



Sponsored through Framework Programme Sixth (Call 5) by



The QUALOSS Consortium consists of: CETIC (BE), Facultés Universitaires 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)

Document Information

Version: 1.0
Date : Jan 13, 10
Pages : 36

Owning Partner:
 URJC

Author(s):
 Daniel Izquierdo
 Jean-Christophe Deprez

Reviewer(s):
 Jacques Flamand
 Santiago Dueñas

To:
 European Commission

Purpose of distribution:
 E.C. Review

Status:

- Draft
- To be reviewed
- Proposal
- Final/Released

Confidentiality:

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

Deliverable ID:

D3.2

Title:

Populated QUALOSS repository (report and electronic database)

Verification of the Standard QualOSS Assessment Method

A deliverable of Task 3.2



Populated QUALOSS repository (report and electronic database)

Deliverable ID: D3.2

Page : 2 of 36

Version: 1.0

Date: Jan 13, 10

Status : Final

Confid : Public

Deliverable: D3.2

Title: Populated QUALOSS repository (report and electronic database)

Executive Summary:

This deliverable aims to summarize all the F/OSS endeavors assessed using the Standard QualOSS Assessment Method (Std QAM) produced by WP4. The Std QAM has evolved through 3 versions, v1.0_RC, v1.0, and the current, v1.1. Fourteen F/OSS endeavors were assessed with the complete Std QAM and many other assessments were only partial. A complete assessment of a F/OSS endeavor means that all the quality attributes of quality model of the Std QAM were assessed where a partial assessment only covers certain quality attributes. The following summarizes the verification process performed on the Std QAM.

Four Assessments using Std QAM v1.0_RC:

- 2 assessments produced preliminary tests, respectively assessing, the FindBugs and K3b F/OSS endeavors
- 2 assessments were initial applications on real world scenarios, respectively on the AdaCore/GCC back-end case study and OSL/Yanloc case study.

Based on the feedback from these first 4 assessments, a new version Std QAM v1.0 was produced.

Nine complete assessments were then performed using Std QAM v1.0:

- 6 assessments were used for benchmarking, that is, used as basis to verify and adapt the thresholds of the risk indicators of the Std QAM v1.0. The 6 F/OSS endeavors assessed for benchmarking were: Evolution, Evince, JMeter, JetSpeed, CVSanaly and Nautilus
- 2 assessments were used to test the Std QAM on various F/OSS endeavor scopes. Whereas the 6 benchmarking assessment considered F/OSS endeavor at the level of a whole F/OSS project, these two assessments scoped F/OSS endeavor on different scope, in particular, based on a particular version of interest and based on a set of F/OSS projects. The former assessment was done on Http v1.3 (and its sub-branches) and the latter assessment was done on Eclipse Platform, which consists of several sub-projects, which together create the Eclipse Platform.
- 1 assessment on a real world case study: the AdaCore/Coverage tool scenario

Finally, using the feedback of the assessments above on v1.0, a new version Std QAM v1.1 was produced. Only small modification were included in v1.1 to guarantee that its applicability would remain feasible. This new version was then used for one last assessment.

One complete assessment was performed using Std QAM v1.1

- 1 assessment on a real world case study, namely, the Freecode/Asterisk study

In addition to these complete assessments, the indicator of each quality attributes were benchmarked with additional F/OSS endeavors. However, this benchmarking effort was not coordinated to assess the same F/OSS endeavors. The most notable effort were undertaken on documentation availability where 17 additional assessments were performed and on community attributes, where the communities of several hundredth of F/OSS project were used to statistically calibrate the indicators. This last benchmarking was done as part of WP4 Task 4.3.

It is important to note, that not all these assessments were under the supervision of WP3, in particular, the assessment of Findbugs and K3b were supervised by WP4 Task 4.4. Furthermore, all the case studies, AdaCore/Gcc backend, OSL/Yanloc, AdaCore/Coverage tool and Freecode/Asterisk were supervised by WP5. Nonetheless all assessments results for Std QAM v1.0 and v1.1 have been gather in the QualOSS repository, viewable at: http://ingrid.cetic.be:33323/qualoss_assessment/. Moreover, the database dumps of the QualOSS repository were packaged together with this document to form the complete deliverable D3.2. This package is available at <http://alcachofe.libresoft.es/qualoss/qualossWP3.2.zip>

	Populated QUALOSS repository (report and electronic database)	Page : 3 of 36
		Version: 1.0
	Deliverable ID: D3.2	Date: Jan 13, 10
		Status : Final Confid : Public

TABLE OF CONTENTS

1. Introduction	4
1.1 Motivation.....	4
1.2 Goals and Outcomes of Task3.2.....	5
1.3 Structure.....	5
2. Methodology.....	6
2.1 Goals of Verification of Task 3.2.....	6
2.2 Dimensions Handled in each Phase.....	7
2.3 Scope of Task 3.2 and Deliverable D3.2.....	8
2.4 Assessment Process for the Task 3.2 Verification	8
2.4.1 Spreadsheets with Measures and Indicators.....	8
2.4.2 Log and Deviation Files.....	9
2.4.3 Debriefing meetings.....	9
2.4.4 Assessment Description Document	9
3. Verification of the Standard QualOSS Assessment Method.....	10
Appendix A: Files Attached with this Document.....	13
Appendix B: Summary of Results.....	14
3.1 Initial Assessment (K3b and Findbugs).....	14
3.2 Benchmarking (Evolution).....	15
3.3 Benchmarking (Eclipse Platform).....	20
3.4 Benchmarking (Evince).....	25
3.5 Benchmarking (Jmeter).....	29
3.6 Benchmarking (Nautilus).....	32
3.7 Benchmarking (CVSAnalY).....	33
3.8 Benchmarking (JetSpeed).....	34
3.9 Testing the Platform (HTTPD 1.3).....	35
3.10 More Examples.....	35

	Populated QUALOSS repository (report and electronic database) Deliverable ID: D3.2	Page : 4 of 36
		Version: 1.0
		Date: Jan 13, 10
		Status : Final Confid : Public

1. INTRODUCTION

This document is the continuation of the work done in QualOSS and more specifically in the work package 3. This work package (WP in the following) has three main goals:

- Identify the FLOSS projects to be measured.
- Deploy the necessary instrumentation in order to execute the measurements.
- Collect and validate data.

The WP3 is divided in three tasks, the first one: **Task 3.1: Measurement Targets Selection** where fifty projects were selected to be studied in **Task3.2: Measurement Tools Deployment and Data Collection** and the results of applying the Standard QualOSS Assessment Method will be validated in **Task3.3: Validation of Data Measurements of Reference advanced FLOSS projects** where feedback from FLOSS community members is collected.

The main outcomes of WP3 are five items: deliverables D3.1, D3.2, D3.3, electronic databases and the assessment reports with the QualOSS assessment results.

This deliverable aims to explain the process which took place during the assessment process, the main difficulties found and the methodology followed.

As a brief introduction, there were defined seven phases that will be later described:

- Initial assessment.
- Initial testing on real cases.
- Initial Benchmarking for the Standard QualOSS Assessment Method (Std QAM in the following) v1.0.
- Additional Benchmarking for Indicator Thresholds of the Std QAM v1.0.
- Testing the Std QAM v1.0 on various assessment dimensions.
- Second wave of Testing on Real Cases.
- Testing the user friendliness of the Std QAM.

Each of the phases contain a list of FLOSS endeavours which were analysed by the Standard QualOSS Assessment Method.

1.1 MOTIVATION

This deliverable focuses on testing the Standard QualOSS Assessment Method (QAM in the following) on a set of FLOSS endeavours with several scopes. Each scope is more detailed in the definition of FLOSS endeavour given in deliverable 4.1, but generally speaking, the FLOSS endeavour definition may refer to an **entire project**, a **part of a project** (for instance a library) and a **family of projects**. Also, the Std QAM aims to be used not just in those aforementioned scopes (focusing on just one project), but also in **product comparison** and **version comparison** (for a given product).

Task 3.2 used FLOSS projects listed in deliverable 3.1 as starting point. However, since QualOSS assessment took more manual effort than expected, it was necessary to reduce the number of assessments.

