


 <p>Qualoss</p> <p>Sponsored through Framework Programme Sixth (Call 5) by</p> <div style="display: flex; justify-content: space-around; align-items: center;">   </div>		Document Information	
		Version: 2.1 Date : Jan 21, 10 Pages : 25	
		Owning Partner: CETIC	
		Author(s): Frédéric Fleurial Monfils Jacques Flamand	
		Reviewer(s): Israel Herraiz (review 1), Kirsten Haaland (review 2) Nicolas Devos (review 2 Version2)	
		To: European Commission	
		Purpose of distribution: Distribution to the European Commission of the final version of public deliverable D2.1	
The QUALOSS Consortium consists of: CETIC (BE), Facultés Notre Dame de la Paix à Namur (BE), Universidad Rey Juan Carlos (ES), Fraunhofer IESE (DE), ZEA Partners (BE), MERIT (NL), AdaCore (FR), PEPITe (BE)			
Status: <input type="checkbox"/> Draft <input type="checkbox"/> To be reviewed <input type="checkbox"/> Proposal <input checked="" type="checkbox"/> Final/Released		Confidentiality: <input checked="" type="checkbox"/> Public - Intended for public use <input type="checkbox"/> Restricted - Intended for QUALOSS consortium only <input type="checkbox"/> Confidential - Intended for individual partner only	
Deliverable ID: D2.1 Title: Requirements and Specifications of the QUALOSS Platform			
Disclaimer: The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.			

	<p>Requirements and Specifications of the QUALOSS Platform</p> <p>Deliverable ID: D2.1</p>	<p>Page : 2 of 25</p> <hr/> <p>Version: 2.1 Date: Jan 21, 10</p> <hr/> <p>Status : Final Confid : Public</p>
---	--	--

Deliverable: D2.1

Title: Requirements and Specifications of the QUALOSS Platform

Executive Summary:

This document presents the specifications of the Qualoss Platform. This lists the requirements for the platform, an overview of the architecture of the platform and how this platform can be configured.

The Qualoss Platform assists in conducting QualOSS assessments and collecting assessment results in a user specified repository. It allows the collection of analysis results (measures and indicators), automated analysis of data sources and the production of a report. The input for this process is a user defined quality model that specifies the quality characteristics of interest to assess for the selected FOSS Endeavor.

The structure of the document is as follows.

- Section 1 gives the scope of the Qualoss Platform and describes common terms
- Section 2 presents a general overview of the platform with its interfaces and constraints.
- Section 3 lists the specific requirements of the platform and the usage scenarios.




	<p>Requirements and Specifications of the QUALOSS Platform</p> <p>Deliverable ID: D2.1</p>	<p>Page : 3 of 25</p> <hr/> <p>Version: 2.1 Date: Jan 21, 10</p> <hr/> <p>Status : Final Confid : Public</p>
---	--	--

TABLE OF CONTENTS

1	Introduction	5
1.1	Purpose	5
1.2	Scope	5
1.3	Definitions, Acronyms, and Abbreviations	7
1.4	References	7
1.5	Overview	7
2	Overall Description	7
2.1	Product Perspective	7
2.1.1	System Interfaces	9
2.1.2	User Interfaces	9
2.1.3	Hardware Interfaces	11
2.1.4	Software Interfaces	11
2.1.5	Communications Interfaces	11
2.1.6	Memory Constraints	12
2.1.7	Operations	12
2.1.8	Site Adaptation Requirements	12
2.2	User Characteristics	13
2.3	Constraints	13
2.4	Assumptions and Dependencies	13
2.5	Apportioning of Requirements	13
3	Specific Requirements	13
3.1	External Interface Requirements	14
3.1.1	User Interfaces	14
3.1.2	Hardware Interfaces	14
3.1.3	Software Interfaces	14
3.1.4	Communications Interfaces	14
3.2	Software Product Features	14
3.2.1	Feature: Check the workflow	14
3.2.2	Feature: Validate the command line	15
3.2.3	Feature: Validated UserDefinedQualityModel	15
3.2.4	Feature: Validate QualityModelMapping	16
3.2.5	Feature: Launch the analyzers	16
3.2.6	Feature: Record the indicators	17
3.2.7	Feature: Calibrate the indicators	18
3.2.8	Feature: Report the Assessment	19
3.2.9	Feature: ImportMeasuresAndIndicators	20
3.3	Performance Requirements	20
3.4	Design Constraints	20
3.5	Software System Attributes	20
3.5.1	Scalability	20
3.5.2	Reliability	20
3.5.3	Availability	21

	<p>Requirements and Specifications of the QUALOSS Platform</p> <p>Deliverable ID: D2.1</p>	<p>Page : 4 of 25</p> <hr/> <p>Version: 2.1 Date: Jan 21, 10</p> <hr/> <p>Status : Final Confid : Public</p>
---	--	--

3.5.4	Security.....	21
3.6	Logical Database Requirements	21
3.7	Other Requirements	23
3.7.1	Usage scenarios.....	23
3.7.2	Platform specifications.....	24

	<p>Requirements and Specifications of the QUALOSS Platform</p> <p>Deliverable ID: D2.1</p>	<p>Page : 5 of 25</p>
		<p>Version: 2.1 Date: Jan 21, 10</p> <p>Status : Final Confid : Public</p>

1 INTRODUCTION

Qualoss is a tool methodology for the assessment of F/OSS Endeavors. As it will be described in deliverable D4.1, a F/OSS Endeavor is composed of:

- set of community members,
- set of work products (including product components)
- set of software processes
- set of tools (support tools as well as libraries used by the product components)

These interact with each other with the goal of producing an outcome, often the F/OSS component (application).

The Qualoss Methodology is a body of methods. Among others, there are methods for assessing the robustness and evolvability of a F/OSS Endeavor, the Qualoss Assessment Methods. Each method will work at a different level of granularity:

1. The lightweight method: It consists in interpreting existing results from already assessed F/OSS endeavors stored in the Qualoss repository.
2. The middleweight method: It consists in assessing a F/OSS endeavor not yet present in the Qualoss repository. This assessment will be based on the standard *Qualoss Quality Model*.
3. The advanced method: It consists in assessing a F/OSS endeavor based on a custom-made quality model. The tailoring on the quality model may vary from small adjustments to the standard Qualoss quality model but can eventually be completely a new quality model adapted to a very specific context.

