Advances in Engineering Software 69 (2014) 1–11

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Empirical study of open source software selection for adoption, based on software quality characteristics

Mohamed Sarrab [a],*, Osama M. Hussain Rehman [b]

[a] *Communication and Information Research Center, Sultan Qaboos University, Oman*
[b] *Electrical and Computer Engineering Department, Engineering Faculty, Sultan Qaboos University, Oman*

ARTICLE INFO

Article history:
Received 16 June 2013
Received in revised form 28 October 2013
Accepted 1 December 2013
Available online 15 January 2014

Keywords:
Information quality
Open source software (OSS)
Quality characteristics
Selection criteria
Service quality
System quality

ABSTRACT

Currently, open source software (OSS) products have started to become popular in the market as an alternative to traditional proprietary or closed source software. Governments and organizations are beginning to adopt OSS on a large scale and several governmental initiatives have encouraged the use of OSS in the private sector. One major issue for the government and private sector is the selection of appropriate OSS. This paper uses new internal quality characteristics for selecting OSS that can be added to the dimensions of DeLone and McLean information systems' model. Through this study, the quality characteristics are organized in a two level hierarchy, which list characteristics and sub-characteristics that are interconnected with three main dimensions: system quality, information quality and service quality. These characteristic dimensions are tailored to the criteria having been built from literature study and standard for software quality and guidelines. This paper presents case study results of applying the proposed quality characteristic on eight different open source software that are divided between open source network tools and learning management systems.

1. Introduction

Open source software (OSS) is software tool that operates under an open source label whose source code should be available and modifiable. In OSS, the software suite must contain the source code orbe available at a freely accessible location, where it may be adapted to individual customized requirements and passon the modified format. Some OSS may reserve re-distribution rights but in other cases it might be free. A distributor or developer might charge for services including special training, installation, programming and technical support, etc. In general, the term OSS refers to software that is freely available, widely accessible and reusable [1], where reusability implies that source codes can be modified to make them work as their users require. In general, developments in information and communication technology support the existence of OSS strongly.

Throughout the world an increasing number of people are using OSS with an open source code, where it generally operates securely and reliably in a stable and cost-effective manner. The increasing popularity of OSS has dramatically changed the software industry in recent years. OSS is often seen as a possible solution to some of the challenges presently faced by many software communities, especially among developing countries. Such challenges include controlling piracy, exerting a greater level of control over acquired software and dealing with broader policy perspectives that pertain to the development of a national domestic software industry.

OSS does not use the same mechanism of development as proprietary software. From a policy perspective, an 'open source' can be defined as a software-licensing model where the software's source code is made available, subsequently modified, redistributed and added-to, although often with certain restrictions. In addition, a range of benefits under commercial arrangements may be made available, such as, updates, training and ancillary software services [2,3].

OSS are usually, though not exclusively, developed by the collaborative efforts of a group of people who contribute components to the final version of the software. Software companies may also produce programs for the open source community. Proprietary or commercial software is developed under commercial rules and policies, in other words, it is licensed for a fee to a customer in binary, object or executable code (either directly or through channels). The company that writes the program usually provides updates, training and other services required by its users so that the software works efficiently.

The software's source code might be made available to a certain number of its users through a special license or an alternative agreement but often remains unavailable to the general public and may not be copied, changed or modified except in a manner provided for under the terms of a prior agreement. The European Union recommended that OSS should be used preferentially; the doubters were made to realize that OSS with its freely accessible

* Corresponding author. Tel.: +968 24143698.
E-mail address: sarrab@squ.edu.om (M. Sarrab).

0965-9978/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.advengsoft.2013.12.001

source code is not just the toy of enthusiastic computer science students but systems designed by professionals for professional use. The many organizations that use open source products include, NASA, Daimler-Chrysler, SONY, Boeing, Sixt, IKEA, the World Council of Churches and the National Association of German Skilled Crafts. Moreover, companies, such as, Hewlett–Packard, Siemens Fujitsu, IBM and Compaq are now selling computers with Linux pre-installed operating systems. These companies make it clear this operating system is just as competitive as any proprietary operating system, such as, Solaris or Windows NT [4].

Each software model (open source software and Commercial Software) represents a viable business strategy for their companies as well as supporting and providing their customers with real advantages. Solutions that the software offers are being continually updated with providers concentrating and improving on a variety of emerging issues and problems, such as, addressing reliability, security and information flow control issues. In fact, OSS has surprised many in the industry by acquiring a good reputation for its reliability, efficiency and functionality [5].

The main objective of this study is to discover criteria that help organizations in their selection, development and evaluation of OSS. In particular, this study aims to identify the most relevant dimension, i.e. system quality, information quality or service quality, which derive or motivate users and IT decision makers in selecting their OSS products. The contribution of this work is based upon the literature study and standard for software quality and guidelines to understand all possible selection criteria or quality characteristics used by potential users to evaluate OSS products for adoption. To achieve the objective of this study, it is necessary to investigate empirically the selection process and quality characteristics of OSS in different application domains. This new empirical knowledge about open source software selection for adoption refers to the qualitative analyses of software quality characteristics.

The proposed model will be used to identify the characteristics considered significant by users and IT decision makers to select and adopt the appropriate OSS product. This research can be used to gain an understanding of the quality characteristics of OSS with the ultimate goal of improving software development practice in industry, particularly of the selection of OSS. This paper provides an overview of open source software and discusses criteria for choosing it while proposing a new internal quality characteristic for selecting OSS that can be added to each DeLone and McLean information system model [6].

2. An overview of open source software

In the 1960s, buying a new computer meant supplying free extra software. Manufacturers were only paid for the computer's hardware. The source codes were freely available and accessible to programmers throughout the world. In 1965 IBM has stopped supplying software source codes with their computers' operating systems. The company started to employ sufficient computer and information technology experts and is able to dispense with external developers.

In 1970s, computer programmers started realizing that they earned a lot of money from software development. The programmers safeguarded their sources of income using license agreements that prohibited or limited users passing software to others. After 1970, freely available software source codes virtually ceased to exist. Software began to be produced behind closed doors [1,7]. All manufacturers started to maintain control over their tools and computer users had to rely and trust the software producers in the event of program code faults, errors or any special modification according to users' wishes or new requirements. In 1984 Richard Stallman from Massachusetts Institute of Technology (MIT) provided the idea of developing a free program package named GNU. The goal of the (MIT) computer experts was to again enable open cooperation between software developers. According to the father of GNU, software is only free when it can be used by anyone without restrictions or limitations. Richard Stallman provided the GNU General Public License (GPL) for protecting the software's freedom. The term 'free' has many meanings in the English language. Because of that for several interested companies the title 'free software' sounded very much like 'free toy'. Many companies were hesitant or shy to have any operating system initiatives that were generously given away to everyone [8,9].

3. Criteria for selecting OSS

This study is based on the DeLone and McLean Information System Success Model (see Fig. 1), which has been used as a useful framework for measuring the organizational abilities of information systems. This model is widely used for understanding and measuring the dimension of information system success [10–12]. The original model of DeLone and McLean consisted of six major success dimensions: system quality (software quality), information quality (source code quality), service quality (expected and received service quality), intention to use (adoption), user satisfaction (process of discovering user satisfaction level) and net benefits (economic evaluation in terms of money and time consumption).

