The FAME Approach: an assessing methodology

FILIPPO EROS PANI*, GIULIO CONCAS*, DANIELE SANNA**, LUCA CARROGU**

*Department of Electric and Electronic Engineering, Agile Group
University of Cagliari
Piazza d'Armi, 09123 Cagliari
ITALY
{filippo.pani, concas}@diee.unica.it http://agile.diee.unica.it

**FlossLab Srl
Viale Elmas 142, 09122 Cagliari
ITALY
{daniele.sanna, luca.carrogu}@flosslab.it http://www.flosslab.it

Abstract: In recent years the increasing of open source solutions extended the business opportunities for the IT companies but at the same time created problems for the choice and evaluation processes. For these reasons the need of approaches and tools to address this issue has become capital. We present FAME (Filter, Analyze, Measure and Evaluate), an iterative approach for open source software assessment, and an idea for a support tool for this methodology. In this paper we analyze the most interesting and important assessment methodologies and the FAME approach that derived by these, more heavyweight, proven approaches developed in a University research and consulting company environment, aims to match the needs of small organizations. The proposed approach has been used by FlossLab, the first spin-off of University of Cagliari, to select the best solutions. This tool can be useful to make easier the whole process, supporting the users in this choice by an interactive approach, particularly in the filtering and analysis phases.

Key-Words: open source software, software evaluation, technology transfer, software quality, assessment model.

1 Introduction

The choice of technologies for own IT investment is fundamental for organizations, because they influence practically all their businesses processes. Therefore, the optimal choice of their architecture and software components is of paramount importance. A wrong choice can lead to dire consequences, such as inefficiencies, information loss, higher maintenance and redesign costs, stop of operational activities, and so on.

In recent years, free/open source software (F/OSS) emerged as a viable solution for software applications [1][2]. The increasing interest in F/OSS is patent in many different contexts like communities of individual users, private firms focusing their attention on this kind of approach, and public institutions. The reasons of the success of F/OSS software include:

- Cost: F/OSS is usually available at no cost for public download.
- Quality: F/OSS source code is available to all. This transparent approach enables to produce high-quality products, achieved through an elaborate peer-review process performed by a large community of users, who act as co-developers to identify and correct software defects and add features [3-7].

The European Commission is currently funding several research projects related to open source and quality, namely, QUALOSS [8] FLOSSMetrics [9], SQO-OSS [10] and QUALIPSO [11].

However, it is very difficult to decide which F/OSS application to adopt inside an organization, because the number of open source projects is strongly increasing. Some products have their own web site as the main distribution mechanism for the software. However, the huge part of F/OSS products are available through portals, which act as repositories of projects. On SourceForge alone, one of the most important repository, more than one hundred thousand projects are hosted. So the myriad of F/OSS products makes actual adoption a real challenge, and it is necessary to have methods to assess and compare these software products [12-17].

In the last years, many F/OSS evaluation methodologies have been proposed to address this issue. All these methodologies have effectiveness as their main goal, and this goal led to a hardly sustainable increase of their complexity, from both the perspective of their costs and needed competences. All evaluation frameworks reported in the literature were devised using an analytic research approach, trying to analyze very many factors [18-22]. For this reason, such frameworks are often not easily applicable to real environments, especially in the case of small organizations.

This paper presents a new F/OSS maturity and reliability evaluation methodology, called FAME for "Filter, Analyze, Measure and Evaluate". FAME is aimed to reduce the evaluation complexity, and thus to be easily usable also by SMEs and small public bodies.

The FAME approach is derived from previous studies performed at the University of Cagliari, characterized by a rigorous, but heavyweight approach to F/OSS selection [18][23][24].

The goals of FAME methodology are to aid the choice of high-quality F/OSS products, with high probability to be sustainable in the long term, and to be as simple and user friendly as possible. The evaluation is not only about technical features of the product and quality of its development community, but it also takes into account a cost-benefit analysis specific of the involved organization.

To support this methodology we are studying a specific tool.

2 Software Assessment Methodologies

Many studies [3][12][17] investigated whether the maturity of the processes employed by distributed, volunteer projects is linked to their success. The results of these studies clearly showed that the maturity of open source processes are correlated to the success of a project. This study identified the importance of the use of version control tools, effective communication through the deployment of mailing lists, and found several effective strategies related to testing.

