskun de la Torre	Updated sections 1.5, 1.6, 2.1, 2.5
00.14	19.11.2014	Izaskun de la Torre	Updated sections 1, 2 and approbation table
00.15	20.11.2014	Izaskun de la Torre	Updated sections 2.2, 2.3, 2.5 and review table
00.16	04.12.2014	Izaskun de la Torre	Updated sections 2.1.2

# 1 Introduction to Toolchain Test Plan

## 1.1 Introduction

This document describes the way in which the Open ETCS platform will be tested in the validation stage, the test strategy covers, the OpenETCS interface and the plugins it works with.

## 1.2 Executive Summary

The main objective is to ensure the requirements of the toolchain, are properly provided, such the user functionality and the interoperability between the different plugins involved in the platform. The current document describes the strategy and objectives of the OpenETCS toolchain validation and details the validation test cases to be executed to the platform. The test cases have been design based on the functional specification and user's guides provided into github.

## 1.3 Intended Audience

The Tool chain Test Plan addresses all the stakeholders who are in the position to interact with OpenETCS tool chain.

- Project Manager
- QA Manager
- Assessors
- Tool chain WP Leader
- Tool chain Development Team
- Tool chain Qualification Team
- Tool chain V&V team

## 1.4 Evolution

This document will be updated regularly with the evolution of the OpenETCS tool chain. The methods and tools to be applied during the development of the OpenETCS toolchain products will be decided based upon the results of the research activities carried out and the needed of the rest of WPs.

The Tool Chain Test Plan document shall be updated whenever:

- tests or the approach for conducting them are changed
- strategies or methodologies used in the Verification and Validation processes are modified
- a new tool is added to the toolchain

- a new tool or technique is incorporated in any of the tasks
- a tool is removed
- a tool or a tool goal is modified

## 1.5 References, Guidelines and Standards

References	
Name	Version/ Edition/ Date
Tool chain Development Plan	00.03, 13.05.2014
Tool chain Qualification Process	1.15, 21.10.2014

Table 1. References

## 1.6 Definitions and Abbreviations

Notation	Description
MDT	Model Development Tools
OBU	On Board Unit
QA	Quality Assurance
TC	Test Case
WP	Workpackage

Table 2. Abbreviations



## 2 OpenETCS Toolchain Test Plan

### 2.1 Test Approach

This section describes the tests execution model, techniques and types of tests to be executed during openETCS tool chain development life cycle.

#### 2.1.1 Tests Execution Model

The openETCS tool chain development is divided into fortnightly sprints, on which objectives are defined and tasks that meet the requirements specified by the other workpackages and/or project goals and needs at that time (sprint) are analyzed, planned and developed. Each sprint is taken as a project in itself with their schedules, tracking daily meetings, etc.

In each iteration the scrum team evolves the product (makes an incremental delivery) from the results completed in previous iterations, adding new objectives / requirements or improving those that have already been completed. A key to guide the iterative and incremental development is the prioritization of objectives / requirements depending on the value provided to the openETCS partners and workpackages.

In the quality control stage of each sprint, unit testing, integration testing, performance, functional and acceptance testing will be made in the test environments ensuring the quality of the openETCS tool chain.

