EVALUATING THE QUALITY OF FREE/OPEN SOURCE ERP SYSTEMS

Lerina Aversano, Igino Pennino and Maria Tortorella
Department of Engineering, University of Sannio, via Traiano 82100, Benevento, Italy

Keywords: Software Evaluation, Open Source, Enterprise Resource Planning, Software Metrics.

Abstract: The selection and adoption of open source ERP projects can significantly impact the competitiveness of organizations. Small and Medium Enterprises have to deal with major difficulties due to the limited resources available for performing the selection process. This paper proposes a framework for evaluating the quality of Open Source ERP systems. The framework is obtained through a specialization of a more general one, called EFFORT (Evaluation Framework for Free/Open souRce projects). The usefulness of the framework is investigated through a case study.

1 INTRODUCTION

Adoption of an Enterprise Resource Planning system could represent an important competitive advantage for a company, but it could be also useless or even harmful if the system does not adequately fit the organization needs. Then, the selection and adoption of such a system cannot be faced in a superficial way. (Fui-Hoon Nah, 2002) schematically summarizes advantages and disadvantages of adopting an ERP system.

Actually, Small and Medium Enterprises – SMEs – have to deal with major difficulties as they have few resources to dedicate to selection, acquisition, configuration and customization of such a complex system. Moreover, ERPs are generally designed to fit needs of big companies. Adoption of Free/Open Source – F/OSS – ERP could partially fill up this gap. With reference to the adoption procedure, from the literature, it emerges that F/OSS ERP are more advantageous for SME (Hyoseob, 2005) (Wheeler, 2009). As an example, the possibility of really trying the system (not just by using a demo), reduction of vendor lock-in, low license cost and possibility of in-depth personalization are some of the advantages.

This paper proposes a framework for the evaluation of F/OSS ERP systems. The framework is obtained through a specialization of a more general one, called EFFORT – Evaluation Framework for Free/Open souRce projects – defined for evaluating open source software projects (Aversano, 2010).

EFFORT is conceived to properly evaluate F/OSS projects and has been defined following the Goal Question Metric (GQM) paradigm (Basili, 1994).

The rest of the paper is organized as follow: Section 2 is dedicated to the analysis of existing models and tools for evaluating and selecting F/OSS projects and ERP systems; Section 3 provides a description of EFFORT and its specialization for evaluating ERP system; Section 4 presents a case study, consisting of the evaluation of Compiere (www.compiere.com), a F/OSS ERP project; concluding remarks are given in the last section.

2 RELATED WORKS

A lot of work has been done for characterizing and evaluating the quality of F/OSS projects.

In (Kamseu, 2009) Kamseu and Habra analyzed the different factors that potentially influence the adoption of an open source software system. They identified a three dimensional model for the quality of open source projects and stated that by focusing on the quality of the development process, community and product, allows achieving a good global project quality.

In (Sung, 2007) Sung, Kim and Rhew focused on the quality of the product adapting ISO/IEC 9126 standard (ISO, 2001) to F/OSS products. Wheeler defined a F/OSS selection process, called IRCA, based on a side by side comparison of different

Aversano L., Pennino I. and Tortorella M. (2010).
EVALUATING THE QUALITY OF FREE/OPEN SOURCE ERP SYSTEMS.
In *Proceedings of the 12th International Conference on Enterprise Information Systems - Databases and Information Systems Integration*, pages 75-83
DOI: 10.5220/0002971900750083

Table 1: Common Elements.

CRITERIA	MODEL					
	Birdogan Kemal	Evaluation Matrix	OSERP Guru	Reuther Chattopadhyay	Zirawani Salihin Habibollah	Wei Chien Whang
Functionality	√	√	√	√		√
Usability	√	√	√		√	√
Costs	√	√	√	√		√
Support Services	√	√			√	√
Vendor's vision	√					
System reliability	√		√	√		√
Interoperability	√					
Market share	√	√				√
Domain knowledge of providers	√		√			
References and reputation of vendors	√					√
Partnership	√					
Integration/Modularity	√		√		√	
Implementation time	√					√
Software methodology	√					
Consulting	√					
Customization and flexibility		√	√	√	√	
Migration		√	√		√	
Technical quality	√	√		√		
Develop activity			√			
Community			√			
Business competitive advantage				√		

software (Wheeler, 2009). The acronyms IRCA comes from the main steps of the selection process: Identify, Read reviews, Compare, and Analyze.

