AN APPROACH FOR FULL SCALE OFF-LINE TESTING TO EVALUATE THE IWIDGET SYSTEM PERFORMANCE

PAULA VIEIRA[1], JOSÉ BARATEIRO[2], DÁLIA LOUREIRO[3], JOÃO COELHO[4], AISHA MAMADE[5]
[1] Laboratório Nacional de Engenharia Civil, Lisbon (Portugal), pvieira@lnec.pt
[2] Laboratório Nacional de Engenharia Civil, Lisbon (Portugal), jbarateiro@lnec.pt
[3] Laboratório Nacional de Engenharia Civil, Lisbon (Portugal), dloureiro@lnec.pt
[4] Laboratório Nacional de Engenharia Civil, Lisbon (Portugal), jpcoelho@lnec.pt
[5] Laboratório Nacional de Engenharia Civil, Lisbon (Portugal), aisha@lnec.pt

ABSTRACT

WIDGET is an ongoing European Commission FP7 project aiming at improved water efficiencies using novel ICT technologies for integrated supply-demand side management, focusing on an integrated approach to water resources management. The project contributes to advance knowledge about smart metering in order to develop novel, robust and cost-effective ICT tools for both water utilities and consumers. Within the project, a set of relevant applications derived from smart metering real-time data was identified, characterized and implemented in a software prototype. This prototype is a critical asset that must be evaluated to ensure its compliance with the requirements specified during system analysis and design and to verify its acceptance according to the end users' needs. The prototype evaluation followed the software engineering best practices, through a set of software tests with close collaboration between consumers, utility stakeholders and software developers. Once individual components (data management and analytical) have passed unit testing, integrated tests took place. The first phase of integrated tests used off-line historical smart metering data as opposed to the second phase (on-line testing) that uses near real-time data. This paper presents the standardized method designed in the project to carry out the off-line testing. The off-line testing method is based on a test scenario/test case approach and includes functional (i.e., tests to verify if functional requirements are met) and non-functional testing (i.e., tests to verify the quality of the software in terms of usability, security or compatibility, for example). For each evaluated application, test scenarios were designed as well as the corresponding test cases (in total, more than 50 test scenarios and 90 test cases). Success criteria for determining whether an observed behaviour of the system is correct and key performance indicators (KPI) to assess the achievement of success criteria were also defined for each test case step. This paper also presents results of the method´s application, which included collecting historical data from a full-scale case study, feeding this data to the prototype, analysing the results and evaluating KPI in order to identify corrections and improvements.

Keywords: Off-line testing, smart meter, test case, water use efficiency

1. INTRODUCTION

iWIDGET is an European Commission funded FP7 collaborative project, aimed at improved water efficiencies through the use of novel ICT technologies for integrated supply-demand side management. The iWIDGET focus is an integrated approach to water resources management and the project intends to contribute to delivering a sustainable, low-carbon society, helping progress towards the Europe 2020 targets on Climate and Energy.

The main scientific challenges for iWIDGET are the management and extraction of useful information from vast amounts of high-resolution consumption data, the development of customised intervention and awareness campaigns to influence behavioural change, and the integration of iWIDGET concepts into a set of decision-support tools for water utilities and consumers, applicable in differing local conditions. In order to meet these aims and challenges, iWIDGET develops research to address the following challenges:

- how best to provide the dynamic accurate measurement and data transfer of useful information about end-user water consumption;
- how best to use consumption data to improve the operation of utilities and influence end-users to modify their behaviour;
- how to arrive at the best business model to convert a promising technology into a useful and cost-effective product;
- how to demonstrate and validate the new methodologies on three case studies in the UK, Portugal and Athens.

This paper presents the main method used for offline testing, which intends to ensure that the software components that are release to the end users comply with the analysis and design requirement, for both functional and non-function requirements. Indeed, it aim is to design and carry out real life full scale testing of the iWIDGET systems in close collaboration with households and utility stakeholders.

