F*l*OSS Managed Data Sources Maturity Level: a first attempt

Junior (Frédéric) Fleurial Monfils Jean-Christophe Deprez
Centre d'Excellence en Technologie de l'Information et de la Communication (CETIC)
{frederic.fleurialmonfils, jean-christophe.deprez}@cetic.be

Abstract

Many organizations have started to integrate Free (libre) Open Source Software and are currently faced with the problem of selecting the components that meet their quality needs, in particular, regarding their evolvability and their robustness. Their assessment is often performed via ad hoc investigations on a few publicly available data sources such as IT newspapers and the internet because of a lack of time and methodology. This paper[1] *identifies and describes some of the major electronic data sources where the information can be extracted during the assessment of the evolvability (and the maturity level) of FlOSS.*

1. Introduction

Free (*libre*) Open-Source Software (F*l*OSS[2]) is becoming increasingly popular as a growing number of organisations are adopting them for their businesses or integrating them in their infrastructures. Selecting the right F*l*OSS piece of software becomes then a business critical task. The major concerns of the organizations regarding F*l*OSS adoption are the functional suitability and the support availability of the F*l*OSS product; more precisely, the organizations want F*l*OSS they can rely on now (F*l*OSS Robustness) and in the future (F*l*OSS Evolvability). To assess these quality characteristics, organizations often collect information from ad hoc sources (newspaper, word of mouth, etc.). This way of working mostly augment their confidence about the popularity of the F*l*OSS they want to use but does not says anything about the evolvability of this F*l*OSS.

There are methodologies designed to assess the quality of F*l*OSS such as the *Open Business Readiness Rating* [1] (OpenBRR), the *Method for Qualification and Selection of Open Source software* [2] (QSOS) and the CapGemini's *Open Source Maturity Model* [3] (OSMM). The typical steps in these methodologies are:

1. Specifying the needs, that is, the list of functional platform or environment requirements.
2. Identifying all the characteristics proposed by the methodology and computing the result of the assessment
3. Applying a threshold or weight to each characteristic
4. Filtering the tools by eliminating the ones outside the thresholds
5. Performing more detailed manual tests to decide on the most appropriate F*l*OSS Product

However, these methodologies assess the *current* suitability of F*l*OSS according to the needs of a potential adopter (individual or organisation) but they fail at addressing whether the F*l*OSS is evolvable, that is, its ability to evolve easily [4] in line with the changing needs of the organization. Furthermore, these methodologies lack a formal notation, and more precisely they do not require a clear localisation of the data on which their proposed measurements are to be performed. Even standards like ISO 9126 [5] do not provide a solution to this issue.

This lack of systematic inventory of the source of information does not allow repeatable measures. For example some of the methodologies require the organization to rank the activity on a particular release. To do this, the organization can base itself on the archives that can be downloaded from the F*l*OSS website, or it can base itself on the available version control system of the F*l*OSS.

In this paper, our contribution is then twofold. First, we list the major electronic data source publicly available managed by the F*l*OSS communities. Second, we provide a maturity level for these electronic data sources related to the accessibility and/or extractability of the information. This maturity level can be useful for two purposes: (1) it gives a first idea about the maturity level of the evaluated F*l*OSS, and (2) it also gives a

[1] Acknowledgment: This work is partly funded by the Belgian CRAQ-155 project (Convention N°EP1A1203000073F-130008) and by the European Union QUALOSS project (Contract N° 33547)

[2] The '*l*' in F*l*OSS is the italicized letter l that stands for *libre*.

confidence level on the measures taken during the assessment.

The paper is organized as follows: Section 2 defines the terms and concepts used. Section 3 explains the purpose of the identification of the FlOSS data sources in the context of the EU QUALOSS Project. Section 4 presents our first attempt to attach maturity levels on a few selected F*l*OSS data sources. Section 5 compares our attempt with related efforts. Section 6 concludes and presents our future works.

2. Definitions

We define below some useful concepts related to F*l*OSS assessment in the context of this paper.

F*l*OSS Endeavour:

- The F*l*OSS product (which itself includes code, documentation, and tests);
- The F*l*OSS community (members closely or remotely connected to the endeavour);
- The process, that is, the rules and division of labor that community members obliged to when performing activities;
- and the tools used by the community during their respective activities.

We prefer to introduce this new notion of F*l*OSS Endeavour to stay away from the term F*l*OSS Project, which has different connotations to different people.

F*l*OSS License: A license listed by the Open Source Initiative (OSI, www.osi.org) or a license that is legally compatible with a license listed by the OSI.

F*l*OSS Product: A piece of executable software that is released under the terms of a F*l*OSS License.

