Applying EFFORT for Evaluating CRM Open Source Systems

Lerina Aversano and Maria Tortorella

Departement of Engineering, University of Sannio
Via Traiano,
82100 Benevento, Italy
{aversano,tortorella}@unisannio.it

Abstract. Free Open Source Software (F*l*OSS) for Customer relationship management (CRM) represents a great opportunity for small and medium enterprises as they can significantly impact their competitiveness. These software solutions can be adopted with success, whatever the size, and offer customized solutions for clients, even with few people working of it. This paper presents an analysis of the quality of CRM F*l*OSS systems using a predefined quality model. The analysis focuses on the main relevant aspect of both open source domain and application domain proposing a trade off assessment among these.

Keywords: FlOSS Free/Open Source Software, Software quality, Software Metrics, CRM Systems, Standard.

1 Introduction

Customer relationship management (CRM) systems are software systems aiming to support enterprises to automate many services from sales to marketing to customer services. They are efficient solutions that, however, were mainly used from big enterprises which could provide deep pockets and time to undertake these huge implementation projects. Nevertheless, even small business organizations can consider the introduction of CRM. The problem is that CRM is too expensive and difficult to be properly implemented, and it seemingly requires radical change. 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 complex systems. Moreover, enterprise software systems are generally designed to fit needs of big companies.

Adoption of Free/Open Source Systems – F*l*OSS – partially fill up this gap. F*l*OSS CRM systems are actually available to any business, whatever the size, and offer customized solutions for clients, even with few people that can be up and running in two to three weeks. This problem was already faced with reference to ERP systems and even in that case the adoption of a F*l*OSS ERP was proved to be very advantageous for SME [2, 3]. As an example, the possibility of really trying the system (not just using a demo), reduction of vendor lock-in, low license cost and possibility of in-depth personalization are some of the advantages.

D. Caivano et al. (Eds.): PROFES 2011, LNCS 6759, pp. 202–216, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Likewise ERP systems, even adopting a Customer Relationship Management systems could represent an important competitive advantage for a company, but it could also be useless or even harmful if the system does not adequately fit the organization needs. Then, the selection and adoption of such a kind of system cannot be faced in a superficial way and evaluation supports are needed. Many quality models for evaluating F*l*OSS systems have been proposed in literature [10-16]. Nevertheless, they do not cover all the relevant aspects of quality and operative context of such systems and are not always applicable to operative contexts. An evaluation of these models is provided in [1].

This paper proposes a framework for a quantitative evaluation of the quality of F*l*OSS CRM systems. The framework is defined by specializing a more general one, called EFFORT – Evaluation Framework for Free/Open souRce projects – defined for evaluating open source software projects [1]. The EFFORT framework was already assessed for ERP Systems [5]. It is conceived to properly evaluate F*l*OSS projects and was defined on the basis of the Goal Question Metric (GQM) paradigm [6].

The rest of the paper is organized as follow: Section 2 presents EFFORT that is a necessary background for discussing the quantitative analysis of the CRM systems; Section 3 provides the specialization of EFFORT for evaluating CRM system; Section 4 presents a case study, regarding the evaluation of 4 open source CRM F*l*OSS projects: SugarCRM (www.sugarcrm.com), CreamCRM (http://sourceforge.net/projects/cream-crm/), Concursive ConnectCRM (www.concursive.com), VTigerCRM (www.vtiger.com). Concluding remarks are given in the last section.

2 Background

EFFORT is a framework defined for evaluating F*l*OSS systems [1]. In this paper. it is considered as a base framework to be specialized to the context of CRM systems.

As told in the introduction, EFFORT was defined on the basis of the GQM paradigm [6]. 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 to define a quality model for F*l*OSS projects, providing a framework to be actually used 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 attractiveness* to its specified catchment area.

Figure 1 shows the hierarchy of attributes that composes the quality model. In correspondence of each first-level characteristic, one *Goal* was defined. Then, the EFFORT measurement framework included three goals. *Questions*, consequentially, mapped second-level characteristics, even if, considering the amount of aspects to take into account, *Goal 1* was 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.

