A Review of Evaluation and Selection of Open Source Software in Electronic Medical Record

Omar Najeh Abdulateef¹, La-sheng YU²*, Belal Najeh Abdullahieef³ and Xiao-peng ZHENG⁴

¹,²,⁴School of computer Science and Engineering, Central South University, Changsha, China
³Middle Technical University, Electrical Engineering Technical College, Baghdad, Iraq

*Corresponding author

Keywords: Electronic medical record, Multi Criteria decision making, Software evaluation criteria, Open source software.

Abstract. The valuation and choice of unsuitable open source software in electronic medical record (OSS-EMR) groups negatively affects the functions of organization and business processes. This study provides an insight into the assessment and choice of OSS-EMR groups in two ways. Firstly, quality characteristics are determined to evaluate OSS-EMR sets according to current frameworks and quality models. Secondly, the capabilities of appropriate choose methods for resolving OSS-EMR groups problems are discussed on the basis of Multi-criteria assessment and select issues are considered for choosing the best OSS-EMR. The obtained results can be attributed to the following: (1) quality characteristics considered for the evaluation of OSS-EMR groups; (2) the gap in the assessment criteria used in EMR and other issues; and Multi-Criteria or Multi-Attribute Decision Making techniques used in framework for evaluating and selecting OSS in health informatics as part of a recommended solution.

Introduction

Quality research in engineering is highly important, and software engineering is included in this aspect. Software systems conform to a set of criteria and definite factors that are considered in quality measurement. Quality is substantial because it can guide important savings in software lifecycle costs [1].

Multiple criteria decision-making (MCDM) models have been developed as part of operational research, integrated computing and mathematics tools to conduct a subjective assessment of presentation criteria by decision-makers [2]. Choosing a software package has become highly complicated because of the following: (I) the availability of considerable software packages in the market makes it difficult to implement the appropriate software packages for the business needs of an organization; (II) incompatibility between different systems of software and hardware; (III) Lack of expertise and technical knowledge for decision makers in choosing the appropriate software; and (IV) advancement in information technology (IT). Software selection tasks are frequently under timetable pressure, and decision makers may have no experience and time to suggest a detailed selection process. Thus, decision-makers cannot use the optimal way to choose the appropriate software [3].

The advantages of MCDM will help accelerate the assessment process, thereby leading to a fast decision. This advice can also lead to increased acceptance of models in various applications. In addition, the modelling quality is evaluated, which is considered by developers as an MCDM issue to facilitate automation. In this case, the MCDM issue can be considered Part of the selection method between the different open source software (OSS) alternatives in accordance with a number of criteria[4]. MCDM goal to help decision makers choose the best alternative, identify potential alternatives between options, and rank alternatives in a descending order based on their performance[5].
This work aims to determine the characteristics, strengths and limitations of the OSS quality model. The results of this study can be used as a guide for anyone planning to use any model.

The rest of this paper is organized as follows: Section 2 shows OSS_EMR. Section 3 describes the current software quality of the open source quality models. Section 4 is a comparative study of MCDM technology. Section 5 discusses the comparisons made and summarizes the sixth part of this article.

Open Source Software-Electronic Medical Record (OSS-EMR)

The (OS) describes open source software (OSS) based on its distribution and use. open source licenses have many features. This license consists of the source code distributed with the product for free distribution, from which the work is derived, while preserving the integrity of the author's original contribution and protecting any individual or group. Contrary to the public field, opening the license agreement and copyright is important to keep the software. The OSS demonstrates the following advantages: It can be universally purchased at a low cost and is by providing the software with the necessary development tools for software, usually through personal and commercial private and government licensing, affordability, local language distribution and participation in a global network. [6, 7].

The OS and medical information network provides a truly global platform for medical informatics researchers to invent and test their inventions in real-world healthcare settings. OS exertion is likely to supply sectors, parts and or effective frameworks [8]. EMR systems store and arrange medical data by electronics. These systems exhibit the following advantages: (a) increased speed in patient interviews and easy obtainment of patient histories and documentation of treatments; (b) great access and security of medical records; and (c) help specialists and pharmacies in decreasing adverse drug events in inpatient settings, and other groups immediately acquire reliable information in ambulatory settings. Although EMR systems exhibit several benefits, they also demonstrate various problems, such as decreased patient encounters at least in the early stages of achievement [9].

