Sponsored through Framework Programme Sixth (Call 5) by

European Commission

Information Society Technologies

Document Information

Version: 1.1
Date : Jan 21, 10
Pages : 18

Owning Partner:
IESE

Author(s):
Martín SOTO (IESE)
Marcus CIOLKOWSKI (IESE)
Jean-Christophe DEPREZ (CETIC)
José Ruiz (ADACORE), Israel HERRAIZ (URJC), Carlos GARCIA CAMPOS (URJC), Raimundas MATULEVICIUS (FUNDP)

Reviewer(s):
CETIC, Adacore

To:
CONSORTIUM

Purpose of distribution:
for E.C. review

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)

Printed
on 12/11/06 at 03:07:02 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: D4.2

Title:

Metrics Systems and Calibrated Quality Models Report

(Metrics and Indicators of the Standard QualOSS Assessment Method)

Deliverable: D4.2

Title: Metrics Systems and Calibrated Quality Models Report

Executive Summary:

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 this end, QUALOSS plan on developing a tooled method for assessing the evolvability and robustness of Free/libre Open-Source Software (F/OSS). In turn, this will facilitate the integration and acquisition process of F/OSS in existing systems.

The fourth work package (WP4) is concerned with the creation of advanced quality models for F/OSS. For this purpose, Task 4.1 already defined the advanced QualOSS quality model. The purpose of Task 4.2 is to calibrate this model by testing it on actual F/OSS endeavors.

The document reviews the general strategy for WP4 as well as the results of Task 4.1. Based on this, the current QualOSS model is summarized, and a summary of the metrics and indicators present in it is presented. A number of appendixes present the detailed results of the metrics and indicators definition process, as well as the data resulting from assessing the two calibration projects FindBugs and K3b.

CHANGE LOG

Ver.	Date	Author	Description
0.1	2009/06/30	Martín SOTO	Initial draft
1.0	2009/17/17	Martín SOTO	Submitted version

APPLICABLE DOCUMENT LIST

Ref.	Title, author, source, date, status	Deliverable Identification

TABLE OF CONTENTS

1 Introduction

This document presents the results of QualOSS Task 4.2, *Calibration of Advanced Quality Models and User Manual Documentation*. The title originally intended for this document (as specified by the project's Description of Work) was “Metrics Systems and Calibrated Quality Models Report”. We have changed it to *Metrics Systems and Calibrated Quality Models Report* in order to make its purpose more clear. In particular, the Standard QualOSS assessment method already contains the advanced metrics produced in Task 4.2.

After this introduction, the first two sections of this document describe the general strategy for QualOSS Work Package 4, and briefly review the work done in Task 4.1, which constitutes the basis for the current task. Following these sections, Section 4 reviews the quality model of the Standard QualOSS Assessment Method, Section 5 presents general information about the set of QualOSS indicators and metrics (including a summary of number of metrics and estimated measurement effort), Section 6 describes the process used to calibrate the metrics, and Section 7 explains how the resulting metric and indicator definitions are laid out. The spreadsheets included with this document contain the full metric and indicator definitions (see below). Section 8 explains how QualOSS indicator values will be combined and visualized on the quality model tree using the visualization facilities of Fraunhofer IESE's Specula tool. Finally, Section 9 closes the document

with a short overview of the future work plan until November 2009. These sections are complemented by two appendices that describe the evaluation process for the FindBugs and K3b projects, which were used as the initial calibration pilots.

In addition to the data in this document, a ZIP file is also provided with the spreadsheets and other documents produced by the QualOSS partners in order to define metrics and indicators. The ZIP file contains a number of spreadsheets (in Open Document format) and some files (in subdirectories) containing additional explanatory material. The spreadsheets are the actual templates used to collect measure data. They contain the detailed procedures used to measure each one the QualOSS metrics, and so can be considered to be the source for the detailed definition of the QualOSS Standard Assessment. Together with Deliverable 4.1, which explains the assessment method at a higher level, they constitute the User's Manual for the Standard Assessment.

The spreadsheets contain also the definitions for the high level quality indicators. In most cases, the values for these indicators are calculated automatically as soon as the measurement data is entered for a particular project. A README file provided in the ZIP file together with the spreadsheets provides further details about their actual use.

2 Strategy for WP4 and Task 4.2

Task 4.1 developed the QualOSS methodology and elaborated the high level part of the standard QualOSS assessment method consisting of the quality model and the role-based questions needed in order to assess the various characteristics of the standard QualOSS model.

Task 4.2, in turn, finalizes the standard QualOSS assessment method. The important outcome of this task is a set of metrics and indicators useful to answer the role-based questions. Subsequently, Task 4.2 also defines how the indicator values of lower-level characteristics can be aggregated to form higher-level indicators to assess the high level of concepts of the quality model. Aggregation can be performed all the way to the root of the quality model to obtain a summary assessment of the robustness and evolvability of the F/OSS endeavor under analysis. The method used to define indicators and their aggregation is reviewed in Section 5. The resulting quality model is described in this document and in its associated tables.

The effort of defining metrics, measurement procedures, indicators and indicator thresholds is intensive and deserves a thorough validation phase. This is the responsibility of Task 4.4. Initially, Task 4.4 consists of a peer-review effort first for the metrics and measurement procedures, second for indicators and third, on actual test runs where metric and indicator values (and colors) are computed on two F/OSS endeavors, namely, the FindBugs and K3b projects.