The modified model covers a wide range of system features [13]. Its quality characteristics of information system success with the new sub-characteristics (internal quality characteristics) for each information system model dimension are based on:

3.1. System quality

System quality implies the desirable characteristics of the OSS product, including availability, reliability, performance, usability and functionality.

- *Availability* of software services and any release of new software's version with new added features. The software's new version release at the targeted or expected time with mainly new functionality. The availability of any books, websites, forums and blogs written about this software available in market.

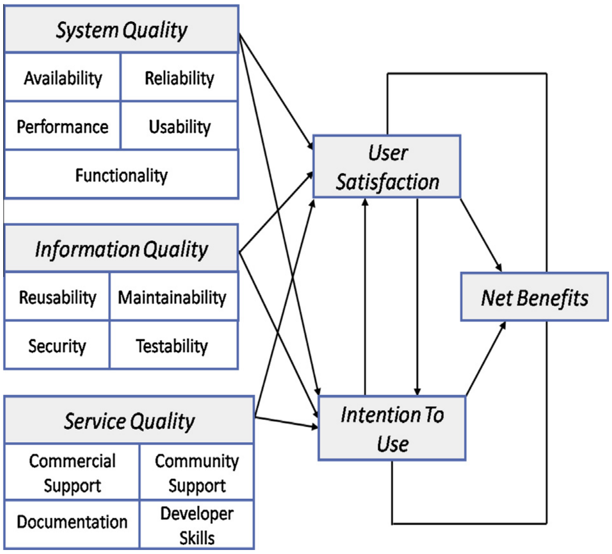

Fig. 1. D&M IS success model with added components of each dimension.

- *Reliability* implies maturity and popularity. Maturity refers to whether the software is new in the market or not. Popularity relates to the different software users and any catalogue, guidebook, manual, and website written about this software being available. Thus, reliable OSS must be designed to be as fault tolerant as possible. Fault tolerance refers to making and supporting the system's function in the presence of faults.
- *Performance* is the most important feature, where every OSS must aim and strive for maximum performance, beeasy to install, configure and operate within a short time.
- *Usability* in OSS refers to its learn-ability, operability, accessibility and user interface. Learn-ability refers to the ease to master and understand the software, where it might even be without using the user manual. Operability is related to how easily the software operates. Accessibility refers to how easily the software is accessed without any other third party software or plug-in.
- *Functionality* refers to achieving the user's expected requirements, i.e. the correct output and the software functions appropriately as required [14–16].

3.2. Information quality

Information quality includes the following desirable characteristic of source code.

- *Maintainability* refers to the software's modularity and modifiability. Software modularity relates to its code structure, readability and how well the software is designed. The software's modifiability is its ability to be customized to meet a user's requirements.
- *Reusability* is very important because it focuses on the easy to reuse or extent the code for further extension or integration. Here the source code can be used again to add new functionalities with slight or no modification.
- *Testability* enables the software to be free of errors. The software's verification and testability is a key aspect that allows the detection of difficulties and defects in OSS. Software testability supports the testing process and facilitates the creation of better quality software.
- *Security* includes confidentiality and integrity. The confidentiality of OSS refers to data and information security and that the software is free of vulnerabilities. Integrity is related to the availability of control mechanisms to ensure the system's integrity. The OSS should provide the level for a user's authentication [17–19].

3.3. Service quality

Service quality is based on SERVQUAL measurement instruments, which are adapted from the field of marketing and has been recognized as an important component of information system success [15].

- *Commercial support*: It is especially important to search for the available support options before deploying the software. This step is even more crucial when there is insufficient experience with OSS. The number of OSS systems with commercial support is relatively small. There is lots of free resources out there but the support available through them is comparable to and sometimes even better than traditional commercial support. Third party commercial support is available from some companies. These companies range from large corporations, such as, IBM and Sun Microsystems; to specialist open source organizations, such as, Red Hat and MySQL and to local firms and independent contractors.
- *Community support*: An active community behind your chosen OSS is very important, because it is always where one will usually go for support, news, advice and tips. The chosen community should have the required skills and knowledge. Both the development team and community should have a good record of performance. Community communication is also important for acknowledging problems and help in solving them. The value of OSS online communities may not always be obvious to a newcomer but are very useful when it comes to making the OSS works better.
- *Documentation*: In terms of quality of service, OSS should provide complete documentation for requirements, architecture, technical and user manual. OSS documentation should consist of the requirements documentation, the statements that identify attributes, capabilities and the qualities of a system. Architecture or design documentation should provide an overview of OSS that includes its relationship to an environment and construction principles to be used in design of OSS components. Technical documentation includes the documentation of codes, algorithms and interfaces. End User documentation involves manuals for the end user and support staff.
- *Developer skills*: Considers the skill set of an individual developer or a group of developers. They should have the appropriate skills to deploy and maintain OSS. If the developers do not have the relevant skills it is possible to employ third party contractors or implement a training program to match the developer's skills to the task [20,21].

3.4. Intention to use

Many of the OSS systems are quite flexible and can be used and modified for different purposes but it is still important to consider their relative strengths and weaknesses. Thus, planning for the current and future use of OSS is important and is a key point to take into account when making an OSS selection.

3.5. User satisfaction

User satisfaction remains an important way of measuring customers' opinions of OSS products and should cover the entire customer experience cycle from information retrieval through purchase, payment, receipt and service. Higher OSS system quality is expected to lead to higher levels of user satisfaction. Its use has led to positive impacts on individual productivity, resulting in higher output throughout the organization. User satisfaction levels can be measured using different techniques, such as, repeat visits and user surveys [10,12].

3.6. Net benefits

Net benefits are an economic evaluation of OSS in terms of money and time consumption. Net benefits are very important measure of success. It captures the effect of the OSS product on the customers, industry, suppliers, employees, economy, organizations and even society. Net benefits are very important in terms of cost and time saving [10,12].

The following topic areas are important when considering open source software:

Hardware compatibility is very important factor in selecting OSS. The chosen OSS has to be able to support used computer parts and the types of devices that are in use. Another crucial factor is the type of operating system that is supported by a computer vendor. Buying from a small seller or local vendor the chance of getting support is slight. On the other hand, buying from a large company, such as, HP, IBM or Dell, one has the possibility to check what OSS operating systems they support [22].

A good reputation for OSS performance and reliability is a very important factor. Some OSS vendors have very good industry reputations, such as, Linux, Apache web server, GNU Compiler Collection (GCC) and Samba. To select an OSS based on reputation one should compare the OSS against its proprietary peers. Good software quality considers all aspects of the software product and it does not necessarily reinvent the wheel, starting from scratch or forcing programmers to learn new languages or using complex data formats [22].

4. Case study

The case study focuses on studying the quality characteristic of selected open source systems. The quality characteristics in the provided model are organized in a hierarchy of two levels described as characteristic and sub-characteristic that are associated with three main dimensions: system quality, information quality and service quality. Both characteristic and sub-characteristics are considered to contribute in some way or other to the dimensions to which they belong. To ensure that these quality characteristics are correctly ranked, a score has been assigned to each of sub-characteristic. The suggested weighting for each criterion is scaled from 1 to 5, where 1 is the least important, 3 is neutral while 5 is the most important criterion.

The suggested selection criteria differ from D&M IS Success Model as information quality involves the source code quality, system quality is referred to software product characteristics and service quality implies the expected and received service quality from software provider. These criteria may contribute to the success of software product adoption among potential users.