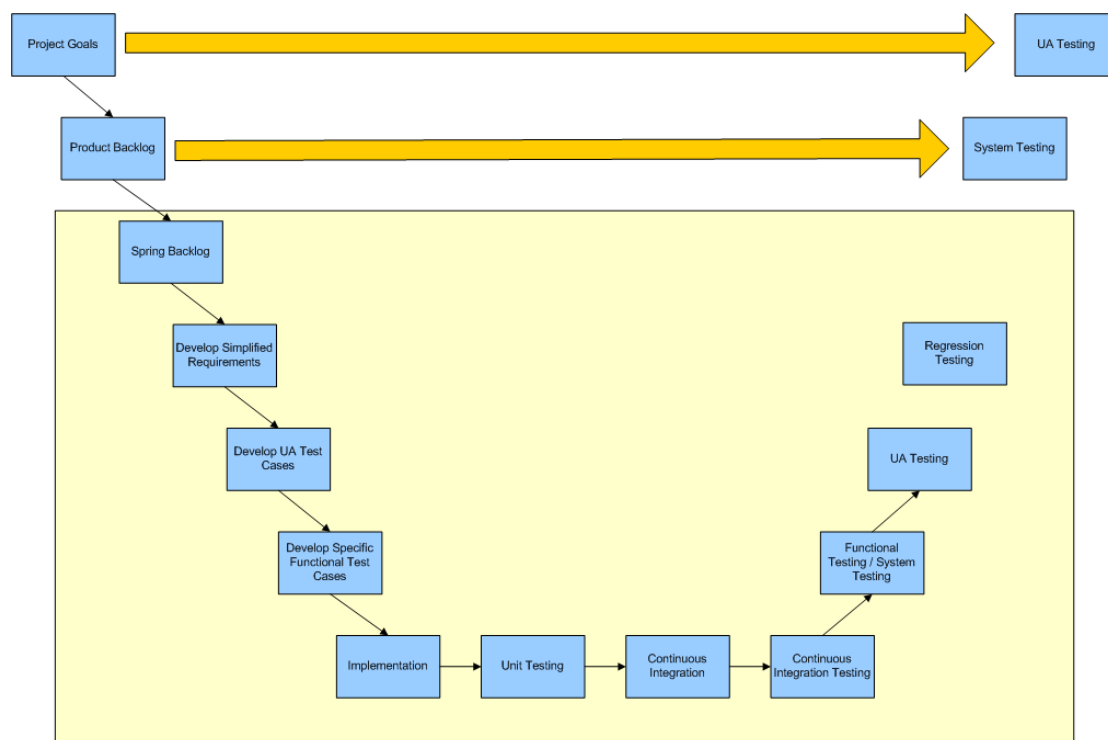


Figure 1. Types of test during the development life cycle

The fact of rerun every test every sprint is time consuming so in each sprint unit testing will be executed continuously and a good regression set based on risk classifications and business value will be collected and executed to give more time to specify and execute test cases for the functionalities implemented during the current sprint. All cross-sections along the subsets of test cases can form the total regression tests that are selected. So, a test suite has to be passed to assure that all functions and components are working and integrated correctly.

If the toolchain reach the stage of functional testing with too critical and / or blocking errors the tool chain will be returned to the unit testing stage

### 2.1.2 Tests Types

The validation has to demonstrate that the openETCS tool chain covers all the requirements and features.

At following the type of tests to be executed will be described.

Unit Tests	
<b>Objective</b>	Test the separate pluggings, and/or independent/isolated code sections.
<b>Method</b>	
<b>Tools</b>	JUnit, part of Jenkins
<b>Responsible</b>	Product Developer
<b>Execution Type</b>	Automatic
<b>Timing</b>	Continuously
<b>Evaluation criteria / Exit Criteria</b>	<ul style="list-style-type: none"> <li>• 100% Test Scripts executed</li> <li>• 100% pass rate of Test Scripts</li> <li>• No open defects</li> <li>• 50% code coverage</li> <li>• the reports generated by the test automation tools contains the minimum variables to allow a proper analysis of the evidence</li> </ul>

Integration Tests	
<b>Objective</b>	Validate that the OpenETCS platform and the pluggings are correctly installed, and their interoperability is correctly working. Validate the integration between different plugins and tools that make up the tool chain in order to ensure their integrated operation is correct.
<b>Method</b>	Bottom-up strategy
<b>Tools</b>	Maven

<b>Responsible</b>	Validation and Verification Team
<b>Execution Type</b>	Automatic and manual
<b>Timing</b>	<ul style="list-style-type: none"> <li>• Automatic tests: Continuously</li> <li>• Manual tests: only on official releases (every 6 months). For each release, a new column will be added to the table toolchain/tool/tests</li> </ul>
<b>Evaluation criteria / Exit Criteria</b>	<ul style="list-style-type: none"> <li>• 100% Test Scripts executed</li> <li>• 100% Test Cases executed in case on manual tests</li> <li>• 100% pass rate of Test Scripts</li> <li>• No open defects</li> </ul>