F*l*OSS Community: Individuals or organizations that have a vested interest in a F*l*OSS Product and that contribute to its promotion and adoption. The roles in the F*l*OSS Community are typically F*l*OSS Maintainer, F*l*OSS Contributor, F*l*OSS Adopter or F*l*OSS Integrator.

F*l*OSS Forge: A large container where the F*l*OSS Community publishes the releases of their products. It can also stores the archives of the communications and give access to the development activities.

F*l*OSS Artifact: An artifact is a (part of a) workproduct, i.e. product of the work, on which measurements are performed.

F*l*OSS Managed Data Source: A container of F*l*OSS artifacts maintained by the F*l*OSS Endeavour.

F*l*OSS Potential Adopter: Individual or Organization that is in the process of evaluating the F*l*OSS Endeavour in order to decide to adopt its F*l*OSS Product or not. They are not yet part of the F*l*OSS Community.

F*l*OSS Assessment: An assessment performed by a F*l*OSS Potential Adopter. The result of this assessment is typically the adoption the F*l*OSS Product, the adherence to the F*l*OSS Community, or the publication of the F*l*OSS Assessment Report.

3. Assessment of a F*l*OSS Endeavour

The EU QUALOSS Project aims at assessing the robustness and the evolvability of F*l*OSS Endeavours. From the interviews we have conducted, the interviewees list the functionalities and the available support as their main concerns.

Identifying the functionalities for a particular product release is of major importance for the F*l*OSS Potential Adopter as this will allow an quick check on whether or not the F*l*OSS Product suits their needs.

The F*l*OSS Potential Adopter seeks for help about the F*l*OSS Product. First, the support is found in the provided documentation. Second, the F*l*OSS Potential Adopter wants to find answers by interacting with the community. Finally, they want to be able to find (local) experts.

The F*l*OSS Potential Adopters are thus looking for F*l*OSS Products that meet their needs and on which they can rely now (Robust) and in the future (Evolvable).

F*l*OSS assessment methodologies like OpenBRR, QSOS or OSMM mostly explain the *why* and the *what* but seldom the *how*. When starting assessment with the current methodologies, the F*l*OSS Potential Adopter has often to find out how to perform the measurement. These methodologies indeed lack a systematic detailed localisation of the F*l*OSS artifacts on which to take the measures.

One of the first and critical steps in the F*l*OSS Assessment Methodologies is then the identification of the data sources managed by the F*l*OSS Endeavour.

As mentioned earlier, the measurements to take will concern the suitability of the evaluated F*l*OSS Product regarding its provided functionalities and the availability of its offered support. These F*l*OSS artifacts can be divided in three categories:

- *Source or executable code of a FlOSS product release:* The source code explicitly defines how the functionalities are implemented but is not the easiest way to decide on the suitability of the functionalities. Concerning the availability of the support, the rule “the code is the documentation” is sometimes the only one applied.
- *Documentation of the product release*: Documentation provides information about the functionalities of the product and ranges from a paragraph describing the F*l*OSS Product on the project homepage, to the various technical or functional user guides or demo material. Regarding the support, the documentation explains how to get, install and use the product.
- *Communications with the community*: The communications with the community can be done through the exchange of e-mails with the developer but can also consist on the consultation of posts on a full featured collaborative web-site or an on-line video (conference). The functionalities of the product can be accessed through the discussion logs, or be requested through the interactions with the community.

The accessibility (ease with which the information can be extracted from the source) and the usefulness (extent to which the information extracted from the source is relevant) of these data sources can provide a quick assessment on the maturity level of the F*l*OSS Endeavour.

4. Maturity level of F*l*OSS Managed Data Sources

The F*l*OSS Artifacts are produced by the different activities performed by the F*l*OSS Community. Typically, these artefacts are found in F*l*OSS Data Sources managed by the F*l*OSS Endeavour. In the context of the EU QUALOSS Project, the evolvability of the F*l*OSS Endeavour (product and community) is assessed as the ability of the endeavour to cope with the evolution of the F*l*OSS users' needs.

This section describes some of the Data Sources managed by the F*l*OSS Endeavour related to the major activities of the F*l*OSS Community. We also associate a maturity level regarding their availability and/or accessibility form. This information is often found on the website of the F*l*OSS Endeavour, or on F*l*OSS Announcement Sites like the Free Software Directory [6].

4.1 Distribution of the product release

The F*l*OSS Endeavour distributes a particular release of its F*l*OSS Product in a F*l*OSS Packaged Distribution. This distribution can be viewed as a snapshot of the work(products) of the F*l*OSS Community at a certain time. The packaged distribution is one of the following:

- A *source package distribution* containing the source code of the product release and all related technical documentation for the developer or maintainer;
- A *binary package distribution* containing the binary version (or executable version) of the product release and all the related documentation for the end user of the product release.