With the huge number of open source software systems available today, two different open source software categories were selected for the case study to demonstrate the feasibility and effectiveness of the proposed approach, i.e. the DeLoneand McLean information system success model with added components of each dimension (see Fig. 1). Two different open source application domains were selected as case studies, i.e. open source network tools and learning management systems.

Currently a huge number of open source network tools are available; therefore, the choice of tools for conducting the case study was important to reflect sufficient assessment of the modified model. In order to demonstrate the applicability of the modified model few of the most common open source network tools were used in this case study, which includes Network Simulator 2, Network Simulator 3, Wireshark and FileZilla. Moreover, there are many learning management systems available today and making the decision which learning platform to choose can be quite overwhelming. To prove the applicability of the proposed approach in different application domains the most popular learning management systems were selected to be evaluated, that includes Moodle, Sakai, ATutor and ILIAS.

The paper discusses various open source software as case studies and evaluates them according to the proposed criteria. The proposed selection criteria are independent of the software types. Eight open source software are evaluated among the two broad categories of considered software from learning management system and network tools. The reason behind considering eight tools is to provide large number of case study examples, to show the feasibility of the provided approach of open source software selection for adoption, based on software quality characteristics. The selected eight tools are weighted based on various research literature surveys, technical reports, performance assessment online articles, tools documentations and user's manuals, users' community feedback, personal knowledge and usage experience. After extracting the required information from the concerned resources, a summarized qualitative assessment is provided for the underlying tools.

The qualitative assessment is converted into scaled form for the internal quality characteristics of the tools. Furthermore, quantitative representations can be used for further analysis, such as averaging the internal quality characteristics of the individual tools. Fig. 2 presents the methodology in order to deduce the scalar representations of the proposed internal quality characteristics for the eight selected open source software. A method of three steps is considered for deducing the scaled values, where the steps are carried out in a sequential manner for all OSS.

The first step covers information gathering based generally on literature study and users' feedback, which also represents bulk of the scaled based analysis. The first step highlights the most common and relatively authentic resources that were relied on inferring the suggested values. While analyzing each software tool, all resources are not taken into account since not all listed resources are available for each tool. For instance, the performance of ns-2 and ns-3 are discussed in various research papers but the same cannot be said for the other two network tools FileZilla and Wireshark. Therefore, the analysis of FileZilla and Wireshark did not include published research papers, but included all other listed resources. The second step as indicated in Fig. 2 concludes the gathered information from step one. The concluded statements are put as *comments* in the case study tables of each tool as given in Tables 1–4 and 8–11 based on the summarized comments of each internal quality characteristic of the tool. Where, it inferring summarized description of the open source software corresponding to the internal quality characteristics. The third step provides a scaled value representation of the concluded comments. In which, the scale (weight) represents the internal quality characteristics of the open source software.

4.1. Network tools

- *Network simulator 2* (ns-2) is a discrete event network simulation tool designed for research in communication networks and licensed under GNU GPLv2. Ns-2 coding is based on C++ &OTcl programming languages that were designed to run on Linux based operating systems but can also be used on Windows platforms using Linux emulators, such as, Cygwin. Various

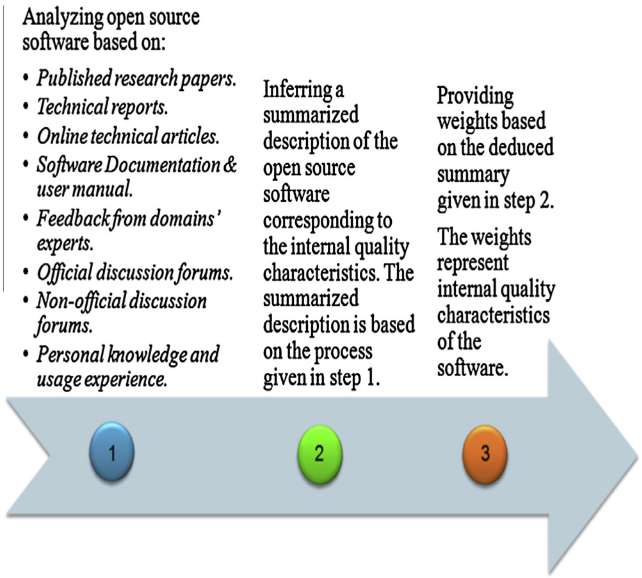
Analyzing open source software based on:
• Published research papers.
• Technical reports.
• Online technical articles.
• Software Documentation & user manual.
• Feedback from domains' experts.
• Official discussion forums.
• Non-official discussion forums.
• Personal knowledge and usage experience.
Inferring a summarized description of the open source software corresponding to the internal quality characteristics. The summarized description is based on the process given in step 1.
Providing weights based on the deduced summary given in step 2.
The weights represent internal quality characteristics of the software.
1
2
3

Fig. 2. Steps of weighting assignment to the internal quality characteristics of the open source software.

Table 1
Quality characteristics for network simulator 2 (ns-2).

Feature	Sub-feature	Scale	Comments
System quality	Availability	5	Ns-2 had consistent maintenance support and proper updates before the introduction of its successor ns-3. Currently ns-2 is maintained lightly. To date, ns-2 is still more popular than ns-3 and has a huge user's community with various updated online support, forums and documents
	Reliability	4	Ns-2 is relatively an old network simulation platform that has been refined constantly with time. It is considered the single most popular network simulation tool to date but also suffers from various bugs and compatibility issues. Although ns-2 has various active issues, it also has a large and global community of users and hence users are able to find ways to go through its faulty components
	Performance	4	Ns-2 can provide good performance with respect to the required simulation time and results accuracy but it is also known to be heavy on resource consumption, such as, memory and processor usage
	Usability	2	Ns-2 is not an easy tool to install and requires a number of configurations before it becomes finally installed. Users may become frustrated due to platforms compatibility issues. Additionally, the learning curve of ns-2 is not an easy one to follow, especially when learning it in solo mode. It can become difficult to understand the library structure of the simulator. Its operation prerequisites good skills in C++ programming
	Functionality	4	Based on a number of network simulation tools comparisons, the results generated by ns-2 are deemed relatively reliable and accurate. In addition, simulation speed is acceptable by most researchers whereas scalability is a major limitation for large networks performance analysis
Information quality	Maintainability	4	Various open source modules are available as freeware components that can be added into ns-2 with few configurations. A non-exhaustive list of ns-2 contributed codes that have not yet been integrated with ns-2 releases can be found. On the other hand, ns-2 is highly modifiable, including its core libraries, due to its open source regime
	Reusability	4	Ns-2 codes are completely reusable but to make use of these codes it requires expertise in the structure of the ns-2 simulator. Generally, ns-2 is known for its complexity inheritance hence making code reuse tasks difficult to achieve
	Testability	3	Ns-2 has a number of active bugs to negotiate but it has a high fault tolerance level, which is one of the features that make it popular
	Security	–	As a simulation platform, ns-2 does not qualify as tool that should be concerned with data security issues
Service quality	Commercial support	1	Various contributions in ns-2 came from individual intellects, professional organizations and academic institutions. Currently ns-2 does not have any commercial support that are constantly developing, trouble shooting and technically supporting this tool
	Community support	5	Ns-2 has a strong community support with various support forums, mailing groups and a large number of researchers and students [33]
	Documentation	4	Ns-2 has a good series of documentation based on different releases. The last document was released in November 2011, for the latest version ns-2.35. It is usually hard to find documentation on other ns-2 source codes that could be added as additional modules
	Developer skills	4	Developers are skillful in developing and integrating new and stable modules into ns-2 but the lack of the support of a specialized commercial company has a negative effect upon managing complexity and architecture organization

Table 2
Quality characteristics for network simulator 3 (ns-3).