In fact, as recommended by current best practices in information systems, such as TOGAF (*http://www.opengroup.org/togaf*) ISO and IEEE standards (e.g., ISO 42010, IEEE 1471), Archimate (*http://www.opengroup.org/subjectareas/enterprise/archimate*), and current works on Enterprise Architecture, or even references on requirements engineering and system analysis, such as the BABOK Guide or the works led by Klaus Pohl, it is clear that designing a new system requires a systematic analysis where we need to involve several stakeholders, analyze their problem, separate their concerns and, only after understanding the problem we can design an adequate solution. Otherwise, the risk of having a software application that does not fit the requirements, needs and business processes of the stakeholders is very high. Technically, this is known as a strategic alignment problem (we can detail that in cross-layer misalignments, with a strong focus on vertical misalignments).

To reduce these risks the software engineering, information systems and architecture communities developed several methods to produce artifacts that represent the system architecture (design). In addition, it is critical develop methods for testing if the implemented solutions comply with the requirements and specifications. This is the core of this paper, where we detail the methodology developed in the scope of iWIDGET to test the software prototypes implemented in this project.

A standardized methodology for performing a comprehensive off-line testing of the prototype iWIDGET system was developed. The project partners, using historical data from a real case study from Barcelos case study (Portugal), applied this methodology. The historical data set was migrated and loaded into the iWIDGET target database, making it possible to run the methodology on top of real/historical data. Applying the methodology, the off-line testing produced a set of results from functional and non-functional testing procedures, which were then analysed and evaluated against the defined performance indicators in order to identify improvements.

In fact, iWIDGET can be seen as a system with two major components: consumer module, specifically designed to provide functionalities to the consumer; and water utility module, specifically designed to address the needs of water utility specialists. As such, since the needs and profile of end-users are slightly different, the off-line testing is also decomposed between the consumer domain and the water utility domain.

This paper focuses on presenting the methodology for off-line testing. To demonstrate its usage, we focus on specific examples and present results from functional tests. Note that the objective is not to highlight on the results achieved from the testing, but rather on the testing method.

The remaining of this paper is organized as follows: First, section 2 details de general methodology for off-line testing. Second, section 3 explains the adopted design for off-line testing, explaining the objectives, type and setup for tests. Then, second 4 illustrates some results, explaining how they were used to improve the iWIDGET prototype. Finally, we conclude in section 5.

2. GENERAL METHODOLOGY

The methodology for off-line testing of the iWIDGET system (Figure 1, Vieira et al., 2013) is based on a test scenarios/test case approach and comprises three main phases: i) preparatory work, ii) off-line tests using historical data and iii) evaluation of the performance of the iWIDGET system during the tests.

In preparatory work, objectives for off-line testing were set and the type of tests to be carried out was defined. Previously, use cases were defined in the project and, for each use case to be tested; the test scenarios where the components were to be tested were defined. Finally, the test cases corresponding to each scenario were specifically designed and described. A test scenario specifies what is the functionality of the system that is to be tested. A test case specifies how it must be tested in detail.

During the off-line test itself, historical data was collected from the Portuguese project case study, fed into the iWIDGET prototype and results were recorded and analysed according to the evaluation criteria established in step iii).

Evaluation work consisted in the definition of success criteria related to the objectives of off-line testing. In order to assess the achievement of the success criteria, key performance indicators were developed and calculated. This performance assessment allowed identifying faults and potential improvements of the iWIDGET systems, which were reported as recommendations to the software development team within the project.

For off-line testing the following pre-conditions were fulfilled:

- A set of consolidated and harmonized of use cases and corresponding functional and non-functional requirements existed.
- Individual components (data management and the analytical components) were ready in prototype version and integrated in the prototype for off-line testing.
- Individual components have passed unit testing.
- Historical data has migrated into the off-line testing prototype database. Data used for off-line testing was a static dataset.

