OP2A
Assessing the Quality of the Portal of Open Source Software Products

Gabriele Basilico, Luigi Lavazza, Sandro Morasca, Davide Taibi, Davide Tosi
Università degli Studi dell'Insubria
{gabriele.basilico, luigi.lavazza, sandro.morasca, davide.taibi, davide.tosi}@uninsubria.it

Keywords: Open Source Software, Trustworthiness, Quality Perception, Certification/Assessment Models.

Abstract: Open Source Software (OSS) communities do not often invest in marketing strategies to promote their products in a competitive way. Even the home pages of the web portals of well-known OSS products show technicalities and details that are not relevant for a fast and effective evaluation of the product's qualities. So, final users and even developers, who are interested in evaluating and potentially adopting an OSS product, are often negatively impressed by the quality perception they have from the web portal of the product and turn to proprietary software solutions or fail to adopt OSS that may be useful in their activities.

In this paper, we define an evaluation model and we derive a checklist that OSS developers and web masters can use to design their web portals with all the contents that are expected to be of interest for OSS final users. We exemplify the use of the model by applying it to the Apache Tomcat web portal and we apply the model to 22 well-known OSS portals.

1 INTRODUCTION

The usage of Open Source Software (OSS) has been continuously increasing in the last few years, mostly because of the success of a number of well-known projects.

However, the diffusion of OSS products is still limited if compared to the diffusion of Closed Source Software products. There is still reluctance to massive adoption of OSS mainly due to two reasons: (1) lack of trust, as final users are often skeptical in trusting and adopting software products that are typically developed for free by communities of volunteer developers that are not supported by large business companies; (2) lack of marketing strategies, as OSS developers often do not pay attention to marketing, commercial and advertising aspects because these activities require a huge amount of effort and are not very gratifying. OSS developers are more focused on and interested in developing competitive software products than creating a commercial network that can support the diffusion of their products. Thus, OSS products may not have the success and the recognition that they should deserve.

Instead, as a mark of quality, commercial software and software producers may claim adherence to well-known standards, such as ISO9001 (ISO, 2008). Such product and process certifications require detailed documentation and clearly defined organizational responsibilities, which are likely to exist only for an established organization with a solid and clear infrastructure. Such an accreditation is not easy to obtain for OSS produced by globally spread individuals or virtual teams who often operate without much infrastructure and / or a formal environment of tools.

The websites and web portals of OSS products may suffer from similar problems, as they are created by non professional web masters who, on the one hand, tend to focus on technicalities that are not relevant for the evaluation of the OSS product from the point of view of the end-user, and, on the other hand, often do not provide in a systematic and exhaustive way the technical information needed by other developers that intend to modify the code or to incorporate it into their products.

Websites and web portals are very important for creating the initial quality perception that end-users or other developers have about an OSS product. A website may be viewed as a shop window: if the window is ordered, clean and well organized customers will probably go inside the shop to either have a look or buy a product. Conversely, if the window is dusty and messy, buyers will not enter the store and they will turn to another store. This may seem an obvious consideration, but OSS portals often do not provide the contents that are most relevant to the end-users (Lavazza et al., 2008), or, if they do, they provide this information in hidden sections of the website, thus not favouring usability (Nielsen, 1999). This may have a strong impact on the diffusion of OSS products.

In this paper, we introduce OP2A (Open source Product Portal Assessment), a model for evaluating the quality of web portals that store OSS products. OP2A can be used as the starting point for objectively certifying the quality of OSS portals. The model is built upon the results of a survey (Del Bianco et al., 2008) – conducted in the context of the European project QualiPSo (QualiPSo, 2011) – carried out to (1) identify the factors most commonly used to assess the trustworthiness and the quality of an OSS product, and (2) understand the reasons and motivations that lead software users and developers to adopt or reject OSS products. The model can be used by OSS developers to assess and improve the quality of their own web portals in order to present their products clearly, and minimize the effort required for presenting and promoting the OSS product in a competitive manner. OP2A takes into account a number of factors that are considered very important for the trustworthiness of an OSS product and describes the way this information should be presented to users that access the web portal of the product. OP2A is based on a checklist that summarizes the factors and simplifies the computation of the site maturity score. The checklist can be used by OSS developers to evaluate the maturity of their web portals and identify the maintenance actions required to meet attractiveness, clarity, and completeness requirements. We applied the assessment model to a real-life web portal (the Apache Tomcat portal) to show (1) the limitations of this portal, (2) how to use the checklist, and (3) how our model can actually drive the improvement of the portal. We also apply OP2A to 22 well-known OSS portals to assess the quality level of these famous projects.

