

OP2A: How to Improve the Quality of the Web Portal of Open Source Software Products

Luigi Lavazza, Sandro Morasca, Davide Taibi, Davide Tosi

Università degli Studi dell'Insubria
{luigi.lavazza, sandro.morasca, davide.taibi, davide.tosi}@uninsubria.it

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 OP2A, an evaluation model and we derive a checklist that OSS developers and web masters can use to design (or improve) 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 47 web sites of well-known OSS products to highlight the current deficiencies that characterize these web portals.

Keywords: Open Source Software, Quality Perception, Web portals quality assessment, Web portal quality model.

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 at no charge 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 [8]. 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 specialized tools.

The websites and web portals of an OSS product 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 [11], or, if they do, they provide this information in hidden sections of the website, thus not favoring usability [13]. 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 certifying the quality of OSS portals. The model is built upon the results of a survey [4, 5] – conducted in the context of the European project QualiPSO [19] – 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 47 well-known OSS portals to assess the general 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 47 OSS portals. Section 4 describes related works in the field of web quality and usability. We conclude and provide an outline for 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 [4, 5] in the context of QualiPSO [19] 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.

We used the results of this survey [4, 5], specifically the trustworthiness factors and the relevance score they obtained in the survey, to derive our OP2A.

2.2 The OP2A Assessment Model

Assessing and 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 quality 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 [4, 5] and the literature that describes well-known usability and accessibility rules for developing websites and web portals [14]. 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 the Appendix and in [17].

The checklist is structured in five areas: company information; web portal information; reasons of assessment; availability of information concerning trustworthiness factors; web 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 in which trustworthiness factors are considered and further detailed into subfactors. In turn, trustworthiness factors are grouped into the following categories:

1. *Requirements*: disk usage, memory usage, supported operating system, etc.;
2. *License*: reference to the license, use conditions, and law conformance;
3. *Documentation*: user documentation, technical documentation, etc. The detailed description of the OSS features is expected to be found here;
4. *Downloads*: the number of downloads and related information;
5. *Quality reports*: information about important properties like Reliability, Maintainability, Performance, Product Usability, Portability, etc.;
6. *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 [4, 5]. 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 [4, 5] 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: $6.45+6.89 = 13.34$.

| <i>Project Information Availability</i> | | <i>Overall Assessment</i> | |
|---|---------------------------------|---------------------------|---------------|
| 3. | License | _____ / 13.34 | |
| | | Presence | |
| | | Y | N |
| | | | Weight |
| - | Main license | | 3.22 |
| - | Sub licenses (if applicable) | | 3.22 |
| - | Law conformance (if applicable) | | 6.89 |

Fig. 1. Excerpt of the checklist for the area “project information availability”, category “License”.

The assessor 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 to a specific OSS product, the total value for the “License” category for that OSS product 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:

$$OP2A_Score\ (\%) = (Tot_Portal_Score / Tot_Applicable_Score) * 100$$

where *Tot_Portal_Score* is the sum of the scores for all the sub-factors that received the Y evaluation (or equally the sum of the scores for the seven categories), while *Tot_Applicable_Score* is the sum of all the scores for the sub-factors that are applicable for the portal under assessment. *OP2A_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 *OP2A_Score* suggests that the quality of the portal is good, while a low value of *OP2A_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.

3 Validation of OP2A

In this section, we illustrate the application of the OP2A model to the Apache Tomcat website [1], and to 47 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.

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.

Figure 4 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– is only 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 (*OP2A_Score* = 66.6%).

3.2 A Proposal for Refactoring the Apache Tomcat Website

As described in Section 3.1, the Apache Tomcat website gained an *OP2A_Score* of 66.6% 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 *OP2A_Score*, so that OSS users and also developers will be able to quickly find all required information, and the probability of adoption/reuse of the Apache Tomcat product will increase.

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

| <i>Original Website</i> | | | <i>Refactored Version</i> | |
|-------------------------|---------------------------|-------------------|---------------------------|---------------------------|
| <i>overall time</i> | <i>quality perception</i> | <i>OP2A score</i> | <i>overall time</i> | <i>quality perception</i> |
| 1h30m | 2 | 98.19 (66.6%) | 26m | 3 |
| 1h18m | 2 | 95.65 (64.8%) | 30m | 4 |
| 1h00m | 2 | 96.99 (65.7%) | 30m | 4 |
| 1h15m | 2 | 94.59 (64.1%) | 25m | 4 |
| 1h10m | 2 | 97.59 (66.2%) | 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. 5, 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 [13].