QSOS – Qualification and Selection of Open Source software – proposes a 5-steps methodology for assessing F*l*OSS projects, defined to make reusable evaluations (QSOS, 2006). The OpenBRR project – Business Readiness Rating for Open Source – has been proposed with the same aim of QSOS (OpenBRR, 2005). This project asks for the execution of the following high level steps: Perform a pre-screening (Quick Assessment); Tailor the evaluation template (Target Usage Assessment); Data Collection and Processing; and Data Translation.

QualiPSo – Quality Platform for Open Source Software – is one of the biggest initiatives related to open source software realized by the European Union. It defines, among other things, an evaluation framework for the trustworthiness of F*l*OSS projects (Del Bianco, 2008).

Generally speaking, some models mostly emphasize product intrinsic characteristics and, only in a small part, the other F*l*OSS dimensions. Vice versa, models have been proposed that try to deeply consider F*l*OSS aspects, offering a reduced coverage to the evaluation of the product.

Regarding the specific context of ERP systems, different collections of criteria for evaluating an open source system were proposed. Some approaches generically regard ERP systems, other ones are specifically referred to F*l*OSS ERPs.

Based on a set of aspects to investigate in a software system, Birdogan and Kemal (Birdogan, 2005) propose an approach identifying and grouping the main criteria for selecting an ERP system.

Evaluation-Matrix (http://evaluation-matrix.com) is a platform for comparing management software systems. The approach follows two main goals: constructing a list of characteristics representing the most common needs of the user; and having at disposal a tool for evaluating available software systems.

Open Source ERP Guru (Open Source ERp Guru, 2008) is a web site offering a support to the users in the identification of an ERP open source solution to be adopted in their organization. It aims at providing an exhaustive comparison among open-source ERP software systems.

In (Reuther, 2004), Reuther and Chattopadhyay performed a study for identifying the main critical factors for selecting and implementing an ERP system to adopt within a SME. The identified factors were grouped in the following categories: technical/functional requirements, business drivers, cost drivers, flexibility, scalability, and other ones specific to the application domain. This research was extended by Zirawani, Salihin and Habibollah (Zirawani, 2009), that reanalyzed it by considering the context of F*l*OSS projects.

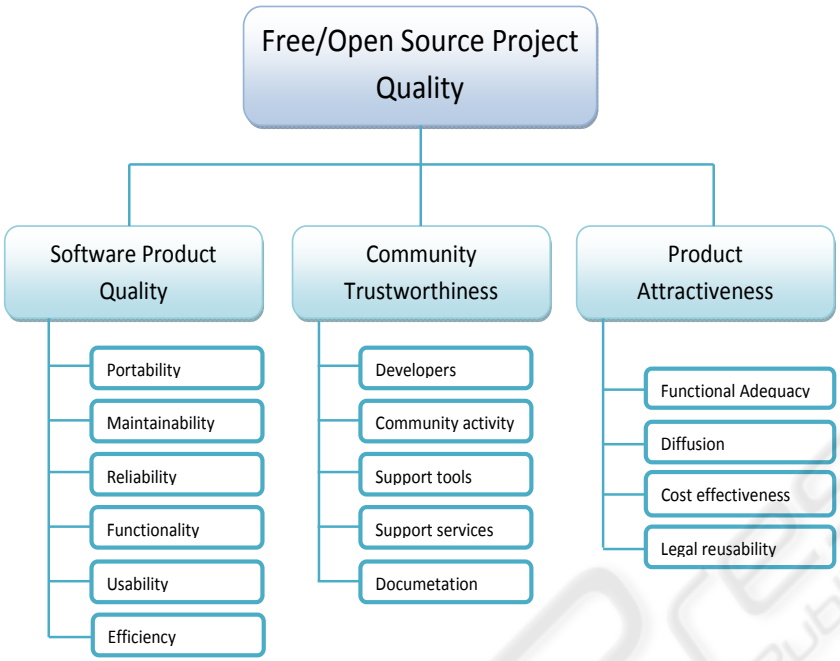

Figure 1: Quality Model defined by EFFORT.