The paper is structured as follows. Section 2 introduces the OP2A assessment model and the related checklist. Section 3 presents the application of the model to the Apache Tomcat web portal and to the 22 OSS portals. Section 4 describes related works in the field of web quality and usability. We conclude and sketch future work in Section 5.

2 THE ASSESSMENT OF OSS WEB PORTALS

In this section, we detail the OP2A assessment model we derived from the results of our survey.

2.1 Which Factors Influence the Quality Perception of OSS Products

We conducted a survey (Del Bianco et al., 2008) in the context of QualiPSo (QualiPSo, 2011) to find out which factors are most commonly used by developers and end-users to assess the trustworthiness of an OSS product. Our goal was to understand the reasons and motivations that lead software users and developers to adopt or reject existing OSS products, and, symmetrically, software developers to develop OSS. We called these factors "trustworthiness factors". Specifically, we focus on the trustworthiness of OSS, since OSS users and developers will not adopt a specific OSS product unless they can trust it. On the other hand, OSS developers need to promote the trustworthiness of their products, so that they may be more appealing to end-users and other developers that want to integrate existing OSS products in their software solutions or build on top of them.

Our survey showed that the most important factor is the satisfaction of functional requirements, followed by reliability and maintainability. The complete ranking of trustworthiness factors is reported in (Del Bianco et al., 2008). Each factor was rated by the interviewees on a 0 to 10 scale, with value 0 meaning "not important at all" and value 10 meaning "of fundamental importance." For each factor, we computed the mean value of the scores assigned by the 151 respondents: the mean scores of functional requirements satisfaction, reliability and maintainability are 8.89, 8.19 and 7.85, respectively. We computed the means, even though the measurement scale of used in the responses is ordinal, strictly speaking, because they are quite representative, for our purpose.

We used the results of this survey, specifically the trustworthiness factors and the relevance score they obtained in the survey, to derive our OP2A.

2.2 The OP2A Assessment Model

Certifying the quality of a web portal can help achieve the goals of different stakeholders. From the developer's point of view, the assessment provides guidelines for the definition of the website structure.

Certified websites speed up the assessment of new OSS products and guarantee the availability of all the needed information for both OSS users and developers that may need to reuse OSS source code. 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.

OP2A is built upon two sources of data: the trustworthiness factors highlighted in (Del Bianco et al., 2008) and the literature that describes well-known usability and accessibility rules for developing websites and web portals (Nielsen and Norman, 2000). 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 certifications. The core of the checklist is reported in Appendix and in (OP2A, 2011).

The checklist is structured in five areas: company information; portal information; reasons of assessment; availability of information concerning trustworthiness factors; portal usability information. So, when using the checklist, the evaluator 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 trustworthiness factors are considered and detailed in subfactors. In turn, 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 the survey (Del Bianco et al., 2008). If a trustworthiness factor is evaluated through subfactors, its value is equally divided among the subfactors.

As an example, Fig. 1 shows an excerpt of the checklist that refers to the "*License*" category. The interviewees of our survey (Del Bianco et al., 2008) 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. For example, if the sub-factor "*Law conformance*" is not applicable, the total value for the "*License*" category is 6.45 instead of 13.34.

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*

where *Tot_Portal_Score* is the sum of all the sub-factors that received the Y evaluation (or equally the sum of the seven categories), while *Tot_Applicable_Score* is the sum of all the sub-factors that are applicable for the portal under assessment. *Tot_Portal_Score* is a valid indicator about the quality of the target portal and can be used by final users and web developers to understand the quality level of the portal. A high value of *Tot_Portal_Score* suggests that the quality of the portal is good, while a low value of *Tot_Portal_Score* indicates that the portal needs refactoring. The checklist suggests how to improve the quality of the portal.

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 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 (Nielsen, 1999): (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) Esthetical and minimalist design; (9) Help users recognize, diagnose, and recover from errors; (10) Help and documentation.

For space reasons, we do not report the details of the heuristics here. See (OP2A, 2011) for more details. In this part of the checklist, we are mainly interested in summarizing well-agreed guidelines (Nielsen and Norman, 2000)(Nielsen, 1999) 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.

