QualiPSo

Quality Platform for Open Source Software

IST- FP6-IP-034763

A5.D1.5.2

The general structure, characteristics, and evolution of OSS products and artifacts (V2)

Gabriele Basilico
Vieri del Bianco
Luigi Lavazza
Sandro Morasca
Davide Taibi
Davide Tosi

Due date of deliverable: 30/06/2011

Actual submission date: 30/06/2011

This work is partially funded by EU under the grant of IST-FP6-034763.

Change History

Version	Date	Status	Author (Partner)	Description

EXECUTIVE SUMMARY

A number of factors are believed to influence the trustworthiness of Open Source Software (OSS). Deliverable D1.5.1.1 identifies and ranks a set of trustworthiness factors according to the relevance they have for OSS developers, modifiers, and final users. Prior to evaluating the trustworthiness of OSS according to this prioritized list of factors, it is necessary to check whether the information needed to evaluate these factors can actually be retrieved, with what degree of completeness, and with what effort.

In this deliverable, we analyze a set of OSS projects that are stored in common OSS repositories, to assess whether OSS repositories provide OSS developers, modifiers, and final users with enough information for them to evaluate OSS projects. Additional help may also come from OSS Repositories of Repositories (RoRs), i.e., repositories that aggregate several OSS projects from other repositories or private websites by extracting data and collecting various measures.

First the investigation documented in this report gathered information from the field about the information on the trustworthiness characteristics that is actually available in OSS portals and repositories. Then, based on the results of this analysis, OP^{2}A (Open source Product Portal Assessment), an assessment model for OSS portals and repositories, was defined.

During the information gathering phase, we investigated 1) OSS repositories and then 2) specific OSS projects, to acquire a better understanding of the trustworthiness characteristics of both OSS repositories and OSS products.

1) By studying a number of important OSS repositories, along with their tools and platforms, we obtained an overview of the structure of OSS repositories. Special attention was paid to the kind of information that is easily accessible (and available) in OSS repositories and can influence developers and final users in selecting OSS projects.

2) Based on the factors identified in D1.5.1.1 [1], we analyzed a number of OSS projects to gather information on the generally available characteristics that can be used for evaluating trustworthiness. The analyzed OSS projects have been selected according to a number of criteria such as their application domains, their developers' community size, and the language used for their implementation. We identified an initial set of 96 OSS projects, out of which 32 were selected for further analysis because of their relevance and their full coverage of the criteria, by also taking into account the effort needed to deeply analyze each of these OSS projects. In this document, we discuss the 32 most relevant projects that have been analyzed so far.

   The analysis of each of the projects was carried out in several steps.

   First, we looked for the factors that were readily available simply by surfing the web sites of the projects, simulating what developers and users usually do when they need to assess OSS products. We carried

out this analysis on a subset of 11 OSS projects out of the 32 we had selected. Analyzing the results of this investigation, we discovered that generally, for most of the trustworthiness factors, the available information is not sufficient to make an objective assessment, although some factors have been ranked in D1.5.1.1 as very important.

Second, we defined a set of different proxy-measures to use whenever a factor cannot be directly assessed on the basis of readily available information. Moreover, some factors are not measurable if developers do not explicitly provide essential information. For instance, this happens for factors like the number of downloads, which cannot be evaluated in a reliable way if the development community does not publish it.

Finally, a new analysis process on the complete set of 32 OSS projects was carried out by using the actual measures and the proxy-measures.

The analysis of the projects shows that users always bump into difficulties when they assess OSS projects. The large majority of the trustworthiness factors was not completely measurable neither by means of the proxy-measures because of lack of information on the project websites.

The results of the previous investigation have allowed us to define OP^{2}A: a model for evaluating the quality of web portals that store OSS products. OP^{2}A can be used as the starting point for a model that reduces the degree of subjectivity involved in the assessment of the quality of OSS portals. Specifically, OP^{2}A classifies OSS portals into three categories (“gold,” “silver,” and “bronze”) plus an additional category (“unclassified”) depending on the completeness and quality of the information available about the important trustworthiness factors. So, OP^{2}A can be used by OSS developers and users to evaluate the quality of OSS portals and identify their strengths and weaknesses, so they can be improved over time.

The results of this WP are beneficial for the other WPs and especially WP5.3, which deals with the identification and refinement of trustworthiness characteristics and factors in OSS products and artefacts. The approach is iterative so that the next version of our analysis will use the measures that are defined in WP5.3 via the Goal/Question/Metric paradigm. Conversely, the results of the next round of analysis will be used to validate and refine the measures defined in WP5.3.

Document Information

IST Project Number	FP6 – 034763	Acronym	QualiPSo
Full title	Quality Platform for Open Source Software		
Project URL	http://www.qualipso.org		
Document URL			
EU Project officer	Michel Lacroix		

Deliverable	Number	A5.D1.5.2	Title	The general structure, characteristics, and evolution of OSS products and artifacts (V2)
Work package	Number	5.2	Title	Analysis of relevant open source projects and artefacts
Activity	Number	5	Title	Trustworthy results

Date of delivery	Contractual	30/06/2011	Actual	30/06/2011
Status	Version 2, dated 30/06/2011		final ☑	
Nature	Report ☑ Demonstrator ☐ Other ☐			
Dissemination Level	Public ☑ Consortium ☐			
Abstract (for dissemination)				
Keywords				

Authors (Partner)	Author1 (Partner1), Author2 Partner2),		
Responsible Author	Author 1	Email	
	Partner	Phone	

TABLE OF CONTENTS

1 INTRODUCTION

The success of Open Source Software (OSS) is due to multiple reasons, ranging from technical qualities to financial, ethical, and political motivations. Nonetheless, the adoption of OSS is still limited. The reason is that, in several cases, OSS fails to convince potential users that its adoption is safe enough and does not involve more risks than purchasing commercial software.