Feature	Sub-feature	Scale	Comments
System quality	Availability	3	Ns-3 is an actively developed network simulation platform. It has a growing user community and its popularity is rapidly increasing and in time is expected to completely replace its predecessorns-2. In addition, there are consistent updates in the form of newer released versions
	Reliability	3	Ns-3 is still considered a new addition to network simulation platforms. It will require few more years to become a major reliable and stable option for network simulations but many researchers from various areas of research in communication networks are shifting from ns-2 and towards ns-3
	Performance	4	Similar to its successor, ns-3 can provide simulations results within a suitable time frame but it also needs to perform a number of configuration steps before having it completely up and running
	Usability	3	Usage of ns-3 will require good programming skills in C++ and python but due its better architectural organization its learning curve is expected to be much easier compared to its predecessor ns-2
	Functionality	4	A good set of research work and papers of a high journal standard are being produced using ns-3. Hence, results generated by ns-3 are relied on by the research community
Information quality	Maintainability	3	Ns-3is limited in terms of its options for available addable modules since its number of users is much less compared to other simulation platforms. Conversely, it is easy to modify the source code as required
	Reusability	5	Ns-3 codes are completely reusable but require expertise in understanding ns-3 structure to make use of the source codes
	Testability	4	Ns-3 is expected to be more stable and fault tolerant than its predecessor ns-2 because its developers have the advantage of learning from the bugs present in ns-2
	Security	–	As a simulation platform, ns-3 does not qualify as tool that should be concerned with data security issues
Service quality	Commercial support	1	Similar to ns-2, ns-3 has no commercial support until now
	Community support	2	The ns-3 community is small relative to other simulation tools but is rapidly increasing
	Documentation	4	A large library of documentation is maintained by the official ns-3 website where various tutorials and online guides are available
	Developer skills	4	Ns-3 development teams are skillful in their work and have the advantage of learning from its ns-2 predecessor

version of ns-2 have been released. The latest version is ns-2.35, released on 4th of November 2011 along with its complete documentation [23].

- *Network simulator 3* (ns-3) is a discrete event network simulation tool based on C++ and Python programming languages. Ns-3 is the successor of ns-2 but with a new development

Table 3
Quality characteristics for Wireshark.

Feature	Sub-feature	Scale	Comments
System quality	Availability	3	Wireshark is not meant for long term heavy throughput network monitoring where its GUI is demanding on system resources, especially memory. Instead, it is meant for low throughput network monitoring or offline packet analysis. In addition, various versions have been launched for Wireshark, initially known as Ethereal
	Reliability	4	Wireshark is the most widely used packet analyzer, especially for professional and educational works. It provides detailed and accurate information on the status of the packets that have been captured. When capturing WiFi packets, the system cannot properly decompose the frames since it requires pre-configurations, which are network card and operating system dependent
	Performance	4	A powerful packet analyzer with a high and accurate performance track record, however, it has high memory consumption, which makes it unsuitable for long time duration live packet capture and analysis unless it is controlled by making a few configuration modifications and re-coding using its command line interface, tshark
	Usability	5	It is a straight forward and easy to use packet analyzer. It can be self-learned. One only requires having a basic knowledge about communication networks
	Functionality	4	Results generated by Wireshark are highly relied upon hence proving its good functionality. It, however, requires a few configurations for usage in WiFi based packet capturing scenarios or else the results are mapped to Ethernet frames by default
Information quality	Maintainability	4	The software is well designed and maintained and the analysis process could be highly customized based on its programmable features using tshark
	Reusability	5	Being an open source tool, Wireshark can be modified and customized. Individuals can contribute their codes among the developer teams
	Testability	4	Wireshark is an established packet analyzer but has a number of errors with which deal, especially when capturing packets from wireless interface cards
	Security	–	As a packet analyzer, Wireshark does not qualify as a tool that should be concerned with data security issues
Service quality	Commercial Support	4	Commercial support, training, and development services are provided by CACE Technologies only at this moment in time
	Community Support	4	Community support is available based on Wireshark user's mailing list. Wireshark is also discussed in many other open forums
	Documentation	5	An organized document system is used to disseminate information on Wireshark, including books, tutorials, video lectures and certifications
	Developer Skills	5	Wireshark has a well-organized developmental approach. It maintains a developer's guide and mailing list and a commercial support organization that brings forth new developments

Table 4
Quality characteristics for FileZilla.

Feature	Sub-feature	Scale	Comments
System quality	Availability	4	Currently FileZilla server software is Windows oriented. It has a good reputation for service availability among its moderately sized user communities. On the other hand, service availability keeps on improving with the release of newer versions
	Reliability	4	FileZilla is a well know and regularly used FTP program. It can provide a reliable service for multiple users simultaneously
	Performance	4	FileZilla has a good performance record through the years where it won many awards, including Project of the Month in 2003 hosted by SourceForge.net
	Usability	5	The software is easy to install, configure and use. Only basic knowledge of communication networks and systems administration is required. The installation process is a straight forward Windows applications setup process. The default configuration at setup will work fine with most users
	Functionality	4	Desired results are achieved by its community of users where it is considered a suitable FTP application for a small to medium amount of file sharing
Information quality	Maintainability	4	The software is well maintained and frequently upgraded. It can be customized to meet ones needs
	Reusability	4	Being available as open source software, it can be reused by other developers but it requires good documentation to understand its various modules
	Testability	3	Good performance has been demonstrated by FileZilla but a few concerns regarding its scalability and security have been raised
	Security	3	Some concerns were raised about users' passwords residing in un-encrypted files hence making it easier for intruders to obtain sensitive information
Service quality	Commercial support	1	FileZilla has no commercial support
	Community support	4	A forum of users is maintained by FileZilla. It is not a large forum but many resources are availableon the internet.
	Documentation	3	A reasonable amount of documentation is available on FileZilla but mainly limited to reports and books.
	Developer skills	5	Good development teams are backing up FileZilla and many versions with many upgrades have been noticed throughout the years. The other noticeable thing about this software is its client side cross platform support for Linux, Mac OS and Windows, which indicates that the development team is well skilled in working with different kinds of operating systems

architecture written from scratch. Ns-3 is not backward compatible with ns-2. The simulator is publicly available under the GNU GPLv2 license for research and development. The latest stable version is ns-3.16 released in December 2012, along with its documentations [24].

- *Wireshark* originally known as Ethereal is a cross-platform packet analyzer. Wireshark is used for wired and wireless networks monitoring, troubleshooting, maintenance and analysis. Wireshark and its terminal based version called TShark are released under the GNU GPL terms [25].

Table 5
System quality of selected open source networking tools.

Tools characteristics	ns-2	ns-3	Wireshark	FileZilla
Availability	4	3	3	5
Reliability	4	4	3	4
Performance	4	4	4	4
Usability	5	5	3	2
Functionality	4	4	4	4

Table 6
Information quality of selected OSS networking tools.