	Populated QUALOSS repository (report and electronic database) Deliverable ID: D3.2	Page : 5 of 36
		Version: 1.0
		Date: Jan 13, 10
		Status : Final Confid : Public

In the end, the important aspects sought by Task 3.2 and WP3 in general, are to provide an efficient strategy for testing the Std QAM and feel confidence that it provides trustworthy results and a broader applicability.

By definition, the Standard QualOSS Assessment Method (Std QAM) assess the risks related to the evolvability and the robustness of a FLOSS endeavour. In order to do so, it assesses risk associated to a FLOSS software product, a FLOSS community and the FLOSS software processes. The analysis of software products (maintainability, reliability and others) is a very well known field in the software engineering area where there are dozens of academic studies (please refer to deliverable 1.2 for more information about related research work). Thus, the most important aspect of the software product and software processes was to show its applicability to the FLOSS world. This was achieved selecting some specific projects, together with diverse FLOSS endeavours.

Regarding to the community aspects, they are much less known. Consequently, for fixing appropriate thresholds on measures to quantify risk, it was necessary to evaluate a larger set of FLOSS endeavours.

1.2 GOALS AND OUTCOMES OF TASK3.2

Goals:

- Deploy measurement tools to use during a QualOSS assessment
- Collect data from a reference set of FLOSS projects.

Outcomes:

- Valid data from reference set of FLOSS projects imported in the QualOSS repository (on CETIC cluster).
- Deliverable 3.2. It explains the strategy followed for testing the Standard QualOSS Assessment Method and present a summary of the test results.
- Electronic data for all QualOSS assessments
 - Electronic databases with results from some analysed projects. Those data bases were created by the QualOSS platform and their scheme is described by deliverable 2.4
 - Electronic reports describing the various assessment operations including deviations and debriefing
- Electronic database with results from community metrics.

1.3 STRUCTURE

The remainder of this deliverable is as follows: next section explains the methodology followed for testing the Standard QualOSS Assessment Method. Secondly the verification section where each type of verification presented in Section 2.2 is reviewed and the important results learned from the *test* assessments is highlighted. Finally the appendixes where detailed information is shown for each *test* assessment.

	Populated QUALOSS repository (report and electronic database) Deliverable ID: D3.2	Page : 6 of 36
		Version: 1.0
		Date: Jan 13, 10
		Status : Final Confid : Public

2. METHODOLOGY

This section describes the methodology followed by the QualOSS consortium for testing the Standard QualOSS Assessment Method (Std QAM) thoroughly. For each assessment, several files have been created in order to log the various actions taken on scoping, taking measures and indicators, recording deviations and suggesting improvement for future version of the Std QAM.

2.1 GOALS OF VERIFICATION OF TASK 3.2

Taking into account the initial target of the Std QAM, the Std QAM was built to specifically work on the case where an enterprise has to face with different types of FLOSS acquisition (some of them addressed in the motivation section). These ways to acquire FLOSS software are defined by various dimensions¹:

- FLOSS collaboration context.
- FLOSS Endeavour Scope.
- FLOSS Assessment Mode.
- FLOSS use.

In consequence, most of the testing effort in Task 3.2 must verify that the Std QAM works for the dimensions listed above. Also, each of the defined dimensions may have several “values”. A value means a specific situation which can be found in each dimension. In the next list, those **in bold** were tested by at least one QualOSS assessment:

- FLOSS collaboration context: **full FLOSS collaboration, FLOSS fork**, FLOSS Takeover, FLOSS exploit
 - Rationale: FLOSS collaboration from a business perspective. A company may be interested in directly working with the community achieving common goals, but also in forking the project for its own interests and other possibilities.
- FLOSS Endeavour Scope: **entire FLOSS project, a part of a FLOSS project, a set of FLOSS projects**
 - Rationale: The scope of the endeavour is defined by the part of the project selected. Since the Std QAM is expected to work on any situation, we can find the analysis of an entire FLOSS project, a part of a FLOSS project and a family of FLOSS projects.
- FLOSS Assessment Mode: **product comparison, version comparison, introspection**
 - Rationale: The Std QAM aims to also work on a product comparison or in a deeper analysis. Hence, there are some scenarios where a company may be interested in comparing some FLOSS endeavours, some version of the same FLOSS product or in studying deeper a own FLOSS endeavour.
- FLOSS use: **integration in a product, integration in a service**, integration in an infrastructure
 - Integration of the FLOSS component in a software product of an organisation.

Clearly, not all possible combinations of the dimension value were explored. Below, the list of combination explored is detailed. Thus, the main goal of WP3 is to verify that the Std QAM is tested appropriately taking into account next steps:

¹ For more information, please refer to deliverable 4.1, where those dimensions were initially defined and deliverable 5.1 where those dimensions were refined.

	Populated QUALOSS repository (report and electronic database)	Page : 7 of 36
		Version: 1.0
	Deliverable ID: D3.2	Date: Jan 13, 10
		Status : Final Confid : Public

- Check the feasibility of the Std QAM (on lab cases and real cases).
- Calibrate indicator thresholds for assessment done for the specific cases for which the Std QAM was built (see above).
- Test to see if the Std QAM also works for other types of assessments subspace (for which it was not initially built).

2.2 DIMENSIONS HANDLED IN EACH PHASE

Task 3.2 was an iterative process where several phases were defined (Initial assessment, initial testing on real cases, initial benchmarking for v1.0, additional benchmarking for v1.0, testing the Std QAM on several dimensions, second wave of testing on real cases and user friendliness). For each phase, a set of dimensions were handled²:

1. Initial Assessment: This first step aims to test the initial Std QAM (v0.3) on the FLOSS project K3B and Findbugs. In both cases, the dimension values assumed are **{Full FLOSS collaboration, Entire FLOSS Project, Product Comparison, Integration in Product}**

2. Initial Testing on Real Cases: These assessments are run on real world cases in order to check the applicability of the Std QAM for enterprises and also try the initial Std QAM on various assessment dimension values. The two FLOSS endeavours assessed were: GCC backend, Yanolc. Each of them explored different dimension values: for GCC backend, the dimension values are **{Full FLOSS collaboration, Part of a FLOSS Project, Version Comparison, Integration in Product}**. For Yanolc, the dimension values are **{FLOSS fork, Part of a FLOSS Project, Product Comparison, Integration in Product}**

After the first two steps, an initial set of feedback was produced and help to create the Std QAM v1.0 used for the subsequent testing steps.

3. Initial Benchmarking for the Std QAM v1.0: The benchmarking step is used to polish the measurement procedures and the indicator thresholds defined in the Std QAM v1.0. The Benchmarking assumed the following dimension values: **{ Full FLOSS collaboration, Entire FLOSS Project, Product Comparison, Integration in Product }**. This benchmarking was done on 6 FLOSS projects: Evolution, Evince, JMeter, JetSpeed, CVSAAnaly, and Nautilus.

4. Additional Benchmarking for Indicator Thresholds of the Std QAM v1.0: In order to further calibrate thresholds of certain indicators, additional assessments were performed. This calibration was performed differently for the different quality characteristics assessed in the Std QAM v1.0. For characteristics of product and software processes, the strategy was to perform a few additional assessments to show that the complete range of colours of the indicators are achievable. For community characteristics, the strategy was different. Threshold for the community indicators are newer than for product and software processes. Thus, there is not much research to help to set initial realistic thresholds for most community indicators. It was then decided to perform the calibration of community indicators on more than 1400 FLOSS community datasets made available by the FLOSSMETRICS project. For all additional benchmarking assessments, the dimension values assumed are **{ Full FLOSS collaboration, Entire FLOSS Project, Product Comparison, Integration in Product }**

5. Testing the Std QAM v1.0 on various assessment dimensions: These assessments test if the Std QAM v1.0 is applicable on other assessment dimension. In particular, the assessment of Eclipse Platform (a umbrella project of many sub-projects) verify the applicability of the Std QAM v1.0 for the dimension values: **{ Full FLOSS collaboration, set of FLOSS Projects, Product Comparison, Integration in Product }** and

² In order to avoid too much information in this section, the explanation of the main results and why those projects were selected is shown in next sections.

