Bulletin of Electrical Engineering and Informatics
Vol. 12, No. 3, June 2023, pp. 1711~1727
ISSN: 2302-9285, DOI: 10.11591/eei.v12i3.4945

Framework for selecting the best software quality model for a smart health application based on intelligent approach

Ashraf Mousa Saleh[1], Odai Enaizan[2]
[1]Department of Software Engineering, Faculty of Computer Science and Informatics, Amman Arab University, Amman, Jordan
[2]Department of Management Information System, Haql University College, University of Tabuk, Tabuk, Saudi Arabia

Article Info

Article history:

Received Nov 25, 2022
Revised Dec 5, 2022
Accepted Dec 16, 2022

Keywords:

Fuzzy delphi
MCDM
Quality factors
Quality models
Software quality

ABSTRACT

There is difficulty in knowing how to weigh the factors of software quality models so that decision-making can be eased. Furthermore, previous work was limited to undertake evaluation and selection of appropriate software quality model based upon multi-criteria in the context of smart health applications. This paper aims to evaluate and select an appropriate model of software quality based on multi-criteria decision-making (MCDM) by three phases of framework. Firstly, investigation of software quality models and factors that were identified based on 'fuzzy delphi'. Secondly, identification of quality models that have uniform multi-criteria so that a decision matrix could be established. Uniform multi-criteria were used in the decision matrix as the basis of the models of quality and the multi-criteria. Subsequently, MCDM approach is adopted and the bases used in the employment of the MCDM approach for the eva luation and selection of the software quality model were technique for order preference by similarity to ideal solution (TOPSIS) and fuzzy analytical hierarchy process (FAHP). The results demonstrated that seven quality factors could be considered as the key factors based upon fuzzy delphi, i.e., usability, maintainability, reliability, interoperability, portability, modifiability, and efficiency. Also, reults shows that McCall is the most appropriate model.

Corresponding Author:

Ahsraf Mousa Saleh
Department of Software Engineering, Faculty of Computer Science and Informatics
Amman Arab University
Amman, Jordan
Email: asaleh@aau.edu.jo

1. INTRODUCTION

Nowadays, applications of smart technology have greater prevalence within mobile health care [1], [2]. Certain smart technologies have been dedicated to particular situations or groups. Most smart technology applications for mobile health care were intended for improving the management of disease, such as monitoring of the status of health/disease, adjustment of dosage of medication and increase of adherence to prescribed medications, and to the changing of health-related behaviours [1], [3]. The term 'smart healthcare' can be considered as referring to a system of health service that utilises technologies such as IoT, wearable devices and the mobile internet for dynamic accessing of information, the connection of institutions, materials of people in relation to healthcare and the active management and response to the needs of medical ecosystems within an intelligent kind of manner.

Software and applications of smart healthcare is able to promote greater interaction between all of the parties within the field of healthcare. It can ensure that participants are able to receive the required services, help the parties in question to make more informed decisions, and can facilitate more rational resource allocation. Smart healthcare software, then, can be considered a construction of information within the medical field that is at a higher stage. There are many benefits to health care software; indeed the assessment of quality with regard to the service of an application of health care is of importance [4]. A failure to assess quality in the services of health care software can lead to discontented users, decreased efficiency and a lower level of usefulness for the software, as well as increased costs due to the likeliness of errors and faults becoming apparent [5]. Insufficient and incorrect information can result in significant damage to the patients and the users of the medical services [6], as such, software quality and the need for the relevant international and national standards has become a focus for lots of different policy makers and researchers [7].

The quality of the software is key to the overall success of a software system and a crucial consideration for project managers, users and developers alike. A central aspect to these matters could be cultural considerations that determine the way success is perceived within the minds of those undertaking evaluation of software quality. Indeed, differences in perspective can main a mismatch between criteria for the success of software amongst the various associated stakeholder groups. The factor that is the highest determining one with regard to achievement is the quality and functionality for the outcome of the software and success in terms of external goals, including user satisfaction. The production of high quality software is essential so that the expectations of customers can be met [8]. The definition for quality of software, as provided within American National Standard Institute (ANSI)/American society for quality control (ASQC) standard (1978), is that it is a totality of characteristics and features of a service or product that have a bearing upon the ability it has to satisfy the needs given. Various models for quality evaluation, such as the model of ISO/IEC9126, have introduced specific definitions for the characteristics and features of software in terms of quality attributes [9]. Attributes of quality are sometimes known as non-functional types of requirement that ought to be given a high degree of consideration throughout the process of software development. However, the achievement of the non-functional types of requirement has a close relationship to software architecture design [10]. As such, software developers need to be encouraged to evaluate quality attribute achievement early within the phase of architectural design since that is more cost effective and convenient than evaluation once the software has been implemented.

