(FP7 614100)

D2.3 Validation Framework

9 July 2014 – Version 1.0

Published by the IMPReSS Consortium

Dissemination Level: Public

Project co-funded by the European Commission within the 7th Framework Programme and the Conselho Nacional de Desenvolvimento Científico e Tecnológico
Objective ICT-2013.10.2 EU-Brazil research and development Cooperation

Document control page

Document file: D2.3 Validation Framework_V1.0.doc
Document version: 1.0
Document owner: Helene Udsen (IN-JET)

Work package: WP2 – Requirements Engineering and SDP Architecture
Task: T2.4 – Validation Framework
Deliverable type: R

Document status: ☒ approved by the document owner for internal review
☒ approved for submission to the EC

Document history:

Version	Author(s)	Date	Summary of changes made
0.1	H. Udsen (IN-JET)	2014-06-06	ToC
0.2	H. Udsen	2014-07-02	Several sections added or edited
0.3	L. Christiansen (IN-JET), H. Udsen	2014-07-04	Completed document ready for internal review
1.0	H. Udsen	2014-07-09	Reviewer comments addressed. Final version submitted to EC.

Internal review history:

Reviewed by	Date	Summary of comments
Enrico Ferrera (ISMB)	2014-07-08	Accepted with minor comments
Peeter Kool (CNET)	2014-07-08	Approved with comments

Index:

1 Executive Summary

This deliverable defines the framework for evaluating the prototypes developed in the IMPReSS project, including results from usability testing, field trials, etc.

The Validation Framework outlines the selected Quality model with metrics and describes planning of evaluation activities relevant to the iterative development process adopted for IMPReSS.

The Framework briefly discusses internal testing and verification processes and describes the planning of User validation activities and Field trial usability testing.

Evaluation results will eventually be reported in deliverable *'D8.5 Platform Analysis and Feedback Report'*.

2 Introduction

2.1 Purpose, Context and Scope of this Deliverable

This deliverable defines the framework for evaluation and validation of the prototypes developed in the project, including results from usability testing, field trials, etc.

The architecture and solutions for the application areas will be defined in deliverables *'D8.1 Specification of Proof-of-Concept Applications'*, *'D8.2 Application Architecture for Energy Management' and* the prototype definitions (*D8.3.1*, *D8.3.2* and *D8.4*).

Evaluation is part of the user-centred development process adopted for the IMPReSS project. A description of this process and an overview of the iterative approach can be found in deliverable '*D2.1.1 Initial Requirements Report'.*

The results will be reported in deliverable '*D8.5 Platform Analysis and Feedback Report'* in M30.

2.2 Content

Section 3 provides an overview of the Framework, including the research methodology, Quality Model, prototyping and validation planning.

Section 4 outlines testing and verification activities.

Section 5 describes planning of user validation and field trial usability testing.

Section 6 contains a brief summary.

3 Validation Framework

Validation is part of the implementation of the user-centred development process which has been adopted for IMPReSS. The main aim is to assure that the IMPReSS solutions adhere to the necessary quality standards for the professional services and meet the needs and requirements of the intended end users.

3.1 Overview of Research Methodology

The IMPReSS project has adopted an evolutionary requirement engineering specification and design methodology, which complies with the following broad template in each cycle of the process:

1. Requirements for development of sustainable applications
2. Developer requirements gathering and engineering
3. Architecture design specification and refinement
4. Enabling technologies research to implement the architecture
5. Prototype development of the platform, system integration and testing
6. Evaluation of the development platform in the real application environment
7. Lessons Learned and change analysis leading to requirements refinement.

This document outlines step 5 and specifically addresses step 6 in the list above.

A detailed account of the requirements engineering process was provided in deliverable *'D2.1.1 Initial Requirements Report'*.