Wei, Chien and Wang (Wei, 2005) defined a framework for selecting ERP system based on the AHP – Analytic Hierarchy Process – technique. This is a technique for supporting multiple criteria decision problems, and suggests how determining the priority of a set of alternatives and the importance of the relative attributes.

In Table 1, a comparison among the models refereed to ERP system is shown for identifying common elements.

The analyzed models are quite heterogeneous, but they have the common goal of identifying critical factors for the selection of ERP systems. The rows of the matrix in Table 1 contain aspects considered in at least one model, while columns refer to the models themselves. The presence of a tick in cell *i,j* means that factor *i* is totally or partially covered by model *j*.

One can observe that Birdogan and Kemal model is the most complete. Criteria considered from the highest number of models regard functionality, usability and costs, followed by support services, system reliability and customizability.

The aim of this paper is to propose an additional framework that oversee the limitations of the previous models. It will represent an exhaustive solution for evaluating the quality of F*l*OSS ERP system with reference to both product and community.

3 PROPOSED APPROACH

In order to adequately support the evaluation of F*l*OSS ERPs, it is necessary to consider that basically these systems belongs to F*l*OSS projects and, obviously, that they are enterprise systems with a specific operative domain. In this direction, the evaluation framework EFFORT, defined for evaluating F*l*OSS systems (Aversano, 2010), has been considered as base framework to be specialized to the context of ERP systems.

As told in the introduction, EFFORT has been defined on the basis of the GQM paradigm (Basili, 1994). This paradigm guides the definition of a metric program on the basis of three abstraction levels: Conceptual level, referred to the definition of the *Goals* to be achieved by the measurement activity; Operational level, consisting of a set of *Questions* facing the way the assessment/achievement of a specific goal is addressed; and Quantitative level, identifying a set of *Metrics* to be associated to each question.

The GQM paradigm helped defining a quality model for F*l*OSS projects, providing a framework to actually using during the evaluation. It considers the quality of a F*l*OSS project as synergy of three main elements: *quality of the product* developed within the project; *trustworthiness of the community* of developers and contributors; and *product attractive-*

ness to its specified catchment area.

Figure 1 shows the hierarchy of attributes that composes the quality model. In correspondence to each first-level characteristics, one *Goal* is defined. Then, the EFFORT measurement framework includes three goals. *Questions*, consequentially, map second-level characteristics, even if, considering the amount of aspects to take into account, *Goal 1* has been broken up into sub-goals, because of its high complexity. For question of space, the figure does not present the third level related to the metrics used for answering the questions.

The following subsections summarily describe each goal, providing a formalization of the goal itself, incidental definitions of specific terms and list of questions. The questions listed in each subsection are can be answered through the evaluation of a set of associated metrics. For reason of space, the paper does not present the metrics, even if some references to them are made in the final subsection. This subsection discusses how the gathered metrics can be aggregated for quantitatively answering the questions.

3.1 Product Quality

One of the main aspects that denotes the quality of a project is the product quality. It is unlikely that a product of high and durable quality has been developed in a poor quality project. So, all the aspects of the software product quality have been considered , as defined by the ISO/IEC 9126 standard.

Goal 1 is defined as follows:

> *Analyze the **software product** with the aim of **evaluating** its **quality**, from the **software engineering**'s point of view.*

Table 2 shows all sub-goals and questions regarding *Goal 1*. As it can be noticed almost all the attributes that the questions reference regards the ISO 9125 standard.

In order to avoid to weight down the exposition, not all metrics of the framework are reported.

3.2 Community Trustworthiness

With *Community Trustworthiness*, it is intended the degree of trust that a user can give to a community, about the offered support. Support can be provided by communities by means of: good execution of the development activity; use of tools, such as wiki, forum, trackers; and provision of services, such as maintenance, certification, consulting and outsourcing, and documentation.