<b>System Tests</b>	
<b>Objective</b>	Validate non-functional requirements. During the System testing non functional requirements will be checked. These requirements are allocated in the [D2.6-9 deliverable].
<b>Method</b>	<ul style="list-style-type: none"> <li>• Develop test sets for performance, load (high data size), etc</li> <li>• During system testing, testing team will use preloaded data which is available on [Tests Data folder] on github.</li> </ul>
<b>Tools</b>	-
<b>Responsible</b>	Validation and Verification Team
<b>Execution Type</b>	Manual
<b>Timing</b>	After functional test is completed
<b>Evaluation criteria / Exit Criteria</b>	<ul style="list-style-type: none"> <li>• 100% Test Cases executed</li> <li>• 100% pass rate of Test Cases</li> <li>• No open defects</li> </ul>

<b>Functional Tests</b>
-------------------------

<b>Objective</b>	validating that the user's workflows and functional requirements provided on [documentation page] and on [D2.6-9 deliverable] are correctly created and to provide clear evidence that the platform performs as it should in every possible environment.
<b>Method</b>	During functional testing, testing team will use preloaded data which is available on github repository. Validation and execution of test set using valid and invalid data Develop a test set of the minimum requirements for the proper tool chain operation
<b>Tools</b>	-
<b>Responsible</b>	Validation and Verification Team
<b>Execution Type</b>	Manual. Test automation can be complex and it is only recommended for some specific functions (eg. Verification and validation of broken links)
<b>Timing</b>	The functional testing will be executed each three months in a deeply way by the testing team that will be created for the toolchain test. The functional testing will be a continuous activity also.
<b>Evaluation criteria / Exit Criteria</b>	<ul style="list-style-type: none"> <li>• 100% Test Cases executed</li> <li>• 100% pass rate of Test Cases</li> <li>• No open defects</li> </ul>

<b>Regression Tests</b>	
<b>Objective</b>	Validate tool chain still works perfectly after corrective actions or new functionality has been applied.
<b>Method</b>	Repeat tests subset (unit, integration, functional, system-load, performance,...-) to check modifications do not cause error where none had and to check that the new functionality do not enter errors. Pass a set of test to check that all functions are working and integrated properly
<b>Tools</b>	Same used tools for each specific test
<b>Responsible</b>	Validation and Verification Team or developer depending on the test
<b>Execution Type</b>	Manual and/or automatic
<b>Timing</b>	in each iteration

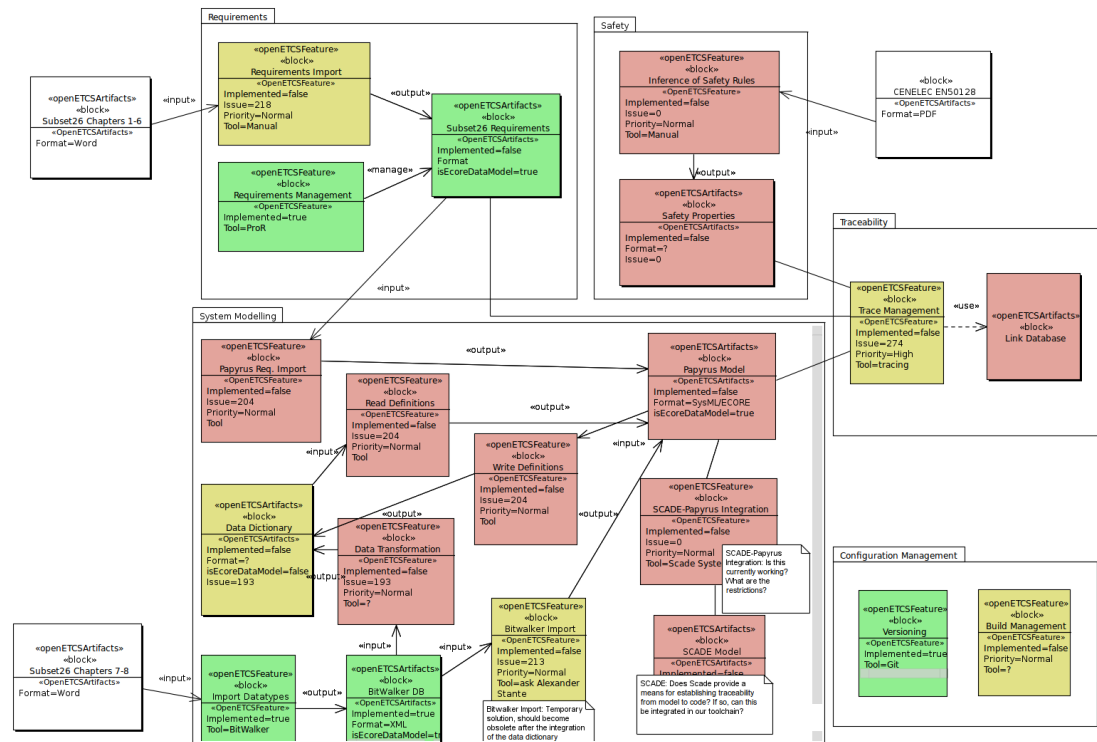
<b>Evaluation criteria / Exit Criteria</b>	<ul style="list-style-type: none"> <li>• 100% Test Scripts executed</li> <li>• 100% Test Cases executed</li> <li>• 100% pass rate of Test Scripts and Test Cases</li> <li>• No open defects</li> </ul>
--	--