Tools characteristics	ns-2	ns-3	Wireshark	FileZilla
Maintainability	4	3	4	4
Reusability	4	5	4	5
Testability	3	4	3	4
Security	–	–	3	–

Table 7
Service quality of selected open source networking tools.

Tools characteristics	ns-2	ns-3	Wireshark	FileZilla
Commercial support	1	1	4	1
Community support	5	2	4	4
Documentation	4	4	5	3
Developer skills	4	4	5	5

- *FileZilla* is a program for transferring files between clients and servers using file transfer protocol (FTP). FileZilla contains separate client and server side programs, where both client and server software are distributed under the terms of GNU GPL license. In this study, we will concentrate on the server side unless explicitly stated, where the server is only available for Windows OS at the moment. On the other hand, FileZilla client program offers multiple platform support [26].

Table 1 shows the quality characteristics based on system quality, information quality and service quality dimensions for Network Simulator 2.

Table 2 shows the quality characteristics based on system quality, information quality and service quality dimensions for Network Simulator 3.

Table 3 shows the quality characteristics based on system quality, information quality and service quality dimensions for Wireshark.

Table 4 shows the quality characteristics based on system quality, information quality and service quality dimensions for FileZilla.

4.2. Discussion of network tools

A system quality scaled assessment for the four OSS tools are presented in Table 5 including an outline that may help decision makers apply a similar technique to other networking related tools they may be interested in for their usage for their corporation. In Table 5, Availability of ns-2 is scaled to 5 since it's the single most popular network simulator whereas ns-3 is scaled to 3 due to its relative lack of popularity [27,28]. Conversely, both network simulators have a consistent updates and releases [29]. In addition, Wireshark Availability is scaled to 3 due to its lack of long duration packet capture support. FileZilla is scaled to 4 with the limitation of server side cross platform support, which is currently available to only Window based systems [30].

The reliability of all four tools is scaled to 4, except for ns-3 due to its relatively lower life span. Performance and Functionality subfeatures are scaled to 4 for all selected networking tools hence giv-

Table 8
Quality characteristics for moodle.

Feature	Sub-feature	Scale	Comments
System quality	Availability	5	It is available in about 78 languages and present in around 216 countries
	Reliability	5	Moodle is thought to be the most popular open source LMS system in use today. Moodle has features that allow it to scale to very large deployments and hundreds of thousands of students, yet it can also be used for a primary school or by an education hobbyist
	Performance	5	Moodle's core design is meant to be as simple and efficient as possible. Research from the eLearning Guild has found that Moodle is the leading LMS platform when it comes to satisfac tion with respect to its ease of installation and time to implement it
	Usability	5	It is the leading LMS platform with respect to satisfaction with ease of use and it offers a great deal of customization through the upload of CSSs. Moodle's themes are freely and easily downloadable where users can upload any tailored CSS
	Functionality	4	Moodle is designed to be highly modular and numerous developers have created plug-ins and other add-ons to increase functionality over the years. Users, however, argue that it relies too heavily on third-party add-ons to create functionality rather than including it as part of the core product
Information quality	Maintainability	3	The reliance on third-party add-ons to create functionality rather than including them as part of the core product can increase the workload for maintaining and updating the software as new versions are released.
	Reusability	3	Moodle has its detractors. Some users complain that its interface design does not offer enough options for rebranding and customizing its look and feel without extensive coding work. It provides, however, ten default course look and feel templates. Institutions can create their own look and feel templates across the entire system
	Testability	4	The software products are usually well-tested due to a huge number of user-base/testers who are geographically distributed and have varied skill sets and could test the module/feature independently
	Security	4	Authentication is modular using a very straightforward module format. The system defines its own authentication module structure. It uses a user access control mechanism with basic username and password authentication. The system can authenticate against a variety of sources, including external databases, LDAP directory servers, IMAP, POP3 and secure NNTP and Unix users through PAM
Service Quality	Commercial support	4	Moodle Partners provide services, such as, hosting, customization, support and training. Moodle Partners contribute 10% of their earnings to support the development and maintenance of the Moodle project
	Community Support	4	Moodle is very popular LMS with nearly 54,000 registered sites representing over 200 countries, 44.3 million users and 4.6 million courses. Moodle's community is both larger and more (inter)active than all other LMSs. It has a strong medium-size end user and developer's community
	Documentation	5	Moodle is better documented in every aspect compared to other LMSs. Administrators, teachers, students, and developers all have better documentation for Moodle
	Developer skills	–	The software developers require skills in PHP

Table 9
Quality characteristics for Sakai.

Feature	Sub-feature	Scale	Comments
System quality	Availability	4	Sakai is available in about 12 languages and is present in approximately 17 countries
	Reliability	3	As a new platform, Sakai has not yet achieved large penetration outside the higher education marketplace. Its reputation for higher-end features and scalability has made it popular with large universities that need a robust solution
	Performance	3	On the negative side, Sakai's critics point out that it lacks comprehensive competency profiling and management, which makes it unsuitable for some large enterprise environments. It can also be challenging to integrate Sakai with other enterprise software systems, such as, talent management, other HR software suites and ERP solutions
	Usability	4	It is a straight, forward and easy to use open source LMS platform. It can easily be customized and rebranded including its collaboration tools
	Functionality	4	Analysts and users consistently rank Sakai at or near the top among open-source LMS platforms for its reporting features functionalities
Information quality	Maintainability	2	Sakai's code is somewhat complex in terms of its structure and readability. The software's modifiability is dependent on its ability to be customized to meet a user's requirements. That makes this system less suitable for simple, rapid deployment projects that require a LMS
	Reusability	4	Sakai has the ability to change the settings of all its tools based on roles. The system, therefore, permits users to customize each tool. It supports code reuse or extent for further extension or integration
	Testability	3	Any individual from the community who is interested in a particular feature can test the developed code for any potential bug(s)
	Security	3	Some concerns have been raised over user authentication but overall Sakai has a good security reputation
Service quality	Commercial support	3	The Sakai eco-system is evolving with a small number of commercial vendors now providing services for the market, such as, hosting
	Community support	3	The Sakai community of users is very active in terms of helping each other
	Documentation	3	A good amount of documentation is available on Sakai
	Developer skills	–	Sakai is written in Java. Fully skilled developers with expertise in Java are required to improve or customize Sakai as required

ing them good characteristics as OSS tools. Finally, the Usability of ns-2 is scaled with 2 due to its complex architecture and difficulty of its usage. Ns-3 is scaled to 3 due to its lower architecture and usage complexity compared to its successor ns-2 [31]. Wireshark and FileZilla are scaled 5 for their usability, since they are both easy to use and deploy [30,32].

Service Quality scaling of the selected networking tools is presented in Table 6. Maintainability of ns-2, Wireshark and FileZilla are scaled to 4 due to good support and contributions by their developers and user community. Ns-3 is scaled to 3 due to its smaller usage domain. Reusability of both ns-2 and FileZilla are scaled to 4 where the former suffers from software design architecture complexity and the latter from lack of strong documentation [25,30,33]. On the other hand, ns-3 and Wireshark are scaled with a Reusability scale of 5 due to their organized software design architecture and good documentation support. Testability of ns-2 and FileZilla is scaled to 3 whereas ns-3 and Wireshark are scaled with a Testability scale of 4, where the scaling of each software was based on fewer of more bugs and fault tolerant issues than the other. FileZilla has a password security concern raised by its user community and hence qualifying it with a Security scale of 3. Ns-2, ns-3 and Wireshark do not qualify for Security assessment, since no data security features are applicable on them [34,35].