Despite the widespread use of open source products in academic and industrial environments, only recently first attempts have been made to evaluate open source products. Some significant contributions are mentioned in this section. All these methodologies have common criteria; they also present various phases and are based on scores.
Some among the main F/OSS evaluation methodologies are the following.

2.1 Business Readiness Rating

Business Readiness Rating (BRR) is being proposed as a new standard model for rating open source software. It is intended to enable the entire community (enterprise adopters and developers) to rate software in an open and standardized way [24].

2.2 The Open Source Maturity Model

The Open Source Maturity Model (OSMM) is designed to help organizations to successfully adopt and implement open source software. It consists in a three-phase process for selecting, assessing and implementing F/OSS products [25].

2.3 The Open Source Maturity Model by Capgemini

In order to be able to determine if an open source product is suitable for an organization, Capgemini developed its Open Source Maturity Model (OSMM) [26]. The OSMM describes how a F/OSS product should be assessed to ensure that the product meets the IT challenges companies face today. The OSMM accomplishes this by linking an extensive product analysis with a thorough review of the company and its IT issues.

2.4 QSOS

In order to have a method of qualification and selection of open software, Atos Origin built an original methodology to evaluate F/OSS software called QSOS [21]. The general process of QSOS is made up of several interdependent steps.

2.5 EFFLOSS

This approach is intended to help IT organization assessing which open source software would be most suitable for their needs [22]. The main limit of the frameworks described above is that they are based on qualitative metrics. The key idea of EFFLOSS is to systematically use quantitative metrics that can be automated, taking advantage of the information that can be found in the Web. In order to allow an easy usage of EFFLOSS, the assessment process is divided in three steps, which are performed sequentially: the first one determines the features that characterize an open source product [27]; the second one identifies the success metrics; the third one assigns a score for each metric.

2.6 NVAF

The NVAF is a framework that aims at giving a contribution to support the public administrations in their decision-making choices.

NVAF (Needs, Values and Assessment Framework) is addressed to public administrations and has a neutral approach to assure impartiality in the adoption choice of IT solution based on proprietary software or open source, and to avoid any kind of discrimination. The framework implements the criteria for evaluating the strategic choice of a solution compared to another. It allows evaluation if the choice increases the value into the environment in question.

3 FAME: Filter, Analyze, Measure and Evaluate Approach

FAME methodology originates from previously described heavyweight open source assessment methodologies. FAME can be considered an evolution of many software comparison methodologies and it focuses on some aspects like the maturity, the durability and the strategy of the organisation around the open source project itself. FAME takes into account many aspects also present in NVAF (Needs, Values, Assessment Framework) [23], intended to support the choice of software applications. NVAF is mainly addressed to public administrations and has a neutral approach to assure impartiality in the adoption of IT solutions based on proprietary software or F/OSS. The framework considers also the social impact of the choice. Another very interesting feature of this methodology is the use of quantitative metrics gathered on the Internet about the projects to evaluate.

All these methodologies, however, are quite heavyweight approaches, suitable for large organizations or research studies, but difficult to adopt by SMEs. For this reason, in the context of a research project of FlossLab, an Italian SME, we decided to devise a new methodology to support SMEs in selecting F/OSS applications, using a simple, structured and tool-based approach, with the University of Cagliari support.

The main idea behind FAME is that the users should evaluate which solution amongst those available is more suitable to their needs by comparing technical and economical factors, and also taking into account the total cost of individual solutions and cash outflows. It is necessary to consider the investment in its totality and not in separate parts that are independent of one another.

This principle is strictly related to a set of conditions to account for, its constraints, disadvantages and benefits. First, the goals of the project need to be defined, and the planning approach has to follow a strategic investment choice. In particular, we also consider all the positive effects registered in the area where the investment takes place. The required activities to obtain these results are:

- identify and evaluate the main constraints and risks;
- identify and evaluate the needs of the involved organization;
- identify and prioritize the key objectives;
- provide a priority framework.

The stakeholders with strategical information are considered as users of the framework. They are in charge to make changes and to approve the choices of a specific project.

The methodology is structured according to four distinct phases.