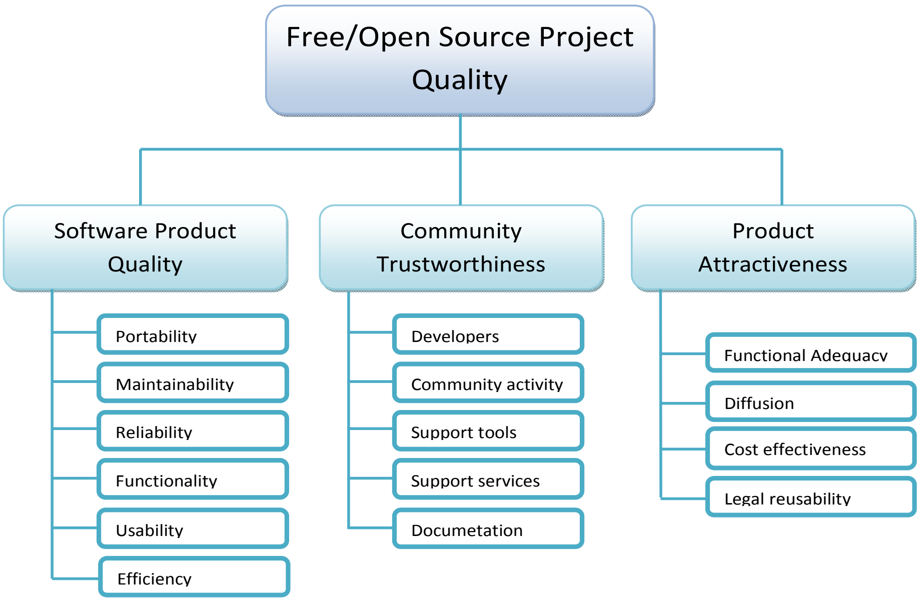

Fig. 1. Hierarchy of the quality model

The following subsections summarily describe the three goals providing their formalization, incidental definitions of specific terms and the list of questions. The listed questions can be answered through the evaluation of a set of associated metrics. For reason of space, the paper does not present all the metrics, and includes some references to them in the final subsection that discusses how the gathered metrics can be aggregated for quantitatively answering the questions.

2.1 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 was developed in a poor quality project. So, all the aspects of the software product quality were considered, as defined by the ISO standard [8, 9].

Goal 1 was defined as follows:

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

Table 1 shows all the 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.

Table 1. Questions about Product Quality

***Goal 1*- Product Quality**	
Sub-goal 1a: *Analyze the software product with the aim of evaluating it with reference to portability, from the software engineer'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 with reference to maintainability, from the software engineer'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 with reference to reliability, from the software engineer'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 with reference to functionality, from the software engineer'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 with reference to 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 with reference to 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?

2.2 Community Trustworthiness

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

Goal 2 was 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 regarding *Goal 2* are shown in Table 2.

Table 2. Questions about Community Trustiworthiness

GOAL 2- **Community Trustiworthiness**	
Id question	**Question**
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?

2.3 Product Attractiveness

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

Goal 3 was related to product attractiveness and formalized as follows:

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

Two elements to be considered for evaluating a F*l*OSS product were *functional adequacy* and *diffusion*. The latter could be considered as a marker of how the product was appreciated and recognized as useful and effective. Other factors that could be considered were *cost effectiveness*, estimating the *TCO* (Total Cost of Ownership) [7], and the *type of license*.

Questions for Goal 3 are shown in Table 3; while, as an example, Table 4 lists the metrics related to question 3.2.

Table 3. Questions regarding Product Attractiveness

GOAL 2- **Community Trustiworthiness**	
Id question	**Question**
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?

Table 4. Metrics related to question Q 3.2

Id Metric	**Metric**
M 3.2.1	Number of thread per year
M 3.2.2	Index of unreplied threads
M 3.2.3	Number of forums
M 3.2.4	Average of threads per forum
M 3.2.5	Average of posts per year
M 3.2.6	Degree of internationalization of the forum
M 3.2.7	Number of trackers
M 3.2.8	Wiki volume
M 3.2.9	Number of frequently asked questions

2.4 Data Aggregation and Interpretation

Once data were collected by means of the metrics, it was necessary to aggregate them, according to their interpretation, for answering the questions and obtaining useful information. Aggregation of answers gives an indication regarding the achievement of the goals.