The process of a new software adoption, and particularly of OSS, usually includes a preliminary step in which the potential users collect information about the products. For this purpose, the web is clearly an extremely valuable and easily accessible source of information. There are many websites that host several projects (Repositories) and others that gather information from repositories collecting metrics and statistics.

Since repositories were created, OSS has increased its diffusion due to the availability of capabilities for developing and collaborating among users and developers.

To simplify the diffusion of OSS, it is therefore necessary to identify, quantify, and assess a number of quality factors that affect trust in OSS products. These factors may be related to the software products as well as to the artefacts produced during software development and the OSS development process.

The investigation documented in this report addresses the following three tasks, which have been carried out to acquire a better understanding of the trustworthiness characteristics of OSS repositories and OSS products and define an OSS portal assessment model.

1. We study a number of important OSS repositories, along with their tools and platforms. The idea is to have an overview of the structure of OSS repositories, with special attention to the kind of information that is easily accessible and available in OSS repositories and that can influence developers and final users in selecting OSS products.
2. Based on the business goals identified in WP5.1 [1], we analyze OSS projects, products, and artefacts in order to gather information on their generally available characteristics that can be used for the evaluation of their trustworthiness.
3. Based on the analysis of OSS products and artefacts, we defined OP^{2}A (Open source Product Portal Assessment), an assessment model that may be used by OSS users to evaluate the quality of OSS portals, and by developers to improve the quality of their own portals.

The commonalities and differences in the characteristics identified and analyzed found in points 1 and 2 above are also used to prepare the ground for other work packages, especially WP5.3, which deals with the identification of trustworthiness characteristics and factors in OSS products and artefacts and their quantification through measures. The approach used starting from WP5.3 on is iterative so that the next version of our analysis will use the measures defined in WP5.3. Conversely, the results of the next project analysis will be used to validate and refine the GQM metrics.

The remainder of this deliverable is organized as follows. In Section 2, we describe the projects selected for the analysis and the rationale used for the choice. In Section 3, we describe the analysis we carried out and we discuss the results of the refined analysis. Section 4 introduces the definition of OP^2A, its characteristics, and shows its use through an application example. In Section 5, we draw the conclusions of this investigation.

2 CONTENT

In our analysis, we are interested in repositories because they are a common way to organize and evaluate OSS projects.

OSS products can be hosted in private websites or in repositories. For instance, the Apache httpd server is hosted in the Apache Software Foundation repository while Linux Kernel is hosted in a private website (www.kernel.org). A repository is a web platform that offers hosting and special tools in order to give anyone the opportunity to share a project. Moreover, each repository gives the possibility to download the software, easily retrieve a lot of useful information, and also monitor the activity of the project.

In addition to the simple repositories, there are several repositories that aggregate many different projects from other repositories or private websites extracting data and collecting various measures (aka Repository of Repositories, abbreviated RoRs).

We are interested in repositories and RoRs because they are the main OSS distribution channels where users look for OSS products. As a consequence, just like with private project websites, repositories and RoRs are the place where trustworthiness factors must be clearly highlighted.

In Section 2 of wd5.2.1v3, we provide details about the available repository tools, the most important platforms, and the repositories and RoRs we analyzed in this deliverable.

Based on the business goals identified in WP5.1, some relevant OSS products and artefacts have been selected, so as to gather information on their general inherent characteristics and the way OSS products and artefacts are used by the software industry.

To carry out an adequate analysis of the projects, the set of projects has been selected by taking into account different types of software applications, generally considered stable and mature.

Specifically, the idea was to choose some OSS products with different characteristics as to:

- Types of programs and applications (from web servers to operating systems, from libraries to Customer Management Systems
- Programming languages
- Developer communities sizes
- Software engineering tools involved in the production process. This characteristic has been taken into account to evaluate whether the automation of the development process affects the trustworthiness of the products
- Age of the products.

The selection criteria aimed at:

- Including a reasonably small set of projects;

- Including at least two projects for every possible value of any possible attribute in each characteristic.

For instance, a specific attribute is the size of the development team. Four possible values were defined: 0 (which indicates an inactive project), no more than ten people, up to 50 people, more than 50 people. Therefore, we took care to include at least two projects for each of the four attribute values, avoiding inactive projects.

The project sets were defined in three phases:

- In the first phase, we selected 96 projects taking in account the projects considered the most important by the partners involved in WP5.2;
- Once the first set was defined, we restricted it to a subset of 32 projects as the most representative ones;

In the third phase, we selected a subset of 11 projects that we used to carry out a quick analysis. This quick analysis aimed at estimating how long a complete analysis would have taken. The complete analysis was later carried out on the set of 32 projects.

3 PROJECT ANALYSIS

3.1 Introduction

The analysis has been carried out in several iterative steps based on the list of factors identified and ranked in D1.5.1.1 [1].

These factors involve quality issues about the product, the process, and the communities of users and developers. Specifically, these factors are related to

- the development phase
- the community activity
- the internal qualities of the product
- the external qualities of the product.

The first round of our analysis was carried out by looking for factor-related information that was available by looking at the project sites.

We discovered that information about most trustworthiness factors is not immediately available and that, if available, it needs some specific measures to be evaluated.

Accordingly, whenever a factor cannot be directly assessed on the basis of the information on the web site, a new set of proxy-measures needs to be defined. Some factors can be assessed in a simple and direct manner, while others call for specific tools. The tool / factors mapping is defined in WP5.5.2 (Identification and construction of measurement tools) as anticipated in the Description of Work.

