ebbits

Enabling the business-based
Internet of Things and Services

(FP7 257852)

D2.6 Validation framework

Published by the ebbits Consortium

Dissemination Level: Public

European Commission
Information Society and Media

SEVENTH FRAMEWORK PROGRAMME

Project co-funded by the European Commission within the 7th Framework Programme
Objective ICT-2009.1.3: Internet of Things and Enterprise environments

Document version: 1.1 Submission date: 2011-09-02

Document control page

Document file: D2.6_Validation_framework_V1.1.doc
Document version: 1.1
Document owner: Comau S.p.A.

Work package: WP2 – Requirements Engineering and Validation
Task: T2.4 – Validation of platform and services
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	S. Björnsson (TNM), M. Jacobsen (TNM)	2011-08-23	Initial version - ToC
0.2	H. Udsen (IN-JET)	2011-08-25	Section 3 added
0.3	M. Franceschinis (ISMB), S. Björnsson, H. Udsen	2011-08-29	Sections 4, 5 and 6 added
0.5	M. Franceschinis, P. Cultrona (COMAU), S. Björnsson	2011-08-29	Section 2, additions to Sections 4, 5 and 6
0.6	P. Cultrona	2011-08-30	Section 1
1.0	H. Udsen	2011-08-30	Minor adjustments, formatting
1.1	P. Cultrona, S. Björnsson, C. Pastrone (ISMB), M. Franceschinis, H. Udsen	2011-09-02	Subsection 2.2 added, reviewer comments addressed Final version submitted to the European Commission

Internal review history:

Reviewed by	Date	Summary of comments
A. Schneider (FIT)	2011-08-31	Approved with comments
P. Kostelnik (TUK)	2011-08-31	Approved with comments

Index:

1. Executive summary

The ebbits project aims to develop architecture, technologies and processes, which allow businesses to semantically integrate the Internet of Things into mainstream enterprise systems and support interoperable real-world, on-line end-to-end business applications. It will provide semantic resolution to the Internet of Things and hence present a new bridge between backend enterprise applications, people, services and the physical world, using information generated by tags, sensors, and other devices and performing actions in the real-world. ebbits opens possibilities to offer a wide range of new business services based on choreography of physical devices, software services, and people that we introduced as Internet of People, Things, and Services (IoPTS).

For this aim, WP2 will manage and undertake the work in carrying out the iterative engineering of requirements, with special focus on the engineering process of initial requirements and reengineering after the end of each iteration cycle.

The main scope of this work package is thus to maintain a continuous discovery and analysis of user centric requirements, needs and prospects, to be used in the design, development, implementation and validation of platform and services. Moreover, the aim is to plan and manage user validation activities and to collect, analyze and document the results.

1.1 Purpose, context and scope of this deliverable

The purpose of this deliverable is to describe the approach that will be adopted for the testing and validation phase of the subsystems and prototypes of the ebbits platform and to describe the procedures and the documentation structure that will be produced during these phases.

The scope of this deliverable is limited to the definition of the validation structure and not to report results from testing or validation procedures. The validation procedures are described taking into account the evolutionary design methodology used in this project with production of a prototype, validation, review of the requirements based also on the results of the validation and design of a next prototype.

Furthermore, also the need of verification and validation at level of the subsystems and then of the prototype has been considered as well.

The validation process will be based on a repetitive procedure that consists of four iterative cycles. Each loop corresponds to one year of the four-year RTD project, with requirements being evaluated and the results fed back into the process together with the observations of developments in technology, market and regulatory standards reported in the related watch reports.

The verification process is based on the V-model[1], a general and well accepted engineering approach to designing complex systems. It takes into account the requirements and the steps needed to transform them into a System design which is then divided into manageable modules.

Finally the different aspects of the internal test and the test to be performed at the manufacturing and farming sites have been analysed and reported including the differences in the validation and the verification of the different envisaged prototypes.

1.2 Deliverable organisation

This deliverable is organised in the following Sections:

- Section 2 presents briefly the background information related to the verification and validation techniques according to the two main application scenarios (manufacturing and farming).
- Section 3 describes the work performed in order to organise the validation activities, taking into account the activities performed in Task 2.4 of WP2 and more specifically in Subtask

[1] http://www.waterfall-model.com/v-model-waterfall-model/

2.4.1 "Validation Planning". In the following a planning of the verification and validation activities with their main goals is presented considering the specific requirements coming from the manufacturing and the farming environment.

- Section 4, entitled "Verification Activities", describes the internal verification procedures. These procedures are focused on the verification of the subsystems and prototypes to be developed tested against the requirements available at the beginning of each iteration.
- Section 5 consists in assuring that the results of the development project - that is, the application developed or, more specifically, the implemented result - actually meets the needs and requirements of users and, in the end, is accepted by them.
- Section 6 focuses on field trials usability testing. The overall aim of the field trials is to assess the effectiveness of the ebbits platform within a manufacturing and a farming scenario. The goal is to conclusively prove the validity of the applications, demonstrating the benefit for the two considered environments.

2. Introduction

The ebbits project main goal is to research and integrate *Internet of Things* (IoT) technologies into the business domain. Business applications developed within ebbits will be able to incorporate physical objects, services and people into mainstream enterprise systems supporting real world as well as online end-to-end interoperability.

The challenging environment containing a massive amount of heterogeneous devices and business applications from the IoT will be managed by using semantic technology that allows automatic processing of information and autonomous collaboration among devices.

ebbits will open new possibilities offering a wide range of novel business services based on the orchestration of physical devices, software services, and people. The latter concept is introduced in the ebbits project as the *Internet of People, Things, and Services* (IoPTS).

This deliverable describes the validation framework. It includes the definition of appropriate metrics and guidelines for usability testing, refinement of the initially defined success criteria, and measurement. It is important to have a well-defined structure for testing and evaluation, agreed by all partners, already at the beginning of the project.