To validate the quality of the refactored version of the website against the original one, we asked ten Computer Science Master’s students, who had never accessed the Apache Tomcat portal before, to surf the original web portal and the refactored one 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 applying the OP2A checklist to 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 to 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 seems to have improved the quality of the portal (that now obtained an *OP2A_Score* near to 100%). 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 relatively low value of 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 empirical study, we selected ten students, following [15]. We are conducting additional experiments with a larger number of students in order to strongly validate these preliminary results.

3.3 Analysis of OSS portals

We analyzed the portals of 47 OSS projects to get a preliminary assessment of their quality. The purpose was not actually to show the feasibility of OP2A, but to provide the evidence that techniques like OP2A are actually needed, as the quality of OSS projects portals is far from adequate, with respect to the quality of OSS products and their ambitions to substantially increase their user basis.

The set of portals has been selected by taking into account different types of software products, generally considered well-known, stable, and mature. In [17] it is possible to find the details of this experimentation.

Figure 2 summarizes the results of the assessment, by reporting the *OP2A_Scores*, computed as the ratio between the obtained score and the total achievable score. The empirical study shows that the quality of the analyzed portals is generally inadequate (the mean value of the *OP2A_Score* achieved by the portals is 57.3%), thus calling for heavy portal refactoring to achieve an acceptable level of quality.

Figure 2 shows that the situation is quite homogeneous: over 26 portals out of 47 achieved an *OP2A_Score* between 40 and 60%. Nineteen portals achieved a 60% or higher score, but only 7 are above 70%.

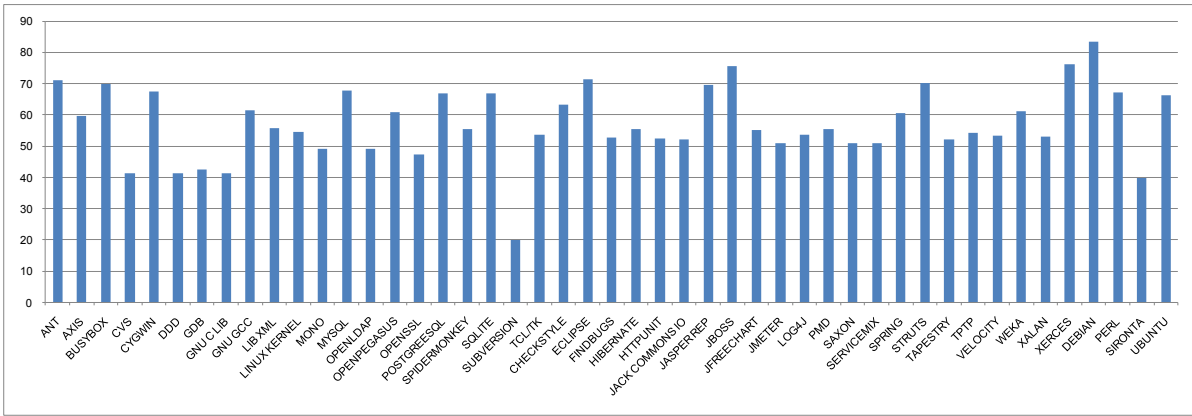


Fig. 2. OSS portal scores.

The overall quality per portal section is described in Figure 3. It is easy to see that the areas of portals with the lowest quality are those concerning requirements, quality reports, licenses, and the community and support. On the one hand, it is natural that most OSS projects rely on spontaneous spreading of information concerning the functionality and quality of their products; on the other hand, it is clearly a problem if a potential user, having heard of product XYZ, looks for the site of XYZ and does not find clear information about the product functionalities, under what license it is released, what quality claims and evidence are available, etc.