3 VALIDATION OF OP2A

In this section, we sample the application of the OP2A model to the OSS product website: the Apache Tomcat website (Apache, 2011), and we apply also the model to 22 well-known OSS portals. The goal of this activity is threefold: (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; (3) providing an evaluation of the quality of well-known OSS products. The Appendix reports on the evaluation results of the Apache Tomcat website. Here, we also propose a refactoring of the portal to improve its quality and visibility.

3.1 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.

The URL http://tomcat.apache.org/ 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" fulfil 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 the Appendix, the Apache Tomcat website earned a *Tot_Apache_Score* = 98.19 over a theoretical *Tot_Applicable_Score* = 147.50. (66.6%).

3.2 A Proposal for Refactoring the Apache Tomcat Website

As described in Section 3.1, the Apache Tomcat website gained a *Tot_Apache_Score* of 98.19 thus indicating that more than 30% of trustworthiness factors have not been taken into account when designing the Apache Tomcat website. In this section, we make a proposal for refactoring the Apache Tomcat portal to improve the quality of the website and to increase the *Tot_Apache_Score*, so that OSS users and also developers will be able to quickly find all required information, and the probability of adoption/reuse the Apache Tomcat product will increase.

Table 1: Evaluation data for the assessment of the Apache Tomcat website

Original Website			Refactored Version	
overall time	quality perception	OP2A score	overall time	quality perception
1h30m	2	98.19	26m	3
1h18m	2	95.65	30m	4
1h00m	2	96.99	30m	4
1h15m	2	94.59	25m	4
1h10m	2	97.59	22m	3
1h14m (avg)	2 (avg)	1.46 (st dev)	27m (avg)	3.6 (avg)

To this end, we need to consider all the factors included in the OP2A checklist. In Fig. 2, 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 comes out by looking at the views of Nielsen (Nielsen, 1999).

To validate the quality of the refactored version of the website, we asked ten master 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 OP2A checklist. Five students evaluated the original Apache tomcat website, and the five 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, the perceived quality of the two versions of the website, and the subjectivity degree of the checklist. In Table 1, we show the time taken by our testers for analyzing the original Apache Tomcat website in column <<*overall time*>>, the users' perceived quality of the Apache Tomcat website in column <<*quality perception*>>, and the total score achieved by applying the checklist on the website in column <<*OP2A score*>>. The other two columns show the overall time, and the quality perception for the refactored version of the website. 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 3.6 after the refactoring activity. These values are in line with the maturity level computed by OP2A. Moreover, the standard deviation (equal to 1.46), computed over the five OP2A scores, suggests the low degree of subjectivity of the proposed checklist.

For our experiment, we selected ten students following (Nielsen and Landaur, 1993). In any case, we are conducting additional experiments with a larger number of students in order to strongly validate these preliminary results.

3.3 Applying OP2A to 22 OSS portals

We also applied the OP2A checklist to 22 additional portals of Java OSS projects. The set of portals has been selected by taking into account different types of software products, generally considered well-known, stable and mature. An author of this paper conducted the assessment of the 22 portals. The purpose of this experimentation is not to shows the feasibility of OP2A, but to provide a preliminary assessment of the quality of these portals. Table 2 summarizes the results of the assessment. Column <<*OP2A Score*>> reports the total score achieved by applying the checklist on each portal, and the ratio (as percentage) between the obtained score and the total achievable score. The experimentation shows that the quality of the analyzed portals is not adequate in general, thus needing a strong refactor of the portal to achieve an acceptable level of quality. In (OP2A, 2011) it is possible to find the details of this experimentation.

4 RELATED WORK

Before committing to using a software product, people want to collect information about the product, in order to be able to evaluate its trustworthiness. Usually, during the selection of software, users and developers collect information about the products from the official websites. This is especially true for OSS products, which are typically distributed exclusively via the web.

The type of the information commonly used by the users when they evaluate OSS projects has been investigated in the last few years, and several OSS evaluation methods have been proposed. Their aim is to help potential adopters to understand the characteristics of the available products, and to evaluate the pros and cons of its adoption. Some of the most known OSS evaluation models are: OpenBRR (Wasserman et al., 2005), QSOS (Atos, 2010), OSMM (Golden, 2005) and OpenBQR (Taibi et al., 2007). OSMM is an open standard aimed at facilitating the adoption of OSS based on the evaluation of some maturity aspect of OSS like documentation, provided support, training availability and third parties integration possibilities. QSOS extends the information to be evaluated by adding new quality areas like the documentation quality and the developer community. Finally, OpenBRR and OpenBQR address additional quality aspects and try to ease the evaluation process.