Finally, Task 4.3 applies data mining and artificial intelligence techniques on F/OSS datasets with the hope of discovering interesting facts on the datasets that can then be used to create new indicators and define thresholds for the existing ones. This is a more exploratory task. Current QualOSS assessment results do not contain enough data points to be used as a basis for data mining. Consequently, FLOSSMETRICS datasets are being used, in particular, those generated by the CVSAnaly tool. These datasets contain records of each interaction of committers with a version

control system such as CVS, SVN or Git. The initial effort of Task 4.3 consists of performing some clustering analysis and also some neural network learning techniques. Clustering is helpful to identify elements in the data sets that share or do not share certain features. This information can then be used to partition the dataset prior to performing additional data mining algorithms on each subset. Neural networks are useful to identify features in a dataset that help predict other features. In the case of QualOSS, features are metrics and indicator values. If the prediction rate is adequate, those features (metrics) having influence on the result can be used to design new "predictive" indicators. For example, if the number of committers can be predicted based on other metrics, it may be possible to design a new indicator for assessing community size and regeneration based on the outcome of data mining.

3 Summary of Task 4.1

During traditional software acquisition, both the quality of the software component of interest and the seriousness and trustworthiness of the enterprise that produces it are considered. An enterprise usually shows its seriousness and trustworthiness by displaying its financial strength and a quality certification such as an ISO9001 certification or a CMMI maturity level.

In the F/OSS world, however, it is not always possible to show seriousness and trustworthiness in this way. Instead, Task 4.1 (in its deliverable D4.1) proposes an alternate fit to the F/OSS context that requires assessing the robustness and evolvability of a F/OSS endeavor. A F/OSS endeavor comprises four sets of entities: work products, community members, software processes and dependencies to other F/OSS endeavors. Therefore, an assessment of the robustness and evolvability of a F/OSS endeavor has to consider quality characteristics of these four sets. Task 4.1 also describes several F/OSS acquisition scenarios. Various dimensions of a F/OSS acquisition scenario may influence the assessment of each characteristic in the quality model. The dimensions are:

- **F/OSS collaboration context:** Full F/OSS collaboration, F/OSS fork, F/OSS exploit, or F/OSS takeover
- **F/OSS endeavor scope:** Whole F/OSS project, a part of a F/OSS project or a set of F/OSS projects.
- **F/OSS component usage:** Integration in a product, a service or an infrastructure.
- **F/OSS assessment mode:** Comparison of versions (of the same F/OSS endeavor), comparison of F/OSS endeavors, introspection.

Task 4.1 then presents the QualOSS assessment methodology, which specifies the requirements that an assessment method has to fulfill to be recognized as a QualOSS assessment method. In particular, a QualOSS assessment method has to respect a specific assessment process and has to record traces of assessments so that it is possible to explain all obtained results.

Next, Task 4.1 elaborates the standard QualOSS assessment method. It is designed to answer many questions shared by F/OSS acquisition scenarios across the dimensions mentioned above. It is built with the full F/OSS collaboration as a target because this is the most comprehensive

F/OSS collaboration context, requiring information on all aspects of a F/OSS endeavor; that is, on the work products, the community, the software processes and the dependencies. In consequence, it is expected to work well for the other collaboration contexts that have more limited needs. It is also built to address the various F/OSS endeavor scopes, component usages and assessment modes.

In addition, the questions to answer when applying the standard QualOSS assessment method are more targeted to the comparison of F/OSS endeavors (that is, with the F/OSS assessment mode equals to *comparison of F/OSS endeavor*), but we expect that it may be applicable to the other two assessment modes (applicability to these three assessment modes will be studied in WP5). Finally, it is also expected that the standard QualOSS assessment method applies to all F/OSS endeavor scopes, and F/OSS component usages.

As a result, the standard QualOSS assessment method proposes a quality model hierarchy of important characteristics to assess in order to determine the robustness and evolvability of a F/OSS endeavor. This standard quality model—the standard QualOSS model for short—is shown in Figure 1. To assess each leaf characteristic of the quality model, the standard QualOSS assessment method identified a series of role-based questions to guide the assessment. These questions are listed in deliverable D4.1 – Annex A. It is then the role of Task 4.2 to complete the standard QualOSS assessment method by identifying the adequate metrics and indicators to answer the questions and the characteristics of the quality model all the way up to the root, which represent the assessment of robustness and evolvability of the F/OSS endeavor in question.

The main purpose of Task 4.2 is to test the model defined in 4.1 by applying it in practice. This document presents the detailed results of this work.

4 General Structure of the Standard QualOSS Quality Model

There are four main characteristics (work products, community members, software processes, tools and dependencies), each with one to four sub-characteristics (Figure 1).