<b>User Acceptance Tests</b>	
<b>Objective</b>	Check that designed user test cases allocated on [Tests Specifications] had the correct results.
<b>Method</b>	Validation and Verification team will write the UAT test cases based on the inputs from End User and project goals
<b>Tools</b>	-
<b>Responsible</b>	End Users
<b>Execution Type</b>	Manual
<b>Timing</b>	after all other levels of testing are done
<b>Evaluation criteria / Exit Criteria</b>	<ul style="list-style-type: none"> <li>• 100% Test Cases executed</li> <li>• 100% pass rate of Test Cases</li> <li>• No open defects</li> </ul>

<b>Exploratory Tests</b>	
<b>Objective</b>	Make sure critical or blocked defects are removed before the next levels of testing can start
<b>Method</b>	Carry out with out any test scripts and documentation
<b>Tools</b>	-
<b>Responsible</b>	Validation and Verification Team
<b>Execution Type</b>	Manual
<b>Timing</b>	in each iteration, once the build is ready for testing. At the beginning of each tests cycle.
<b>Evaluation criteria / Exit Criteria</b>	No open Critical and/or Blocked severity defects

## 2.2 Test Items

The OpenETCS tool chain consists of components. A Component is what the user perceives as an atomic tool aspect of openETCS. Some special Components satisfy infrastructure needs and are called Cross-Cutting Concerns. All the components are hosted by the tool platform (eclipse MDT). The OpenETCS tool page lists all the available components. A guideline to follow the defined life-cycle for OBU development will complete the tool chain.



**Figure 2. Tool Chain overview (20.02.14) –**  
**Green Block: Implemented**  
**Yellow Block: Work in Progress**  
**Red Block: Not started**  
**White Block: External Artifacts**

The features may be implemented by one or more tools and may also be implemented as plugins. All the features and plugins that make up the current tool chain and that should be tested are allocated in [Toolchain repository].

In addition to this, the person in charge of each test item is identified in the information allocated in [test management folder] as it is shown in Figure 4.

## 2.3 Features to be tested

The features to be tested are identified in [test specifications] folder.

## 2.4 Item Pass / Fail Criteria

A test is considered passed when the results obtained are the expected results shown in the Test Case. If any of the expected results are not met, the test is considered failed.

## 2.5 Test Environment

The environments where is going to be tested the toolchain are based on different operating systems:

- Windows 64
- Windows 32
- Linux 64
- Linux 32
- MacOS 64
- MacOS 32

To perform the testing activities of the toolchain the following tools will be used:

Tool	Functionality
GitHub	Configuration Management tool. This tool will be used to maintain under control all the selected configuration items (code, documentation,...), their versions and historial.
Issue Tracker	Tool for the Bug Management and Tracking. The errors found during testing activities will be reported in this tool
Jenkis	Open source continuous integration tool. This tool will be used to execute Unit tests.
Maven	Build automation tool. This tool will be used to execute some integration tests.
Eclipse	Eclipse is a development tool and will be used for test, in functional testing, in an easy and agile way the Open ETCS pluggins.
Openoffice cal	This tool will be used to specify the test cases.