Table 2: Assessment of 22 portals of well-known OSS products.

Analyzed Portal	OP2A Score
Checkstyle	93.34 (63.32%)
Eclipse	105.4 (71.50%)
Findbugs	77.79 (52.77%)
Hibernate	82.08 (55.68%)
HttpUnit	77.44 (52.53%)
Jackarta Commons	76.92 (52.18%)
Jaspert Report	102.64 (69.62%)
JBoss	111.57 (75.68%)
JFreeChart	81.43 (55.24%)
JMeter	75.20 (51.01%)
Log4J	79.52 (53.94%)
PMD	78.16 (53.02%)
Saxon	71.28 (48.35%)
Spring Framework	89.35 (60.61%)
Struts	103.52 (70.22%)
Tapestry	73.00 (49.52%)
TPTP	82.08 (55.68%)
Velocity	72.76 (49.36%)
Weka	82.08 (55.68%)
Xalan	71.60 (48.57%)
Xerces	110.56 (75.00%)
Servicemix	71.28 (48.35%)

The evaluation process of all these methods is mainly organized into an evaluation step and a scoring step. The evaluation step aims at collecting the relevant information concerning the products from the OSS website. In this phase, the goal is to create an "identity card" for every product with general information, functional and technical specifications, etc. The quality aspects of the selected products are evaluated and a score is assigned according to the evaluation guidelines provided by each method. In the scoring phase, the final score is computed by summing all the scores calculated in the previous step.

In (Golden, 2005) a method for OSS quality certification is proposed. Like the other evaluation methods, it is based on the evaluation of a common set of information but differs whilst the process is based on ISO/IEC 9126. The biggest problem of the evaluation model is the definition of the information to be evaluated. This information has been defined according to experience and the literature, but they are often unavailable and not useful for most users. In order to reduce the set of information to be evaluated, we carried out a survey (Del Bianco et al., 2010) to study the users' perception of trustworthiness and a number of other qualities of OSS products. We selected 22 Java and 22 C++ products, and we studied their popularity, the influence of the implementation language on trustworthiness, and whether OSS products are rated better than Closed Source Software products.

Another important research field for this paper is the website certification. In 2001, a certification schema proposal for Italian Public Administration website quality has been defined (Minelle et al., 2001). This certification model is based on a set of information that Public Administration websites must publish on their own website. The set of information has been defined by investigating the quality aspect –e.g., usability and accessibility– of 30 Italian Public Administration websites.

Since 1994 the World Wide Web Consortium (W3C), defined several standards, guidelines and protocols that ensure the long-term Web growth and accessibility to everybody, whatever their hardware, software, language, culture, location, or physical or mental ability.

In 2008, W3C released the second version of the "Web Content Accessibility Guideline", aimed at making Web contents more accessible (W3C, 2008).

Usability is defined by the International Organisation for Standardisation (ISO) as: "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use“. Some usability studies show problems in Sourceforge (Sourceforge, 2011): Arnesen et al. (Arnesen et al., 2000) showed several problems, mainly concerning the link structure and the information organization. Another study (Pike et al., 2003) identified usability problems both with Sourceforge and with the Free Software Foundation (FSF, 2011) website by means of eye tracker techniques (Jacob, 1991).

Currently, the vast majority of OSS websites does not provide the information needed by end-users. OP2A aims at ensuring both the availability of information and its accessibility.