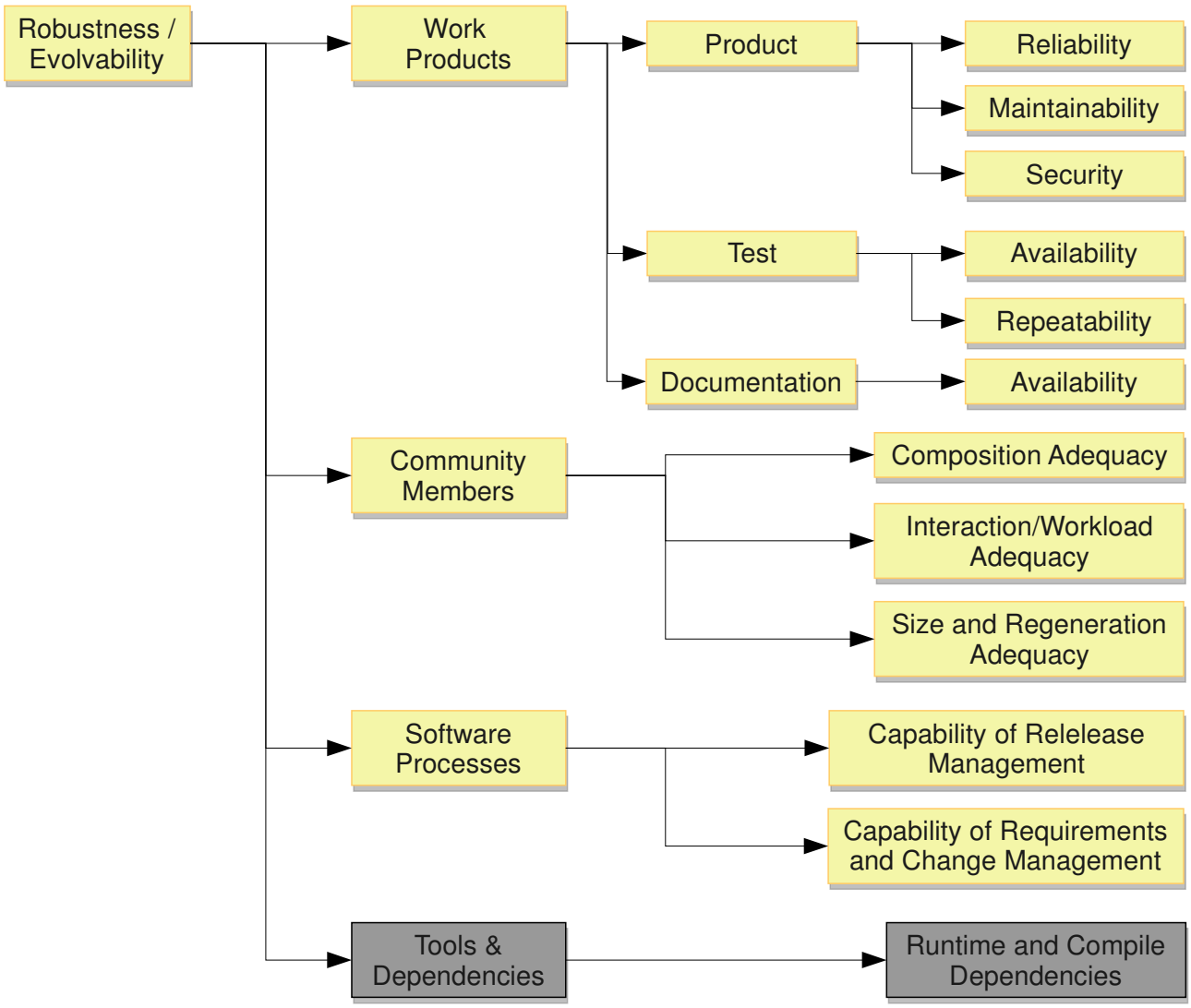


Figure 1: QualOSS model: Work product, Community members and Software processes. Gray boxes denote indicators that will be defined until the end of the project

The following table summarizes the number of metrics and indicators in these categories. For each group, the number of metrics and indicators that must be collected or calculated manually is provided separated from the number of metrics and indicators that can be obtained or calculated automatically. Notice that, in some cases, automatic metrics or indicators may require some manual work. The last columns shows the total number of hours necessary for collecting the data for a single F/OSS endeavor.

Area	Metrics		Indicators		Effort (hours)
	Manual	Automatic	Manual	Automatic	
Work Products / Product / Reliability		9		4	6
Work Products / Product / Maintainability	22	11		15	4
Work Products / Product / Security		38		9	1
Work Products / Test	40			9	1
Work Products / Documentation	41[1]			6	8
Community Members	16	7			8
Software Process	69			17	6
Total	147	65	0	60	34

[1]A total of 572 individual document checks will be performed in order to calculate these metrics.

Details on the definition of these sub-characteristics and their corresponding indicators and metrics can be found in the appendixes of this document as explained in Section 1.

5 QualOSS Quality Modeling Concepts: Metrics and Indicators

The quality model of the standard QualOSS assessment method is composed of three types of interrelated elements (see Figure 2): *quality characteristics, metrics (or metrics)*, and *indicators*. *Quality characteristics* correspond to the concrete attributes of a product or community considered relevant for assessing the robustness and evolvability of a F/OSS endeavor. *Metrics* correspond to concrete aspects that can be measured on various assets provided by a F/OSS endeavor expected to help in the assessment of work product, community, software processes, or dependencies characteristics. Finally, *indicators* define how to aggregate and evaluate the measurement values to obtain a consolidated information that can be readily used by decision makers when performing an assessment.

