Numericanal Project

Technical Evaluation of the Pilots

Functional Tests Scenarios

Authors	Sanae Saadaoui	30/10/2015
Verification	Mohamed Boukhebouze	02/11/2015

CETIC ASBL – Centre d'Excellence en Technologies de l'Information et de la Communication
Aéropole – Rue des Frères Wright n°29/3 - 6041 Charleroi (Belgique) – Tél. : +32-71-490 700 – Fax : +32-71-490 799
e-mail :info@cetic.be- TVA BE 0474 549 932

Document Control

Creation Date	04/06/2015
Last Modification Date	11/02/2015
Version	1.0

Document Versions History

Version	Date	Description	Authors
0.1	01/07/2015	First Draft of the document	Sanae Saadaoui
0.2	30/10/2015	Second Draft of the document	Sanae Saadaoui
1.0	02/11/2015	Review the document	Mohamed Boukhebouze

References

V&V	Verification and Validation

Abbreviations

V&V	Verification and Validation

Confidentiality Level

Public	
Restricted to a specific audience (Numericanal Project Members)	**x**
Confidential (for CETIC members only)	

Contents

1. ***Executive Summary*** ____ ***4***
2. ***Introduction*** ____ ***5***
3. ***Functional Tests Scenarios*** ____ ***6***

3.1 Test Case Design ____ **6**

3.2 Requirement Analysis ____ **6**

3.3 Test Case Template ____ **7**

3.4 Tests cases ____ **9**

3.4.1 Pilot of SRS ____ 9

3.4.2 User tests ____ 10

1. Executive Summary

The main objective of the NUERICANAL project is to enhance safety matters for pleasure boats and to offer them facilities and logistical help when needed. The ultimate goal of the project is to give the basis for an integrate system to make communication and connexion between all pilot systems possible.

In this document, we presented various test cases for the NUMERICANAL pilot's software. We first describe the Verification & Validation goals, as well as the methodology used for designing the test cases.

In order to obtain software architecture of good quality, a verification and validation process is used to enhance the productivity of the developers and improve the quality of the final outcome, a prototype or product. The verification and validation process (a generic term for the checking procedure which ensures that the software meets the original requirements and those of the software owner) proposes a continuing evaluation, with two objectives: the discovery of defects in the system, and the assessment of whether or not the system is usable in an operational situation. This can be done in three or more cycles, alongside with the development of the software components, with improvements and more details as the project takes place.

2. Introduction

In the following, some scenario tests are described as an example for the SRS pilot based on the use cases described in the SRS document.

Due to very high level of requirement description, the document gives a template and an example of how to insatiate it for real cases. This will help each partner to apply it to its pilote depending based on more detailed requirements.

The pilot one is the pilot of SRN which objective is to give the localisation of dangerous zones and some recommendations on how to navigate near them. The main users are pleasure boats navigating in the Amsterdam region.

The second pilot is the pilot of VNF which purpose is to develop an application to manage events on inland waterways and to offers different services to the ships crossing the France waterways.
The third pilot is the Eijsden-Margaten which purpose is to facilitate leisure boats life by offering them an AIS-like system and provide services to help them in their logistical issues.

3. Functional Tests Scenarios

3.1 Test Case Design

The "Numericanal_SRS" document presents the specifications of the four pilots of the Numericanal project. The level of detail is not homogeneous because it depends on the available information on the pilots. All the pilots are not at the same level of development.

3.2 Requirement Analysis

A very high percentage of software projects fail because of an incomplete study of requirements. Important factors have to be taken into account:

- Lack of contribution of the end user.
- Requirement changes.
- Requirements can be time-sensitive.
- Unclear requirements, with multiple sources.
- Requirements are sometimes difficult to express in words.