Both the new measures and the original ones, defined for OSS product trustworthiness, are listed. Each project has been evaluated according to the definitions of these measures. These measures directly refer to the possibility of evaluating the factor by looking into the project website. For example, taking into account the "Functional requirements satisfaction degree", the basic indicators assessment does not refer to the requirement satisfaction but to the possibility of evaluating the satisfaction of the requirements.

Some factors cannot be measured in an objective way, so the evaluation has to be done by using a subjective ordinal scale. For example, taking into account the *Feature List availability*, the difference between the availability of a poor free text description (where you can find all the features) and a comprehensive feature list will be measured with a subjective ordinal scale.

Other factors are not measurable, unless the developers provide essential information. For instance, the *number of downloads* cannot be evaluated in a reliable way if the community of developers does not explicitly publish this information.

3.2 Results

The main goal of the analysis is to obtain the information that is available through the websites of the projects.

Some factors have been analyzed, while analysis of other factors requires the development of ad-hoc tools.

In this section, we summarize the analysis we carried out on the subset of 32 projects. The analysis has the same structure of the Questionnaire presented in [1] with reference to the four categories: Economics, Development, Quality, and Customer. For each factor, we show the importance given by the users in [1] along with the results of the analysis. The complete analysis is available in Section 5.3 of wd5.2.1v3.

3.2.1 Economics: Economic Issues When Choosing OSS

In [1], Section 3.2.3 *"Economics: Economic Issues When Choosing OSS"* discusses two economic factors: ROI and TCO.
Here, we do not analyze these factors due to their subjectivity.

3.2.2 Developments: OSS Development Process

License Issues When Choosing OSS: fairly high importance

The majority of the projects use a GPL/LGPL license (48% GPL and 17% LGPL); seven projects use an Apache License while the remaining projects use other types of licenses.

As expected, this important factor is properly reported into the analyzed web portals.

The availability of tools for developing, modifying, or customizing OSS products: fairly high importance

We can see that more than 50% of the projects have special purpose-build tools and more than 75% of projects have some ad hoc configuration and management tools such as tools for managing users, or plug-ins and less than 35% of the projects use generic tools such as Text editors.

As expected, this important factor is properly reported into the analyzed web portals.

The availability of best practices for the OSS products: low importance

As expected, best practices were not available mostly in all projects (only one project out of 32 provides best practices on its website) while more than half of the projects provide some code examples listed on the website.

The availability of technical documentation / user manual: low importance.

As expected, almost every project has an up-to-date user documentation (manuals, getting started guides and installation guides) and there is a good level of interaction between users and developers by means of forums and mailing lists. Conversely, the situation of the technical documentation is not as good: approximately half of the projects provide technical documentation,

forums and mailing list, while only less than half of the projects have updated F.A.Q. and technical forums.

We point out that OSS users and developers consider product and design documentation as a major issue. We suggest that developers make technical documentation always available and up-to-date.

Environmental issues: low importance

Environmental issues describe software and hardware capabilities of each component of the environment. Due to the high subjectivity of this factor, we excluded it in our analysis.

The availability of training and guidelines: very low importance

Guidelines and training guidelines are mostly available and updated on the project websites. Only 7 projects out of 32 have out-of-date guidelines and one project does not provide any guideline.

Only 8 projects out of 32 provide professional training courses.

The mid-long term existence of a user community: high importance

Unfortunately, the dimension of the user community is not measurable unless explicit data is provided on the website. In our analysis, only 2 projects out of 32 report the community size. Not all websites clearly show data on the vitality of a community in correlation with the number of patches and releases. Some websites show only the number of patches/releases of the last 6 months, others only the total number of patches/releases, and some show both data. An interesting result is the availability of several community groups identified through different mailing lists (technical related, user related, translator related).

Despite our expectation, data related to the size and the vitality of the communities are not sufficiently highlighted in the considered web portals. We suggest that developers clearly show this information.

The mid-long term existence of a maintainer organization / sponsor: very low importance

The analysis carried out on this factor shows that the vast majority of the projects have several maintainers/sponsors. As reported in wd5.2.1v3, only 8 project out of 32 (25%) do not have a maintainer/sponsor or a supporting organization.

The short-term support: fairly high importance

As expected, the short term support is mostly assessable. As we can see, most of the projects publish their bug-trackers and provide professional services that can guarantee a short-term resolution of bugs.

The reputation of the OSS provider

This factor is not analyzed, due to its high subjectivity.

The distribution channel: negligible importance.
As expected, all the projects freely provide their source code via the Internet. The vast majority provides the source code, the binaries and access to the repository.

Few projects are available via CD/DVD or p2p (such as torrent, eMule).

The programming language uniformity: low importance

More than 60% of the projects use exactly one programming language but only half of these projects explain why they use one particular language or why they use multiple languages for their projects.

The existence of a sufficiently large community of users that can witness its quality: fairly high importance

As expected, most of the projects provide an official forum (20 out of 32), while the others have forums with a lot of activity (in some cases with more than 100.000 posts).

We suggest that developers always maintain active forums and vital communities.

The existence of benchmarks / test suites that witness for the quality of OSS: low importance

Most of the projects do not show if they use any test framework and test suites. 18% of projects try to encourage the community to contribute to their quality efforts and 41% show links to articles with the results of benchmark studies.

3.2.3 Quality

This information was collected in order to check the availability of the quality related factors that OSS users take into account when selecting OSS products.