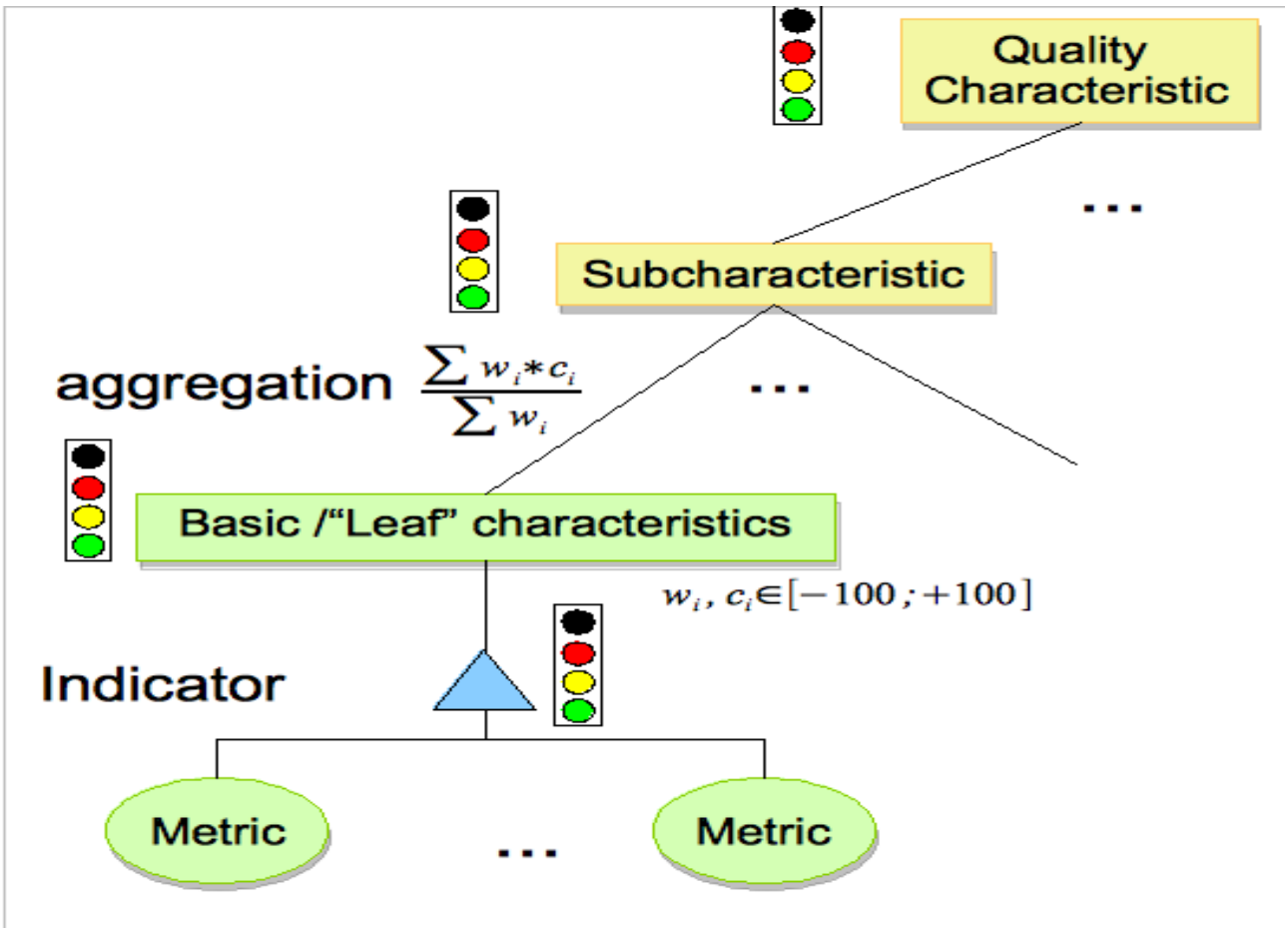


Figure 2: Characteristics, sub-characteristics and metrics in the standard QualOSS quality model

The quality characteristics in the model are organized in a tree hierarchy of several levels that are referred to as characteristics and sub-characteristics for simplicity. The sub-characteristics are considered to contribute in one way or another to the main characteristic they belong to.

One important role of indicators is to interpret their underlying metric values on the following scale:

- ***Green***: *No or minor risks for the measurement object in relation to a given characteristic*. Sizable existing work or measurement object of sufficient quality is present. Alternatively, the F/OSS endeavor exhibits a predictable or desirable behavior (e.g., less than 5% of existing work needs to raise its quality or, alternatively, evidence of unpredictable or undesirable behavior should be inferred for at most 5% of total assets studied).
- ***Yellow:*** *Significant risk for the measurement object in relation to a given characteristic*. Existing work/measurement object has flaws or exhibits some unpredictable or undesirable behavior. For a yellow label, less than 30% of the total existing work should require rework. Alternatively, evidence of unpredictable or undesirable behavior should be inferred for at most 30% of all assets studied.

- ***Red***: *Critical risks for the measurement object in relation to a given characteristic*. Existing work/measurement object has serious flaws (i.e., it cannot really be used; it is beyond "threshold of pain"). However, some of it can be saved with a still large amount of effort. Alternatively, the F/OSS endeavor exhibits a high level of unpredictability. For a red label, less than 70% of the total existing work should require rework. Alternatively, evidence of unpredictable or undesirable behavior should be inferred for at most 70% of all assets studied.
- ***Black***: *Prohibitive risks for the measurement object in relation to a given characteristic.* This is somewhat equivalent to “discard and start from scratch”. None or only a very small portion of the existing work/measurement object (if there is any) can be used. For a black label, only 5% of less of the total existing work would not require rework. Alternatively, evidence of unpredictable or undesirable behavior is rampant, only 5% of the behavior showed desirable level for the total assets studied.

This ordinal scale is mapped onto an interval ranging from -100 to +100 with even spacing of indicator levels. The purpose of this is to (a) facilitate aggregation, and (b) enhance the interpretation with quantitative information. The resulting mapping is the following:

- Green: +100; range (50; 100]
- Yellow: +33; range (0; 50]
- Red: -33; range (-50; 0]
- Black: -100; range [-100; -50]

For high-level Indicators (indicators of indicators), an aggregation function is used, where each indicator and each characteristic receives a weight w_i. Values are aggregated by computing a weighted mean of underlying indicators or (sub-) characteristics. As mentioned above, these interpretation values are normalized on a scale between -100 and +100.