To ease the assessment of F/OSS endeavors, for the middleweight and advancedweight methods, a platform will be used. This platform, the *Qualoss Platform*, will store in a structured way the measures and indicators gathered during an analysis or will launch tools to automatically gather measures and compute quality indicators.

1.1 PURPOSE

The purpose of this document is to present the different components of the *Qualoss Platform* and for each component, to describe:


- its purpose
- its interactions
- its configuration

We emphasize that this document is not an implementation guide but in its current form it is a requirements document.

1.2 SCOPE

The software products that will be produced as part of the *Qualoss Platform* are the following:

- The *Qualoss Configurator*
- The *Qualoss Importer*
- The *Qualoss Analyzer*
- The *Qualoss Reporter*
- The *Qualoss Database Connector*
- The *Qualoss tool Connector*

	<p>Requirements and Specifications of the QUALOSS Platform</p> <p>Deliverable ID: D2.1</p>	<p>Page : 6 of 25</p> <hr/> <p>Version: 2.1 Date: Jan 21, 10</p> <hr/> <p>Status : Final Confid : Public</p>
---	--	--

The *QualOSS Configurator* is used to configure the QualOSS platform. The *Qualoss Importer* is used to gather measurements and indicators collected separately in spreadsheets and stores them inside the *Qualoss Database*. The value of those indicators can then be retrieved from the database in order to assess the quality of a given F/OSS Endeavor.

As an alternative, the *Qualoss Analyzer* can collect measurements from different tools thanks to various *Qualoss tool Connector*, and store them inside the *Qualoss Database*. The stored measurements are used by the *Qualoss Reporter* to compute indicators and produce reports that will be used as a mean to assess the quality of a given F/OSS Endeavor.


Relevant benefits: The *Qualoss Platform* allows a generic way to collect measures and indicators across various data sources. It is also highly configurable. The measurements needed for the analysis are performed independently from the *Qualoss* platform and collected in separate spreadsheets (one per characteristic). The advantage of this approach (in comparison with a full automated process) is that the analysis process can be better controlled and that some specific aspects of the artefacts can be taken into account.

The goal of the *Qualoss Platform* is to ease the quality assessment of a F/OSS Endeavor. The standard *Qualoss Quality Model* will be provided for the middleweight assessment, it will be the default quality model compute by the *Qualoss Platform*.

The entry point of the *Qualoss Platform* is provided by command line tools that will launch the correct components based on the command line arguments and configuration files.

The objectives of this platform are threefold:

- To reduce the time needed to assess and analyse different aspects of a F/OSS Endeavor analysis
- To store in a structured way the information gathered during F/OSS Endeavor analysis
- To ease the definition of user-defined quality models and the integration of measurement tools

	<p>Requirements and Specifications of the QUALOSS Platform</p> <p>Deliverable ID: D2.1</p>	Page : 7 of 25
		Version: 2.1 Date: Jan 21, 10
		Status : Final Confid : Public

1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

CSV	Comma-separated value. CSV is an open data storage format.
CVS	Concurrent Versions System
FLOSS	Free <i>libre</i> Open Source Software
FLOSS Endeavor	Endeavor around a FLOSS. The Endeavor is composed of the tools, people, processes and products managed by the Endeavor.
FlossMetrics	EU Funded Project FlossMetrics (FP6-033982) FLOSSMetrics stands for Free/Libre Open Source Software Metrics. The main objective of FLOSSMETRICS is to construct, publish and analyse a large scale database with information and metrics about libre software development coming from several thousands of software projects, using existing methodologies, and tools already developed. The project will also provide a public platform for validation and industrial exploitation of results. See www.flossmetrics.org
HTML	HyperText Markup Language
MySQL	MySQL is an Open Source RDBMS
PDF	Portable Document Format
PostgreSQL	PostgreSQL is another Open Source RDBMS
RAM	Random Access Memory
RDBMS	Relational DataBase Management System
XML	Extensible Markup Language

1.4 REFERENCES

Qualoss Description of Work (also referred to as the Technical Annex to the E.C. Contract)

1.5 OVERVIEW

The document is divided in 2 sections:

- Section 2 briefly describes the *Qualoss Platform* and its intended use.
- Section 3 lists the specific requirements for the *Qualoss Platform* and its configuration.


2 OVERALL DESCRIPTION

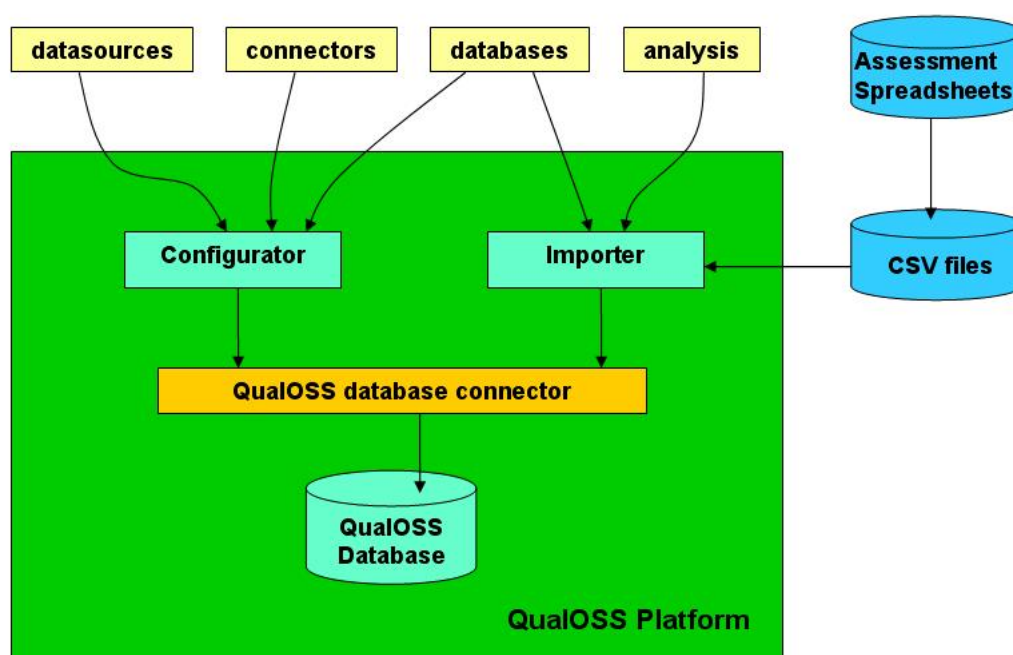
This section describes the general factors that affect the product and its requirements. This section contains background information, but does not state specific requirements.