5 CONCLUSIONS

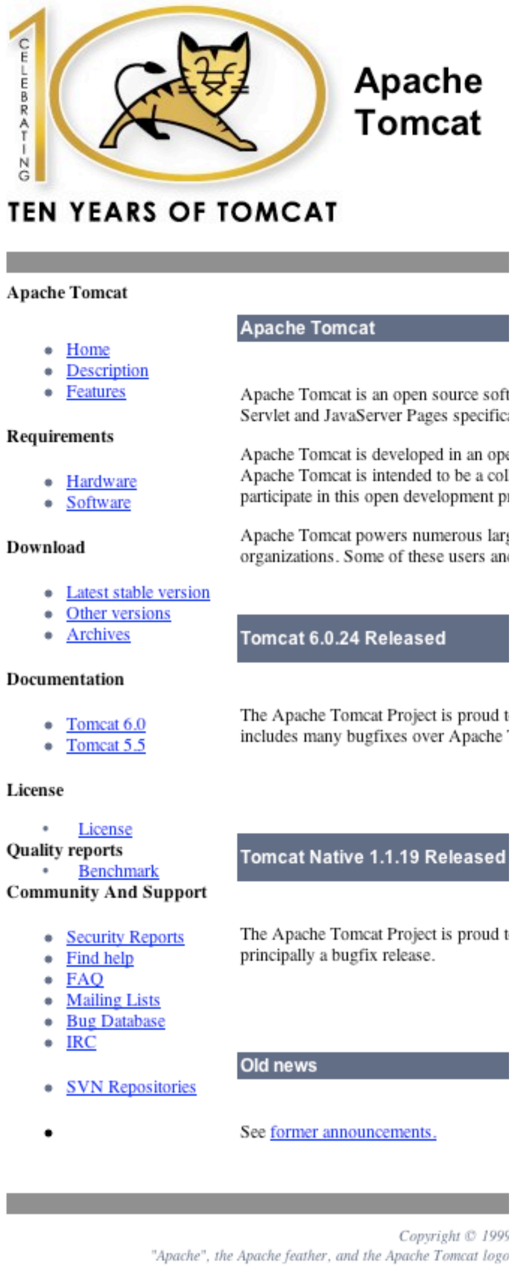
A survey that we conducted in the context of the QualiPSo European project led to the identification of the trustworthiness factors that impact on the choices of users in adopting OSS products. On such basis, we defined the OP2A assessment model, which contains a checklist that OSS developers and web 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 OP2A through its application to the Apache Tomcat website, to show the simplicity and the actual potentialities of the model and of the checklist, and we evaluated the quality of 22 OSS portals. 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) are not applicable in the case of closed source software.

We are conducting additional experiments and we are applying OP2A to other OSS web portals to understand whether: (1) the weight of subfactors should be refined, for example asking OSS developers and users to weight also subfactors; (2) the checklist needs refinements, for example detailing/adding/removing subfactors from/to the checklist; (3) using degrees of presence of factors, instead of yes/no values;

ACKNOWLEDGMENTS

The research presented in this paper has been partially funded by the IST project Qualipso (http://www.qualipso.eu/), sponsored by the EU in the 6th FP (IST-034763); the FIRB project ARTDECO, sponsored by the Italian Ministry of Education and University; and the projects “Elementi metodologici per la specifica, la misura e lo sviluppo di sistemi software basati su modelli” and “La qualità nello sviluppo software,” funded by the Università degli Studi dell’Insubria.



Apache Tomcat

- Home
- Description
- Features

Requirements

- Hardware
- Software

Download

- Latest stable version
- Other versions
- Archives

Documentation

- Tomcat 6.0
- Tomcat 5.5

License

- License

Quality reports

- Benchmark

Community And Support

- Security Reports
- Find help
- FAQ
- Mailing Lists
- Bug Database
- IRC
- SVN Repositories

Apache Tomcat

Apache Tomcat is an open source soft
Servlet and JavaServer Pages specific

Apache Tomcat is developed in an op
Apache Tomcat is intended to be a col
participate in this open development p

Apache Tomcat powers numerous lar
organizations. Some of these users an

Tomcat 6.0.24 Released

The Apache Tomcat Project is proud t
includes many bugfixes over Apache

Tomcat Native 1.1.19 Released

The Apache Tomcat Project is proud t
principally a bugfix release.

Old news

See former announcements.

Copyright © 199
"Apache", the Apache feather, and the Apache Tomcat logo

Figure 2: Apache Tomcat website Refactoring.

REFERENCES

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.

Duijnhouwer F-W., Widdows C., "Expert Letter Open Source Maturity Model". Capgemini, 2003.

Atos Origin, "Method for Qualification and Selection of Open Source Software (QSOS)". Web published: http://www.qsos.org (Last visited: Jan., 2011).

Taibi D., Lavazza L., Morasca S., "OpenBQR: a framework for the assessment of OSS", Proceedings of the Third International Conference on Open Source Systems (OSS), Limerick, Ireland, June 2007. Springer, pp. 173–186.

