Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 7, July 2015, pg.364 – 374

RESEARCH ARTICLE

Towards a New Evaluation Model to Improve Open Source Software - Application in Moroccan SMEs

Youssef Ait Houaich[1], Mustapha Belaissaoui[2]

[12]SIAD laboratory, University Hassan I, ENCG, Settat, Morocco

[1] yaithouaich@gmail.com, [2] mustapha.belaissaoui@uhp.ac.ma

Abstract— Open Source Software (OSS) is widely used as it offers several advantages such as cost saving, security and ability to modify the source code, which encourages companies to adopt it [1]. Morocco like many countries has been hit by the global crisis that affected its economy; several companies of different sizes have closed and employees were left without a fixed workplace. The objective of this study is to propose a solution to improve the Moroccan economy by the introduction of Open Source Software and to encourage especially small and medium enterprises to adopt it, and also to benefit from the advantages offered by this technology. In this paper, we propose a new model baptized Easiest Open Source Evaluation Model (E-OSSEM) whose goal is to allow decision makers to select the best product that meets the business needs and facilitate its adoption.

Keywords— Decision Making, Evaluation Criteria , Evaluation Process, Free/Libre Open Source Software, Moroccan economy, Open Source Software development.

I. INTRODUCTION

Open Source is a word introduced in 1998 [2]. The Open Source Software (OSS) is any type of software which allows the participants to collaborate without any restrictions and to gain access to its source code with complete license rights as defined by the Open Source Initiative (OSI) [3][4]. The Free Software Foundation (FSF) describes four types of benefits to be respected by an OSS which provide the ability to use the program, the ability to adopt and modify the source code, the possibility of redistributing the modified version of the program and the ability to distribute copies [5]. Today, open source software has become increasingly used by many public and private organizations (e.g. the governments of Brazil, Malaysia, France, and Canada) [6]. The OSS offers many benefits and profitability (e.g. more than $55 billion annual economic is achieved through the adoption of free software, following a report by the Standish Group) [7] [8]. According to bibliographic researches, it has been found that the quality and performance are different in each open source product. In addition, the OSS offers several advantages such as: stability, security and errors can be located and processed very quickly given the large number of developers involved in the project and monitoring the functioning of the application. Therefore, the costs are reduced as most OSS requires less hardware resources.

Morocco's economy consists of 51.4% of services, 32.2% of industry and 16.5% of agriculture. Like many countries, Morocco has been hit by the global crisis that affected its economy. Accordingly, several companies of different sizes have closed and employees were left without fixed work (the unemployment rate reached 9.6% in 2014) [9]. After research visits that we have made on about 200 Moroccan SMEs that operate in different areas, we found that only 34.5% know the definition of OSS, 51% confuse between OSS and free software that they can download for free, and 14.5% have never known this word. Among 34.5% companies which are familiar with OSS, only 76.8% are using this technology in their IT systems of which 79% are using only “OpenOffice" to replace the proprietary products used without a license so as not to invest in an expensive product, which confirms that Moroccan SMEs do not benefit from the advantages offered by the OSS and its exploitation is very limited in the office solution. Moreover, the survey on SMEs showed, on the one hand, that they involve small IT profile to process basic requests such as computers repairing, application installation, network connectivity ... etc., and on the other hand, they use pirated software which currently faces legal problems with the owners, whereas others chose to go through a basic or manual work and avoid using software without a license. In this paper, we propose a new evaluation model baptized E-OSSEM to improve the IT

system of SMEs by using OSS whose objective is to help decision makers to select the most suitable product that meets their needs.

The paper is organized in the following way: Section 2 deals with related work in the evaluation of OSS maturity. In section 3, we present our new evaluation model baptized E-OSSEM and its architecture. Section 4 describes the results of OSS deployment performed on 20 Moroccan SMEs. Finally, we give a conclusion about our study.

II. RELATED WORK

Measuring the quality of a developed software is a key element. To meet this, researchers have proposed evaluation models measuring the quality of proprietary software. Evaluation models define the functional and non-functional requirements and allocate criteria to achieve relevance ranking. With the introduction of the open source technology, traditional evaluation models cannot be applied because of the OSS concept (e.g. Change the source code, software developed by a community ... etc.), and usually the knowledge and recommendation are the main selection criteria of an OSS [10]. Today, there are many open source projects that meet the same features, so it turns out necessary to choose the best mature product for deployment. For these reasons, some quality assessment models of OSS have been proposed to help decision makers choose the best software. Through the study of the needs of change and the evolution environment of the OSS concept, we will overfly six models.