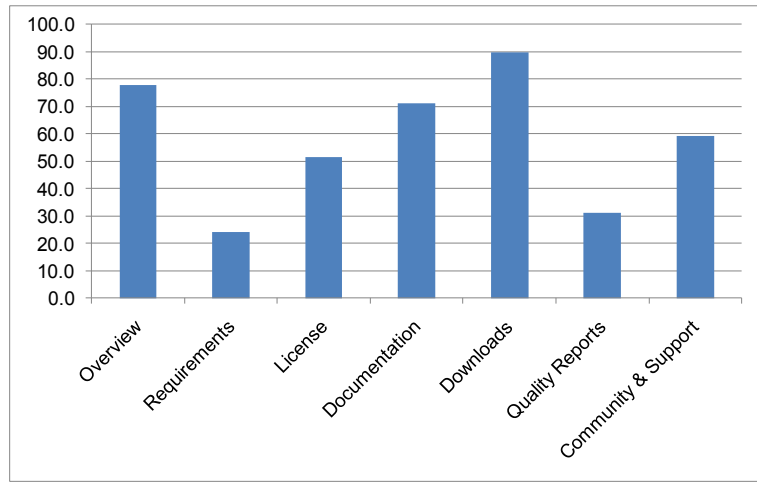


Fig. 3. Average scores per portal section.

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 understand the characteristics of the available products, and evaluate the pros and cons of its adoption. Some of the most known OSS evaluation models are: OpenBRR [23], QSOS [3], OSMM [7] and OpenBQR [21]. 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. 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 [7] 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. To reduce the amount of information to be evaluated, we carried out a survey [10] 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 [12]. 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) has 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 [22]. 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 [20]: Arnesen et al. [2] showed several problems, mainly concerning the link structure and the information organization. Another study [18] identified usability problems both with Sourceforge and with the Free Software Foundation [6] website by means of eye tracker techniques [9].

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 47 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 to understand whether: (1) the weight of subfactors should be refined, for example asking OSS developers and users to weight also subfactors; (2) define targeted weights for the different OSS stakeholders such as managers, developers, and end-users; (3) collecting additional statistical data about the analyzed portals; (4) 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

The Apache Software Foundation
<http://www.apache.org/>

Search the Site Search Site

Apache Tomcat

- [Home](#)
- [Taglibs](#)

Download

- [Which version?](#)
- [Tomcat 6.x](#)
- [Tomcat 5.5](#)
- [Tomcat Connectors](#)
- [Tomcat Native](#)
- [Archives](#)

Documentation

- [Tomcat 6.0](#)
- [Tomcat 5.5](#)
- [Tomcat Connectors](#)
- [Tomcat Native](#)
- [Migration Guide](#)

Problems?

- [Security Reports](#)
- [Find help](#)
- [FAQ](#)
- [Mailing Lists](#)
- [Bug Database](#)
- [IRC](#)

Get Involved

- [Overview](#)
- [SVN Repositories](#)
- [Mailing Lists](#)
- [Wiki](#)

Misc

- [Who We Are](#)
- [Heritage](#)
- [Apache Home](#)
- [Resources](#)
- [Contact](#)
- [Legal](#)
- [Sponsorship](#)
- [Thanks](#)

Apache Tomcat

Apache Tomcat is an open source software implementation of the [Java Servlet](#) and [JavaServer Pages](#) technologies. The Java Servlet and JavaServer Pages specifications are developed under the [Java Community Process](#).

Apache Tomcat is developed in an open and participatory environment and released under the [Apache Software License](#). Apache Tomcat is intended to be a collaboration of the best-of-breed developers from around the world. We invite you to participate in this open development project. To learn more about getting involved, [click here](#).

Apache Tomcat powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations. Some of these users and their stories are listed on the [PoweredBy](#) wiki page.

Tomcat 6.0.24 Released 2010-01-21

The Apache Tomcat Project is proud to announce the release of version 6.0.24 of Apache Tomcat. This release includes many bugfixes over Apache Tomcat 6.0.20.

[Download](#) | [ChangeLog](#) for 6.0.24

Tomcat Native 1.1.19 Released 2010-01-05

The Apache Tomcat Project is proud to announce the release of version 1.1.19 of Tomcat Native. This version is principally a bugfix release.

[Download](#) | [ChangeLog](#) for 1.1.19

Old news

See [former announcements](#).

Fig. 4. Original Apache Tomcat website.



Apache
Tomcat



TEN YEARS OF TOMCAT

Search the Site

[Contact](#) | [Legal](#) | [Sponsorship](#)

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 software implementation of the [Java Servlet](#) and [JavaServer Pages](#) technologies. The Java Servlet and JavaServer Pages specifications are developed under the [Java Community Process](#).