Goal 2 is defined as follows:

> *Analyze the offered **support** with the aim of **evaluating** the **community** with reference to the **trustworthiness,** from the **(user/organization) adopter**'s point of view.*

Questions about *Goal 2* are shown in Table 3.

Table 2: Questions about Product Quality.

Sub-goal 1a: *Analyze the software product with the aim of evaluating it as regards portability, from the software engineering's point of view*	
Q 1a.1	What degree of adaptability does the product offer?
Q 1a.2	What degree of installability does the product offer?
Q 1a.3	What degree of replaceability does the product offer?
Q 1a.4	What degree of coexistence does the product offer?
Sub-goal 1b: *Analyze the software product with the aim of evaluating it as regards maintainability, from the software engineering's point of view*	
Q 1b.1	What degree of analyzability does the product offer?
Q 1b.2	What degree of changeability does the product offer?
Q 1b.3	What degree of testability does the product offer?
Q 1b.4	What degree of technology concentration does the product offer?
Q 1b.5	What degree of stability does the product offer?
Sub-goal 1c: *Analyze the software product with the aim of evaluating it as regards reliability, from the software engineering's point of view*	
Q 1c.1	What degree of robustness does the product offer?
Q 1c.2	What degree of recoverability does the product offer?
Sub-goal 1d: *Analyze the software product with the aim of evaluating it as regards functionality, from the software engineering's point of view*	
Q 1d.1	What degree of functional adequacy does the product offer?
Q 1d.2	What degree of interoperability does the product offer?
Q 1d.3	What degree of functional accuracy does the product offer?
Sub-goal 1e: *Analyze the software product with the aim of evaluating it as regards usability, from the user's point of view*	
Q 1e.1	What degree of pleasantness does the product offer?
Q 1e.2	What degree of operability does the product offer?
Q 1e.3	What degree of understandability does the product offer?
Q 1e.4	What degree of learnability does the product offer?
Sub-goal 1f: *Analyze the software product with the aim of evaluating it as regards efficiency, from the software engineering's point of view*	
Q 1f.1	How the product is characterized in terms of time behaviour?
Q 1f.2	How the product is characterized in terms of resources utilization?

Table 3: Questions about Community Trustiworthiness.

Q 2.1	How many developers does the community involve?
Q 2.2	What degree of activity has the community?
Q 2.3	Support tools are available and effective?
Q 2.4	Are support services provided?
Q 2.5	Is the documentation exhaustive and easily consultable?

3.3 Product Attractiveness

The third goal has the purpose of evaluating the attractiveness of the product toward its catchment area. The term *attractiveness* indicates all the factors that influence the adoption of a product by a potential user, who perceives convenience and usefulness to achieve his scopes.

Goal 3, related to product attractiveness, is formalized as follows:

> *Analyze* ***software product*** *with the aim of* ***evaluating*** *it as regards the* ***attractiveness*** *from the* ***(user/organization) adopter****'s point of view.*

Two elements that have to be considered, during the selection of a F*l*OSS product, are *functional adequacy* and *diffusion*. The latte could be considered as a marker of how the product is appreciated and recognized as useful and effective. Other factors that can be considered are *cost effectiveness*, estimating the *TCO* (Total Cost of Ownership) (Kan, 1994), and the *type of license*.

Questions for Goal 3 are shown in Table 4.

Table 4: Questions about Product Attractiveness.

Q 3.1	What degree of functional adequacy does the product offer?
Q 3.2	What degree of diffusion does the product achieve?
Q 3.3	What level of cost effectiveness is estimated?
Q 3.4	What degree of reusability and redistribution is left by the license?

This goal is more dependent from application context respect the others. That is why every kind of software products come to life to satisfy different needs. With reference to a real-time software, for instance, the more is efficient the more is attractive. Such a thing is not necessarily true for a word processing software, to which the user requires ease of use and compliance of de facto standards.

For this reasons, Goal 3, that mainly regards the way a software system should be used for being attractive, strongly depends on the application domain of the analysed software system and needs a customization to the specific context.

Therefore, the EFFORT framework was extended and customized for making it to be customized to the context of the ERP systems for taking into account additional attraction factors that are specific to this context. The customization of EFFORT required the insertion of additional questions referred to *Goal 3*. In particular following aspects were considered:

- *Migration between different versions of the software*, in terms of support provided for switching from a release to another one. In the context of ERP systems, this cannot be afforded like a new installation, because it would be too costly, taking into account that such a kind of systems are generally profoundly customized and host a lot of data;
- *System population*, in terms of support offered for importing big volumes of data into the system;
- *System configuration*, intended as provided support, in terms of functionality and documentation, regarding the adaption of the systems to specific needs of the company, such as localization and internationalization. Higher the system configurability, lower the start-up time;
- *System customization*, intended as support provided, without direct access to source code, for doing alteration to the system, such as the definition of new modules, installation of extensions, personalization of reports and possibility for creating new workflows. This characteristic is very desirable in ERP systems.

Table 5 shows questions that extend *Goal 3*. As it can be noticed, the new questions are referred to the listed characteristics.

Table 5: Specialization of EFFORT for evaluating ERP systems.

Q 3.5	What degree of support for migration between different releases is it offered?
Q 3.6	What degree of support for population of the system is it offered?
Q 3.7	What degree of support for configuration of the system is it offered?
Q 3.8	What degree of support for customization of the system is it offered?

3.4 Data Aggregation and Interpretation

Once data are collected by means of the metrics associated to each questions, it is necessary to aggregate them, according to the interpretation of the metrics, so one can obtain useful information for answer the questions themselves. In addition, aggregation of answers gives an indication regarding the achievement of the goal.

In doing aggregation, one has to take into account some issues, listed below:

- Metrics have different type of scale, depending on their nature. Then, it is not possible to directly aggregate measures. To overcome that, after the

measurement is done, each metric is mapped to a discrete score in the [1-5] interval, where: 1 = inadequate; 2 = poor; 3 = sufficient; 4 = good; and, 5 = excellent.

- An high value for a metric can be interpreted in a positive or a negative way, according to the context of the related question; even the same metric could contribute in two opposite ways in the context of two different questions. So, for each metric the interpretation is provided.
- Questions do not have the same relevance in the evaluation of a goal. A relevance is associated to each metric in the form of a numeric value in [1-5] interval. Value 1 is associated to questions with minimum relevance, while value 5 means maximum relevance. The definition of the relevance markers depend on the experience and knowledge of the software engineer. In the baseline version of EFFORT, this feature tries to consider relevance of quality characteristics respect to F*l*OSS context. They can change when the customized framework is applied. In fact, their definition should depend on the king of software system to be evaluated and its application domain. Then, a second level of relevance indicators has been added to consider the relevance of the F*l*OSS projects in ERP context.

For the aggregation a specific function is defined so that it takes into account the observations above. In particular, some it is set that:

- $rFlOSS_{id}$ represents the relevance indicator in F*l*OSS context associated with question *id* (sub-goal for Goal 1);
- $rERP_{id}$ indicates the relevance indicator in ERPs context associated with question *id* (sub-goal for Goal 1);
- Q_g is the set of questions (sub-goals for goal 1) related to Goal *g*;

The aggregation function for the Goal *g* is defined as follows:

$$q(g) = \frac{[\sum_{id \in Qg}(rFlOSS_{id} + rERP_{id}) * m(q)]}{\sum_{id \in Qg}(rFlOSS_{id} + rERP_{id})} \quad (1)$$

where $m(q)$ is the following aggregation function for metrics of question *q*:

$$m(q) = \frac{\sum_{id \in Qg} i(id)*v(id) + [1-i(id)]*[v(id) \bmod 6]}{|M_q|} \quad (2)$$

where $v(id)$ is the score obtained for metric *id* and $i(id)$ is its interpretation:

$$i(id) = \begin{cases} 0 \text{ if metric has negative interpretation} \\ 1 \text{ if metric has positive interpretation} \end{cases} \quad (3)$$

and M_q is the set of metrics related to question *q*.