An attribute of quality, however, cannot be considered in isolation. There is a need to consider various relationships to other attributes of quality required within software under development; the reason for this is that enhancement of one attribute of quality always leads to hindrance or enhancement of other attributes of quality. So, various models of software quality have been developed with the same goal of achieving evaluation of software quality [11]. However, the definitions for factors of quality of software are inconsistent and there are contradictions between the models that have been developed. A number of models of quality have been put forward for testing software quality with the most popular (as used for this paper) being the model of Jim McCall, the ISO 9126 model and the W. Boehm model.

Recent research has noted that a lack of a comprehensive model for quality stems from neglect of all the various aspects to quality [12]. Even standard models of quality are not sufficiently comprehensive for use in various engineering tasks [13]. Final product functionalities are not considered by McCall. Within functionality, usability, reliability, performance, and supportability (FURPS) final product quality is not affected by portability. No measurement approaches related to factors of quality are not present within a number of models, including Boehm and Dromey. Reusability is not considered by [14]. No solution was provided by [15] for coping with subjectivity in factors of quality and the dependence of them upon the experience and knowledge of experts [13], nor for the various entities at different knowledge levels [16].

The concept of software quality is, indeed, a complex one that is impacted upon by various aspects of the process of software development. Different engineering tasks are involved in software quality such as the domain of application and testing, and the quality of code, design, process and product [13]. Furthermore, various actors have involvement with the concept of quality based upon the roles they play, such as architect, designer or analyser, at various times, such as the phases of establishing requirements, the analysis and the design, and with a variety of artefacts, including code, documents and models. A comprehensive model is needed that addresses all of the aspects involved from software quality estimation to its recognition, representation and assessment [11]. As such, this paper has the aim of evaluating and selecting an appropriate model of software quality in accordance with the 17 factors of quality. Integrating the methods of technique for order preference by similarity to ideal solution (TOPSIS), fuzzy analytical hierarchy process (FAHP) and fuzzy delphi, there is evaluation and selection of appropriate models of quality to accord with factors of quality within an application of smart health care context.

2. MODELS OF SOFTWARE QUALITY

Lots of models of quality have put forward links between external factors/attributes of quality and internal properties of software. It was mentioned by [8] that well-known models of software quality include

ISO 9126, FURPS, Dromey, Boem and McCall. A model of quality was put forward by [17] with hierarchies of measures, criteria and factors, and eleven external factors of quality proposed, namely: flexibility, maintainability, portability, testability, integrity, usability, efficiency, reliability, correctness (functionality), interoperability, and reusability. A model of quality was proposed by [18] that consisted of 3 characteristic levels: the high-level, the intermediate-level and the primitive-level. The model introduced 7 external factors of quality for 3 characteristics at the high-level, i.e., maintainability (modifiability, understand ability and testability), as-is utility (usability, efficiency and reliability) and portability. Each external attribute is mapped to lots of measures (primitive sub characteristics).

Another model that is similar to the Bohem and McCall models is the FURPS/FURPS+ model [18]. The FURPS+ version has been extended with rational software [19]. Another model that is similar to the quality models of Boehm and McCall and FURPS+ is the Dromey quality model [20]. The Dromey model has a focus upon relationships between attributes and sub-attributesand provides mapping of the software properties towards the attributes of software quality. The standard model of ISO (1991) defines quality in terms of it being a set of characteristics of the product at 2 levels, i.e., the internal level and external level. Six attributes of quality are included within the ISO model: reliability, functionality, maintainability, usability, portability, and efficiency. Those attributes of quality are refined to more software product sub-attributes. An ISO 9126 model variant is squid [21], however it depends upon there being a database of past-experience so that the relationship between external and internal quality characteristics can be established. A comparison of models of software quality factors from within the literature is shown in Table 1.