Software Quality

Many researches are focused on software quality to acquire substantial accurate software quality models. Software quality research is produced because of users’ needs for system products with high quality. Meanwhile, software quality systems are defined as the degree of process, component or system that accomplished the specified requirements and meet the expectations and needs of users [10]. This study aims to create new designs based on the current OSS quality evaluation models with respect to their quality features, methodology and field of application. In this section, most main software quality models used to assess the OSS are briefly reviewed.

**McCall's Model.** (also called as McCall’s triangle of quality) is one of the software evaluation models. McCall presented eleven factors, divided into three (review, transitions and product operations). based on the software development cycle [11]. The main weaknesses of the McCall’s model are its exclusion of functionality, thereby diminishing user insights, security and vendor.

**The Boehm Model.** exhibits most of the characteristics of the McCall model, which are similar to the hierarchical quality list classified as high-level stage, intermediate and primitive characteristics. The high-level characteristic is It is called a "general utility" and represents a general model of quality [11]. However, this model is not focused on functionality, usability and security, which are important characteristics for measuring software.

**Dromey Model.** is one of the models used in evaluating software quality, model demonstrates four fundamental characteristics, namely, correctness, internal effective deployment of a component based on its intended use, contextual and descriptive. However, the model does not give an explicit explanation about the evaluation approaches or metrics to be used [11]. In addition, no studies exist regarding the manner by which the model is used in various applications [10].
FURPS Model. is a model for evaluating software, and its quality is divided into two categories: functional (F) and non-functional (URPS) [11]. However, the study does not consider portability, security and vendor quality. In addition, the model excludes the necessary metrics or assessment methods [10]. The dimensions of this model are hierarchical and classified as follows: feature and performance qualities, conformance, serviceability, reliability, aesthetics, durability and perception. Several of these scopes can be subjectively considered.

The Open Business Readiness Rating (OpenBRR), was created to help organizations find the optimal OSS and accelerate the evaluation procedure. The model content demonstrates four categories used for specific measures, and a weighting is given to each metric [12]. However, the model did not focus on the quantitative measure, which is the major part of the quality-in-use model [13]. In addition, the model excludes the functionality and vendor criteria of OSS EMR.

In OS Maturity Model (OMM), the study aims to check the usability and quality of Free/Libre OS evaluation models. [12]. However, the study did not consider functionality, transferability and operability.

(Sung et al.) Model. was evaluated for use in the selection of OSS for expansion purposes and includes four features (functionality, usability, portability and reusability) All these features and sub-features are matched with ISO 25010. Functional commonality, understanding and lawfulness are beyond the scope of the standard system [12]. However, the study did not determine the functionality criteria of OSS EMR, maintainability and Vendor criteria.

OS Maturity Model (OSMM). is designed as a tool that can be used to identify and compare appropriate options for the OSS software to mature a product built into the enterprise. This sample displays application and product indicators. Product indicators are divided into four groups (i.e. acceptance, use, integration and product). The model does not evaluate internal quality, particularly source code, which is highly important [12]. In addition, the model did not include functionality and efficiency of OSS EMR.

OS-UMM by Raza, et al. [14], presented a usability maturity model by considering usability related problems of OS projects. Moreover, the model tests the degree of coordination between OSS and its usability aspects. However, the authors only focused on usability model and did not consider other criteria, such as functionality, portability, reliability and security.

The SQO-OSS Model. can be used to evaluate an OSS quality, which supports decisions about whether or not to use them. The model assesses source code and hierarchical and community processes, which are key factors in the quality of open source software [12]. Furthermore, the model focuses on source code and community around a project [15]. However, the study did not consider functionality, transferability and usability.

The QualOSS Model. is created for assessment of FLOSS, which concentrates on robustness and evolvability. This model demonstrates two categories, namely, community- and product-related [13]. However, the study does not comprise any criteria in the quality in use [16].

Qualification and OSS Selection (QSOS) Model. specializes in comparing and choosing OS and free software. The model is performed in four independent iterative level, namely definitions, assesses, qualifications and chosen. However, the study is only focused on maintainability characteristic [12].

Sarrab and Rehman [17], presented a model using new internal quality features for the selection of OSS and added them to McLean and DeLone models. In this study, the quality features are categorised into two hierarchal levels, which exhibit characteristics and sub-characteristics consistent with the three properties: information quality, system quality and service quality. However, the functionality criteria are inappropriate for EMR, and the definitions of the metrics and approaches used in the model in practice are obscured.