Every requirement has to be matched by one or more test cases. For this, a simple checklist questionnaire can be answered for every requirement; this will yield a deep analysis of the requirements. This is done for the higher-level requirements, which encompass the lower level requirements. Recall that the document from the requirements specification phase identifies what the software is intended to do. Thus the requirements should also :

- Be traced to the user's needs.
- Completely satisfy the capabilities specified in the concept document.
- Be evaluated for their technical merits.
- Identify all external and internal interfaces to the software.

Category	Description of Requirement
1. Clarity	Are the requirements clear and unambiguous?
2. Standards	Have all requirements standards been followed?
3. Completeness	Are all the requirements complete?
4. Level of Detail	Are the requirements not affected by design constraints?
6. Functionality	Are functions correctly specified? Are inputs and outputs clearly specified? Are functions logically independent and highly cohesive?
7. Performance	Are the performance requirements for timing, memory, and resource utilisation clearly defined?
8. Testability	Are the requirements testable and verifiable?
9. Modifiability	Are the requirements uniquely structured such that any necessary changes to the requirements can be made easily, completely, and consistently?

Table 1 Requirements Analysis Criteria

3.3 Test Case Template

The following template should be used when creating a new test case. It contains all the useful information needed to specify a test. The grey rows may appear after the test has been defined. They may even only appear in the ultimate iteration, when testing the final prototype, as the code needs to be written to have a proper schedule and owner.

The following template can be applied to any type of tests, integration tests, system test etc.

Test Case Number. Version	Ordinal of the test case, no special numbering policy is enforced **Example**: # 1.1
Test Case Title	Short name that describes the test case **Example**: End-Use encoding the details of a new event
Module tested, **Keyword**	Name of the module (or a big functionality) to be tested **Example**: Event Manager
Related Requirement(s) or Use Case	The name of the requirement (or functional or a non-functional specification) this test case is used for.
Target	Short description about the topic that we are going to test.
Initial Conditions	Initial conditions that we need to set before to start the test
Expected Results	List the results you expected to happen when the test is run.
Test Case Status	Status of the test case. **Possible values**: Pending, In Progress, Executed, or Unknown (N/A).
Owner	This is the person ultimately responsible for the test case.
Assigned	Person the test case is assigned to.
Steps	List of the exact steps that the assigned person must follow to perform the test case.
Feedback receiver	This is the person to notify with feedback of the test case.
Start date	Initial date of the test case.
End date	End date of the test case.
Comments	Additional optional information on the test case
Conclusions	When test results require explanation, the optional "Conclusions" field may be added

Table 2 Test Case Template

3.4 Tests cases

The following table lists all test cases that can be applied to the pilots and briefly describes possible restrictions due to planned implementation.

3.4.1 Pilot of SRS

Number	Title	Restriction
1.1	DZ menu is displayed to the End-user in the portal	Fully applicable
2.1	Risk information shown to the End-User	Fully applicable
7.1	Provider including risk into the SLA offer	Fully applicable
8.1	Provider engaging in Fault-Tolerance	Fully applicable
30.2	For readability, initial Test Case **"Provider receives service requests"** has been divided in two. Test 30.2 is **"Provider receives service requests (from the Broker layer)"**	Fully applicable for D2.1
51.1	Software Accuracy	Partially applicable in WP2

T able 3 Test Cases applicable on SRS PILOT

3.4.2 User tests

Test Case Number.Version	**1.1**
Test Case Title	DZ Administration menu is displayed to the Administrator-user in the application
Module tested, Keyword	Collect data on Dangerous zones DZ
Related Use Case(s)	UC001; UC002 ; UC003; UC004; UC008
Target	The system displays the Administration menu to allow the user to administrate in a simple way data on dangerous zones.
Initial Conditions	The administrator is logged.
Expected Results	The application displays the Administration menu with the name of the logged administrator depending on his access wrights
Test Case Status	UNKOWN
Owner	SRS developer
Steps	1) Start by selecting Display Administration Menu give a SCREEN showing the operation **Result**: The administration menu is displayed Give a SCREEN of the displayed Administratio Menu

Table 4 Test Cases applicable on SRS User