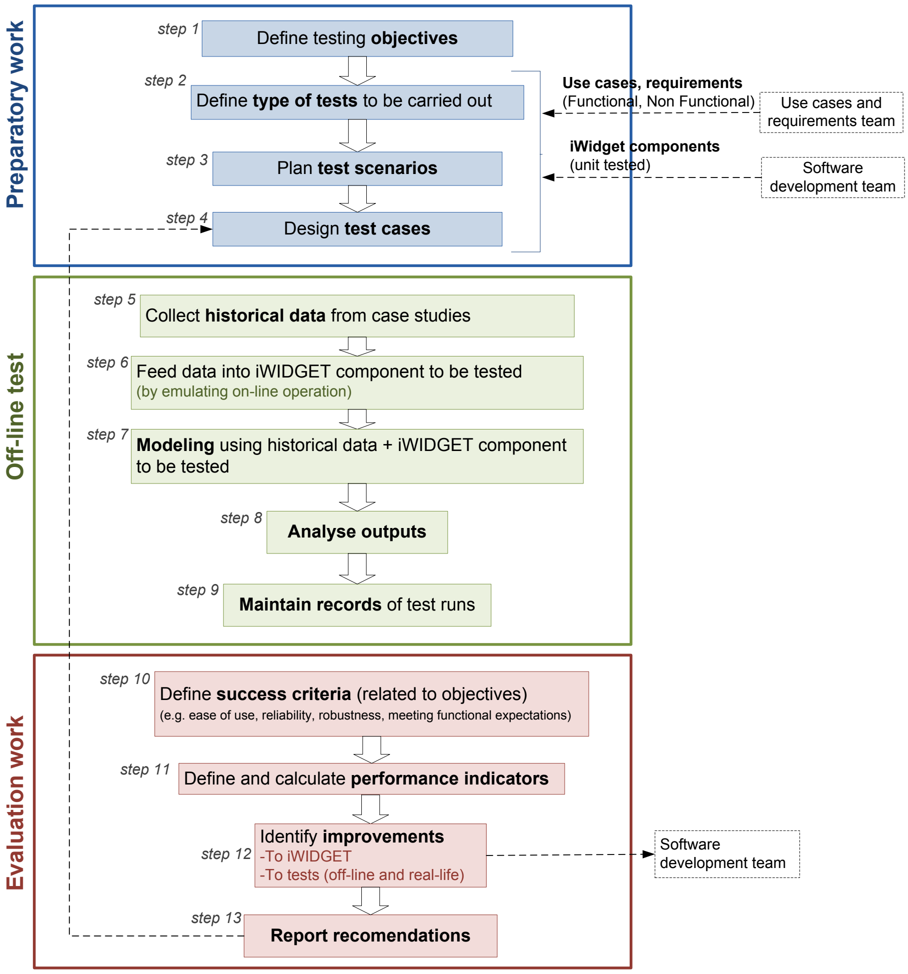

Figure 1 – Methodology for off-line testing of the iWIDGET system

3. DESIGN OF THE OFF-LINE TESTS

3.1 Objectives and type of tests

The main objective of off-line testing was the evaluation of the operational readiness of the iWIDGET system with historical data, previously migrated to the off-line testing prototype database. Specific objectives were to test the system in meeting the requirements of each use case and to evaluate the resources required to run the iWIDGET system to meet those requirements.

Following the objectives, black box tests were performed on a system level. Black box tests examine the functionality of the software without peering into its internal structures or algorithms.

Off-line testing included functional (F) and non-functional (NF) testing. Functional tests verify if functional requirements are met, by feeding input to the functions and examining the output. Non-functional tests verify compliance with non-functional requirements and, in the case of iWIDGET, assessed the quality of the software in terms of loading, performance, compatibility, scalability, usability, documentation, security and availability. Considering the objectives of the iWIDGET project, an option was made not to perform extensive NF testing but to select tests that ensured that the basics the aspects important for a successful online testing were covered.