2.1 Background

The background and context of the work performed and described in this deliverable follow the first phase of the project design and aims at identifying a uniform framework for the verification and validation of the various subsystems and prototypes which will be built during the whole duration of the project.

This deliverable is the result of the activities performed under task T2.4 "Validation of platform and services" of WP2, more specifically subtask 2.4.1 "Validation planning".

Verification and validation are essential in the life cycle of any embedded system platform. Its development is not complete without rigorous testing and verification that the implementation is consistent with the specifications. Verification and validation (V & V) have become important, especially in software, as the complexity of software in systems has increased, and planning for V & V is necessary from the beginning of the development life cycle. Over the past 20 to 30 years, software development has evolved from small tasks involving a few people to enormously large tasks involving many people. Because of this evolution, V & V has similarly also undergone a change. Previously, V & V was an informal process performed by the software engineer himself. However, as the complexity of systems increased, it became obvious that continuing this type of testing would result in unreliable products. It became necessary to look at V & V as a separate activity in the overall software development life cycle. The V & V of today is significantly different from the past as it is practiced over the entire software life cycle. It is also highly formalized and sometimes activities are performed by organizations independent of the software developer. In addition, V & V is very closely linked with certification because it is a major component in support of certification.

While the terms verification and validation are used interchangeably in papers and texts, there are distinct differences in their terminology. According to the IEEE Standard Glossary of Software Engineering Terminology, verification is defined as "The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase." Validation, on the other hand, is defined as "The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements." So verification simply demonstrates whether the output of a phase conforms to the input of a phase as opposed to showing that the output is actually correct. Verification will not detect errors resulting from incorrect input specification and these errors may propagate without detection through later stages in the development cycle. It is not enough to only depend on verification, so validation is necessary to check for problems with the specification and to demonstrate that the system is operational. Finally, certification is "A written guarantee that a system or component complies with its specified requirements and is acceptable for operational use."

Considering the production systems, the field trial involves production optimisation and energy awareness in manufacturing industries.

The aim of this demonstration is to develop and validate an ebbits online optimisation business application using an enhanced optimisation metric[2] by adding energy consumption and CO2 emission to the traditional efficiency metrics used in TPM (Total Productive Maintenance) and LEAN Manufacturing. The ebbits platform will demonstrate interoperability of traditional large scale manufacturing equipment with a range of heterogeneous and proprietary auxiliary equipment such as energy monitors, environmental sensor networks, etc.

The other field trial involves lifecycle management in the food chain. The ebbits platform will support continuous monitoring of food products regardless of its actual geographical location. The application will show how comprehensive traceability of products can be devised based on widespread availability of wireless networks and smart home infrastructures. Information will seamlessly be collected from farms, food processing plants, the logistic chain, and public databases and presented to the consumer in a usable and inclusive way.

2.2 Stakeholder analysis

At the start of a new project an important initial step is to identify the stakeholders and their needs and interests. Stakeholders can be individuals, groups and organisations who may directly or indirectly influence or be influenced by the project, both in terms of the development process and the project outcomes.

Though two application domains, Automotive Manufacturing and Food Traceability, have been selected as initial focus, the potential of the project outcomes is much wider, and therefore the stakeholders are an extremely divers group. Consequently the list of stakeholders in this analysis is by no means considered comprehensive; it only serves to illustrate some of the more obvious candidates with particular emphasis on the selected domains.

Four main stakeholder categories have been identified:

- ***The Concept Owner*** licenses the right to use the ebbits to industrial enterprises or service providers.

- ***Service Providers*** are organisations that establish the commercial ebbits platform and offer ebbits applications to enterprises or organisations in the forms of Software as a Service, Platform as a Service or Infrastructure as a Service.

- ***Business Partners*** are companies and organisations concretely taking part in the project and benefiting from its outcomes. Their business processes are ubiquitously interwoven and interacting with the aim of optimising and executing the business strategies.

- ***End Users*** are all the people and entities working and interacting with the ebbits applications, typically on a daily basis.

It is not trivial or easy to match each entity to the appropriate category of stakeholders. Nonetheless, on the basis of the categories shown above, several stakeholders have been identified for the two application domains. Table 1 summarises this classification, showing also how entities may appear in more than one category depending the point of view.

[2] OEEE: Overall Equipment and Energy Efficiency Index

Stakeholder Category	Entities in Automotive Manufacturing	Entities in Food Traceability
Concept Owners	Providers (e.g. ebbits partners)	Providers (e.g. ebbits partners)
Service Providers	Providers (e.g. ebbits partners)	Providers (e.g. ebbits partners)
Business Partners	Manufacturing plant managers* Suppliers Customers Providers (*When considered organisations)	Feedstuff suppliers* Farmers* Slaughterhouses* Processing and packaging enterprises* Wholesalers* Transporters* Retailers* (*When considered organisations)
End Users	Machine operators Line supervisors Area supervisors Maintenance crew Manufacturing plant managers** (** When seen as individuals working for the enterprise)	Feedstuff suppliers** Farmers** Slaughterhouses** Processing and packaging enterprises** Wholesalers** Transporters** Retailers** Consumers (** When seen as individuals working for the enterprise)

Table 1: Stakeholder categories in ebbits

3. Validation Framework

Validation is an essential part of a user-centred development process. The validation work is performed in Work Package 2 under *Task T2.4 Validation of platform and services*, which consists of three subtasks. The present validation framework is the outcome of *Subtask 2.4.1 Validation planning*, and the agreed principles and guidelines will be adhered to in *Subtask 2.4.2 Application field trials* and *Subtask 2.4.3 Deployment preparation*.