Table 1. Comparison of quality models

Factor/attribute	McCall	Boehm	Dromey	FURPS	ISO
Maintainability	x		x		x
Flexibility	x	x			
Testability	x				
Correctness	x				
Efficiency	x	x	x		x
Reliability	x	x	x	x	x
Integrity	x			x	
Usability	x		x		x
Portability	x	x	x		x
Reusability	x		x		
Interoperability	x				
Human engineering		x			
understand ability		x			
Modifiability		x			
Functionality			x	x	x
Performance				x	
Supportability				x	

3. PREVIOUS RESEARCH

A framework was proposed by [22] that used a framework with the method of TOPSIS so that models could be ranked in accordance with efficiency, maintainability and reliability so that the best model of software quality could be selected based upon those three parameters. This study, however, has a focus upon just three software quality factors based solely upon the TOPSIS method. Remaining software factors that belonged to aforementioned models of quality were neglected by the framework. FAHP was applied by [23] as a software quality model selection method. The fuzzy prioritisation method is used by the study for the inspection of the FAHP method application for the selection of best model based upon the requirements. The method proposed by [24] for choosing best model of software quality was under an environment of IFSs (intuitionistic fuzzy sets). The study undertook a comparative study along with previously developed approaches for the verification of the proposed method results. An approach was put forward by [25] for performing prioritisation for attributes of system quality that involved application of the process of analytic hierarchy, which is able to prioritise attributes of quality based upon the relative importance of them.

Based upon the survey of literature, it can be seen that a number of studies have applied hybrid fuzzy delphi, TOPSIS and FAHP in the evaluation and selection of appropriate model of software quality based upon comprehensive factors of software quality for well-known models of software quality. Quality of software is an issue that is of a paramount importance for all software stakeholders within a particular establishment and, because of customer demand, there is a rapid increase in overall demand [26]. In recent decades, there has been an exponential increase in importance in using quality software [27]. Software is seen as being a tool by software users that enables them to undertake their daily activities more easily and employ it in performing tasks that are sensitive [28].

Use of software of lower quality can, indirectly and/or directly, endanger lives as well as result in huge losses for the software users [29]. So, many models of software quality have been put forward for the evaluation of software quality; however, as yet, no one particular model has been accepted widely as being a benchmark for the assessment of software quality. The recent for that is that those models fail to address all of the important attributes of software quality that stakeholders would be keenly interested in and are, rather, tailored for the meeting of the requirements of a specific project. So that the requirements of the stakeholder(s) can be addressed, it has been proposed that customised models of software quality are used [30]. Those custom-built models of quality offer different types of benefits for the research community and software industry and, hence, fail to cover a broad scope of attributes of quality. It has been shown by those researchers how there is gaining importance for the employment of models of software quality within the software development field. Nonetheless, there has been ranking of the attributes of quality and so there is difficulty in knowing weightings for all of the attributes within the process of decision making. Multi-criteria decision-making (MCDM) along with fuzzy delphi is employed within this research for the evaluation and selection of the attributes of software quality and the ranking of them.

4. METHOD

This study has the primary aim of investigating the way in which the most appropriate model of software quality can be evaluated and selected for the development of smart health applications. There will be identification of appropriate models based upon multi-criteria through use of MCDM. The method presented is based upon multi-criteria (derived from experts in software quality assurance) as a frame of reference. Firstly, the models of software quality that were reviewed in our study included ISO, FURPS, Dromey, Boehm, and McCall which had been identified from in previous research. The models of software quality identified with the factors of quality (multi-criteria) are to be the primary components for the decision matrix to be developed. A crossover is represented in the context between factors of quality (multi-criteria) as the criteria in question and the alternatives of software quality models. Following on, the basis for the technique of MCDM is integration of FAHP and TOPSIS for evaluation and selection of an appropriate model for software quality. Figure 1 depicts the new framework for evaluation and selection.