Functional tests were derived from use cases and aimed to test the main functionalities of each use case, i.e., to evaluate functional requirements, tests were based on scenarios defined for each detailed level use case. Non-functional tests were derived from requirements, i.e., to evaluate non-functional requirements, tests were directly driven by and checked against requirements in the form of scenarios (requirements-based testing); NF tests were done using selected use cases (e.g. more complex UC in terms of data processing). Performance indicators were linked to the requirements.

In total, more than 50 test scenarios and 90 test cases were designed. Figure 2 shows an example of a test case and corresponding KPI. The complete set of test scenarios and test cases can be found in Vieira et al., 2013. Each test case was characterized with the following information: test case ID, corresponding test scenario, test case description, sequence of steps, related requirements that have to be met, tester profile required.

Use Case	Test scenario	Test case	Test case steps	Requirement	Success criteria	Key Performance Indicator
WU_UC02.1: Obtain real-time water balance data	#1 Obtain real losses for a specific month	#1.1	1. Log on to iWIDGET using a password	Logon	Successful logon without errors	Pass/fail
			2.Select network sector	Network sector selection	Successful selection of option	Pass/fail
			3. Choose a time period from a calendar	Time period selection	Successful selection of option	Pass/fail
			4. Select the components of the water balance to be calculated	Water balance components selection	Successful selection of option	Pass/fail
			5. iWIDGET calculates the result	Execute calculations	Correct result is calculated	Pass/fail
			6. iWIDGET presents a table and a report with the results	Display information	Correct display of results	Pass/fail
			7. Print the report	Print	Successful print of report	Pass/fail
			8. Save data	Download	Successful data saving	Pass/fail

Figure 2 – Test case for the iWIDGET system (example for a functional test)

Each step in the test case is related to the specific requirement from the iWIDGET analysis. In fact, the traceability between requirements and test case steps adds critical information for further analysis. For instance, there might be steps that fail due to requirements that were underestimated. If a test case fails in such scenario, the decision maker can provide an informed decision to determine the requirement must be implemented to correct the failure or, as an alternative, might change the requirement from the analysis specification. Indeed, it is critical to separate the analysis (problem space) from the design (solution space) avoiding misalignments between the problem and the solution that is being implemented. That fact is explicit by the relation between the requirement (problem space) and the testing which is validating a prototype (solution space) against the desired requirements.

3.2 Setup for the tests

The tests were carried out by several project partners (LNEC, UNEXE, NTUA, AGS, IBM and SAP). To have some comparability of results, at least two partners tested a certain widget. For the utility domain, partners involved in testing had the scientific knowledge to assess the correctness of results. The water utility partner tested all widgets in the utility domain. For the consumer domain, testing was also made by the remaining testing partners. In some cases, the same partner who developed the widget also tested it. In this case, tests were done by team members who did not develop the software.

Different testers profiles were defined following the actors identified in the use case description: consumer, water utility - billing and customers management staff, water utility - maintenance staff, water utility - network operation staff, water utility - public relations and communication staff, water utility - strategic, tactical and operational planning staff).

As persons from outside the iWIDGET project did some of the tests, thus having access to data from case studies, data confidentiality issues were considered. To safeguard data confidentiality, testers who were not part of the project team did not know the origin of the historic data that they had access to.

The execution of a functional off-line test was composed of two parts: the functionality test itself and the usability enquiry, where users report about their experience when testing the system.

During the functionality test, tasks were executed by users (the testers) instructed to carry out a sequence of predefined steps according to the test scenarios and test cases previously defined. When the tester was someone from outside the project, he was accompanied by a project partner team member, called the observer. The observer took notes and recorded user testing.

Usability enquires use subjective measures to gather feedback from the users perspective on the more qualitative aspects of the tests related to users expectations. The usability enquiry comprised the execution of a standardized questionnaire using a slightly modified version of the System Usability Scale (SUS) approach (Brooke, 1996; Bangor et al., 2009) widely used in the IT domain. After using the system under test, the user filled a questionnaire rating the answers according to a 5-point Likert scale. Figure 3 presents the enquiry made to each user.