A. Capgemini Open Source Maturity Model

The Capgemini Open Source Maturity Model "C-OSMM" [11] is a model developed by the Capgemini Company in order to help the decision makers determine which open source product is suitable for their needs; the aim is to keep immature products away. The selection is done by evaluating two product levels: product indicator and application indicator. For Application indicator, it is necessary to consider different aspects like environment, current and future users' requirements, cultures...etc. The product indicator defines twelve attributes organized in four groups (Product, Integration, Use, and Acceptance). The results of the two levels of evaluation obtained by assigning a relevance value allow having a final score. This evaluation allows knowing the capacity of each product to meet the company's needs.

B. Open Business Readiness Rating

The Open Business Readiness Rating "OpenBRR" [12] is a model developed by Spike Source, the Center for Open Source Investigation at Carnegie Mellon West in association with Intel Corporation. The OpenBRR proposes four phases of software assessment:

- Quick assessment filter: The objective of this first step is to reduce the free software selection list by removing those which do not meet the basic criteria. Users can use the proposed list of filtering criteria and may add their own for any particular situation.
- Target Usage assessment: Classifies the categories as a first step, according to their interest (1 "the highest, 12" the lower ") and selects the first 7 categories to allocate a percentage to each one, respecting 100%.
- Data Collection & Processing: collects measured data used in the corresponding category and calculates the weighting applied for measurements.
- Calculates the BRR score: By using the category ratings and the functional weighting factors.

C. Open Source Evaluation Model

The Open Source Evaluation Model "OSEM" [13] developed by David A. Wheeler. It proposes four steps of software assessment:

- Identify: The consultation of published lists which provide software valued as mature or experienced as safe tool is recommended.
- Read reviews: Consult various forums to know the advantages and disadvantages of each solution based on user feedback.
- Compare: comparing key attributes of the software with the needs of the user, these attributes are: functions on, cost, market share, support, maintenance, reliability, performance, scalability, usability, security, flexibility / customizability, interoperability and legal / licensing issues.
- Analyze: an analysis of possible software candidates by testing them in a situation that represents the workload.

D. Navica Open Source Maturity Model

The Navica Open Source Maturity Model "N-OSMM" [14] is a model developed by Navica-Golden whose objective is to evaluate the maturity of a product in three steps (Assess vital product elements, Define a weighting factor for each element, and Calculate the product's overall maturity score).

E. Qualification and Selection of Open Source Software

The Qualification and Selection of Open Source Software "QSOSS" [15] is a model developed by Atos-Origin Company which is composed of four stages: (Definition, Evaluation, Qualification, and Selection).

F. Software Quality Open Source Software Evaluation

The Software Quality Open Source Software Evaluation SQO-OSS [16] is a model developed by Samoladas and al which is composed of two steps: (Defining the evaluation model - attributes and their metrics), and defining the aggregation method (the evaluation categories and their profiles).

Although, the models studied in the evaluation of OSS provide a certain number of criteria analysis for a better selection, none of them gives an importance to interoperability which is a relevant criterion. It is necessary to mention that proprietary and OSS should meet the businesses continuity by permitting interoperability between different systems.

III. OUR APPROACH

The objective of our approach, as illustrated in Fig. 1, is to propose a new model baptized Easiest Open Source Evaluation Model (E-OSSEM) which is inspired from the models already presented in the previous section. Our goal is, on the one hand, to use OSS and benefit from its advantages, on the other hand, to allow decision makers select the best product without the intervention of IT experts. In addition to explicit support for conceptual model components, we present the core elements of the E-OSSEM (definition, identification, qualification and selection).

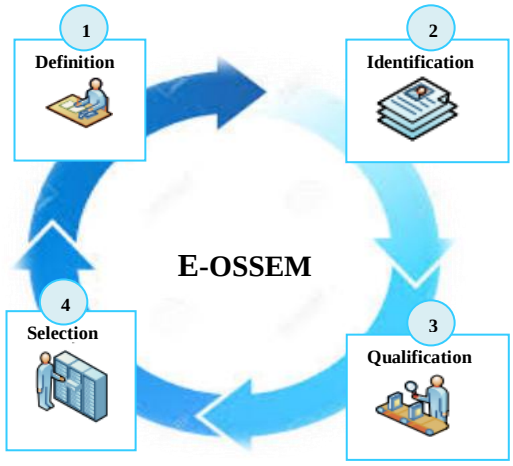

Fig. 1 Our proposed assessment model (E-OSSEM).