3.1 Filtering

The first problem to deal with in an evaluation methodology is the one connected to the choice of the candidate projects to introduce in the following assessment phases. In fact, it turns out unthinkable and totally counter productive to carry out any kind of evaluation on an excessive number of solutions.

From these considerations it follows that necessarily before the real evaluation phase it's needful a selection phase that allows to reduce in a consisting way the number of options.

The first operation to complete in such process is therefore the choice of which are the projects that satisfy the minimums requirements (which does not come within a quality evaluation or however a detailed evaluation) connected to particular requirement of the organization that carries out the evaluation. This operation is of fundamental importance and must be as simple and fast as possible as it allows to reduce considerable and in an immediate way the solutions to take under investigation in the real phase of evaluation, with consequent cost reduction in terms of time and resources of the entire process.

The critical elements of the census operation can themselves be summarized in two main aspects: what to search for (domain understanding) and where to search (identification of possible repositories and citation of eventual studies).

Such information will allow to construct a general profile on the solution of our interest and therefore to define an Identity Card of the project by means of which we will be able to carry out a totally qualitative but highly effective evaluation for the filtering operations.

3.2 Analysis

From the user point of view this phase aims to understand which solution can satisfy the needs of the organization considered in order to guarantee an effective and efficient productivity.

About the needs analysis, the key is to seek the gap between the current situation and the desired situation and then to focus resources where they're most needed. The analysis must determine root causes.

The approach wants to give a correct and complete identification of the business objectives that we want to get, with the focus on the needs to satisfy. The output will be established by a hierarchy based on priority of objectives, obtained from strategic evaluations and the domain characteristic and constrains.

The process of needs definition has been divided in three main steps. The first step has involved an analysis of the literature and of the Italian and international regulation and guidelines. The second one has involved the creation of a survey for the decision makers according to the results of the phase one. In the third step the stakeholders define the more interesting needs and its priority based on the needs knowledge base.

In the third step we can record the necessity that the survey has to have some open areas to identify other potential needs not yet found or specific of the organization. The complete individuation of the survey areas is still on and can be modified.

The surveys have been organized in two macro-areas, they recognize the main type of needs. Therefore the final components, based on the information are the following:

- technical and functional analysis;
- economical and social analysis.

3.3 Measurement

The measurable elements come from the needs. FAME turns the high level stakeholder's evaluations in technical-functional values and economical-social values. A weight and a metric is associated to these elements and the comparative analysis among the solutions is possible.

The technical-functional elements are classic in literature, and so we show only the economical-social elements concerning the evaluation of potential benefits for the citizens, the enterprises and local organizations. They take in consideration the social and educational elements, so like the offer to wide access to information, the increase of capacity and information skills of the citizens. But it is very important to consider all business implications around the territory, like the increase of local capacity and skills with important repercussions on the development of the local enterprises.

We have adopted the procedures of the economical analysis following the cost-benefit analysis. This analysis is very difficult and expensive to follow in-depth way, so we have adopted only the methodological approach. This approach is used to make firstly an estimate of benefits, secondly of costs, then both, also those intangibles. Because a cost-benefit model shows if the system benefits justify your implementation, so it is necessary firstly to make clear the value coming from the adoption of a solution rather than another. This analysis aims to find the typical real costs of a project choice and to evaluate and compare incidental saving costs for the public administrations, so the public administrations with these savings can offer further services. A correct procedure of an economic comparison should be completed both with the starting costs and the services costs to the support, the training, but also migration, installation and management costs, adapting, maintaining, and so on.

The costs compared to benefits are simpler to find. The right costs analysis should take in account also the TCO (Total Cost of Ownership). It considers all direct and indirect costs. All software has a TCO including the price of sell, hardware and software upgrades, maintenance, technical support and learning (time and frustration are complex to measure). After we have assigned a cost to each single item, this cost will be normalized. Here we suggest a possible solution and how our normalized score is calculated:

$S = W*(Cmin + Cmax - Cij)/Cmax$

let be: *[S]* normalized score; *[W]* maximum assignable score; *[C_min]* lower price; *[Cmax]* higher price; *[Cij]* price to be normalized.

FAME associates to the need of type outsourcing the technical-functional elements of type "supplier reliability" with high weight. This is because the organization, which we have analyzed, assigned a high priority to that need.