Use Case: *use case ID and name*
Tester (name/organization):
Date of test run:

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	1	2	3	4	5
2. I found the system unnecessarily complex	1	2	3	4	5
3. I thought the system was easy to use	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	1	2	3	4	5
5. I found the various functions in this system were well integrated	1	2	3	4	5
6. I thought there was too much inconsistency in this system	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	1	2	3	4	5
8. I found the system very cumbersome to use	1	2	3	4	5
9. I felt very confident using the system	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	1	2	3	4	5
11. I found the user documentation comprehensive, appropriate and well-structured	1	2	3	4	5
12. When using the system, I would need more support by e-mail or phone.	1	2	3	4	5

Figure 3 – Usability enquiry for off-line tests

Off-line functional testing was done in two phases so that end-users (consumers and water utilities staff) tested a version, which had already been corrected for the main faults:

- Phase 1) Testing done by project partners team members not directly involved in the software development, i.e, in code writing.
- Phase 2) Testing done by end-users with adequate profile: for the consumer domain, widgets testers are domestic consumers; for the utility domain, widgets testers are staff members from water utilities.

Non-functional tests were carried out only by the project partners' team members and not by end-users.

The historical data used for off-line testing came from Barcelos case study. Barcelos is a town located in Northern Portugal. The case study consists of three District Metered Areas within the town. Network flow and pressure data are available from a SCADA system for these areas; alongside with telemetry systems that collect real-time water consumption data for 311 households. This data was provided by project partner AGS (the water utility in Barcelos) and covers a 2 years period starting from January 2009 to October 2010. For household analytics testing, water consumption data from households was used. The raw data is cumulative water consumption in litres, logged in with a 15-minute time step.

4. RESULTS FROM TESTS

Figure 4 presents an example of the results from the off-line functionality tests for a specific use case. These functionality tests were performed according to the procedure described in section 2. As can be seen in the Figure, results include the number of times the test case was executed, the number of total failures, the steps where failures occurred and the final status of the test case, i.e. if it passed or if it failed in the test. According to the criteria used during this off-line testing, a test case fails if a failure occurred in at least one of its steps.

Considering all test cases for the consumer domain, the most common types of failures found during the tests were related to print and save/download data. These failures occurred in most of the use cases. Other failures were related to the following requirements:

- Selection of options;
- Data input;
- Display of results;
- Correct calculation of results.

These results were communicated to the analysts and software developers to implement corrective actions. Note that this can be done either on the problem space (requirements definition) or on the solution space, developing the software components to correct or add functionalities to the iWIDGET system.

Use Case ID	Test case ID	# Executions	# Failures	# Passes	Steps where failing	Status
C_UC01.1	1.1	3	3	0	4. Select 15-minutes resolution from the time resolution list 8. Print the report 9. Download data	FAIL
	2.1	3	3	0	8. Print the report 9. Download data	FAIL
	3.1	3	3	0	9. Print the report 10. Download data	FAIL
	4.1	3	3	0	9. Print the report 10. Download data	FAIL
	5.1	3	3	0	9. Print the report 10. Download data	FAIL
	6.1	3	3	0	9. Print the report	FAIL
	7.1	3	3	0	9. Print the report	FAIL
	8.1	3	3	0	8. Print the report	FAIL
	9.1	3	3	0	9. Print the report	FAIL

Figure 4 – Example of functional test results

As for the consumer domain, 20 use cases from the water utility domain that were also off-line tested. From those test cases, several failures were also detected, especially concerning the following requirements:

- Selection of input options;
- Selection of analytics type;
- Display of results;
- Correct calculation of results;
- Print data;
- Save/download data.

The failure 'incorrect calculation of results' which can be considered more serious than the others occurred in a significant number of use cases (7 out of a total of 20 use cases). To eliminate this failure the implemented algorithms have to be checked and corrected by software developers (this is an example of failure where the only option is to act on the solution space).