3.1 The iterative development process

The ebbits project has adopted an evolutionary requirement engineering, specification and design methodology, which includes a well-defined validation process. The work will consist of four iterative cycles as the one outlined in the picture below. Each cycle corresponds to one year of the four-year RTD project, with requirements being evaluated and the results fed back into the process together with the observations of developments in technology, market and regulatory standards reported in the related watch reports.

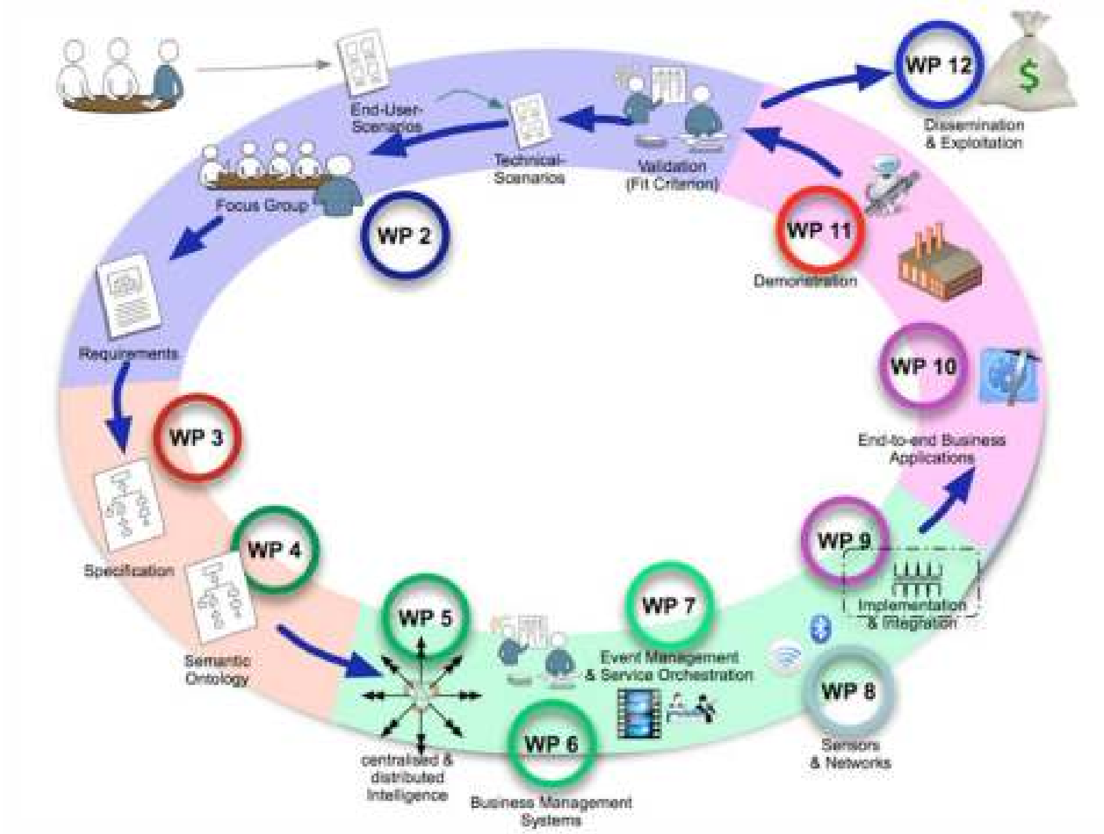

Figure 1: The iterative development cycle for ebbits

At the end of each annual cycle a prototype is planned with the specific purpose of illustrating the following aspects:

- End of year 1: First prototype of the ebbits platform serving as proof-of-concept
- End of year 2: Prototype II based on the production optimisation scenario
- End of year 3: Prototype III incorporating the food traceability scenario and additional elements from the production optimisation scenario
- End of year 4: Fully operational prototype, combining all elements from the two domains into the final platform prototype and demonstrator

3.2 Validation planning

The ebbits validation framework outlines a structured approach to software testing and user validation, including the definition of appropriate metrics and guidelines for usability testing and identification of success criteria for the field trials.

The user validation plan is inspired by the guidelines recommended by VNET5[3], including the three steps in the user validation process shown below. In this context user validation encompasses all elements of validation activities, and the users may be any of the stakeholder categories outlined in Section 2.2, depending on what stage of the process is involved.

1. Planning user validation
2. Carrying out validation activities according to this plan
3. Making decisions founded on the validation results (e.g. redesign, correction of errors, implementation, release)

During platform development, validation is carried out to detect possible deviations from the original objectives and to provide feedback to the development team for early corrective action. This will also facilitate tight, result-oriented monitoring of project status.

The user validation plan will be modified and adapted to changes in the development process whenever needed.

The overall validation activities in ebbits consist of three distinctly different elements:

- Verification – testing if the software is free of bugs
- User Validation – evaluating if the services meet the expectations and requirements of its intended users
- Usability testing – assessing the quality of use of the applications

A general overview of these elements is presented here, whereas the selection of specific methods for verification and validation activities will be described in Section 4 and Section 5, respectively.

Software verification (debugging and testing) is a quality control process used to evaluate if a system component complies with regulations, specifications or conditions imposed at the beginning of the current development phase. It is performed at laboratory level by the technical partner(s) responsible for the component.

Verification is the answer to the question: Have we built the system right? (i.e., does it fulfil the requirement specification?). Thus, verification is the process of evaluating a subsystem or system to test if the products of a given development phase satisfy the conditions imposed at the start of that phase.

User Validation is partly done at laboratory level, with internal technical partners analysing each software module and verifying its consistency, alone and as part of the overall architecture. Then the assessment of performance criteria is done with user partners and technical partners who have not contributed to the implementation, providing evaluations of the (stable) components and prototypes from different points of view.