The degree to which a OSS product satisfies / covers functional requirements: fundamental

Unexpectedly, the situation is quite negative: less than half of the projects provide a comprehensive list of supported functionalities and product samples (such as screenshots, static or dynamic demos, excerpts of code). The majority of the projects (19 out of 32) do not discuss functional requirements (or often the provided information given is incomplete). Only release notes are well shown (59% of projects).

We suggest that developers focus their attention on this fundamental factor, discussing and reporting how their products satisfy/cover functional requirements.

External quality – Performance: high importance

Unexpectedly, the majority of the projects do not provide any information (for example, by means of specific documentations, reports of performance tests, benchmarks).

We suggest that developers pay attention to this important factor, discussing and reporting the performance of their product.

External quality – Maintainability: fairly high importance.

Unexpectedly, the only measure that is easily retrievable from the analyzed web portals is the existence of maintenance releases. Other measures are almost never retrievable, while half of the projects report the usage of some coding standards.

We suggest that developers highlight this important factor in their web portals, discussing and reporting how their products are maintainable.

External quality – Portability: fairly high importance.

As expected, the analysis shows that more than 70% of the projects use a portable language (e.g. Java) but only 38% of the projects show their supported environments.

External quality – Reliability: very high importance.

Unexpectedly, almost no project reports its development status and only half of the projects use some coding standards and check it regularly.

We suggest that developers highlight this important factor better in their web portals, discussing and reporting data that demonstrates the degree of reliability of their products.

Internal Quality - Complexity: low importance

Information about product complexity is - when provided at all - mostly incomplete.

Internal Quality - Modularity: high importance.

In line with the desires of users and providers, more than 60% of the analyzed projects provide plug-ins or interfaces that increase the modularity of the project.

Internal Quality - usage of Standard Architecture: high importance

Unexpectedly, only half of the projects provide some architectural documentation and a description of the adopted standards. Moreover, only 14 out of 32 web portals describe the design patterns applied in the project.

We suggest that developers simplify the retrieval of this important factor in their web portals, increasing the information and the documentation related to architectural choices they used.

Standard compliance: high importance.

Unexpectedly, the possibility to assess the standard compliance reflects the already discussed quality factors. Two measures are in common with factor *"Internal Quality – Reliability"*: the use of standards during the coding phase, and the proper compliance to standardization. Another measure we considered for this factor is the availability of information about the implemented standards (e.g., HTTP 1.0, SQL 97…). Unfortunately, only half of the projects report data about the compliance of the project with available standards.

We suggest that developers point out this important factor in their web portals, discussing and reporting data about the used standards in order to improve the global comprehension of the overall project.

Self containedness: low importance.

More than 70% of projects can run out of the box without any other tool or library. As expected, some projects use third parties products but only half of them describe integration issues in their documentation.

Interoperability: very high importance.

As expected, most of the projects are equipped with information about interoperability issues (e.g., whether they communicate or not with other systems and if they provide plug-ins or interfaces).

Human interface language / localization: low importance.

Only 11 projects out of 32 provide localization support and support more than one language.

3.2.4 Customer

By customer, we mean the person that has requested a service, a system, a library, a tool, etc. We intend to stress out a purchaser role.
An analysis of the customer-related factors is impossible due to the subjectivity of this category.

4 THE OP²A MODEL

Assessing the quality of a web portal can help achieve the goals of different stakeholders. From the developer's point of view, the model provides guidelines for the definition of the website structure. OSS web masters may benefit from the website model used in the assessment, because it helps assess if all the product's contents are correctly organized and published in their portals: they can simply compute the maturity level of their web portal, and then, improve the "goodness" and "attractiveness" of the portal, if needed.

Based on the experience gained during the project analysis (see Section 3), we defined the Open source Portal and Product Assessment (OP²A), a model to help producers in creating websites for OSS projects.

OP²A is built upon trustworthiness factors identified in WP5.1 and the results of project analysis carried out in Section 4 and 5 of wd5.2.1v3.

OP²A has been defined with emphasis on simplicity and ease of use. To this end, we defined a checklist that OSS developers and web masters can use to determine the maturity level of their own OSS web portals. OP²A is thus a tool for self-assessment, rather than an instrument for formal assessment. The core of the checklist is reported in Appendix H of wd5.2.1v3.

The checklist is structured in five areas:

1. company information;
2. portal information;
3. reasons of assessment;
4. availability of information concerning trustworthiness factors;
5. portal usability information.

When using the checklist, the evaluator (the OSS web master or the OSS developer) first inserts general information about the company, about the portal under analysis and the reasons of assessment. Then, the evaluator goes through a sequence of entries that drive developers and web masters to identify whether contents and data related to the relevant trustworthiness factors are published in their OSS web portal.

Specifically, the core of the checklist is the evaluation of the project information availability (the fourth area of the checklist) in which the trustworthiness factors are considered and detailed in sub-factors. Trustworthiness factors are grouped into the following seven categories:

1. *Overview*: general description of the product, without dwelling too much on the details, as only an overview of the software is needed;
2. *Requirements*: disk usage, memory usage, supported operating system, etc.;
3. *License*: reference to the license, use conditions, and law conformance;
4. *Documentation*: user documentation, technical documentation, etc.;

5. *Downloads*: the number of downloads and related information;
6. *Quality reports*: Reliability, Maintainability, Performance, Product Usability, and Portability aspects are addressed;
7. *Community & Support*: the availability of various forms of support and the possible existence of a community around the project are investigated.

Every item of the information availability area is associated with a weight. Items corresponding to trustworthiness factors are weighted according to the average grade obtained in WP5.1. If a trustworthiness factor is evaluated through sub-factors, its value is equally divided among the sub-factors.