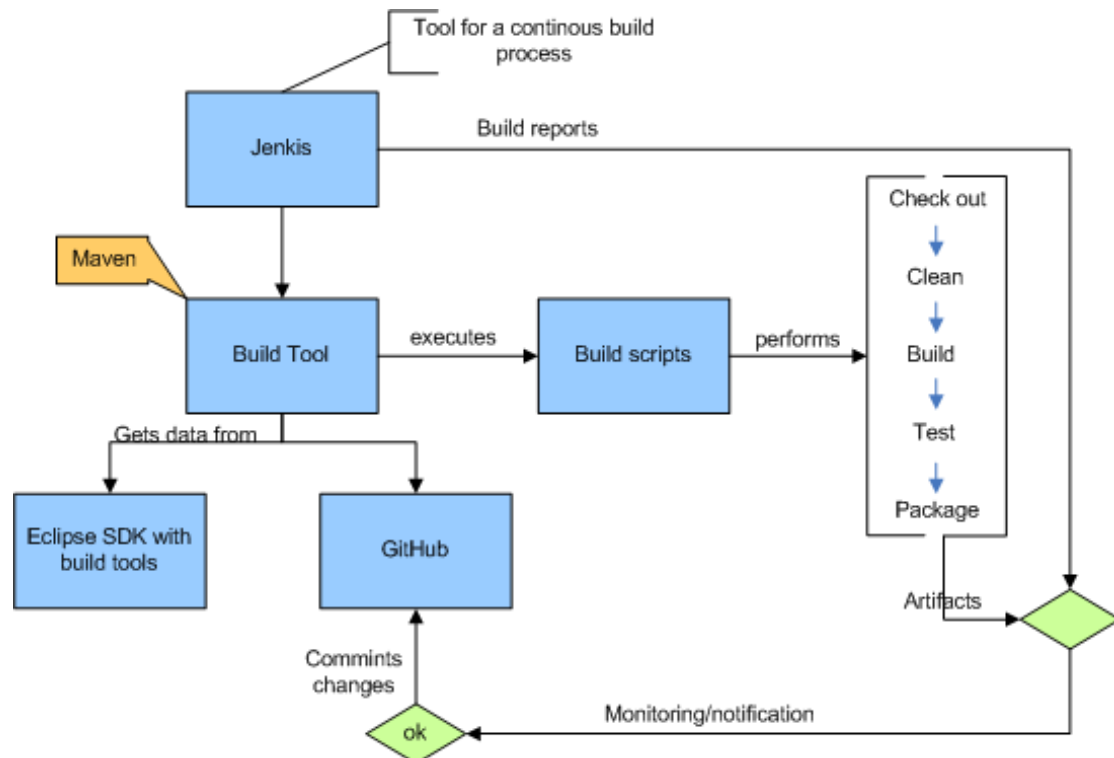


Figure 3. Test environment tools

As previously mentioned, GitHub will be used for configuration management. At following, the proposed structure to manage the testing process of tool chain supported by GitHub tool in order to assist in quality control and tool chain testing, to provide support for the planning and execution of the tests consistently, while keeping track of the requirements, versions, tasks, configurations and defect management, is shown:



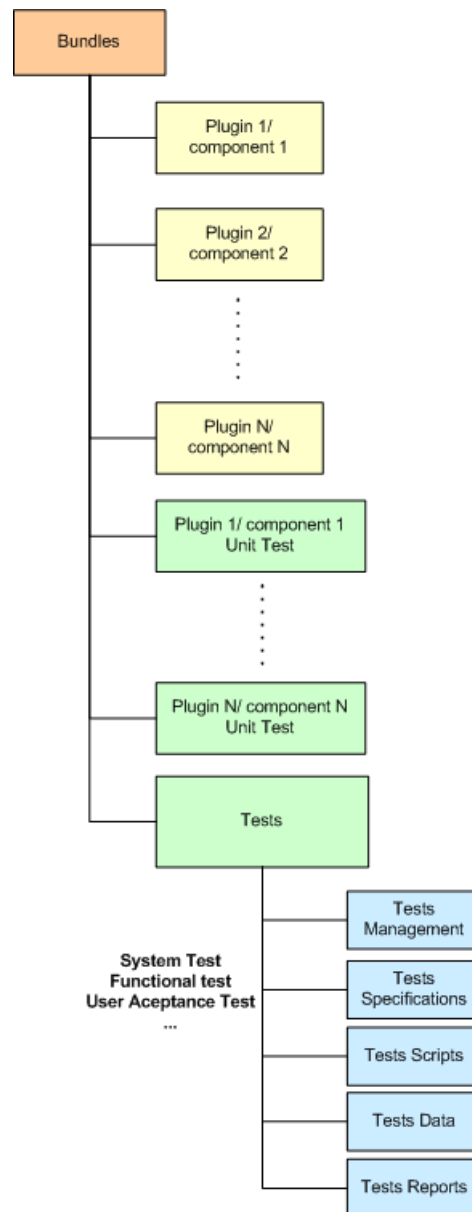


Figure 4. Test Information Management

## 2.6 Test Deliverables

Throughout the tool chain testing process a set of documents is created in order to keep track of the activities:

- **Test Plans:** A master test plan and sub-test plans for each plugin or feature will be developed. These documents will contain information about the scope, approach, objectives, features to be tested, resources, tools and schedule of testing activities. These documents must be prepared in accordance with the EN50128 Standard. A Test Plan Template is provided in Appendices .1
- **Test Specifications:** This document will contain all the information about the tests. For each of them, the ID, name, description and the requirements which the cases validate will be included. In addition, each Test Case will include information about the Entry and Exit

specification, a Description of the Event to be performed and other information such as type or Needs Test Environment for conducting the test. The Test Specification Structure is provided in the Appendices .2 and the test cases information is located in [Test Specification folder]

- **Test Results Reports:** The results obtained after the Test Cases execution will be collected in a Summary Test Report. For each test performed their unique identifier, the date and time which has been successfully executed or not and whether it is passed or failed state shall be indicated. A Test Results Report Template is provided in the Appendices .3. In case the tests are executed automatically by a tool that creates its own report, this will also be used as Test Result Report. An example of this last condition is the unit test result report which can be found in [Test Report folder].
- **Test Scripts:** All created scripts identified for use in test should be under configuration management tool.
- **Test Data:** All necessary data identified for use in test should be under configuration management tool. These data or the reference where they are are located in [Test Data folder]
- **Test Incident Report:** This document will contain the list of errors found during the execution on test cases. For each error will appear the record of the ticket live.
- **Test Logs:** This document will contain the logs of the system during the OpenETCS platform testing.

## 2.7 Schedule

The plan of tasks related to the activities of the toolchain tests is detailed below:

Activity	Start Date	End Date
Test Plan Elaboration	14/08/2014	19/09/2014
Test Plan Review	24/09/2014	26/09/2014
Test Plan Corrections	29/09/2014	30/09/2014
Test Plan Approval	01/10/2014	01/10/2014
Test Cases Specifications		
Documentation Plugin Test Cases Specifications	29/09/2014	02/10/2014
Tracing Plugin Test Cases Specifications		
Data Dictionary Plugin Test Cases Specifications		
Test Cases Review		
Documentation Plugin Test Cases Review		
Tracing Plugin Test Cases Review		
Data Dictionary Plugin Test Cases Review		
Test Cases Execution		
Documentation Plugin Test Cases Execution		
Tracing Plugin Test Cases Execution		
Documentation Plugin Test Cases Execution		

Test Results Report		
Documentation Plugin Test Results Report		
Tracing Plugin Test Results Report		
Data Dictionary Plugin Test Results Report		
Regression testing		

## 2.8 Responsibilities

Role	Responsibilities	Person in charge
WP7 Leader	<ul style="list-style-type: none"> <li>• Review and approval of the toolchain testing strategy, approach, and plans</li> <li>• Review of testing results report and defects to determine the impact to overall tool chain and plugins development and implementation schedule</li> </ul>	Michael Jastram
Testing Manager	<ul style="list-style-type: none"> <li>• Develop Test Strategy and Test Plan</li> <li>• Coordinate the development of testing deliverables</li> <li>• Review and approve testing deliverables</li> <li>• Monitor and report on the status of testing activities</li> <li>• Coordinate testing activities</li> <li>• Defect management</li> </ul>	
SW Development Team	<ul style="list-style-type: none"> <li>• Design Unit Test Cases</li> <li>• Execute Unit Test Cases</li> <li>• Review test cases and results for completeness</li> </ul>	Plugin or Feature Owner