Apache Tomcat is developed in an open and participatory environment and released under the [Apache Software License](#). Apache Tomcat is intended to be a collaboration of the best-of-breed developers from around the world. We invite you to participate in this open development project. To learn more about getting involved, [click here](#).

Apache Tomcat powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations. Some of these users and their stories are listed on the [PoweredBy](#) wiki page.

Tomcat 6.0.24 Released

2010-01-21

The Apache Tomcat Project is proud to announce the release of version 6.0.24 of Apache Tomcat. This release includes many bugfixes over Apache Tomcat 6.0.20.

[Download](#) | [ChangeLog for 6.0.24](#)

Tomcat Native 1.1.19 Released

2010-01-05

The Apache Tomcat Project is proud to announce the release of version 1.1.19 of Tomcat Native. This version is principally a bugfix release.

[Download](#) | [ChangeLog for 1.1.19](#)

Old news

See [former announcements](#).

Fig. 5. Apache Tomcat website Refactored.

References

1. Apache Tomcat Web Site: <http://tomcat.apache.org/> (Last visited: Jan. 2011).
2. 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.
3. Atos Origin, "Method for Qualification and Selection of Open Source Software (QSOS)". Web published: <http://www.qsos.org> (Last visited: Jan., 2011).
4. 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).
5. Del Bianco V., Lavazza L., Morasca S., Taibi D.: "A Survey on Open Source Software Trustworthiness", IEEE Software, IEEE Computer Society. To appear.
6. FSF Free Software Foundation Web Page: www.fsf.org. (Last visited: Jan. 2011).
7. Golden, B., "Making Open Source Ready for the Enterprise: The Open Source Maturity Model", from "Succeeding with Open Source", Addison-Wesley, 2005.
8. International Standardization Organization. "ISO 9001:2008". Web published: www.iso.org. (Last visited: Jan. 2011).
9. 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.
10. 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.
11. 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).
12. 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.
13. Nielsen, J., "User Interface Directions for the Web", Communications of the ACM, Vol. 42, No. 1, 1999, pp. 65-72.
14. Nielsen, J., Norman, D.: "Usability On The Web Isn't A Luxury", *InformationWeek*, February 2000.
15. 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.
16. 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.
17. OP2A: <http://www.op2a.org> (Last visited: Jan. 2011).
18. Pike, C., Gao, D., Udell, R., "Web Page Usability Analysis", Technical Report, December 2003, Pennsylvania State University.
19. QualiPSo Web page: <http://www.qualipso.org> (Last visited: Jan. 2011).
20. Sourceforge Web page: <http://www.sourceforge.net> (Last visited: Jan. 2011).
21. 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.
22. W3C recommendation, Web Content Accessibility Guidelines (WCAG) 2.0, 2008. Web published: <http://www.w3.org/TR/WCAG20/> (Last visited: Sept. 2010).
23. Wasserman, A., Pal, M., Chan, C., "Business Readiness Rating Project", Whitepaper 2005 RFC 1, http://www.openbrr.org/wiki/images/d/da/BRR_whitepaper_2005RFC1.pdf (Last visited: Jan., 2011).

APPENDIX

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

| Project information availability | | Overall Assessment | | |
|----------------------------------|---|----------------------|---|--------|
| 1 | Overview | 22.86 / 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 | 2.86 / 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 / 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 | 20.25 / 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 | 9.00 / 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 | 17.31 / 37.11 | | |
| | | Presence | | |
| | | Y | N | Weight |
| 6.1 | Reliability | 1.64 / 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 | 1.96 / 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 | 0.00 / 7.34 | | |
| | • Performance tests and benchmarks (if applicable) | | x | 3.67 |
| | • Specific performance-related documentation | | x | 3.67 |
| 6.4 | Product Usability | 7.20 / 7.20 | | |
| | • Ease of installation/configuration | x | | 3.60 |
| | • ISO usability standard (ex. ISO 14598) | x | | 3.60 |
| 6.5 | Portability | 6.51 / 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 | 12.58 / 26.52 | | |
| | | Presence | | |
| | | Y | N | Weight |
| 7.1 | Community | 4.88 / 14.52 | | |
| | • Size of the community | | x | 7.20 |
| | • Existence of mid / long term user community | | x | 2.44 |
| | • 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 | 7.70 / 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%) | | |