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 subsystem 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 the field trials made with end users, where controlled conditions are needed to ensure valid and interpretable results. The applications may be a final product/service or a software development toolkit, the users being end users or developer users, respectively. The field trials are undertaken in Work Package 10.

[3] www.vnet5.org

The quality criteria that users apply to assess the value of an application or service include performance parameters such as functionality, effectiveness, efficiency and reliability, and also more subjective factors like look and feel, learning effort required and added value.

4. Verification Activities

Verification is the quality control activities that are used to evaluate every document, code, requirements and specifications. Many verification activities are already outlined in the "Description of Work" for the project, i.e. internal reviewing of deliverables and contents of deliverables. This Section only describes software verification. Further details are specified in deliverable *D9.1 Test and integration plan*, and the results will be reported annually in the deliverable series *D9.2.x Integration report x*.

4.1 The V model for software development

The V-model (shown in Figure 2) is a general and well accepted engineering approach to designing complex systems. The left part of the V illustrates how the requirements are transformed into a System design which is then divided into manageable modules. The right part illustrates how each design activity has a corresponding testing/verification activity.

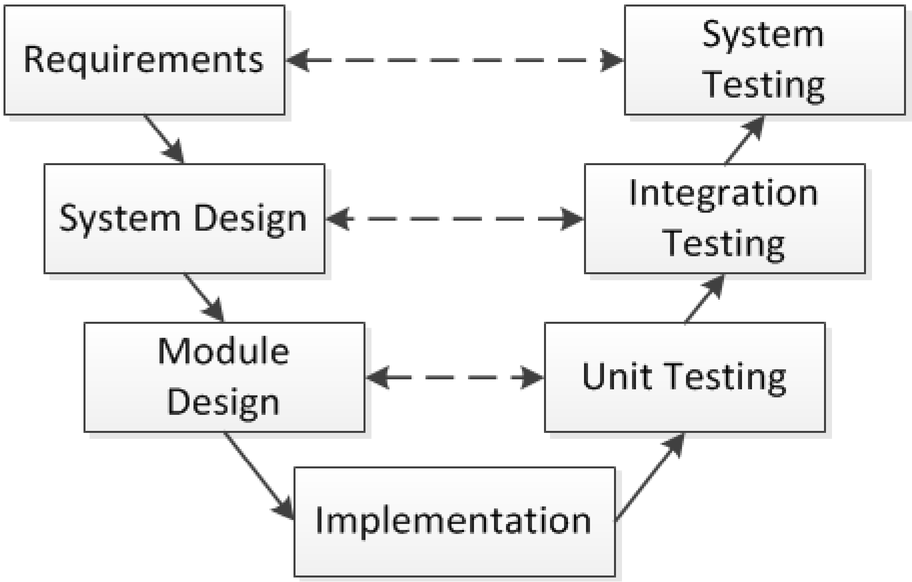

Figure 2: V-model for system development

This way of illustrating the system design process allows us to easy identify the different levels of testing. These tests are performed by the ebbits participants in order to verify the correctness of the implementation and make sure requirements are fulfilled.

4.2 Low level testing

This deliverable will not describe how the two lower levels of testing, unit testing and integration testing, are conducted as the ebbits project uses various tools and frameworks. There are many methods and tools available for applying these tests. Examples of unit testing tools are JUnit[4], NUnit[5] and XUnit[6]. Integration test are often done with functional testing tools. Examples of integration testing tools are FitNesse[7] and SWTBot[8] As a guideline it is the developers' responsibility to do unit testing and the work package leaders' responsibility to do integration testing. To make these tests

4 http://www.junit.org/
5 http://www.nunit.org/
6 http://reflex.gforge.inria.fr/xunit.html
7 http://fitnesse.org/
8 http://www.eclipse.org/swtbot/

predictable, repeatable, efficient and automated, if possible, the ebbits partners have committed to use continuous integration as well as unit testing.

4.3 System testing

System testing compares the current design with the ideal system described in the requirements. This is the single most important step in verifying the system. The requirements describe in detail the system, both its functionality and its qualities, and therefore the system testing verifies that the correct system was built. A complete list of requirements and requirement categories, both functional and non-functional, can be found in *D2.4 Initial requirements report*.

As described in this deliverable the requirements engineering is not a phase or a stage but a continuous effort to describe the ideal system. This methodology is an integral part of the ebbits project and a large part the work in Work Package 2 revolves around this.

The requirements re-engineering work is focused around the demonstration of the prototypes. The deliverables that guide this re-engineering work are:

- **Prototype application specifications**

  Before every prototype this deliverable describes the prototype specification and which requirements the prototype is expected to fulfil. This will then guide the final work towards the prototype. These deliverables have numbers D2.5.1, D2.5.2 and D2.5.3.

- **Change request and re-engineering reports**

  After the prototype demonstration this deliverable will concentrate information from lessons learned, verification/validation activities and watch reports changed requirements. Change requests can be about new requirements, changes in current requirements and removing current requirements. These deliverables have numbers D2.8.1, D2.8.2 and D2.8.3.

- **Updated requirements reports**

  These change requests from D2.8.X are then analysed and the ebbits requirements updated as needed. It is important to verify the total effect of the changes and identify impact on the work packages. This deliverable then publishes the updated version of current requirements for the ebbits platform. These deliverables have number D2.9.1, D2.9.2 and D2.9.3.

4.4 Metrics

The non-functional requirements describe the quality and characteristics of the system. These requirements need to be measureable and include description on how they are measured. In the requirements database the fit criteria is the quantifiable goal for the requirement. The fit criteria describes how the requirement is quantified it does not describe how to apply tests. A more detailed approach to describing measurements and other information for the non-functional requirements is needed. An example description is given in Table 2.