3.2 Quality Model

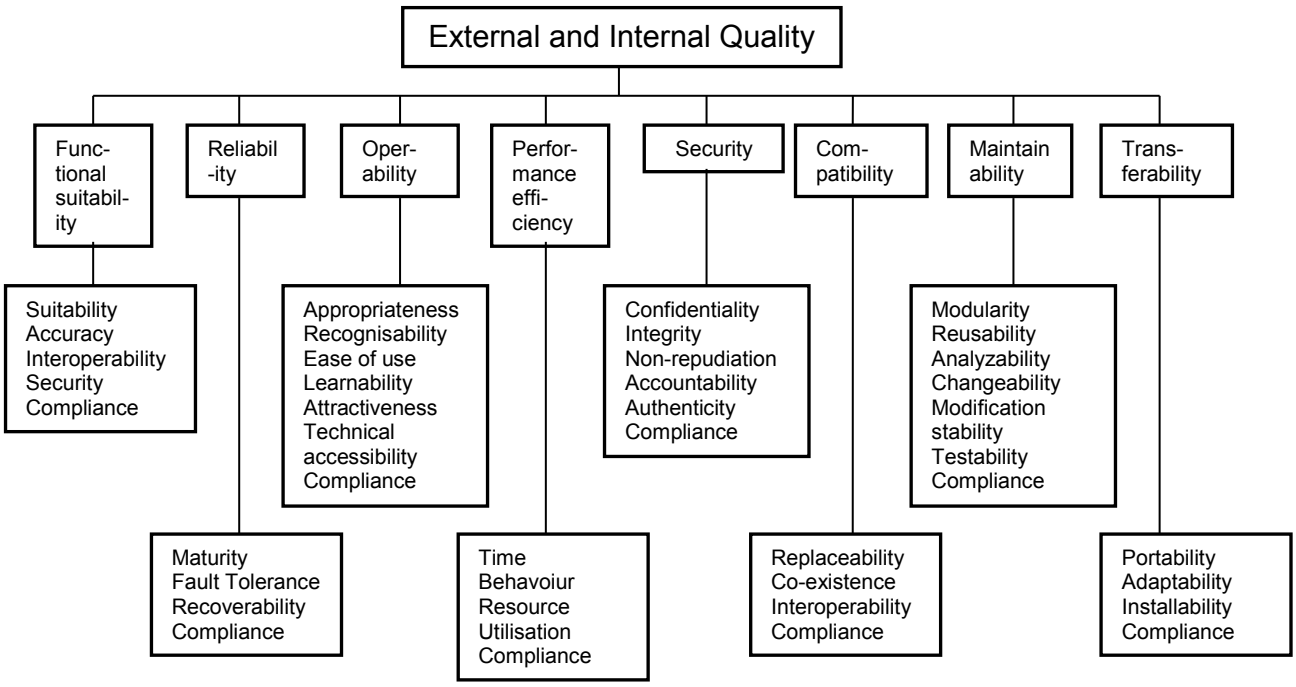

Figure 1: ISO/IEC 25010:2011 Quality Model for internal and external quality

A number of different software quality models exist, a more recent one being ISO/IEC25010 (ISO, 2011). This is a fairly comprehensive model, which is not necessarily relevant for IMPReSS in its entirety. ISO/IEC25010 will be used as inspiration and as a checklist to ensure that the applicable parameters are taken into account. The stage in the development process and the components implemented in the particular prototype under test will determine which parameters are considered in each case.

The Quality taxonomy of ISO/IEC25010 is shown in Figure 1.

3.3 Metrics

The quality of an implementation may be measured by quality characteristics or key indicators.

Quantitative metrics are defined as being directly measureable, e.g., in numbers. These comprise properties like performance (capacity, speed, etc.), system and deployment characteristics (uptime of equipment, network and software, response times, transmission error rates) or measurement uncertainty (noise, bias). Some of the quantitative metrics can stem from requirements of user acceptance. These will be defined for the individual user tests.

Qualitative metrics relate to the end user's judgment of the functionality implemented. The perceptions of the end user will be assessed through dedicated questionnaires suitable for rating the answers, for example:

1. Did the display of energy consumption live up to your expectations?

   Very good/good/fair/unacceptable

2. I feel more in control of electricity brownouts with the equipment

   Fully Agree/Agree/ok/completely disagree

For evaluation of the quality parameters outlined in the model above, ISO/IEC 25010 includes a software product quality measurement reference model, mathematical definitions of quality measures, and practical guidance for their application. Examples are given of internal and external measures for software quality, and measures for quality in use.

The evaluation plans and specification of complete metrics sets will be developed based on scenarios, use cases and test cases.

3.4 Prototyping

During the first 6 months the work will be dedicated to proof-of-concept and mock-up prototypes, which will be useful for facilitating the requirement engineering process.