A. Definition

The definition phase is of great importance, as it describes the real needs of any company that considers adopting a new OSS. For a better description and information gathering, it is important to check and analyze the following aspects: "functional, technical and strategic". The functional component describes the necessary business functions (e.g. accounting, sales, CRM …etc.) to provide a list of software that may help in meeting the needs. The technical component allows the acquisition of information about secondary elements that can contribute to the success of the system selection (e.g. for people using a current system that will be replaced, we must know their expectations in order to avoid resistance to change). The strategic component should be taken into account since some companies may have contracts with a proprietary software provider that requires the use of certain system only.

B. Identification

The objective of the identification phase is to determine the OSS general characteristics in order to create a data sheet describing all the key elements. To achieve this, we present by mind map the identification groups with indicators to evaluate the product score (see Fig. 2).

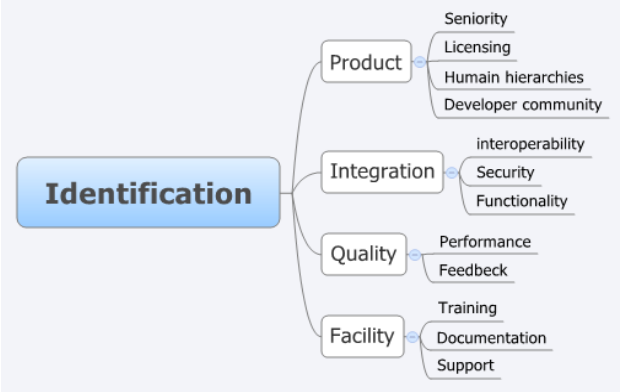

Fig. 2 Groups and indicators of "identification" phase.

1) Product: According to our study in Moroccan SMEs, many decision makers avoid deploying OSS because they fear that the software community drops out its development. The "product" group includes indicators that present information on the development and maintenance of target software which helps to determine whether there is a risk of suspension or not. For this, we present below a group of four criteria:

- Seniority: The age of OSS is very relevant. The company gains more confidence when it knows that the product will continue to exist and also will continue to be supported by its developers. In other words, software that is reviewed on a regular basis has fewer errors and bugs and is, therefore, more stable. So, it is very important to take into account the time that the software spent in the market in comparison with the others since a newer one would always be more utilized.
- Licensing: OSS is presented with a set of rules. It comes with a license that we must understand and accept first. There are several licenses that organize the use of open source (e.g. GPL, BSD and Apache ... etc.). When thinking of adopting open source software, decision maker has to be very careful and check the delivered license before, especially if he/she wants to modify the source code for a specific use (e.g. a government that has decided to migrate to OSS, and wishes to adopt it in the various state services, changes may contain confidential information which should not be disclosed), because some licenses prohibit the reuse of a modified code with a proprietary license and require reproduction with public access. For a better understanding, we start with the "GNU LGPL V3" license that allows the modification of the source code but prohibits the licensing change for the developed program. Another example is the BSD license, which allows the change of the modified source code to make it private.
- Human hierarchies: Human organization within the community is very important. A project has little chance of surviving if it is controlled by one person. In other words, an organization where tasks are distributed among several members whose mission is to develop and improve the OSS project is likely to succeed.
- Developer community: An OSS community has many people who develop software and make it available under a license. They usually use experience to provide continuous monitoring, functionality tests and user support as well as to improve the product and make it competitive. Consequently, we suggest considering four groups. The first group consists of individuals (GI) who develop a project with no guarantee of support or development continuity. The second is the organization (OR) offering effective management of the project life cycle, a vigilant support for various applications and also a website offering structure. The third is the foundation (Fdn) of an open source profit. Commercial organization (Commer) is the last group; its goal is to develop OSS by adding specific features with more support and maintenance options. The latter is a paid service

2) Quality: The "quality" group confirms the ability of OSS to meet the expected needs of users and also to measure its performance. To achieve this, we present below a group of two criteria:

- Performance: The IT system is the company backbone. In fact, every company wants to have an efficient IT system that helps achieving its objectives. It is usually the first concern of the IT department. Adopting an OSS helps not only reducing the costs but also optimizing the hardware resources. For specific business needs (e.g. Web application, ERP, CRM ... etc.), it is necessary to analyze the performance of the potential application before making a decision. So, we suggest inquiring about the application response time (e.g. time spent to return the result of a query) and also about the number of transactions that the system can process within a defined time. We also should consider the need for hardware resources for the proper functioning of the system (e.g. HDD, memory, storage, processor…etc.).
- Feedback: There is a dedicated forum for each OSS. It is used to share the experience regarding the usage of the application between stakeholders (developers and users). Exchanging OSS experience permits learning about its features, communicating new versions and improvements, sharing documentation, which confirms that OSS is transparent. As mentioned before, well-organized communities establish control mechanisms for continuous product improvements among the indicators. From the OSS Website, we quote the result of quality assurance (QA) indicator which reflects the user's satisfaction. This criterion provides a clear idea about the system and its behavior. On the other hand, it is advisable to surf into different discussion forums to gather more information about user's opinions.

3) Integration: The "integration" group represents the technical control of OSS. It helps decision makers to choose the appropriate OSS product which would operate with their companies' software already deployed. It also assists them to know all the served features and the levels of security. So, we suggest three criteria:

- Interoperability: Today, it is becoming more and more necessary to select computer hardware and software that validate their ability to function and communicate with other suppliers products. This is called interoperability. The different OSS maturity assessment models do not treat the interoperability thoroughly.
- Security: The advantage of using OSS is that you can access to its source code, analyze and detect any abnormalities that may present a security risk. As the source code is analyzed by hundreds of people within the community, developers provide a rapid response to any critical demand compared to proprietary software. We

acknowledge that we cannot reach a 100% safety level. So, there should be a continuous follow-up to improve the system with upgrades, patches ... etc.

- Functionality: it is vital to determine the different features that a piece of software has to offer in terms of safety, license, regulatory, support, and documentation. We take as an example a commercial company that has deployed new customer management software CRM with some features. Later on, it turns out that the system cannot be used to configure e-mails and send messages to customers; this requires either special development which would increase the adoption costs or selection of an appropriate solution.

4) Facility: When selecting a new software proprietary or OSS, it is essential to check existing resources that facilitate deployment, usage and product support. For this, we present below three criteria:

- Training: Any change in management requires a good organization to succeed. When adopting a new OSS that will replace another system (e.g. the case of the implementation of a multi-module ERP system), it is strongly recommended to train a staff and improve their skills. For this reason, we insist on "training" as a criterion in the maturity model. So, we should investigate in advance about the training available for the targeted OSS before its selection.
- Documentation: it is a paperwork used to identify a system and linked safely to its destinations. As stated before, it is necessary to be sure of the availability of documentation that would serve in two different parts: the first is the "user's documentation" describing how to use the software and its various features. The second is the "system programmer's documentation" that explains the source code and how to modify it for any need by adding or changing its functioning. In addition, it is important to check the availability of the FAQ tool (Frequently Asked Questions) which provides some advice about the application usage. It goes without saying that the OSS forums remain of great help to users who need assistance.
- Support: it is the key element for the survival of any system. It is a decisive factor that provides solutions especially to the IT teams who want to ensure the smooth running of business systems and continuity of production. According to the survey we have conducted among 200 Moroccan SMEs, 56% of the businesses that have responded do not wish to set up an OSS because for them it is developed by a team of volunteers and not by a professional organization. There is also a fear that it might not be as efficient as possible. To remedy to this situation, the OSS communities have begun to offer detailed descriptions of the developed systems so as to understand its functionality and use it easily. Therefore, some commercial companies offer paid support service for OSS (e.g. OpenERP) with the possibility to have access to 24/7 support. As far as support requests are concerned, we can classify them into three categories: the first is an emergency "High", for example the shutdown of the electronic payment system which may have a negative impact on the turnover of the company. The second is of a "Medium" emergency requiring rapid intervention to prevent the loss of confidential data. Finally, an emergency "Low" when it concerns simple errors of application that require only a fix.