In Table 7, Service Quality for the selected networking tools is presented. As a OSS, ns-2, ns-3 and FileZilla do not have any consistent Commercial Support with respect to development, technical support and troubleshooting, hence a scale of 1 is assigned to all three networking tools. Wiresharkis commercially supported by a company named CACE Technologies, hence providing it with a scale of 4. Among all four selected tools, ns-2 surpasses all others with its huge user community hence providing it with a community support scale of 5 [24,36]. In comparison, ns-3 is still new in the OSS market and lacks popularity and a large user community, therefore, it was assigned a community support scale of 2. Wireshark and FileZilla each have a reasonably large community of users, hence they were assigned a community support scale of 4. The volume of documentation forns-2 and ns-3 is large but there is a lack of professionally written books, they were assigned a community support scale of 4. Wireshark has good, professional support documentation with respect to books, manuals, certification facilities and guidelines, hence providing it with a scale of 5 for community support. FileZilla does not have strong documentation and was scaled 3 for community support.

The developers for all selected OSS tools were rated as skilled. Ns-2 and ns-3, however, were scaled 4 due to a lack of dedicated commercial support and, therefore, a dedicated development process. Wireshark and FileZilla are scaled 5 due to dedicated commercial development and client side cross platform support respectively [25,26].

4.3. Learning Management Systems (LMS)

Learning management systems are called Course Management Systems (CMS) or Virtual Learning Environments (VLE). Most of them, especially the open source systems, are designed for the formal education environment. Open source ELMS are gaining increased recognition and adoption by education organizations and some companies [37]. With the huge number of learning management systems available today, making the decision about which platform to choose can be quite overwhelming. Below is a brief introduction to four of selected Open source ELMS currently available.

- *Moodle* is an LMS for producing Internet-based course websites. It is an abbreviation for "Modular Object-Oriented Dynamic Learning Environment" although when it was first developed the M stood for Martin's after its developer, Martin Dougiamas. Moodle has been around for over ten years. Its first version was published in August 2002. It is written in PHP [38].
- *Sakai* is a free educational software platform and is used for teaching, research and collaboration. It was designed by universities to solve university issues. It was built by MIT, Stanford and Berkeley amongst others so they did not need to use home-grown systems or pay vendors. Sakai aims to suit group projects and describes itself as a Collaborative Learning Environment (CLE). It is written in Java [39].
- *ATutor* is a platform to develop and deliver online courses. It differs slightly from the other education learning management systems as it is actually a learning content management system (LCMS). The platform is particularly conceived for people with

Table 10
Quality characteristics for ATutor.

Feature	Sub-feature	Scale	Comments
System quality	Availability	4	ATutor is available in about 20 languages and is present in approximately 58 countries
	Reliability	3	Reports of concerning its reliability are encouraging and steps, such as; the use of language caching to help minimize the load in rendering the user interface on high traffic systems maximizes responsiveness
	Performance	4	Administrators can install or update ATutor in minutes or develop custom templates to give ATutor a new look
	Usability	5	ATutor is designed with accessibility and adaptability in mind thus; the biggest feature about ATutor is its accessibility and interoperability
	Functionality	3	There are no modules as such. All functionality resides in the core application. Various add on modules are available to extend its functionality. It has monolithic architecture. All functionality resides in the core of the application
Information quality	Maintainability	3	ATutoris well designed and its code is structured so as to be readable. The modules make it possible for course designers to turn tools on or off. The system allows developers to create new features and integrate them into ATutor. There is no system module architecture. Extensions must be made part of the application and are tightly coupled
	Reusability	3	Institutions can apply their own institutional images, headers and footers across all courses or categories of courses. Instructors can change the navigation tabs and tool icons that are available. Its source code can be used again to add new functionalities
	Testability	2	Whenever there is an ATutor release a broad range of time consuming manual tests are conducted to ensure the stability of the release
	Security	3	Authentication is not modular and strongly tied to the database throughout the system code. Administrators and instructors can set courses to be publicly accessible or can protect access to individual courses with a username and password. The system has a password reminder option. User logins can be encrypted with SSL
Service quality	Commercial support	4	ATutor provides support services through atutorspaces.com. These include hosting and technical support, as well as custom development and theme design
	Community support	3	The ATutor community is not that popular. It has a patcher module that has brought significant benefit to its community, as it allowed community developers to develop their features into their own patches and have them submitted for inclusion in the public source code. It has a medium size end user community but no developer community. Currently the development of ATutor is closed
	Documentation	3	Instructors can access an online manual, context sensitive help and an instructor support forum hosted on the product provider's site. Generally, it has acceptable end-user documentation
	Developer skills	-	Software developers require skills in PHP, MySQL and web server software, such as, Apache or Microsoft IIS

disabilities (tools with very good accessibility) and accessibility is its best feature. In addition, user navigational patterns can be tracked so instructors can see students use of the site and students can track their own use. It is written in PHP [40].

- *ILIAS* (IntegriertesLern-, Informations- und Arbeitsko operations-System) (German for Integrated Learning, Information and Work Cooperation System). It is an LMS based on the concepts of Personal Desktop and Repository. It supports learning content management (including SCORM 2004 compliance) and tools for collaboration, communication, evaluation and assessment. ILIAS was published and offered for learning at the Cologne Faculty of Business Administration, Economics and Social Sciences. It can be run on any server that supports PHP and MySQL. It's written in PHP [41].

Table 8 shows the quality characteristics based on system quality, information quality and service quality dimensions for Moodle.

Table 9 shows the quality characteristics based on system quality, information quality and service quality dimensions for Sakai.

Table 10 shows the quality characteristics based on system quality, information quality and service quality dimensions for ATutor.

Table 11 shows the quality characteristics based on system quality, information quality and service quality dimensions for ILIAS.

4.4. Discussion of Learning Management Systems (LMS)

The above mentioned open source LMS possess different levels of quality characteristics. They have different importance levels for system, information and service qualities. Moodle has by far has the best system quality among the selected LMS. It is the most available and popular one. It is available in about 78 languages and in approximately 216 different countries compared to the ELMS Sakai, ATutor and ILIAS, which are available in about 30 languages and present in less than 60 countries. Due to its popularity, Moodle is the most reliable open source LMS system in use today. It can be scaled for small or large deployments. The other LMS have not yet equaled the large pen etration of Moodle. The performance characteristics of Moodle were scaled to 5 because of its simplicity and satisfac tion with ease of installation and time to implement it. Sakai scaled to 3 because of its lack of comprehensive competency profiling and manage ment. ILIAS scaled to 2 because it has some logistical installation issues that need to be considered [42] (see Tables 12–14).

The usability of Moodle and ATutor, were scaled to 5. This reflects satisfac tion with Moodle with respect toits ease of use and comprehensive of customization and the good accessibility and adaptability of the design of ATutor. ILIAS was scaled to 2 due to its lack of interoperability, integration and poor accessibility standards. The functionality of Moodle and Sakai is ranked 4, because many developers try to create plug-ins and other add-ons to increase their functionality over the years. The functionality of ATutor and ILIAS was scaled to 3 because there are no modules in ATutor and in ILIAS most functionality resides in ad hoc modules without interface consistency and its code is a bit [43,44].