6 Process for Metric and Indicator Definition

In order to define and calibrate our metrics and indicators we have applied the method shown in Figure 3. It consist of six steps:

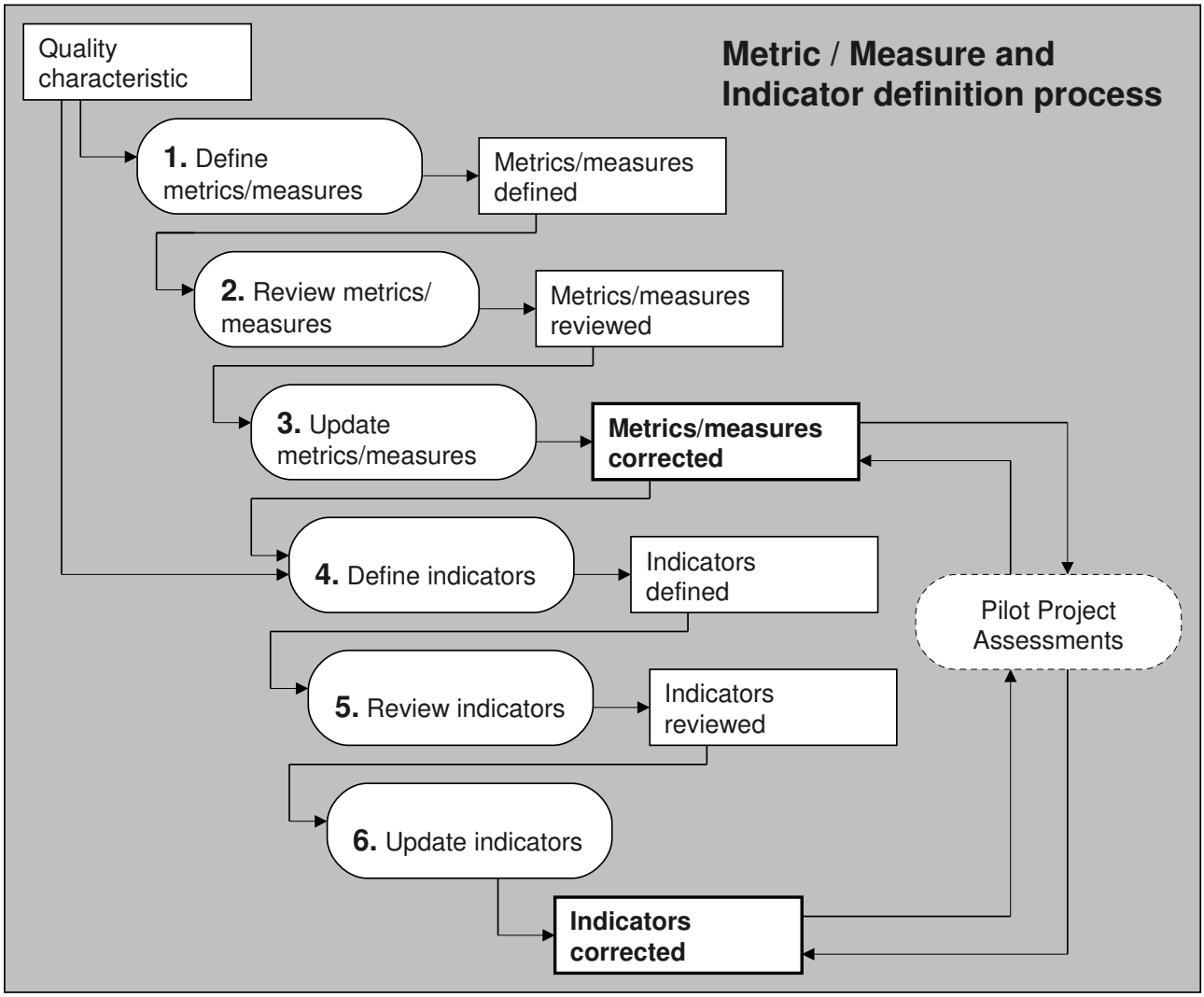


Figure 3: Process to define metrics and indicators

1. *Define metrics.* The input for the first step is the quality characteristics defined in the quality model. The step results in a set of defined metrics for each input characteristic.
2. *Review metrics.* The input for this activity is the report defining metrics. The responsible partner (different from the one who defined metrics) reviewed the input and gave comments to the defined metrics. The step results in the review report.
3. *Update metrics.* The input for this activity is the review report. The partner who initially defined the metrics, addresses the review comments. The step results in the definition of the metrics to each quality characteristic.
4. *Define indicators.* This activity has two inputs: firstly, it includes the characteristics of the quality model, secondly, it is the metrics for this quality characteristic. The step results in the definition of the indicators for the quality characteristic.

5. *Review indicators.* The input for this step are the indicators defined in the previous step. The responsible partner (different from the one who defined indicators) reviewed the input and gave comments about the indicators. The step results in the indicator review report.
6. *Update indicators.* The input of this step is the indicator review report. The partner who defined indicators addresses the comments from the review report. The step results in the indicator definition updated according to the review.

7 Structure of the Metric and Indicator Definitions

The metrics and indicators defined using the process specified above are described in detail in a set of OpenDocument spreadsheet documents (actually, workbooks containing several spreadsheets). Each document corresponds to a particular area of the model. Inside the document, at least two spreadsheets will be found, namely, one for metric definitions and one for indicator definitions. Additional spreadsheets may be present that are referenced by the main definitions for specific purposes.

Important fields in metric-definition spreadsheets are:

- *Name:* Name of the metric.
- *Scale:* Measurement scale.
- *Data source/artifact type:* Specification of the data repository(ies) or artifact(s) used as a source for the metric. Depending on the metric, only one of these two fields could be relevant.
- *Measurement procedure:* The actual procedure used to obtain the metric's value, i.e., running a particular tool or performing certain manual steps.