4.5 Standards

The ebbits platform covers many domains and has to comply with many standards. The standards range from food safety standards in the traceability domain to personnel safety standards in manufacturing to network standards. This deliverable will not cover all the standards as they are too many and as the project evolves the standards affecting the project will change. Among the ebbits participants we have experts in the fields of concern and we will rely on them to make sure the ebbits project adheres to standards where necessary. A continuous watch of market and regulatory standards is taking place in Subtask 2.3.4.

Requirement No.	Quality dimension	Measure	Unit of Measurement	Critical Value	Required Value	Optimal Value	Methods
22	Operational	Number of devices connected	Number of devices	300	500	>1000	Stress testing the network with many devices.
37	Performance	Wireless communication bandwidth	Mbit/s	24	54	>54	Bandwidth tested under various conditions
51	Operational	Deployment speed of a network	Number of devices per hour	15	20	>20	To be determined
61	Usability	Real-time activities of plant	User friendliness rating	Below average	Above average	Above average	Questionnaire

Table 2: Assessment criteria examples for non-function (qualitative) requirements

5. Validation Activities

5.1 Basic concepts of user validation

User validation is one of the basic components that must be included in a validation framework. Its main goal consists in assuring that the results of the development project - that is, the application developed or, more specifically, the implemented result - actually meets the needs and requirements of users and, in the end, is accepted by them. In principle, users may belong to any of the stakeholder categories listed in Section 2.2.

User validation evaluates if we have built the right system and fulfilled the end users' needs and wishes. In this light, validation is the process of evaluating a subsystem or a system at the conclusion of the complex development process. Such evaluation is carried out based on impressions and opinions of external entities (stakeholders) to establish whether the developed system satisfies specified user needs. User validation touches several topics, comprising for instance analysis of user needs and user satisfaction measurement and is considered a mature approach, based on scientific knowledge and proven and tested methods.

It is important to underline that the assessment of performance measurements is carried out in collaboration between user partners and technical partners, who have not themselves contributed to the implementation. This prevents risks of 'conflicts of interest' and ensures that the final result is a fair and unbiased evaluation from different points of view.

5.2 Approach to user validation process

The user validation process is developed along three chronologically consecutive and deeply interleaved steps:

1. Planning the validation, which occurs at the end of each iterative cycle
2. Carrying out suitable validation activities according to the above mentioned plan, both before and after the prototype demonstrator is available
3. Making decisions, on the basis of the validation results, as part of the requirements re-engineering work

The validation framework organised as above guides the collection of information about the project specific objectives, requirements and constraints which may to a certain extent limit the choice of appropriate methods for user validation (different methods measure different quality dimensions).

5.3 The User validation process

The user validation process is repeated multiple times, in correspondence with the several cycles. The realisation of the process is substantially similar in all validation cycles and foresees fixed steps (detailed in subsections 5.3.1 through 5.3.6 below) to be followed:

- A preliminary preparation part
- An (internal) verification activity and/or an (internal or external) validation activity in cooperation with (expert) end users (who may be user partners)
- Collection and subsequent analysis of the outcomes
- Definition of actions based on these results, fed back into the loop for the next step in the process

5.3.1 Planning and preparation of validation activities

The first, preliminary, part of this step consists in defining and briefly describing the subject of the validation, including platform and components to be validated. In addition, the specific requirements, organized using the Volere template, which need to be validated are identified.

Preparing a non-ambiguous, clear, precise User Validation Plan presents several advantages, as underlined here:

- It is possible to make comparisons among different ways of performing user validation activities, as they will occur later in the project. Such comparisons and evaluations of these different ways can be carried out easily, until the most effective and efficient approach is found.
- Formal schedules might help to identify critical factors, included time, cost, personnel skills and qualifications that will need a concentration of effort and the commitment of project partners and users.
- The existence of a plan is a powerful psychological means in the role of persuader. Indeed, it could contribute to generating a commitment amongst the development team by visually demonstrating that the plan has been well reflected and discussed. In other words, the plan makes clear that and how state-of-the-art user validation is achievable.

Another fundamental prudence that must be taken into account in this process is to ensure that the person/people who will be leading the evaluation process has/have sufficient skills and experience in the methods used, in order not to compromise the final success of the overall process. If necessary, it could be worth bringing in some outside expertise. Precautions and careful attention must be devoted to the following issues:

- In order to get a final result whose outcome is really reliable, sound method and conduct are essential. In particular, it is as easy as dangerous to get misleading results from small or unrepresentative samples, or from evaluator bias, or by over-generalising from single instances. Thus, these aspects must not be underestimated.
- It is important to liaise with the design & development and technical teams over timing and status of what is to be evaluated.
- The readiness of designs and prototypes for testing is critical, so timing must be evaluated being as realistic as possible.
- The technical constraints of the product must be fully and correctly understood without any ambiguity. Similarly, the time and budget constraints for making changes must be reliably estimated. Finally, it must be discovered which changes can be made more easily and which ones request much more difficulties.
- When users have to be involved in some process phase, related arrangements must be introduced with reasonable (enough) advance, and then they should be kept.
- It must always be remembered that in general it is not easy to find users who are suitable for the process and who have, at the same time, enough time availability to participate in the user validation process.
- There are PR aspects for the project when involving outside people in testing.

The validation templates, to be prepared before the evaluation activities take place, identify the actor(s), i.e. the test person/people. It is likewise useful to draft the corresponding user scenarios, the use cases or the test cases that such actors need to go through as part of the validation process. This allows customisation of the validation procedure, selecting from already existing methods that are considered appropriate.