C. Qualification

The principle of "qualification" phase is to assign a score for each criterion to obtain an overall score to facilitate choosing the most appropriate software. For more precision, the evaluation is done by providing a score from 0 to 5 (see Table I).

TABLE I. DEFINITION OF VALUES

0	Unacceptable
1	Weak
2	Acceptable
3	Good
4	Very good
5	Excellent

1) Seniority: As explained before, the age of OSS is very important. Software that has existed for years is more likely to provide a stable version and gives confidence to the continuity of its existence. We present in the table below the scores attributed (see Table II).

TABLE II. MEASURES OF "SENIORITY"

Seniority by year	**Score**
0 < OSS < 3	0
3 < OSS < 5	1
5 < OSS < 7	2
7 < OSS < 9	3
9 < OSS < 12	4
12 < OSS	5

2) Licensing: The type of license distributed with OSS is a relevant element of selection. In this context, we have avoided the software available with a license that does not meet the company's needs. For this criterion, we propose to attribute two scores: score (0) for software available with a license that does not meet the needs and score (5) for software that responds to the request (see Table III).

TABLE III. MEASURES OF "LICENSING"

	Soft 1	Soft 2
The request	BSD	BSD
The software license	BSD	GPL
Score	5	0

3) Human hierarchies: For human hierarchies, here we assign score (0) for a community controlled by a single person and score (5) for an organization that delegates tasks among its members (see Table IV).

TABLE IV. MEASURES OF "HUMAN HIERARCHIES"

Community organizations?	Score
One responsible	0
Shared responsibility	5

4) Developer community: As far as developer community is concerned, we assign score (0) for software developed by an unknown person, score (1) to a single developer, score (2) to a group of individuals, score (3) to a non-profit organization, score (4) to a foundation, and score (5) to a commercial organization (see Table V).

TABLE V. MEASURES OF "DEVELOPER COMMUNITY"

Community types	Score
Unknown	0
Single developer	1
Group of individuals (GI)	2
Non-profit organization (OR)	3
Foundation (Fdn)	4
Commercial organization (Commer)	5

5) Performance: The performance of OSS depends on the service provided. According to Database Management System, one of the important criteria is the response time to a request sent. We therefore assign score (0) for a response more than six seconds, score (1) for an response from four to six seconds, score (2) for a response from two to four seconds, score (3) for response from one to two seconds, score (4) for a response from fifty millisecond to a second, and finally, score (5) for an response less than fifty milliseconds (see Table VI).

TABLE VI. MEASURES OF "PERFORMANCE"

Number of post	Score
RT > 6s	0
4s < RT < 6s	1
2s < RT < 4s	2
1s < RT < 2s	3
50ms < RT < 1s	4
RT < 50ms	5

6) Feedback: We assign a score to this criterion by taking into account the number of topics posted in the OSS forums. For scoring, we assign (0) for a number of posts less than five thousands, (1) for a number of posts less than twenty thousands, (2) to a number of posts less than forty thousands, (3) for a number of posts less than sixty thousands, (4) for a number of posts less than eighty thousands, and (5) for a number of posts more than eighty thousands (see Table VII).

TABLE VII. MEASURES OF "FEEDBACK"

Number of post	Score
Posts < 5000	0
5000 < posts < 20000	1
20000 < posts < 40000	2
40000 < posts < 60000	3
60000 < posts < 80000	4
80000 < posts	5

7) Interoperability: To measure interoperability, it is sufficient to confirm whether OSS is compatible with the existing technology. Moreover, information can be found very easily. We assign score (0) to software that does not meet the

company needs and score (5) for software that is compatible with the technology already deployed at the company level (see Table VIII).

TABLE VIII. MEASURES OF "INTEROPERABILITY"

	Soft 1	Soft 2
Is the product compatible with the technologies deployed?	No	Yes
Scores	0	5

8) Security: We find in the literature different levels of risk security namely high, medium and low. We are only interested in high risk that can cause financial loss to a company as a result of serious errors. For scoring the different cases, we can retrieve relevant information from communities releases or other information sources (e.g. websites security consulting). Moreover, we assign score (0) to a number of risks greater than four in the last twelve months, score (1) to a number of risks equal to four, score (2) to a number of risks equal to three, score (3) to a number of risks equal to two, score (4) to a number of risks equal to one, and score (5) to non-recovered risks (see Table IX).