Important fields in indicator-definition spreadsheets are:

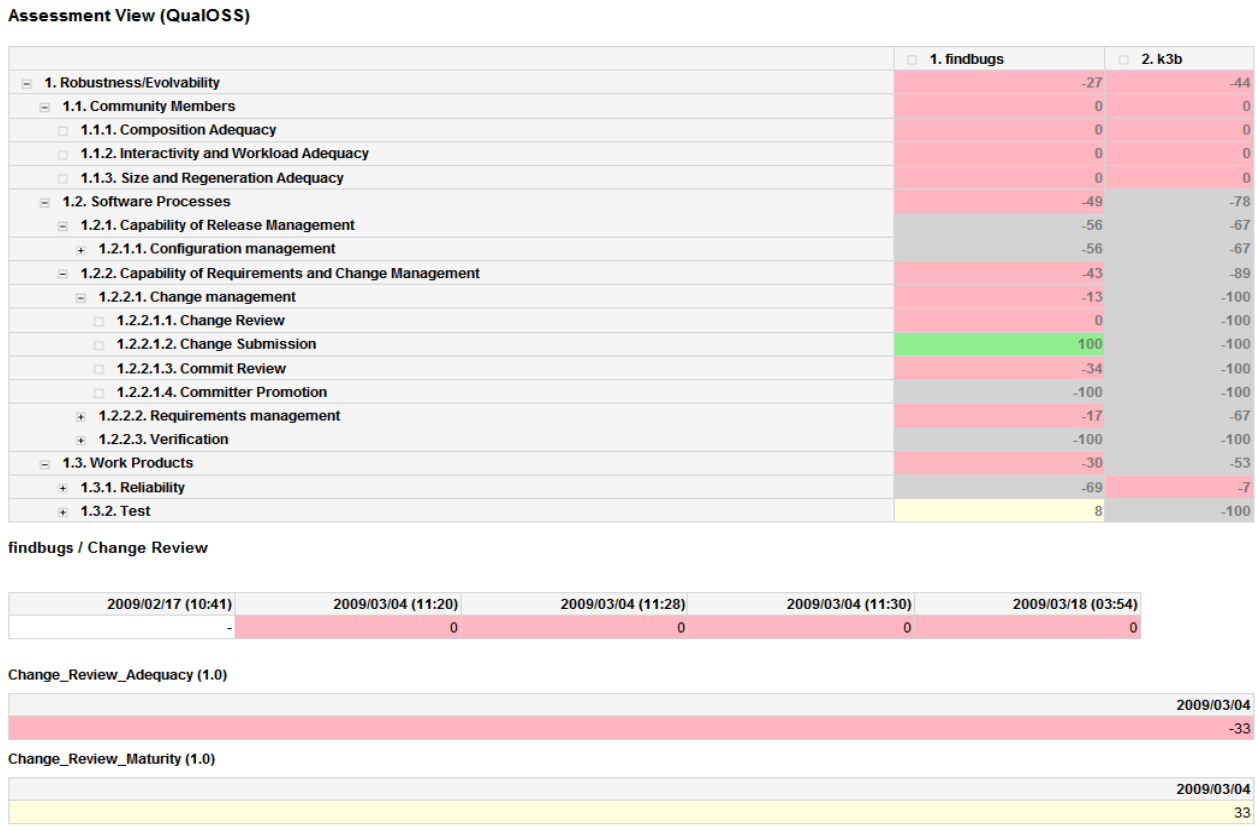
- *Name:* Name of the indicator.
- *Metrics used:* Metrics used to compute this indicator.
- *Rationale:* Explanation of the indicator's purpose.
- *Indicator rule:* Mathematical rule or procedure used to calculate the indicator's value.

8 Visualization Concept

The interpretation of each quality characteristic is visualized on a tree-like structure, reflecting the quality model structure and starting with the top node (Robustness/Evolvability), see Figure 4. From a technical viewpoint, the visualization platform is able to gather data from various databases, such as the QualOSS platform, and visualize those. It is implemented as a web application; therefore, anyone with Internet access and a browser can use it.

Drill-down capabilities allow to view details of the underlying quality characteristics. A click on a leaf characteristics will display the underlying indicator values. Currently, metric values are not

visualized, but we will implement that before the end of the project. The idea is that metric values can be accessed in order to understand an indicator value. Indeed, some of the hypotheses of D5.1 test the sufficiency of drill down capability provided by the QualOSS assessment results.



Assessment View (QualOSS)

	1. findbugs	2. k3b
1. Robustness/Evolvability	-27	-44
1.1. Community Members	0	0
1.1.1. Composition Adequacy	0	0
1.1.2. Interactivity and Workload Adequacy	0	0
1.1.3. Size and Regeneration Adequacy	0	0
1.2. Software Processes	-49	-78
1.2.1. Capability of Release Management	-56	-67
1.2.1.1. Configuration management	-56	-67
1.2.2. Capability of Requirements and Change Management	-43	-89
1.2.2.1. Change management	-13	-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
1.2.2.3. Verification	-100	-100
1.3. Work Products	-30	-53
1.3.1. Reliability	-69	-7
1.3.2. Test	8	-100

findbugs / Change Review

2009/02/17 (10:41)	2009/03/04 (11:20)	2009/03/04 (11:28)	2009/03/04 (11:30)	2009/03/18 (03:54)
-	0	0	0	0

Change_Review_Adequacy (1.0)

2009/03/04
-33

Change_Review_Maturity (1.0)

2009/03/04
33

Figure 4: Result visualization for a QualOSS assessment

The screenshot visualizes the result of the FindBugs and K3b measurement. Clearly, the two projects display significant risks for an enterprise that would want to integrate them in a product, a service or an infrastructure. In particular, the behavior related to software processes and to the community are quite unpredictable. It may also be that further calibration of the indicators is necessary.

9 Conclusions and Future Work

In the remainder of this project, a number of tasks will be achieved as part of WP4.