As an example, Figure 1 shows an excerpt of the checklist that refers to the "*License"* category. The interviewees of our survey assigned to factor "*Type of license*" an average grade of 6.45 and to factor "*Law conformance aspects*" an average grade of 6.89. In the checklist, we have three items: "*Law conformance aspects*," which is a factor, so it has the weight obtained through the survey, and "*Main license*" and "*Sub-licenses*", which are sub-factors of "*Type of license*" and thus get half of the weight that was obtained for factor "*Type of license*" in the survey.

The total value for the "*License*" category of the checklist is: 6.45+6.89 = 13.34.

Project Information Availability		**Overall Assessment**		
	3. License	**_____ / 13.34**		
		Presence		
		Y	**N**	**Weight**
	• Main license			**3.22**
	• Sub licenses (if applicable)			**3.22**
	• Law conformance (if applicable)			**6.89**

Figure 1 ***Excerpt of the checklist for the area "project information availability," category "License"***

The evaluator evaluates the availability of each type of information by ticking the box "*Y*" if the information is available, "*N*" otherwise. Some trustworthiness factors and sub-factors may be not applicable to the target portal: if a factor is not applicable, its weight is not meaningful to compute the final score of the portal. When this process is completed and all the entries have been checked, the evaluator simply sums the values of the information classified as available: the result is the actual total score of the portal. The weighted percentage of covered factors is equal to:

$$(Tot\_Portal\_Score / Tot\_Applicable\_Score) * 100.$$

Referring to our previous example, if the web portal under analysis provides only information about the *main license* used in the project and *sublicenses* and *law conformance* aspects are applicable but not published on the web portal, the final score for category "*License*" will be equal to 3.22.

The last area of the checklist details usability aspects of the web portal. Specifically, we check whether the aspects that affect the site usability have

been met. This part of the checklist is divided into ten subparts, one for each of Nielsen's usability heuristics [26]: (1) Visibility of system status; (2) Match between system and the real world; (3) User control and freedom; (4) Consistency and standards; (5) Error prevention; (6) Recognition rather than recall; (7) Flexibility and efficiency of use; (8) Esthetic and minimalist design; (9) Help users recognize, diagnose, and recover from errors; (10) Help and documentation.

In this part of the checklist, we are mainly interested in summarizing well-agreed guidelines [26][27] for creating high quality web sites, to simplify the work of OSS developers that should keep in mind these aspects when designing the web portal.

4.1 The Assessment Levels

The OP^{2}A assessment model provides three assessment levels organized in increasing levels of web portal maturity: Bronze, Silver, and Gold. In addition, there is an "unclassified" level. Each level represents the score computed for the trustworthiness factors listed in the checklist, and a maturity level is achieved only when a certain score is obtained. We defined three assessment levels in compliance with the levels and the intervals defined in the OpenBRR maturity model [28], an assessment methodology for rating OSS. OpenBRR is characterized by five levels with the following maturity intervals (computed as the cumulative sum of the single weights divided by the maximum score in percentage):

1. less than 65%, maturity level = 1 (unacceptable)
2. 65% - 80%, maturity level = 2 (bad)
3. 80% - 90%, maturity level = 3 (acceptable)
4. 90% - 96%, maturity level = 4 (very good)
5. greater than 96%, maturity level = 5 (excellent)

In OP^{2}A the intervals are defined as in OpenBRR, except that level 4 and level 5 are merged into the single maturity level "Gold," as shown in Figure 2. Of course, this is just a preliminary definition of levels and intervals. We are conducting several experiments of web portal evaluations to understand whether these levels are too strict and we are studying if the literature suggests different levels that better fit our model.

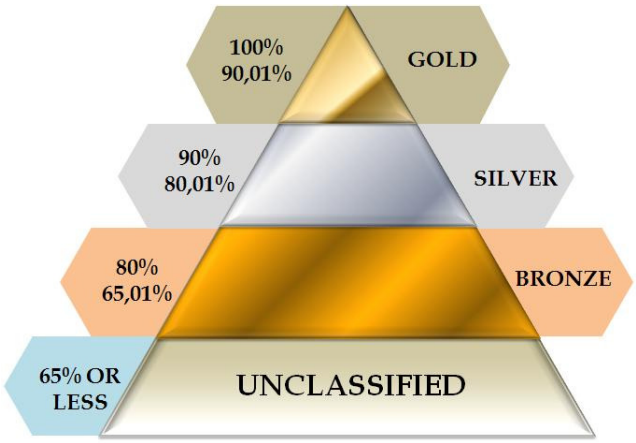

Figure 2 ***OP²A maturity intervals***

4.2 Model Validation

In this section, we show the application of the OP²A model to a well-known OSS product website: the Apache Tomcat website (http://tomcat.apache.org) The goal of this activity is twofold: (1) showing the simplicity of OP²A and the real support provided by the checklist; (2) showing how it is possible to actually improve the quality of the web portal by refactoring it according to the indications provided by the analysis. Here, we propose a refactoring of the portal to improve its quality and visibility.

4.2.1 Applying OP²A to the Apache Tomcat Website

Apache Tomcat is an open source servlet container developed by the Apache Software Foundation. It provides a platform for running Web applications developed in Java. We decided to take Apache Tomcat as an example because of its notoriety and diffusion.

Figure 3 shows the Apache Tomcat website at the time of writing. A quick look at the home page shows a very long menu on the left, with several links grouped by topic. We notice a lack of the general product description. On the home page the overview says: "*Apache Tomcat is an open source software implementation of the Java Servlet and JavaServer Pages (jsp) technologies...*" but an inexperienced user or developer may not understand if Apache Tomcat is just a utility or a set of libraries for Java Servlet and jsp or something else able to manage jsp.

