USQO-FOSS quality model: Utilization based Software Quality Observatory for evaluation of Free and Open Source Software

P. Vijaya[1], Satish Chander[2] and G. Raju[3]

Abstract— **In today's fast-moving and competitive industrial environment, software requirement is a pivotal issue. Free Open Source Software (FOSS) is freely distributable software which provides greater values to the users/companies with great zeal and high hostility. Even though many advantages are provided by FOSS, it has not been effectively used in real applications. In this paper, an attempt is made to find the factors affecting the effective utilization of FOSS. We proposed Utilization based Software Quality Observatory for evaluation of Free and Open Source Software (USQO-FOSS) quality model to analyze the relationship between the software quality factors. Our quality model evaluates all aspects of the FOSS development, both the product, and community. Moreover, we investigate the effectiveness of the USQO-FOSS quality model over some working groups using two metrics UGAM and OSSUI to measure the quality of user experience. Research findings show that FOSS is a viable software solution for commercial software dependent companies.**

Index Terms— **Free Open Source Software, Open Source Software Utilization Index, Usability goal achievement matric, Open Source utilization Index**

I. Introduction

FROM the time of conception [7], 'Open Source' has been considered as an active area of research. Progressively more commercial software dependent companies have taken a keen interest in the open source software. According to Open Source Initiative (OSI), Open Source Software is defined as freely distributable software which is freely available to the users [6] and so it is called Free and Open Source Software (FOSS).Some of the open source software's are Apache Web server, Linux, Web language PHP, database server MySQL, office suite [8] etc. In the FOSS, the source code which is the recipe of the intended software applications is freely available to its users [3]. But in Software's other than the Open source software which is referred as Proprietary software is closed source software, wherein the source code of the software strictly belongs to the vendor. The FOSS is considered free because the source code is publically available which are share via the internet [3]. The question ascends how the OSS company detain the profit if it is provided free access to the users available, the solution is the FOSS community [9-10] conferring to which the users (resource) of software inadvertently debugs the errors on whereas in proprietary software's financial resources are needed for the testing and quality assurance of software's. The software is peddled with the support and maintenance cost is provided by the user resources which increases the profit of FOSS.

In recent years, the adaptation of the FOSS among organizations and users are increasing because of the advantages such as cost reduction, security, reliability, etc. when compared to counterpart software's [11]. By providing software assistance as per the need, the revenue and the profit of the companies are also increased by the FOSS [4]. However, sufficient utilization of the FOSS in Information technology and Research and Development sectors are low [3]. The problem is concerned with the nature of FOSS which includes open access to the source code, shared artifact repositories, peer review of committed code, asynchronous global development and lack of formal support, etc., traditional software quality models may not be sufficient. The source code is the important problem in the FOSS. Companies prefer the source code with fewer bugs and errors which reduce the development cost of the software. Another factor for the rejection of the FOSS is that it does not provide software packages in related to the intended operation, and also they are alleged with the lack of reliable on-going technical support.

To make the practical application of FOSS more efficient in different software dependent sectors, it is significant to understand the needs of the OSS users. Users in OSS are programmers. The utilization improves only when the FOSS provides necessary satisfaction. The main factor responsible for effective utilization of OSS is the OSS quality. OSS quality evaluation validates the whether FOSS is suitable for the appropriate needs and also confirms whether it has been matured enough for the deployed software drive. Evaluations of FOSS are performed based on software models in which the software metrics are used along with expert's opinions. The high-quality OSS automatically leads the effective utilization of the OSS [12]. The advent of FOSS has rendered the traditional quality evaluation models non-applicable to some extent since they cannot be used to evaluate both the software and the community as a whole.Many of the studies were conducted for the effective utilization of FOSS [13-14]. Most of the existing studies fails to address the structural relationship between the factors related to the quality utilization. Existing models have the rather simple structure of only a few factors.