	Populated QUALOSS repository (report and electronic database) Deliverable ID: D3.2	Page : 8 of 36
		Version: 1.0
		Date: Jan 13, 10
		Status : Final Confid : Public

the assessment of Apache Http 1.3 tests the applicability of the Std QAM v1.0 for the dimension values { **Full FLOSS collaboration, entire FLOSS Project, Version Comparison, Integration in Product** }.

6.Second wave of Testing on Real Cases: 2 additional assessments are performed, the first one is a FLOSS endeavour undertaken by AdaCore and several other partners that develop a software coverage tool and the second is an assessment of the Asterisk endeavour for the Freecode company. The dimension values for the coverage tools are: { **Full FLOSS collaboration, entire FLOSS Project, Introspection, Integration in Product** } and for Asterisk, they are: { **Full FLOSS collaboration, entire FLOSS Project, Product Comparison, Integration in Service** }

7.User Friendliness of the Std QAM: This type of testing took place during several of the 6 testing steps above. It consists of verifying the clarity and non-ambiguity of the documentation of the Std QAM. In order to perform this verification, QualOSS assessment are performed by people who did not create the Std QAM and its measurement procedures. This type of testing was first performed during step 2, a people from AdaCore and OSL (involved in QualOSS but neither in defining measures nor the assessment process of the Std QAM) performed the assessment. Their feedback was used to create the Std QAM v1.0. A second friendliness test was then performed in an assessment of step 4 (the JMeter assessment) where a person (from the consortium) had to apply measurement procedure that he did not define. A third friendliness test also took place during step 6 in the assessments of the coverage tool and of Asterisk. .

2.3 SCOPE OF TASK 3.2 AND DELIVERABLE D3.2

The testing steps described in Section 2.2 were not all under the supervision of Task 3.2. In particular, step 1 was performed as part of Tasks 4.2, Steps 2 and 6 were supervised by WP5. As for the other steps 3, 4, 5, and 7, they were supervised by Task 3.2 and they reported their results to WP4, in particular, Task 4.4 so it could use the information generated by Task 3.2 to describe the internal validation process and make recommendations for producing the Std QAM v1.1.

Although all of these steps were not controlled by Task 3.2, they were still part of overall testing of the Std QAM and thus, needed to be presented in this deliverable. However, the specific outcomes of steps 1 and 2 are presented in deliverable D4.4. The feedback from step 6 will only be gathered at the very end of the QualOSS project and will be described in deliverable D5.2. In consequence, information about step 1, 2, and 6 are not covered in any more depth in this deliverable.

2.4 ASSESSMENT PROCESS FOR THE TASK 3.2 VERIFICATION

All tests mentioned in Section 2.1 followed the assessment process of the Std QAM described in deliverable D4.1. This process did not change between v0.3 and v1.0 of the Std QAM. Important aspects of this assessment process and how it was executed during the assessment of Task 3.2 are briefly describe in the subsections below. For details, each assessment has a corresponding file that describes the actions taken during the assessment. In particular, every operation of the workflow prescribed by the assessment process is presented and the person(s) who performed it are mentioned.

2.4.1 Spreadsheets with Measures and Indicators

All the versions of the Std QAM propose a set of spreadsheet documents. These spreadsheet documents specify how to perform the Std QAM, in particular, how to perform measurement and how to compute risk indicator for each characteristic of the quality model of the Std QAM. Each spreadsheet document addresses different characteristics of the quality model used in the Std QAM. In particular, there is one spreadsheet for the assessment of product maintainability, one for the assessment of product reliability, one for security, one for documentation availability and completeness, one for test availability and repeatability, one for all community characteristics, and one for all software process characteristics. In total, there are 7 spreadsheet documents

	<p>Populated QUALOSS repository (report and electronic database)</p> <p>Deliverable ID: D3.2</p>	Page : 9 of 36
		Version: 1.0
		Date: Jan 13, 10
		Status : Final Confid : Public

Assessment performed in Task 3.2 was performed collectively by several people. However, the responsibilities were well divided as one person was responsible for performing the assessment part of a single spreadsheet document.

2.4.2 Log and Deviation Files

Log files are necessary to log every step made during the assessment process. Each partner should create one, in order to keep a trace of the important information capture during assessments. This step is also interesting for those who want to verify the value of indicators or measures as it may be the case in Task 3.3 where community members will be asked to review the assessment results.

The deviation file may be created to record deviation to the process however, for Task 3.2, assessors were ask to follow rigorously the assessment process and the measurement procedures. Only very small deviations were acceptable. In consequence, few deviation files were created for the assessment of Task 3.2. When created, a deviation file contain description of the particular situation that required deviating from the assessment or measurement process.

2.4.3 Debriefing meetings

The debriefing meetings are done after a whole assessment process is finished. The leader of an assessment asks other partners to comment on the main issues that happened during the assessment process . The result of debriefing is collected and appears as an annex of the Assessment Description Document. An important aspect of debriefing is to collect suggestions for improvements so new version of the Std QAM can be improved in the future.

2.4.4 Assessment Description Document

For each analysed project, at least a set of spreadsheets is provided which contains the results of the assessment. Besides, it is provided a document which contains all the information about each step done during the assessment project for each partner, a general story about it and conclusions from the debriefing.

Finally, extra files are given to provide a general overview of the assessment process. For instance, debriefing documents, log documents and so on. More information about the directory structure for those files can be found in appendix A.

	Populated QUALOSS repository (report and electronic database) Deliverable ID: D3.2	Page : 10 of 36
		Version: 1.0
		Date: Jan 13, 10
		Status : Final Confid : Public

3. VERIFICATION OF THE STANDARD QUALOSS ASSESSMENT METHOD

In this section, we review each type of verification presented in Section 2.2 and highlight the important results learned from the *test* assessments. The detailed results of each assessment and their conclusion, are given in Appendix B of this document and the assessment reports are delivered in separate files.

Verification I: Initial test of the "v1.0 release candidate" of the Standard QualOSS Assessment Method on K3b & Findbugs

The assessment of K3b & Findbugs showed that most measurement procedures proposed in the "v1.0 release candidate" of the Standard QualOSS Assessment Method were feasible. Certain measures could not be taken due to lack of data but the measures still seemed relevant for the purpose of the assessment (risk of collaborating with a FLOSS endeavour). The main concerns were the indicator thresholds. They were immature and needed calibration and more testing on various type of FLOSS endeavour scopes (not just complete project). D4.4 provides the details on these assessments, which were performed under the supervision of WP4 (and not WP3).

Verification II: Real Test - Wave 1: Initial test of the "v1.0 release candidate" of the Standard QualOSS Assessment Method on Gcc-Backend and Yanolc.

The assessment Gcc-Backend and Yanolc were conducted in parallel to those of K3b and Findbugs. Gcc-Backend and Yanolc were also assessed using "v1.0 release candidate" of the Standard QualOSS Assessment Method. These assessments highlighted that some tools were not available for measuring C coupling. The v1.0 release candidate seemed to be able to take measurement on partial FLOSS projects. In both cases, Gcc-backend and Yanolc client were only part of the FLOSS project. The main feedback was the lack of documentation for some measurement procedures, the lack of automation for computing indicators, which should be done directly by the measurement spreadsheets. The lack of standardization between the measurement spreadsheets made is also hard to jump from one part of an assessment to the next.

Based on Verification I and II, a feedback report was generated by Task 4.4 and was then feedback to Task 4.2 so the various developers of the Standard QualOSS Assessment Method could make small adaptation to v1.0 release candidate and create v1.0 (for internal release only). Release 1.0 was done by Task 4.2 and was included in deliverable D4.2.

Verification III - Initial benchmarking of the Std QAM v1.0.

The assessment of 6 FLOSS projects was undertaken. They were useful for calibrating the current Indicator thresholds. However the number of assessments were too few for the assessment part with less background on which to base their calibration, notably, for the assessment of Community and of Documentation. In consequence, it was recommended that these two assessment parts performed additional calibration effort.