In doing aggregation, some issues needed to be considered. These are listed below:

- Metrics have different types of scale, depending on their nature. Then, it was not possible to directly aggregate measures. After the measurement was done and to overcome that problem, each metric was 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 could be interpreted in either positive or 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, the appropriate interpretation was provided for each metric.
- As questions did not have the same relevance in the evaluation of a goal, a relevance marker was associated to each metric in the form of a numeric value in interval [1-5]. Value 1 is associated to questions with the minimum relevance, while value 5 means maximum relevance. The definition of the relevance markers depended on the experience and knowledge of the software engineer and the organizational needs and requirements.

The aggregation function for Goal *g* was defined as follows:

$$q(g) = \left[\sum_{id \in Q_g} r_{id} * m(id) \right] / \sum_{id \in Q_g} r_{id}$$

where:

r_{id} is the relevance associated to question *id* (sub-goal for goal 1);

Q_g is the set of questions (sub-goals for goal 1) related to goal *g*.

m(q) is the aggregation function of the metrics of question *q,* defined as:

$$m(q) = \left\{ \sum_{id \in M_q} i(id) * v(id) + [1 - i(id)] * [v(id)\ mod\ 6] \right\} / |M_q|$$

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

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

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

3 EFFORT Specialization

CRM solutions are enterprise software systems whose goal is to learn more about customers' needs and behaviors in order to develop stronger relationships with them, and facilitate acquiring, enhancing and retaining of customers. Although CRM has emerged as a major business strategy for e-commerce, little research was conducted in

Table 5. Integration of the EFFORT specialization

EFFORT Integration for Goal 1			
Id Question	**Question**	**Id Metric**	**Metric**
1a.1	What degree of adaptability does the product offer?	**1a.1.2**	Number of Application Servers supported
1a.3	What degree of replaceability does the product offer?	**1a.3.1**	Availability of functionality for creation of data backup
		1a.3.2	Availability of functionality for restoration of data backup
		1a.3.3	Availability of backup services
		1a.3.4	Number of file formats for the reporting
		1a.3.5	Number formats per the importing of data
1b.1	What degree of analyzability does the product offer?	**1b.1.3**	Number of Package
		1b.1.4	Number of Class
		1b.1.6	Methods for Class
		1b.1.7	Javadoc density (MB)/NOC
1b.2	What degree of changeability does the product offer?	**1b.2.1**	Lack of methods cohesion, as defined by Henderson-Sellers
		1b.2.2	Efferent coupling
		1b.2.3	Afferent coupling
		1b.2.5	Average value of the number of methods per class
1b.3	What degree of testability does the product offer?	**1b.3.2**	Average value of the height of the inheritance tree
		1b.3.3	Average of the Number of subclass for class
		1b.3.6	Average of the Number of attribute for class
		1b.3.7	Average of the Number of override method for class
		1b.3.8	Average number of test drivers respect to the number of classes

evaluating the available CRM solutions. In particular, in the best of the authors' knowledge, research did not propose either comparative analyses of open source solutions or analyses of the quality of such a kind of products. With the aim of overcoming this lack, the EFFORT framework was specialized to this specific application context.

As already stated in the previous section, EFFORT needs to be specialization to the context of the CRM systems before being applied for evaluating such a kind of systems. The specialization tasks was performed at the level of questions or goals. The first kind of task regarded the integration of additional metrics for answering some baseline EFFORT questions; while the intervention at the goal level concerned the extension of some goals with the adding of further questions.

In particular, with reference to the integration, Table 5 shows all the metrics that were added for answering some baseline questions. A large part of integration was performed with reference to Goal 1 and all the choices are explained by the strategic role played in the context of the CRM systems by both data and necessity of integrating an application with the information system of an organization. Therefore, additional metrics were considered for evaluating the Adaptability and Replaceability (and, consequentially, Portability), as it can be evicted from Table 5. In fact, the Number of Application Servers supported was considered for the adaptability characteristic. Whereas, Availability of functionality for backup and restore data, Availability of backup services and Numbers of reporting formats were taken into account for the Replaceability characteristic. Evaluating the Analysability required the addition of a Javadoc density metric as all the analysed CRM systems were based on the Java technology. Moreover, as the analysed software systems were based on the object-oriented paradigm, metrics relate to this paradigm were considered. This required the instantiation of the measurement framework with the adoption of the specific object-oriented metrics.