Tool chain Validation and Verification Team	<ul style="list-style-type: none"><li>• Design Test Cases</li><li>• Execute Test Cases</li><li>• Elaborate Test Reports</li></ul>	
---	---	--

# **Appendices**

## **.1 Test Plan Template**

### **.1.1 Introduction**

#### **.1.1.1 Executive Summary**

#### **.1.1.2 Intended Audience**

*This section of the Test Plan should list the audience for which the document has been written. Also mention distribution restrictions and levels of confidentiality*

#### **.1.1.3 Evolution**

#### **.1.1.4 References, Guidelines and Standards**

*Provide a complete list of all documents and other sources referenced in the Software Test Plan*

#### **.1.1.5 Definitions and Abbreviations**

*Specify definitions of all terms and acronyms required to properly interpret the Test Results Report*

### **.1.2 Toolchain/Plugin Test Plan**

#### **.1.2.1 Test Approach**

*Identify the types of testing to be performed along with the methods and criteria to be used in performing test activities. Describe the specific methods and procedures for each type of testing.*

#### **.1.2.2 Test Items**

*Describe the items/features to be tested that are within the scope of the test plan*

#### **.1.2.3 Features to be Tested**

*Identify all toolchain features and combinations of features to be tested*

#### **.1.2.4 Item Pass / Fail Criteria**

*Specify the criteria to be used to determine whether each item has passed or failed testing*

#### **.1.2.5 Test Environment**

*Describe the tools and high level environment used for the testing activities*

#### **.1.2.6 Test Deliverables**

*Identify the deliverable documents from the test process*

#### **.1.2.7 Schedule**

*Identify the high level schedule for each testing task. Establish specific milestones for initiating and completing each type of test activity. Summarize when the testing activities will be done*

#### **.1.2.8 Responsibilities**

*Identify the groups responsible for managing, designing, preparing, executing, witnessing, checking, and resolving test activities*

## .2 Test Specification Template

Test Case Identifier:	
Test Objective	<i>Describe the purpose of the test case. Provide a brief description</i>
Test Items	<i>Describe the items or features (e.g., requirements, code, ...) to be tested by the test case</i>
Input Specifications	<i>Identify all inputs required to execute the test case.</i>
Test Steps	<i>Describe the series of individually numbered steps that are to be completed in sequential order to execute the test.</i>
Expected Test Results (Output Specifications)	<i>Identify all outputs required to verify the test case. Describe what the system should look like after the test case is run.</i>
Environmental needs	<i>Identify any environment requirement (e.g. operating system, tools,</i>
Inter-case Dependencies	<i>List any prerequisite test cases that would create the test environment or input data in order to run this test case. Also, list any post-requisite test cases for which the running of this test case would create the test environment or input data.</i>



### **.3 Test Results Template**

#### **.3.1 Introduction**

##### **.3.1.1 Purpose**

*Provide the purpose of the Test Result Document*

##### **.3.1.2 Audience**

*This section of the Test Results Report should list the audience for which the document has been written. Also mention distribution restrictions and levels of confidentiality*

##### **.3.1.3 References**

*Provide a complete list of all documents and other sources referenced in the Software Test Plan*

##### **.3.1.4 Definitions and Abbreviations**

*Specify definitions of all terms and acronyms required to properly interpret the Test Results Report*

#### **.3.2 Toolchain or plugins Test Summary**

##### **.3.2.1 Objectives, Scope**

*In the next section the objectives of the testing activities of the specific iteration are explained*

##### **.3.2.2 Methodology**

*In this section in which way the toolchain, and plugins requirements or features have been proven, the technique and tools that have been used, etc. will be explained*

##### **.3.2.3 Results**

*This section provides a summary of the results of the specific iteration testing of the toolchain and identifies all resolved issues. A list of requirements, test cases and the state, if passed or failed, will be created. In failed case the error number reported shall be identified*

Test Identifier	State	Release version	Execution Date	Requirement covered Identifier

##### **.3.2.4 Evaluation**

---

*This section provides an overall evaluation of the testing process*