The *Qualoss Platform* is a collection of components. This platform is a workflow engine that ties together various tools. It uses a database to store its information. The remaining part of Section 2 describes the overall platform and its environment.


2.1 PRODUCT PERSPECTIVE

The following figure gives an overview of the Qualoss platform for the use case of “Importing”.

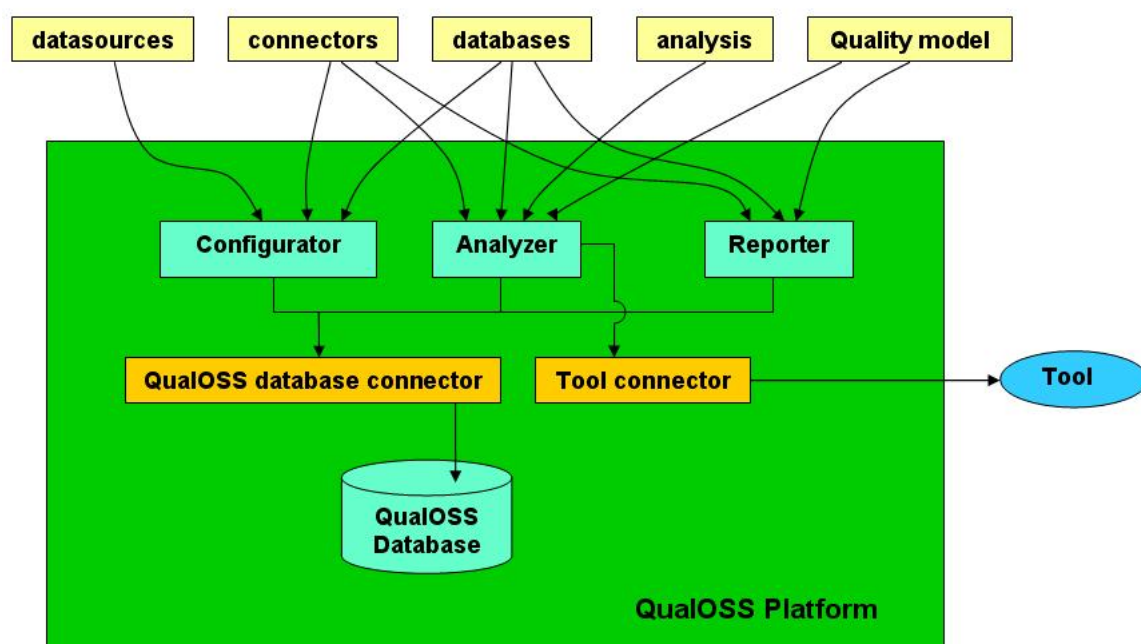
	Requirements and Specifications of the QUALOSS Platform	Page : 8 of 25
		Version: 2.1 Date: Jan 21, 10
		Status : Final Confid : Public
	Deliverable ID: D2.1	



The *Qualoss configurator* is used to configure the *Qualoss* platform. The *Qualoss Importer* is used to gather measurements and indicators collected separately in spreadsheets and stores them inside the *Qualoss Database*. The yellow boxes stand for configuration files that are used for configuring the platform and/or for importing the assessment results.

	<p>Requirements and Specifications of the QUALOSS Platform</p> <p>Deliverable ID: D2.1</p>	Page : 9 of 25
		<p>Version: 2.1 Date: Jan 21, 10</p> <p>Status : Final Confid : Public</p>

The following figure gives an overview of the Qualoss platform for the of use case of “Automatic Analysis” :



During the analysis, the needed measures are automatically collected by the appropriate tools that are launched by the platform using the appropriate connectors. As an additional functionality, the reporter allows to compute and list the values of indicators described by a quality model file.


2.1.1 System Interfaces

Database: The *Qualoss Platform* stores the collected information in a database. The detailed schema of this database is given in section 3. The kind of data base server, the connection parameters and the name of the schema are defined by configuration file, which makes possible to collect the result of an assessment in a “raw” data base, and, after verification, to collect them in a “validated” data base.

Connectors: The *Qualoss Platform* contains interfaces with different external tools through connectors. Each tool has a specific connector. A connector is in fact a mean to “metrify” the output of a given tool, that is, allow the extraction of metrics from the results produced by the run of the given tool. Each connector will be called in turn and will be executed by the *Qualoss Analyzer* if needed. An option at the command line will specify whether launch of the connector should be forced or not (in this case, the result stored in the database is used). For further details see *Qualoss Analyzer* options list.

2.1.2 User Interfaces

In this version, the *Qualoss Platform* is used at the shell prompt and will be implemented as a set of command line tools. An overview of the necessary commands to enable the user to configure, control and analyse a FIOSS Endeavors are provided.

	<p>Requirements and Specifications of the QUALOSS Platform</p> <p>Deliverable ID: D2.1</p>	Page : 10 of 25
		Version: 2.1 Date: Jan 21, 10
		Status : Final Confid : Public

Here are the tools that are provided:

- 1) The *Qualoss Configurator*: to configure the Qualoss platform in order to collect assessment data and/or to launch the analysis based on the user defined characteristics
- 2) The *Qualoss Importer*: to gather Qualoss measures and indicators collected in intermediate files
- 3) The *Qualoss Analyzer*: to launch an automatic analysis based on the user defined characteristics
- 4) The *Qualoss Reporter*: to report an automatic analysis according to the user defined quality model

When launching the tools, the user will see the version of the tool and how he can access the inline help. For example: the user will have an access to an inline help by issuing the “<command> --help” command.