Furthermore, this benchmarking exercise showed that measure values should be accompanied by a measure status. This would help to indicate whether a measure value could not be taken due to some missing data (FLOSS endeavour's responsibility) or due to a problem in measurement procedure or measurement tools. Subsequently, indicators can use that information to compute a measure coverage. In addition to measure coverage, the benchmarking revealed a need to weigh indicator based on their information value. In consequence, it was recommended for later version of the Std QAM (eg for v1.1) to

	Populated QUALOSS repository (report and electronic database) Deliverable ID: D3.2	Page : 11 of 36
		Version: 1.0
		Date: Jan 13, 10
		Status : Final Confid : Public

create a field “confidence” for each indicator in order to see how much confidence one should put in the indicator value's capability to answer its assigned questions and goals.

These results are provided in electronic database. Firstly the QualOSS platform database: “dump_qualoss_assessment1_0” and secondly, the spreadsheets and other files related to these analysis.

Verification IV - Additional Benchmarking .

Additional measures and indicators have been evaluated separately for the different characteristics for further calibration. For documentation, measures and indicators were further calibrated on a set of 17 FLOSS projects. This enables adjusting the weight to assign to each type of documents. For community a significant statistical calibration took place (around 1400 projects initially took part of the calibration experiment) as part of D4.3, which then produced new threshold values for many community-related risk indicators.

This information can be found in electronic format. In first place, the extra analysis for documentation can be found in the directory “other-analysis/documentation”. The hundreds of projects studied for acquiring expertise in the analysis of communities, can be found in the file “[community_metrics_dumps.zip](#)”.

Verification V - Testing the Std QAM v1.0 various endeavour scope.

Although the assessment of Gcc-backend and Yanolc could be performed on FLOSS endeavour that were only part of project, it was still important to test whether the assessment method could be done on set of FLOSS project and also on a part of a FLOSS project delimited by version number. These respective cases were covered by assessing Eclipse Platform, which is truly an umbrella project for several separate FLOSS projects, and by assessing Http 1.3 where data for the other version needed to be ignored.

The outcome of this test showed that the scoping operation of an assessment was delicate and needed synchronization between assessment parts in particular, if each part is performed by a different assessor. Indeed, part of the Eclipse assessment needed to be redone in order to synchronize the datasets used to measure the product and the community. Surprisingly, most measures and indicator applied to these various context. In other words, measures and indicators could be computed and returned quite coherent results in comparison to the benchmarking assessment.

Verification VI: Second Wave of Testing on Real Cases.

Test of the Std QAM v1.0 took place on the AdaCore coverage tool. The same assessor as for Gcc-Backend performed the assessment. The conclusion were fairly good. All the problem identified on the Std QAM were solved. Additionally, although time was restricted, it was also important to test the new upcoming v1.1 and make sure that the visualization tool also worked with the new version. This test took place when assessing Asterisk in the context of the Freecode cases study. Overall, the assessment performed smoothly. The main remark consisted in the update of community indicators, in pointing out some missing documentation. Furthermore, the spreadsheet of the assessment of documentation did not automate the computation of Measure coverage using measure status, and of indicator confidence using measure coverage plus the calibrated indicator weight.

Verification VII: User friendliness - Assessment of JMeter (and also Gcc Backend, Yanolc)

	<p>Populated QUALOSS repository (report and electronic database)</p> <p>Deliverable ID: D3.2</p>	<p>Page : 12 of 36</p> <hr/> <p>Version: 1.0</p> <p>Date: Jan 13, 10</p> <hr/> <p>Status : Final</p> <p>Confid : Public</p>
---	--	---

User friendliness verification attempted to see if the Std QAM could be performed by people who did not create the method. The initial part of this was performed by assessing Gcc Backend, Yanolc using v1.0 release candidate of the Std QAM. There was clearly a lack of documentation, standardization between the various part of the assessment, and even lack of automation in the computation of Indicator. Developers of the Std QAM addressed these problems and created Std QAM v1.0. Afterwards, another The assessment another friendliness verification took place using the Std QAM v1.0 for assessing JMeter. The standardization of the measurement spreadsheets and the automation of the Indicators values were well automated.

However, the documentation was still lacking and created ambiguity hence subjectivity in the assessment. It was therefore recommended that v1.1 propose a standard documentation for each part of the assessment which could be further extended to provide a full documentation for each part of the assessment in the Std QAM.

	Populated QUALOSS repository (report and electronic database) Deliverable ID: D3.2	Page : 13 of 36
		Version: 1.0
		Date: Jan 13, 10
		Status : Final Confid : Public

APPENDIX A: FILES ATTACHED WITH THIS DOCUMENT

As aforementioned, during each assessment, a set of files have been created to follow the analysis of each FLOSS endeavour. Those files are delivered as a ZIP file "[qualossWP3.2.zip](#) "

The directories' hierarchy is as follows:

Project

Assessment_Results: It contains all the spreadsheets organized by version and quality attribute.

DataSources: It contains the files with the selection of the data sources. Most of this discussion took place in the internal mailing lists, what explains why few files are found here.

Debriefing: This directory contains the debriefing summary.

Deviations: This directory contains the deviations files if happened.

HowTo: This directory contains a description of how to run the scripts files. In most of the cases there is just few files. However the main information is found in the Jmeter project, where the roles were changed and documentation was created.

Logs: The log files.

Additionally the following database dumps are delivered :

[dump_qualoss_assessment1_0](#) : dump of the QualOSS platform data base for assessments version 1.0

[dump_qualoss_assessment1_1](#) : dump of the QualOSS platform data base for assessments version 1.1

[community_metrics_dumps.zip](#) : it contains the community metrics for around 1400 projects.

APPENDIX B: SUMMARY OF RESULTS

There are several kind of results. Those provided in electronic format and those provided in spreadsheets. We think that it is important to have both results in order to show how the process is carried out.

With respect to the electronic database, there are two kinds of database, the formal QualOSS database with full assessment and another electronic database with just community metrics. The first approach is the one with final results from the QualOSS consortium, while the second one is also attached as a way to reuse and show the results about the study made to update the thresholds in the community quality attribute.

As a summary of the whole set of results, next there is a list of projects and their results, also provided in electronic databases hosted at CETIC servers. The results of assessments appear in the different visualizations that have been iteratively used during the assessment process.

3.1 INITIAL ASSESSMENT (K3B AND FINDBUGS)

Results

Next figure shows the results in the old visualization model. It is necessary to remark that in the following versions it has changed to a 1-4 score, but using the same colors approach. This figure shows the assessment results obtained for k3b and Findbugs.

Assessment View (QualOSS)

	1. findbugs	2. k3b
1. Robustness/Evolvability	2	-30
1.1. Community Members	53	15
1.1.1. Interactivity and Workload Adequacy	47	-20
1.1.2. Size and Regeneration Adequacy	58	50
1.2. Software Processes	-46	-78
1.2.1. Capability of Release Management	-56	-67
1.2.1.1. Configuration management	-56	-67
1.2.1.1.1. Release Management	-11	-33
1.2.1.1.2. Release Planning	-100	-100
1.2.2. Capability of Requirements and Change Management	-36	-89
1.2.2.1. Change management	10	-100
1.2.2.1.1. Change Review	0	-100
1.2.2.1.2. Change Submission	100	-100
1.2.2.1.3. Commit Review	-34	-100
1.2.2.1.4. Committer Promotion	-100	-100
1.2.2.2. Requirements management	-17	-67
1.2.2.2.1. Enhancement Proposal	-100	-100
1.2.2.2.2. Issue Management	66	-34
1.2.2.3. Verification	-100	-100
1.2.2.3.1. Testing	-100	-100
1.3. Work Products	-0	-28
1.3.1. Documentation	0	0
1.3.1.1. Availability	0	0
1.3.1.1.1. Documentation information availability	-33	-33
1.3.1.1.2. Documentation type availability	33	33
1.3.2. Product	-10	15
1.3.2.1. Maintainability	50	36
1.3.2.2. Reliability	-69	-7
1.3.2.2.1. Importance_of_corrections	-100	33
1.3.2.2.2. Stability_Evolution	-7	-53
1.3.2.2.3. coding_convention_violation	-100	0
1.3.3. Test	8	-100
1.3.3.1. Availability & Coverage	-83	-100
1.3.3.2. Repeatability	100	-100

Difficulties found

	<p>Populated QUALOSS repository (report and electronic database)</p> <p>Deliverable ID: D3.2</p>	Page : 15 of 36
		Version: 1.0
		Date: Jan 13, 10
		Status : Final Confid : Public