4 CASE STUDY

In order to assess the usefulness of the EFFORT framework and its customization for the evaluation of F*l*OSS ERP projects, they have been applied for the evaluation of Compiere, a F*l*OSS ERP project. Compiere is one of the most diffused ERP Open Source System. Therefore, it has been considered as a relevant case study for validating the framework applicability. In particular, a comparison has been carried out between the results obtained evaluating Compiere by using the baseline version of the EFFORT framework, and those reached evaluating the system by applying the customized version of the framework.

Table 6: Results regarding Compiere Product Quality.

QUALITY CHARACTERISTIC	RELEVANCE		SCORE	
	FLOSS	ERP	BASELINE	SPECIALIZED
Portability	**3**	**2**	**4,1**	**3,57**
Adaptability			5	3,33
Installability			2,64	2,64
Replaceability			4,67	4,75
Maintainability	**3**	**4**	**2,83**	**2,83**
Analyzability			3	3
Changeability			2,8	2,8
Testability			2,5	2,5
Technology cohesion			3	3
Reliability	**3**	**5**	**4,42**	**4,46**
Robustness Maturity			4,16	4,16
Recoverability			4,67	4,75
Functionality	**5**	**5**	**4,13**	**3,96**
Functional adequacy			3,25	3,25
Interoperability			5	4,67
Usability	**4**	**4**	**3,28**	**3,28**
Pleasantness			2	2
Operability			4	4
Understandability			3,89	3,89
Learnability			3,25	3,25
PRODUCT QUALITY	***EFFORT BASELINE VERSION***		***3,77***	
	EFFORT SPECIALIZED VERSION			***3,66***

In the following, a summary of the evaluation results in table form are given for each goal

described in previous section. The data necessary for the application of the framework were mainly collected by analysing the documentation, software trackers, source code repositories and official web sites of the project. Moreover, some other data were obtained by analyzing the source code and using the product itself. Finally, further data source considered are some very useful web sites, such as *sourceforge.net, freshmeat.net* and *ohloh.net*.

The "in vitro" nature of the experiment did not allow a realistic evaluation of the efficiency, so it has been left out. Tables 6, 7 and 8 synthesize the obtained results. They list all the quality characteristics and the score obtained for each of them by applying the relevance indexes, also listed, of both baseline EFFORT and its customized framework.

Table 6 shows results regarding the quality of Compiere software product.

It can be observed that Compiere product is characterized by more than sufficient quality. With a detailed analysis of the sub-characteristics, it can be noticed that the product offers a good degree of portability and functionality, excellent reliability and sufficient usability. Concerning product quality, the lowest value obtained by Compiere is related to the maintainability.

With reference to the reliability, the characteristic with higher score, a very satisfying value was achieved by the robustness, in terms of age, small amount of post release bugs discovered, low defect density, defect per module and index of unsolved bug. An even higher value was obtained for the recoverability, measured in terms of availability of backup and restore functions and services.

Concerning maintainability, the lower score, it was evaluated mainly using CK metrics (Chidamber, 1991), associated to the related sub-characteristics. For instance, the medium-low value for testability of Compiere depends on high average number of children (NOC) of classes, number of attributes (NOA) and overridden methods (NOM), as well as little availability of built in test functions. Values of cyclomatic complexity (VG) and dept of inheritance tree (DIT) are on the average.

It was observed that global scores obtained with the two different relevance criteria, are substantially the same for Compiere product quality. There is just a little negative variation considering both relevance indexes. Moreover, a better characterization of some aspects was done, knowing the application domain. In fact, other metrics were considered, other than the ones considered by the general version of EFFORT. For that reason, Table 6 presents two columns of scores: the "BASELINE" one is obtained by considering metrics from EFFORT general version only, while the "SPECIALIZED" column contains results from the evaluation by means of all metrics from the EFFORT customized version.