Table 1: Common options for all the tools


Option	Mandatory	Default	Description
help	no	N/a	Prints the list of options and exits
version	no	N/a	Prints the version number of the <i>Qualoss Analyzer</i>
verbose	no	no	Indicates whether the details of the execution of the tool should be displayed to the standard output.
quiet	no	yes	Inverse function of verbose
debug	no	no	Generate debug information
fast	no	no	Indicates whether the Analyzer should optimise its running
log	no		Directory for log files
config	no		get options from configuration file
raw-config	no		get options from raw configuration file
generate-config	no	no	generate config file based on the provided options

Table 2: Specific options for the Qualoss Configurator

Option	Mandatory	Default	Description
connectors	no	config/default/connectors.ini	Path to connectors specifications
databases	no	config/default/databases.ini	Path to databases connections
datasources	no	config/default/datasources.ini	Path to datasources definitions
init	no	no	Force the initialization of the database

Table 3: Specific options for the Qualoss Importer

Option	Mandatory	Default	Description
analysis	no	config/default/analysis.ini	Path to analysis specifications
databases	no	config/default/databases.ini	Path to databases connections
measures	no	at least one or both parameters	Path to CSV file with measures

	<p>Requirements and Specifications of the QUALOSS Platform</p> <p>Deliverable ID: D2.1</p>	<p>Page : 11 of 25</p> <hr/> <p>Version: 2.1 Date: Jan 21, 10</p> <hr/> <p>Status : Final Confid : Public</p>
---	--	---

indicators	no	can be specified	Path to CSV file with indicators
fields	no	config/default/fields.ini	Specify the fields mapping of the CSV files
force	no	no	Force the insertion of measures and / or indicators

Table 4: Specific options for the Qualoss Analyzer

Option	Mandatory	Default	Description
analysis	no	config/default/analysis.ini	Path to analysis specifications
databases	no	config/default/databases.ini	Path to databases connections
connectors	no	config/default/connectors.ini	Path to connectors specifications
quality-model	no	config/default/qualitymodel.yaml	Path to quality model
force	no	no	Force the insertion of measures

Table 5: Specific options for the Qualoss Reporter

Option	Mandatory	Default	Description
analysis	no	config/default/analysis.ini	Path to analysis specifications
databases	no	config/default/databases.ini	Path to databases connections
quality-model	no	config/default/qualitymodel.yaml	Path to quality model
format	no	csv	text xml yaml csv text
output	no	stdout	Output format

2.1.3 Hardware Interfaces

The Qualoss platform will have no hardware interfaces.

2.1.4 Software Interfaces

The Software interfaces are provided by the *Qualoss Connectors*.


2.1.5 Communications Interfaces

There is no specific communication interface since the *Qualoss Platform* is a set of tools that can be run independently. In fact, this Qualoss specifications requirements documents the process that the user has to follow so as to evaluate a given F/OSS Endeavor. See Section on Operations below.

The architecture of the components is as follows, depending on the use case :

Importing use case:

Once the Qualoss platform is configured using the *Qualoss Configurator*, the measures and indicators can be collected in spreadsheets that, after conversion into an intermediate csv format, can be stored (imported) into the *Qualoss Database* using the *Qualoss Importer*. The Qualoss Importer uses another connector, the

	<p>Requirements and Specifications of the QUALOSS Platform</p> <p>Deliverable ID: D2.1</p>	Page : 12 of 25
		Version: 2.1 Date: Jan 21, 10
		Status : Final Confid : Public

Qualoss Database Connector to store the results in the *Qualoss Database*. After that, the appropriate SQL commands can be used to select measures and indicators for a specific F/OSS endeavour.

Automatic Analysis use case:

Each *Qualoss Connector* is activated in due time by the *Qualoss Analyzer*. This analyzer uses a selection algorithm to select the appropriate metrics based on the provided Quality Model.

The *Qualoss Analyzer* uses another connector, the *Qualoss Database Connector* to store the results in the *Qualoss Database*.

The *Quality Model Reporter* can be used to extract the relevant data from the *Qualoss Database* and produces the report as requested by the user.

As shown, the various components are run consecutively and each component stores its results in a central database. The communication interface is then an open connection with the RDBMS to this central database. The local area network is used as a mean to let the client component communicate with the database server.

2.1.6 Memory Constraints

Currently no real estimation of the minimum memory and disk space is available. The first version of the platform will run on a cluster with nodes having 4 Gigabytes of RAM and the database server will have 32 Gigabytes of RAM.

2.1.7 Operations

The operations to be performed on the *Qualoss Platform* consist of the installation, update and calibration of the measurement tools, the backup and recovery of the database and the tuning of the basic configuration files of the *Qualoss Platform*.


Installing measurement tools: New measurement tools can be installed on the platform in order to be used during an analysis. The *Qualoss Platform Administrator* will use tools, scripts and methods described in the *Qualoss Platform Administration Guide* in order to download compile and install the tools that are part of the default platform, that is, the set of tools that have been calibrated during the Qualoss Project.

Updating measurement tools: New versions of measurement tools can be installed on the platform. The *Qualoss Platform Administration Guide* will describe the necessary operations and steps to be performed by the *Qualoss Platform Administrator*.

Backup of the Qualoss Database: The *Qualoss Database Administrator* needs to backup the database schema and data at regular time intervals.

Recovery of the Qualoss Database: The *Qualoss Platform Administrator* can recover a backedup database after a crash of the system or when a change of database or database server has occurred.

Tuning of basic configuration files of the Qualoss Platform: The *Qualoss Platform Administrator* can tune the execution of the analysis by modifying the configuration file of the Qualoss Platform

	<p>Requirements and Specifications of the QUALOSS Platform</p> <p>Deliverable ID: D2.1</p>	Page : 13 of 25
		Version: 2.1 Date: Jan 21, 10
		Status : Final Confid : Public

2.1.8 Site Adaptation Requirements

The Qualoss Platform will provide a set of tools, scripts and methods to perform the following functions:

- Configure the tools to be used during the analysis.
- Check the availability of the measurements that are to be collected by inspecting the database.
- Configure the user-defined Quality Model.
- Export the measurements to various formats like XML and CSV.