Both projects were the first one to be assessed by the QualOSS method what showed a lack of several points to be improved in next steps:

- Scoping: In some cases (specifically for k3b) there were found several data sources for the same quality attribute what complicated the assessment process. For instance, the bug tracking system (BTS) found in k3b showed a lack of information when analysed. It was fixed when a new data source for the BTS was chosen to analyse the project. This took place since k3b has used several forges and different web sites to announce their work. For instance, at the very beginning the chosen BTS was the one hosted at www.sourceforge.net, however it was realised that this project really uses the Bugzilla provided by the KDE project.
- On the contrary, for Findbugs, all of its infrastructure was found at SourceForge what facilitated the assessment process.
- Assessment procedure: When analysing the data sources, some of the tools were not mature enough, what showed a delay in the assessment process. Specifically, the community tools were continuing improving what provoked several database schema changes.
- Spreadsheets: The spreadsheets initially used in these assessments showed a lack of information for some situations. The experience retrieved here was used in the following to improve the spreadsheets used for the version 1.0.
- Glossary: Some specific terminology was used during these assessments what mean that it was necessary to create a common glossary among the partners in order to use the same approach (or at least as similar as possible) in following assessments.

Conclusions

Taking into account the difficulties found during both assessments, several pieces of advice were retrieved for future assessments. It is worth enough to mention that all of this would be used during the creation of the v1.0 assessment method.

3.2 BENCHMARKING (EVOLUTION)

Results



Table . Results of the Evolution project

Robustness and Evolvability of the Evolution project	Work product	Product	Maintainability	Percentage of accepted enhancement proposals
				Rapidity of implementation of enhancement proposals
				Evolution of change in code between major releases
				Evolution of change to public interfaces between major releases
				Evolution of number of lines of code between successive releases
				Percentage of unassigned issues
				Rapidity of issue resolution
				Evolution of change in code between minor releases
				Average number of patch per issue
				Evolution of change to public interfaces between minor releases
				Average efferent coupling per defined types
				Average efferent coupling per low level module
				Average cyclomatic complexity per defined routine
				Percentage of commented algorithm
				Evolution of cyclomatic complexity of defined routines between successive releases
				Reliability
			Stability_Evolution_Specific_Version	
			Importance_of_Corrections	
			Coding_convention_violation	
			Security	Global_track_record_of_NVD_Entries_over_time
				Global_track_record_of_High_severity_NVD_Entries_over_time
				Global_track_record_of_Medium_severity_NVD_Entries_over_time
				Predictability_of_yearly_trend_of_High_severity_NVD_Entries_over_major_releases
				NVD_Entry_Status_of_selected_release
				High_severity_NVD_Entry_Status_of_selected_release
				Track_record_of_NVD_Entries_for_selected_minor_over_time
				Track_record_of_High_severity_NVD_Entries_for_selected_minor_over_time
				Predictability_of_the_high_severity_security_issues_over_minor_release_track_record
				Predictability_of_security_issues_over_minor_release_track_record
				Predictability_of_yearly_trend_of_nvd_entries_over_major_releases
				Security_issues_in_code
				Track_record_of_high_severity_nvd_entries_for_selected_minor_over_correction_releases
			Track_record_of_nvd_entries_for_selected_minor_over_correction_releases	
Documentation	Availability	Document type availability		
		Document type availability (weighted)		
		Document information availability		
		Document information availability (weighted document types)		
		Document information availability (weighted organisation and completeness and document types)		



Populated QUALOSS repository (report and electronic database)

Deliverable ID: D3.2

Page : 17 of 36

Version: 1.0

Date: Jan 13, 10

Status : Final

Confid : Public

Document information availability (document types and weighted organisation and completeness)



Table (continued). *Results of the Evolution project*

Robustness and Evolvability of the Evolution project	Work product	Test	Test Availability and Coverage	Likelihood_Future_Test_Reports
				Alternative_likelihood_future_test_report_2
				Composite_Indicator_For_Project_Manager
				Test_Report_Availability_For_Desired_Release
				Environment_Test_Availability
				Unit_Test_Coverage_Adequacy
				System_Test_Coverage_Adequacy
				Composite_Indicator_For_Testers
				Unit_Test_Suite_Adequacy
				System_Test_Suite_Adequacy
				Alternate:Test_Suites_Availability
	Test Repeatability	Ease_Of_Testing_For_Testers		
	Community members	Size and Regeneration Adequacy	Indicator Evolution first bug submitted by registered people	
			Indicator Evolution first commit submitted by registered people	
			Indicator Evolution first no source code commit by registered people + evolution first post in mailing lists + evolution first no bug report submitted by registered people	
			Indicator Evolution new core contributors	
			Indicator Evolution core member leaving core team	
			Indicator Evolution balance core team	
			Indicator Average committers longevity	
			Indicator Evolution code contributors who submitted patches and changes	
			Indicator Total code contributors who submitted patches and changes	
			Interactivity and Workload Adequacy	Indicator Evolution number of events
				Indicator Evolution number of commits
		Indicator Percentage people working old releases		
		Indicator Territoriality		
		Indicator Lines per committer + bugs per committer + emails per committer		
		Indicator Lines per committer		
		Indicator Percentage number handled files		
		Composition Adequacy	Indicator Name companies reporting bugs	
			Indicator Name companies committing	
			Indicator Name companies with community leaders	
			Indicator Name companies committing in module	
			Indicator Roles filled by community members	
Indicator Are roles adequate				
Indicator Is code expert enough				

Table (continued). *Results of the Evolution project*

Robustness and Evolvability of the Evolution project	Software processes	Capability requirements of change and management	Change_Submission_Maturity
			Change_Review_Maturity
			Change_Review_Adequacy
			Committer_Promotion_Maturity
			Commit_Review_Maturity
			Commit_Review_Adequacy
			Enhancement_Proposal_Maturity
			Enhancement_Proposal_Adequacy
			Issue_Management_Maturity
			Issue_Management_Adequacy
			Testing_Process_Maturity
			Testing_Process_Adequacy
		Capability of release management	Release_Planning_Maturity
			Release_Planning_Adequacy
			Release_Management_Maturity
			Release_Management_Adequacy
			Release_Backport_Maturity

Difficulties found

Database import: Problems importing the spreadsheets to the QualOSS database.

Quality attributes missed: Some of the quality attributes were not assessed during the process. They have been ignored because of the delay. All of them which appear with white background are the non-analysed ones.

Delay in the assessment: This assessment took several weeks to be completed. However not all the partners were full-day working on this activity. Nevertheless, the delay of this first steps was strong.

Revision of metrics and indicators: Some indicators and metrics needed to be reviewed again for future versions in order to accomplish all the tasks this happened mostly for the Community and Security quality attributes.

Conclusions

A debriefing activity took place after analysing the whole project. This debriefing appeared to be the first of the benchmarking session which aimed to improve the QualOSS method.

It is worth to mention the fact that the benchmarking projects were selected because there were well known by the QualOSS consortium, so the QualOSS consortium expected some specific results for all of them. For more information, please refer to each of the final documents for each project.

Finally, it is necessary to mention the scoping process. In this case, some pieces of advice were addressed in order to improve the scoping process. However, some problems were found during the scoping of the Evolution project:

- Responsible person needs to create the history table including different versions of the project. The evaluators choose from this table the versions depending on assessment

	<p>Populated QUALOSS repository (report and electronic database)</p> <p>Deliverable ID: D3.2</p>	<p>Page : 20 of 36</p> <hr/> <p>Version: 1.0</p> <p>Date: Jan 13, 10</p> <hr/> <p>Status : Final</p> <p>Confid : Public</p>
---	--	---

information availability. The rule - to select as higher version as possible. The limitation - overall assessment still might have be done having different scope.

- For some quality characteristics (e.g., maintainability, reliability) the very precise scope can be defined and agreed between evaluators. For other quality characteristics (e.g., doc. availability, software processes) the broad scope can be defined. But this broad scope needs to be as close to the selected project version as possible. In both cases it very important to document the actual scoping information in the log file.
- Some metrics (e.g., maintainability) are taken on the project distribution, other (e.g., community) are taken on the version control system. This makes difficult to achieve the common scope. The priority must be given to the version control system. Limitation - the specific and precise URL must be provided.
- Review of the scoped information worked very well for documentation availability. However it was failure for community members metrics.

