

Guideline

for testing VCD System release 2.1



Log of Changes

Version	Date of Change	Changed By	Summary of Change
0.1	14.02.2012	Daniel Reiser	Initial version
0.2	02.03.2012	Daniel Reiser	* Inserted Schematron validation and test system information * Updated test focus section * Inserted cross-browser testing information

Purpose

This document provides a guideline for a common approach of testing the VCD System release 2.1. It defines the general workflow, tasks and tools to be used in order to execute the defined test cases.

Related Resources

#	Resource title	Reference
[1]	VCD Testing Framework	Word document
[2]	VCD 2.1 Test case summary	Excel document
[3]	Test Framework Overview	PowerPoint presentation
[4]	BSCW root folder for testing	BSCW folder
[5]	Concept database ¹	Excel document

¹ The concept data base provides definitions, translations, user explanations, business rules and implementation guidelines and is a core specification artifact relevant for testing the VCD System.

Table of Contents

Log of Changes	1
Purpose.....	1
Related Resources.....	1
1. Introduction	3
2. Execution of test cases and documentation of test results.....	3
2.1. Components	3
2.2. Focus.....	4
2.3. Input data	5
2.4. Tools	5
2.5. Test case assignments	6
3. Reporting of bugs and improvements.....	6
4. Test result analysis	7
5. Bug-fixing, implementation.....	7
6. Regression testing (invokes step 1.)	7

1. Introduction

The VCD testing framework (ref. [1]) already defines general aspects of the VCD testing activities. This document is complementing the framework by providing a common and structured approach to execute concrete test cases, to document executions and bugs or improvements and to analyze test results which are then input for a bug-fixing/implementation cycle.

The general testing workflow will consist of the following phases, each of them is further described in the subsequent chapters:

- 1.) Execution of tests cases and documentation of test results
- 2.) Reporting of bugs and improvements
- 3.) Test result analysis
- 4.) Bug-fixing, implementation
- 5.) Regression testing (invokes step 1.)

2. Execution of test cases and documentation of test results

The focused test groups are conformance testing/validation and functional-/non-functional testing of the VCD components. For each group, detailed test cases have been defined in pDAF. The capabilities provided by pDAF will be used by each test user to document the test case execution.

2.1. Components

The components to be tested are:

- VCD Designer
- VCD Builder
- VCD Viewer
- EVS-NVS service interface
- EVS reasoner components
- NVS-AT
- NVS-IT
- NVS-FR (see VCD Designer and VCD Builder)
- Ontology
- Schematron validation rules
- Generic Test Systems

2.2. Focus

The behavior and functioning of tested components shall be tested with respect to the following aspects:

- Functionality
 - Does the application correctly perform all required actions (requirements-driven testing according to defined user stories and functional requirements of each component)?
 - Does the application correctly process user input (both valid and invalid data) and visualize corresponding information (e.g. error messages, input validation messages, etc.)?
 - Are all information displayed on the GUI? Can detailed information be accessed?
 - Is the communication between EVS and NVS working correctly?
 - Is similar functionality implemented in the same way throughout different components (e.g. filter mechanisms to display criteria, evidences, etc.)?
 - Are implementation guidelines realized?
- Usability
 - Does the application display appropriate information (instructions, detailed information, etc.) so that a user knows what to do?
 - Are additional editing functionalities, search features, views, etc. required to better support the user in filling in his information or finding relevant information?
 - How long does the application need for a response after the user has triggered an action?
- Texts and translations
 - Is the information displayed in the correct language regarding the selection of the user language?
 - Is criteria and evidence information displayed in the correct language regarding EU-level, CA national and EO national level?
- VCD ontology
 - Is information regarding legal references, titles, names, etc. of evidences and criteria correct? Are translated values correct as well?
- Platforms and system environments

- Cross-browser testing: Are GUI elements displayed correctly in different web browsers?
- Tests should be performed at least for the most common browser versions: ²
 - Firefox 9.x, 10.x
 - MS Internet Explorer 8.x, 9.x
 - Chrome 16.x and higher
 - Safari 5.x
 - Mobile Safari 5.x
 - Opera 11
- Conformance testing and validation
 - Do the validation rules implement all required business rules?
 - Are test systems and Schematron rules correct?
 - Are created instances of VCD Package and VCD Package Skeletons valid regarding VCD Schema version 1.3.1?
 - Are T-/TC-/TCE-Skeletons exchanged between EVS and NVS valid?