2.2 USER CHARACTERISTICS

The intended users of the Qualoss Platform are:

- **Administrators:** with technical expertise for the installation and configuration of tools on Unix or Windows operating systems.
- **Consultants:** self trained, and they will train end-users in the interpretation of measures and indicators collected by the Qualoss platform.
- **Experts:** skills to configure Quality Models that suits their specific user needs.
- **End users:** consult indicators collected by the Qualoss platform .

2.3 CONSTRAINTS

The main constraints of the Qualoss Platform:

- **Reliability requirements:** The prototype of the platform will be hosted on CETIC's cluster and will benefit from this environment. Service Level Agreements have still to be defined.
- **Criticality of the application:** The application is not per se critical. The data and information that could be extracted from the database can be of high value. Hence an authentication mechanism will be used on the database engine.
- **Safety and security considerations regulatory policies:** none
- **Hardware limitations:** Only the amount of RAM is relevant. The application will be designed to run on a Linux platform
- **Interfaces to other applications:** These interfaces will be handled through the Qualoss Connectors.
- **Parallel operation:** In the first phase, the application will only be sequential.
- **Audit functions:** A logging mechanism will be applied. This functionality can be enabled or disabled using command line switches.
- **Higher-order language requirements:** The programming language will be Python, an object oriented language.
- **Signal handshake protocols:** none.

2.4 ASSUMPTIONS AND DEPENDENCIES


The following factors affect the requirements:

As said, the current prototype version will be deployed on CETIC's cluster. However, future version will only require a single server so as to allow any company to use the Qualoss Platform.

2.5 APPORTIONING OF REQUIREMENTS

The following requirements may be delayed until a future version of the system:

- Access the tool through a web interface.
- Download of generated reports.
- Search for past analysis requested by other users.

	<p>Requirements and Specifications of the QUALOSS Platform</p> <p>Deliverable ID: D2.1</p>	Page : 14 of 25
		Version: 2.1 Date: Jan 21, 10
		Status : Final Confid : Public

3 SPECIFIC REQUIREMENTS

This section describes all the Qualoss Platform requirements at a level of detail that enables the correct implementation and test of the platform.

3.1 EXTERNAL INTERFACE REQUIREMENTS

3.1.1 User Interfaces

The Qualoss Platform is composed of a set of scripts that can be launched from the command line of a shell, using a command line interface. There is no graphical user interface available.

The tools are connected together through a configuration file describing the sequence of their execution.

The entry point to the Qualoss platform consist of several the Python scripts.

3.1.2 Hardware Interfaces

There is no hardware interface. The Qualoss Platform does not use any hardware except for the connection with a possibly remote database server.

3.1.3 Software Interfaces

Each connector (QualossConnector) has an interface to a program such as Checkstyle.

3.1.4 Communications Interfaces

The Qualoss Platform communicates only through database/flat file connectors. There is no plan to provide information to PDAs, GSMs etc.

3.2 SOFTWARE PRODUCT FEATURES

This section presents the features of the product.

3.2.1 Feature: Check the workflow

Purpose

The Qualoss Platform is in fact a workflow engine that takes a description of the tasks that it has to perform and excutes these task one at a time following the provided order.

The provided workflow should allow the correct execution of the analysis. Hence, some tests and checks are needed before starting the analysis.

Stimulus/Response Sequence


The user specifies the Workflow Configuration.

Associated Functional Requirements

3.2.1.1 Functional Requirement: Tasks pre-requisites are satisfied

Introduction: Sometimes the execution of a task needs the execution of another task to have happened before it. The purpose of this feature is to check that the run of the needed tasks have been specified before the execution of a specific task.

Input(s): The QualossWorkflowConfiguration files.

	<p>Requirements and Specifications of the QUALOSS Platform</p> <p>Deliverable ID: D2.1</p>	Page : 15 of 25
		Version: 2.1 Date: Jan 21, 10
		Status : Final Confid : Public

Processing: Each task checks that their pre-requisites are defined before them.

Output(s): Each missing pre-requisite for a task is displayed

3.2.1.2 Functional Requirement: Tasks mandatory options are provided

Introduction: Each task can take a number of options (optional or mandatory ones). When a task has mandatory options, they should be provided in the configuration files or as a parameter of the corresponding scripts.

Input(s): The QualossWorkflowConfiguration files.

Processing: Each task checks that their mandatory options are defined.

Output(s): Each missing mandatory option for a task is displayed

3.2.2 Feature: Validate the command line

Purpose

The Qualoss Platform, when launched, validates the options. A matrix of the options, their combination, their default value and their behaviour will be designed.

Stimulus/Response Sequence

The user launches an analysis on the server.

Associated Functional Requirements

3.2.2.1 Functional Requirement: An output option is required

Introduction: Computing a lot of information for nothing is not interesting. Hence, an "output" option is always needed from all the options provided at command line or in the configuration file.

Input(s): Command line options.

Processing: The application checks that there is an option from the "output" category, either provided at the command line or from the configuration file.

Output(s): An error message is printed if there is no "output" option.


3.2.2.2 Functional Requirement: Command line options respects the options matrix

Introduction: The command line options can be combined in different ways. A matrix with the possible combination is defined for the available options.

Input(s): Command line options.

Processing: The application checks that the provided options respect the matrix.

Output(s): An error message is printed if the combination is not allowed in the matrix.

	<p>Requirements and Specifications of the QUALOSS Platform</p> <p>Deliverable ID: D2.1</p>	Page : 16 of 25
		Version: 2.1 Date: Jan 21, 10
		Status : Final Confid : Public

3.2.3 Feature: Validated UserDefinedQualityModel

Purpose

The user has the possibility to provide his own quality model based on the default Qualoss Quality Model. The purpose of this feature is to check that the provided quality model respects the rules established by the Qualoss methodology.

Stimulus/Response Sequence

The user specifies at the command line the name of their Quality Model through the option --qualitymodel.