FAME relates to each Need being in the questionnaire a set of Measurable elements with correspondent metrics, so to each element *Ei* is related an objective metrics of evaluation *Mi*.

3.4 Evaluation

The decision choice should be taken by comparing the values of the needs among the different solutions, and using the weights and the objective metrics for the assessment of the found elements. In order to compare so different objects, we choose to adopt a systematic comparison among the scores of the solutions.

The approach is based on associating a weight to a value, it reflects the relative weight of the value in the overall assessment in accordance with $\sum Wi=Wtot$, where *Wi* is the weight associated with the element *Ei* and *Wi /Wtot* is the relative weight in the assessment based on the importance coming from the needs analysis and the priority one's associated. The priority comes from the evaluation given in the questionnaire to the need from which the value from come.

From this phase FAME obtains the table that is instanced with the analyzed need and it shows how also the weights depend on the needs. Then we compare the different eligible solutions. The project choice will be determined throughout the metrics *M* used with the weights *W* associated with the element *E*, in accordance with FAME. Each solution will have a final score like summation of the scores assigned with each evaluated element. Finally the organization will choose the project solution with the major score.

Practically, a score *Pij* is assigned to each proposed solution *Sj*, where $0<=Pij <=Wi$ for each element *Ei* based on metric Mi associated with that element. Then each solution will have a total score *Pjtot*. The solution with the major score *Pjtot* will be the chosen solution, in fact it will satisfy much more the needs of the administration. The solution with the maximum *Ptot* will be the best because the evaluation comes from the needs analysis and elements measurement and the assessment of that organization.

3.5 The support tool

Purpose of the support tool is to simplify and automate as much as possible the application of a determined evaluation methodology. The tool have to be easy to use for non professional people and guide the users in the selection among various software alternatives and it will have to assist him in the evaluation phase.

This tool should implement the steps of the FAME framework, trying to follow the logic of the simplified method and proposing automatism that allows a faster application and supplies support to who will be engaged in the selection of the new IT solution.

In the Filtering phase the system have to support the user in the collection of information about the candidate products through an user interface of data insertion and a module of data import from repositories of open source projects, like SourceForge.

The user can define and insert through an iterative filtering process, further characteristics with relative values and in this way select further candidate projects, all through comfortable interface of selection.

In the Analysis phase the user has to be guided through a tree in which the leaves represent the needs to measure and the nodes the several macro area of such needs.

In the Measurement phase the user has to be guided through a wide view of what have to be evaluated and with which parameters such process must be executed.

Finally, in the Evaluation phase the collected data will have to be inserted, after normalization, in an appropriate mask created on purpose by means of the list of the candidate software and the list of the metrics selected in the previous steps; the application will calculate the result taking into account the expressed priorities.

4 Conclusion

In this paper we presented FAME, a simplified methodology for F/OSS project assessment, and the description of a possible support tool.

FAME is derived from quite heavyweight methodologies, but was simplified in order make simpler its use. FAME is a new-generation evaluation methodology, able to be tuned to specific needs and without fixed and pre-defined score systems.

Its main original characteristics are: the iterative approach, the integration with a support tool, a supported filtering phase to pre-select project candidates, with automatic collection of data, easing the whole evaluation process; an analysis of the needs of the organization performed and in practice able to configure the evaluation process according to the actual organization's needs, both technical and economic; a phase that explicitly considers various viable metrics, including possible quantitative measurements of software repositories and Web hits related to the projects; a final evaluation phase that blends together different metrics, making them comparable and consistent, and performs the final evaluation.

References:

[1] Feller, J., Fitzgerald, B., Hissam, S. and Lakhani, K., Perspectives on Free and Open Source Software, *MIT Press,* Cambridge, MA, 2005.

[2] Free/Libre and Open Source Software: Survey and Study: Final Report", 2003, available at: http://FLOSS.infonomics.nl/

[3] Senyard, A. and Michlmayr, M., How to have a successful free software project., *APSEC, IEEE Computer Society*, pp. 84–91, 2004.

[4] Antoniades, I. P., Stamelos, I., Angelis, L. and Bleris, G. L., A novel simulation model for the development process of open source software projects, *International Journal of Software Projects: Improvement and Practice (SPIP), special issue on Software Process Simulation and Modeling*, 2003.