The download area is well structured, but it contains too much information, while users usually want to be presented with a link for downloading the latest stable version of the product. Nevertheless, we scored this area as good in our checklist, because it provides all the information required by OSS final users and developers.

Other areas like "problems?", "get involved," and "misc" fulfill several entries of the checklist. More than 90% of the information is correctly shown on the website for the categories: *Overview*, *License*, *Documentation* and *Downloads*. Conversely, we noticed that information about *Requirements* and *Quality Reports* – such as reliability, maintainability, performance and product usability

– are marginally discussed on the Apache Tomcat website. The current version of the website covers 60% of the category *Community&Support*. In conclusion, as shown in Appendix I of wd5.2.1v3, the Apache Tomcat website earned a *Tot_Apache_Score* = 98.19 over a theoretical *Tot_Applicable_Score* = 147.50. Thus, the ratio is 66.6% and the Apache Tomcat website is classified at the "Bronze" level.

In Appendix I above mentioned we show the OP^{2}A checklist filled in with the Apache Tomcat results.

4.3 A Proposal for Refactoring the Apache Tomcat Website

As described in Section 6.2.1, the Apache Tomcat website did not obtain the best assessment level in OP^{2}A. In this section, we illustrate a proposal for refactoring the Apache Tomcat portal to the OP^{2}A gold level, so that OSS users and also developers will be able to quickly find all required information, and the probability of adoption or reuse of the Apache Tomcat product will increase.

To get the gold OP^{2}A level, we need to consider all the factors included in the OP^{2}A checklist. In Fig. 4, we propose a new menu structure for the home page. This menu is shorter than the original one and enables users to reach the most important information directly from the home page. The idea of grouping all the information derives from the Nielsen's views [26].

To validate the quality of the refactored version of the website, we asked eight Master's students, who had never accessed the Apache Tomcat portal before, to preliminary surf the original web portal and the refactored one for 10 minutes and rank their perception of the quality of the website, in a scale from 1 (poor quality) to 4 (very good quality). Then we asked our sample to fill out the OP^{2}A checklist. Four students evaluated the original Apache tomcat website, and the four other evaluated the refactored version. We were interested in observing the ease of the information retrieval process, the time taken to fill out the checklist, and the perceived quality of the two versions of the website. In Table 1, we show the users' perceived quality of the Apache Tomcat website in column "quality perception," and the time taken by our testers for analyzing the original Apache Tomcat website and the refactored version in column "overall time." Based on these results, we can state that the refactoring actually improved the quality of the portal. It is interesting to observe that the quality perception is actually increased from an average value of 2 to an average value of 3.5 after the refactoring activity. These values are in line with the maturity level computed by OP^{2}A. Since the small number of people involved does not support statistical significance, we are conducting additional experiments with a larger number of students in order to validate these preliminary results.

Original Website		Refactored Version	
overall time	quality perception	overall time	quality perception
1h 30 min	2	26 min	3
1h 18 min	2	30 min	4
1h 00 min	2	30 min	4
1h 10 min	2	22 min	3
1h 14 min (avg)	2 (avg)	27 min (avg)	3.5 (avg)

Table 1 Time necessary to complete the OP²A checklist and users' perceived quality