In this paper, we present a new novel software quality evaluation model called USFO-FOSS quality model. The utmost intention is to evaluate the Free and Open Source

[1,2]Department of Computer Science & Engineering, Waljat College of Applied Sciences, P.O. 197, Rusayl, PC-124, Sultanate of Oman. (phone: 968-24446660; e-mail: vijaya@waljat.net;satish@waljat.net)

[3]Department of Information Technology, Kannur University, Kannur, Kerala, India. (e-mail: kurupgraju@gmail.com)

Software and to find what affects the FOSS utilization. The USQO-FOSS quality evaluation model validates the direct and indirect effects of the quality factors over the utilization of FOSS. Furthermore, we investigate the quality model over some working groups including students to quantify the effectiveness of the proposed model approach using the open source utilization index OSSUI and UGAM.

The organization of this paper is as follows. In section II, literature of the existing research works related to Open Source Software Evaluation model is presented. The proposed USQO-FOSS quality model is deliberated in section III with the quality factors considered for the quality evaluation. The quality evaluation metrics considered for the utilization evaluation are discussed in section IV. Evaluation of proposed quality model is provided in section V. Section VI concludes this paper.

II. Literature review

In this section, the literature of the research works related to Software evaluation model is deliberated. Accordingly, eight research papers [1-5, 18-20] regarding the quality model for Open source software evaluation is reviewed and analyzed. The quality investigation of Free and open source software's (FOSS) in the organizations expresses the certain aspects affecting the effective utilization.

M. Dollores Gallego et al. [1] have presented a user acceptance model for evaluation of Opens source software. In their study, they identified the variables and factors that have a direct effect on the individual attitude towards OSS adaptation. In line with their study, user demands OSS with high quality, wide capabilities, and flexibilities, because users remark that the OSS are easier to use and more useful. But the model of [1], doesn't answer the question whether the OSS satisfy the need of the user requirements. Bernard Golden et al. [2] have presented an Open Source Maturity Model for evaluation of the open source products. They assumed the fact that the quality of the OSS is proportional to the maturity. They evaluated the OSS using six factors concerned more to the maturity such as product software, support, Documentation, Training, Product integrations and Professional services. Each of the quality factors has some weights and are evaluated based on some score. Despite the fact, the OISMM model is simple; it is criticized for the left out of the important software artifacts such as source code itself.

SoYoung Sohn et al. [3] have demonstrated an OSS utilization model based on structural equation model. They analyzed the relationship between the quality factors based on the ISO9126 and OSS utilization. The Open Source utilization Index was also used for validation of the user experience. In their method, the community factors of the OSS are left out. On the contrary, Georgios et al. [4] have presented an SQOSS quality model for OSS utilization evaluation with community factors. It was found to be an automated software evaluation system. In addition to community factors, the product factors of OSS responsible for OSS development were also evaluated in their approach. Variable with minimal human intervention and metric oriented are considered for quality evaluation which seems to be a shortcoming. Anirudha Joshiu et al. [5] have presented an evaluation model for OSS utilization measurement. In their quality model, two metrics UGAM and LOL which measures the extent of the software product design based on the user experience goals were used for assuring the quality of the user experience. The two metric inter-relatedly expresses the effectiveness of the factors responsible for the effective OSS utilization.

Haim Bar, *et al.* [18] developed and, open-source concept mapping software called R-CMap, which is implemented in R. This software provided a graphical user interface to guide users through the analytical process of concept mapping. The R-CMap software allowed users to generate a variety of plots, including cluster maps, point rating and cluster rating maps, as well as pattern matching and go-zone plots. Benjamin De Leener *et al.* [19] have introduced the Spinal Cord Toolbox (SCT), a comprehensive OSS dedicated to the processing of spinal cord MRI data. SCT built on previously validated methods and included state-of-the-art MRI templates and atlases of the spinal cord, algorithms to segment and register new data to the templates, and motion correction methods for diffusion and functional time series. Ankita Bansal [20] has analyzed the effectiveness of hybridized search based algorithms for change prediction using OSS. The models were constructed models using machine learning techniques and compared the performance of these models with the models constructed using Search Based Algorithms. The validation was carried out on two open source Apache projects, Rave and Commons Math.

A. Challenges

Based on the literature, some of the challenges are found in the existing FOSS utilization evaluation model. They are listed below;

- Factors considered for the quality evaluation is well thought-out to be the crucial challenge in most of existing quality evaluation model of OSS [1-5].
- Factors comprising both the product and community aspects of the OSS are necessary for quality evaluation.
- Utilization index is another challenge in the OSS quality model [1&4]. The utilization metrics should measure the quality of the user experience without any negligence of software artifacts.

Although there were many studies done, as per the literature is found that what most of these studies failed to address is a structural relationship among the factors related to OSS utilization. The prerequisite is to construct an evaluation model with a simple structure having different quality factors comprising the product (code) and Community aspects of FOSS.

III. USQO-FOSS quality model for Free and Open Source Software (FOSS)

In this section, the detailed description about the proposed USQO-FOSS quality model is presented. Primarily, the quality factors considered for the FOSS utilization evaluation is deliberated. The quality factors considered for the evaluation of FOSS quality in the proposed quality model is taken from the factors of SQO-OSS [4] and SEM-OSS [1] model.

A. Quality factors based on SQO-OSS (Software Quality Observatory for Open Source Software) model

The quality factors based on the SQO-OSS model is deliberated in this section. In SQO-OSS [4], quality factors related to the source code base of the OSS and community of the OSS are taken into account for the construction of the quality model. The factors interrelated to the code and community are constructed with the help of the Goal Question metric [15]. In perspective of source code base of OSS, the quality factors are preferred related to the characteristics which quantify the OSS evaluation. The quality factors related to the source code in SQO-OSS are maintainability, reliability, and security. In similar fashion, the factors regarding the community of OSS are also chosen. The factors for measuring the community quality are chosen as the community factors. The community quality factors considered in SQO-OSS model are mailing list, Developer base, and Documentation. In SQO-OSS model, the factors for evaluation are preferred based on the quality assurance questionnaire. Each of the characteristic factors has some sub-characteristics of OSS product quality. The sole disadvantage of the SQO-OSS model is a limitation in factor consideration. The factor regarding the portability, utilization, etc. is not considered for evaluations which are responsible for effective utilization of OSS.

B. Quality factors based on SEM-OSS (structural equation model for Open Source Software)

In this section, the quality factors based on the SEM-OSS evaluation model is deliberated. In SEM-OSS model [3], the factors are selected for quality evaluation by taking some entities into account such as users view, quality assurance, and sharing factor. In SEM-OSS evaluation model, characteristics of the ISO/IEC 9126 quality model are considered as quality factors for OSS utilization evaluation. ISO/IEC 9126 quality model consist of seven factors related to the source code base. In addition to the factors of ISO/IEC 9126, some latent and measurable variables are also used as the quality factors in SEM-OSS model. The added variables are sharing, and utilization. The additional latent variable is added along with the efficiency factor of the ISO 9126 model to determine the cause and effect relationship to the OSS utilization. Likewise, each of the focused quality factors has some sub-attributes with interrelationship. The quality factors in the SEM-OSS model are maintainability, usability, sharing, reliability, portability, functionality, efficiency, and utilization. The solitary disadvantage of the SEM-OSS model is negligence of the quality factors relied on the community aspects of the OSS.

C. USQO-FOSS quality model

The proposed USQO-FOSS quality model is discussed in this section. The factors preferred for the quality evaluation of OSS utilization is also highlighted in this section.

The tree view of the proposed USQO-FOSS quality model with the quality factors selected for the utilization evaluation is given in figure 1. The proposed USQO-FOSS model is inspired by the SEM and SQO-OSS model. In the proposed model, an attempt is made to integrate multiple quality factors for the quality evaluation of OSS with a simple structure. Moreover, the OSS evaluation model comprising both the community factors and code factors by OSS utilization is the main intent of the proposed model. In USQO-FOSS model, nine factors related to product code of OSS and one factor related to community concept of OSS is selected as the quality factors. The source code base quality factors are sharing, reliability, maintainability, usability, portability, functionality, utilization, efficiency, and security. It may seem that community factors considered for the proposed model are less, but the sub-metrics of the community factor covers all factor related to the community base of FOSS. The advantages of the proposed USQO-FOSS model are i) It doesn’t require heavy user interference for the evaluation of FOSS utilization ii) Automatic evaluation because of the two metric collections iii) Focuses on quality factors related to the source code as well as the community iv) Simple since only the factors which can be measured automatically are chosen for the evaluation.

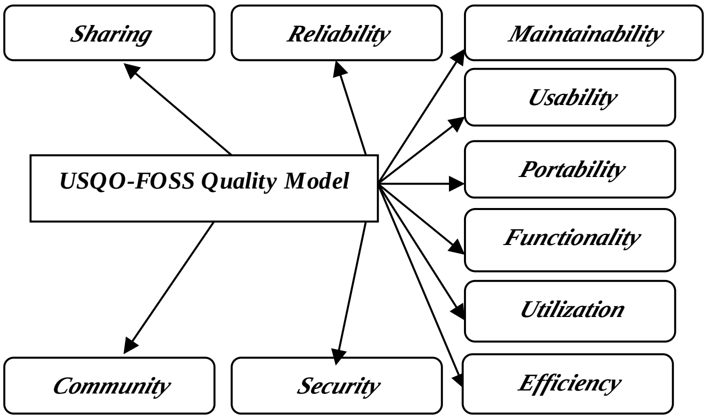

Figure.1 Proposed USQO-FOSS Quality Model

The quality factors and the sub-attributes of the respective quality factors which are engaged in the proposed quality model USQO-FOSS model are listed in table 1. Each of the quality factors has some of the sub-attributes which are thought ought to be the measurement variables for the quality evaluation.

TABLE.I QUALITY FACTORS

Quality factor	Sub-Attributes
Maintainability	Analyzability Changeability Stability Testability
Portability	Adaptability Installability Replaceability Reusability
Usability	Understandability Learnability Operability
Reliability	Maturity Fault tolerance Recoverability
Functionality	Suitability Accuracy Interoperability License
	Developer network

Sharing	Flexibility
Utilization	Development cost Openness Willingness to use
Community	Documentation Mailing List Developer base
Security	------------

Definition:

The definition of the quality factors focused in the proposed USQO-FOSS quality model is listed below;

Maintainability:

The competence of the FOSS against the modification is called the maintainability. The modifications include single corrections or adaptation to changing environment.

Portability:

The competence of the FOSS to be transferred from one environment to another environment is called the portability.

Usability:

The competence of the FOSS to easily learn, understand and operate is called the usability.

Reliability:

The competence of the FOSS to maintain its level of performance in changing conditions is called reliability.

Functionality:

The capability of FOSS to provide functions under specific needs is called functionality.

Sharing:

The characteristic of FOSS to communicate between knowledge seekers and knowledge providers is called sharing.

Utilization:

The competence of the FOSS to satisfy the software developers their intended projects is called utilization.

Security:

The competence of the FOSS against null differences is called security,

Community:

The competence of FOSS susceptible for the users to debug the errors in the software is called community.

The quality factors picked in the proposed model are interrelated to each other. For example, sharing factor has more impact on the portability, functionality and reliability factors. The FOSS effective utilization is possible only if certain conditions of all the handy factors are attained. Each of the quality factors has some of the sub-attributes specifying the quality characteristics responsible for effective utilization. The quality factors are measured by measuring the sub-attributes of the factors. In the proposed USQO-FOSS model, factors resemble the sub-attributes of the quality factors. Each of the sub-attributes is considered as the measured variables. The measurement value for evaluation from the concerned factor is chosen between the scale interval 0 and 100. ‘0’ represents the worst possible measurement for the evaluation on account of destined parameter, ‘25’ represents the quite bad measurement of intended parameter, ‘50’ represents the undecided state of measurement, ‘75’ represents the measurement has good enough impact on FOSS utilization, and ‘100’ represents the best possible measurement for evaluation.

IV. Quality evaluation metrics based on UGAM and OSSUI

The description of the quality evaluation metrics reflected in the proposed USQO-FOSS quality model is presented in this section. The quality evaluation metrics thought out in the proposed quality model is Usability Goal Achievement Metric (UGAM) [4] and Open Source Software Utilization Index [1].