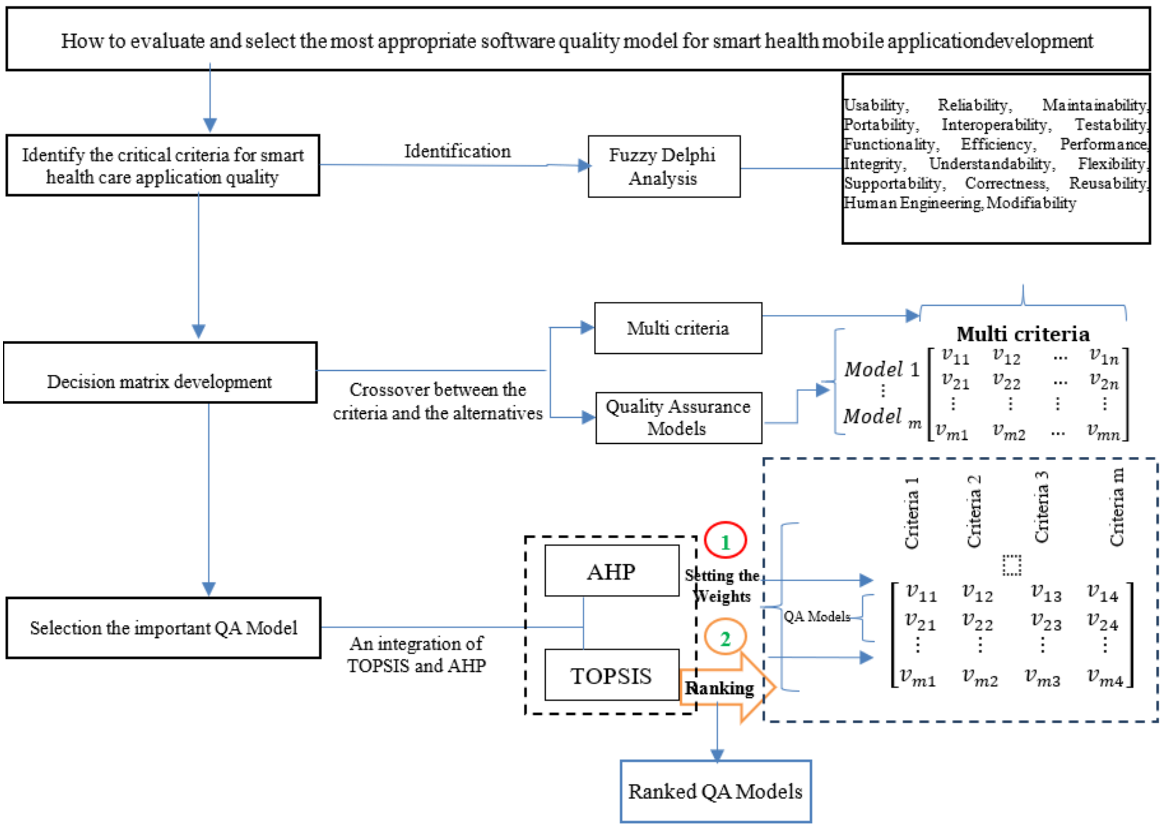

Figure 1. A new evaluation and selection framework

Within the initial section, previous studies are reviewed that investigated models of software quality. A number of issues, however, limited our study scope. Furthermore, there is only application of this research to the aforementioned models of software quality. So, here there are 2 steps: firstly, identification of alternatives from software quality models identified within previous studies, i.e., ISO, FURPS, Dromey, Boehm and McCall and secondly, introduction of the factors of quality that fuzzy delphi identified. There are 5 models of software quality that are integrated within this study with factors of quality within one single decision matrix. The next step, then, involved defining multi-criteria for the models of quality from ISO, FURPS, Dromey, Boehm, and McCall.

The development of a decision matrix is based upon crossover between the quality factors (multi-criteria) and the models of quality (Belinda *et al*. n.d.). As a consequence, lots of models of software quality have been put forward for evaluation of software quality; however, as yet, there is no wide acceptance of a particular model that could serve as a benchmark in the assessment of software quality. The reason is that the existing models fail to address all of the key attributes of software quality that could be of interest for the relevant stakeholders. In practice models are tailored specific to the requirements of a particular project. In order to address the requirements of stakeholders, custom models of software quality have been put forward that offer different kinds of benefit to the research community and the software industry. However, their coverage is not a broad scope for all attributes of quality.

The ranking of the attributes of software quality can greatly assist developers in the selection of the best quality of attribute in the evaluation of developed software. Previous studies have not ranked attributes of quality, and this has led to there being several different customised models of software quality being put forward. It has been shown by the researchers that a great deal of importance is now given to use of models of software quality within software development. Nonetheless, quality attributes were not ranked and so there was difficulty in knowing the weight that each attribute had within the process of decision making. This study utilises an analytic hierarchy process, which is a multi-criteria type of decision-making tool for evaluating the attributes of software quality and ranking them. There has not been ranking of most of the attributes of software quality that have been used for assurance of software quality so there is difficulty in noting key attributes to employ in evaluating software products.