Typically a packaged distribution contains the following workproducts:

- *Test files* containing the unit test cases of the product release;
- *Source files* containing the source code of the product release;
- *Documentation files* containing the functional or technical documentation of the product release;
- *Build files* needed to build the executable version of the product release;
- *Executable files* needed to have a running version of the product release;
- *Resource files* such as Images, Video, etc.

Maturity level of a F*l*OSS packaged distribution: The maturity levels capture the ease with which the F*l*OSS Potential Adopter can evaluate a running version of the F*l*OSS Product on their environment. The goal is to evaluate the suitability of the functionalities and the quality of the support (help functionality) of the product by running it.

1. A source archive: the packaged distribution is almost exclusively dedicated to the potential developer contributor. The potential end-user is forced to compile the distribution to use the product release;
2. A source archive and a binary archive targeting a single platform: the packaged distribution contains the product release ready to be used;
3. Source archives and binary archives targeting multiple platforms: the product release can be used easily on various platforms;
4. Source archives and binary archives containing documentation: the documentation

of the product release is also provided in the archive;

5. Nightly built source and binary archives: the F*l*OSS Potential Adopter has access to the latest versions of the product release;
6. Multi licensed source and binary archives: the F*l*OSS Potential Adopter can chose their license and hence can include the F*l*OSS Product Release in their own product.

4.2 Versioning of the product releases

To ease the management of the different product releases, the F*l*OSS Endeavour stores them in a F*l*OSS Version Control Repository. It allows access to artifacts produced in various F*l*OSS Product Releases. It can be as simple as the list of the Packaged Distributions or a true Version Control System such as CVS or Subversion. Version Control Systems also logs the actions (who, when, what) and then allow detailed analysis of the activities of the F*l*OSS Community regarding a F*l*OSS Product.

Maturity level of the FlOSS Version Control Repository:
The maturity levels capture the way different versions of F*l*OSS Artifacts can be identified and/or retrieved. The goal is to see the evolution of the implementation of the functionalities and the evolution of the quality of the documentation (the first available support).

1. A “Change Log” section or a “TODO” Section in FlOSS Product Documentation: The generic files such as README, TODO or NEWS provide minimum information about the (functional) content of the product release;
2. A list of packaged distributions on the F*l*OSS Website giving the history of the functionalities can be used to assess the suitability of the product;
3. A list of packaged distributions on a major FlOSS Forge allows the retrieval of the previous releases and the manual comparison of the executable product and documentation;
4. A version control system without a web interface: the access to a particular version is easier and requires the ckeck out of the version of interest;
5. A version control system with a web interface eases the access to the implementation of the versions;
6. A project management system gives an on-line and integrated access to the code contents, the issues, the documentation, etc.

4.3 Management of the issues

The F*l*OSS Endeavour often logs the issues encountered in the adoption of his F*l*OSS Product by getting feedback from the F*l*OSS Adopters and logging them in a F*l*OSS Issues Log. The F*l*OSS Issues Log gives access to the issues reported by the F*l*OSS Community over F*l*OSS Product Releases. Regarding the functionalities, the issue list also gives an idea of the current functionalities that have some problem. In the simple form, it can be a file in a Packaged Distribution that contains known issues, or this can be a true Issue Tracking System like a Bug Tracking System or any Ticketing system. Issue Tracking Systems logs the life-cycle of the issue from its reporting to its fix.

Maturity level of the F*l*OSS Issues Log:
The maturity levels capture the ease with which issues on particular release of the F*l*OSS Product can be tracked and the quality of the support provided by the F*l*OSS Community to solve the issues and answer the requests of the users. This also allows the F*l*OSS Potential Adopter to have an idea on the evolution of the F*l*OSS Product thanks to the request for changes that are typically tracked in these issue logs. The potential adopter can then see if the product will meet its future needs in terms of functionalities.

1. A “Change Log” section or a “TODO” section in documentation files of the product release mentioning the current issues and the new functionalities;
2. A “Known Issues” section in documentation files of the product release;
3. A generic mailing list archive containing the handling of the issues reported about the product release;
4. A dedicated mailing list archive containing the handling of the issues reported about the product release;
5. A generic issue tracking system on major FlOSS Forge
6. A dedicated issue tracking system on F*l*OSS Website

4.4 Interaction with the community

A F*l*OSS Discussion Log keeps track of the discussions, i.e. threads of messages in reply to an original message posted to the address associated to this discussion list. The replies are sent to the original poster and to the (registered) members of the distribution list. The messages sent to the distribution lists (web forums, newsgroups, mailing list) can be accessed through various client softwares such as a web browser, a news reader or an e-mail client.

