Deliverable D600.11

ProSEco Assessment Methodology

First Version

WP 600

Grant Agreement number:	NMP2-LA-2013-609143
Project acronym:	ProSEco
Project title:	Collaborative Environment for Eco-Design of Product-Services and Production Processes Integrating Highly Personalised Innovative Functions
Funding Scheme:	IP
Date of latest version of Annex I against which the assessment will be made:	30.09.2014 (Version I)
Project co-ordinating Partner:	Fundación Tecnalia Research & Innovation
Project co-ordinator contact details:	Mikel Sorli, Dr +34 946 400 450 mikel.sorli@tecnalia.com
Project website address:	www.proseco-project.eu
Start date of the project:	01.10.2013
Duration:	48 months
Responsible of the Document:	USAL
Document Ref.:	D600.11 ProSEco Assessment Methodology
Version:	V1.1
Issue Date:	13.02.2015

DISSEMINATION LEVEL

PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

CHANGE HISTORY

Version	**Notes**	**Date**
0.1	Creation of the document	10.06.2014
0.2	Final structure	14.07.2014
0.3	Updated content (input received from ONA)	30.07.2014
0.4	First draft submitted to partners for feedback	11.09.2014
0.5	Modified version submitted to partners for feedback	05.10.2014
1.0	Final version of the document	09.10.2014
1.1	Updated version of the document	13.02.2015

This publication was completed with the support of the European Commission under the 7th Framework Programme. The contents of this publication do not necessarily reflect the Commission's own position.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

ACRONYMS

BC	Business Case
CA	Consortium Agreement
CAM	Computer Aided Manufacture
CNC	Computer Numerical Control
DoW	Description of the Work
EDM	Electronic Document Management
GQM	Goal Question Metric
KPI	Key Performance Indicator
LCA	Life Cycle Assessment
i.e.	id est (in English = that is to say)
ICT	Information and Communication Technology
IPR	Intellectual Property Rights
PES	Product Extension Services
PSS	Product Service System
RTD	Research and Technological Development
S & T	Scientific and Technological
SME	Small and Medium-sized Enterprise
UI/UX	User Interface/User Experience
w.r.t.	With respect to
WP	Work Package

1 Executive Summary

This deliverable provides a proposed assessment methodology for the evaluation of PES development and the deployment environment. This assessment methodology comprises of four main parts:

1. What to assess
2. Assessment Criteria
3. Assessment Settings
4. Assessment Techniques and Methods

The section on "*what to assess*" describes the technical components that are to be evaluated within the PES development environment, the PES process for each business case and the PES deployment environment. The Goal Question Metric (GQM) approach has been used to identify the goals that will be used as basis for the assessment.

The *assessment criteria* highlight different aspects that will be evaluated to ensure better service quality, smooth functionality and the expected final outcome. The key attributes that will be evaluated are technical quality, utility, likeability and ease of use of the system, and business benefits.

The section on the *assessment settings* covers both the early prototype in laboratory conditions and a full prototype assessment in an industrial environment using a number of techniques and methods.

The *techniques and methods* that will be considered for the assessment of the PES are *verbal protocol*, *focus group*, *usability questionnaire*, *observation* and *interviews.* Furthermore, the *McCall Quality Model,* which provides a set of matrices, is used to evaluate the technical quality of the engineering tools, the PES development process and the PES deployment environment.

Finally, we present how the proposed assessment methodology will be used to define *test scenarios* and *test cases* for evaluating associated technologies in business settings, taking ONA and ALB, as an example. Since the main purpose of this deliverable version is to focus in the methodology for the assessment, only the BC involving ONA and ALB (BC4) was provided as a preliminary example. Other test scenarios and test cases for the remaining BCs will be included later in the final version of this deliverable (D600.12: ProSEco Assessment Methodology-Final Version).

Model templates for defining test scenarios and test cases for ONA and ALB are included in Annex 2 – Test Scenarios Templates and Annex 3 – Test Case Templates respectively.

2 Introduction

2.1 Document Purpose

The main purpose of this document is to present an approach for evaluating the PES development environment and the business solutions that will be developed as the PES deployment environments. This document clearly demonstrates what needs to be assessed, the assessment criteria, the assessment settings, the assessment techniques and methods. The proposed assessment criteria (goals) are presented in terms of technical assessment, utility, likeability of the system and business benefits. The assessment settings will consider two conditions: evaluation of the early prototypes under laboratory conditions and evaluation of more mature prototypes under realistic industrial settings. Although the ProSEco project is planning to evaluate its technical solutions under four business cases, this document has taken one of them as an example to illustrate the assessment techniques and methods, namely the ONA and ALB business case (BC4). This approach will be further developed for the other business cases during the final deliverable D600.12 on Assessment Methodology (Final Version).

2.2 Approach Applied

The starting point for defining the assessment methodology is the ProSEco architecture that has already been presented in the Public ProSEco Concept document (please refer to Figure 3 in D100.4). This ProSEco architecture illustrates the planned technology tools within the PES development environment, workflows and the key technical components within the PES deployment environment.

Due to the complex nature of the ProSEco architecture, which is intended to provide a range of engineering tools to create sophisticated business solutions, it is important to establish a sound approach for evaluation that describes the context, purpose, intended KPIs, etc. With this in mind, the evaluation methodology is based on the following approaches:

- The assessment methodology is described using four distinctive parts – what to assess, the assessment criteria, the assessment setting and assessment techniques and methods
- The Goal Question Metric (GQM) approach will be used to set the goals of assessment for which questions and metrics will be defined
- The McCall Quality Model will be used to conduct a technical assessment to evaluate the quality of the PES development and deployment environments
- The assessment will be carried out in two settings: evaluation of the early prototype under laboratory conditions and the evaluation of the full prototype under realistic industry settings
- Use cases will be used to design the test scenarios and test cases for each specific business case

2.3 Document structure

The document consists of:

- Section 1: the Executive Summary with a short and concise overview of the overall content of the whole document
- Section 2: the introduction which describes the purpose of this document, the position of this document with respect to the whole project, and provides a brief overview of the contents of the document
- Section 3: explains the proposed methodology framework for the assessment of the PES development and deployment environment. It also describes the GQM model for setting the goals of the assessment
- Section 4: highlights what will be assessed during the PES development and deployment phases
- Section 5: presents the assessment criteria in terms of technical assessment, utility of the system, likeability of the system, and business benefits
- Section 6: describes the assessment settings for early and full prototype assessment
- Section 7: elaborates the assessment techniques and methods such as verbal protocol, focus group, usability questionnaire, observation and interviews
- Section 8: provides test scenarios and test cases, particularly for ONA and ALB (BC4)

- Section 9: provides conclusions and summary
- Annexes
    - Annex I, References
    - Annex II, Test Scenarios templates for ONA and ALB
    - Annex III, Test Case templates for ONA and ALB

2.4 Document objective and background

The objective of this deliverable is to present the ProSEco assessment methodology elaborated by the project's academic partners with the support of the industrial partners. The overall objective of this document is to establish a methodology for evaluating the technology solutions produced by the ProSEco consortium and providing feedback to the technology developers of the ProSEco consortium. This evaluation feedback will ensure that the ProSEco software tools and environments are usable by both business solution developers as well as the final users of the ProSEco solutions, in terms of functionality and usability.

The input to this document is the ProSEco Architecture described in the deliverable D100.4 and the business cases and requirements described in the deliverable D100.2. Building on the knowledge gathered in these deliverables, the proposed evaluation framework in this deliverable presents the initial ideas for evaluating the technology components presented within the ProSEco Architecture. We hope to further refine our evaluation methodology when the nature of the technology, workflows and the business context become more mature during the next phase of the project.

3 Overview of the ProSEco Assessment Methodology and Approach

This document provides a proposed assessment methodology and a possible approach for the evaluation of the PES development and deployment environment, as exhibited in

.

This methodology will be used to evaluate the technical components and the development and deployment environments that are described in the Public ProSEco Concept document (D100.4) which address a generic ICT solution for supporting the extension of the Product Service System (PSS) by enabling the dynamic composition of core services and application specific services adaptable to the different meta products and associated production processes in order to compose innovative PES solutions.

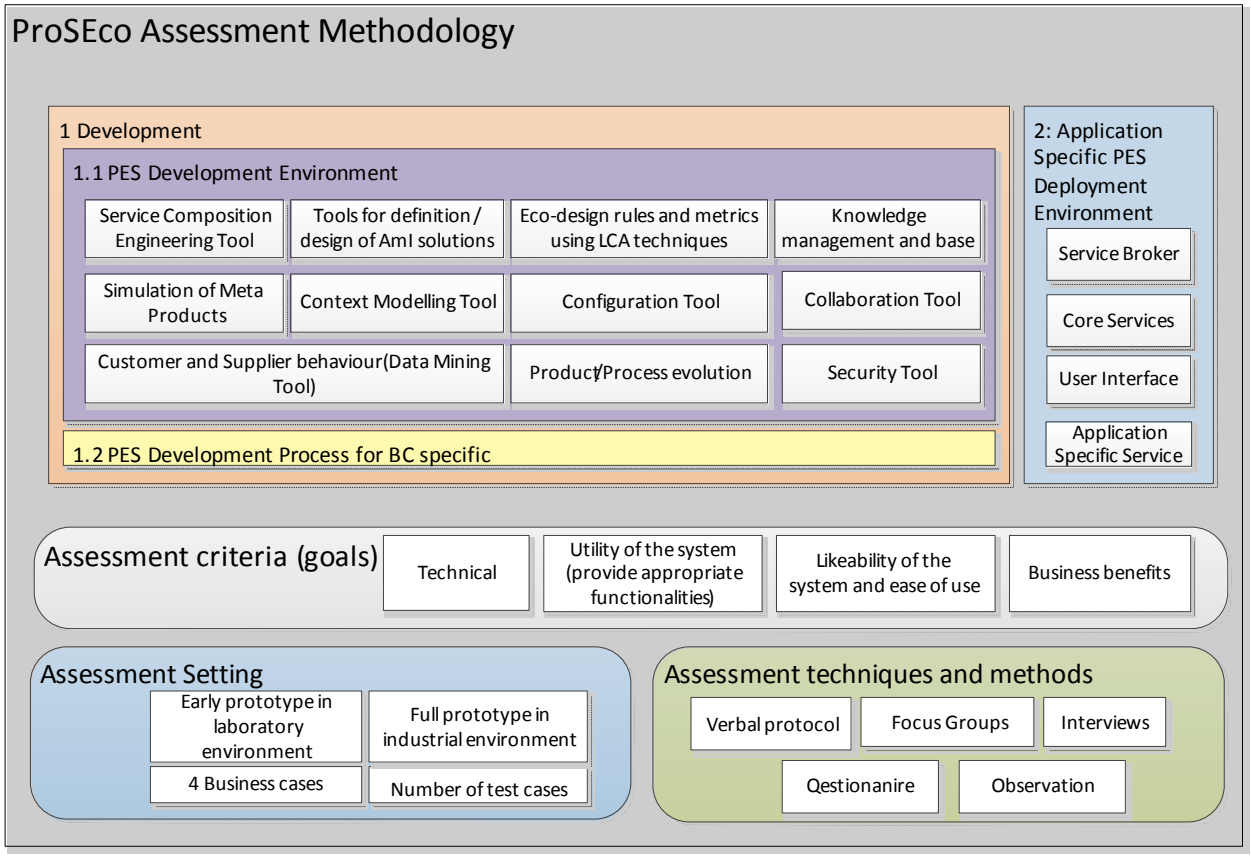

Figure 1- Overview of the ProSEco assessment methodology

As presented in the

, the ProSEco assessment methodology can be divided into four main distinctive levels:

- **What to assess:** this section is further divided into two main parts: 1) Assessment of the PES Development Environment, and 2) Assessment of the PES Deployment Environment. The development part consists of assessing the engineering tools as well as the BC specific PES development process. The Business Specific PES Deployment Environment comprises the assessment of a service broker, core services, a user interface and business specific services (details are provided in Section 4)
- **Assessment criteria:** this section sets out the main goals for the assessment. This highlights why assessment is required both at the development level and the deployment level. The key attributes of the assessment criteria are technical assessment, assessment of the utility of, and the likeability of, the system, and its ultimate business benefits (details in Section 5)
- **Assessment setting:** this section outlines the structure/setting for the assessment. This includes the evaluation of the early prototype in the laboratory and then testing the full prototype in the industrial environment. Testing all four business cases, as well as identifying a number of test cases, is also a part of assessment settings (details in Section 6)

- **Assessment techniques and methods:** this section describes the possible techniques and methods that can be adopted for the assessment of the ProSEco engineering tools in the PES development environment, the PES development process and the resulting PES (details in Section 7)

The assessment/ evaluation framework should be aligned with the organization's objectives (all four BC in our case) [7]. Different approaches have been developed to guide the definition and implementation of the assessment framework. Going from goals to metrics has proven to be successful for the evaluation of complex software systems. Lajos et al. in their paper [8] suggest following an assessment method that is defined in a top-down fashion. They further support their statement by saying that a bottom-up approach has several drawbacks because there are many characteristics in software, but selecting the important ones is not straightforward without well-defined top-level goals.

One of the well-known approaches, based on goal-oriented assessment is the Goal Question Metric (GQM) approach which was developed by Basili and Weiss and expanded by Rombach [1]. GQM has become a de facto standard for the definition of measurement frameworks [9]. One of the reasons for its success is that it is adaptable for use by many different organizations and in many environments, as confirmed by the large number of companies that have used it (e.g. Philips, Siemens, NASA, Hewlett Packard, Motorola) [1][10]. An overview of the approach is provided in the next section.

3.1 The Goal Question Metric (GQM) Approach

Considering all the above mentioned factors, and also to provide an approach that is flexible for all four business cases (Volkswagen, Electrolux, Desma and ONA/ALB), it has been agreed with the consortia to utilize the GQM (Goal Question Metric) as an approach for the assessment of the PES development environment and the resulting PES.

This approach has been defined as a top-down approach starting with specific goals to be measured, then questions are assigned to measure these goals and, finally, metrics are specified for the data collection. In general, the GQM method consists of four phases: the first one is the planning phase to select, define, characterize and plan a project for measurement application, resulting in a project plan. The second phase is the definition one, to identify a goal, all the questions, related metrics and hypotheses of the measurements. The third one is the data collection phase to define the data collection forms. The fourth phase is the interpretation one to process the collected data with respect to the defined metrics into measurement results and provide answers to the defined questions, after which the goal attainment can be evaluated [11], as demonstrated in Figure 2.

In the ProSEco assessment, particularly, four goals have been identified to be measured. These are:

- **Goal 1 – Technical Assessment:** this mainly involves the quality assessment of the PES development and deployment. Quality assessment of software is associated with a number of attributes that should be evaluated. These are correctness, reliability, efficiency, portability, integration and interoperability of the tools and the framework
- **Goal 2 – Utility of the system:** this measurement focuses on the functionality provided in the system. This measures the usefulness of the PES development environment and the resulting PES. This will also help to evaluate the efficiency to predict the following:
    - Do end-users find the PES development environment and resulting PES useful for their work?
    - Do the PES development environment and resulting PES meet the needs of the end user?
    - Do the PES development environment and resulting PES provide the right functionality as expected?
- **Goal 3 – Likeability of the system and ease of use:** this measurement covers the general likeability of the PES
- **Goal 4 – Business benefits:** this measurement covers the perceived business benefits and business performance

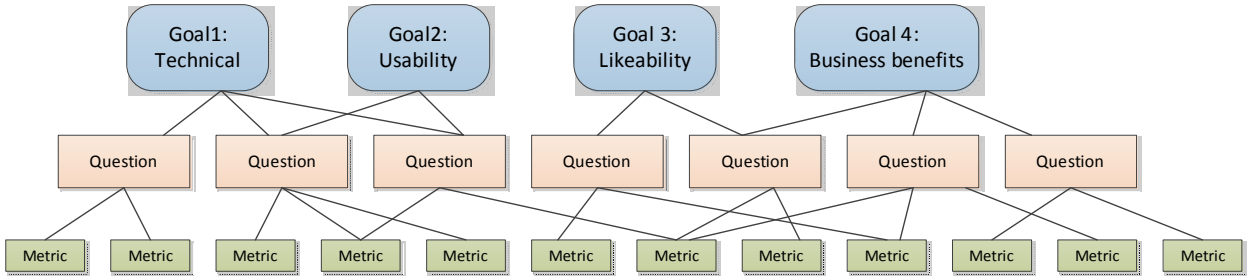

Figure 2- Use of GQM as an approach for the ProSEco assessment

Figure 2 explicitly demonstrates the four main goals of the assessment of the PES development environment and resulting PES. The goal is refined into several questions. Each question is then refined into metrics. The same metric can be used in order to answer different questions under the same goal. For instance, to check if the performance of the process is improving or not, the metric can be defined as: $\frac{Current\ average\ cycle\ time}{Baseline\ average\ cycle\ time} x100$ [1]. Several GQM models can also have questions and metrics in common, making sure that, when the measure is actually taken, the different viewpoints are taken into account correctly (i.e. the metric might have different values when taken from different viewpoints).

4 What to assess

As presented in Figure 3, the ProSEco project aims at evaluating/assessing the PES in two distinctive phases:

1) Assessment at the Development Phase: this phase is further divided into two levels:
   1.1) PES Development Environment: this includes the assessment of engineering tools based on their functionality
   1.2) PES Development Process for BC Specific: this includes the customization of the generic tools to comply with the specific requirements of the Business Cases
2) The Deployment Phase comprises: the developed PES for specific business cases

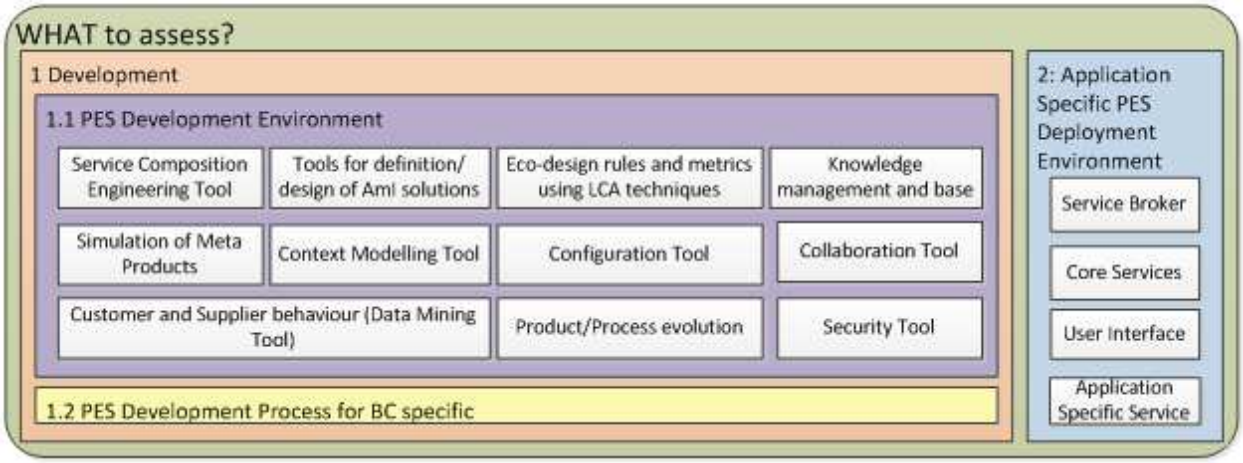

Figure 3: What to Assess

The assessment process is needed to ensure the effective extension of the PES within different sectors (automotive, home appliances, automation equipment and machine tooling) by means of the innovative combination of AmI technology, Lean and Eco-design principles, LCA techniques, operational research techniques and collaborative knowledge management technology. Furthermore, it is important to assess the quality of the engineering tools within the PES development environment as well as the resulting PES.

The literature survey reveals that, in software engineering, there are many types of software quality models, each of which comprises different quality characteristics and features or factors. By referring to international standards of quality measurements, Durgesh et al. in their paper [12] state that *"Quality comprises all characteristics and significant features of a product or an activity which relate to the satisfying of given requirements"* [13] and *"Quality is the totality of features and characteristics of a product or a service that bears on its ability to satisfy the given needs"* [14]. One of the more renowned predecessors of today's quality models is the quality model presented by Jim McCall et al. [15-17] (1977) which is also known as the General Electric's Model and is primarily aimed towards system developers and the system development process. This model has been widely used in software maintainability [18], complex software models [19] and object-oriented systems [20].

Given the nature of our project, which is a complex, service-oriented software system, the McCall quality model is used, as shown in the Figure 4. This model is used to demonstrate three major perspectives for defining, identifying and testing the quality of the PES development and deployment model. It is based on product revision (ability to undergo changes), product transition (adaptability to new environment) and product operations (its operational characteristics).

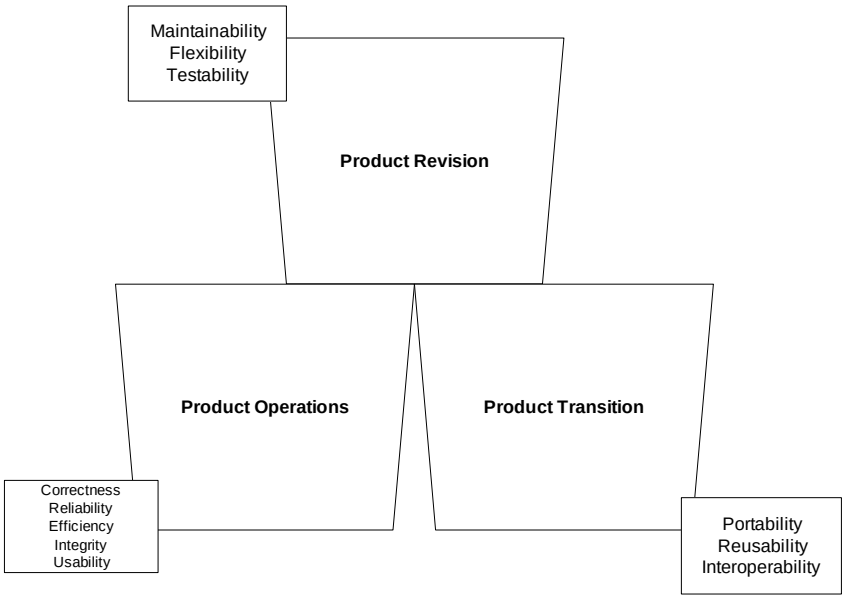

Figure 4: The McCall Quality Model

Product revision comprises maintainability (locating the problem in the system and fixing it in the working environment), flexibility (ease of modifying the code, model or software module) and testability (the ease of testing the tools, programs, and final product to make it error free according to user requirements).

Product transition refers to portability (transferring the program/ tool/product from one environment to another), reusability (the ease of reusing the tools and the software in different contexts) and interoperability (the effort required to couple the system to another system) using appropriate standards.

The third dimension, product operations, depends on correctness (whether or not the product is fulfilling the user/client requirement specifications), reliability (the system's ability to survive in all haphazard conditions), efficiency (tools' implementation efficiency, efficiency of the use of resources, i.e. processor time, storage etc.), integrity (protection of the program or the system from unauthorized access), usability (ease of implementing and using the software).

The scope of the assessment for the PES development and deployment mainly lies in "Product Operation" where the PES will be assessed for its correctness, reliability, efficiency, integrity, utility, and likeability. The PES could also be assessed for portability and interoperability at the end.

The PES will be assessed at the development level for the quality assurance of the engineering tools and at the process level for each specific BC, as shown in the Figure 3. The PES development environment will be assessed in terms of technical assessment, utility and likeability of the system as shown in the Figure 5.

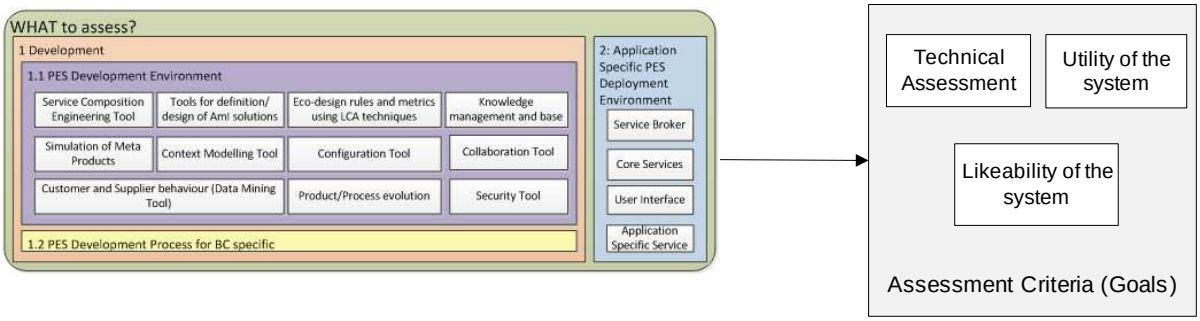

Figure 5: Assessment Criteria for PES Development

Similarly, the PES deployment environment will be assessed in terms of technical assessment, utility and likeability of the system in addition to its business benefits, as shown in the Figure 6.

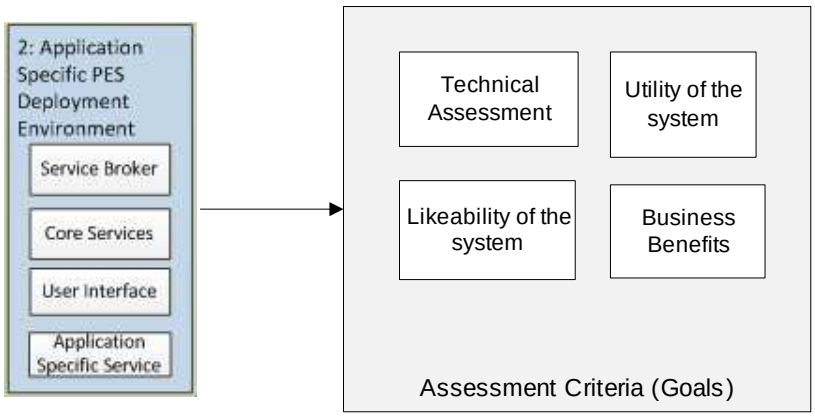

Figure 6: Assessment Criteria for PES Deployment

5 Assessment Criteria (Goals)

This section highlights the criteria against which the PES and its associated development tools and processes will be evaluated to ensure better service quality, smooth functionality and the expected final outcome. As discussed in the beginning of this report, the ProSEco project will assess the PES development environment and the resulting PES.

The following are the four aspects that will be tested during the evaluation:

- Technical Assessment
- Utility Assessment
- Likeability Assessment
- Business Benefits

5.1 Technical Assessment

The quality of software is made up of a composite of characteristics such as correctness, reliability, efficiency, portability, integration and interoperability. It further extends to the technical attributes of the analysis, design and code models. For this project, the quality or the technical functionality of the engineering tools, the PES development process for BC specific and the resulting PES will be measured in terms of adherence to a set of metrics or to set of attributes used to distinctively evaluate the quality of service-oriented systems. The following is a set of quality measures which will help to technically evaluate the above:

- **Correctness:** whether or not engineering tools and the PES are fulfilling the user/client requirement specifications
- **Reliability:** the reliability of the PES to survive in all haphazard conditions
- **Efficiency:** implementation efficiency, efficiency of the use of resources i.e. processor time, storage, etc.
- **Portability:** to check the ease of moving components and the whole system between different environments
- **Integration:** integration at component level as well as at system level
- **Interoperability:** whether or not the information is passing through the components and systems correctly and is using appropriate standards for information communications
- **Compatibility:** different components are operating accurately in the same platform

5.2 Utility of the system

This aspect of the evaluation will assess whether or not the engineering tools, the PES development process for BC specific and the resulting PES are providing the right functionality. It will assess the following:

- Does each engineering tool provide the correct functionality to carry out the intended task in the PES development process? Are there any missing functionalities?
- Is the workflow that invokes various tools flexible enough to provide seamless PES development tasks involving teams?
- Does the overall PES development environment facilitate team collaboration effectively to analyse data and discuss ideas to create new PES?
- Does the resulting PES have the right functionality expected during the design to offer eco products and services with benefits to both the producer and the consumers?

5.3 Likeability of the system and ease of use

This assessment will cover issues relating to general likeability and ease of use. It will assess whether or not all of the customers/clients are able to use the PES easily and effectively. Likeability of the PES can be measured in terms of user/customer satisfaction. This will explore aspects such as:

- Is the overall user interface of the PES development environment easy to understand and use?
- Are the interfaces of the individual engineering tools easy to use?
- Are the interfaces attractive and do they provide a good look and feel?

5.4 Business benefits

At the end, the PES will also be assessed as a full prototype in an industrial setting in terms of customer satisfaction, improved competitiveness and performance for each particular BC. This will also help provide a way of measuring the success of the PES in meeting the goals of the business. The resulting PES will also be evaluated for its expected benefits that are mentioned in the Description of Work document (DoW).

6 Assessment Setting

The assessment setting consists of two main parts: early prototype and full prototype assessments. However, a number of pilot studies, in the initial testing phase, will take place with a group(s) of researchers to check the system's functionality and to ensure that the users will not encounter any difficulties in using the system. Furthermore, these pilot studies will serve as a trial to ensure that all methods and techniques used are clear.

6.1 Early Prototype in a laboratory environment

As explained in the DoW, the early prototype of the PES development environment and the resulting PES will be assessed by the industrial partners in a laboratory environment using real data. During these tests, the industrial partners will be supported by the RTDs' partners to define test data and use the key functionality of the prototype. The test results will be used to refine the prototype, serving as a basis for the full prototype. While defining the test data for the early prototype evaluation, the industrial partners will be trained in using the PES. The RTD partners will elaborate key tests to cover all the functionality offered by the PES and forms (such as questionnaires, interviews and so on) to document the test results. Once the industrial partners have filled in the appropriate forms and questionnaires, providing the fulfilment of the envisaged functionality, task leaders ONA and DESMA will compile the feedback from the industrial partners in each task related to Testing T620 and Assessment T630.

6.2 Full Prototype in an industrial setting

The full prototype testing will be performed in real business situations (as opposed to the laboratory approach, used in the early prototype evaluation). The full prototype testing will involve more users than the early prototype testing, aiming at covering all possible functionality and collecting diverse opinions. The objective is to evaluate the PES in the real business environment where it will be used on a daily basis.

7 Assessment Techniques and Methods

This section introduces a number of techniques and methods that could be used for the assessment. The selection of the appropriate techniques will be discussed in the final version of this deliverable, based on the test cases.

- **Verbal Protocol:** this is a think aloud protocol where users vocalise thoughts, goals, perceptions, opinions, feelings and talk about their actions whilst performing tasks [2]. This method allows the evaluator to understand how the participant approaches a particular task and their reasoning when interacting with the interface. Any differences between the expectations on how to conduct a specific task and the actual sequence of the steps required in order to complete that task will be highlighted using this technique

- **Focus groups:** focus groups usually involve 6-8 participants and 1-2 moderators. The interaction between participants can lead to insights about certain issues which would not usually emerge from the use of individual interviews [3]. Participants could either be the same type of end user or different stakeholders from an organisation. The moderator has specific goals and objectives for a focus group session and usually guides the direction of the discussion

- **Usability Questionnaire:** a custom-made usability questionnaire could be used to assess general usability issues experienced when using collaborative technology, specific usability issues associated with interaction devices and displays, and to compare a number of technologies/interfaces/devices to determine users' personal preference. These questionnaires can include open and closed questions, rating scales and multiple choice questions [4]

- **Observation:** evaluators usually observe participants interacting with technologies, products and systems in order to gather information about a task, identify general and specific usability issues and monitor their behaviour [5]. This method could take a place with one evaluator taking notes and observing participants performing the required tasks. Multiple evaluators can ensure the consistency and accuracy of data collection. Video and voice recording can be used in this method if participants are talking loud as they perform their tasks which can be used later on to analyse the collected data

- **Interviews:** interviews can be organised to be structured, semi-structured or unstructured [6]. They allow participants to express their thoughts in their own words and can provide insight into their thought processes. Structured interviews consist of a specific, defined set of questions. Open-ended unstructured interviews allow more of a conversation between the evaluator and the participant. This would allow the interviewer to ask broader or more detailed questions about the topics that arise during the interview and would allow the participant to provide more additional information. Contextual interviews involve interviewing the user in context while they are conducting their daily work activities

Assessment could be performed automatically and remotely by means of scripts that capture specific events in the interaction between the services and the industrial partners' systems. These actions may be registered in a log file that can be queried and analysed remotely. If executed automatically, the used scripting tool will be indicated in the Testing Procedure. Also, the assessment could be performed manually and on-site by following the steps identified in a testing procedure.

8 Defining Test Scenarios & Test Cases (BC4)

The testing will be performed using test scenarios and test cases derived from the use cases defined in the D100.2 (Business Cases & Requirements Analyses). For each of the Business Cases, some use cases have been defined to describe the Business Case functionality. Test cases will be derived from these use cases' content. This way, there is a direct traceability between the requirements, use cases, and test cases. This can further be explained by Figure 7, where the top level scenario represents the needs or the requirements of a stakeholder. This number of requirements will be further refined to a set of features by the expertise that a stakeholder requires in the proposed system. Features in case of ProSEco can be defined as:

- Collection of AmI data
- Context aware modelling of the data
- Data mining application
- Defining eco-innovation strategies
- Security configuration
- PES configuration and composition
- PES deployment
- Collaboration333

These features, then, map down to a number of use cases that describe the functional requirements of the PES. From the use cases, test scenarios and test cases are derived for the testing and assessment of the PES. Before creating a test case, we need to identify all the scenarios for the given use case. A scenario is an instance of the use case. It describes one specific path through the flow of events.

Since the purpose of this deliverable is to propose a methodology for the assessment of the PES development environment and its resulting PES, only one BC was provided as an example and discussed in this deliverable—the BC4, involving industrial partners ONA and ALB. Other test scenarios and test cases for other BCs will be included later in the final version of this deliverable D600.12 (ProSEco Assessment Methodology-Final Version).

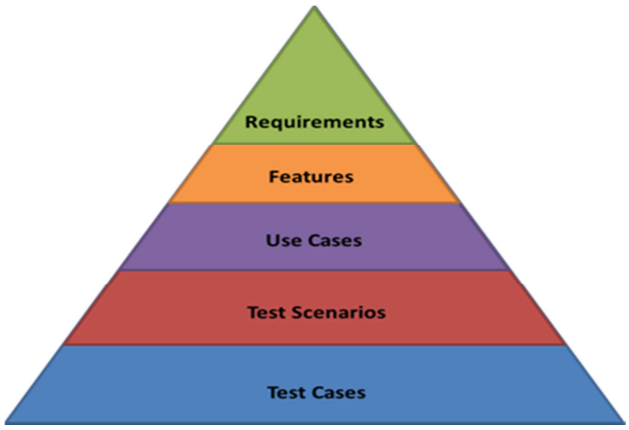

Figure 7: Traceability link from requirements to test cases (Ref: ProSEco Concept Deliverable)

8.1 Business Requirements and Features in BC4

According to the ONA and ALB (BC4) user requirements described in deliverable D100.2 (Business Cases & Requirements Analyses), a desired future scenario has been depicted for both companies. BC4 testing and assessment methodology has to be developed inside that future context in order to define the main BC4 test scenarios.

The Company Alberdi nowadays works taking into account the customer product design specs, that is, they do not undertake any design process. However, Alberdi would like to help their customers to define, in a collaborative way, a better product design according to their manufacturing process rules and taking into account the final user requirements while trying to reduce the cost and/or the environmental impact through the overall life cycle. The future scenario can be seen in Figure 8.

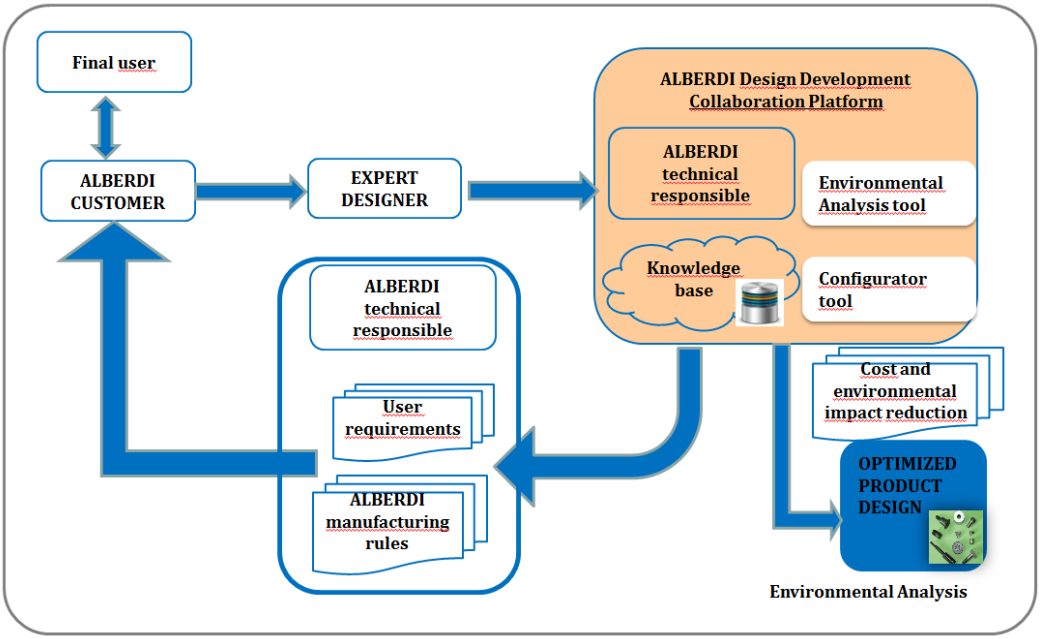

Figure 8: Alberdi future scenario (Ref: Business Case & Requirements Analysis Deliverable)

The Company ONA has their own product and would like to see the influence of eco-driven methodology and ProSEco services in a new machine development incorporating lean principles and eco design strategies in the design process. Some of the areas upon which ONA wants to focus include:

- ***Product customization and Environmental commitment.*** Customized EDM solutions are in the core business of ONA so the company is the world leading manufacturer of large & customized EDM machines including ecological filtering systems. Now ONA envision the possibility of collaborating with customers during the design process and in the testing of how Eco-design principles and LCA techniques could help reduce general costs and the environmental impact of the EDM machines and solutions.
- ***Monitoring, remote control as new services.***

The main interests for ONA and Alberdi in the BC4 context can be summarized as:

- improving the process for extracting user requirements in order to better include them into the custom product design process
- testing how Eco-design principles and LCA techniques could help to reduce the environmental impact of their products
- to explore the potential of the new control and supervising services (PES) implemented on their machines to influence future design

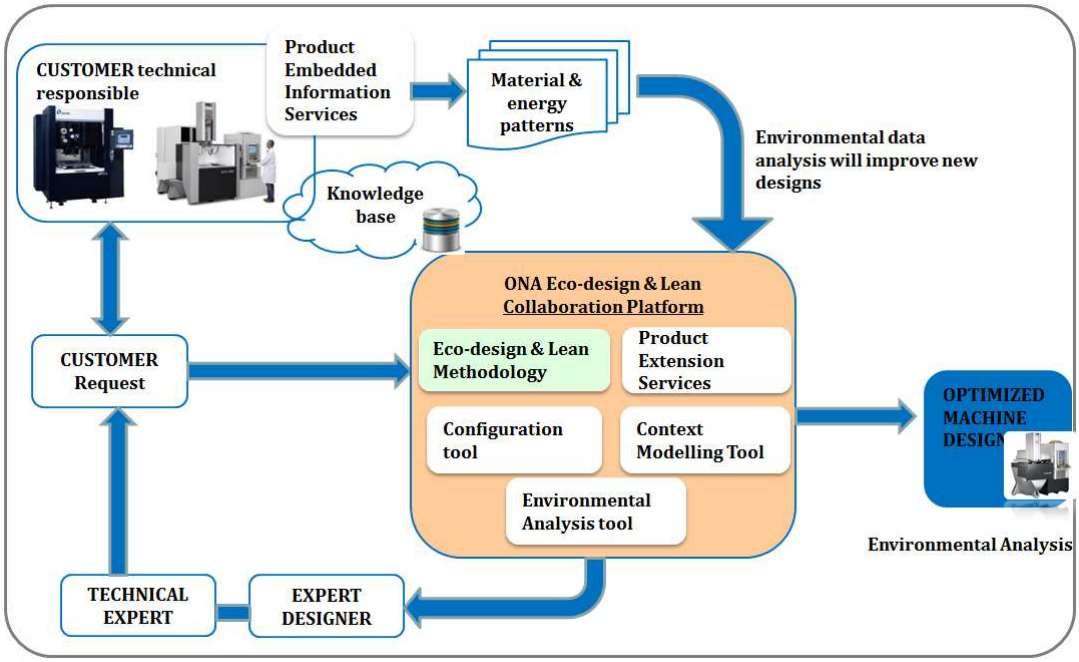

Figure 9: ONA future scenario (Ref: Business Cases & Requirements Analysis Deliverable)

8.2 Use Cases for ONA & ALB (BC4)

The following high-level use cases have been proposed in BC4 (deliverable D100.2 – Business Cases & Requirements Analysis):

1. **Co-creation with the customer:** custom products/solutions' design
2. **Build an ontology** based on the user-patterns, market issues taking into account environmental aspects in order to be used as a knowledge based repository in the co-creation process with the customer
3. **Select raw data from ONA machines** and possible operations that can be performed over the raw data to be used in future designs
4. **Build an ontology based on raw data (from sensors or Ami data)** and define the context model for a specific set of services

8.3 Test Scenarios for ONA & ALB (BC4)

From the testing and assessment point of view, two main testing scenarios are proposed, particularly for ONA and Alberdi:

1. ***Co-creation with customer***. Personalized products/services design.
   - Scenarios for solutions and simulation
   - Extracting customer requirements in the early stage of the Project (the "offer" moment)
   - Risks assessment
   - Knowledge base, learning from previous experience, repository solutions
   - ECO impact assessment
   - Collaborative Environment
     - Customer/Commercial Area
     - I+D. Design
     - Operations / quality / manufacturing

2. ***META-CNC***.
   - WEB enabled CNC services
   - Monitoring data. Filtering useful information. Security concerns when accessing data
   - Role of third party developers. New opportunities on "app" paradigm

- Personalized UI/UX. Customer role
- CNC-CAM: «From model to part» concept
- E-doc, tele-service/ diagnosis and support
- Machine Integration in complex manufacturing environment. Automation and CELL controller concepts.
- User behaviour analysis

Testing scenario 1 is aligned with the major key drivers defined in the ProSEco project and probably could also include the testing of scenario 2. However, keeping in mind the nature of the proposals for the other BCs in the current state of the project, it has been considered that the meta-product concept has a special relevance by itself. This separation into two main scenarios could help the testing purposes because it is reasonable to expect some synergies with the other BCs specially for test scenario 2, where the core idea behind it is to extract and to analyze potentially useful data from a complex system in order to generate new services that could be in the root of new business opportunities (Figure 10).

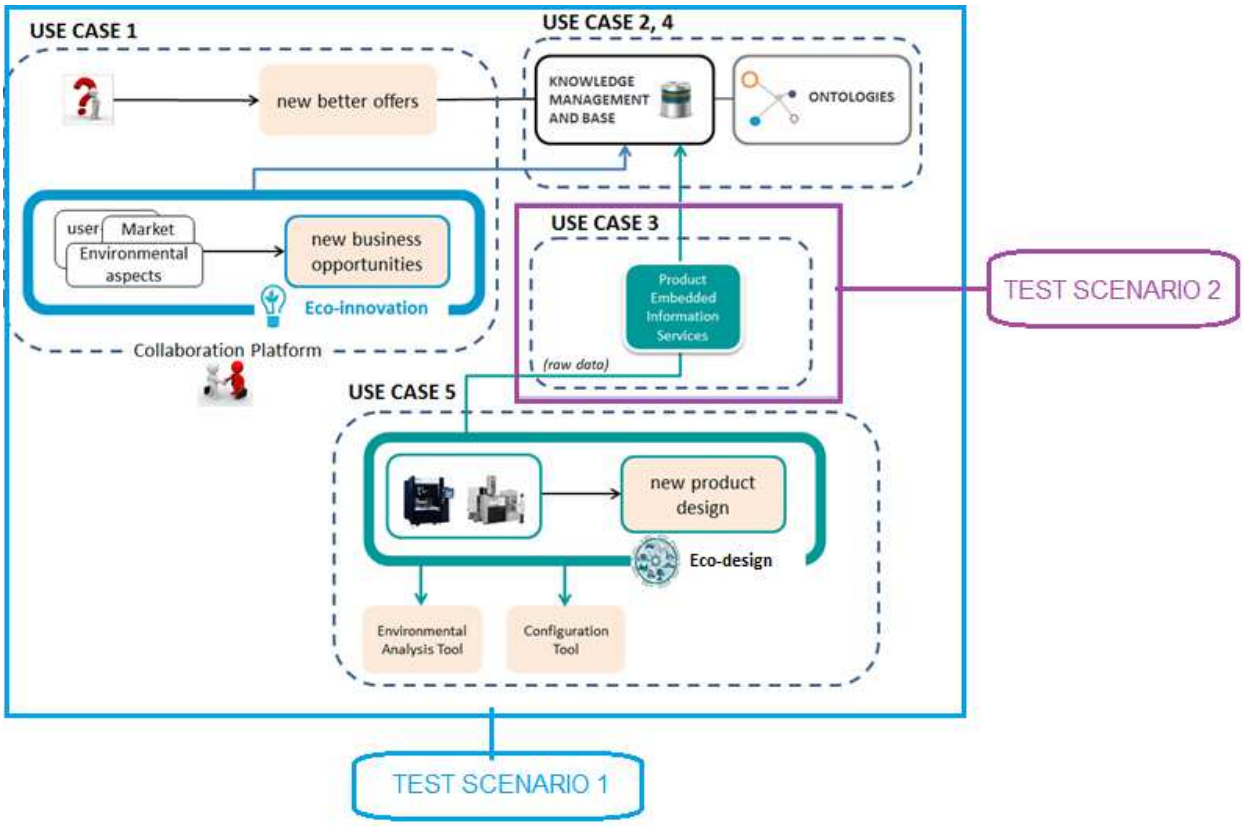

Figure 10: Use Cases integration with the ProSEco Environment (Ref: Business Cases & Requirements Analysis Deliverable)

These test scenarios will be defined to provide the following information:

1. What to assess within each scenario?
    a. Here, the activities will be monitored, clearly identifying:
        i. actors involved in the activity
        ii. systems involved in the activity
        iii. the (inter)actions made by the actors
    b. Which indicators will be measured?

Also, the test cases could have the following characteristics:

1. It is univocally identified
2. It is matched against the scenarios where it will be executed
3. It is traced against the related requirement

4. It indicates if it is performed manually or automatically. If performed automatically, the mechanisms for automation and scripts must be indicated
5. It indicates the collection of data used in order to provoke (trigger) a desired state or output
6. It indicates when to perform the test (maybe the activity assessed happens on a daily, weekly or monthly basis)
7. It indicates whether a specific precondition is needed for the Test Case to be executed (e.g. special actors involved, interdependencies of activities, state of a determined application). At this point it may be necessary to design a dummy process in order to provoke the desired state
8. It indicates the number of times the Test Case must be executed in order to obtain a valid result and to ensure a statistical significance of the results

8.4 Templates for Defining Test Scenarios and Test Cases

According to the BC analysis and knowledge of the ICT providers and RTD partners, a list of main requirements have been identified. These requirements are provided in detail in D100.2 (Business Case & Requirement Analysis document).

Based on the use cases that are identified in Figure 10, templates for the test scenarios are designed which are provided in Annex 2 – Test Scenarios Templates. According to Figure 7, as a final step for the evaluation of the system, templates for test cases are also designed, which are provided in Annex 3 – Test Case Templates. Please note that these templates (test scenarios and test cases) provide a basic idea of workflow at an early stage; further amendments could be applied at the time of specification and implementation.

8.4.1 Test Scenarios and Test Cases for Other BC's

As explained earlier, the ProSEco project is aiming at evaluating its technical solutions under the four business cases, however, in this deliverable, ONA & Alberdi business case is taken as an example to provide a template for designing the test scenarios and test cases based on the use cases. This approach will further be augmented in the final version of the assessment methodology, D600.12 [Month 24] where test scenarios and test cases for the rest of the business cases will be provided, as illustrated in Figure 11.

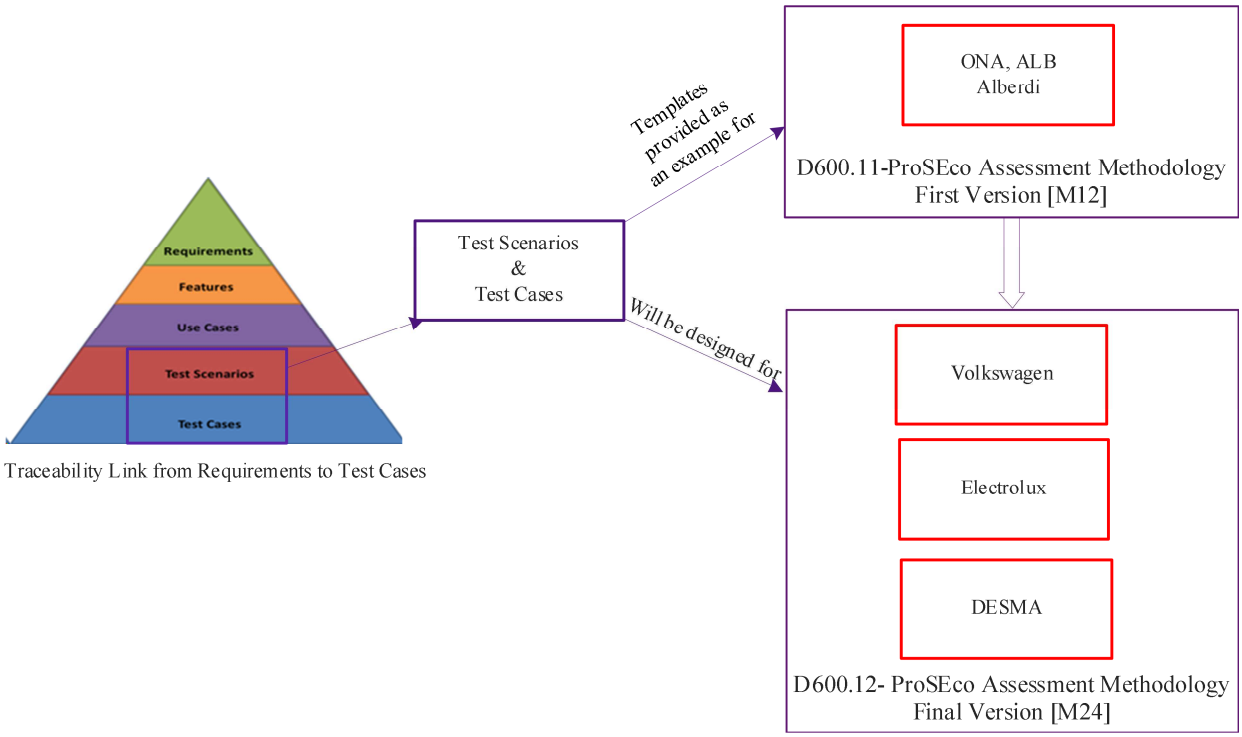

Figure 11: Current and future milestones for test scenarios and test cases for all BCs

9 Conclusions

This document provides an overview of the ProSEco proposed assessment methodology which comprises four main parts: what to assess, the assessment criteria, the assessment settings, and the assessment techniques and methods. In addition to the Goal Question Metric approach selected to identify the goals for the evaluation of the PES, each one of the four main parts of the methodology was discussed in detail. What needs to be assessed gave the explanation for the two main phases of the assessment methodology, while the assessment criteria defined the technical assessment, utility and likeability in addition to measuring the business benefits. The assessment setting was presented to cover both early and full prototypes using a number of possible techniques and methods to carry out the evaluation during the PES development and deployment. In the final section of the document, use cases for BC4 were used as a guideline to highlight possible test scenarios and test cases with some exemplary templates for test scenarios and test cases. The assessment methodology discussed here and its application to all BCs will be refined and finalized in the final version of this deliverable. This will include a detailed methodology and framework including test scenarios and test cases for the rest of the BCs.

10 Annex I – References

1. Basili, V.R., *Software Modeling and Measurement: the Goal Question Metric Paradigm*, in *Computer Science Technical Report Series CS-TR-2956*. 1992, University of Maryland, College Park, MD.
2. Bainbridge, L., *Verbal protocol analysis*, in *Evaluation of human work*, J.R. Wilson and E.N. Corlett, Editors. 1990, Taylor and Francis: London.
3. Morgan, D.L. and R.A. Kreuger, *When to use focus groups and why*, in *Successful Focus Groups*, D.L. Morgan, Editor. 1993, Sage: London, UK.
4. Patel, H., et al., *Human centred design of 3D interaction devices to control virtual environments.* International Journal of Human-Computer Studies, 2006. **64**(3): p. 207-220.
5. Spagnolli, A., D. Varotto, and G. Mantovani, *An ethnographic, action-based approach to human experience in virtual environments.* International Journal of Human-Computer Studies, 2003. **59**: p. 797-822.
6. Fontana, A. and J.H. Frey, *The Interview: From Structured Questions to Negitiated Text*, in *Hnadbook of Qualitative Research*, K.D. Norman and Y.S. L., Editors. 2000, Sage Publications Ltd.: London, UK. p. 645-672.
7. Van Solingen, R., and Berghout, E. Integrating Goal-Oriented Measurement in Industrial Software Engineering: Industrial Experiences with and Additions to the Goal/Question/Metric Method (GQM), Proceedings of the Seventh International Software Metrics Symposium (METRICS 2001), (London, England, April 4-6, 2001), IEEE Computer Society, Los Alamitos, CA, 2001, 246-258.
8. Schrettner, L.; Jeno Fülöp, L.; Beszédes, A; Kiss, A; Gyimóthy, T., "Software Quality Model and Framework with Applications in Industrial Context," *Software Maintenance and Reengineering (CSMR), 2012 16th European Conference on* , vol., no., pp.453,456, 27-30 March 2012
9. Van Solingen R., and Berghout, E. The Goal/Question/Metric Method – A Practical Guide for Quality Improvement of Software Development, McGraw-Hill Publishing Company, Maidenhead, England, 1999.
10. Solingen V., R., Basili, V., Caldiera, Gianluigi, and Rombach, D. H. Goal Question Metric (GQM) Approach, Encyclopedia of Software Engineering (Marciniak, J.J. ed.), online version @ Wiley Interscience, John Wiley & Sons, 2002.
11. Yu L.; Zhang Y.; Lu L.; Liu C., "Using three-tier strategy based on GQM to generate test case metrics," *4th IEEE International Conference on Software Engineering and Service Science (ICSESS), 2013* , vol., no., pp.204,207, 23-25 May 2013
12. Samadhiya, D.; Su-Hua Wang; Dengjie Chen, "Quality models: Role and value in software engineering, *2010 2nd International Conference on* " *Software Technology and Engineering (ICSTE)*, vol.1, no., pp.V1-320,V1-324, 3-5 Oct. 2010
13. German Industry Standard DIN 55350 Part 11
14 ANSI Standard (ANSI/ASQC A3/1978)
15. McCall, J. A., Richards, P. K., and Walters, G. F., Factors in Software Quality", Nat'l Tech.Information Service, no. Vol. 1, 2 and 3, 1977.
16. Marciniak, J. J., Encyclopedia of software engineering, 2vol, 2nd ed., Chichester : Wiley, 2002.
17. Kitchenham, B. and Pfleeger, S. L., "Software quality: the elusive target [special issues section]", IEEE Software, no. 1, pp. 12-21, 1996.
18. Ahmed, M.A; Al-Jamimi, H.A, "Machine learning approaches for predicting software maintainability: a fuzzy-based transparent model," *Software, IET* , vol.7, no.6, pp.317,326, December 2013
19. Pedrycz, W.; Peters, J.F.; Ramanna, S., "Design of a software quality decision system: a computational intelligence approach," *Electrical and Computer Engineering, 1998. IEEE Canadian Conference on* , vol.2, no., pp.513,516 vol.2, 24-28 May 1998

20 Chuan Ho Loh; Lee, S.P., "Predicting Quality of Object-Oriented Systems through a Quality Model Based on Design Metrics and Data Mining Techniques," *Information Management and Engineering, 2009. ICIME '09. International Conference on* , vol., no., pp.239,243, 3-5 April 2009

11 Annex 2 – Test Scenarios Templates

Table 1: Template for test scenarios

Test Scenario ID	BC4_TS1
Test Scenario Name	***Co-creation with customer***. Custom products/solutions design.
Actors	ALBERDI Customer (i.e. ONA) ALBERDI/ONA Offers Service Provider *The ONA customer* who specifies the new machine requirements. *The ONA technical responsible* who guarantees the manufacturability of the machine (taking into account the ONA manufacturing rules). *An expert designer* who introduces the technical and environmental aspects of the machine parts into the design process. *An eco-design expert* who analyses the whole life cycle of the machine and selects the best eco-design strategies to optimize the process
Description	<Short and clear description (as a bullet list for example) of the use case>
Trigger	<List the triggers that cause this use case to be executed>
Preconditions	<Indicate any possible preconditions that have to be met before this use case>
Postconditions	<Indicate any possible post-conditions that have to be met after this use case>
Normal Flow	<Describe the normal flow between the different types of users of the system and the various ways that they interact with the system>
Alternative Flows	<Describe alternative flows, if any>
Exceptions	<Specify exceptions that may occur and under which conditions within the depictured use case>
Frequency of Use	<Indicate how often the execution of such a use case is happening – daily, monthly, every second week etc.>
Business Rules	<Specify any specific business rules that are applied and needed for this use case>
Special Reqs	<Identify any additional requirements, such as non-functional requirements, that may need to be addressed during design or implementation>
Assumptions	<List any assumptions that were made in the analysis that led to writing the use case description>
Notes and Issues	<List any additional comments about this use case or any remaining open issues>

Test Scenario ID	BC4_TS2
Test Scenario Name	***META-CNC***.
Actors	ONA ONA Service Provider ONA Customer's Machines
Description	<Short and clear description (as a bullet list for example) of the use case> In this use case diagram the following use cases can be identified: • *Select Data from ONA Machines* – the Service Provider is presented

	with the list of existing possible data that he can have access to from ONA and may choose which one he wants to use in his service *(internal service for ONA or external one to be offered to the ONA customers)* • *Select basic operations to be done over the RAW data* – The Service Provider has the possibility of choosing from an array of simple operations which one he wants to have made upon the data that he chose in the Use Case "Select data" • *Negotiation* – The Service Provider may request ONA for data that is not listed (when using Use Case "Select data") and his request has to be negotiated with the ONA Customer (i.e. whether he accepts giving the data / adding the needed sensors in case they do not exist yet) • *Configure/update data list* – ONA configures or updates the list of data according to the sensors that are available in the ONA machines and which he wants to make available to the Service Providers
Trigger	For the use cases in this use case diagram the triggers are the following: • *Select Data* – the Service Provider wants to create a new service • *Select basic operations to be done over data* – the service provider has chosen the data that he needs but needs more complex data based on the selection • *Negotiation* – the Service Provider request ONA for data that is not listed *Configure/update data list* – ONA wants to provide a list of data available in the ONA machines.
Preconditions	The preconditions for this use case diagram are: • for the use cases *Select data, Select basic operations to be done over data* and *Negotiation* – the data list must be already configured (use case *Configure/update data list*) • Take into account the customer view on this extraction of data from his machines. Ontology to manage the raw data accordingly
Postconditions	Examples for post-conditions: • (When a design process ends) A design document signed by all the parts participants is obtained • (When a design process ends) A design document approved by all the participants is kept in the Document Management System. (When a design process ends) A signed Manufacturing Process Request is obtained.
Normal Flow	The normal flow of actions for the use cases in the current use case diagram is the following: • ONA starts with *Configure/update data list* for the creation of the data list • Given the data list, the Service Provider may start the process of creating a new service by choosing which data to use (*Select* data) and which data operations should be done over that data (*Select basic operations to be done over data*) • If the Service Provider needs some more information that is at the moment not available in the data list then he may enter in *Negotiation* with ONA At any point of the process, asynchronously, ONA may update the data list via the *Configure/update data list* use case
Alternative Flows	<Describe alternative flows, if any>

Exceptions	<Specify exceptions that may occur and under which conditions within the depictured use case>
Frequency of Use	The different use cases have the following frequency of use: • *Select Data* – this is used every time a Service Provider wants to start building a new service • *Select basic operations to be done over data* – this may be used every time a Service Provider wants to start building a new service • *Negotiation* – this may be used when a Service Provider wants to start building a new service and the available data does not correspond to the ones he needs *Configure/update data list* – this is used when ONA is setting up the list of data according to the sensors that are available in their machines and afterwards when they want to update this (due to machine change or to negotiation with the Service Provider).
Business Rules	ONA Customer preferences about the use of his machines' raw data by an external service provider.
Special Reqs	<Identify any additional requirements, such as non-functional requirements, that may need to be addressed during design or implementation>
Assumptions	The assumption for this use case is that the ONA machines exist with installed sensors for the collection of data and that this data is collected and sent to the ONA backend.
Notes and Issues	In the To-Be situation the machine data from ONA would be available for Service Providers to use and build services with it (upon authorisation from the machine customers/users)

12 Annex 3 – Test Case Templates

Table 2: Test case template

Test Case ID	BC4_TC1	**Scenario ID**	BC4_TS1	**Req. ID**	r_BC4_fun_01
Title	***Co-creation with customer***. Custom products/solutions design. Offer stage.				
Summary					
<Short and clear description of the feature to be tested, the general expected output, indicators to be measured and acceptance criteria> The outcome of this use case will be the definition of the scope and the critical requirements of a custom project, working in cooperation with the customer.					
Related Test Case	<Indicate whether the test needs to be run after a specific test case. Is the input of the test the output of another one?>				
Test Data	<Specific data that must be used in the testing procedure>				
Acceptance Criteria	<Describe the situation, event, state or output that must be obtained in order to consider a valid execution of the Test Procedure>				
Preconditions	<Environmental conditions needed to perform the test procedure. Necessity of executing a specific Test Procedure before this one in order to provoke a specific output will be indicated here>				
Automated	<If automated, indicate the name of the tool and/or script used. If automated, there is no need to complete the following steps>	**Repeat**	<Indicate the number of repetitions needed. Leave a blank space to annotate the number of repetitions that is being executed. The Tester has to print the Testing Procedure as many times as indicated in this field>		

Steps				
#	**Action**	**Expected Output**	**Output**	**Passes (Y/N)**
1	<Describe the action needed to execute the testing procedure>	<Indicate the expected output of the previous action>	<Space for the tester to handwrite the obtained result>	"Y" If expected output and Output matches. "N" If not.
2	---	---	---	---
Summary & Comments				
<Summary of the results and special events that may occur during the testing process: errors or exceptions found, unexpected behaviours, etc.>				

Test Case ID	BC4_TC2	**Scenario ID**	BC4_TS1	**Req. ID**	r_BC4_fun_01/2
Title	Knowledge based repository for co-creation process				
Summary					
<Short and clear description of the feature to be tested, the general expected output, indicators to be measured and acceptance criteria> **Build an ontology** based on the **user-patterns**, *market issues* taking also into account *environmental aspects* in order to be used as a knowledge based repository in the co-creation process with					

the customer.

Related Test Case	<Indicate whether the test needs to be run after a specific test case. Is the input of the test the output of another one?>
Test Data	<Specific data that must be used in the testing procedure>
Acceptance Criteria	<Describe the situation, event, state or output that must be obtained in order to consider a valid execution of the Test Procedure>
Preconditions	<Environmental conditions needed to perform the test procedure. Necessity of executing a specific Test Procedure before this one in order to provoke a specific output will be indicated here>

Automated	<If automated, indicate the name of the tool and/or script used. If automated, there is no need to complete the following steps>	**Repeat**	<Indicate the number of repetitions needed. Leave a blank space to annotate the number of repetitions that is being executed. The Tester has to print the Testing Procedure as many times as indicated in this field>

Steps

#	Action	Expected Output	Output	Passes (Y/N)
1	<Describe the action needed to execute the testing procedure>	<Indicate the expected output of the previous action>	<Space for the tester to handwrite the obtained result>	"Y" If expected output and Output matches. "N" If not.
2	---	---	---	---

Summary & Comments

<Summary of the results and special events that may occur during the testing process: errors or exceptions found, unexpected behaviours, etc.>

Test Case ID	BC4_TC3	**Scenario ID**	BC4_TS2	**Req. ID**	
Title	META CNC				

Summary

<Short and clear description of the feature to be tested, the general expected output, indicators to be measured and acceptance criteria>

Select raw data from ONA machines and possible operations that can be performed over the raw data to be used in future designs

Related Test Case	<Indicate whether the test needs to be run after a specific test case. Is the input of the test the output of another one?>
Test Data	<Specific data that must be used in the testing procedure>
Acceptance Criteria	<Describe the situation, event, state or output that must be obtained in order to consider a valid execution of the Test Procedure>
Preconditions	<Environmental conditions needed to perform the test procedure. Necessity of executing a specific Test Procedure before this one in order to provoke a specific output will be indicated here>

Automated	<If automated, indicate the name of the tool and/or script used. If automated, there is no need to complete the following steps>	**Repeat**	<Indicate the number of repetitions needed. Leave a blank space to annotate the number of repetitions that is being executed. The Tester has to print the Testing Procedure as many times as indicated in this field>

Steps				
#	**Action**	**Expected Output**	**Output**	**Passes (Y/N)**
1	<Describe the action needed to execute the testing procedure>	<Indicate the expected output of the previous action>	<Space for the tester to handwrite the obtained result>	"Y" If expected output and Output matches. "N" If not.
2	---	---	---	---

Summary & Comments
<Summary of the results and special events that may occur during the testing process: errors or exceptions found, unexpected behaviours, etc.>

Test Case ID	BC4_TC4	**Scenario ID**	BC4_TS1	**Req. ID**	r_BC4_fun_03/4
Title	New component-based design (eco-design)				

Summary
<Short and clear description of the feature to be tested, the general expected output, indicators to be measured and acceptance criteria>

Related Test Case	<Indicate whether the test needs to be run after a specific test case. Is the input of the test the output of another one?>
Test Data	<Specific data that must be used in the testing procedure>
Acceptance Criteria	<Describe the situation, event, state or output that must be obtained in order to consider a valid execution of the Test Procedure>
Preconditions	<Environmental conditions needed to perform the test procedure. Necessity of executing a specific Test Procedure before this one in order to provoke a specific output will be indicated here>

Automated	<If automated, indicate the name of the tool and/or script used. If automated, there is no need to complete the following steps>	**Repeat**	<Indicate the number of repetitions needed. Leave a blank space to annotate the number of repetitions that is being executed. The Tester has to print the Testing Procedure as many times as indicated in this field>

Steps				
#	**Action**	**Expected Output**	**Output**	**Passes (Y/N)**
1	<Describe the action needed to execute the testing procedure>	<Indicate the expected output of the previous action>	<Space for the tester to handwrite the obtained result>	"Y" If expected output and Output matches. "N" If not.
2	---	---	---	---

Summary & Comments
<Summary of the results and special events that may occur during the testing process: errors or exceptions found, unexpected behaviours, etc.>