ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 8, August 2015

Open Source Software Evaluation Model for Smes

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

SIAD Laboratory, ENCG, University Hassan I, Settat, Morocco [1, 2]

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, it is therefore necessary to propose solutions to help our economy to survive and improve. The OSS has been adopted in many developed countries [2], as its introduction and implementation has saved a lot of money and has offered various advantages. After having conducted a survey, we have proposed a new model to decision makers in order to help them choose the best open source software available that exactly meets their needs, which we have nicknamed Easiest Open Source Software Evaluation Model "EOSSEM" [3]. Following new research carried out in collaboration with IT experts, it was necessary to add new relevant features.

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 [4]. 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) [5][6]. 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 [7]. Today, open source software has become increasingly used by many public and private organizations (e.g. the governments of Brazil, Malaysia, France, and Canada) [2]. 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) [8] [9]. 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.

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 present our evaluation model baptized E-OSSEM and the result of the latest research conducted that have helped to introduce two new and relevant selections criteria.

The paper is organized in the following way: Section 2 exposes our evaluation model baptized E-OSSEM. In section 3, we present the results of the latest research and the enhancements made. Section 4 describes the application of certain attributes of our model. Finally, we give a conclusion about our study.

II. Our approach

Our new model baptized Easiest Open Source Evaluation Model E-OSSEM [3] (see Fig. 1)allows 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

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 8, August 2015

intervention of IT experts. We present bellow the core elements of the E-OSSEM.

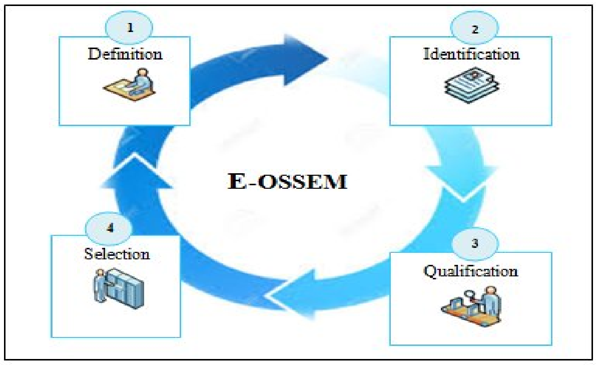

Fig 1. Our proposed assessment model (E-OSSEM).

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).

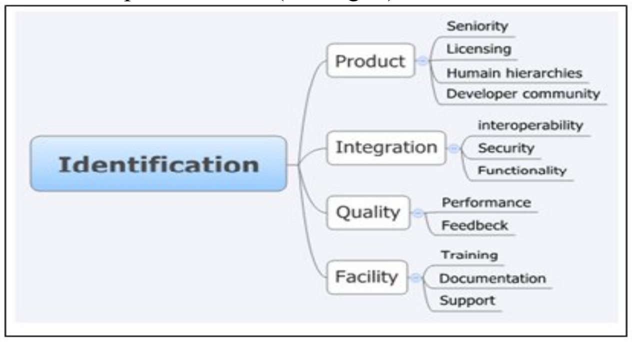

Fig 2. Groups and Indicators of "Identification" Phase.

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

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 8, August 2015

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 functioningof the system (e.g. HDD, memory…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. Asthe 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

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 8, August 2015

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) foran 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"

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 8, August 2015

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. That's why, we'd better choose the software "Y" which has the highest score (See Table XIV).

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. 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).

TABLEXIV.FIRST LEVEL COMPARISON

Evaluation group	Evaluation criteria	Product "X"	Product "Y"

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 8, August 2015

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

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.

III.NEW CONTRIBUTION

We conducted further survey in second quarter 2015 with decision makers and IT experts to test our E-OSSEM model in new environments. It was proved that our evaluation model is very relevant and greatly facilitates the selection of the most suitable open source software.

A. Survey's results

We have presented our evaluation model to the thirty decision makers participating in our study, they have recommended us to add new criteria in order to facilitate the selection of the most fitting open source software (see Fig.3). After having benefited from the guidance and experience of the various participants and in coordination with IT experts, it was decided to enhance our evaluation model so as to present a complete solution that meets the different needs. So we have added the two most requested criteria (evolution and program code).

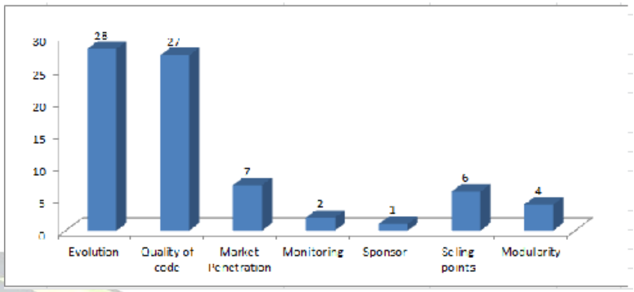

Fig. 3.The different selection criteria recommended by decision makers

B. New criteria

Following different recommendations, it was necessary to update our model by adding two relevant criteria which are:

- The evolution of open source software
- The program code.

As it is noticed bellow, "evolution" is added to the "product" group; while "program code" is added to the "quality" group (see Fig. 4).