TABLE IX. MEASURES OF "SECURITY"

Number of major risk (reported during the last twelve months)	**Scores**
Non recovered risk	5
Number of risks = 1	4
Number of risks = 2	3
Number of risks = 3	2
Number of risks = 4	1
Number of risks > 4	0

9) Functionality: As part of the Information Systems governance, we should ensure alignment with the company's strategy. It is necessary to ensure that the OSS that we wish to adopt provides all the functionality required by users. We assign score (0) for software that offers non-requested feature, score (3) for a system that offers some features only, and score (5) for software that perfectly meets the needs (see Table X).

TABLE X. MEASURES OF "FUNCTIONALITY"

Features ensured	**Scores**
No functionality	0
Some features	3
All features	5

10) Training: For some companies, training is needed for users in the case of an implementation of very complicated modular software (e.g. ERP). We assign score (0) to non-existent institutes that can provide training modules on a target product, and score (5) if the training is largely available (see Table XI).

TABLE XI. MEASURES OF "TRAINING"

Training	**Scores**
Not available	0
Available	5

11) Support: The open source software support is a relevant criterion. We identify three types of available support service in the literature: self-support made by the user himself, support provided by the community that has developed the software, and paid support offered by a commercial organization. We assign score (0) for self-support, score (3) for the community support as we can require a service quality, and score (5) for the paid support (see Table XII).

TABLE XII. MEASURES OF "SUPPORT"

Supports	**Scores**
Self-support	0
Community support	3
Paid support	5

12) Documentation: After a bibliographic research, we have found that the documentation is a key element in the OSS selection. We assign score (0) for the lack or absence of documentation, score (3) for documentation that may not have been updated, and score (5) for a recent updated documentation (see Table XIII).

TABLE XIII. MEASURES OF "DOCUMENTATION"

Documentation	Scores
No documentation	0
Document not up to date	3
Document up to date	5

D. Selection

After collecting all the necessary information, the decision maker would have the opportunity to gather the numerical values assigned to each attribute for a final score. For that, we can classify the different OSS evaluated by an order of relevance, which means that the software which meets the needs will get the highest score. For example, we want to introduce new OSS for the management of CRM activities. We have choice between software "X" and "Y" which propose the same features. It is necessary to collect the relevant information that corresponds to our evaluation model (see Table 14). That's why, we'd better choose the software "Y" which has the highest score (See Table XIV).

TABLE XIV. FIRST LEVEL COMPARISON

Evaluation group	Evaluation criteria	Product "X"	Product "Y"
Product	Seniority	2	4
	Licensing	5	5
	Human hierarchies	0	5
	Developer community	3	1
Integration	Interoperability	5	5
	Security	2	4
	Functionality	3	5
Quality	Performance	2	4
	Feedback	1	0
Facility	Training	5	5
	Documentation	5	3
	Support	0	3
	Sum	**33**	**44**

It is possible to have two programs with the same final score, a case never approached by other evaluation models letting the choice to the end user to select only one solution without being necessarily the best. Following our different assessments, we conclude that both programs having the same final score do not necessarily have the same attribute values (see table below). In this case, we propose that decision makers carry out a second evaluation and utilize only the five most relevant criteria for their needs (e.g. security, documentation, support, type of license and interoperability), and hence, choose the most mature software (see Table XV).

TABLE XV. SECOND LEVEL COMPARISON

Evaluation group	Evaluation criteria	Product "X"	Product "Y"
Product	Seniority	2	4
	Licensing	5	5
	Human hierarchies	5	5
	Developer community	3	1
Integration	*Interoperability*	5	5
	Security	2	4
	Functionality	3	5
Quality	Performance	4	4
	Feedback	4	0
Facility	Training	5	5
	Documentation	5	3

	Support	2	4
	Sum	**45**	**45**
	Sum of the second evaluation	**19**	**21**

For special cases, decision makers can add other evaluation criteria to better evaluate the software while maintaining the same notation used for the other criteria.

IV. EXPERIMENTS

We present in this paper a case study carried out on a Moroccan company which wants to deploy a supervision system to check their network devices configured in several subsidiaries in Africa. This project requires the use of OSS and should meet functional requirements. In this context, we have conducted a preliminary study to keep only four OSS solutions (Nagios [16], Zabbix [17], Cacti [18] and Zenoss Core [19]). To better clarify our approach, we will present only the most relevant criteria. We also provide the results of our study made on different Moroccan SMEs.