Thereafter, two iterations will be carried out as follows:

Prototype 1 (M14)

- SDP Components: IDE for model-driven development, communication management on an ARM gateway, data analytics modules, context engine, mixed criticality applications
- Evaluation pilot: Teatro Amazonas energy management system

Prototype 2 (M26)

- SDP Components: Context modelling IDE, machine learning, network and event debugging, Model-driven IDE
- Evaluation pilot: University Campus energy management system

The prototype applications will be developed for evaluation of the architecture, the components and the development tools. The results and the Lessons Learned will be fed back to the developers for further enhancement of the solutions.

All information will be used to update and refine the requirements specification, and, if required, an updated architecture design specification.

3.5 Validation Planning

The validation framework outlines an agreed structure for software testing and user validation, including definition of appropriate metrics and guidelines for usability testing, refinement of the initially defined success criteria, and measurement.

The overall validation activities in IMPReSS consist of three different elements:

1. Verification; to test if the software is free of bugs
2. User validation; to evaluate if the services meet the expectations and requirements of its intended users
3. Usability testing; to assess the quality of use of the applications.

The general approach for the project is described here, whereas the selection of specific methods for each verification and validation activity will be described in subsequent sections.

The *software verification* (debugging and testing) is a quality control process that is used to evaluate whether or not a system component complies with regulations, specifications or conditions imposed at the start of a development phase. It is always performed at the laboratory level by the technical partner(s) responsible for the component.

Verification is the answer to the following question: Have we built the system right? (i.e., does it match the requirement specification?). Thus, verification is the process of evaluating a sub-system or system in order to check whether the products of a given development phase satisfy the conditions imposed at the start of that phase.

The *user validation* element is partly done at laboratory level, with internal technical partners analysing each software module and verifying its consistency alone and inside the overall architecture. Then the assessment of performance measurements is done with end users, whether these are professional users or not.

User validation is the answer to a different question: Have we built the right system? (i.e., is this what the end users need and want?). Thus, validation is the process of evaluating a sub-system or system at the end of the development process in order to establish whether it satisfies specified user needs.

The third element, *usability testing*, assesses the quality of use in field trials involving real-life end users. This must take place under controlled conditions to assure that valid and interpretable results are obtained.

A validation framework specifically for energy distribution will be provided by partner CHESF.

4 Planning of Testing and Verification Activities

Verification is a quality control process applied to evaluate whether or not an artefact, product, service or system complies with regulations, specifications or conditions imposed at the start of a development phase. Verification is often an internal process.

4.1 Software Test Environment

The software testing process should be integrated as much as possible with the production of the software itself. As the software development follows an iterative cycle, testing procedures should follow the development closely to give precise feedback to the developers as soon as possible, thus helping to identify and repair problems early in the design and implementation phase.

In order for the testing process to be efficient and for the results to be objective and repeatable, the testing should preferably be based on automated methods and tools, or at least on methods minimising the human factor in the testing process.

4.2 Structure of the internal Tests

The testing of the IMPReSS platform is an iterative process which involves multiple steps and targets at all layers of the platform. The testing procedure is performed in parallel with the development of the platform and also evolves in parallel with it. Its goal is to guarantee that the platform meets the requirements that have been specified by the end users, in a stable and seamless way, to test that the platform adheres to standards wherever this is required and that the system is able to cope with exceptions without crashing or unrecoverable problems.

The following kinds of tests are expected to be performed on the platform:

- **Unit tests**. Using automated unit tests and test cases it can be verified that the functionality of the system meets the user requirements. Unit tests are performed or repeated when adding new functionality to make sure that no bugs are introduced and that fixed bugs remain fixed
- **Integration tests**. High level tests will be performed to verify that the various system components interact and integrate in a seamless way based on a coherent semantic model
- **Internet of Things tests.** Most of the components developed in the project are intended for use in an IoT scenario, meaning that they cannot make assumptions about how they are being used and by which applications. Therefore they need also to be tested from an IoT perspective and to conform to established IoT standards and principles
- **System tests**. The final tests will be performed on the integration of all physical and/or logical subsystems in order to realise the envisaged prototypes of the development platform. The prototypes will be tested against the functionalities specified in the requirements.

More details pertaining to the verification process will be available in deliverable *'D7.1 Integration and Test Plan'*.

5 Planning of User Validation Activities

Validation is a quality assurance process of establishing evidence providing a high degree of assurance that a product, service or system accomplishes its intended requirements. This often involves acceptance of fitness for purpose with end users and other stakeholders.