With reference to Goal 3, Table 6 contains the integration performed for understanding the economical advantage when a CRM F/OSS system is adopted. This advantage does not depend just on the license costs, but also on those costs to be spent

Table 6. Integration for instantiating EFFORT for Goal 3

EFFORT Specialization of Goal 3			
Id Question	**Question**	**Id Metric**	**Metric**
Q 3.3	What degree of economical advantages is estimated?	**3.3.5**	Cost of migration among different versions
		3.3.6	Cost of system population
		3.3.7	Cost of system configuration
		3.3.8	Cost of system customization
Q 3.5	What degree of support for migration between different releases is it offered?	**3.5.1**	Availability of functionality for creation of data backup
		3.5.2	Availability of functionality for restoration of data backup
		3.5.3	Availability of backup services
		3.5.4	Availability of documentation of migration between versions
		3.5.5	Availability of automatic migration tools
		3.5.6	Availability of documentation for migrating the database
Q 3.6	What degree of support for population of the system is it offered?	**3.6.1**	Number of standard formats for importing data
Q 3.7	What degree of support for configuration of the system is it offered?	**3.7.1**	Availability of a wizard for configuring the system
		3.7.2	Number of supported languages
		3.7.3	Availability of documentation for supporting the starting setup
		3.7.4	Availability of documentation for supporting language configuration
Q 3.8	What degree of support for customization of the system is it offered?	**3.8.1**	Availability of functionality for installing an extension from the user interface
		3.8.2	Availability of functionality for creating a new module from the user interface
		3.8.3	Availability of functionality for customization of the user interface
		3.8.4	Availability of functionality for creating customized report
		3.8.5	N° of standard template for the creation of report
		3.8.6	Availability of documentation for the product customization

for both adapting the adopted software system to own needs and maintaining it by installing updated versions or adoption of new releases.

Goal 3 mainly regarded the way a software system should be used for being attractive. Then, it strongly depended on the application domain of the analysed software system and needs a customization to the specific context. Therefore, in the CRM context, the EFFORT framework was extended and customized taking into account additional specific attraction factors by considering additional questions referred to *Goal 3*. In particular the 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 CRM systems, this cannot be addressed like a new installation, as it would be too costly, taking into account that such a kind of system is generally customized and hosts a lot of data.
- *System population*, in terms of support offered for importing big volumes of data into the system.
- *System configuration*, intended as support provided, in terms of functionality and documentation, regarding the adaption of the system 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 the source code, for doing alteration of the system, such as the definition of new modules, installation of extensions, personalization of reports and possibility of creating new workflows. This characteristic is very desirable in CRM systems.

Table 6 shows questions that extend Goal 3. As it can be noticed, the new questions are referred to the listed characteristics.

During the instantiation of EFFORT, a relevance regarding the CRM context was associated to each metric by using numeric values of the [1-5] interval. The EFFORT relevance factors that were additionally considered are the following:

- $rFlOSS_{id}$, representing the relevance indicator in the F*l*OSS context associated with question *id* or sub-goal *id* of Goal 1;
- $rCRM_{id}$, indicating the relevance indicator in the CRM context associated with question *id* or sub-goal *id* of Goal 1;
- Q_g, the set of questions (or sub-goals for Goal 1) related to Goal *g*;

Then, the aggregation function for evaluating Goal *g* is defined as follows:

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

where *m(q)* is the aggregation function of the metrics of question *q*.

4 Results

After specializing the EFFORT framework, an analysis of CRM F*l*OSS was performed. It permitted to choose the following four CRM systems considered for being evaluated: SugarCRM (www.sugarcrm.com), CreamCRM (http://sourceforge.net/

projects/cream-crm/), Concursive ConnectCRM (www.concursive.com), VTigerCRM (www.vtiger.com). These systems were selected from the top ten classification of open source CRM systems [4].

SugarCRM_is an open source Customer Relationship Management (CRM) software that runs on Windows/Linux. SugarCRM is web-based and can be accessed with a web browser locally connected or set up for the internet access by any machine with a browser and an internet connection. SugarCRM for Microsoft Windows requires the open source software Apache Web Server, MySql Database and PHP. Functionality includes sales force automation, marketing campaigns, support cases, project management, calendaring, documents and more.

Cream CRM is a multilingual application designed for supporting the following services: tracks sale orders, payments, shipments, services, online and print subscriptions, and the effectiveness of promotional campaigns. Modules allow communication with customers via newsletters, email and a Web interface. Cream CRM runs on FreeDSB, Linux and Windows 2000/XP. It is written in Java and JavaScript.

vTiger CRM is built upon the LAMP/WAMP (Linux/Windows, Apache, MySQL and PHP) architecture, with the main development team based in Chennai, India. vTiger CRM includes SFA (Sales Force Automation), customer-support and -service, marketing automation, inventory-management, multiple database support, security-management, product-customization, calendaring and email-integration features. It also offers add-ons and support for other add-ons. Vtiger is written in JavaScript, PHP and Visual Basic. It is compatible with ADOdb, MySQL and PostgreSQL databases.

Concursive offers two robust products that enable businesses and other organizations to harness the power of collaboration, leverage social networks and manage their activities. ConcourseConnect is a platform that enables the creation of true commercial networks. Commercial networks are the intersection of social networking, web 2.0 collaboration and sales and marketing tools. ConcourseConnect enables organizations to create dynamic communities to connect various stakeholders and manages the entire ecosystem with an integrated management console and backend CRM tools.

Figure 2 and 3 show the results regarding the product quality obtained by applying the specialized framework. In particular, Figure 2 shows that vTiger CRM exhibits the best product quality. This results are also graphically confirmed by Figure 3 highlighting the detailed results regarding each characteristic of Goal 1. The higher value of the **Portability** was achieved by SugarCRM, that is 3.4. Indeed, all the portability metrics achieved good values for this system. On the contrary, the portability value reached by Cream CRM was 1.83 that is relatively low. This fact mainly depended on the lack of functionality for the creation and backup of data, deficiency of installation manuals, and customization of installation procedures that assumed a default value in just the 25% of cases for the data import/export. The value of Concursive was medium, 2.7. This was mainly due to the high values of metrics M1a3.1, M1a3.2, M1a3.3, M1a3.4, M1a3.5, related to the software replaceability and data import export, and to the low value of metrics, related to question Q1a2, regarding the knowledge required about the third part software system and guide effectiveness.

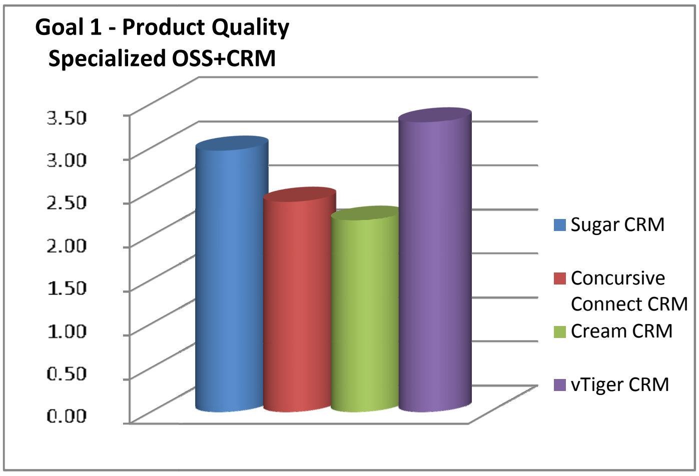

Fig. 2. Overall results obtained with reference to the Goal 1- Product Quality

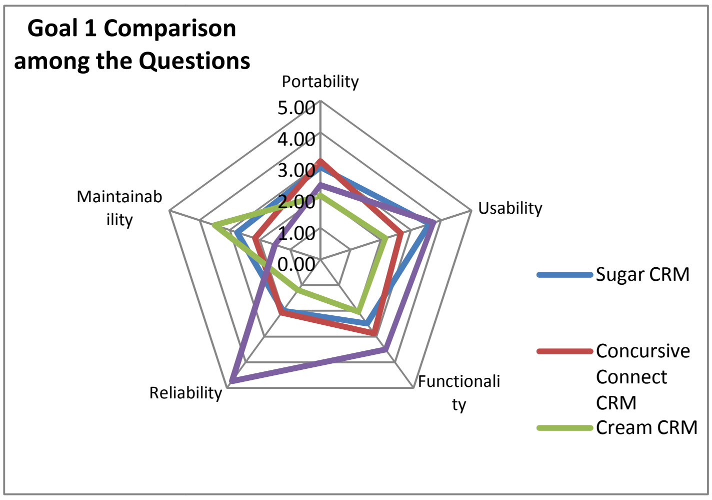

Fig. 3. Detailed results obtained with reference to the Goal 1- Product Quality

With reference to the **Maintainability,** the value achieved by SugarCRM is 2.5, that was higher than the value 3 obtained by CreamCRM. This was due to the value of Methods per Class (M1b1.6), that was very high in CreamCRM, and even to the good value 3 obtained by the index of LOC evaluated in Q1b3. As it can be noticed, Maintainability was very low for vTiger CRM, as it exhibited a very low value of modifiability (Q1.b.2), testability (Q1.b.3) and technology concentration (Q1.b.4).

Reliability generally achieved low values: value 2 for SugarCRM; value 1.7 for Cream CRM. These results were essentially penalized by the lack of backup services and bug tracker, that influence the values of the metrics related to Q1c. In any case,

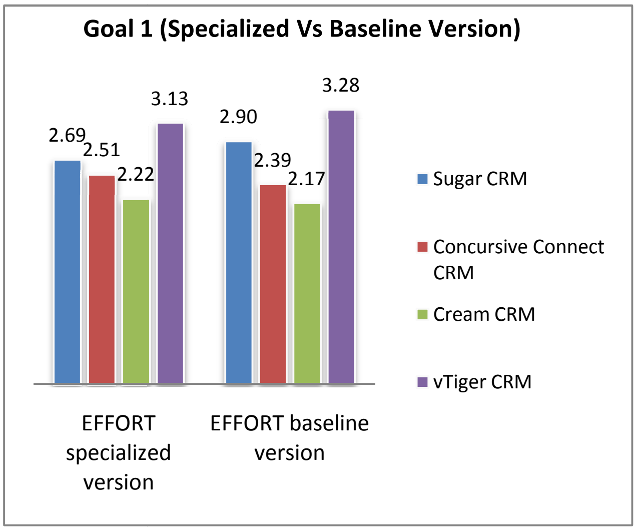

Fig. 4. Comparing the results of the Product Quality between the Specialized Framework and Baseline Framework

Reliability reached a high value for vTiger for the high capability of robustness and recoverability.

The best value regarding the **Functionality** characteristic was reached by vTiger that is 3.33; while the lowest value was reached by Cream CRM, that is 1.97, essentially due to the lack of web services support.

Similar values of **Usability** were reached by SugarCRM and vTiger CRM; while the worst value was still that one obtained by Cream CRM, mainly due to the limited documentation and videos available for the training of the product.

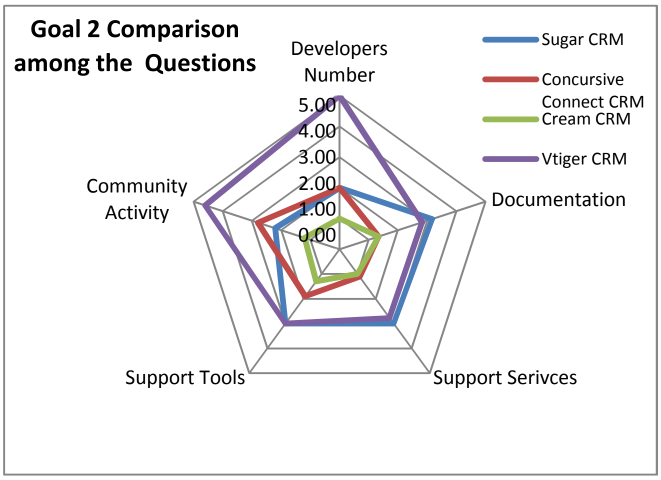

Fig. 5. Comparing the results of the Community Trustworthiness for each question

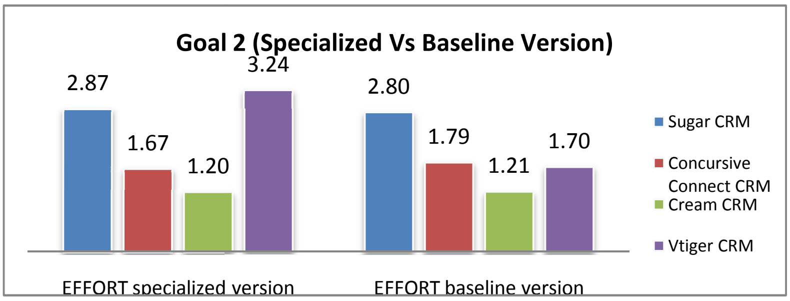

Fig. 6. Comparing the results of the Community Trustworthiness between the Specialized Framework and Baseline Framework

Overall, observing Figure 4, it can be noticed that the CRM system achieving the best results is vTiger CRM, even with the EFFORT specialized version. While the worst results are those of Cream CRM, as emerged by the data analysed for Goal 1.

Moving to the Goal 2, regarding the Community Trustworthiness, the results confirm the outcomes above. Indeed, vTiger CRM obtains the best results even with reference to Goal 2; while the worst value is achieved by Cream CRM that just obtains the value 1.2. The reason of this result is mainly due to the lack of supporting services and poor use of the forums, as shown in Figure 5.

Figure 6 shows that vTiger CRM achieved better results with the EFFORT specialized version instead of the baseline one. Actually, this was due to the better results obtained for those aspects having an higher relevance score in the specialized version.

Results of Goal 3, related to the Product Attractiveness, are shown in Figure 7 and Figure 8. In this case, the differences among the results obtained by the different CRM systems are less relevant. Nevertheless, the overall results confirm that vTiger CRM achieved the best values, and the lowest values are again those of Cream CRM.

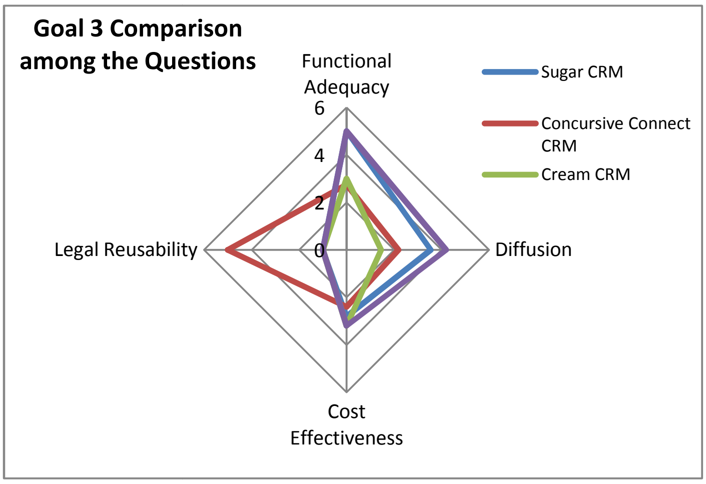

Fig. 7. Comparing the results of the Product Attractiveness for each question

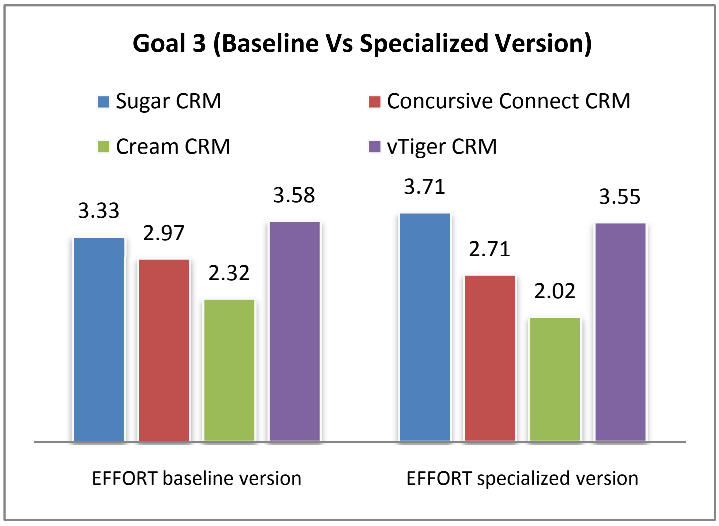

Fig. 8. Comparing the results of the Product Attractiveness between the Baseline Framework and Specialized Framework

Some reasons of the low results of CreamCRM are the following: question Q 3.1 regarding the Functional Adequacy achieved a very low value because not all the modules work correctly; question Q 3.2, regarding the diffusion, reached a low value due to the few visibility on the net; question Q 3.5, regarding the migration support, assumed a low value mainly for the lack of support for automating the migration and data backup.

5 Conclusion

The introduction of an open source CRM system into an organization can lead to the 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 process of evaluating and adopting a CRM system is desirable.

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

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

The usefulness of the framework is described through its concrete application. Indeed, EFFORT was used for evaluating four CRM open source systems, selected among the most diffused F/OSS CRM. The obtained results are quite good for product quality and product attractiveness. They are less positive with reference the community trustworthiness. The interpretation of the obtained results is strictly connected to the adopting enterprise's environment, needs and requirements.

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. In particular, EFFORT will be extended for better considering evolution aspects.

References

1. Aversano, L., Pennino, I., Tortorella, M.: Evaluating the Quality of FREE/OPEN Source Project. In: ENASE (Evaluation of Novel Approaches to Software Engineering) Conferences, INSTICC (2010)
2. Hyoseob, K., Boldyreff, C.: Open Source ERP for SMEs. In: ICMR 2005. Cranfield University (2005)
3. Wheeler, D. A.: How to evaluate open source software/free software (OSS/FS) programs (2009), http://www.dwheeler.com/oss_fs_eval.html#support
4. Hakala, D.: The Top 10 Open-Source CRM Solutions. Focus Expert Network (2009), http://www.focus.com/briefs/crm/top-10-open-source-crm-solutions/
5. Aversano, L., Tortorella, M.: Evaluating the quality of free/Open source systems: A case study. In: Filipe, J., Cordeiro, J. (eds.) ICEIS 2010. Lecture Notes in Business Information Processing, vol. 73, pp. 119–134. Springer, Heidelberg (2011)
6. Basili, V.R., Caldiera, G., Rombach, H.D.: The goal question metric approach. Encyclopedia of Software Engineering. Wiley Publishers, Chichester (1994)
7. Kan, S.H., Basili, V.R., Shapiro, L.N.: Software quality: an overview from the perspective of total quality management. IBM Systems Journal (1994)
8. International Organization for Standardization, 2001-2004. ISO standard 9126: Software Engineering – Product Quality, part 1-4. ISO/IEC
9. International Organization for Standardization, ISO standard ISO/IEC 25000:2005, Software Engineering – Software product Quality Requirements and Evaluation, SQuaRE (2005)
10. Golden, B.: Making Open Source Ready for the Enterprise: The Open Source Maturity Model. In: Succeeding with Open Source, Addison-Wesley Publishing Company, Reading (2005)
11. Kamseu, F., Habra, N.: Adoption of open source software: Is it the matter of quality? PReCISE PReCISE, Computer Science Faculty, University of Namur, rue Grandgagnage, Belgium (2009)
12. OpenBRR. Business Readiness for Open Source. Intel (2005)
13. QSOS. Method for Qualification and Selection of Open Source software. Atos Origin (2006)
14. Samoladas, I., Gousios, G., Spinellis, D., Stamelos, I.: The SQO-OSS quality model: measurement based open source software evaluation. In: IFIP 20th World Computer Congress, Working Group 2.3 on Open Source Software, OSS 2008, Springer, Heidelberg (2008)
15. Spinellis, D., Gousios, G., Karakoidas, V., Louridas, P., Adams, P.J., Samoladas, I., Stamelos, I.: Evaluating the Quality of Open Source Software. Electr. Notes Theor. Comput. Sci., vol. 233, pp. 5–28. Springer, Heidelberg (2009)
16. Sung, W.J., Kim, J.H., Rhew, S.Y.: A quality model for open source selection. In: Sixth International Conference on Advanced Language Processing and Web Information Technology, China, pp. 515–519. IEEE Press, Los Alamitos (2007)