Del Bianco V., Chinosi M, Lavazza L., Morasca S., Taibi D. "How European software industry perceives OSS trustworthiness and what are the specific criteria to establish trust in OSS". Deliverable A5.D.1.5.1 – QualiPSo project – October 2008. Web published: www.qualipso.eu/node/45. (Last visited: Jan., 2011).

Lavazza L., Morasca S., Taibi D., Tosi D., "Analysis of relevant open source projects". Deliverable A5.D.1.5.2 – QualiPSo project – October 2008. Web published: www.qualipso.eu/node/84. (Last visited: Jan., 2011).

Correia J.P., Visser J., "Certification of technical Quality of Software". Proceedings of the OpenCert Workshop, Italy, 2008.

Mockus A., Fielding R. and Herbsleb. "A case study of OS software development: the Apache server". Proceedings of the 22nd International Conference on Software Engineering (ICSE), May 2000, Limerick, pp. 263-272.

Del Bianco, V., Lavazza, I., Morasca, I., Taibi, D., Tosi, D. "An Investigation of the users' perception of OSS quality". Proceedings of the 6th International Conference on Open Source Systems (OSS), May/June 2010, Notre Dame, IN, USA.

Stamelos, I., Angelis, L., Oikonomou, A., Bleris, G.L., "Code Quality Analysis in Open-Source Software Development", Information Systems Journal, 2nd Special Issue on OS Software, 12(1), January 2002, pp. 43-60.

Koch, S. and Schneider, G. "Results from software engineering research into OS development projects using public data". Web published. (Last visited: Jan. 2011). http://opensource.mit.edu/papers/koch-ossoftwareengineering.pdf

Mockus, R.T. Fielding, J.D. Herbsleb, 2002: "Two Case Studies of Open Source Development: Apache and Mozilla". ACM Transactions on Software Engineering and Methodology Vol.11, No. 3, 2002, pp. 309-346.

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

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

Goode S., "Something for Nothing: Management Rejection of Open Source Software Australia's Top Firms", Information & Management, vol. 42, 2005, pp. 669-681.

Minelle, F., Stolfi, F. e Raiss, G., "A proposal for a certification scheme for the Italian Public Administration web sites quality", W3C – Euroweb Conference. Italy, 2001.

Golden B., "Making Open Source Ready for the Enterprise: The open Source Opportunities for Canada's Information and Communications Technology Sector", Information & Management, vol. 42, 2005, pp. 669 – 681.

Ruffatti G., Oltolina S., Tura D., Damiani E., Bellettini C., Colombo A., Frati F., "New Trends Towards Process Modelling: Spago4Q", Trust in Open-Source Software (TOSS), Limerick, June 2007.

Sourceforge Web page: http://www.sourceforge.net (Last visited: Jan. 2011).

QualiPSo Web page: http://www.qualipso.org (Last visited: Jan. 2011).

Wasserman, A., Pal, M., Chan, C., "Business Readiness Rating Project", Whitepaper 2005 RFC 1, http://www.openbrr.org/wiki/images/d/da/BRR_white paper_2005RFC1.pdf (Last visited: Jan., 2011).

International Standardization Organization. "ISO 9001:2008". Web published: www.iso.org. (Last visited: Jan. 2011).

Golden, B., "Making Open Source Ready for the Enterprise: The Open Source Maturity Model", from "Succeeding with Open Source", Addison-Wesley, 2005.

Apache Tomcat Web Site: http://tomcat.apache.org/ (Last visited: Jan. 2011).

W3C raccomandation, Web Content Accessibility Guidelines (WCAG) 2.0, 2008. Web published: http://www.w3.org/TR/WCAG20/ (Last visited: Sept. 2010).

Arnesen L. P., Dimiti J. M., Ingvoldstad J., Nergaard A., Sitaula N. K., "SourceForge Analysis, Evaluation and Improvements", Technical Report, May 2000, University of Oslo.

Jacob R. J., "The use of eye movements in human-computer interaction techniques: what you look at is what you get". ACM Transaction on Inf. Syst., Vol.9, No. 2 (Apr. 1991), pp. 152-169.

Pike, C., Gao, D., Udell, R., "Web Page Usability Analysis", Technical Report, December 2003, Pennsylvania State University.