5.1 Approach to User Validation

The user validation process consists of three steps:

1. Planning the validation at the end of each iterative cycle.
2. Carrying out validation activities according to the plan when the prototype demonstrator is available
3. Making decisions on the basis of the validation results (e.g., redesign, error correction, start of implementation, release) as part of the requirements re-engineering work.

The process followed is similar in all validation cycles: an initial preparatory part identifying and involving the end users, selecting the specific requirements to be validated and writing test scripts and/or questionnaires. After the evaluation the results are analysed and fed back to the developers.

5.2 Quality Parameters and Criteria

Which quality parameters are relevant depend on the application domain and the development stage of the system under evaluation. However, many quality parameters are generic in their nature and would apply to both IMPReSS applications domains – and indeed many other development products. Examples of some generic quality parameters are shown in Table 1.

Quality parameter	Measurement	Unit	Method(s)	Critical Value	Required Value	Optimal Value
Security and Privacy	Rating by users and experts Number of vulnerabilities	Global rating Number of vulnerabilities	Questionnaire	Global rating below average Too many vulnerabilities	Above average A minimum number of vulnerabilities	Highest values No vulnerabilities
Added Value	Rating by users/ stakeholders	Global positive impression	Questionnaire, Positioning, Interview with stakeholders	Below average	Above average	Highest value
Error correction	Number of errors on number of task executed	Number	Testing measurements and statistical analysis	Above 5 errors every 100 tasks	3 errors every 100 tasks	Below 1 error every 100 tasks
Integration	Inspection by experts and developers	Integration problems detected	Programming of simple applications including domain models and rules	> 2 problems	1-2 problems	0 problems

Table 1: Examples of generic quality parameters

For each user validation process and application domain, specific quality parameters should be defined. Many of these parameters will appear on the overall list defined by the ISO/IEC 25010 Quality Model taxonomy, see Section 3.2.

5.3 Prototype 1 – Teatro Amazonas Energy Pilot

The first domain application will be deployed in Manaus at the Teatro Amazonas Opera House to demonstrate the potential of a smart system for reducing energy usage and CO_2 footprint in an existing public building. A detailed application domain analysis for the Opera House can be found in deliverable *'D2.1.1 Initial requirement Report'*.

5.3.1 Stakeholders

Stakeholders for the IMPReSS solutions were identified in deliverable 'D2.2.1 SDP Initial Architecture Report'. Relevant for user validation are:

- **Developer end users:** The Application Developers who use the IMPReSS SDP to develop IMPReSS-enabled applications. Target applications are energy efficiency systems addressing the reduction of energy usage and CO_2 footprint, within the context of the Internet of Things (IoT)
- **Integrators**: The Solution Integrators, who install, configure, deploy application, and connect them to other external services and hardware components. Different people or organisations may play the role of integrators. Integrators must have special interfaces, such as web-based GUIs or smartphone/tablet apps, so that they are easily able to configure the system to operate under different circumstances in different environments
- **Application end users**: The final users, who are affected by the solution, such as audience of a theater or employees of a company (with different skills and positions) These people can interact with the solution by means of different interfaces (web-based, apps) for configuring certain parameters and receiving real-time information.

5.3.2 Metrics

User validation and acceptance testing may be done at various stages of the development process.

One measure is the percentage of requirements implemented in the solution being evaluated, compared to the total numbers of requirements specified.

Of specific relevance is the assessment by the end users of the User Requirements implemented in any particular Release of the solution under development.

The form shown in Table 2 can be used to record the results of user validation of typical User Requirements defined for the Teatro Amazonas Energy Pilot.