A. UGAM

UGAM is the usability goal achievement metric which is used to measure the quality of the user experience based on the quality factors. In the proposed model, UGAM metric measures the user experience in the effective utilization of FOSS. The sole advantage of the UGAM metric is that it can be used in different ventures irrespective of the domain or application. The UGAM is the weighted sum of all the quality factors of FOSS. The formula for the UGAM calculation is given by;

$$UGAM = \sum_F (W_F . \times S_F) \Big/ \sum_F W_F \qquad (1)$$

Where, W_F is weight of the quality factors and S_F is the score of the concerned quality factors F. The weight value of the quality factor is of the range [0, 5]. The weight values are assigned to the sub-attributes based on their effect on the FOSS utilization (typical importance). For example, utilization quality factor is significant for FOSS evaluation so the weight of 5 is assigned to it, whereas replaceability is provided with weight 0 which is the sub attribute of portability, since it is inferior for FOSS utilization evaluation.

B. OSSUI

Open Source Software Utilization index (OSSUI) is other evaluation metric used for the quality evaluation of FOSS utilization in the proposed USQO-FOSS model. OSSUI is based on the ACSI concept [16]. In our proposed model, OSSUI is related to all the 27 measurement variables. The formula for the OSSUI calculation is given by;

$$OSSUI = \frac{\sum_F S_F Y_F - \sum_F S_F}{(r-1) \times \sum_F S_F} \qquad (2)$$

where, S_F is the measured variable, Y_F is mean of the measured variable, r is Likert scale range, usually within (0, 5). Both the OSSUI and UGAM are calculated for the measured variables varying from 0 to 100 in a group. In the proposed USQO-FOSS quality evaluation model, the value of F is 27, since 27 sub-attributes of the factors (Table 1) is taken as the factors for evaluating the FOSS utilization.

V. Evaluation

The evaluation of the proposed USQO-FOSS evaluation model is presented in this section. To evaluate the performance of the proposed USQO-FOSS quality model, a survey is conducted on five different strategies of working groups. The validation of the proposed model is performed over the Weka 3 OSS using the metrics USSI and UGAM.

The analysis results in the factors which have direct and indirect effects on effective utilization of the Weka 3 OSS as well as the quality of the user experience.

The data collected from different strategies of working groups are measured variables which are analyzed with utilization index OSSUI and UGAM.

The detailed description of the Weka 3 software and strategies of working groups considered for evaluation are discussed below.

A. Weka 3: open source machine learning software

In this section, a brief description about the Weka 3 software taken for evaluation of proposed USQO-FOSS quality model is presented.

The Waikato Environment for Knowledge Analysis (WEKA) OSS [17] is software which is the comprehensive collection of machine learning algorithms and data pre-processing tools. It provides the users the quick try out of different machine learning methods for the new data set. The workbench of Weka 3 software includes the algorithms for regression, classification, clustering, association rule mining, and attribute selection which makes Weka 3 a better option for data mining applications. Weka 3 has a simple tool kit with easy plugin mechanism which facilitates the programmers to integrate new learning algorithm with the help of the Weka graphical user interfaces. This is the reason, Weka 3 software is chosen as the FOSS for the utilization evaluation of proposed USQO-FOSS model.

B. Strategy

The strategy considered for the evaluation of proposed USQO-FOSS evaluation model over the Weka 3 software evaluation is discussed in this section.

The proposed USQO-FOSS quality model for FOSS evaluation is validated using the Weka 3 software. This study analyses the Weka 3 Utilization with data collected from 5 different strategies of working groups. From each of strategy, the quality factors from the programmers are measured and evaluated with the utilization metrics to find the effect of the FOSS utilization. The strategies of working groups considered for the survey are listed as;

- Strategy 1: Group of 10 people who are professional programmers with 10 to 15 years programming experience.
- Strategy 2: Group of 10 people who are professional programmers with 5 years programming experience.
- Strategy 3: Group of 10 people with programming skill but without experience.
- Strategy 4: Group of 10 people who are studying programming skills.
- Strategy 5: 10 Students of computer science engineering

The data concerned to the considered quality factors (Source code as well as community) are collected from the above mentioned strategy for the analysis of effective utilization of Weka 3 software. Before validating the proposed USQO-FOSS model, confirmatory analysis is conducted for validating the relationships among the measurement variables.

C. Sample UGAM and OSSUI score

The sample UGAM score and OSSUI score for the data collected from one of the five different strategies of working groups is presented in this section.

Table 2 lists the sample UGAM score and OSSUI score for the measured variables collected from the strategy two working group of people who are professional programmers with five years' experience. In table 2, P represents the people in the group. The UGAM and OSSUI score is calculated using the equation (1) and (2).

TABLE.II SAMPLE UGAM

Strategy 1	UGAM Score	OSSUI Score
P1	58.4986	48.8462
P2	53.2824	40
P3	53.2824	38.0769
P4	43.8302	40.9615
P5	55.586	47.1154
P6	49.641	41.1538
P7	52.7879	51.1538
P8	54.2083	45.1923
P9	57.9799	44.8077
P10	57.8308	51.3462

D. Analysis

In this section, analysis of proposed USQO-FOSS quality model over the Weka 3 FOSS using the UGMA and USSI metric are presented.

i) Analysis based on UGMA:

The analysis based on the UGMA score of the proposed UGSQO-FOSS evaluation model is presented in this section. The analysis is deliberated regarding mean and variance value of UGMA score.

Mean:

Table 3 depicts the Mean UGAM score of various strategies. From the table 3, it is clear that working group of people with 10 to 15 years of experience attained the maximal UGAM mean score of 47.269. The minimal UGAM mean score of 37.788 is attained by strategy five working group of CS students. From the table 3, it is clear that better UGAM score relies on the functionality, community of the OSS users.

TABLE.III MEAN UGAM SCORE

Strategy	Mean UGAM Score
1	47.269
2	40.404
3	46.308
4	48.231
5	37.788

Variance:

The Variance UGAM score of different strategies of the working group considered for the evaluation is given in table 4. The minimal variance UGAM score corresponds to better utilization of Weka 3 software. Survey results that strategy one working group of people with 10 to 15 years user experience attained the minimal variance UGAM score of 26.305. The worst case UGAM variance score of 438 is attained in strategy 5.

TABLE.IV. VARIANCE UGAM SCORE

Strategy	Variance UGAM Score
1	26.305
2	36.051
3	124.93

4	204.5
5	438.02

ii) Analysis based on OSSUI:

In this section, the analysis of the proposed USQO-FOSS quality model over Weka 3 software using OSSUI score is deliberated. The analysis of OSSUI is presented in terms of Mean and variance OSSUI score.

Mean:

Table 5 shows the mean OSSUI score of different strategies of the working group considered for the evaluation of proposed USQO-FOSS utilization model over Weka 3 software. The best case OSSUI mean score value of 53.232 is attained by the strategy 1. The worst case OSSUI mean score value of 42.318 is attained by the strategy 5. The mean OSSUI score resembles the fact that better utilization depends on the programming skills of the OS users.

TABLE.V. MEAN OSSUI SCORE

Strategy	Mean OSSUIScore
1	53.232
2	44.909
3	49.728
4	50.933
5	42.318

Variance:

The variance OSSUI score value of different strategies of the working group considered for the evaluation of proposed USQO-FOSS utilization model over Weka 3 software is given in table 6. Table 6 shows that programmers with 10 to 15 years of experience have increased Weka 3 software utilization compared to another group of working people. The best case variance OSSUI score value of 22.106 is attained by strategy 1 and worst case variance OSSUI score value of 179.12 is attained by strategy 4.

TABLE.VI.VARIANCE OSSUI SCORE

Strategy	Variance OSSUIScore
1	22.106
2	27.622
3	84.154
4	179.12
5	37.788

From the analysis, it is clear that strategy with better reliability, sharing, efficiency, community and security factors attained maximal OSSUI and UGAM score. Based on the survey, it is found that the working groups of programmers with the better community as well as the source code based quality factors attained higher UGAM score and OSSUI score.

VI. CONCLUSION

In this paper, we have presented a new Free and Open Source Software quality model called USQO-FOSS quality model. The model was constructed centered on Software Quality Observatory and Open Source Software model. The evaluation of OSS quality was performed over ten different quality factors comprising both communities as well as product aspects of the FOSS. The performance of proposed evaluation quality model is analyzed over Weka 3 FOSS. The evaluation of the quality model is facilitated with two utilization metrics; Open Source Software Utilization Index (OSSI) and Usability Goal Achievement Metrics (UGAM). The study reveals the direct and indirect effect of the quality factors in FOSS utilization. The result reveals that quality factors such as functionality, sharing, efficiency, and community have a significant effect on Weka 3 FOSS utilization. This implies that if functionality, sharing, efficiency, and community factors are improved, the FOSS utilization also increases.

REFERENCES

[1] M. Dolores Gallego, Paula Luna, Salvador Bueno, "User acceptance model of open source software", Computers in Human Behavior, vol. 24, pp. 2199–2216, 2008.

[2] Golden.B., "Making Open Source Ready for the Enterprise: The Open Source Maturity Model", Extracted From Succeeding with Open Source, Addison-Wesley Publishing Company, 2005.

[3] So Young Sohn, Min Seok Mok, "A strategic analysis for successful open source software utilization based on a structural equation model", The Journal of Systems and Software, vol. 81, pp. 1014–1024, 2008.

[4] Ioannis Samoladas, Georgios Gousios, Diomidis Spinellis, Ioannis Stamelos, "The SQO-OSS Quality Model: Measurement Based Open Source Software Evaluation", Open Source Development, Communities and Quality, vol. 275, pp. 237-248, 2012.

[5] Anirudha Joshi, N.L. Sardaa, Sanjay Tripathi, "Measuring effectiveness of HCI integration in software development processes", The Journal of Systems and Software, vpl. 83, pp. 2045–2058, 2010.

[6] Dinesh Kumar, Dr. C.S. Lamba, "Open Source Software Evaluation Model for Learning Management System ", International Journal of Emerging Trends & Technology in Computer Science, vol. 2, no. 4, pp. 62-72, 2013.

[7] opensource.org, Open Source: http://www.opensource.org. Accessed 8th October 2010.

[8] Hissam S, C.B.Weinstock, Daniel Plakosh and Jayatirtha Asundi, "Perspectives on Open Source Software", Software Engineering Institute, technical report, CMU /SEI-2001-TR-019, Carnegie Mellon University, Nov 2001.

[9] Wheeler D. A., "How to evaluate open source software / free software (oss/fs) programs", http://www.dwheeler.com/oss fs eval.html, 2010.

[10] Hertel, G., Niedner, S., HerrmannS., "Motivation of software developers in open source projects: an internet-based survey of contributors to the Linux kernel", Research Policy, vol. 32, no.7, pp. 1159– 1177, 2003.

[11] Ruffin, C., & Ebert C., "Using open source software in product development A primer", IEEE Software, vol. 21, no. 1, pp. 82–86, 2004.

[12] Carbone, G., & Stoddard.D, "Open source enterprise solutions", New York: Wiley, 2001.

[13] Hars, A., & Ou, S., "Working for free? Motivations for participating in open-source projects", International Journal of Electronic Commerce, vol. 6, no. 3, pp.25–40, 2002.

[14] Johnson. J. P., "Open source software: Private provision of a public good", Journal of Economics and Management Strategy, vol. 11, no. 4, pp. 637–662, 2002.

[15]Solingen.R.V, "The goal/question/metric approach", Encyclopedia of Software Engineering, vol. 2, pp. 578-583, 2002.

[16] Fornell, C., Johnson, M.D., Anderson, E.W., "The American customer satisfaction index: nature, purpose, and findings", Journal of Marking, vol. 60, no.4, pp. 7–18, 1996.

[17] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H., "The WEKA data mining software: an update", ACM SIGKDD explorations newsletter, vol. 11, no. 1, pp. 10-18, 2009.

[18] Haim Bar, Lucas Mentch, "R-CMap—An open-source software for concept mapping", Evaluation and Program Planning, Vol. 60, pp. 284–292, February 2017.

[19] Benjamin De Leener, Simon Lévya, Sara M. Dupont, Vladimir S. Fonov, Nikola Stikova, D. Louis Collins, Virginie Callot, Julien Cohen-Adad, "SCT: Spinal Cord Toolbox, an open-source software for processing spinal cord MRI data", NeuroImage, October 2016.

[20] Ankita Bansal, "Empirical analysis of search based algorithms to identify change prone classes of open source software", Computer Languages, Systems & Structures, Vol. 47, Part 2, pp. 211–231, January 2017.