A. Optorg group

Optorg [20] has long been a recognized name in international trade, with nearly a century of experience in specialized distribution. This skill has paved the way for the success of two of their subsidiaries: Tractafric Equipment Corporation, a Caterpillar dealer in Central Africa and Morocco, and Tractafric Motors Corporation, a major automobile distributor in Africa.

Our objective is to suggest the best OSS using our model (E-OSSEM) for enabling network monitoring of all connected equipment.

1) *Identification:* At this step, we determine the general characteristics of the OSS using only "community, seniority, interoperability and documentation" criteria (see Table XVI, XVII, XVIII and XIX).

TABLE XVI: THE COMMUNITY CRITERIA INFORMATION

OSS	Community type
Zabbix	OR
Nagios	Commer
Cacti	Commer
Zenoss Core	OR

TABLE XVII: THE SENIORITY CRITERIA INFORMATION

OSS	Seniority
Zabbix	10 years
Nagios	15 years
Cacti	12 years
Zenoss Core	13 years

TABLE XVIII: THE INTEROPERABILITY CRITERIA INFORMATION

OSS	Interoperability
Zabbix	Yes
Nagios	
Cacti	
Zenoss Core	

TABLE XIX: THE DOCUMENTATION CRITERIA INFORMATION

OSS	Documentation
Zabbix	Yes
Nagios	
Cacti	
Zenoss Core	

2) *Qualification:* We attribute a score to each criterion following its value (see the table XX).

Table XX: Scoring

OSS	Criteria	Scores	Final score
Zabbix	community	3	17
	seniority	4	
	interoperability	5	
	documentation	5	
Nagios	community	5	20
	seniority	5	
	interoperability	5	
	documentation	5	
Cacti	community	5	19
	seniority	4	
	interoperability	5	
	documentation	5	
Zenoss Core	community	3	18
	seniority	5	
	interoperability	5	
	documentation	5	

3) *Selection:* After assigning a score for each evaluation criterion, it would be easy to calculate the overall sum which classifies the different solutions by order of maturity. In this case, we choose the open source Nagios, it got 20 points.

B. Results of other experiments

We have deployed Nagios for many Moroccan companies that wish supervising their active equipment installed in different premises. Today, more than 1,500 facilities are monitored in real time allowing IT managers to deal with and solve over 4000 technical problems (see Fig. 3).

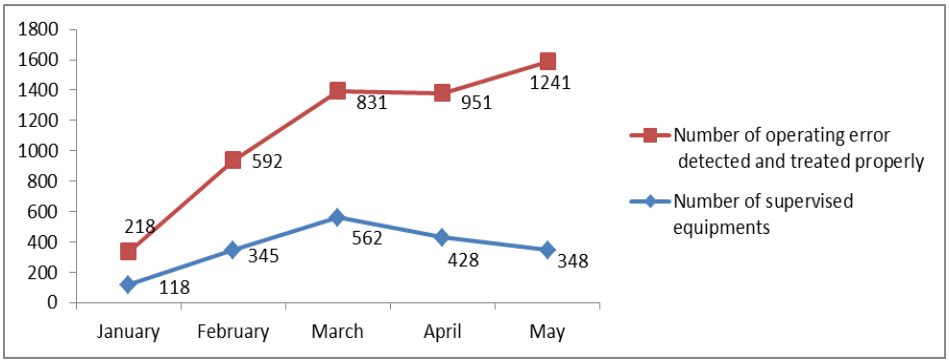

FIG. 3 NUMBER OF SUPERVISED EQUIPMENT AND DETECTED INCIDENTS BY NAGIOS TOOL

After the study carried out on some Moroccan companies, we present bellow a synthesis of the most requested supervision services by them (see Fig.4).