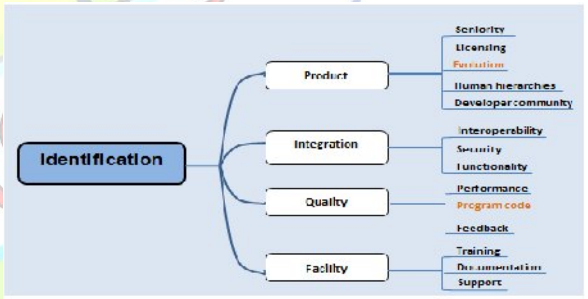

Fig. 4. The new groups and indicators of "identification" phase.

1. Definition of new criteria

- Evolution: Evolution of an open source project indicates whether its leaders have a clear vision and well defined action plan. Each project starts with limited functionality and requires improvement actions, such as error fixing, add or change features ... etc. This criterion is essential in choosing the best open source solution. Open source projects typically offer several versions of the software, with a description of the profits gained from each of them.
- Program code: The program code enables decision makers to have an idea about the quality of open source software, it determines whether the developers applied the general rules of development as the code

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 8, August 2015

structure , line number, coding complexity, interaction with components of the machine (eg processor, memory, input / output interfaces ... etc.)

2. *Qualification of the new criteria*

- Evolution: As it has been explained before, the evolution of OSS is a relevant information. Decision makers are more willing to work with developers who conceive new versions of a software since it brings new functionality and is more likely to exist in time. We present in the table below the scores attributed (see Table XVI).

TABLE XVI.MEASURES OF "EVOLUTION"

Number of versions developed (since the launch of the project)	Scores
Number of versions > 40	5
20 < Number of versions ≤ 40	4
10 < Number of versions ≤ 20	3
5 < Number of versions ≤ 10	2
1 < Number of versions ≤ 5	1
Number of versions = 1	0

- Program Code: After a bibliographic research, we have found that the program code is a key element in the OSS selection. We assign score (0) for the absence of method application, score (2) for a method used but not fully respected, and score (5) for a method used and respected scrupulously (see Table XVII).

TABLE XVII.MEASURES OF "PROGRAM CODE"

Program code	Scores
No method applied	0
A method used but not fully respected	2
A method used and respected scrupulously	5

IV. EXPERIMENTS

We present in this paper a case study carried out on a Moroccan company which wants to deploy a backup system to save its relevant data (user or server). In this context, we have conducted a preliminary study to keep only three OSS solutions (Bacula [10], Amanda [11] and Backuppc [12]). To better clarify our approach, we will present only the most relevant criteria.

A. *Identification*

At this step, we determine the general characteristics of the OSS using only "community, seniority, evolution and program code" criteria (see Table XVIII, XIX, XX and XXI).

TABLE XVIII: THE "COMMUNITY" INFORMATION

OSS	Community type
Bacula	GI
Amanda	OR
Backuppc	SD

TABLE XIX: THE "SENIORITY" INFORMATION

OSS	Seniority
Bacula	15 years
Amanda	14 years
Backuppc	13 years

TABLE XX: THE "EVOLUTION" INFORMATION

OSS	Number of versions
Bacula	13
Amanda	4
Backuppc	2

TABLE XXI: THE "PROGRAM CODE" INFORMATION

OSS	Program code structure
Bacula	A method used and respected scrupulously
Amanda	A method used and respected scrupulously
Backuppc	A method used but not fully respected

B. *Qualification*

We attribute a score to each criterion following its value (see the table XXII).

Table XXII: Scoring

ISSN2394-3777 (Print)
ISSN2394-3785 (Online)
Available online atwww.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 8, August 2015

OSS	Criteria	Scores	Final score
Bacula	community	2	15
	seniority	5	
	evolution	3	
	program code	5	
Amanda	community	3	14
	seniority	5	
	evolution	1	
	program code	5	
Backuppc	community	1	9
	seniority	5	
	evolution	1	
	program code	2	

C. 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 Bacula [10], it got 15 points.

V. Conclusion

After analyzing the data collected during our study launched among thirty small and medium enterprises, we draw the following two conclusions:

- Our proposed model showed its efficiency and relevance, which allowed the various decision makers select the best open source software that perfectly meets their expectations.
- New selection criteria will be integrated for better effectiveness of our model.

References

[1]. Fitzgerald, B.: 'A Critical Look at Open Source', Computer, 2004, 37 (7),pp. 92-94.

[2]. Feller, J., Fitzgerald, B., Hissam, S. and Lakhani, K. (eds.), (2005). Perspectives on Free and Open Source Software, MIT Press, Cambridge: MA.

[3]. Youssef Ait Houaich, Mustapha Belaissaoui.Towards a New Evaluation Model to Improve Open Source Software - Application in Moroccan SMEs. International Journal of Computer Science and Mobile Computing (IJCSMC). Vol. 4, Issue. 7, July 2015, pg.364 – 374. ISSN: 2320–088X

[4]. Opensource.org (2010), Open Source: http://www.opensource.org. Accessed 8th October 2010.

[5]. 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.

[6]. Eric S (2010), Raymond http://www.catb.org/esr/open-source.html. Goodbye, free software; hello, "open source" accessed 8th October 2010.

[7]. 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.

[8]. St. Laurent, Andrew M. (2008). Understanding Open Source and FreeSoftware Licensing. O'Reilly Media. p. 4. ISBN 9780596553951.

[9]. Rothwell, Richard (2008-08-05). "Creating wealth with free software". Free Software Magazine. Retrieved 2008-09-08.

[10]. http://www.bacula.org/

[11]. http://www.amanda.org

[12]. http://backuppc.sourceforge.net/