[5] Feller, J. and Fitzgerald, B., A framework analysis of the open source software development paradigm, *ICIS 2000*, pp. 58–69.

[6] Gonzalez-Barahona, J. M., Pérez, M. A. O., Quiros, P. d. l. H., Gonzalez, J. C. and Olivera, V. M., Counting potatoes: the Size of Debian 2.2, *Upgrade - The European Online Magazine for the IT Professional*, Vol. II, No. 6, December 2001, pp. 61-67.

[7] Mockus, A., Fielding, R. T. and Herbsleb, J., A case study of open source software development: the Apache server, *ICSE '00: Proceedings of the 22nd international conference on Software engineering*, *ACM Press*, New York, NY, USA, 2000, pp. 263-272.

[8] QUALOSS, http://www.qualoss.org, 2008.

[9] FLOSSMetrics, http://flossmetrics.org, 2008

[10] SQO-OSS, http://www.sqo-oss.eu, 2008.

[11] QUALIPSO, http://www.qualipso.org, 2008.

[12] Michlmayr, M., Software process maturity and the success of free software projects, in Zielinski, K. and Szmuc, T. (Eds.), *Software Engineering: Evolution and Emerging Technologies*, *IOS Press*, 2005, pp. 3-14.

[13] Stewart, K. J. and Ammeter, T., An exploration study of factors influencing the level of vitality and popularity of open source projects, in. Applegate, R. L and De Gross, J. I. (Eds.), *Proceedings of the Twenty-Third International Conference on Information Systems*, 2002, pp. 853-857.

[14] Weiss, D., A large crawl and quantitative analysis of open source projects hosted on sourceforge, *Research Report RA-001/05*, Institute of Computing Science, Poznań University of Technology, Poland, 2005.

[15] Weiss, D., Measuring success of open source projects using web search engines, *OSS2005, Proceedings of the The First International Conference on Open Source Systems*, 2005, pp. 93-99.

[16] Crowston, K., Annabi, H. and Howison, J., Defining open source software project success, *International Conference on Information Systems* (*ICIS*)*,* 2003, pp. 327–340.

[17] Crowston, K., Annabi, H., Howison, J. and Masango, C., Towards a Portfolio of FLOSS project success measures, in *Workshop on Open Source Software Engineering, International Conference on Software Engineering*, Edinburgh, Scotland, UK, 2004. From http://flosspapers.org/180

[18] Cau, A., Concas, G. and Marchesi, M., Extending OpenBRR with automated metrics to measure object oriented open source project success, *The Workshop on Evaluation Frameworks for Open Source Software,*

collocated in *The Second International Conference on Open Source Systems*, 2006.

[19] Ciolkowski, M. and Soto, M., Towards a Comprehensive Approach for Assessing Open Source Projects, in *Proceedings of the international Conferences on Software Process and Product Measurement* (Munich, Germany), Lecture Notes In Computer Science, SpringerVerlag, Berlin, Heidelberg, Vol. 5338, 2008, pp. 316-330.

[20] Deprez, J. C., Alexandre, S., Comparing assessment methodologies for free/open source software: OpenBRR & QSOS, *Lecture Notes in Computer Science*, Springer, 2008.

[21] Method for Qualification and Selection of Open Source software (QSOS), version 1.6, Atos Origin, 2006, available at: http://qsos.org

[22] Cau, A., EFFLOSS: An Evaluation Framework for Free/Libre Open Sourc*e*, PhD Thesis, 2007.

[23] Mannaro, K., Concas, G., Marchesi, M., NVAF: un Framework per una valutazione di tipo comparativo delle soluzioni software nelle Pubbliche Amministrazioni, *The Second International Conference on Open Source Systems*, Esperta Workshop, Como, Italy, 2006.

[24] Business Readiness Rating, A Proposed Open Standard to Facilitate Assessment and Adoption of Open Source Software, 2005, available at http://www.openbrr.org

[25] Golden, B., *Succeeding with Open Source*, Addison-Wesley Professional, 2004.

[26] Open Source Maturity Model, 2000, available at: http://www.seriouslyopen.org

[27] Jones, C., *Software Assessments, Benchmarks, and Best Practices*, Addison-Wesley Information Technology Series, 2000.