Maturity level of the F*l*OSS Discussion Log:
The maturity levels capture the ease with which the F*l*OSS Potential Adopter can obtain help by communicating with the F*l*OSS Community. Often, the potential adopter has access to the archives of the discussions or is asked to consult the FAQ. The goal of this data source is mostly related to the available support.

1. A developer's e-mail address;
2. A generic mailing list archive: the discussions are archived;
3. A dedicated mailing lists archives;
4. An IRC Channel Log, Newsgroup or FAQ;
5. A blog or news site;
6. An on-line Webforum.

5. Related works

There are various F*l*OSS assessment methodologies such as OpenBRR, QSOS or OSMM. The typical steps in these methodologies are:

1. Specifying the needs, that is, the list of functional platform or environment requirements.
2. Identifying all the characteristics proposed by the methodology and computing the result of the assessment
3. Applying a threshold or weight to each characteristic
4. Filtering the tools by eliminating the ones outside the thresholds
5. Performing more detailed manual tests to decide on the most appropriate F*l*OSS Product

In these methodologies, the F*l*OSS Potential Adopter has no rigorous means to select the artifacts on which to take measures. He cannot provide objective and repeatable results for their assessment.

This paper lists some of the major data sources investigated in the QUALOSS project. The identified sources should be listed along with the assessment result to allow the traceability and the repeatability of the measures.

6. Conclusion and future work

The purpose of the QUALOSS project is to assess the robustness and evolvability of F*l*OSS Endeavours by performing a large number of measures in an automatic way. The maturity levels associated to the data sources provide (1) a confidence factor for the measures that are to be performed during the QUALOSS assessment and (2) an idea on the effort that has to be allocated on the development of extraction tools.

- The confidence factor can be explained by the fact that the recall and the precision of the results drawn from specialised and dedicated data sources will be better than the recall and precision drawn from more general purpose data sources. For example, the number of issues will be more precise and easier to compute if the F*l*OSS Endeavour has put in place a bugtracker than if the issues are only managed in a mailing list archive.
- Some of the measurements required in the F*l*OSS Assessment Methodologies entail a huge amount of manual work. It is then necessary to automate as much as possible the extraction of data and their evaluation. The identification of the data sources managed by the F*l*OSS Endeavour is then of major importance. The maturity level of these data sources gives an indication on the ease with which the data could be extracted. For example, the number of changes performed on the source code of a two successive stable versions of a product is easier to compute from a version control system than from a list of the packaged distributions.

The list of data sources presented in this paper is not exhaustive and the F*l*OSS Potential Adopter is free to augment it. This is one of the first task of the EU QUALOSS Project.

The maturity level associated to the presented F*l*OSS Managed Sata Sources, will be validated during the second phase of the EU QUALOSS project. They will be part of the measurement methodology designed to estimate the evolvability of a F*l*OSS Endeavour.

The maturity levels have also to be put in the perspective of the identified processes of the F*l*OSS Endeavour and investigated in the context of the large number of activities performed a F*l*OSS Community. This work will be done applying the activity theory [7] in the context of F*l*OSS.

Finally, the assessment of the maturity level of the F*l*OSS listed on the Free Software Directory could be a good case study. Indeed, this directory has already listed the major communication means for the various F*l*OSS Endeavours that it contains.

7. References

[1] OpenBRR.org (SkipeSource, Carnegie Mellon West, Intel), Business Readiness Rating for Open Source, 2005, BRR_whitepaper_2005RFC1.pdf. (http://www.openbrr.org)

[2] Atos Origin, Method for Qualification and Selection of Open Source software (QSOS) v1.6, April 2006, Licensed under GNU Free Documentation License . (http://www.qsos.org/download/qsos-1.6-en.pdf)

[3] Frans-Willem Duijnhouwer and Chris Widdows (Capgemini), Open Source Maturity Model 1.5.3, August 2003, (Expert Letter) (http://www.seriouslyopen.org)

[4] Evolvability, Tim Berners-Lee, 7th International WWW Conference, 1998

[5] ISO/JTC 1/SC 7, ISO/IEC 9126 Software Engineering - Product Quality (Part1-4), 2001.

[6] Free Software Directory, Free Software Foundation, (http://directory.fsf.org)

[7] Bjørke, S.Å. (2005) 'The Concepts of Communities of Practice, Activity Theory and Implications for Distributed Learning' in Lager, R. (ed) *Artikler om høgskolepedagogikk, Skriftserien nr 118*, Agder University College, Kristiansand 2005, ISBN: 82-7117-558-0