For instance, there are different scores for adaptability and replaceability (and, consequentially, for portability). In fact, the number of supported DBMS and availability of a web client interface were considered for the adaptability characteristic. Whereas, availability of functionality for backup and restore data, availability of backup services and numbers of reporting formats have been taken into account for the replaceability characteristic. Those aspects are not significant for other kind of software products.

In Table 7, data regarding community trustworthiness are reported. In the case of Goal 2, as well as in the one of Goal 3, the hierarchy of the considered characteristics has one less level.

Moreover, aspects of this goal are completely generalizable for all F*l*OSS projects so anything of this part of the EFFORT framework changes, but the relevance.

The score obtained by Compiere for community trustworthiness is definitely lower with respect to product quality. In particular, community behind Compiere is not particularly active; in fact, average number of major release per year, average number of commits per year and closed bugs percentage assume low values. Support tools are poorly used. In particular, low activity in official forums was registered. Documentation available free of charge was of small dimension; while the support by services results was more than sufficient, even if it was available just for commercial editions of the product. This reflects the business model of Compiere Inc., slightly distant from traditional open source model: product for free, support with fee. This time the evaluation by means of the specialized version of EFFORT (in this case consisting just of different relevance pattern) gives better results for the Compiere community. That is the main reason for which the availability of support services was considered more important than community activity, in the ERP context.

As mentioned before, product attractiveness is the quality aspect more dependent from operative context of the product itself. In this case, it was extended the relative goal with other four questions, as explained in section 3.3. The aim was to investigate how this can influence the evaluation.

Table 7: Results regarding Community Trustworthiness.

QUALITY CHARACTERISTICS	RELEVANCE		SCORE
	FLOSS	ERP	
Developers Number	2	1	2
Community Activity	4	2	2,60
Support Tools	5	4	2,44
Support Services	2	4	3,44
Documentation	4	4	1,67
COMMUNITY TRUSTWORTHINESS	*EFFORT BASELINE VERSION*		***2,36***
	EFFORT SPECIALIZED VERSION		***2,43***

Results are showed in Table 8. Compiere offers a good attractiveness, especially if the score obtained from the analysis done with the EFFORT customized version is considered. In particular, a sufficient functional adequacy and excellent legal reusability are obtained, because of the possibility left to the users in choosing the license, even if it is a commercial one. Compiere product results quite widespread. The last thing was evaluated measuring: downloads number, freshmeat popularity index, sourceforge users rating number, positive sourceforge rating index, success stories number, visibility on google, official partners number, as well as number of published books, experts review and academic papers.

Table 8: Results about Product Attractiveness.

QUALITY CHARACTERISTIC	RELEVANCE		SCORE	
	FLOSS	ERP	BASELINE	SPECIALIZED
Functional Adequacy	5	5	3,25	3,25
Diffusion	4	3	4	4
Cost Effectiveness	3	5	2,40	3,22
Legal Reusability	1	5	5	5
Migrability	-	5	-	3,67
Data Importability	-	5	-	5
Configurability	-	2	-	3,89
Customizability	-	4	-	4,67
PRODUCT QUALITY	*EFFORT BASELINE VERSION*		***3,42***	
	EFFORT SPECIALIZED VERSION			***3,96***

Concerning costs, EFFORT baseline just considered the possibility to have the product free of charge, and amount to be spent for an annual subscription. As this is not sufficient for ERP systems, a customization was considered for including also costs for customization, configuration, migration between releases and population of the system. This explains different the scores about cost effectiveness in Table 8. The characteristics described above are also independently considered.

As an ERP system, Compiere provides an excellent customizability and data importability, as well as a good configurability and migrability. High values for those characteristics contribute to increment attractiveness, that goes from 3,42 to 3,96.

5 CONCLUSIONS

The introduction of an ERP system into an organization can lead to obtain an increase of its productivity, but it could also be an obstacle, if the implementation is not carefully faced. The availability of methodological and technical tools for supporting the adoption process is desirable.

The work presented in this paper is related to the presentation of EFFORT, a framework for the evaluation of F/OSS projects, and its customization to explicitly fit the ERP software system domain. The specialization mainly regarded product attractiveness characterization.