Associated Functional Requirements

Introduction: The user can provide his quality model through the weights given to the various characteristics and sub-characteristics. Therefore the sum of the weights must be equal to 100%.

Input(s): The configuration file elements 'characteristics' and 'sub-characteristics'.

Processing: Checks against the values of the 'weight' attributes.

Output(s): An error message is printed if the sum of the weights is not equal to 100%.

3.2.4 Feature: Validate QualityModelMapping

Purpose

The user can provide a quality model mapping that specifies the tools that will be used to collect the needed information.

Stimulus/Response Sequence

The users specifies the QualityModelMapping

Associated Functional Requirements

Functional Requirement: All tools are calibrated

Introduction: The Qualoss Platform defines the way how the tools should compute every indicator.

Input(s): The configuration file sections.

Processing: Every section related to the tools is analysed. There should be a mapping function that explains how to compute the needed indicator from the results of the tools.

Output(s): All sections that do not contain a mapping function are reported


3.2.5 Feature: Launch the analyzers

Purpose

In order to provide the assessment, the analyzers should be launched.

Stimulus/Response Sequence

The configuration file describing the analysis to be launched has been provided.

	<p>Requirements and Specifications of the QUALOSS Platform</p> <p>Deliverable ID: D2.1</p>	<p>Page : 17 of 25</p> <hr/> <p>Version: 2.1 Date: Jan 21, 10</p> <hr/> <p>Status : Final Confid : Public</p>
---	--	---

Associated Functional Requirements

3.2.5.1 Functional Requirement: All analyzers are available

Introduction: The analyzers that have to be launched are available on the platform.

Input(s): The configuration file specifying the analyzers to be launched.

Processing: The command line of the analyzers to be launched is tested. If the process cannot be launched a record is generated.

Output(s): The name of the missing analyzers is recorded.

3.2.5.2 Functional Requirement: Only needed analysis is performed

Introduction: Not all analysis needs to be computed. This is important to reduce the load on the main server but also to make the system modular.

Input(s): The configuration file specifying the analyzers to be launched.

Processing: There is a table that holds the following information (analyzer, analyzer version, project, project version, date of last analysis).

Output(s): Only the analyzers that have not been launched since a defined period of time on the same version of a project will be candidate to be launched.

3.2.5.3 Functional Requirement: Connectors are processed in turn

Introduction: All the connectors defined in the configuration file are processed one after the other in the specified order.

Input(s): The configuration file specifying the analyzers to be launched and their order.

Processing: The configuration file is checked and the sections are sorted according to their order.

Output(s): Connectors/Analyzers that do not contain the order are not executed.

3.2.5.4 Functional Requirement: A record is created for every indicator that cannot be computed

Introduction: If there is no mean to compute an indicator, a clear message should be printed explaining this. This is required because this information will be used at report generation time.

Input(s): The configuration file specifying the indicators to be computed.


Processing: The command line of the analyzers to be launched.

Output(s): The name of the missing indicator is recorded.

3.2.6 Feature: Record the indicators

Purpose

The purpose of the Qualoss Platform is to provide reports on the quality of the FLOSS. This assessment is based on the value of the indicators. These indicators are computed from the value of the measurements collected on various project data sources.

	<p>Requirements and Specifications of the QUALOSS Platform</p> <p>Deliverable ID: D2.1</p>	<p>Page : 18 of 25</p> <hr/> <p>Version: 2.1 Date: Jan 21, 10</p> <hr/> <p>Status : Final Confid : Public</p>
---	--	---

Stimulus/Response Sequence

The user launches the analysis and the workflow configuration file has been validated.

Associated Functional Requirements

3.2.6.1 Functional Requirement: Check availability of measurements

Introduction: All the needed measurements for an indicator have to be available, because the value of this indicator is the combination of these measurements.

Input(s): Resulting measurements from the analyzers.

Processing: A check is done on every measurement record required for this indicator.

Output(s): All missing measurements for a defined indicator are reported.

3.2.6.2 Functional Requirement: Records are stored in the database

Introduction: The measurements performed by the analyzers are transformed into indicators. The measurements and the indicators are stored in the database.

Input(s): Resulting measurements from the analyzers.

Processing: The measurements are combined and the combination of these values is the indicator required by the Quality Model.

Output(s): The indicator is stored on the correct table.

3.2.7 Feature: Calibrate the indicators

Purpose

The purpose of the calibration is to validate that the measurements are in line with the context of the FLOSS Endeavor.

Stimulus/Response Sequence

Associated Functional Requirements

3.2.7.1 Functional Requirement: Manual calibration is allowed to experts


Introduction: The indicators should be validated. They could be validated manually by experts. To decide on the validation, the expert bases itself on the context of the FLOSS Endeavor.

Input(s): The indicators in the correct table of the RawDatabase.

Processing: The measurements and the indicators are checked against facts found in the FLOSS Endeavor.

Output(s): The indicators are calibrated and transposed to the ValidatedDatabase.

V1.1 : no calibration / validation implemented (but it is possible to use a “test” database and a “production” database after validation.

	<p>Requirements and Specifications of the QUALOSS Platform</p> <p>Deliverable ID: D2.1</p>	<p>Page : 19 of 25</p> <hr/> <p>Version: 2.1 Date: Jan 21, 10</p> <hr/> <p>Status : Final Confid : Public</p>
---	--	---

3.2.7.2 Functional Requirement: Automatic calibration is allowed to expert processes

Introduction: The indicators should be validated. They could be validated automatically by expert processes. To decide on the validation, the expert processes base themselves on rules and other indicators found in the context of the FLOSS Endeavor.

Input(s): The indicators in the correct table of the RawDatabase.

Processing: The measures and the indicators are checked against facts found in the FLOSS Endeavor.

Outputs: The indicators are calibrated and transposed to the ValidatedDatabase. ==> calibration /second data base not implemented yet

V1.1 : not implemented

3.2.7.3 Functional Requirement: Automatic warning on FLOSS facts

Introduction: Some facts and measurements taken on a FLOSS Endeavor could seem strange and should be reported: for example a huge number of commit for a single account, or only one account that is committing to the CVS repository.