Key	Requirement Type	Priority	Summary	Fit Criterion	Results
(JIRA-REF)	User	Critical	Management system for Electrical energy used	The electrical energy can be displayed for at least one Power Meter. The Power Meter can switch the power to a device on and off	

Key	Requirement Type	Priority	Summary	Fit Criterion	Results
(JIRA-REF)	Functional	Major	Devices connected to a Power Meter	At least 5 out of 10 device types have been connected to a Power meter	
(JIRA-REF)	User	Major	Device properties	All devices connected to the system are registered in a Device Properties DBA with description of its properties	
(JIRA-REF)	Functional	Major	The Power meter measuring capabilities	The measuring and calculation of power for each device can be displayed at least every 60 seconds.	
(JIRA-REF)	User	Blocker	Management system display building and surroundings	Preselected drawings of building and areas around the buildings can be displayed and selected from a pick list	
(JIRA-REF)	User	Critical	Management system display of the devices	Information on 4 out of 6 devices can be displayed	
(JIRA-REF)	User	Major	Management system view of measured data	A single device can be selected and the real-time data displayed	

Table 2: Examples of User requirements for validation – Teatro Amazonas Pilot

5.4 Prototype 2 – University Energy Pilot

The second prototype pilot for electrical energy management will be deployed at the Federal University of Pernambuco in Recife. As for Prototype 1, this application domain is analysed in deliverable *'D2.1.1 Initial requirement Report'*.

5.4.1 Stakeholders

The first two groups of stakeholders are the same as for Prototype 1, i.e., **Developer end users** and **Integrators**.

The third group is also largely similar, comprising:

- **Application end users**: The final users who are affected by the solution, such as university professors, students and staff or employees of a company as for Prototype 1. These people can interact with the solution by means of different interfaces (web-based, apps) for configuring certain parameters and receiving real-time information.

5.4.2 Metrics

Obviously, the user validation processes for the University Pilot are in principle the same as for the Teatro Amazonas Pilot, see Section 5.3.2.

Table 3 shows examples of User Requirements defined for the University Energy Pilot.

Key	Requirement Type	Priority	Summary	Fit Criterion	Results
(JIRA-REF)	User	Critical	Visualisation of the energy consumption	The electrical energy consumption for a preselected area can be displayed on the public displays.	
(JIRA-REF)	User	Major	Display of the temperature and humidity	Measure the temperature and humidity in the rooms and outside the building with at least one device	
(JIRA-REF)	User	Major	Display of energy saving	The saving of energy can be shown for the displayed area for a preselected period	
(JIRA-REF)	User	Major	Display of the devices	The devices can be displayed on the building drawing	
(JIRA-REF)	User	Major	Display of historical data	Historical data can be displayed for at least one device	

Table 3: Examples of User requirements for validation – University Pilot

5.5 Field Trial Usability Testing

As part of user validation, quality in use may be assessed by usability testing in field trials in real environments. Field trials are the ultimate demonstrations, which should convince stakeholders and end users, whether these are professional or not, and open up for business opportunities.

Field trials can range from very simple stand-alone demonstrators to fully functional and integrated solutions, the latter being quite complex, time-consuming and expensive.

The scope of field trials in IMPReSS will be decided at a later stage.

5.5.1 Planning of Field Trials

Usability testing will assess the quality of use of the IMPReSS infrastructure, platform and applications.

The overall aim of field trials is to assess the effectiveness of the IMPReSS platform in the two domains, with the goal of proving the applicability of the solutions, demonstrating the benefit for all the targeted parties and assessing the impact at personal, organisational and societal level.

Field trials provide information for subsequent management decisions regarding performance and other features of the platform. To get interpretable and valid results from the field trials careful planning is necessary. Test conditions, instructions of users, data analysis procedures, and benchmarks for comparison have to be defined.

As part of a field trial, user manuals must be compiled, from a Field Trial test programme or a set of test cases, preferably based on the relevant scenario and use case descriptions. A support function must be established to aid users in case of technical difficulties.

5.6 Reporting of User Validation Activities

The results from user validation, usability testing and results from field trials will be reported in the deliverable *'D8.5 Platform Analysis and Feedback Report'*.

6 Summary

The Validation Framework outlined in this document illustrates the evaluation activities planned for the prototypes developed in the IMPReSS project as part of the user-centric iterative development process. The prototype pilots and their components are summarised, and the adopted quality model and metrics described. Both internal and external quality parameters will be considered, as will quality in use.

Evaluation may be done by stakeholders both internal and external to the project, while user validation is always focused on the intended end user, e.g., by field trial usability testing. Both evaluation and validation activities seek to verify that the requirements implemented in the different prototypes are satisfied. The results will be analysed and fed back into the development process as part of the requirements re-engineering work. The outcome from the evaluation and validation activities will be reported in deliverable '*D8.5 Platform Analysis and Feedback Report'.*

7 References

(ISO, 2011) ISO/IEC 25010:2011: Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality model