Qualoss

Sponsored through Framework Programme Sixth (Call 5) by

European Commission

Information Society Technologies

Document Information

Version: **2.6**
Date : 03/20/09
Pages : 26

Owning Partner:
AdaCore

Author(s):
José Ruiz (AdaCore)
Rishab Ghosh (Merit)
Ruediger Glott (Merit)
Kirsten Haaland (Merit)
Jean-Christophe Deprez (CETIC)
Jacques Flamand (CETIC)

Reviewer(s):
Martín Soto (Fraunhofer IESE)
Raimundas Matulevicius (FUNDP)

To:
European Commission

Purpose of distribution:
Second Submission to address E.C. Reviewers' comment (from Period 2)

The QUALOSS Consortium consists of: CETIC (BE), Facultés Notre Dame de la Paix à Namur (BE), Universidad Rey Juan Carlos (ES), Fraunhofer IESE (DE), ZEA Partners (BE), MERIT (NL), AdaCore (FR), PEPITe (BE)

Printed
on 03/19/09 at 03:26:59 PM

Status:

[ ] Draft
[ ] To be reviewed
[ ] Proposal
[X] Final/Released

Confidentiality:

[X] Public - Intended for public use
[ ] Restricted - Intended for QUALOSS consortium only
[ ] Confidential - Intended for individual partner only

Deliverable ID: **D5.1**
Title:

Case Study Design and Pilot Projects

Deliverable: D5.1

Title: Case Study Design and Pilot Projects

Executive Summary:

One of the most important aspects in QualOSS is the validation of the QualOSS methodology and of the QualOSS assessment methods developed in the project, with a particular emphasis, in the initial case studies, on the validation of the standard QualOSS assessment method. To achieve such a validation, a set of case studies are devised to verify whether or not particular business goals are reached. A set of suitable pilot projects are identified to assess the applicability and utility of the QualOSS methodology and QualOSS methods.

People directly involved in the pilot projects will be interviewed to better understand the general context in which the QualOSS methodology and QualOSS assessment methods will be applied. These interviews will help to verify several hypotheses regarding user satisfaction and profitability. First, they will allow for comparing the results obtained from QualOSS assessments against human perception of the robustness and evolvability of the F/OSS endeavors assessed in each case study. Second, these interviews are also useful to study user satisfaction with the results obtained from the standard QualOSS assessment method and eventually of other more advanced QualOSS assessment methods.

CHANGE LOG

Ver.	Date	Author	Description
0.1	03/07/2007	José Ruiz	Initial version
0.2	15/05/2008	José Ruiz	Update with new methodology and case studies
0.3	06/06/2008	Rishab Ghosh / Rüdiger Glott / Kirsten Haaland	Update with hypotheses, revised methodology, and case studies
0.4	25/07/2008	José Ruiz / Jean-Christophe Deprez	Version for revision including all the proposed case studies
1.0	01/10/2008	José Ruiz / Rishab Ghosh / Jean-Christophe Deprez	Update with results of internal revision (JDC: Sanity Check)
2	09/02/2009	José Ruiz / Jean-Christophe Deprez	Update hypotheses and usage scenarios to address EC Reviewers' comments
2.1	25/02/2009	Rüdiger Glott / Jacques Flamand / José Ruiz	Update methodology, case studies and hypotheses
2.2	05/03/2009	Martín Soto / José Ruiz	Update document after first internal review
2.3	10/03/2009	Raimundas Matulevicius / José Ruiz	Second internal review. Correcting style and syntax errors.
2.4	19/03/2009	Jean-Christophe Deprez / Rüdiger Glott / José Ruiz	Complete Restructuring of D5.1
2.5	20/03/2009	Raimundas Matulevicius / Jean-Christophe Deprez / José Ruiz	Update after third internal review
2.6	20/03/2009	Jean-Christophe Deprez	Sanity Check before E.C. submission

TABLE OF CONTENTS

1. Introduction

1.1 Motivation

The strategic objective of the QualOSS project is to enhance the competitive position of the European software industry by providing methodologies and tools for improving their productivity and the quality of their software products. To achieve this objective, QualOSS notes that many organizations integrate Free *libre* Open Source Software (F/OSS) in their systems hence QualOSS's goal is to facilitate the identification of the most robust and evolvable F/OSS development endeavors whose F/OSS components are worth integrating in industrial software products and systems. In the end, the QualOSS methodology and its QualOSS assessment methods will ease the selection of high quality open source components. The overall effect will be increased productivity and higher dependability for the industrial software products integrating F/OSS components.

To achieve this goal, QualOSS proposes to build a high level methodology to benchmark the quality of open source software. In particular, the QualOSS project delivers an assessment methodology for gauging the evolvability and robustness of open source software endeavors.

This fifth work package (WP5) verifies that the QualOSS methodology, its QualOSS assessment methods and the accompanying tools can be used to verify whether particular business goals set for the studied projects are reached or not. The first task of WP5 (T5.1) presents the broad context of the case studies and selects the pilot projects to be analyzed. Furthermore, T5.1 lists the hypotheses that will be checked by each case study. The second task (T5.2) of WP5 consists in a set of interviews and the application of QualOSS assessment methods on the selected F/OSS endeavors. In a first phase, T5.2 applies the standard QualOSS assessment method on the F/OSS endeavors. In a second phase, the adaptation of the standard QualOSS assessment method into more advanced methods and the application of these advanced methods will be studied. The results obtained from T5.2 are used in the final task (T5.3) to report the results of the case studies and to argue the validity of the hypotheses being tested.

1.2 Objectives

The goal of T5.1 is to design the case studies to be conducted in T5.2 and T5.3. In particular, it must identify the hypotheses to test, and also design the general protocol to use in case studies. Second, T5.1 also describes pilot projects identified so far on which these studies will be conducted.

The main goal of the case studies developed in WP5 is to determine whether the QualOSS methodology and its assessment methods respond to industrial needs, in particular, they must satisfy industrial users, and it must also be determined that obtaining QualOSS assessment results is profitable in term of effort or monetary cost. The hypotheses detailing how to determine user satisfaction and profitability are presented in details in Section 4.

The list of objectives of Task 5.1 are therefore:

- To identify the goals to reach to determine whether the QualOSS methodology and its QualOSS assessment methods respond to industry's demand.
- For each goal, to enumerate the hypotheses to test in the case studies to verify whether or not the goals are reached.
- To present a general protocol that provides the general guidelines to use when conducting all case studies.
- To describe the industrial pilot projects on which the case studies will be conducted. A discussion on how the general protocol is applied in each each case study is also included.

1.3 Approach

This document defines the case studies to be conducted to validate the QualOSS platform and methodology, defining the background, objectives and goals of the study, as well as the hypotheses to validate. Deliverables D5.2 and D5.3 will apply the methodology and verify the obtained results to validate the objectives.

1.4 Strategy of WP5

T5.1 identifies the set of general hypotheses and a general protocol to follow in all our cases studies. The goal is that this general framework, made of hypotheses and general protocol, will steer our various hypotheses in a similar way so that their results can be aggregated at the end of WP5 to identify certain trends (e.g. all case studies were interested about code reliability and 80% of the users were satisfied with the results obtained from the standard QualOSS assessment method for code reliability). Such an aggregation from several case study results is only possible if the various studies were performed in a compatible way. By providing a set of general hypotheses and a general protocol, T5.1 proposes a first action to help conduct our case studies in a compatible fashion.

T5.2 uses the general hypotheses and the general protocol created during T5.1, and refines specific hypotheses and a specific protocol for each case study. It is important that T5.2 controls how the refinement takes place because this is how compatibility across case studies will be guaranteed. Once the refinement has produced the specific hypotheses and specific protocol, including the specific questionnaires for a particular case, then T5.2 also has the responsibility to collect the data for that case study.

T5.3 first analyzes the data for each case study individually and assesses whether the specific hypotheses verify or not. Subsequently, T5.3 compiles also the results of the various case studies together in order to identify trends. Clearly, prior to performing this aggregation, it should first be analyzed whether or not the case studies were finally conducted in a compatible way that allows for aggregating their results in a sensible way.

1.5 Structure of the Deliverable

The rest of the deliverable is structured as follows:

Section 2 provides the definition of terms used in the document. Section 3 specifies the way how the QualOSS methodology is applied to the different case studies. Section 4 presents the hypotheses to verify in order to determine the level of user satisfaction and the profitability of applying QualOSS assessment methods. Section 5 describes the general protocol to be used in the case studies. All the industrial pilot projects used in the case studies are described in Section 6. Section 7 specifies the exact F/OSS acquisition scenarios addressed by each pilot. Detailed information on F/OSS acquisition scenarios are presented in Deliverable D4.1. Finally, Section 8 gives some conclusions.

2. Terminology / Glossary

FIOSS Endeavor. F/OSS Endeavor is defined by 1) a set of work products, 2) the F/OSS community creating, updating and using these work products, 3) the tools used to act on these work products or to build or run the software product, and 4) the set of development processes executed by the community, these processes include rules and a division of labour accepted and followed by community members when interacting and creating work products.

3. QualOSS Methodology and WP5

The QualOSS methodology can be used at different levels of detail, with different degrees of effort involved, depending on 1) the time one wants to invest in analyzing (time to take a decision, time to adopt the

technology), 2) the cost involved, and 3) the risk in taking the decision (that is, the impact once the decision is made).

People will not invest a big effort in an analysis with restricted impact (such as selecting an e-mail reader for personal use of just a very few employee) because the decision can be later reverted without much work (replacing an e-mail reader can take less than one hour). On the other hand, there are decisions that may affect the viability of a project (such as the operating system to be used in an embedded system) where it is reasonable to spend weeks in its analysis.

Deliverable D4.1 presents the QualOSS methodology and defines the standard QualOSS assessment method, which is expected to take an effort of a few person-days (between 5 and 10 days) to obtain the assessment results for a F/OSS endeavor. This corresponds to a middleweight application of the QualOSS methodology. In some situation it is however possible to exploit existing results, if they are shared publicly. Accessing and reading results is much shorter. It corresponds to a lightweight QualOSS assessment method. When accessing existing results from previous assessments, the important aspect is to check the F/OSS acquisition scenario that drove this previous assessment, and make sure that they are in-line with the current situation.

Finally, a heavy-weight methodology is used when a company is willing to invest some initial effort in customizing the standard QualOSS assessment method into a specific QualOSS assessment method better adapted to the company's F/OSS acquisition scenarios. Creating a specific QualOSS assessment method requires initial effort, hence it is considered a heavyweight application of the QualOSS methodology. Such additional effort may be warranted if the assessment results are so important that the viability of a project of even of a company is at stake. Customization of the standard into a specific assessment method may also be warranted if a company wants to create a F/OSS assessment programme whereby all F/OSS endeavors whose F/OSS components are used in or by the company (product, services, or infrastructure) must be assessed in a specific way adapted to the company's vision of how to assess risk of collaborating with the F/OSS world. For instance, customization effort may add new characteristics to the quality model, or may require new indicators or measures to be taken. Spending additional effort is adequate when creating a custom-made F/OSS assessment programme in a company because this customization effort is done initially, and then the specific QualOSS assessment method will be applied many times in the future.

In WP5, the initial phase (from early 2009 until end of April 2009) focuses on testing the standard QualOSS assessment method on the various hypotheses listed in Section 4. If the outcomes of the case studies show that the results produced by the standard QualOSS assessment method are not satisfying to users, or that obtaining these results is not profitable, then customization will have to be applied on the standard QualOSS assessment method. These new specific QualOSS assessment methods will be studied in the second phase of WP5 (from April 2009 until the end of the project). These studies, in particular, when verifying the hypotheses on profitability, will examine the time taken to adapt the standard QualOSS assessment method into more specific methods.

4. Case Study Goals and Hypotheses

For the QualOSS methodology and its QualOSS assessment methods to be successful in industry, it is important to verify the following:

- User Satisfaction: the capability of the QualOSS methodology and its QualOSS assessment methods to satisfy users, with emphasis on such capability for the standard QualOSS assessment method.
- Profitability: the profitability of applying the standard QualOSS assessment method and the profitability of adapting the standard QualOSS assessment method into more specific QualOSS assessment methods.

Each of these two points are further explored in their respective subsection below. All hypotheses presented in these subsections can be applied to cases covering the various F/OSS endeavor acquisition scenarios

described in deliverable D4.1. In particular, case studies can be conducted on pilot projects covering 1) any F/OSS use context (integration in a product, a service or an infrastructure), 2) any F/OSS collaboration context (Full F/OSS Collaboration, F/OSS Fork, Exploit or Takeover), 3) any assessment mode (product comparison, version comparison or introspection), and 4) any F/OSS endeavor scope (whole, part of or set of F/OSS projects). Hence, for each industrial pilot, it is very important to identify accurately the F/OSS use context, the F/OSS collaboration context, the assessment mode, and the F/OSS endeavor scope being studied. Later, this will enable the research team to assess whether user satisfaction or profitability is dependent on the use context, collaboration context, mode or scope.

In addition to contexts, modes and scopes, it is also very important to record what QualOSS assessment method, and which version of it, was used to obtain the QualOSS assessment results examined by a case study.

To keep track of the various dimensions mentioned above, Section 7 fills the matrix below. The columns represent the F/OSS endeavor scope, while the rows specify the F/OSS collaboration context. The other information that needs to be given are: the F/OSS assessment mode, the use mode, and the QualOSS assessment methods used. In addition, it is also important to describe the specifics of the pilot, that is, the name of the firm, the name of the F/OSS project(s) of interest with the targeted version(s).

In summary, to specify a case study completely, the matrix below must be filled with tuples listing the following information:

- Firm's name, e.g. AdaCore
- (F/OSS project, version)*, e.g. ((GCC back-end, 4.2.4), (GCC back-end, 4.3.4))
- F/OSS assessment mode, e.g. version comparison
- F/OSS use e.g. will the selected F/OSS component be integrated in a product, a service or an infrastructure
- (name of the QualOSS assessment method, version) e.g. (standard QualOSS assessment method, v1.0)

	Whole F/OSS Project	Part of F/OSS Project	Set of F/OSS Projects
Full F/OSS Collaboration		(AdaCore, ((GCC-back-end, 4.2.4), (GCC-back-end, 4.3.4)), version-comparison, product-integration, (standard QualOSS assessment method, 1.0))	
F/OSS Fork			
F/OSS Exploit			
F/OSS Take-over			

Table 1: Dimensions in case studies

Section 7 fills this table with all pilot case studies presented in Section 6. The set of all possible values for the various dimension present above multiple to 108 possibilities. This does not even include dimensions such *decision impact* (e.g. little, significant, major, critical) or *cost of alternate solutions* (e.g. cheap, moderate, large, prohibitive). Clearly, values of these two dimension may have an influence on whether or

not the effort spent on a QualOSS assessment is judged profitable or not. For instance, one is likely to allow for a QualOSS assessment to take longer if the impact decision is critical and where there is no other alternative than the considered F/OSS components beside a complete re-implementation (hence, this would likely be considered prohibitive). If the values of these two dimensions are also included then the number of combinations explodes by a factor of 16 to 1728 possibilities.

Clearly, WP 5 does not have the ambition to test all these possibilities. Indeed, only a few cases (4 or 5) will be studied selected. The case studies explored are selected based on how realistic they are (the one currently listed in Section 7 all come from true industry situations) and also based on opportunities.

4.1 User Satisfaction

User satisfaction is studied in relation to the results of QualOSS assessments. In this context, the user is not asked to judge the applicability of QualOSS assessment methods, but rather to judge the quality of results obtained from QualOSS assessment method. Studying the applicability of QualOSS assessment methods, and in particular, its profitability, is covered by hypotheses in the next section.

In a first phase, user satisfaction is studied on the results obtained from the standard QualOSS assessment method. The outcome of these studies is to determine if and where the standard QualOSS assessment method has limitations.

In a second phase, the new advanced QualOSS assessment methods created to address the limitations of the standard QualOSS assessment method are also studied with regards to their capability to provide results satisfying users' needs.

With regard to user satisfaction, three groups of hypotheses are identified in order to judge whether the QualOSS methodology and its QualOSS assessment methods propose a satisfactory solution to users. As just mentioned, in a first phase, the three groups of hypotheses test the results produced by the standard QualOSS assessment method, and in a second phase, the results of advanced QualOSS assessment methods, which were adapted to address the limitation of the standard QualOSS assessment method.

To study satisfaction of results, the following topics must be addressed:

- Satisfaction with the Quality Model (Tree from the root until the leaf characteristics but not including indicators).
- Satisfaction with the Indicators.

These three topics are addressed in their respective subsections.

4.1.1 Hypotheses on the Quality Model Satisfaction

These hypotheses evaluate the satisfaction with the quality model used in a QualOSS assessment method. As previously mentioned, during the first phase case studies will focus on testing these hypotheses for results of the standard QualOSS assessment method and then, in a second phase, for results of specific QualOSS assessment methods. A first kind of hypotheses is about completeness:

- Quality Model Completeness - QMC-all: It is achieved if the Quality Model of the selected QualOSS assessment method includes all factors (node in the quality model tree) that one relies on to make a decision regarding F/OSS acquisition.

This hypothesis can be further broken down to study systematically if certain parts of the quality model are more satisfactory or interesting than others:

- QMC-wp: It is achieved if the Quality Model of the selected QualOSS assessment method include all characteristics of various work products (code, test and documentation) that one relies on to make decisions regarding F/OSS acquisition.

- QMC-cm: It is achieved if the Quality Model of the selected QualOSS assessment method includes all characteristics related to the community and its members that one relies to make decision regarding F/OSS acquisition.
- QMC-sp: It is achieved if the Quality Model of the selected QualOSS assessment method includes all characteristics of software processes that one relies on to make decisions regarding F/OSS acquisition.
- QMC-td: It is achieved if the Quality Model of the selected QualOSS assessment method includes all characteristics of tools and dependencies that one relies on to make decisions regarding F/OSS acquisition.

A second kind of hypotheses verifies the minimality of the quality model of the selected QualOSS assessment method:

- Quality Model Minimality – QMM-all: It is achieved if the Quality Model of the selected QualOSS assessment method contains the minimal set of characteristics required by the various roles for assessing the targeted F/OSS acquisition scenario(s) (which is/are determined by F/OSS acquisition context, assessment mode and F/OSS endeavor scope).
  - QMM-wp: It is achieved if the Quality Model of the selected QualOSS assessment method contains the minimal set of work product characteristics required by the various roles for assessing the targeted F/OSS acquisition scenario(s) (which is/are determined by F/OSS acquisition context, assessment mode and F/OSS endeavor scope).
  - QMM-cm: It is achieved if the Quality Model of the selected QualOSS assessment method contains the minimal set of community characteristics required by the various roles for assessing the targeted F/OSS acquisition scenario(s) (which is/are determined by F/OSS acquisition context, assessment mode and F/OSS endeavor scope).
  - QMM-sp: It is achieved if the Quality Model of the selected QualOSS assessment method contains the minimal set of software process characteristics required by the various roles for assessing the targeted F/OSS acquisition scenario(s) (which is/are determined by F/OSS acquisition context, assessment mode and F/OSS endeavor scope).
  - QMM-td: It is achieved if the Quality Model of the selected QualOSS assessment method contains the minimal set of tool/dependency characteristics required by the various roles for assessing the targeted F/OSS acquisition scenario(s) (which is/are determined by F/OSS acquisition context, assessment mode and F/OSS endeavor scope).

Additional information to take into account during case studies

Verifying the hypotheses QMC and QMM requires interviewing people in industry that fill the roles identified by the selected QualOSS assessment method. Prior to conducting interviews, it will be important to identify accurately the F/OSS acquisition context, the assessment mode and the F/OSS endeavor scope addressed in the case study.

It is also important to recall the F/OSS acquisition scenario(s) targeted by the selected QualOSS assessment method. The standard QualOSS assessment method was initially built to target all assessment modes and F/OSS endeavor scopes in a Full F/OSS Collaboration context.

When verifying the hypotheses above, two situations may arise: 1) the case study is in a Full F/OSS Collaboration context, or 2) the case study is NOT in a Full F/OSS Collaboration context.

In the first case, it is expected that QMC and QMM hypotheses verify to a high degree. However, the second situation is actually equivalent to studying whether the standard QualOSS assessment method is capable to target additional F/OSS acquisition contexts than originally intended. In such case, it is still worth verifying QMC and QMM hypotheses, but the same degree of satisfaction would not be expected. Indeed,

Full F/OSS collaboration is expected to be more extensive in its assessment needs than the other contexts, hence the QMM-all hypothesis is not expected to verify in other contexts. For example, in a F/OSS Fork context, QMM1.2, on community, will probably not verify because in a fork situation an enterprise will likely not care much about the past and existing community (hence, community related characteristics would not be considered relevant in F/OSS Fork context). However, QMM-1.1, on work product characteristics, may still be relevant to a F/OSS Fork context. Therefore, when faced with an actual case study, it is important to review the set of hypotheses proposed above to eliminate or refine them to fit the case study at hand.

For building the protocol, it is also worth noting that interview questionnaire on the hypotheses above does not need to include actual results of applying a QualOSS assessment method, but only to know the quality model of the QualOSS assessment method. So, for example, to test the hypotheses on the standard QualOSS assessment method, the interview questionnaire only needs to know about the quality model of the standard QualOSS assessment method, but actual results from the application of the standard QualOSS assessment method on the given case does not need to be available.

Incidentally, as mentioned above, the interview with regard to quality model satisfaction should only be performed once the accurate F/OSS acquisition is known. Otherwise, we could mix F/OSS acquisition scenarios and this could make the interview results useless.

4.1.2 Hypotheses on Indicator Satisfaction

Hypotheses in this section evaluate the satisfaction with the indicators used by a QualOSS assessment method. As previously mentioned, during the first phase case studies will focus on testing these hypotheses on the standard QualOSS assessment method, and then, in a second phase, on other specific QualOSS assessment methods created to address limitations of the standard method.

To study indicator satisfaction, we differentiate between:

- low-level indicators used to assess leaves in the quality model tree, and
- high-level indicators, composed of low-level indicators, to assess other nodes in the quality model tree.

This distinction is important because adapting low-level indicators is much more complex, while adapting high-level indicators can be apprehended by users. Thus, it seems more important for the provided set of low-level indicators to be judged satisfactory by users.

The first group of hypotheses relates to low-level indicators, while the next one verify the appropriateness of role-based aggregation for high-level indicators. Finally, the last group of hypotheses determines whether users are satisfied with how indicators are computed.

The first hypotheses study whether the low-level indicators of a QualOSS assessment method are adequate for a given F/OSS acquisition context, assessment mode and F/OSS endeavor scope:

- Leaf Characteristic Convincingness – LCC: It is achieved if the user finds convincing the value (and colour) of each leaf characteristic obtained from the selected QualOSS assessment method.
- Low-level Indicator Sufficiency – LIS: It is achieved if, for each leaf characteristic, the indicators found in the selected QualOSS assessment method are sufficient to assess the characteristic for the various roles in the targeted F/OSS acquisition scenario(s) (which is/are determined by F/OSS acquisition context, assessment mode and F/OSS endeavor scope).

It is worth noting that the hypothesis above must be tested on each leaf characteristic of a quality model. Using a granularity at the level of leaf characteristics enables detailed analysis where each leaf characteristic can be given an individual score with regard to user satisfaction.

The following hypothesis studies if role-based aggregation (for example by providing different weighted average per role) is an appropriate technique for computing high-level indicators:

- High-level Indicator Aggregation – HIA: It is achieved if users for each role identified in a QualOSS assessment method find appropriate the role-based aggregation to summarize scores from leaf

characteristics up to the root of the quality model tree (for example, the roles in the standard QualOSS assessment method are: product manager, project manager, analyst/developer, tester and technical writer).

The last group of hypotheses concerns user satisfaction with respect to how low-level indicators are computed, and with the level of details provided to justify the score (and colour) obtained by each indicator:

- Measure Aggregation Satisfaction – MAS: It is achieved if the rule for a given low-level indicator is a good way to assess risks related to the corresponding leaf characteristic.
- Indicator Drill-down Capability – IDC: It is achieved if, for each role specified in a QualOSS assessment method, the user is satisfied with results reaching a level of detail at the low-level indicator scores and colours.
- Measure Drill-down Capability – MDC: It is achieved if, for each role specified in a QualOSS assessment method, the user is satisfied with results reaching the level of detail of measures.

Note that this last hypothesis could turn out to be not sufficient for some users. This would then mean that users want yet a finer level of detail. In particular, they may want to know information on the input, the processing and the output of how a measure was generated.

Additional information to take into account during case studies

Testing the hypotheses above relies on interviewing people filling the various roles defined in a QualOSS assessment method. Furthermore, studying the validity of these hypotheses requires sharing the results of QualOSS assessment with users in order to get their opinion. In turn, these hypotheses can only be fully tested once you have obtained the desired QualOSS assessment results.

One possible approach to study (LCC) convincingness of value (and colour) attributed to each leaf characteristic is to take the opinion of a user with regards to each characteristic, and then compare it to the score obtained using QualOSS. When scores are different, we can then check if the user is convinced by the score proposed by the QualOSS assessment and whether or not he has switched his opinion to match with the QualOSS assessment results.

Regarding the other five hypotheses (LIS, HIA, MAS, IDC and MDC), the number of measures involved in indicators is too large to ask users to give their opinion on each indicator and measure. Furthermore, the satisfaction of measure aggregation and of the details of the results can only be apprehended in front of actual results. In other words, it would be too hard for a user to determine if a particular measure aggregation used in an indicator rule is satisfactory, or if results are detailed sufficiently without being shown actual results.

So, an appropriate way to study these five hypotheses above is to confront users with actual results, and then ask them to analyze certain indicators, first to verify that they agree with how they aggregate measures and specify thresholds, and second, to let them drill down in the details of various indicators and their measures.

Another very interesting point on testing sufficiency (LIS) is to figure out if indicators that can be obtained automatically (that is, through automatic measurement procedures) are sufficient to assess a characteristic, or if the manual measurements add value and are worth the added effort.

4.2 Profitability

Studying profitability consists in verifying how much QualOSS assessment results benefit users in comparison to how much is required to obtain them. In particular, assessing profitability is a mix of 1) how important a user views each characteristic, 2) the anticipated amount of effort needed to obtain each of these characteristics and indicators, and 3) how much effort is really needed to assess these characteristics and indicators.

The identified hypotheses on profitability are:

- Characteristic Effort Profitability (CEP) – It is achieved if characteristics of a QualOSS assessment method are profitable, that is, they take less time to compute than perceived by the user.
- Indicator Effort Profitability (IEP) – It is achieved if indicators of a QualOSS assessment method are profitable, that is, they take less time to compute than perceived by the user.
- Specific Assessment Profitability (SAP) – It is achieved if adapting the standard QualOSS assessment method into a specific assessment method and then applying this specific assessment method is profitable, that is, if it takes less time than perceived by the user.

As it was true for user satisfaction, profitability is influenced first by the actual QualOSS assessment method selected to obtain results, and second, by the specifics of an F/OSS acquisition scenario, that is, its context, mode, and scope. For example, the standard QualOSS assessment method may turn out to be profitable in a Full F/OSS Collaboration context but not in a F/OSS Fork context. Indeed, the standard QualOSS assessment method assesses not only the products and work products but also the community and software process. These last two aspects are likely of little interest to a F/OSS Fork context. So, effort spent on evaluating software processes and various community characteristics may be considered wasted time that renders the QualOSS results unprofitable. Fortunately, in such a scenario, merely eliminating the useless part of the assessment on software processes and community may switch the situation back to profitable.

Another more subtle study could consist of verifying if the the standard QualOSS assessment method is profitable in Full F/OSS Collaboration, but when different F/OSS endeavor scopes are used. Tools focus on measuring the whole dataset found in a repository, but they may have difficulties taking measures on just a part of that repository repository, for example, requiring tedious manual operations prior to taking the measures. This could then cause an assessment effort for a part of a F/OSS endeavor to explode.

Additional information to take into account during case studies

CEP can be studied by asking users how much time or money they would spend to know the assessment of each characteristic. However, there are too many indicator to proceed the same way with IEP. In turn, the appropriate approach is probably to ask a user if the time usually taken to assess a characteristic is adequate; if it is not, it is then time to drill down to the level of indicators to find out which indicator they would prefer spending time and money on.

5. General Protocol

This general protocol describes at a high level the various steps to take when conducting the QualOSS case studies. Clearly, we could design a protocol to verify each hypothesis in isolation. However, it would be too time consuming, and the interviewees from firms would likely not adhere to this approach. Therefore, the approach suggested by this general protocol is to compile the list of questions needed to validate the various hypotheses into a comprehensive and targeted interview guideline with some questionnaire-like elements in order to evaluate characteristics and indicators. Depending on the specifics of the case study, it is anticipated that each interviewee will have to answer two or three questionnaires during the various steps of the protocol.

The steps of the Protocol are as follow.

Step 0. Identify accurately the case to study.

In the terms of the QualOSS methodology, this step consists of executing the first task of the assessment process, that is, *Initiate the Assessment*.

Identify the context. From the view point of case studies, the important outcomes of this step are to identify:

- firm's name,
- F/OSS endeavor(s) to study (names and versions),
- F/OSS collaboration context,
- F/OSS assessment mode, and

- F/OSS endeavors scope.

Identify the objectives. The task of *initiating an assessment* has additional objectives with respect to the business goals. In particular, it requires:

- the identification of the reason for the assessment, and also
- a description of how QualOSS assessment results are planned to be used.

Although not directly useful to conduct case studies, these two points are very useful to understand the complete context in which a QualOSS assessment is performed. This will later help to present and to explain the results of the case studies.

Planning. Step 0 must:

- identify who will be interviewed from the firm. This list does not need to be frozen. However, it should identify all potential interviewees
- identify the selected people's roles in the company and eventually their roles in the F/OSS endeavor(s) of interest. In particular, if they are already involved in the FlOSS endeavor(s) in question or if they are not, the roles they may wish to have in the future
- produce an overview planning of the case study, which specifies when remaining Steps of the protocol are planned to start and to end

Actions. Step 0 requires the following actions in order to get the information mentioned above:

- Conducting interviews to capture all information mentioned above. A draft questionnaire for guiding this type of interview is provided in Appendix A.
- Analyzing interview information to extract the list of outcomes of Step 0 specified above.

Step 1. Adapt material to the current case study.

The existing material must be adapted to fit with the current case study. In order to adapt this material adequately, the following sub-steps must be performed in order:

Sub-step 1.1: Refine the hypotheses of Section 4 to mention the exact F/OSS context, mode, scope as well as the QualOSS assessment method under study.

Sub-step 1.2: Refine the overall strategies for verifying hypotheses into operational strategies adapted to the current case study.

Currently, a brief description of overall strategies to use for verifying hypotheses are suggested at the end of each subsection of Section 4 in clauses titled “Additional information to take into account during case studies”. During Step 1.2, it is time to refine the overall strategies into an operational strategy for the given case.

Sub-step 1.3: Guidelines and questionnaire templates for conducting the interviews of Steps 2 and 4 must be instantiated into specific questionnaires. A part of questionnaire template for pre-QualOSS assessment is presented in Appendix B. It is still evolving. A template for post-QualOSS assessment is currently being developed.

The important task in sub-step 1.3 is to verify that the operational questionnaire is in sync with the refined hypotheses and the operational strategies by doing the following:

- First, a mapping must be done between specific hypotheses and questions whose answers are needed to validate these hypotheses.
- Second, based on the specific strategy as well as the constraints of people's schedule, decision on how to operate the interviews is devised. For example, decisions regarding when to schedule the set of interviews for the case study must be made, how to partition an interview e.g. into a phone interview followed by paper questionnaire, etc.

Sub-step 1.4: Schedule the interviews with the various interviewees.

Step 2: Capture information pre-QualOSS assessment.

This step consists of conducting interviews based on the operational questionnaires for pre-QualOSS assessment interviews. Furthermore, an initial consolidation of the results also takes place.

Step 3: Execute the selected QualOSS assessment method for the particular pilot project.

This step executes the QualOSS assessment method on the F/OSS endeavor(s) of interest to the industrial pilot.

Step 4: Capture information post-QualOSS assessment.

This step consists of conducting interviews based on the operational questionnaires for post-QualOSS assessment interviews. Furthermore, an initial consolidation of the results also takes place.

Step 5: Compile the Information from interviews to facilitate the verification of hypotheses.

The consolidation of the information resulting from Steps 2 and 4 is done during this step. It is expected that the results from the QualOSS assessment are already structured to facilitate the verification of hypotheses in the next step.

Step 6: Verify hypotheses and present initial conclusions.

Every hypothesis is verified and a first interpretation of results is proposed. This interpretation makes use of information on the context, mode and scope as well as on the QualOSS assessment method selected to present a fine grain analysis of the results.

Step 7: Elaborate final conclusions based on feedback from interviewees.

Results and initial conclusions are presented to all interviewees of the pilot and their impressions are collected in order to elaborate the final conclusions of the case study.

Step 8: Compare results of all case studies.

The case studies conducted in WP5 are too few to infer any generalization. However, comparing and contrasting the results from the various case studies may help identifying trends. Knowing these trends may help people to make a better use of the QualOSS methodology and its assessment methods. It tests also the applicability of the QualOSS assessment to a range of very different interests that may trigger a QualOSS assessment.

6. Pilots for Case Studies

The different industrial pilots are presented following a common schema. They describe the actors involved, the F/OSS endeavors under analysis, the objectives, and how the results of the QualOSS assessment will be used. Each of the following subsections structures its description as follows.

1. Executive summary. Short outline of the case study.
2. Background
   a) Overview of the F/OSS endeavor(s) under analysis
   b) Overview of the actor (company or person) directing the case study
3. Business needs, objectives and goals. This section will describe the help that the company or person involved in the case study expects from QualOSS.

6.1 AdaCore/GCC Version to Use as Back End for GNAT

6.1.1 Executive summary

AdaCore uses the GNU Compiler Collection (GCC) back end for its flagship product, the GNAT compiler for Ada. Its target market is primarily interested in robustness and reliability, so quality metrics for the GCC back

end related to its robustness are of paramount importance. AdaCore needs to upgrade to new versions of the GCC back end regularly (every one or two years) and the process to select the appropriate GCC version is critical for its business.

This case study will analyze the different versions of the GCC back end in the context of AdaCore's usage, and it will study whether the QualOSS methodology can help improving the decision process and increasing the confidence in the selected version.

6.1.2 Background

GCC includes front ends for various programming languages (C, C++, Objective-C, Fortran, Java, etc.) and back end code generators for many target architectures (ARM, x86, PowerPC, SPARC, etc.) and most platforms available today (different flavors of Unix, Windows, embedded platforms, etc.). The GNAT Ada compiler uses the GCC back end code generators and provides the Ada front end to GCC.

GCC is one of the most important tools in the development of free software. GCC is free software, distributed under the GNU General Public License (GNU GPL), and its first release was in 1987. It runs on most platforms available today, and can produce output for many types of processors.

There is a very big community of users/maintainers behind GCC; there are hundreds of different contributors, and tens of people maintaining and contributing to GCC on a daily basis. Most contributors work for big companies with vested interests in making GCC work in their environment (RedHat, Google, IBM, AMD, Intel, etc.).

AdaCore, founded in 1994, provides commercial, open software solutions for Ada to build robust and reliable software. Currently, AdaCore uses GCC version 4.1 for the GNAT release (released in Feb. 2007), and it is now in the process of transitioning to GCC 4.3 (released in March 2008).

AdaCore has used many different GCC back ends in its history (4.1.3, 3.4.3, 3.2.3, 2.8.1, etc.), and the way to choose the appropriate GCC version has been based on verifying the reputation of the different versions. This decision is critical for AdaCore in terms of both risk (not enough robustness of the compiler distributed to customers) and access to recent enhancements (new fixes, targets, optimizations, etc.). In addition, the cost of migration (around 18-24 Person-Month) is not negligible, so it is of crucial importance to have a good degree of certainty about the new versions to be used as back-end code generators.

6.1.3 Business needs, objectives and goals

AdaCore needs to use the most recent version of the GCC back-end in order to take advantage of recent fixes and improvements, and new targets made available in each new release. However, the target market of Ada is that of reliable systems, where robustness is a must. Hence, the GCC back-end chosen for GNAT releases need to provide the desired robustness.

The main goal of this case study is to use the QualOSS platform and methodology to get information about the GCC back-end status in terms of robustness.

This information will be primarily used by the group of AdaCore's back-end experts. This is a group of around six people at AdaCore that follow the GCC patches and mailing lists on a daily basis, and they get a feeling about the status of the GCC releases. They analyze the status (in terms or new problems, fixed problems, new features, and new targets) and the “reputation” of the GCC release (based on the comments exchanged in the GCC mailing lists).

Later, when the GCC back-end is being integrated within the GNAT technology, AdaCore pass its test suite to validate the whole tool-chain. At this stage AdaCore would detect potential problems related to the GCC back-end. The risk of detecting problems at this late stage is twofold: first, the migration effort is around 18-24 Person-Month, so if AdaCore considered to drop a problematic GCC back-end it would be a waste of effort; second, it may be the case that it is AdaCore's customers the ones detecting the problems, with the risk of customers generating bad code and the damage to AdaCore's image.

Therefore, AdaCore's back-end experts would like to have most of the information about the robustness of the GCC back-end as early as possible. This way, they can use this information to decide about the new GCC version to be used, and to increase their confidence when the decision has been taken. This is where the QualOSS platform and methodology can help.

More concretely, the goal in this case study is to analyze version 4.3 of the GCC back-end to assess whether it is suitable or not in terms of robustness. Another interesting analysis that could be done is about other GCC versions. For example, AdaCore skipped version 4.2 because it did not provide many new features and it was considered not robust enough. It would be interesting to see whether the use of the QualOSS methodology could give some indication.

A second group that could be interested in the results provided by QualOSS is the AdaCore Quality Assurance team. They would like to know in advance if a new release (GNAT front-end plus GCC back-end) may be problematic in order to devise a proper validation campaign, a potential beta release to increase field testing, and a backup plan in case something goes wrong.

A third group is AdaCore's customers who may obviously be interested in the quality of the back-end code generator. Many customers are dealing with the trade-off between robust versus efficient code (old versions are better tested and tend to be more robust, while new versions come with better optimizations). They would be interested in any indication about the quality of the code generator, and QualOSS can give them such indications.

A fourth group is the GCC community, who is obviously interested in any feedback about the evolution of GCC in terms of robustness.

In order to determine whether the QualOSS methodology improves the decision process and/or increases the confidence in the selected version, the QualOSS results will be provided to the GCC experts at AdaCore (who can play both the role of GCC developers and GCC back-end integrators) so that they can manipulate them and assess their validity based on their expertise.

6.2 Freecode/Asterisk

6.2.1 Executive summary

Freecode is a Norwegian company that develops F/OSS but it has its focus on the provision of training courses, advice, and implementation and maintenance services for a number of F/OSS products, including Operating systems such as Linux, FreeBSD, OpenBSD and NetBSD; databases (PostgreSQL and MySQL); application servers such as Tomcat, JBoss and Zope, Apache web server; publishing solutions (Joomla!, Drupal, Typo3, eZ publish, Master Desk); Samba file, print and domain server; firewalls based on Linux or OpenBSD; IDSer (Snort and STAT); SAN, NAS and backup solutions (www.freecode.no). The company has a very strong attitude towards F/OSS and emphasizes the gain of control achieved by their clients through the use of F/OSS.

Another field in which Freecode is active and that provides the context in which this case study is carried out is the provision and implementation of complete telecommunication infrastructures. The company is specialized in Asterisk. The purpose of the interview will be to investigate how Freecode performs within the Asterisk endeavor in order to secure and improve the quality of its products and services. As a side aspect we will also make a comparison between an old Asterisk version (1.4.17) and the latest release (6.1.04). The reason to compare these two versions is that we have also access to a user community of Asterisk 1.4.17 that collaborates with Freecode. It enables us to cover developers, service providers, and users, thereby investigating the reasons for keeping the old version and comparing the old version to the newest one. In other words: the Freecode case provides important input for other case studies on the same F/OSS endeavor.

6.2.2 Background

Asterisk is the world's leading open source Private Branch eXchange (PBX) telephony engine and telephony applications toolkit (www.asterisk.org). Originally (in 1999), it has been developed and is distributed by Digium. Asterisk is written in C and, as typical for many F/OSS products, it is largely platform independent. Asterisk is released under a dual license model, using the GNU General Public License (GPL) as a free software license and as a proprietary software license to permit licensees to distribute proprietary unpublished system components (Wikipedia.org).

There is a very big community of users/maintainers behind Asterisk, including individuals as well as firms.

As outlined above, PBX is only a part of Freecode's activities, although Freecode is clearly an Asterisk expert that deals with this program now for a number of years, and it has deep understanding of many releases. Freecode usually focuses on implementing Asterisk as available at the project's homepage (http://www.asterisk.org/) but it has also developed a number of patches for Asterisk that have been created in order to counter security issues in the standard Asterisk distribution. However, the company did not succeed so far to convince the Asterisk community to incorporate its patches, so that Freecode now owns a set of patches that, if implemented, forms a sort of fork of Asterisk. However, Freecode highly appreciates community support and therefore prefers to implement the public available trunk version of Asterisk instead of their own version, which does not receive support from the community.

6.2.3 Business needs, objectives and goals

Freecode finds itself in sort of a dilemma when a client asks for the implementation of Asterisk. The company has strong expertise in F/OSS and encountered a number of security issues aligned with Asterisk. These issues seem to be typical for PBX programs in general. There are two ways to deal with these problems: either 1) removing the problematic components from the Asterisk version that is implemented at the client's site, or 2) fixing the bugs. Although, for obvious reasons, Freecode prefers the second option, it often has to pick the first one because the company could not convince the community to incorporate a number of patches Freecode has developed in order to fill the security gaps. This problem exists although the company highly appreciates collaborating with the Asterisk community and supports it actively.

Freecode is therefore not only interested in 1) a comparison and evaluation of different versions of Asterisk with regard to their services and service quality, but also in 2) a SWOT (Strengths, Weaknesses, Opportunities and Threats) analysis of their engagement in the Asterisk endeavor. The company wants to know if collaboration with the community can be improved and what could be done in order to get their patches incorporated in the trunk version of Asterisk. The latter would save the company both effort and money when implementing Asterisk.

6.3 Océ PRISMAspool to use the LPR client of the yanolc project

6.3.1 Executive summary

Océ delivers the Océ PRISMAspool software product that provides an enhanced spooling system for Windows environments. PRISMAspool uses the Line Printer Remote (LPR) protocol in 2 use cases. The implementation currently uses the LPR client delivered with the Windows operating systems. A new LPR client should be delivered with a next version of PRISMAspool in order to solve some restrictions. This case study will analyze the adequacy of using a subset of the Open Source project *yanolc* (Yet ANOther LPR Client) as a basis for this delivery. As PRISMAspool is used for professional printing, for transactional and mailing environment, a top most quality and reliability is required.

6.3.2 Background

Océ supplies digital printing systems, software and services for the production, reproduction, distribution and management of documents, in colour and black-and-white, in small format and in wide format, for

professional users in offices, educational institutions, industry, advertising and the graphics arts market. Océ is a leading supplier of these systems world-wide.

Océ PRISMAspool is a software product that provides an easy-to-use yet powerful spooling system for Windows environments. It provides all features required for heavy-duty professional printing and is mainly used in transactional and mailing applications, suited both for enterprises and print service providers.

PRISMAspool uses the LPR protocol in several use cases. Those use cases are currently realized using the LPR client delivered with the Windows operating systems:

- Use case 1. Submission of a print job to the PRISMAspool spool server; the MS Windows lpr command (lpr.exe) can be used for submission. However, the size of the print job is limited to 2 Gb, which is a critical limitation in case of colour production jobs.
- Use case 2. Submission from PRISMAspool to a so-called “printer command”. It is possible to configure PRISMAspool to drive printers using different connection protocols. One of those is the so-called “command printer” - meaning that printing the job leads to the execution of a command that can be specifically configured for the printer. A commonly used command is lpr.exe because some printers behave as an LPR server. Again, the MS Windows lpr command is used with its limitation with respect to job size, but also with the fact that the “User Identification” is not supported as command parameter. This is also a critical limitation because the identification is displayed on the panel of Océ printers.

A new LPR client should be delivered in a next version of PRISMAspool; as an alternative to a proprietary development, Océ investigates the possibility of using the Open Source project yanolc as a basis for this development. yanolc has been chosen because of its adequate functionality, but also because its licensing scheme satisfies Océ's needs.

6.3.3 Business needs, objectives and goals

As mentioned before, Océ needs to deliver a new LPR client, and they consider using the project yanolc as an alternative to a proprietary development. Their business needs can be summarized as follows:

- Deliver only an lpr client for job submission without additional functionality such as job follow-up.
- The lpr client must be at least as reliable as the client delivered with the Windows system.
- The lpr client must provide at least the same functionality as the client delivered with the Windows system, without the restrictions mentioned here above.
- In case of any bugs / issues / extension request, Océ R&D must have the same reaction capability as for its proprietary developments.
- Taking the know-how and possibly adapting the Open Source project must be significantly less expensive than developing a proprietary LPR client component.

The goal of this case study is to analyze version 1.2.11 of the yanolc project, and its dependencies, to help Océ taking the decision of going to an Open Source-based versus a proprietary development.

The product manager is primarily interested in the reliability of the project in order to guarantee the topmost quality to the Océ customer. Additionally, the benefit in term of development cost (necessary adaptation of yanolc versus proprietary development) must be significant.

The project developers are primarily interested in the maintainability and the extensibility of the code.

6.4 CETIC/PCF/PFB (ON HOLD)

6.4.1 Executive summary

This series of case study analyzes the usefulness of the advice on how to turn the Tabellio project into a robust and evolvable FlOSS endeavors. These pieces of advice will be elaborate by applying the QualOSS methodology to the Tabellio project.

The Tabellio project consists of developing an information system for drafting, managing and publishing legislative and parliamentary documents. The final solution must also include other generic tools commonly found in public local administration to enable the sharing of information between regional and local administrative authorities. Tabellio is mandated by two Belgian parliaments and is being developed by several consulting firms. CETIC, a QualOSS partner, acts as the client quality assurance team that advises the two Belgian parliaments on software product and software process quality matters. One of CETIC's task consists of advising and guiding the two parliaments to increase the chances of the Tabellio project to become a robust and evolvable FlOSS endeavor. In turn, CETIC will apply the QualOSS methodology and the results will be inspected to determine their perceived usefulness.

6.4.2 Background

The Tabellio project consists of developing a parliamentary software application for two Belgian parliaments. The resulting web application must include an information system for drafting, managing and publishing legislative and parliamentary documents. It must also include other generic tools commonly found in public local administration to enable the sharing of information between regional and local administrative authorities. This whole project is split in several parts and the development of each part was attributed to different consulting firms.

The contract of Tabellio imposes to release the final solution under a F/OSS license. However, nothing else regarding the composition or the structure of the final F/OSS endeavor(s) is mentioned in the contract. From a technological stand, the contract suggests but does not impose the use of the Python, Zope and Plone stack and indeed, the selected contractors have all elected that suggested solution.

Tabellio involves the following types of teams:

- The *client* sets the requirements of the desired application including the need to release it under a F/OSS license. This role is filled by two Belgian parliaments, namely, the “Parlement de la Communauté Française de Belgique” and the “Parlement Francophone Bruxellois”.
- The *client quality team* advises the client on what quality standards to apply during the Tabellio project and also performs certain quality checks ordered by the client. This role is filled by CETIC.
- The *consulting firms* analyze users' needs and implement the application. This roles is filled by Software AG, Entr'Ouvert, and Pragsys SARL. The largest part of the solution is implemented by Software AG. Each consulting firm has its own quality team. These quality teams propose and agree with the quality standard set by the client quality team for Tabellio.

The Tabellio project started in June 2007 and as specified in the contract, the first initial delivery of the application was initially planned for November 2008. However, it is already acknowledged that this initial delivery will be late. A date in Spring 2009 seems more likely. This initial delivery of the application will be considered “under review” for one calendar year during which the consulting firms will be obligated to correct bugs and to adapt the functionality to match with requirements free of cost. After this one-year time span, the delivery will be considered final. At that moment, the contract requires another year of cost-free corrective maintenance from the consulting firms.

6.4.3 Business needs, objectives and goals

The two main reasons for wanting to release the final solution of Tabellio under a F/OSS license are:

- to avoid consulting firm lock-in for the development of the application and for future evolution to the application, and
- to collaborate with other parliaments.

These two objectives were already considered as reached when the project was awarded to the various consulting firms. Firstly, three independent consulting firms are working on the development of the initial

solution; hence, the first point above is fulfilled. Secondly, the two Belgian parliaments started their collaboration during Tabellio; hence, the second point above is also fulfilled.

However, the two parliaments agree that Tabellio could be even more successful with regards to these two objectives. In particular, they hope that other francophone parliaments, in particular, those from developing nations, will consider using the Tabellio application or at least certain Tabellio components. In addition, Tabellio would be considered yet more successful if other parliaments not only used but also contributed corrections or additional functionality to the Tabellio application.

Concerning the avoidance of consulting firm lock-in, Tabellio would be considered more successful if consulting firms other than the three already involved would later win bids of future developments.

Recognizing that Tabellio could increase its success beyond the two points mentioned above, the two Belgian parliament have requested CETIC to provide advice regarding the modality for setting up Tabellio so that it would increase its chances of becoming a robust and evolvable F/OSS endeavor. In this scenario, the emphasis of robustness and evolvability will be put on the parliaments' objectives i.e., collaboration with other parliaments and avoidance of consulting firm lock-in.

Tabellio and QualOSS are due to end around the same time, it is therefore unlikely that Tabellio will have already been adopted by additional parliaments and even less chances that these parliament will have contributed to it. Likewise, the two Belgian parliaments will not have had the time to submit new calls for tender to verify that new consulting firms could respond. In summary, we will lack the time to obtain clear and objective evidences to show that the advice from the QualOSS methodology were fruitful. In turn, we plan for these alternate case studies:

From the client's viewpoint (the two parliaments):

- Determine the perceived usefulness of the advice, i.e., how the two Belgian parliaments perceived that the given advice were useful with regards to their objectives.
- Determine the perceived completeness and coherence of the advice.

From the client quality team's viewpoint (CETIC):

- Determine the perceived systematic approach and completeness of the QualOSS methodology.
- Determine the efficiency with which the QualOSS methodology could be applied.

From the consulting firms viewpoint (Software AG, Entr'Ouvert, Pragsys SARL)

- Determine the degree to which the pieces of advice can feasibly applied by the consulting firms.
- Determine the degree to which the pieces of advice are presented convincingly.

In addition, to the studies above, objective evidences of new collaborations, or of new consulting firms performing new development, will be reported if they happen earlier than anticipated.

7. FLOSS ACQUISITION SCENARIOS AND QUALOSS CASE STUDIES

Table 1 presented in Section 4 explains how we are keeping track of the various factors that can potentially affect the results of case studies. Below, this table is presented with the list of case studies proposed. Those in regular black font are fully defined case studies that have already started while those in grey, italic font need to be further defined and have not been initiated yet.

	Sets of projects	**Single FIOSS project**	**Part of a FIOSS project**
Full FIOSS Collaboration	Ref. sec. 6.4 (CETIC/PCF/PFB, (Plone Parliamentary App., v1), Introspection, Product-Integration, (Standard QualOSS assessment method, v1.0))	Ref. sec. 6.2 (Freecode, (Asterisk, latest), Introspection, Service-Integration, (Standard QualOSS assessment method, v1.0))	Ref. sec. 6.1 (AdaCore, ((GCC-back-end, 4.2.4), (GCC-back-end, 4.3.4)), Version-Comparison, Product-Integration, (Standard QualOSS assessment method, v1.0))
FIOSS Exploitation		*(CETIC ,* *(OBM Groupware, ?)* *Endeavor-Comparison,*	

		Infrastructure-Integration, (standard QualOSS assessment method, v1.0))	
FIOSS Fork			Ref sec. 6.3 (Océ, (yanolc, 1.2.11), Endeavor-Comparison, Product-Integration, (Standard QualOSS assessment method, v1.0))
FIOSS Takeover			

Table 2: Scenarios addressed by case studies

The case studies testing the user satisfaction and profitability of the standard QualOSS assessment method at AdaCore, Océ and Freecode are planned to start in February 2009 and to run until April 2009. Other studies for the standard QualOSS assessment method and for newly developed, specific QualOSS assessment methods will take place from May 2009 until the end of the QualOSS project in November 2009.

It is worth noting that FlOSS takeover are rare and in turn, it is doubtful that any such a case can be identified in the remaining time-frame of the QualOSS project.

8. Overall Conclusions

User satisfaction and profitability are paramount if the QualOSS methodology, its assessment methods and its platform are to be used in industry. Task 5.1 presents a list of hypotheses to verify user satisfaction and profitability of the QualOSS assessment methods and the QualOSS assessment results they produce.

Furthermore, a general protocol specifies the general guideline to follow in each case studies. Certain of the steps in the protocol are generic, and they must be refined into operational actions prior to conducting the case studies. This refinement effort is left to Task 5.2.

Deliverable D4.1 defined a set of F/OSS acquisition scenario according to several different dimensions. The number of possible combinations is 108, so not all combinations can be addressed during WP5. The set of case studies identified in this deliverable address the scenarios considered important. Furthermore, all case studies have their roots in real industrial scenarios.

The AdaCore/GCC, the Océ Lab/Yanolc and the Freecode/Asterisk case studies have all started in February and are planned to end in April 2009. In addition to these three case studies, another two industrial pilot will be studied in the period from May to November 2009.

It is also worth emphasizing that one hypothesis is concerned with the profitability of customizing the standard QualOSS assessment method into more specific QualOSS assessment method.