In Task 4.3, the emphasis will continue on community related datasets and, in particular, on finding features (metrics) that can used to predict other important metrics on which community evolution indicators can be based. Another important part of this task is to refine the thresholds for the existing indicators. In addition, effort to identify the most adequate partitioning of the FLOSSMETRICS dataset in order to obtain highly reliable prediction, i.e., with a high rate of certainty, will continue. In Fall 2009, the relevant metrics (features) of community datasets that are

good predictors will be identified. Subsequently, these good predicting metrics will be used to create new evolution indicators for the standard QualOSS assessment method.

Task 4.4 requires a round of peer-review to validate the outcome of Task 4.2 (which may have required adapting metrics and indicators based on WP3 and WP5 feedback. Furthermore, a last round of validation is planned in October. This last validation will check that new indicators from Task 4.3 are correctly transposed for automated computation in the QualOSS platform. Furthermore, Task 4.4 also reviews the user guide accompanying the standard QualOSS assessment method.

Finally in connection to the implementation of the QualOSS platform in WP2, effort will be directed to improving the interfacing between the QualOSS platform and the visualization tool Specula. Currently, the interfacing requires transferring QualOSS assessment results into a spreadsheet prior to being processed with Specula. In the future version, Specula will directly interact with the QualOSS results repository.

10 APPENDIX I: FindBugs Assessment Results

10.1 Study Object - FindBugs

FindBugs is a static analysis tool to find bugs in Java programs. FindBugs is distributed under the terms of the Lesser GNU Public License and is a free software. As of February 2009, FindBugs has been downloaded more than 500,000 times from SourceForge.net.

10.2 Assessment Process

This appendix presents the data gathered when assessing the FindBugs project. The assessment of the FindBugs project is closely related to the metric and indication process. The metrics and indicators produced during the mentioned process are the input for the FindBugs assessment activities.

The FindBugs assessment process consisted of the following steps, as illustrated in Figure 5:

1. *Analyse FindBugs and gather data*. This step has two inputs. The first input is the FindBugs itself. The second input is the metrics defined during the metric and indicator definition process. During the step the evaluators are gathering data and evaluation experience about the FindBugs project according to the metrics. The step results in the assessment data on the FindBugs project and evaluator's experience on the assessment.
2. *Calculate indicators and define their interpretation*. This step also has two inputs. The first input is the FindBugs assessment data. The second input is the indicators defined during the metric and indicator definition process. During the step the evaluators are calculating indicator values and define their interpretation (green, yellow, red or black). the step results in the indicator interpretation data and evaluators experience on the assessment.

3. *Update metrics according to the assessment experience*. This step defines the feedback on the metrics used during the FindBugs assessment. The primary input of this step is the evaluator's experience acquired during the assessment process. The output is the updated metrics.

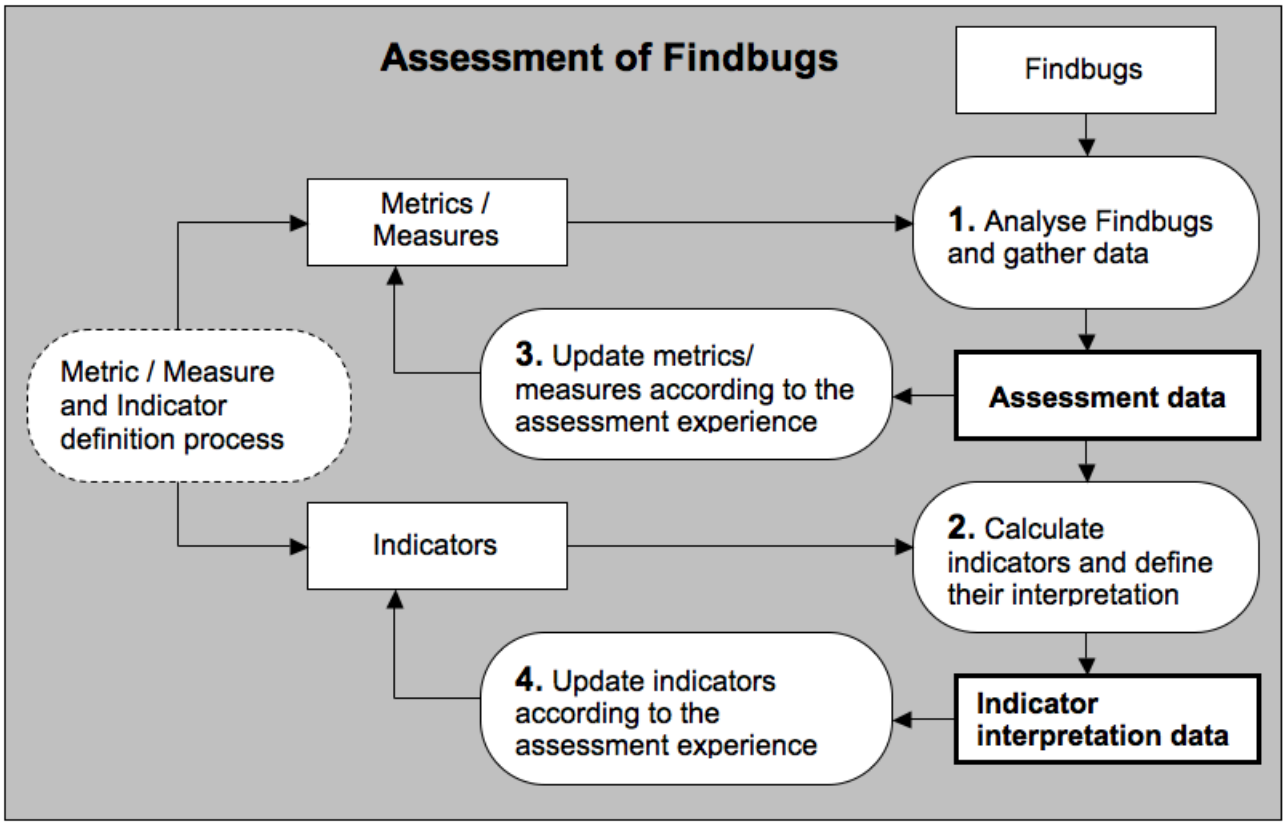


Figure 5: FindBugs assessment process

4. *Update indicators according to the assessment experience*. This step defines the feedback on the indicators used during the FindBugs assessment. The primary input of this step is the evaluator's experience acquired during the assessment process. The output in the updated indicators.