2.3. Input data

In order to have appropriate input data for the test cases for VCD Designer/Builder, EVS GUI and NVS IT, the historical call for tenders elaborated by the legal task force should be used.

For the VCD Viewer test cases, VCD Package and VCD Package Skeleton test instances created during execution of aforementioned test cases should be used. In order to ensure validity of created test instances, each of them should pass the conformance/validation test cases.

2.4. Tools

pDAF as the general test case management tool is providing details about test cases and test steps. It is also used to document the test executions. For conformance testing and validation, the generic test systems shall be used complemented by appropriate tools supporting Schematron validation (e.g. Altova, Oxygen, etc.).³ Documentation of test execution is also done in pDAF.

² The different web browsers and versions have been selected on their popularity according to http://www.w3schools.com/browsers/browsers_stats.asp

³ The format in which the VCD Schematron rules are provided is currently not supported by the test systems, therefore external Schematron validators have to be used for Schematron validations. It is planned to update the test systems with Schematron support in the beginning of March. By then, the test systems will be capable of executing all defined tests and validations for TCE-Skeleton and VCD Package instances.

2.5. Test case assignments

The following table provides an overview about which test cases should be executed by which project partner:

Test area and test target	Responsible	Deadline
<Pdaf-Name/ID>		

3. Reporting of bugs and improvements

In addition to the documentation of test executions, the results have to be reported in JIRA. There, bugs and improvements are submitted for each component.

Information about each reported issue should consist at least of:

- Summary: General description of what is being reported
- Component: The test target
- Affects version: The version of the test target (Version 2.1 for software components, Version 1.3.1 for VCD Schema-related issues)
- The issue type shall be:
 - o “Bug” or “improvement” for software components
 - o “Request for change” or “missing translation key” for VCD Schema, Ontology or Translations (i.e. Concept DB)
- Description: Detailed description of the issue.
 - o Bugs shall be described in a way that it can be reconstructed, e.g. information about provided input data and other circumstances should be described. This should be complemented by attaching screenshots to the issue.
 - o Improvement: What is required to achieve something which is currently not supported?
 - o Request for change: Which additional elements or information are needed?
 - o Missing translation key: Which translation and value is missing?

Whenever an issue has been created, its ID needs to be referenced in the test execution report in pDAF. For this, pDAF provides a corresponding input field to refer to bugs in JIRA (Atlassian JIRA) and to provide the ID of the created issue, which is the very last part of the URL of the created JIRA issue.

For each discovered bug, the pDAF test execution failed, meaning that the corresponding checkbox has to be activated in the test execution report page displayed in pDAF. If information elements or translations are missing, this should also be reported as “failed” in pDAF. Improvements have to be treated differently, i.e. the test execution succeeded, however the link to the improvement issue should be provided in the commenting fields of the pDAF test execution report.

4. Test result analysis

After all tests have been executed, a compound analysis of all test results will be performed. This includes:

- General net meeting, in which each test user should report about his/her test case executions, reports and issues.
- Final assignment of each issue to the correct component is required (e.g. in case a bug has been reported for the wrong component because its actual source is in that other component).
- Check if all issues have the correct type and contain all required information
- Discussions and approvals / rejections of reported issues in different task forces

5. Bug-fixing, implementation

Approved issues will be implemented in software components, VCD Schema update, Concept DB and language file update.

6. Regression testing (invokes step 1.)

Test shall be executed again in order to ensure that bug fixes, improvements and RfCs have been implemented correctly and that they do not cause malfunctions in other parts of the components. Accordingly, the subsequent phases after step 1 have to be executed again.