ISO/IEC 9126 Model. is a standard quality model for evaluating single software. These standard exhibits six specific key characteristics of software quality assessment. [18]. The software quality model is classified into three types: internal, external and quality in use. Therefore, the ISO model works as a starting stage for software assessment. The basic quality property can be a part of the
ISO product software under study [11]. The model did not determine the functionality criteria and vendor of OSS EMR.

**ISO 25010 Model.** This standard was developed in 2010 to replace the ISO 9126 model. The ISO 25010 exhibits advance product quality characteristics [19] [4]. The model did not determine the functionality criteria and vendor of OSS EMR.

<table>
<thead>
<tr>
<th>Quality characteristics</th>
<th>McCall</th>
<th>Gravin</th>
<th>Boehm</th>
<th>Douglass</th>
<th>FURPS</th>
<th>QRMM</th>
<th>OS-MOSS</th>
<th>OpenBRR</th>
<th>Sung et al.</th>
<th>Qual OSS</th>
<th>ONM</th>
<th>SQOSS</th>
<th>Sustainability</th>
<th>PROCESS Maturity</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functional suitability</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>12</td>
</tr>
<tr>
<td>Reliability</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>10</td>
</tr>
<tr>
<td>Usability</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>5</td>
</tr>
<tr>
<td>Performance/Efficiency</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>7</td>
</tr>
<tr>
<td>Operability</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>5</td>
</tr>
<tr>
<td>Security</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>8</td>
</tr>
<tr>
<td>Compatibility</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>5</td>
</tr>
<tr>
<td>Maintainability</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>12</td>
</tr>
<tr>
<td>Transferability/Portability</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>9</td>
</tr>
<tr>
<td>Supportability</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>Correctness</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>Integrity</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>Testability</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>Flexibility</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>Reusability</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>2</td>
</tr>
<tr>
<td>Interoperability</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>Human engineering</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>Modifiability</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>Understand-ability</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>Feature quality</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>Conformance</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>Aesthetics</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>Perception</td>
<td>x</td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>Effectiveness</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>Efficiency, safety/satisfaction</td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>0</td>
</tr>
<tr>
<td>Sustainability</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>5</td>
</tr>
<tr>
<td>PROCESS Maturity</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td>4</td>
</tr>
</tbody>
</table>

As shown in Table 1, most current models have common features, whilst some focus on a specific software property, and some are highly extensive, and the quality characteristics are unclear and difficult to measure. These common features are reliability, functionality, efficiency, usability, portability, security, maintainability and which acquired high scores of 13, 13, 8, 9, 13, 8 and 10, respectively. These fundamental quality characteristics are measured for software assessment. Therefore, the model exhibits these fundamental quality characteristics. Some quality characteristics demonstrate middle-level ones, such as operability, compatibility, sustainability and process maturity, which acquired scores of 5, 5, 6 and 5, respectively, and some of these quality models unsatisfied the standard score.

Many standards and models, such as Dreamy, Boehm, McCall and ISO/IEC 9126 standard, which is developed into ISO/IEC 25010, produced to improve the software evaluation quality have been mentioned. These models exhibit several common quality characteristics, such as reliability, efficiency, maintainability and portability [20]. Therefore, our study will adopt the ISO/IEC 25010.

According to [21], software quality models exhibit weaknesses. These weaknesses are found in many evaluators, which may complicate the quality of the program. Different residents may have different goals, and each suggests a specific quality model. In general, assessors choose their
characteristics, sub-characteristics and attributes, that are unnecessary from standard software models. All sub-characteristics are occasionally assessed as if they are equally significant to the program.

Measuring the quality of the OS software process is difficult. Most OSS measurement software is not effectively documented, thereby making it difficult to determine the quality of the OSS development process. The optimal way to evaluate OSS is to consider the software product used and software quality [21]. An easy way to evaluate the software is studying the quality in use. However, measuring the software quality is inappropriate. Considering the external and internal measurements according to the usual quality characteristics is necessary.

**Decision making (DM)**

Although some decisions are easy to make, other decisions are also difficult and cause loss of strength and energy. Similarly, making a good decision greatly varies depending on the information that the decision-maker has. In some conditions of the decision, considerable data may be needed, whilst in other cases, it may not require substantial information. Sometimes few data may be available; thus, the decision can be intuitively made.