Input(s): The measurements from the tables in the database, some thresholds and ratios defined in the configuration file.

Processing: The measurements are validated against the defined rules.

Output(s): All measurements that seem strange are reported to the standard error and to a dedicated table in the database and used during the generation of the report.

V1.1 : not implemented

3.2.8 Feature: Report the Assessment

Purpose

When all the indicators are computed, the report can be generated based on the Quality Model configuration file provided by the user.

Stimulus/Response Sequence

Generate report is enabled and all the needed indicators for the Quality Model are present in the database

Associated Functional Requirements

3.2.8.1 Functional Requirement: All the needed indicators are present in the database


Introduction: To produce the report, the indicators have to be present in their correct location inside the database.

Input(s): The indicators computed for a certain version of a FLOSS Endeavor.

Processing: Based on the provided Quality Model, the list of indicators is extracted and checked against the available indicators inside the database.

Output(s): All missing indicators are listed and a message is printed at the standard error and recorded in a table of the database.

3.2.8.2 Functional Requirement: Comments are generated based on indicators value

	<p>Requirements and Specifications of the QUALOSS Platform</p> <p>Deliverable ID: D2.1</p>	<p>Page : 20 of 25</p> <hr/> <p>Version: 2.1 Date: Jan 21, 10</p> <hr/> <p>Status : Final Confid : Public</p>
---	--	---

Introduction: The automation of the generation of the report does not prevent an accurate comment on the value of an indicator.

Input(s): A list of sentences/comments that depends on the thresholds of indicators/measurements.

Processing: All indicators/measurements are checked against the list and, based on their value, the appropriate comment is selected.

Output(s): The selected comment is inserted in the report.

3.2.9 Feature: ImportMeasuresAndIndicators

Purpose

The user has the possibility to import the value of measures and indicators collected in an intermediate file. The purpose of this feature is to check that the provided information respects the rules established by the Qualoss methodology before storing them in the Qualoss database.

Stimulus/Response Sequence

The user specifies at the command line the name of the measures file and/or of the indicators file.

Associated Functional Requirements

Introduction: The user can provide measures and indicators collected in an intermediate file.

Input(s): The intermediate files containing measures or indicators.

Processing: Check the contents of the intermediate files

Output(s): An error message is printed if columns are missing or if the referenced datasources / artefacts are missing in the Qualoss database.

3.3 PERFORMANCE REQUIREMENTS

None

3.4 DESIGN CONSTRAINTS


Concerning the design constraints, the Qualoss Platform will run on Linux.

3.5 SOFTWARE SYSTEM ATTRIBUTES

The following sections list the attributes of the Qualoss Platform that have to be objectively verified. These are the non-functional requirements that the Qualoss Platform should satisfy.

3.5.1 Scalability

No real requirements since the analysis will be launched on the CETIC's cluster. See below in the section 3.5.3. on Availability.

	<p>Requirements and Specifications of the QUALOSS Platform</p> <p>Deliverable ID: D2.1</p>	Page : 21 of 25
		Version: 2.1 Date: Jan 21, 10
		Status : Final Confid : Public

3.5.2 Reliability

Maturity: *The frequency of failure of the system could be quite high in the prototyping phase. During exploitation, the frequency of failure of the system should be less than 1 failure per 100 import action or report generation.*

Fault tolerance: *If a user asks for a report based on its QualityModel, when the task has been accepted, the user will be presented a report. This means that even if a connection problem arises during the computation of the results, a report will be generated and available for the user.*

For every indicator of the UserDefinedQualityModel that should be computed to produce information, an indication should be present in the report. This means that every indicator will have a value or an indication that the computation was not possible, either as “Not appropriate” or “Could not be computed”

3.5.3 Availability

The Qualoss Platform is a command launched at the shell. The availability is thus the one of the shell. However, some analysis could take a long time to proceed. Utilities such as the one provided on the operating system could be used to request the amount of available memory before starting an analysis.

Note: During the EU Qualoss project, the QualossPlatform will be hosted on CETIC's environment which is a cluster. This means that it should be possible to launch tasks that will be distributed over the different machines composing the CETIC's cluster.

3.5.4 Security


Factors protecting the Qualoss software from accidental or malicious access, misuse or modification:

- **Cryptography:** Apart from the required SSH login and remote communications, no cryptography is used inside the Qualoss Platform.
- **Authorization and Authentication:** to access the Qualoss Platform, the user has to authenticate himself on the machine. Authorization to execute the Qualoss script is granted on users belonging to a specific group.
- **Activity logging:** All the actions performed by the launching of the Qualoss Platform can be saved in a log file.
- **Restrictions on intermodule communications:** Every communication between modules should use an identifier and this identifier should be recognized by the destination module.
- **Data integrity checks:** Checks on the database are used. Prepared statements and stored procedures should also be used if available on the database management system.

3.6 LOGICAL DATABASE REQUIREMENTS

The database contains, among others, the following information:

- *Endeavor (name, version)*
- *Connector to Tool (name, version)*
- *DataSource and Artefacts*
- *Metric (name)*
- *Measurement (metric, value)*
- *Indicator (value, color)*

	<p>Requirements and Specifications of the QUALOSS Platform</p> <p>Deliverable ID: D2.1</p>	Page : 22 of 25
		Version: 2.1 Date: Jan 21, 10
		Status : Final Confid : Public

Record database time. *The time when every measurement was entered in the database must be recorded.*

Link measurement to artefact. *Every measurement must be associated to the artefact on which it was taken.*

Link measurement to metric. *Every measurement must be associated to a metric. (where a metric = a measurement unit).*

Link metric to artefact. *Every metric must be associated to an artefact type.*

Check type of artefact and metric. *The type of an artefact on which measurement may be taken must be known and must match with the type associated to metric. For example, an artefact to measure has the type source-code file and the metric cyclomatic complexity is also associated to the type source-code file hence the measurement of the artefact for its cyclomatic complexity can take place. On the other hand, the metric number of publications, which is associated to the type publication list, cannot take place on the given artefact of type source-code file.*

Unique artefact Floss. *Every artefact belong to a single FLOSS project and when possible, to a single releases of a FLOSS project.*