FSF Free Software Foundation Web Page: www.fsf.org. (Last visited: Jan. 2011).

OP2A: http://www.op2a.org (Last visited: Jan. 2011).

Nielsen, J., and Landauer, T. K.: "A mathematical model of the finding of usability problems," *Proceedings of ACM INTERCHI'93 Conference*, 1993, pp. 206-213.

APPENDIX

Here, we present the core of the OP2A checklist and the results of its application to the Apache Tomcat website.

Project information availability		Overall Assessment		
1	**Overview**	**____ / 29.09**		
		Presence		
		Y	N	Weight
	• Product general description	x		3.92
	• Product age	x		3.92
	• Best Practices		x	6.23
	• Features high level description	x		4.29
	• Detailed Features description	x		4.29
	• License	x		6.44
2	**Requirements**	**____ / 8.59**		
		Presence		
		Y	N	Weight
	• Hardware requirements			
	• Disk usage		x	1.43
	• Memory usage	x		1.43
	• Min CPU required		x	1.43
	• Other HW requirements		x	1.43
	• Software requirements			
	• Supported operative systems	x		1.43
	• Required 3rd parties components (if applicable)		x	1.43
3	**License**	**____ / 13.34**		
		Presence		
		Y	N	Weight
	• Main license	x		3.22
	• Sub licenses (if applicable)	x		3.22
	• Law conformance (if applicable)	x		6.89
4	**Documentation**	**____ / 20.85**		
		Presence		
		Y	N	Weight
	• Technical documentation			
	• Code documentation (javadoc, etc.)	x		0.60
	• Code examples	x		0.60
	• Architectural documentation	x		0.60
	• Documentation on customization	x		0.60
	• Installation guide	x		0.60
	• Technical related F.A.Q.	x		0.60
	• Technical forum	x		0.60
	• Technical related mailing list	x		0.60
	• Testing documentation		x	0.60
	• Documentation about additional tools for developing, modifying or customizing the product (if applicable)	x		6.84
	• Security aspects analysis (if applicable)	x		6.21
	• User documentation			
	• User manual	x		0.60
	• Getting started guide	x		0.60
	• User related F.A.Q.	x		0.60
	• Mailing list	x		0.60
5	**Downloads**	**____ / 12.00**		
		Presence		
		Y	N	Weight
	• Download page	x		3.00
	• The download page is easily reachable	x		3.00
	• More than one archives	x		3.00
	• Specified the dimension of each downloads		x	3.00

6	**Quality reports**	**____ / 37.11**		
		Presence		
		Y	N	Weight
6.1	**Reliability**	____ / 8.20		
	- Correctness		x	1.64
	- Dependability		x	1.64
	- Failure frequency		x	1.64
	- Product maturity	x		1.64
	- Robustness		x	1.64
6.2	**Maintainability**	____ / 7.86		
	- Code size		x	1.96
	- Standard architectures (if applicable)		x	1.96
	- Language uniformity	x		1.96
	- Coding standard (if applicable)		x	1.96
6.3	**Performance**	____ / 7.34		
	- Performance tests and benchmarks (if applicable)		x	3.67
	- Specific performance-related documentation		x	3.67
6.4	**Product Usability**	____ / 7.20		
	- Ease of installation/configuration	x		3.60
	- ISO usability standard (ex. ISO 14598)	x		3.60
6.5	**Portability**	____ / 6.51		
	- Supported environments	x		2.17
	- Usage of a portable language	x		2.17
	- Environment-dependent implementation (e.g., usage of hw/sw libraries)	x		2.17
7	**Community & Support**	**____ / 26.52**		
		Presence		
		Y	N	Weight
7.1	**Community**	____ / 14.52		
	- Size of the community		x	7.20
	- Existence of mid / long term user community			
	- Trend of the number of users		x	2.44
	- Number of developers involved	x		2.44
	- Number of posts on forums / blogs / newsgroups	x		2.44
7.2	**Training and Support**	____ / 12.00		
	- Availability of training			
	- Training materials	x		2.54
	- Official training courses (if applicable)		x	2.54
	- Bugs number	x		1.72
	- Number of patches / release in the last 6 months	x		1.72
	- Average bug solving time		x	1.72
	- Availability of professional services (if applicable)	x		1.72
Total Score:		**98.19 / 147.50 (66.6%)**		