3.3 BENCHMARKING (ECLIPSE PLATFORM)

Results



Table . Results of the Eclipse Platform project

Robustness and Evolvability of the Evolution project	Work product		Maintainability	Percentage of accepted enhancement proposals
				Rapidity of implementation of enhancement proposals
				Evolution of change in code between major releases
				Evolution of change to public interfaces between major releases
				Evolution of number of lines of code between successive releases
				Percentage of unassigned issues
				Rapidity of issue resolution
				Evolution of change in code between minor releases
				Average number of patch per issue
				Evolution of change to public interfaces between minor releases
				Average efferent coupling per defined types
				Average efferent coupling per low level module
				Average cyclomatic complexity per defined routine
				Percentage of commented algorithm
				Evolution of cyclomatic complexity of defined routines between successive releases
	Product	Reliability	Stability_Evolution	
			Stability_Evolution_Specific_Version	
			Importance_of_Corrections	
		Coding_convention_violation		
		Security	Global_track_record_of_NVD_Entries_over_time	
			Global_track_record_of_High_severity_NVD_Entries_over_time	
			Global_track_record_of_Medium_severity_NVD_Entries_over_time	
			Predictability_of_yearly_trend_of_High_severity_NVD_Entries_over_major_releases	
			NVD_Entry_Status_of_selected_release	
			High_severity_NVD_Entry_Status_of_selected_release	
			Track_record_of_NVD_Entries_for_selected_minor_over_time	
			Track_record_of_High_severity_NVD_Entries_for_selected_minor_over_time	
	Predictability_of_the_high_severity_security_issues_over_minor_release_track_record			
	Predictability_of_security_issues_over_minor_release_track_record			
	Predictability_of_yearly_trend_of_nvd_entries_over_major_releases			
	Security_issues_in_code			
	Track_record_of_high_severity_nvd_entries_for_selected_minor_over_correction_releases			
	Track_record_of_nvd_entries_for_selected_minor_over_correction_releases			
Documentation	Availability		Document type availability	
		Document type availability (weighted)		
		Document information availability		
		Document information availability (weighted document types)		
		Document information availability (weighted organisation and completeness and document types)		

	<p>Populated QUALOSS repository (report and electronic database)</p> <p>Deliverable ID: D3.2</p>	<p>Page : 22 of 36</p> <hr/> <p>Version: 1.0</p> <p>Date: Jan 13, 10</p> <hr/> <p>Status : Final</p> <p>Confid : Public</p>
---	--	---

				<p>Document information availability (document types and weighted organisation and completeness)</p>
--	--	--	--	--



Table (continued). Results of the Eclipse Platform project

Robustness and Evolvability of the Evolution project	Work product	Test	Test Availability and Coverage	Likelihood_Future_Test_Reports		
				Alternative_likelihood_future_test_report_2		
				Composite_Indicator_For_Project_Manager		
				Test_Report_Availability_For_Desired_Release		
				Environment_Test_Availability		
				Unit_Test_Coverage_Adequacy		
				System_Test_Coverage_Adequacy		
				Composite_Indicator_For_Testers		
				Unit_Test_Suite_Adequacy		
				System_Test_Suite_Adequacy		
				Alternate:Test_Suites_Availability		
				Test Repeatability		
	Ease_Of_Testing_For_Testers					
	Community members	Size Regeneration Adequacy and	Indicator Evolution first bug submitted by registered people			
				Indicator Evolution first commit submitted by registered people		
				Indicator Evolution first no source code commit by registered people + evolution first post in mailing lists + evolution first no bug report submitted by registered people		
				Indicator Evolution new core contributors		
				Indicator Evolution core member leaving core team		
				Indicator Evolution balance core team		
				Indicator Average committers longevity		
				Indicator Evolution code contributors who submitted patches and changes		
				Indicator Total code contributors who submitted patches and changes		
				Interactivity and Workload Adequacy	Indicator Evolution number of events	
						Indicator Evolution number of commits
						Indicator Percentage people working old releases
		Indicator Territoriality				
		Indicator Lines per committer + bugs per committer + emails per committer				
		Indicator Lines per committer				
		Indicator Percentage number handled files				
		Composition Adequacy	Indicator Name companies reporting bugs			
				Indicator Name companies committing		
				Indicator Name companies with community leaders		
				Indicator Name companies committing in module		
				Indicator Roles filled by community members		
				Indicator Are roles adequate		
				Indicator Is code expert enough		

	Populated QUALOSS repository (report and electronic database) Deliverable ID: D3.2	Page : 24 of 36
		Version: 1.0
		Date: Jan 13, 10
		Status : Final Confid : Public

Robustness and Evolvability of the Evolution project	Software processes	Capability requirements of and change management	Change_Submission_Maturity
			Change_Review_Maturity
			Change_Review_Adequacy
			Committer_Promotion_Maturity
			Commit_Review_Maturity
			Commit_Review_Adequacy
			Enhancement_Proposal_Maturity
			Enhancement_Proposal_Adequacy
			Issue_Management_Maturity
			Issue_Management_Adequacy
			Testing_Process_Maturity
			Testing_Process_Adequacy
		Capability release management	Release_Planning_Maturity
			Release_Planning_Adequacy
			Release_Management_Maturity
			Release_Management_Adequacy
			Release_Backport_Maturity

Difficulties found

The main difficulty found during the Eclipse Platform assessment was the scoping. The Eclipse projects is huge and is divided in some subprojects. This is also the case of the Eclipse Platform which is again divided in more subprojects. As addressed during the Evolution's debriefing, main problems where related to the scoping of the assessment.

In first place, there are main differences between the release version and the CVS version. It is common to find extra material in the CVS version and in the case of the Eclipse Platform, we realized that there were extra directories in the CVS version. It implied an agreement among the different partners to work together on an specific set of directories (reliability, security, maintainability and community are some of these quality attributes). However that agreement was made after a first draft for the Eclipse project was done.

Finally, another issue was related to the data sources, where the bug tracking system was too big to be analysed in less than few weeks. It provoked the analysis of other data sources and not directly using the Bugzilla provided by the Eclipse project. It was necessary to create some minor scripts to translate other databases version to the current understandable by the QualOSS tools.

Again, this FLOSS endeavour took more time than expected. It worked in this way since some of the indicators had to be retrieved during the assessment process.

Conclusions

The QualOSS consortium strongly recommend to have a first scoping session among all the people involved in the assessment process. This will save time among all the partners and will contribute to have a better common idea about the rest of the quality attributes.

	<p>Populated QUALOSS repository (report and electronic database)</p> <p>Deliverable ID: D3.2</p>	<p>Page : 25 of 36</p> <hr/> <p>Version: 1.0</p> <p>Date: Jan 13, 10</p> <hr/> <p>Status : Final</p> <p>Confid : Public</p>
---	--	---

3.4 BENCHMARKING (EVINCE)

Results



Table. Results of the Evince project

Robustness and Evolvability of the Evince project	Work product	Product	Maintainability	Percentage of accepted enhancement proposals
				Rapidity of implementation of enhancement proposals
				Evolution of change in code between major releases
				Evolution of change to public interfaces between major releases
				Evolution of number of lines of code between successive releases
				Percentage of unassigned issues
				Rapidity of issue resolution
				Evolution of change in code between minor releases
				Average number of patch per issue
				Evolution of change to public interfaces between minor releases
				Average efferent coupling per defined types
				Average efferent coupling per low level module
				Average cyclomatic complexity per defined routine
				Percentage of commented algorithm
				Evolution of cyclomatic complexity of defined routines between successive releases
			Reliability	Stability_Evolution
				Stability_Evolution_Specific_Version
				Importance_of_Corrections
				Coding_convention_violation
			Security	Global_track_record_of_NVD_Entries_over_time
				Global_track_record_of_High_severity_NVD_Entries_over_time
				Global_track_record_of_Medium_severity_NVD_Entries_over_time
				Predictability_of_yearly_trend_of_High_severity_NVD_Entries_over_major_releases
				NVD_Entry_Status_of_selected_release
				High_severity_NVD_Entry_Status_of_selected_release
				Track_record_of_NVD_Entries_for_selected_minor_over_time
				Track_record_of_High_severity_NVD_Entries_for_selected_minor_over_time
				Predictability_of_the_high_severity_security_issues_over_minor_release_track_record
				Predictability_of_security_issues_over_minor_release_track_record
				Predictability_of_yearly_trend_of_nvd_entries_over_major_releases
				Security_issues_in_code
				Track_record_of_high_severity_nvd_entries_for_selected_minor_over_correction_releases
				Track_record_of_nvd_entries_for_selected_minor_over_correction_releases
Availability	Document type availability			
	Document type availability (weighted)			
	Document information availability			
	Document information availability (weighted document types)			
	Document information availability (weighted organisation and completeness and document types)			