11 APPENDIX II: K3B ASSESSMENT RESULTS

11.1 STUDY OBJECT - K3B

K3b (from KDE Burn Baby Burn) is a CD and DVD burning application for Linux systems. As of February 2009, K3b was downloaded more 1,200,000 times on SourceForge.net. K3b is free software released under the GNU General Public License.

11.2 Assessment Process

This appendix presents the data gathered when assessing the K3b project. The assessment of the K3b project is closely related to the metric and indication process. The K3b assessment process consisted of the following steps, as illustrated in Figure 6:

1. *Analyse K3b and gather data*. This step has two inputs. The first input is the K3b itself. The second input is the metrics defined during the metric and indicator definition process. During the step the evaluators are gathering data and evaluation experience about the K3b project according to the metrics. The step results in the assessment data on the K3b project and evaluator's experience on the assessment.

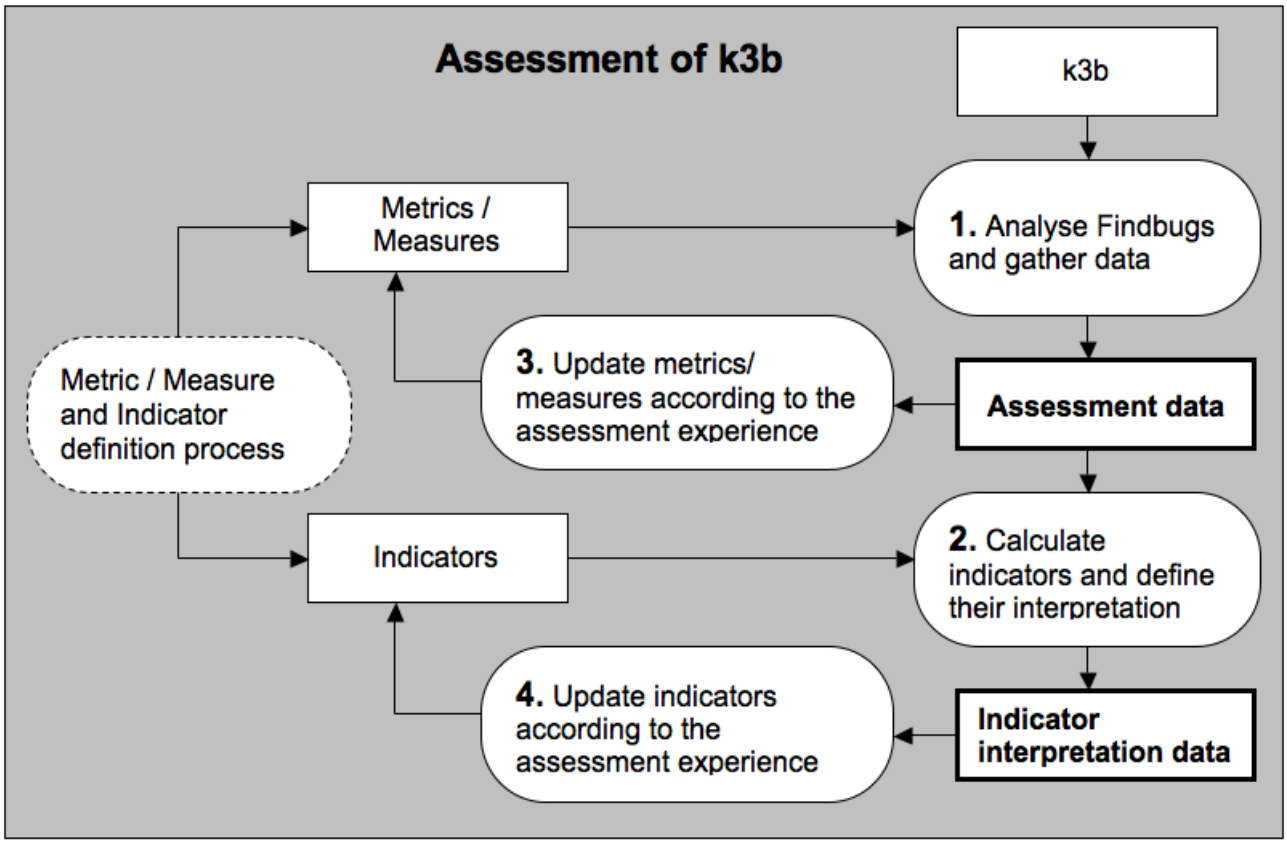


Figure 6: K3b assessment process

2. *Calculate indicators and define their interpretation*. This step also has two inputs. The first input is the K3b assessment data. The second input is the indicators defined during the metric and indicator definition process. During the step the evaluators are calculating indicator values and define their interpretation (green, yellow, red or black). the step results in the indicator interpretation data and evaluators experience on the assessment.
3. *Update metrics according to the assessment experience*. This step defines the feedback on the metrics used during the K3b assessment. The primary input of this step is the evaluator's experience acquired during the assessment process. The output is the updated metrics.

4. *Update indicators according to the assessment experience*. This step defines the feedback on the indicators used during the K3b assessment. The primary input of this step is the evaluator's experience acquired during the assessment process. The output in the updated indicators.