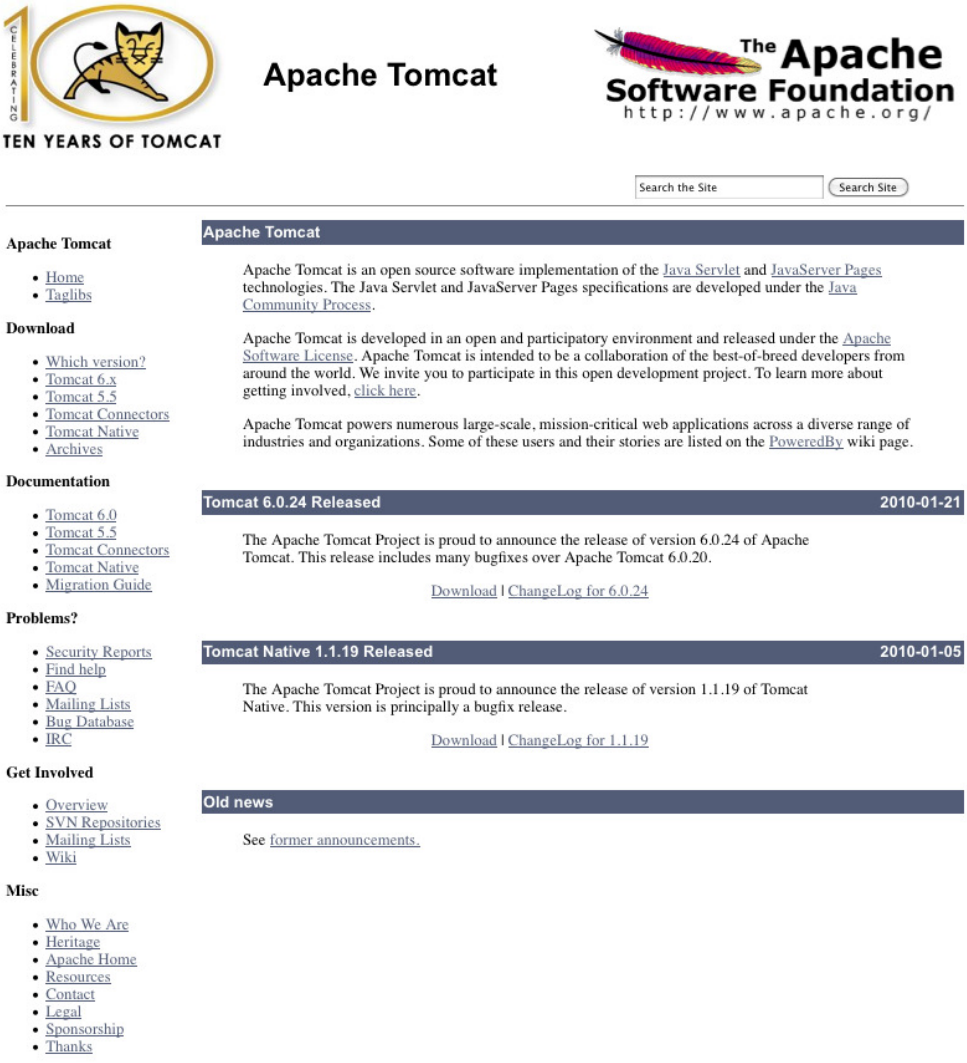

Figure 3. *Actual Apache Tomcat home page*

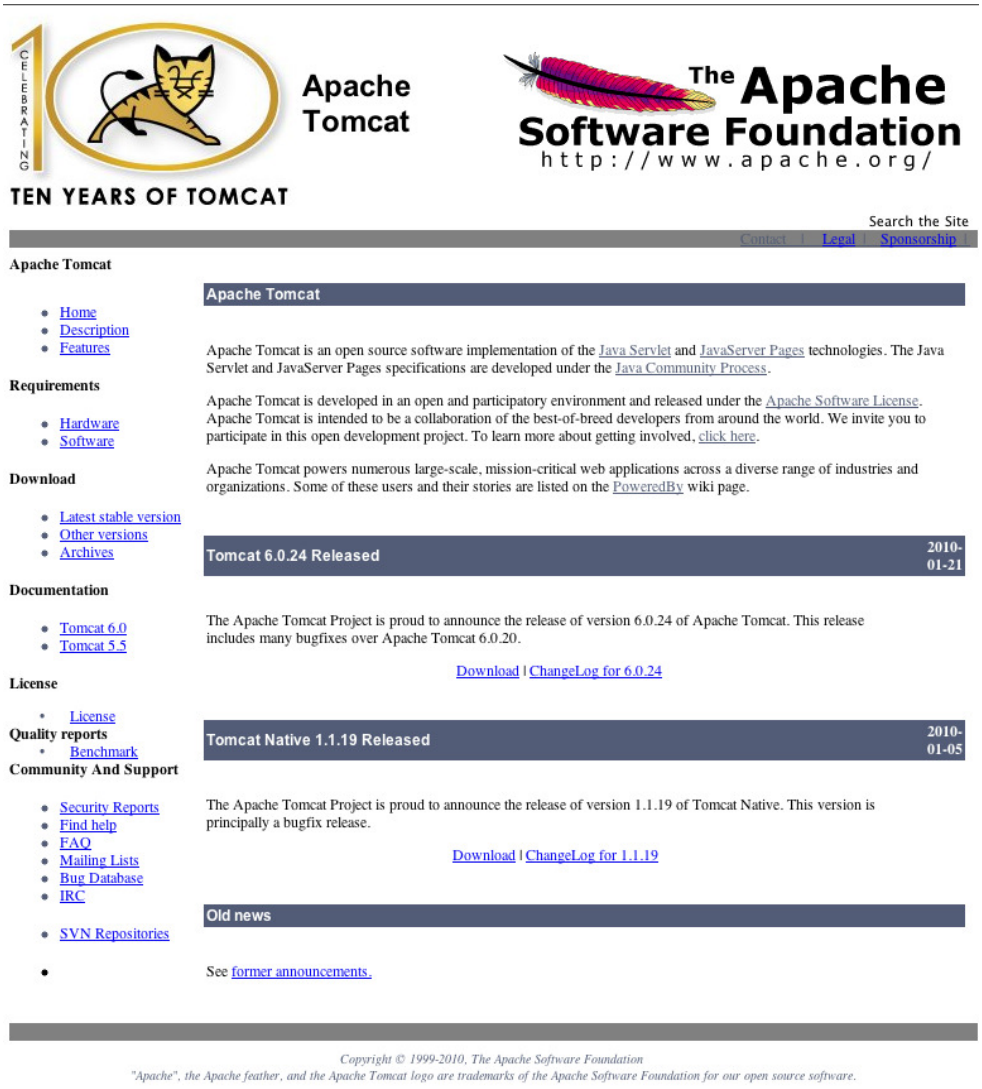

Figure 4. ***The proposed refactoring for the Apache Tomcat website***

5 CONCLUSIONS

The investigation presented in this document focuses on understanding the trustworthiness characteristics of OSS repositories.

Commonalities, differences in their characteristics, and usage are identified and analyzed to prepare the ground for other work packages, especially WP5.3, that deals with the definition of relevant factors for trustworthiness.

The achievements of these results have been obtained through the analysis of the main OSS repositories and of a fairly extensive set of OSS products and artefacts. For this type of analysis, we have therefore gathered information on the generally available characteristics of OSS projects, thus allowing the evaluation of their trustworthiness.

The analysis of OSS repositories shows the commonly available services and highlights commonalities and differences between repositories.

The second part of this work was carried out taking into account the factors generally perceived as the most important by the users (see [1]). Initially we check whether the factors was easily available on the project website. This analysis shows that the vast majority of the factors need some measures to be specified. Then a checklist of measures has been specified in order to assess each factor for each project. In this report, we also provide a set of guidelines and recommendations that developers should follow when developing and releasing OSS products.