The proposed framework is compliant with ISO/IEC 9126 standard for product quality. In fact it considers all of characteristics defined by the standard model, but in-use quality. Moreover, it considers major aspects of F/OSS projects and, has been specialized for ERP systems.

The applicability of the framework is described through a case study. Indeed, EFFORT was used to evaluate Compiere, one of the most diffused F/OSS ERP. The obtained results are quite good for product quality and product attractiveness. They are less positive with reference the community trustworthiness.

Future investigation will regard the integration in the framework of a questionnaire for evaluating customer satisfaction. This obviously includes more complex analysis. In particular, methods and techniques specialized for exploiting this aspect will be explored and defined.

In addition, the authors will continue to search for additional evidence of the usefulness and applicability of the EFFORT and customizations, by conducting additional studies also involving subjects working in operative realities.

REFERENCES

Fui-Hoon Nah, F., 2002. *Enterprise Resource Planning solutions and management*. Idea Group Inc (IGI).

Hyoseob, K., Boldyreff, C., 2005. *Open Source ERP for SMEs*. In *ICMR 2005*, *The International Conference on Manufacturing. Research*. Cranfield University.

Basili, V. R., Caldiera, G., & Rombach, H. D., 1994. *The goal question metric approach*. Encyclopedia of Software Engineering. Wiley Publishers.

Kamseu, F., Habra, N., 2009. *Adoption of open source software: Is it the matter of quality?* PReCISE, University of Namur.

Sung, W. J., Kim, J. H., Rhew, S. Y., 2007. A quality model for open source selection. In *ALPIT 2007*, *Sixth International Conference on Advanced Language Processing and Web Information Technology* IEEE Comp. Soc. press

International Organization for Standardization, 2001-2004. *ISO standard 9126: Software Engineering – Product Quality, part 1-4*. ISO/IEC.

Wheeler, D. A., 2009. *How to evaluate open source software/free software (OSS/FS) programs*. http://www.dwheeler.com/oss_fs_eval.html#support

QSOS, 2006. *Method for Qualification and Selection of Open Source software*. Atos Origin.

OpenBRR, 2005. *Business Readiness for Open Source*. Intel.

Del Bianco, V., Lavazza, L., Morasca, S., Taibi, D., 2008. The observed characteristics and relevant factors used for assessing the trustworthiness of OSS products and artefacts. *QualiPSo*.

Birdogan, B., Kemal, C., 2005. Determining the ERP package-selecting criteria: The case of Turkish manufacturing companies. In Business Process Management Journal, 11(1), Emerald Group Publishing Limited, pp. 75 – 86.

Open Source ERp Guru, 2008. *Evaluation Criteria for Open Source ERP selection*. http://opensource erpguru.com/2008/01/08/10-evaluation-criteria-for-open-source-erp/

Reuther, D., Chattopadhyay, G., 2004. *Critical Factors for Enterprise Resources Planning System Selection and Implementation Projects within Small to Medium Enterprises*. In *International Engineering Management Conference 2004*, IEEE Comp.Soc.press.

Zirawani, B., Salihin, M. N., Habibollah, H., 2009. Critical Factors to Ensure the Successful of OS-ERP Implementation Based on Technical Requirement Point of View. In *AMS 2009*, *3rd Asia International Conference on Modelling & Simulation*, IEEE Comp.Soc.press.

Wei, C. C., Chien, C. F., Wang, M. J. J., 2005. An AHP-based approach to ERP system selection. In *International Journal of Production Economics*. 96(1). Elsevier press, pp. 47-62.

Kan, S. H., Basili, V. R., Shapiro, L. N., 1994. Software quality: an overview from the perspectiveof total quality management. In *IBM Systems Journal*.

Chidamber, S. R., Kemerer, C. F., 1991. Towards a metrics suite for object-oriented design". In *ACM SIGPLAN Notices*, 26(11), ACM press, pp. 197 - 211.

Aversano, L., Pennino, I., Tortorella, M., 2010. Evaluating the Quality Of Free/Open Source Projects. In 5th International Conference on Evaluation of Novel Approaches to Software Engineering.