5.3.2 Analysis of user needs, requirements and preferences

Collecting and determining user needs and requirements, and overall their subsequent analysis, is the preliminary action of the user validation process. The carefulness and precision paid at this phase is fundamental for the quality of the final result. Indeed, requirements and needs of different groups of users are analysed through detailed studies of the demonstrator application context, based on the involvement of expert users from the application domain. When considering the set of requirements, it is worth mentioning that, beside the subset of traditional ones, other only apparently less important and often forgotten requirements, specifically concerning security needs and realistic business models, have to be investigated.

The wideness of the set of requirements which are dealt with, along with the overall complexity associated to each single requirement in terms of heterogeneous characterizing attributes suggest that suitable requirements management tools should be adopted. Volere and GForge have been selected for managing the user requirements. Volere is a template for organising the requirements storing, while GForge is a freeware tracking tool commonly used for bug tracking, issue tracking and project management. GForge also provides a source code repository based on Subversion (SVN) as well as support for continuous integration. It is a powerful and extensible platform covering all aspects of the software development workflow. To illustrate requirement validation, an example of a suitable validation form is shown below in Table 3. In the following some additional details explaining the use of the validation form are provided.

The fields 'Type', 'Priority', 'Summary', 'Rationale' and 'Fit criterion' are the same as in the Volere template. The 'ID' fields allow different references to Requirements, Test cases, Use cases and Scenarios and allow the unequivocal identification of the specific requirement.

Fit criterion is usually filled in with open text. In case the fit criterion has to be measured by means of a laboratory test, the validation template must clearly indicate information such as test method, statistical processes to be applied, the number of trials necessary for a trustworthy result, boundary conditions and any other data necessary to conduct a reliable experiment.

If the fit criterion refers to a quality dimension which is not related to a numerical measurement (e.g., user satisfaction or user acceptance) a questionnaire must be prepared, aimed at measuring the specific aspect. Roughly speaking, the questionnaire works as a translator from qualitative to quantitative information. The questionnaire should investigate aspects like provided functionality, added value and other related topics. Such a questionnaire will provide information for depicting a quality space through performance, productivity and added-value dimensions from a user point-of-view.

ID	Req. No.:	Test case No.:		Use case No.:		Scenario Ref.:	
Type		**Priority**					
Summary							
Rationale							
Fit criterion							
	1st cycle	**2nd cycle**		**3rd cycle**		**4th cycle**	
Result							
Comment							

Table 3: Example of Validation Form

The needs of users are not statically frozen; while precise requirements, constraints and some preferences are to be considered fixed and thus maintained as they are along time independently of the conditions, a certain amount of flexibility is given such that one attribute may be traded for another attribute.

When referring to the questionnaire, it must be considered that the more experienced with respect to a service is the user the larger is the amount of information that is contained in his/her answering to the questions. In particular, the richness and the completeness of the answer is a precious source for the analysis of the user preferences in terms of trade-offs. Although a meaningful (quantitative) analysis demanding that a substantial amount of data is available is outside the scope of the project, simple interview and rating techniques may allow the collection of enough data able to give an indication of the trade-offs which users consider when selecting services or products for use and purchase.

If the questionnaire is well conducted and the users involved in answering are suitably chosen, the result of the questions campaign could help understand how valuable is to add specific quality features to a service; it could also indicate which main quality features users would like to see integrated into application packages.

5.3.3 Performing validation activities

Once the validation template and questionnaire are completed, the test person/people, assisted by the working group who prepared the evaluation activities, has/have to follow the instructions explaining details about how to perform the user validation. This operation can consist of a laboratory test or of a trial of the application before answering the questions.

At this stage of the process members of the design & development team and other stakeholders must be involved in observing the evaluation activities. While they are passive with respect to the actual validation process, this participation represents the most effective way for promoting feedback into design and gets people to act on results.

In the initial development phase, user interfaces guidelines for the design of information presentation and navigation structures, along with samples which illustrate them, are highly effective. When used in a more rigorous form, style guides must be complemented by a review process, which tests for the adherence to the guidelines. While this is usually and preferably done by an expert review, the important condition to be respected is that it must not be carried out by the developers themselves.

The reason is always the same: to separate designers from evaluators. And it is at the basis of the next constraint: evaluation process success is more likely to be achieved if design reviews during the early development phase are managed by those project's experts who are not involved in the development effort. They use checklists and test the system according to the defined use cases, playing the role of a fair and unbiased user. They conclude their engagement by reporting the results directly to the developers; in addition, they involve the developers directly in the design review.

Users are a precious source because the quality and the richness of their impressions and the feedback on services they provide are central to the convergence to high-quality services. It is therefore important to avoid presenting them with incomplete and defective software. Otherwise they could become frustrated and unable to provide much constructive feedback. Definitively, users should only be involved in tests when the development team is confident of the quality of the result of the development itself. Summarising in this light, some good practices are as follows: user tests should be well planned and correspond to minimum methodological constraints; experts should help with the planning of test sessions; finally, in order to demonstrate shortcomings of the application and problems of users and to convince developers, it can be useful to videotape relevant episodes of test sessions for later review and presentation to the developers.

In conclusion, it is worth stressing once more that the tests foreseen in the user validation process are carried out in order to identify as many problems as possible, and then to find a better solution immediately after: mistrust and be suspicious of lack of problems emerging by users feedback. Concerning the number of subjects to be used for testing, it can be initially small, but for conclusive tests a minimum in the range of 8 users is required.

5.3.4 Data analysis

After completing the test trials and questionnaire submission results must be at last analysed and evaluated. Obviously the laboratory measurements and the questionnaire responses will be matched with distinct data analysis methodologies based on different approaches. While the first will hopefully result in immediate numbers, manageable in a much easier way, the examination of questionnaires will pass through both quantitative analysis (i.e., statistical calculations drawing from multiple choice questions) and qualitative analysis (based on comments and observations emerging from open questions).

A fruitful data analysis requires that all the steps in data processing, elaboration and management are carried out based on an impartial attitude, aim for speed and sufficient rigour must be balanced, and focus on the things that matter for the product's success must be carefully considered.

Regarding execution times, the basic rule is that even if very accurate, 'too-late' results are no use to the project. By rather simplifying with a slogan, same-day analysis is the norm in much commercial evaluation work.

5.3.5 Data analysis results fed back into the loop

The validation results will contribute to the success of the project because all the user feedback will be shared with the system developers, in line with the iterative approach characterising the validation framework. Users may be end users or belong to other stakeholder categories (importantly user partners) depending on the situation.

The data emerging from the previous analysis will be distributed to the consortium partners, and, acting as a new and updated system input, they will serve in refining the original user requirements and in improving the system characteristics. If all the indications so far exposed have been implemented, the process should move to convergence as cycles pass. Note that also the last iteration, where the assessment result provides the basis for a list of further recommendations, shares the user requirements refinement process.

The findings at the end of the cycles must be discussed with the development team which will exploit them to possibly correct unsatisfying aspects. Also the implications for how the product (and the project) will achieve its quality goals need to be shared with the development team, and this should be done as early as possible. The chief value of user validation is how it improves design.

5.3.6 User validation report

The user validation report is foreseen to contain a description of the overall experience with the use of the platform, to report the results of the usability test, the scenarios workflow validation and the performance evaluation. Specific problems, inconsistencies or bugs at any level should be reported in order to be properly addressed in the next release and also new functionalities addressing specific user needs not yet included in the current requirement specifications should be clearly added and listed. Finally, in an effort of overall user satisfaction evaluation, it should be quantitatively calculated and reported, and specific suggestions should be fed back to the technical team.

The report should be framed in such a way that it will appear meaningful to all relevant stakeholder groups. In doing that, and in particular in the formal exposition of the User Validation Report, it must be remembered that document readers in general will have varying levels of technical, business and ergonomic understanding.

The possible content of a User validation report is shown below.

User validation report

1. Description of the development project

    The "service"

    Objectives, requirements and constraints of the development project

    Phase in which the application was validated and development status

    Objectives of the user validation

    Critical success factors

    Constraints for user validation

2. The quality strategy

    Validation questions agreed with the users of the validation results

    The validation scenario

    Focus of the assessment

    Quality dimensions and assessment criteria

3. Users, tasks, and context of use

Description of user groups

Description of the tasks users intend to perform with the application

Description of the context of use

4. Methods for user validation

The user validation plan

Description of the validation procedure

Analysis of the validation results

Recommendations and conclusions

5.3.7 Timeline for the validation process

Timelines should be defined for each iteration cycle, which is one year in the ebbits project. Such timelines should be correlated with the anticipated progress and deadlines for the development work. The length of the four different steps may of course differ from one cycle to the next and from one project to another. Figure 3 shows a possible timeline, illustrated for the first iteration.

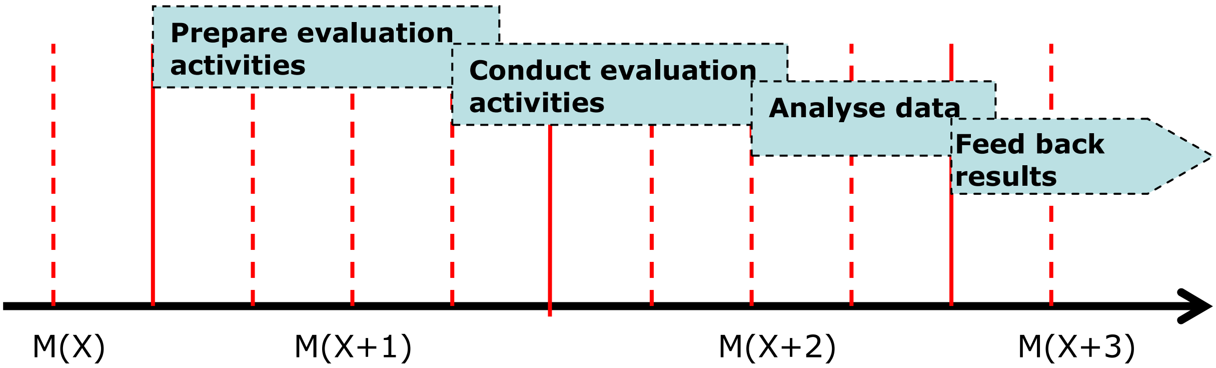

Figure 3: Timeline – First iteration

6. Usability Testing Activities

In the context of a project, usability tests concern and are conducted on the platforms, as well on the prototypes composing the platform itself and developed during the project. Such usability tests are executed by means of ad-hoc field trials and their purpose is the analysis of the platform in terms of usability by the final end user(s).

More precisely, the output of usability tests is the assessment of the quality of use of the platform project. Depending on the definition, the scope and the application areas of the project, usability tests are generally realized in different ways which are however oriented to evaluate the effectiveness of the developed solution. For instance, the choice about the number of users involved (small, medium, large) to detect user problems and deficiencies of the prototypes can depend on available resources (money, time, etc.) and in general it will result in a trade-off between deepness and accuracy of the result and its cost broadly speaking.

Usability tests are functional to the user validation process since when a platform undergoes such trials, the verification process has already been proven successful.

The validity of the platform/application must be demonstrated in a wide sense, by proving and gaining on the field the actual benefits for the different categories of users.

As already mentioned in the user validation framework description, feedback from the users of the platform are collected in order to give retrofit to the developers for redesigning, if and when necessary, user interfaces or functionality of the platform.

It is necessary to spend relevant amounts of time in preparing usability tests, because only a careful planning can guarantee valuable results in terms of clearly interpretable and useful indications. Some aspects that cannot be neglected in tests designing are the specification of test conditions, the preparation of instructions for users, the definition of procedures for subsequent data analysis, the provision of benchmarks for comparison purposes. Indeed, it is as initially burdensome as subsequently fruitful to give users a small introductory training regarding the system and its functionalities before proceeding with a test.

The system evaluation will be based on specific usability metrics, in order to have objective and quantitative data for analysis of the usability test. In addition, in order to make possible common assessment and aggregation of data involving different field trials, smart management procedures are followed. All the results of the evaluation sessions, along with the overall comments regarding the platform, are at last included in the validation report.

As previously said, the usability tests are tools with the purpose of achieving a quality assessment, which will decide for the success or failure of the platform. Nevertheless, quality is not a mere metric. First of all, it can reflect subjective opinions since it involves different people with different sensibilities. Secondly, it can concern multiple aspects. For this reason, quality (of use) is quantified through a number of quality dimensions which are combined together. A typical example of set of quality dimensions is for instance the following one:

- performance (effectiveness - the ability to actually carry out tasks successfully, and efficiency - the cost in terms of time and other factors for carrying out the task)
- subjective assessment (affect) of the quality of an application
- learning effort required using a system
- cognitive workload
- added value

In the following sections we apply the previous concepts referring to the ebbits platform and to the two main ebbits scenarios: manufacturing and traceability. The ebbits platform will support end-to-end business applications based on connectivity to and monitoring of a product in its entire lifecycle, i.e. from the early manufacturing stage to its end-of-life. Real-time lifecycle management implies that the manufacturer has online access to the product from the moment it starts the production process and until its effective end-of-life. If this would be possible for all products, it would dramatically change the business performance of a wide range of industries. Equipment and machinery already today features online asset management: From manufacturing equipment, large building heating and cooling installations to vending machines and pumps; all leading manufacturers have online access capabilities to their products. With the IoPTS, this capability will be extended to also include smaller products, including consumer products such as food and pharmaceuticals. The functionalities and sustainability of the ebbits platform will thus be demonstrated in two field trials.

6.1 Field trial for Automotive Manufacturing

The main objective of this section is to present user testing activities concerning the specific manufacturing environments taken into consideration by the ebbits project.

One field trial involves production optimisation and energy awareness in manufacturing industries.

The aim of this demonstration is to develop and validate an ebbits online optimisation business application using enhanced optimisation metrics[9] by adding energy consumption and CO_2 emissions to the traditional efficiency metrics used in TPM (Total Productive Maintenance) and LEAN Manufacturing. The ebbits platform will demonstrate interoperability of traditional large scale manufacturing equipment with a range of heterogeneous and proprietary auxiliary equipment such as energy monitors, environmental sensor networks, etc.

Here the general term manufacturing means the use of machines for the production of goods.

As already described, the production process is performed by a machine composed of several stations, each of them is composed of several devices that interacting among them contribute to the realization of the final product. Basically in a production line it is usual to find PLCs, robots, drives, fieldbuses, valve packs, sensors, HMIs and a SCADA system.

6.1.1 Hardware

The control hardware architecture includes, in general, specific devices such as PLCs, CPUs, Expansion Racks, Servo Drives, Robot Controllers, Valve packs, Human machine interfaces (HMI) and Supervisory systems (SCADA). All these devices are integrated together, by means of specific communication bus and protocols.

6.1.2 Software

The software architecture contains applications such as PLC, to implement control, diagnostic and safety functions, HMI/SCADA to interface the system with the operator and Factory Information System to connect the system with the upper software management systems (i.e. MES).

6.1.3 Communication

The communication infrastructure is composed of all the connection systems between the devices and the upper control and management levels. Inside the production plant this kind of connection is called fieldbus.

6.2 Success criteria for Automotive Manufacturing

6.2.1 Hardware

To match the Critical Success Factors it could be important to consider specific hardware parameters such as:

- CPU speed: in order to reach high level of performance and real-time constraints
- Memory: to store the application program and a high amount of data coming from the field.

[9] OEEE: Overall Equipment and Energy Efficiency Index

- Availability of expansion slots: to ensure future expansion requirements
- Availability of different connection means: to guarantee a high flexibility

6.2.2 Software

To match the Critical Success Factors it could be important to consider specific software aspects like:

- Development of PLC program, Servo drive and safety application following the customer's specifications
- HMI and SCADA applications must be intuitive and user-friendly
- The integration of the MES with the manufacturing environment must be easy and must ensure a full overview and control of the production plant

6.2.3 Communication

To match the Critical Success Factors it is important to consider specific connection aspects like:

- Easy interconnection reached by a standardized protocol
- User-friendly configuration, i.e. easy integration of new devices inside the manufacturing control architecture

6.3 Field trial for Food Traceability

The other field trial involves lifecycle management in the food chain. The ebbits platform will support continuous monitoring of food products regardless of its actual geographical location. The application will show how comprehensive traceability of products can be devised based on widespread availability of wireless networks and smart home infrastructures. Information will seamlessly be collected from farms, food processing plants, the logistic chain, and public databases and presented to the consumer in a usable and inclusive way.

6.4 Success criteria for Food Traceability

The usability tests and field trials need to be tailored to the expected users of the system. The users in this scenario come from various groups. There are the professional users, i.e. farmers, authorities, wholesalers and others. But what separates this scenario from manufacturing is that we have costumers that are not trained in using the system. They will have to be able to use the system with minimal instructions and guidance. The third type of users is development users, that is, users that will develop application that interact with the ebbits framework.

The success criteria for Traceability field trials are:

- Consumers:
  - Can without help familiarise with the interface
  - Can with limited help retrieve relevant information about the product
- Professional users:
  - Can get in-depth information about products or producers
  - Can identify the source of products
- Developer users:
  - Can identify information in ebbits
  - Can retrieve information from ebbits network