The results extracted from our analysis point out interesting issues concerning the availability of factors that are believed to affect the perceived trustworthiness of the project.

Some of these results confirm and give evidence to support previous beliefs, while others are surprising and unexpected.

Quite noticeably, most of the expected indications involve technical issues. Most factors are assessed directly or indirectly via the identified measures, while others need some tools to be developed. Taking into account the development related factors, there are some problems in retrieving the majority of the factors. Only around half of the projects have technical documentation, forums and mailing list available, while only less than half of the projects have updated F.A.Q. (Frequently Asked Questions) and technical forum. The same problems appear when we have to check for the availability of best practices and the programming language uniformity. Some factors are often (but not always) available: the availability of training, the availability of tools for modifying, customizing OSS products, and the distribution channel. Taking in account community activity, the situation is fairly negative. The size of user communities is not measurable unless the websites do not provide the number of participants. In the set of projects that has been analyzed, only 2 projects out of 32 provide information over the size of their community, and not all projects clearly show patches and releases; some projects inform only of the number of patches/releases in the last 6 months, others only the of total number and finally a last group shows both.

An interesting result is the availability of several community groups identified via different mailing lists (technical related, user related, translator related…). Unexpectedly, the situation about documentation is quite good from the user's side: almost every project has updated documentation (user manuals, getting started guide and installation guides) and there is a good level of communication between users and developers through forums and mailing lists. Considering product quality, there are no factors completely measurable, in some case because tools should be developed for this goal, in others because of the lack of information provided in the project websites. Almost no projects provide any information about their performances, maintainability, reliability and complexity. On the other hand, half of the projects show the usage of standard architectures, the availability of interfaces and plug-ins and its interoperability, the possibility to run without any other tools or library and their standard compliance. No project gives any information on code complexity. Some factors are not analyzed because of their subjectivity: all economic and customer related factors, the environmental issue and the reputation of the vendor.

The main contribution of this report is an exhaustive analysis of a set of representative projects useful for the next tasks of A5 "*Trustworthy Results*". Another important output of this work will be a list of recommendations that will be given as input to the open source community in order to improve their product quality that includes the information that should be available on the project websites.

Finally, we defined the OP^{2}A assessment model, which contains a checklist that OSS developers and web masters can use to design their web portals so that all the contents that are expected by OSS users are actually provided. We exemplified the use of OP^{2}A through its application to the Apache Tomcat website, to show the simplicity and the actual potentialities of the model and of the checklist. Preliminary results suggest that the model can be effectively used to improve the quality of OSS web portals.

The proposed evaluation model can be applied also to the websites of closed source products. Of course, a few trustworthiness factors –namely those addressing source code qualities, like maintainability– are not applicable in the case of closed source software.

6 REFERENCES

[1] V. del Bianco, M. Chinosi, L. Lavazza, S. Morasca, D. Taibi: "How European software industry perceives OSS trustworthiness and what are the specific criteria to establish trust in OSS", October 2007, available on-line at http://qualipso.semanticdesktop.org/xwiki/bin/view/Wiki/WP51.

[2] Gartner OSS Symposium - December 2006.

[3] T.J. McCabe, "A complexity measure", IEEE Transactions on Software Engineering, Dec. 1976.

[4] In Internet. URL: http://news.netcraft.com/archives/web_server_survey.html. Last visit: August 2008

[5] In Internet. URL: http://www.apache.org. Last visit: August 2008

[6] In Internet. URL: http://www.codeplex.com. Last visit: August 2008

[7] In Internet. URL: http://www.codehaus.org. Last visit: August 2008

[8] In Internet. URL: http://www.objectweb.org. Last visit: August 2008

[9] In Internet. URL: http://www.rubyforge. Last visit: August 2008

[10] In Internet. URL: http://www.savannah.gnu.org. Last visit: August 2008

[11] In Internet. URL: http://www.sourceforge.net. Last visit: August 2008

[12] In Internet. URL: http://ossmole.sourceforge.net. Last visit: August 2008

[13] In Internet. URL: http://www.gforge.org. Last visit: August 2008

[14] In Internet. URL: http://dev.libresource.org/. Last visit: August 2008

[15] In Internet. URL: https://gna.org/projects/savane. Last visit: August 2008

[16] In Internet. URL: http://trac.edgewall.org/. Last visit: September 2008

[17] In Internet. URL: http://www.berlios.de. Last visit: September 2008

[18] In Internet. URL: http://freshmeat.net. Last visit: September 2008

[19] In Internet. URL: http://directory.fsf.org. Last visit: September 2008

[20] In Internet. URL: http://freshmeat.net. Last visit: September 2008

[21] In Internet. URL: http://www.ohloh.net. Last visit: September 2008

[22] In Internet. URL: http://www.sourcekibitzer.org. Last visit: September 2008

[23] In Internet. URL: http://swik.net. Last visit: September 2008

[24] In Internet. URL: http://www.fortifysoftware.com/. Last visit: September 2008

[25] QualiPSo Annex I - "Description of Work" M13-M30

[26] Nielsen, J., "User Interface Directions for the Web", Communications of the ACM, Vol. 42, No. 1, 1999, pp. 65-72.

[27] Nielsen, J., Norman, D.: "Usability On The Web Isn't A Luxury", *InformationWeek,* February 2000.

[28] O'Reilly CodeZoo SpikeSource community initiative, being sponsored by Carnegie Mellon West Center for Open Source Investigation and Intel. "Business Readiness Rating for Open Source – A Proposed Open

Standard to Facilitate Assessment and Adoption of Open Source Software", August 2005.