The set of scenarios and hypotheses covered in the case studies describe in this document provide a wide range of exercises to study the QualOSS methodology and its assessment method. The use of interviews at different moments, and with different people, provides a very powerful tool to assess what is expected, what is needed, and what is valuable. All these factors build up an exhaustive, rigorous, and attractive validation for the QualOSS methodology.

The results of these case studies will show the degree of user satisfaction and profitability achieved by the proposed QualOSS methodology. Comparing and contrasting the results from the various case studies help in two ways. First, it may identify ways to improve the QualOSS quality model. Second, it may identify better ways to use the QualOSS methodology and its assessment methods.

9. Appendix A – Sample Questions to Ask in Step 0

The initial interview may take place face-to-face, by email, by phone or a mix of these techniques. The following questions must be answered after these initial interviews:

- To which degree is your firm dealing with F/OSS? We'd like to largely differentiate the following options:
    - Is your firm fully integrating F/OSS development processes (and maybe F/OSS communities) in its way of doing business? Is F/OSS more or less the core of your firm's business model(s)?
    - Does your firm integrate F/OSS components in its products without any further interest in F/OSS and collaboration with F/OSS communities?
    - Does your firm integrate F/OSS components in its product(s) by creating a fork of the current F/OSS component (i.e. is there interest in an "own" version of the F/OSS component, being independent from the existing F/OSS community around this component)?
    - Does / did your firm want to take over an F/OSS product and community in order to have more control on how it develops?
- Does your firm want to integrate F/OSS on the firm's infrastructure?
- Is your firm interested in selling services on F/OSS component(s) without engaging too much in the development of the F/OSS product itself?
- What interest does your firm have in the interviews, i.e. what do you expect from us? Would you like us to provide some guidance on how to monitor your firm's involvement in an existing F/OSS product (and, maybe, its community) or how to set up an own F/OSS product with a community related to it? (This would go in the direction of a SWOT analysis, covering the examination of work products, community members, tools and software processes)

10. Appendix B – Sample Interview Guideline for Step 1

The main interview is carried out, covering all aspects that are measured in the standard QualOSS assessment method. In this way, we make sure that the QualOSS assessment method is well addresses by the interview (including quality indicators, characteristics, and metrics), and we can compare the human and the QualOSS model by means of examining:

- how the respondent's organization usually organizes an evaluation of a F/OSS endeavor and how much time and money is usually needed for this,
- how important these measures are evaluated by the interviewee,
- what characteristics he misses,
- how he personally assesses the F/OSS endeavor under scrutiny with regard to the characteristics measured by the standard QualOSS assessment method
  - the main instrument in order to measure the human perception of the software quality characteristics is a form in which the interviewee assigns colours according to the QualOSS assessment method to each measured item (see Appendix C)
  - this form must be filled in by the interviewee twice, the first time in the beginning of the interview (after a short intro phase) and again at the end of the interview, without seeing the responses given in the first assessment; the purpose of this double check is to see how the conversation about quality characteristics and measures influences the human perception of these measures. For instance, in the beginning a respondent might think that measures regarding the community of the F/OSS endeavor are unimportant, but after being confronted with the items that can be measured in this respect he might think that some of these characteristics are very important. We assume that the same will happen when a company later decides to run the QualOSS assessment platform and becomes more familiar with the measures. If it turns out that the conversation about the characteristics indeed biases the human perception the manual provided with the QualOSS platform should take account of this and prepare the user so that he knows that familiarizing with the measures before the platform runs is very important. Alternatively, the user might be advised to run the platform repeatedly, with different settings after he might have found that he ignored some measures in the first run that later turned out to be important
  - besides the bias that possibly results from the interview / measurement procedure itself, the human perception of software quality measures must also be validated with regard to deviances of the human perception of the characteristics that are measured from the QualOSS assessment. For this purpose, the evaluation forms will be cross-checked by the QualOSS research team. In a first step, the interviewer summarizes all statements the interviewee made regarding the different quality characteristics. This summary will be passed on to a research team member that does not know what colours the interviewee assigned to the different characteristics. Based on the statements of the interviewee, the neutral team member tries to guess what colour the interviewee has assigned to the various characteristics. Thus, this cross-check is basically a test of coherence of the interviewee's assessment. Coherence is important because it is implied by the QualOSS assessment, i.e. the QualOSS platform operates on the basis of a coherent set of measures and thresholds. This, in turn, requires that we are able to figure out how the interviewee would have replied if his perception was coherent, otherwise the comparison of human perception and the QualOSS measurement would be an arbitrary subject. In other words: In case of discrepancies between the interviewee's perception and the QualOSS assessment method we will use the neutral team member's colour assignment as a reference instead of the assignment the interviewee did
- we will keep track of differences between the human perception and the QualOSS assessment method as well as between the interviewee and the neutral team member. In order to become able to explain these possible deviations and particularly incoherence of the interviewee's assessment the interview will go on with a detailed discussion of each quality characteristic after the interviewee's first colour assignment. This provides us with the necessary amount and quality of information for the neutral team

member to make his guess, and at the same time it provides us with insights and background information that we need in order to understand why the interviewee is incoherent. For instance, an interviewee could describe how vivid and important the community for a F/OSS endeavor is but may give it a “yellow” in the assessment, instead of a green that a neutral observer might have given. It is the task of the interviewer to find out why there is this discrepancy, either during the interview or in an additional interview after the main interview and after the results of the QualOSS assessment are available (see below). For instance, it can be that the interviewee tends to neglect to some extent the importance of community measures because he is so much part of it that he takes a functioning community as granted. Or he might have made some community experience in the past that never occurred again but nevertheless biased his otherwise positive perception. Being able to explain these differences is crucial for assessing the authenticity of the QualOSS assessment. We assume that later, companies might find similar deviations of the QualOSS assessment results from what they had expected (in this case, the firm has to reconsider its own views too, and it has to find the explanations about why their expectations were not met). In order to gather the required background information, the discussion of the measures in detail will be open enough to going beyond the standard QualOSS assessment and touch issues that may not deal with the measurement directly, but have an indirect powerful influence on human perception. This provides also some insights about the directions in which an advanced QualOSS assessment might be developed.

11. Appendix C – Sample Questionnaire-like element to measure Human Perception

We would like to ask you to compare Asterisk version 1.4.17 and version 1.6.0.5. :

Imagine you were a company that wants to use this version for its products or services and tell us your personal opinion on how each of the software characteristics in the table on the next page must be considered with regard to its impact on robustness and evolvability.

Please use the following colour scheme:

green	no risk at all / the characteristic has a very positive impact
yellow	some risk / the characteristic has by and large a positive impact
red	quite a risk / the characteristic has by and large a negative impact
black	substantial risk / the characteristic has a very negative impact

Example:

If you think that there are significant security issues in one of the software versions you should assess this characteristic as "black"

If the software version is very secure in principle but bears some security issues if other factors, such as a specific platform or something similar, come into play, you should assess it as "red"

If there are more than just a few security issues and if it might be impossible or hard to avoid these problems you should assess the software version as "yellow"

If the software version is 100% secure (on all platforms etc.) you should assess it as "green".

Please try to assess all characteristics. Only if you really don't know how to assess a characteristic leave the respective row in the table below empty.

	ASESSMENT			
CHARACTERISTICS	No Risk	Some Risk	Quite a Risk	Substantial Risk
Work Products				
Code				
Maintainability				
Reliability				
Security				
Documentation				
Availability				
Structuredness				
Completeness				
Test				
Availability & Coverage				
Repeatability				
Community members				
Size & regeneration adequacy				
Interactivity & workload adequacy				
Composition adequacy				
Software Processes				
Requirements management capability				
Change management capability				
Release management capability				
Support management capability				
Community management capability				
Tools & Dependencies				
Run-time and compile time dependencies				
FIOSS endeavour compatibility				