The information quality of the above mentioned LMS are similar. The maintainability of Moodle, ATutor and ILIAS are scaled to 3 because their software enjoy a similar level of structure, readability and design. Sakai is scaled to 2, which is lower than others due to its code complexity. The reusability of Sakai and ILIAS are higher than Moodle and ATutor, because they support software reuse and easily extend for further extension or integration. Moodle and ATutor have few limitations with respect to their user interface design [17,18]. In terms of testability, Moodle and ILIAS were scaled to 4, because of their amenability for the testing process, which facilitates the creation of better quality software. On the other hand, Sakai and ATutor are scaled to 3 and 3 respectively, because of their test process and time consumption in the manual test. The final criterion of information quality is the software security in which Moodle and ILIAS were scaled 4, because of their flexible and straightforward

Table 11
Quality characteristics for ILIAS.

Feature	Sub-feature	Scale	Comments
System quality	Availability	4	ILIAS is available in about 26 languages and is present in approximately 53 countries
	Reliability	4	The concept of ILIAS is to offer an ever adaptable online environment for learning. It provides reliable learning management system solutions, such as, course management, collaboration groups, tests, learning content repository, webcasting and online surveys
	Performance	2	ILIAS has some logistical installation issues that need to be considered. In some cases several days may be spent trying to install and configure it on Linux in conjunction with its dozen or more required third party applications
	Usability	2	ILIAS currently suffers from limitations common to many OSS products. These include: lack of interoperability and integration; poor accessibility standards
	Functionality	3	Most functionality resides in ad hoc modules without interface consistency. It is a bit complex tool in terms of its code structure and it lacks support for audio, video conferencing and time-zone or date localization which affects its overall functionality
Information quality	Maintainability	3	There is module architecture but it is a tightly coupled one. The core functionality in modules is inconsistent. Its architecture is confusing and undocumented. Each module is implemented as a class that exposes an arbitrary interface and behaves as a special case
	Reusability	4	ILIA Seasily extends its functionality using feature modules. Educators can quickly assemble, package and redistribute Web-based instructional content, easily retrieve and import pre-packaged content and conduct their courses online. Students learn in an adaptive learning environment
	Testability	4	Any individual from the community who is interested in a particular feature can test the developed code for any potential bug(s). The common ground testing could be carried out even after new versions are released
	Security	4	ILIAS has flexible support for authorization roles with role templates and user interfaces. It uses PEAR Auth library for authentication. Modules are of medium, which is modular and developed as part of the PEAR project. Complexity, but good examples and documentation are available
Service quality	Commercial Support	3	Some companies offer services and support for ILIAS
	Community Support	2	There is a small community of developers. Non-German users have little chance of influencing the product's development via participation in the various help forums. Generally, there is a small but active end user and developer community
	Documentation	2	A drawback is that this platform is almost entirely used by German speakers, which means less documentation. Poor English documentation affect stool popularity. Thus, it has limited end-user and developer documentation
	Developer Skills	-	ILIAS is a PHP/MySQL based package thus the software. Developers require skills in PHP/MySQL. The system also works with Oracle databases

Table 12
System quality of selected open source LMS.

Tools characteristics	Moodle	Sakai	ATutor	ILIAS
Availability	5	4	4	4
Reliability	5	3	3	4
Performance	5	3	4	2
Usability	5	4	5	2
Functionality	4	4	3	3

Table 13
Information quality of selected open source LMS.

Tools characteristics	Moodle	Sakai	ATutor	ILIAS
Maintainability	3	2	3	3
Reusability	3	4	3	4
Testability	4	3	2	4
Security	4	3	3	4

Table 14
Service quality of selected open source LMS.

Tools characteristics	Moodle	Sakai	ATutor	ILIAS
Commercial support	4	3	4	3
Community support	4	3	3	2
Documentation	5	3	3	2
Developer skills	0	0	0	0

support of authentication and authorization security control mechanisms. Conversely, Sakai and ATutor scaled 3, because of some weaknesses in their access control mechanisms [19,45].

Moodle scored highest while ILIAS scored lowest for service quality. Moodle and ATutor have the best commercial support in terms of the services provided by their partners. Sakai and ILIAS have limited commercial support, such as, hosting and technical support. In terms of community support Moodle is a very popular LMS and is larger and more active than all other LMSs. The Moodle community is full of discussion boards in many different languages; therefore, users can communicate with people from in/outside their organization. Sakai and ATutor are less popular but they have very active communities in terms of helping each other. ILIAS has a small community of developer because non-German users have little chance of influencing the product's development via participation in the various help forums. Moodle is a well-documented LMS compared to other ones. Here administrators, teachers, students and developers all have better documentation compared to Sakai and ATutor with their small amount of available documentation [12,20]. There are many support options for Moodle users. On the Moodle site you can find books, manuals and documents including frequently asked questions and a forum where you can ask questions. The forums usually have plenty of other users that are available and even computer programmers who are happy to offer help, advice and recommendations whatever the issue. ILIAS has less documentation because it is almost entirely used by German Speakers [21]. Developer skills refer to the skill set of an individual developer or a group of developers. The above mentioned tools do not qualify as ones that concern the skills of developers. The developers skills for each tool depends of the used language in the tool development process, such as, Moodle, ATutor and ILIAS, which use PHP while Sakai uses Java [43,44].

This case study illustrates how to apply additional quality characteristic that added to the DeLone and McLean information system models as an internal constraint to be included for users and IT decision makers in an OSS selection process to solve the problem of selecting the preferred open source tool. From the case study it can be concluded that ns-2 has the better characteristics out of the four above network tools while Moodle is the best of the analyzed learning management systems [46,47].

5. Conclusion

The study's objective was to identify the most and relevant dimension (system quality, information quality or service quality) that derive or motivate users and IT decision maker in choosing their OSS products. The paper relied on the literature study, standard for software quality and guidelines to understand all possible selection criteria or quality characteristics used by these potential users in evaluating OSS products for adoption purposes. The paper uses an additional quality characteristic that added to DeLone and McLean information system model, which is the internal constraint included for IT decision makers on OSS adoption. The hierarchical characteristic of OSS selection criteria might act to boost users' confidence in OSS product adoptions in the future. Applying the proposed model to an OSS related criteria is operable and effective in solving the selection issue in practice, which is demonstrated in the case study of eight different OSS that are divided between open source network tools and learning management systems.

This study concentrated on the quality characteristic for the selection of an OSS, however, future work, which the authors intend to perform, entails enhancing the existing work by extending this study to include other DeLone and McLean information system model components, such as, Intention to Use. Here, the work will consider planning for the current and future use of OSS and User Satisfaction, which is very important for measuring customers' opinions of OSS products, which can be measured using different techniques, such as, repeat visits and user surveys. The last component in any future research will be Net Benefits, which is closely related to economic evaluation of OSS in terms of money and time consuming. Furthermore, the proposed model characteristic dimensions are tailored to the criteria build from literature study, standard for software quality and guidelines, which may help to identify the characteristics consider by the users and IT decision makers for selecting and adopting an OSS product. The contribution of this research depended on the literature study, standard for software quality and guide lines. A comprehensive survey, therefore, should be conducted to identify the main characteristics of each OSS to be selected or adopted. Moreover, it would be valuable to review all case studies on selected tools by the experts that already used them.

References

[1] Ettrich M. Open source software: a guide for small and medium enterprises. Berlin: Thormann&Goetsch GmbH; 2001.