	<p>Populated QUALOSS repository (report and electronic database)</p> <p>Deliverable ID: D3.2</p>	<p>Page : 27 of 36</p> <hr/> <p>Version: 1.0</p> <p>Date: Jan 13, 10</p> <hr/> <p>Status : Final</p> <p>Confid : Public</p>
---	--	---

				<p>Document information availability (document types and weighted organisation and completeness)</p>
--	--	--	--	--



Table (continued). Results of the Evince project

Robustness and Evolvability of the Evince project	Work product	Test	Test Availability and Coverage	Likelihood_Future_Test_Reports	
				Alternative_likelihood_future_test_report_2	
				Composite_Indicator_For_Project_Manager	
				Test_Report_Availability_For_Desired_Release	
				Environment_Test_Availability	
				Unit_Test_Coverage_Adequacy	
				System_Test_Coverage_Adequacy	
				Composite_Indicator_For_Testers	
				Unit_Test_Suite_Adequacy	
				System_Test_Suite_Adequacy	
	Alternate:Test_Suites_Availability				
	Community members	Size Regeneration Adequacy and	Interactivity and Workload Adequacy	Test Repeatability	Ease_Of_Testing_For_Testers
					Indicator Evolution first bug submitted by registered people
					Indicator Evolution first commit submitted by registered people
					Indicator Evolution first no source code commit by registered people + evolution first post in mailing lists + evolution first no bug report submitted by registered people
					Indicator Evolution new core contributors
					Indicator Evolution core member leaving core team
					Indicator Evolution balance core team
					Indicator Average committers longevity
					Indicator Evolution code contributors who submitted patches and changes
					Indicator Total code contributors who submitted patches and changes
	Composition Adequacy				Indicator Evolution number of events
					Indicator Evolution number of commits
					Indicator Percentage people working old releases
					Indicator Territoriality
					Indicator Lines per committer + bugs per committer + emails per committer
					Indicator Lines per committer
					Indicator Percentage number handled files
					Indicator Name companies reporting bugs
					Indicator Name companies committing
					Indicator Name companies with community leaders
					Indicator Name companies committing in module
					Indicator Roles filled by community members
					Indicator Are roles adequate
					Indicator Is code expert enough

	Populated QUALOSS repository (report and electronic database) Deliverable ID: D3.2	Page : 29 of 36
		Version: 1.0
		Date: Jan 13, 10
		Status : Final Confid : Public

Robustness and Evolvability of the Evince project	Software processes	Capability requirements of change management	Change_Submission_Maturity
			Change_Review_Maturity
			Change_Review_Adequacy
			Committer_Promotion_Maturity
			Commit_Review_Maturity
			Commit_Review_Adequacy
			Enhancement_Proposal_Maturity
			Enhancement_Proposal_Adequacy
			Issue_Management_Maturity
			Issue_Management_Adequacy
			Testing_Process_Maturity
			Testing_Process_Adequacy
			Capability release management
		Release_Planning_Adequacy	
		Release_Management_Maturity	
		Release_Management_Adequacy	
		Release_Backport_Maturity	

Difficulties found

Not much difficulties were found. Most of the partners got used to work with the QualOSS methodology and results were obtained in few days.

However it is necessary to mention that the uploading process was not still user friendly enough, what meant to spend some time polishing the spreadsheets in order to upload the content to the QualOSS database.

Conclusions

This FLOSS endeavour's analysis showed that most of the tools were ready for production. Also, the process to retrieve metrics and create spreadsheets.

3.5 BENCHMARKING (JMETER)

Results

Work Product – Code – Maintainability (2,5)

Percentage of accepted enhancement proposals (3.5)

Rapidity of implementation of enhancement proposals (3.5)

Evolution of change in code between major releases (0.5)

Evolution of change to public interfaces between major releases (0.5)

Evolution of number of lines of code between successive releases (0.5)

Percentage of unassigned issues (3.5)



Populated QUALOSS repository (report and electronic database)

Deliverable ID: D3.2

Page : 30 of 36

Version: 1.0

Date: Jan 13, 10

Status : Final

Confid : Public

Rapidity of issue resolution (3.5)

Evolution of change in code between minor releases (0.5)

Average number of patch per issue (3.5)

Evolution of change to public interfaces between minor releases (0.5)

Average efferent coupling of high level modules (2.5)

Average efferent coupling of low level modules (2.5)

Average cyclomatic complexity per defined routine (3.5)

Percentage of commented algorithm (1.5)

Evolution of cyclomatic complexity of defined routines between successive releases (3.5)

Work Product – Code – Reliability (1.75)

Stability_Evolution (1.5)

Stability_Evolution_Specific_Version (3.5)

Importance_of_Corrections (0.5)

coding_convention_violation (1.5)

Work Product – Code – Security (2.8)

Indicator:Global_track_record_of_NVD_Entries_over_time (3.5)

Indicator:Global_track_record_of_High_severity_NVD_Entries_over_time (1.5)

Indicator:Global_track_record_of_Medium_severity_NVD_Entries_over_time (3.5)

Indicator:Predictability_of_yearly_trend_of_NVD_Entries_over_major_releases (1.5)

Indicator:Predictability_of_yearly_trend_of_High_severity_NVD_Entries_over_major_releases (1.5)

Indicator:NVD_Entry_Status_of_selected_release (3.5)

Indicator:High_severity_NVD_Entry_Status_of_selected_release (3.5)

Indicator:Track_record_of_NVD_Entries_for_selected_minor_over_time (3.5)

Indicator:Track_record_of_High_severity_NVD_Entries_for_selected_minor_over_time (3.5)

Work Product – Documentation – Availability & Completeness (2.2)

Document type availability (3.5)

Document type availability (weighted) (3.5)

Document information availability (1.5)

Document information availability (weighted document types) (1.5)

Document information availability (weighted organisation and completeness and document types) (1.5)

Document information availability (document types and weighted organisation and completeness) (1.5)

Work Product – Test – Availability (0.5)

Likelihood_Future_Test_Reports (0.5)

	Populated QUALOSS repository (report and electronic database)	Page : 31 of 36
		Version: 1.0
	Deliverable ID: D3.2	Date: Jan 13, 10
		Status : Final
		Confid : Public

Test_Report_Availability_For_Desired_Release (0.5)

Environment_Test_Availability (0.5)

Unit_Test_Coverage_Adequacy (0.5)

System_Test_Coverage_Adequacy (0.5)

Unit_Test_Suite_Adequacy (0.5)

System_Test_Suite_Adequacy (0.5)

Work Product – Test – Repeatability (2.5)

Alternate:Test_Suites_Availability (2.5)

Ease_Of_Testing_For_Testers (2.5)

Community – Size & Regeneration (2.8)

Indicator Evolution first bug submitted by registered people (3.5)

Indicator Evolution first commit submitted by registered people (3.5)

Indicator Evolution first no source code commit by registered people + evolution first post in mailing lists + evolution first no bug report submitted by registered people (3.5)

Indicator Evolution new core contributors (1.5)

Indicator Evolution core member leaving core team (2.5)

Indicator Evolution balance core team (1.5)

Indicator Average committers longevity (3.5)

Indicator Evolution code contributors who submitted patches and changes (3.5)

Indicator Total code contributors who submitted patches and changes (2.5)

Community – Activity & Workload (2.9)

Indicator Evolution number of events (3.5)

Indicator Evolution number of commits (3.5)

Indicator Percentage people working old releases (1.5)

Indicator Territoriality (1.5)

Indicator Lines per committer + bugs per committer + emails per committer (3.5)

Indicator Lines per committer (3.5)

Indicator Percentage number handled files (3.5)

Software Processes – Change & Requirement Management (1.6)

Change_Submission_Maturity (3.5)

Change_Review_Maturity (2.5)

Change_Review_Adequacy (0.5)

Committer_Promotion_Maturity (0.5)

	Populated QUALOSS repository (report and electronic database) Deliverable ID: D3.2	Page : 32 of 36
		Version: 1.0
		Date: Jan 13, 10
		Status : Final Confid : Public

Commit_Review_Maturity (0.5)

Commit_Review_Adequacy (0.5)

Enhancement_Proposal_Maturity (1.5)

Enhancement_Proposal_Adequacy (2.5)

Issue_Management_Maturity (2.5)

Issue_Management_Adequacy (0.5)

Testing_Process_Maturity (1.5)

Testing_Process_Adequacy (2.5)

Software Processes – Release Management (0.7)

Release_Planning_Maturity (0.5)

Release_Planning_Adequacy (0.5)

Release_Management_Maturity (1.5)

Release_Management_Adequacy (0.5)

Release_Backport_Maturity (0.5)

Difficulties found

This assessment aimed to check how user friendly it was for not experienced people to gather metrics included in the QualOSS method . In order to achieve this, each partner changed its role by other. It means that people involved in the QualOSS' project had to calculate a new set of metrics. For instance, people in charge of the processes metrics had to retrieve the documentation ones.

Main difficulties appeared at the very beginning in most of the quality attributes due to the fact of a lack of documentation and scoping. The latter showed to be vital for all the partners and changing the point of view of each of them helped to better understand future difficulties that may be find for new assessors.

With respect to the documentation, each partner had to create, first of all, a log file, explaining detailed problems found during the assessment and also specific documentation explaining how to carry on the assessment for the given quality attribute.

Conclusions

The Jmeter assessment achieved to create or improve specific documentation for each quality attribute.

3.6 BENCHMARKING (NAUTILUS)

Results

Hight Risk **Medium Risk** **Small Risk** **No Risk**

Robustness and Evolvability of the Evolution project AVG 2.286	Work product AVG 2.017	Product AVG 2.258	Maintainability AVG 1.885
			Reliability AVG 2.5

	Populated QUALOSS repository (report and electronic database) Deliverable ID: D3.2	Page : 33 of 36
		Version: 1.0
		Date: Jan 13, 10
		Status : Final Confid : Public

			Security AVG 2.389
		Documentation AVG 2.167	Availability AVG 2.167
		Test AVG 1.625	Test Availability and Coverage AVG 0.75
			Test Repeatability AVG 2.5
	Community members AVG 2.556	Size and Regeneration Adequacy AVG 2.611	
		Interactivity and Workload Adequacy AVG 2.5	
		Composition Adequacy AVG 0	
	Software processes AVG 2.284	Capability of requirements and change management AVG 1.667	
		Capability of release management AVG 2.9	

Difficulties found

No main difficulties were found during the assessment process. It is worth to mention the scoping section, where at the very beginning was a misunderstanding with the data sources and the project itself. There exist another Nautilus in the SourceForge web site. However, after some interchanged emails the scope became correct.

Conclusions

The main conclusion for this project was that it was easy to carry on an assessment for a given project (after having a correct spreadsheet, a set of user friendly tools and a correct scope),.

3.7 BENCHMARKING (CVSANALY)

Results

High Risk Medium Risk Small Risk No Risk

Robustness and Evolvability of the Evolution project AVG 1.775	Work product AVG 1.462	Product AVG 1.929	Maintainability AVG 2.233	
			Reliability AVG 0.833	
			Security AVG 2.722	
		Documentation AVG 1.833	Availability AVG 1.833	
			Test AVG 0.625	Test Availability and Coverage AVG 0.75
				Test Repeatability AVG 0.5
	Community members AVG 2.754	Size and Regeneration Adequacy AVG 2.722		
		Interactivity and Workload Adequacy AVG 2.786		
		Composition Adequacy AVG 0		

	Populated QUALOSS repository (report and electronic database) Deliverable ID: D3.2	Page : 34 of 36
		Version: 1.0
		Date: Jan 13, 10
		Status : Final Confid : Public

	Software processes AVG 1.109	Capability of requirements and change management AVG 0.917
		Capability of release management AVG 1.3

Difficulties found

Not main difficulties were found for this assessment. However in the set of product metrics we found a lack of tools to analyse Python source code, (at least at the same level as others for C or Java).

Conclusions

Again, CVSanaly was, in terms of metrics and indicators, easy to measure and to upload results to the QualOSS database.

3.8 BENCHMARKING (JET SPEED)

Results

Hight Risk	Medium Risk	Small Risk	No Risk
------------	-------------	------------	---------

Robustness and Evolvability of the Evolution project AVG 1.937	Work product AVG 1.99	Product AVG 2.052	Maintainability AVG 2.433
			Reliability AVG 1
			Security AVG 2.722
		Documentation AVG 2.167	Availability AVG 2.167
		Test AVG 1.75	Test Availability and Coverage AVG 1
			Test Repeatability AVG 2.5
	Community members AVG 2.397		Size and Regeneration Adequacy AVG 2.722
			Interactivity and Workload Adequacy AVG 2.071
			Composition Adequacy AVG 0
	Software processes AVG 1.425		Capability of requirements and change management AVG 1.75
		Capability of release management AVG 1.1	

Difficulties found

Not main difficulties were found. Again, it is necessary to mention that the scope process was the most complicated part since there were two projects named as JetSpeed. JetSpeed 1 and JetSpeed 2 where the second one was the continuation of the first one (previously abandoned).

Conclusions

	Populated QUALOSS repository (report and electronic database) Deliverable ID: D3.2	Page : 35 of 36
		Version: 1.0
		Date: Jan 13, 10
		Status : Final Confid : Public

Generally speaking, this project was quickly analysed and results were rapidly obtained.

3.9 TESTING THE PLATFORM (HTTPD 1.3)

As aforementioned, this study aimed to check the user friendly and current situation of the QualOSS method for a specific version of an specific FLOSS endeavour. As seen before, the final projects showed a easy process to be created. Thus, in this section we tried to deepen a bit more and try to focus on real cases, such as the case of HTTPD 1.3. So, our question was, *is the QualOSS platform ready for specific version of specific FLOSS endeavours?*.

Results

Hight Risk
Medium Risk
Small Risk
No Risk

Robustness and Evolvability of the Evolution project AVG 2.435	Work product AVG 2.27	Product AVG 2.852	Maintainability AVG 2.7
			Reliability AVG 2.5
			Security AVG 3.357
		Documentation AVG 1.833	Availability AVG 1.833
	Test AVG 2.125	Test Availability and Coverage AVG 0.75	
		Test Repeatability AVG 3.5	
		Size and Regeneration Adequacy AVG 1.833	
	Community members AVG 2.31	Interactivity and Workload Adequacy AVG 2.786	
		Composition Adequacy AVG 0	
	Software processes AVG 2.725	Capability of requirements and change management AVG 2.75	
Capability of release management AVG 2.7			

Difficulties found

This assessment did not find any difficulty during its assessment. The main issue was related to the scoping process, where the specific version was needed for some of the quality attributes. However, this is necessary in all of the assessments done so far.

Conclusions

The QualOSS method worked with this new approach.

3.10 MORE EXAMPLES

Extra assessments were run for some of the metrics to guarantee that the range of colours were covered.

Specifically :

	<p>Populated QUALOSS repository (report and electronic database)</p> <p>Deliverable ID: D3.2</p>	<p>Page : 36 of 36</p> <hr/> <p>Version: 1.0</p> <p>Date: Jan 13, 10</p> <hr/> <p>Status : Final</p> <p>Confid : Public</p>
---	--	---

Documentation: 17 assessments

Process metrics: 2 assessments

Community metrics: 2 assessments plus around 1400 projects analysed for the indicator definition.

Reliability metrics: 2 assessments.