After using the widgets, users filled a usability enquiry, with 12 questions, to assess the quality perception when using the system. Answer spores rated according to a 5-point Likert scale. As an example, Figure 5 shows the results of the 12 questions for a specific use case. For each question, the Figure shows the average for all answers and the range of answers from the users that performed off-line testing on this specific use case.

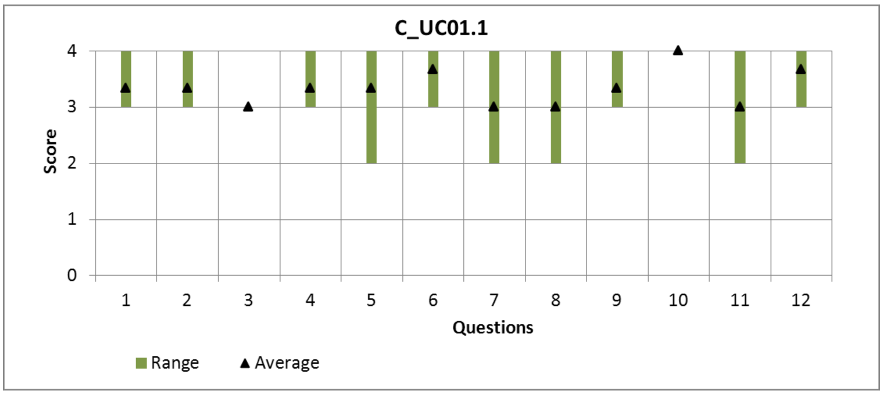

Figure 5 – Example of functional test results

5. CONCLUSIONS

This paper presented the off-line testing methodology adopted to provide a real-life and full scale testing of the iWIDGET system. The proposed method is based on the best practices on the software engineering filed, assuming distinct test scenarios and test cases to assess the system functionality and quality (non-functional properties). While ensuring a comprehensive assessment, the testing scenarios must also be aligned with the system requirements. On the other hand, the effective testing requires a close collaboration with the system analysts, software developers and end users (households and utility stakeholders).

The widgets, both from the consumer domain and the water utility domain did not always provide a correct behaviour with respect to the functional requirements for tested conditions and input parameters. Some minor failures were found during off-line testing but also major failures (incorrect calculation of results) that require changes in the algorithms of the widgets to be made by software developers. As a first understanding, this might seem as bad results but, in fact, that is what is expected from system testing. Indeed, detected problems on testing methods also mean that the testing procedures were done with a high-level of detail. As such, potential problems were identified in advance, making it possible to correct them and improve the overall system quality before releasing it to the end-users.

All issues identified by this testing were reported to the software developers and system analysts and are being corrected. Indeed, the communication between the testing and developers' team allow a continuous improvement of iWIDGET as a whole system. This way, improvements in the widgets could be made by software developers as soon as test results were available.

Also, we would like to remark that testing must be seen as a critical process and application improvement activity, where anomalies are detected in advance, making it possible to correct (or being aware of) functional and non-functional issues before deploying the system to the final users.

In general, testing activities must be seen as an important process to support the continuous system improvement, where anomalies and users' behaviour must be tracked to provide informed decisions about new or updated versions of any system.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 318272.

The authors of this paper would like to acknowledge all members of the iWIDGET consortium, especially those that were involved on the off-line test execution.

REFERENCES

Bangor, A., Kortum, P., and Miller, J. (2009). Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies*. 4(3), 114-123.

Brooke, J. (1996). SUS: a quick and dirty" usability scale. *In* P. W. Jordan, B. Thomas, B. A. Weerdmeester, & A. L. McClelland. Usability Evaluation in Industry. London: Taylor and Francis.

Vieira P., Barateiro J., Loureiro D., Kossieris P. and Thies H. (2013). Methodology for offline testing. Task 3.2 Report. Project iWIDGET.