[2] Hasan N. Issues and challenges in open source software environment with special reference to India. In: International conference on academic libraries, (ICAL-2009); 2009. p. 266–71.

[3] Zuliani P, Succi G. Migrating public administrations to open source software. In: Proceedings of international conference e-Society 2004 IADIS. Avila, Spain; 2004. p. 829–32.

[4] Sarrab M, Elbasir M, Elgamel L. The technical, non-technical issues and the challenges of migration to free and open source software. Int J Comput Sci Issues (IJCSI) 2013;10(2(3)):464–96.

[5] Security in the Information Society – An initiative of the German Federal Ministry for Economy and Technology of The Federal Ministry of the Interior and the Federal Ministry for Security in Information Technology: "What is open source?". <http://www.sicherheit-im-internet.de/>.

[6] Sarrab M, Elbasir M, Elgamel L. An overview of the technical challenges of migration to open source software. Free and open source software conference (FOSSC-13). Muscat Oman; 2013. p. 39–42.

[7] Thomas D. Going open source software in IT opportunities and challenges. J Object Technol 2005;4(2):7–13.

[8] Huda M. Migration to FOSS: challenges. In: First free open source software conference. Sudan; 2009.

[9] PeckA.Free/Open Source's Key Programming Languages. Datamation, QuinStreetInc; 2009. <http://www.datamation.com/osrc/article.php/3828096/FreeOpen-Sources-Key-Programming-Languages.htm>.

[10] Delone W, McLean E. The DeLone and McLean model of information systems success: a ten year update. J Manage Inform Syst 2005;19(4):9–30.

[11] Petter S, DeLone W, McLean E. Measuring information systems success: models, dimensions, measures, and interrelationships. Eur J Inform Syst 2008;17(3):236–63.

[12] Chamili K, Jusoh Y, Yahaya J, Chepa N. Selection criteria of Open Source Software for adoption in Malaysia. Asian Trans Basic Appl Sci 2012;2(2):9–13.

[13] Sarrab M, Rehman O. Selection criteria of open source software: first stage for adoption. IJIPM: Int J Inform Process Manage 2013;4(4):51–8.

[14] Taibi D, Lavazza L, Morasca S. OpenBQR: a framework for the assessment of OSS. IFIP international federation for information proceedings. Open source development, adoption and innovation; 2007. p. 173–86.

[15] Bianco V, Lavazza L, Morasca S, Taibi D, Tosi D. An investigation of the user' perception of OSS quality. IFIP international federation for information proceedings. Open source software: New Horizons, IFIP Advances in Information and Communication Technology; 2010. p. 15–28.

[16] Atos origin, method for qualification and selection of open source software (QSOS) version 1.6, April 2006. <http://www.qsos.org/>.

[17] Soto M, Ciolkowski M. The QualOSS open source assessment model: measuring the performance of open source communities. In: Proceedings of the 3rd international symposium on empirical software engineering and measurement (ESEM2009); 2009. p. 498–501.

[18] Carlo D, Jesús M, Edmund H, Werner K, Bernard L, Ben L, et al. Working group on Libre Software. Free software/open source: information society opportunities for Europe. Version 1.2; 2000. <eu.conecta.it/paper.pdf>.

[19] Sarrab M. Runtime verification of information flow: policy-based runtime verification of information flow control. LAP LAMBERT Academic Publishing; 2011.

[20] Samoladas I, Gousios G, Spinellis D, Stamelos I. The SQO-OSS quality model: measurement based open source software evaluation. Open source development, communities and quality. The International Federation for Information Processing IFIP, vol. 275; 2008. p. 237–48.

[21] Nichols D, Twidale M. The usability of open source software. First Monday; 2003, 8(1–6).

[22] Reyes F. Selecting an open source operating system. Open source software technical articles. Open logic helping enterprises use open source software; 2009. <http://www.openlogic.com/wazi/bid/188150/Selecting-an-Open-Source-Operating-System>.

[23] ns-3 project. ns-3 Manual: Release ns-3.16; 2012. <http://www.nsnam.org/docs/release/3.16/manual/ns-3-manual.pdf>.

[24] Teerawat I, Ekram H. Introduction to network simulator ns2. 2nd ed. New York: Springer; 2011.

[25] Wireshark documentation. <http://www.wireshark.org/docs/>.

[26] Source Forge. <http://sourceforge.net/potm/potm-2003-11.php>.

[27] The network simulator ns-2: mailing list. <http://www.isi.edu/nsnam/ns/ns-lists.html>.

[28] ns-3 documentation. <http://www.nsnam.org/documentation/>.

[29] Rehman OMH, Bourdoucen H, Ould-Khaoua M. Impact of ns-2 as FOSS Simulation tool for research in vehicular ad-hoc networks. Free & open source software conference (FOSSC-13). Muscat, Oman; 2013. p. 34–8.

[30] FileZilla. The free FPT solution. <https://filezilla-project.org/>.

[31] Weingartner E,Vom L, Wehrle K. A performance comparison of recent network simulators. Communications, 2009. ICC '09. IEEE; 2009. p. 1–5.

[32] Gerald C, Laura C. Wireshark 101 Essential skills for network analysis. podbooks.com; LLC; 2013.

[33] Martinez F, Toh C, Cano J, Calafate C, Manzoni P. A survey and comparative study of simulators for vehicular ad hoc networks (VANETs). Wirel Commun Mobile Comput 2011;11(7):813–28.

[34] Lucio G, Paredes-Farrera M, Jammeh E, Fleury M, Reed M. OPNET modeler and ns-2 – comparing the accuracy of network simulators for packet-level analysis using a network testbed. WSEAS Trans Comput 2003;2(3):700–7.

[35] ns-3 releases. <http://www.nsnam.org/releases/>.

[36] ns-2 Contributed code. <http://nsnam.isi.edu/nsnam/index.php/Contributed_Code>.

[37] Almargni E, Abdalla G. Design and simulation of wireless network using ns-2. In: 2nd International conference on computer science and information technology (ICCSIT'2012), Singapore; 2012. p. 157–61.

[38] Moodle. Consulted on the 9th June 2012. <http://moodle.org/>.

[39] Sakai project. Consulted on the 11th June 2012. <http://www.sakaiproject.org>.

[40] ATutor. Consulted on the 9th June 2012. <http://atutor.ca/>.

[41] ILIAS. Consulted on the 9th June 2012. <http://www.ilias.de/>.

[42] Fernandes S, Antonio C, Luis S, Pantelis M., FLOSS in technology-enhanced learning, InSuEdu 2012. 1st International symposium on innovation and sustainability in education Thessaloniki, Greece, October 2012.

[43] Martin D. Catalyst IT Limited. Technical evaluation of selected learning management systems, The Open Polytechnic of New Zealand; May 2004.

[44] Aarthy K. Open source software development process for the development of open source E-learning systems school of electronic engineering, Dublin City University; August 2012.

[45] Don M. Vendors of learning management and E-learning products, Trimeritus eLearning Solutions; May 2013.

[46] Ali H, Claudio E, Tin Y, Juan M, Ulf K, Mario G. Deployment and Evaluation of a wireless mesh network. In: Second international conference on advances in mesh networks, 2009. MESH; 2009. p. 66–72.

[47] Al-Ajlan A, Zedan H. Why Moodle. 12th IEEE international workshop on future trends of distributed computing systems, 2008. FTDCS '08; 2008. p. 58–64.