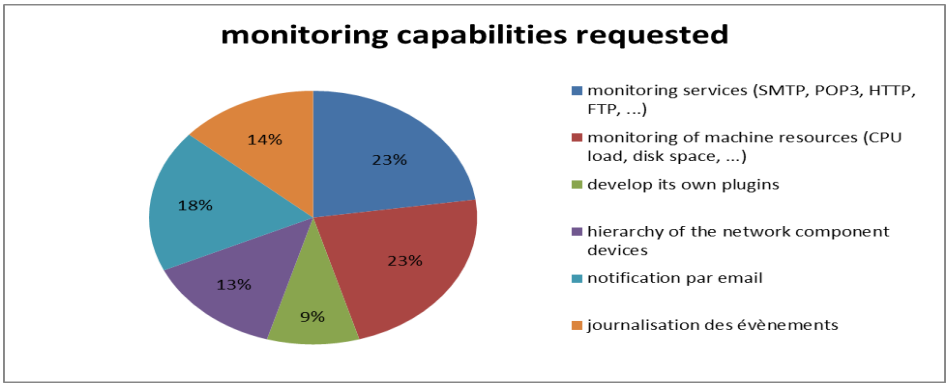

FIG. 4 SUPERVISION SERVICES SOLICITED BY MOROCCAN COMPANIES

V. CONCLUSION

Following our literature review, we conclude that some decision makers still avoid deploying open source software for different reasons, namely the ignorance of its relevance. However, there are some companies which use OSS to meet different business objectives such as security and costs savings. Our goal from this paper is to encourage more Moroccan SMEs to adopt open source software in order to improve their productivity and get benefit from the advantages offered by this technology.

For all these reasons and to help non-technical decision makers in selecting the most suitable software, we propose a new simplified model baptized E-OSSEM.

REFERENCES

[1] Fitzgerald, B.: 'A Critical Look at Open Source', Computer, 2004, 37 (7),pp. 92-94.
[2] Opensource.org (2010), Open Source: http://www.opensource.org. Accessed 8th October 2010.
[3] Finding Open options, An Open Source Software Evaluation Model with a Case Study on Course Management Systems, Berg K., Master thesis, Tilburg University, August 2005.
[4] Eric S (2010), Raymond http://www.catb.org/esr/open-source.html. Goodbye, free software; hello, "open source" accessed 8th October 2010.
[5] Foundations, Principles and Techniques, Pohl K. G. Bockle, and F. J. Vander Linden, Software Product Line Engineeri,ng, Secaucus, NJ, USA: Springer Verlag New York, Inc., 2005.
[6] Feller, J., Fitzgerald, B., Hissam, S. and Lakhani, K. (eds.), (2005). Perspectives on Free and Open Source Software, MIT Press, Cambridge: MA.
[7] St. Laurent, Andrew M. (2008). Understanding Open Source and FreeSoftware Licensing. O'Reilly Media. p. 4. ISBN 9780596553951.
[8] Rothwell, Richard (2008-08-05). "Creating wealth with free software". Free Software Magazine. Retrieved 2008-09-08.
[9] I.S. Hauge, Ø., Osterlie, T., Sorensen, C.-F., and Gerea, M.: 'An Empirical Study on Selection of Open Source Software – Preliminary Results'. Proc. ICSE Workshop on Emerging Trends in FLOSS Research (FLOSS '09), Vancouver, Canada, 2009.
[10] Duijnhouwer, F., and Widdows, C.: 'Open Source Maturity Model', Capgemini Expert Letter, 2003.
[11] Www.openbrr.org: 'Business Readiness Rating for Open Source, RFC 1', 2005.
[12] Wheeler, D.A.: How to Evaluate Open Source Software / Free Software (OSS/FS) Programs, online: http://www.dwheeler.com/oss_fs_eval.html, accessed on: 8 September 2009.
[13] Navica Open Source Maturity Model , http : // web . archive . org / web / 20080507024544 / http : //www.navicasoft.com/pages/osmm.htm.
[14] Atos Origin: Method for Qualification and Selection of Open Source software (QSOS) version 1.6 , Technical Report (2006).
[15] Ioannis Samoladas, Georgios Gousios, Diomidis Spinellis, and Ioannis Stamelos. The SQO-OSS quality model: Measurement based open source software evaluation. In Ernesto Damiani and Giancarlo Succi, editors, Open Source Development, Communities and Quality — OSS 2008: 4th International Conference on Open Source Systems, pages 237–248, Boston, September 2008.
[16] http:\\www.nagios.org/
[17] http:\\www.zabbix.com
[18] http:\\www.cacti.net
[19] http:\\www.zenoss.org
[20] http:\\www.optorg.com