Apart from the subjective and fundamental aspects of the decision-making process, structured ways also exist to assist decision-makers. The decision analysis aims to help decision-makers and create techniques but not replace them. An analysis of its decision can be identified as a modelling process, methodology, evaluation and choice of appropriate action for a particular decision problem. Decision analysis often involves a wide range of tools and a basic approach in which the decision-maker reduces the problem to manageable parts to make it easier. Decision makers must also obtain an intensive understanding of the problem, analyze the complex conditions and identify the procedure that corresponds to their knowledge and values through this process [22].

Making decisions in various aspects using the OSS-EMR choice process is often a difficult task. A simple decision-making is insufficient for this selection process because decision-makers often face complex, multi-criterion problems with conflicting objectives.

At this stage, MCDM is considered highly suitable for considering and comparing many alternatives and criteria. MCDM, a branch of operations research, is a common decision-making process that deals with conclusion problems under considerable decision criteria [23]. This method is a decision-making technique commonly used by decision makers to deal with multiple standard problems. This technique aims to consider the way the decision maker views a multi-standard problem. In this method, a mathematical model is created because the amount of information in a multi-attribute problem is excessive for humans. The appropriate thing to do is to allow the decision maker to focus on small parts of the problem. The way the decision maker views the multi-attribute problem is also defined as data for decision-makers [24].

MCDM methods are recommended solutions owing to their ability in dealing with technical problems in the real world and help decision makers to organize the problems that must be resolved as soon as possible, execute analysis and solution and arrange alternatives and comparisons in multiple platforms. Therefore, a suitable alternative described in previous studies was selected [25]. These methods are seemingly appropriate to resolve the problem of choosing the OSS-EMR package. The MCDM methods aim to choose an optimal alternative using decision matrix (DM), classify the viable alternatives amongst a set of available ones and rank the alternatives in a decreasing order based on their performance [25, 26].

Choosing an optimal software process from the software presented is an important aspect of IT system management. The selection process can be considered an MCDM problem, which can solve various and inconsistent characteristics amongst pre-determined decision alternatives [27]. Numerous decision-making theories have been successfully applied in various fields in the last few decades. Versatility and diversity of the MCDM program have assisted decision-makers. MADM/MCDM allows the application of multiple conflicting criteria. This part provides a critical valuation of the existing MCDM approaches and explains the manner by which to understand the
OSS-EMR selection. This work chooses new studies associated with decision-making and selection techniques [3, 28-31].

**Limitations of MCDM Techniques**

According to Jadhav and Sonar [3], Triantaphyllou and Lin [32], Zaidan, et al. [26] and Whaiduzzaman, et al. [33], the features, in addition to MCDM performances, can be summarized as follows: Weighted scoring method (WSM) is easy to understand and utilise. However, the method of assigning the criterion weights is arbitrary. Therefore, the task becomes hard to resolve the number of attributes increases. Another problem with WSM comes from using numerical scalar to get the final result.

Analytic hierarchy process (AHP) technique was used for software selection and assigning weight owing to its a flexible and effective tool for solve qualitative and quantitative multi-criteria problems. In addition, the method procedures can be applied in group and individual decision-making. However, the mathematical calculations are difficult due to substantial pairwise comparisons of criteria and alternatives, when the number of alternatives or standards changes, decision makers need to re-evaluate the alternatives. The alternative ranking depends on the alternatives considered for assessment. Therefore, adding or removing alternatives can change the final ranking (rank reversal problem)[25, 34].

The method of Eliminate and choose to express reality (ELECTRE) can apply qualitative and quantitative criteria. This method gives a premise to an entire request of various choices. The Vise Kriterijumska Optimizacija I Kompromisno Resenje (VIKOR) method compromise arrangement utilize linear normalisation [33, 35, 36]. Nevertheless, the values are based on not only the criterion assessment but also the overall function to balance the distance between the ideal solution and its inverse.

Additionally, TOPSIS is practically connected with the issues of separate alternatives. This technique is one of the pragmatic methods for explaining real word issues. The advantage of TOPSIS is that it can determine the best alternative immediately [27, 33, 36-38]. However, TOPSIS does not perform weight extraction and consistency checks on decisions. On this basis, TOPSIS satisfies the need for double comparisons, and the limit constraint may not significantly dominate the procedure. Consequently, this technique would be appropriate for cases with important alternatives and criteria, especially for quantitative or objective information.

In a fuzzy-based methodology [33, 35], Decision Maker can take advantage of linguistic terminology to evaluate alternatives to improve basic decision-making techniques by considering ambiguity in the human decision-making process. However, processing a fuzzy indicator and rating values for all alternatives is difficult. In addition, this method does not provide the advantage of checking the consistency of decisions. Table 3 provides decision making technical explain. According to the above-mentioned review, many MCDM methods and its issues were utilised. As indicated by Triantaphyllou in MCDM investigation, ‘there may not be a single MCDM method that can always ensure the best decision’ [39]. The issues of these techniques are outlined in either the ranking of alternatives or the calculation of weights. In this manner, we will utilise an integration technique between two methods to comprehend these problems.

Although AHP or Fuzzy can be used to set criterion weights, the latter is better than the former because the former is time-consuming and the number of mathematical calculations and comparisons increases with the increase of criteria and alternatives, thereby making the task difficult. In addition, decision-makers need to conduct a re-evaluation when the number of criteria or alternatives changes [5]. Furthermore, ELECTRE, TOPSIS and VIKOR systems can be utilised when ranking alternatives. In this study, TOPSIS is chosen to be combined with Fuzzy to solve the weight problems of TOPSIS. In addition, TOPSIS can immediately identify the highly suitable alternative, according to Whaiduzzaman, et al. [33]. Furthermore, this method is also utilised when ranking the software packages [40].
Existing Frameworks

Several frameworks were utilized for assessing and selecting software in the current study, each of which exhibits advantages and disadvantages.

The framework of Sanga [21] emphasizes on only three software features, namely, usability, maintainability and deployability. However, the study neglected functionality, security, transferability and reliability characteristics.

The framework of Jadhav and Sonar [3] reviewed the suitable software quality model in general software. Furthermore, the study used a hybrid knowledge-based system (HKBS) to specify the rank and weights of software. However, some metrics are unclear and difficult to measure because of limited IT users. In addition, the functionality criteria are unsuitable for the scope of the medical domain. Moreover, the HKBS process exhibits time-consuming calculation. The duration of training in this case is between 15 and 20 minutes and increases when the number of criteria and alternatives increase [41].

The EFFORT framework is the project evaluation by which the EFFORT frame is created to assess the quality and functionality of OSS systems. The term EFFORT is an abbreviated form of Evaluation Framework for free/OS projects. This framework helps in the assessment of community trustworthiness and product attractiveness and quality [42]. The strength of EFFORT model depends on its ability to consider the key quality aspects of OSS, which include the program itself, the community created around the program and its attractiveness to the user [12]. However, the framework did not comprise operability and security. In addition, the weights of the attribute were arbitrarily assigned. Considerable criteria make it difficult to assign weight [41].

The framework of Zaidan, et al. [41] proposes evaluation and selection framework for medical software. The study focuses on six criteria, namely, security, functionality, usability, user support, customisation and developer support. Moreover, this study aims to evaluate medical software packages for customisation using TOPSIS and AHP to choose and rank the alternatives. However, this work did not consider the reliability, portability, maintainability and vendor support criteria. Furthermore, AHP was used to set the criterion weight. However, this method is time-consuming because the number of pairwise comparisons and mathematical calculations increased with the increased number of alternatives and criteria. In addition, the decision-makers need to re-evaluate the alternatives when the number of alternatives or criteria changes [5].

The current framework requires improvement in terms of the evaluation criteria, which are associated with EMR and the techniques used to set the criterion weights, in addition to those used for ranking the alternatives, according to the preview review.

Conclusion

Various aspects related to the OSS-EMR evaluation and selection were explored and studied. In this work, comprehensive insights into the evaluation and selection of OSS-EMR programs were presented in two ways: Firstly, the quality characteristics were determined to evaluate OSS-EMR packages from published papers according to the current quality models and frameworks. Secondly, the capabilities of selection methods that are appropriate for solving the issues of OSS-EMR packages were presented on the basis of the multi-criteria evaluation and selection problem. In addition, the gaps on the evaluation criteria used for EMR and other problems were discussed. Moreover, MADM/MCDM techniques used to evaluate and select OSS in health informatics were specified as part of the recommended solutions. This study direction is significant because it will help administrators and decision-makers in the field of medical informatics to select the highly suitable and appropriate OSS-EMR for their needs.

Summary Points

What was already known

- Investigated 15 software quality models of open-source with its limitations.
• Investigated 4 frameworks for evaluation and selection of open-source with its limitations.
• Investigated MCDM techniques with its limitations.

Acknowledgment
This work was supported by the National Nature Science Foundation of China under Grant Z201G10110G20003

References