Record Uri of artefact. *The universal location of every artefact must be recorded in our database. This information can eventually be decomposed in several fields if needed. The important aspect of this requirement is to have the capacity to recover every artefact measured from its original storage.*

Record measurement tool. *Record the measurement tool and its version used for computing a measurement.*

Record user for measurement. *Record the person who performed the measurement.*

Allow measurement comments. *Record the description of the measurement environment and how the measurement tool was used, for example, the command line used and the content of configuration files for the measurement tools.*


	<p>Requirements and Specifications of the QUALOSS Platform</p> <p>Deliverable ID: D2.1</p>	Page : 23 of 25
		Version: 2.1
		Date: Jan 21, 10
		Status : Final Confid : Public


Table 5: The first version of the schema of the Qualoss Database:

Table	Description
analysis	An analysis is performed on a FOSS Endeavor. A release is a human label given to a specific revision. It is possible to give various names to the same revision. A revision is a particular version of an entity on which measurements will be taken. For example, this can be the revision number 23465 of a given file. It has to be linked to a given scope, because this version of the file has been produced for a given purpose.
datasource	A data source is the container holding entities to be measured. It can also be considered as the type of an artefact. Examples of data sources: <ul style="list-style-type: none"> • issue tracking system • distribution list
datasource_type	The data source types configured on the platform.
artefact	An artefact is the entity to be measured by a tool. Examples of artefact: <ul style="list-style-type: none"> • file • committer • issue
artefact_type	The artefact types configured on the platform.
metric	A metric is the definition of a measurement taken on a given data source. Example of metrics: <ul style="list-style-type: none"> • Number of Lines Of Code (scope: Class) • Number of Lines Of Code (scope: Endeavor) • Number of Methods (scope: Class) • Number of Methods (scope: Endeavor) • Number of Parameters (scope: Method)
measure	A measure is the value of the measurement taken on a given source by a given tool for a given revision or imported from an intermediate file
indicator	An indicator is the value and the colour of an indicator imported from an intermediate file
connector	A tool is used to compute the value of the measurement. Example of tools: <ul style="list-style-type: none"> • CVSAAnaly • Checkstyle
connector_configuration	Configuration data used for a specific run of a tool

3.7 OTHER REQUIREMENTS

3.7.1 Usage scenarios

The tool is (in the first phase) a set of command line tool.

	<p>Requirements and Specifications of the QUALOSS Platform</p> <p>Deliverable ID: D2.1</p>	<p>Page : 24 of 25</p> <hr/> <p>Version: 2.1 Date: Jan 21, 10</p> <hr/> <p>Status : Final Confid : Public</p>
---	--	---

Usage Scenario 1: Access the platform

The user can access the platform by connecting on the server on which the Qualoss platform runs. To do this the user enters his login and password for this server.

- *“anonymous” access: A special account “anonymous” allows the retrieval (search, select, download) of publicly available reports.*
- *“admin” access: A special account “admin” allow the configuration of the Qualoss platform itself*
- *“expert” and “advanced” access: these two accesses allow the request and retrieval of personal and publicly available reports*

Usage Scenario 2: Import measures and indicators

It is possible to import the result of the analysis of a F/OSS Endeavor collected in an intermediate file by issuing the correct command at the command line prompt.

Usage Scenario 3: Request analysis

It is possible to request the analysis of a F/OSS Endeavor by issuing the correct command at the command line prompt.

Usage Scenario 4: Retrieve analysis

Retrieve a particular analysis by looking for the desired one on the output folder created by the analysis.

3.7.2 Platform specifications

Overview

The user configures the Qualoss platform with the appropriate configuration files (databases.ini, connectors.ini and datasources.ini) using the Qualoss Configurator. Based on this, the user can either store the value of measures and indicators for a specific analysis by using the Qualoss Importer component or can launch an automatic analysis based on a UserQualityModel by using the Qualoss Analyzer (that itself uses the various *Qualoss Connectors* to launch, extract and transform the available data from the tools), and then get a report by using the Qualoss Reporter Component.

Configuration File: DatabaseConnector


This configuration file lists the information needed to connect to the Qualoss database e.g.

- *Type of database server*
- *Schema name*
- *Connection parameters*

Configuration File: Analysis

The needed information for describing the project under analysis e.g.

- *the name of the project*
- *the “data sources” of the project (information can be found on various web sites, ohloh, sourceforge, FSF Directory, ...)*
 - *url of CVS*
 - *url of SVN*
 - *url of bugtracking system*
 - *address of mailing lists and their roles*

	<p>Requirements and Specifications of the QUALOSS Platform</p> <p>Deliverable ID: D2.1</p>	<p>Page : 25 of 25</p> <hr/> <p>Version: 2.1 Date: Jan 21, 10</p> <hr/> <p>Status : Final Confid : Public</p>
---	--	---

Configuration File: ToolsConnectors

This configuration file lists the tools and specifies their configuration.

- *Path to the executable*
- *Default options*
- *Options to launch the analysis, given the location of the data to analyse*
- *Options to extract the results of the analysis*
- *Options to extract a particular "metric" out of the results of the analysis*

Configuration File: QualityModel

This quality model is an XML or INI file containing the Qualoss tree and the weights allocated to each of the nodes.

- *Indicators, derived Measurements and Measurements.*
- *Definition of the levels of colours.*
- *Weight of these indicators.*

Component: QualossConfigurator

The QualossConfigurator component takes the DatabaseConnector, Analysis and ToolConnectors configuration file to initilys the Qualoss database.

Component: QualossImporter

The QualossImporter component import measures and/or indicator informtion from intermediate files using the Analysis configuration file

Checks:

- *Check that all source and artefact referred in the intermediate files are present in the Qualoss database*

Component: QualossAnalyzer

The engine uses the UserQualityModel as its configuration. It uses the required connectors to

Checks:

- *configuration is valid*
- *tools are available*
- *metrics exist*
- *dependencies are already analysed*

Component: QualossReporter

The reporter is used to compute the indicators accordingly with the quality mode, to store them in the data base and to generate a report

Checks:

- *rules are followed*
- *needed data are computed and stored in the database*