In the following phase, a new methodology for decision making is developed that is based upon the issues that were identified within the previous phases. MCDM was defined by [31] and [32] as 'an extension of decision theory that covers any decision with multiple objectives. As such, MCDM is used as a methodology for the assessment of alternatives from individual and frequently conflicting criteria which are combined to conduct an appraisal overall. Furthermore, MCDM was defined by [33] as "an umbrella term to describe a collection of formal approaches, which seek to take explicit account of multiple criteria in helping individuals or groups explore decisions that matter".

Multi-criteria forms of analysis, which can be considered an operational research sub-discipline which gives explicit consideration to numerous criteria within decision-making environments, happens within various actual medical record situations [33]. There are several different types of useful technique that may be applied within actual issues for MCDM. Those techniques assist decision makers (DMs) in their organisation of outstanding problems and help them in the provision of prioritisation, scoring and the analysis of the alternatives [34]. As such, within this current study, there is performance of scoring from the suitable alternatives. There is reviewing of several methods of MCDM. The methods of MDCM that are most popular use a number of concepts to accord with the work of [35], [36]. However, it is our understanding that none of those methods was utilised in the evaluation and selection of models of software quality based upon MCDM.

The MCDM/MADM methods may also be used for solving scoring problems in the selection of models based upon multi-criteria within applications of smart mobile health. In any ranking of MCDM/MADM, there ought to be defining of the fundamental terms with inclusion of an evaluation matrix (EM) or decision matrix, with criteria and alternatives [35]. EM comprising alternatives (m) and criteria (n) ought to be created. Given intersection of each of the alternatives and the criteria (xij), the matrix is obtained (xij) m×n.

$$DM/EM = \begin{array}{c} \\ A_1 \\ A_2 \\ \vdots \\ A_m \end{array} \begin{array}{c} \begin{array}{ccc} C_1 & C_2 & \dots & C_n \end{array} \\ \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} \end{array} \quad (1)$$

Where in: C_1, C_2, to C_n represent criteria against which there is measurement of each alternative performance, i.e., the factors of quality; x_ij is representative of value for alternative A_i in respect to criterion C_j; and C_j possesses a weight W_j with A1, A2 to Am representing possible alternatives as rated by the DMs, i.e. ISO, FURPS, Dromey, Boehm and McCall.

Evaluation of appropriate models for quality within the application for smart health is undertaken through use of the criteria for factors of quality involving reliability, usability, portability, maintainability, efficiency, modifiability and interoperability. Each model of quality, however, has factors of quality and each type of DM has different weightings for their particular factors [37]. So, the selection of suitable factors based upon multiple perspectives can be problematic.

The evaluation of models of software quality that are based upon numerous attributes can be considered a complex and essential MCDM problem. However, the selection of models, i.e., ISO, FURPS, Dromey, Boehm, and McCall, is a problem that is multi-criteria, with each of the models deemed to be an alternative that is available for DM. Taken from that perspective, it is suitable to use TOPSIS for cases involving numerous alternatives and attributes [38], [39]. Specifically, the application of TOPSIS has convenience when there is provision of quantitative or objective data. There is utilisation of the method of TOPSIS to rank the models of quality based upon multi-criteria analysis. A significant shortcoming with TOPSIS, however, is lack of provision regarding elicitation of weight and checking for judgement consistency. So, an effective technique is required by TOPSIS in order to acquire relative importance amongst various criteria in regard to an objective; such a technique is provided by analytical hierarchy process (AHP). So, there is adoption of AHP for calculation of attribute weight. The one that is most suitable amongst the methods of MCDM/MADM recommended is employed in ranking amongst existing alternatives. Integration of TOPSIS (identified method of MCDM/MDM) and AHP are employed as the basis for application of the algorithm proposed for settling the complexity involved with issues of multi-attribute selection accompanying a variety of medical records [32]. Figure 2 shown illustrates steps for integrated fuzzy delphi with TOPSIS and FAHP. Within the sections that follow there is illustration of the steps utilised with the TOPSIS and AHP methods.

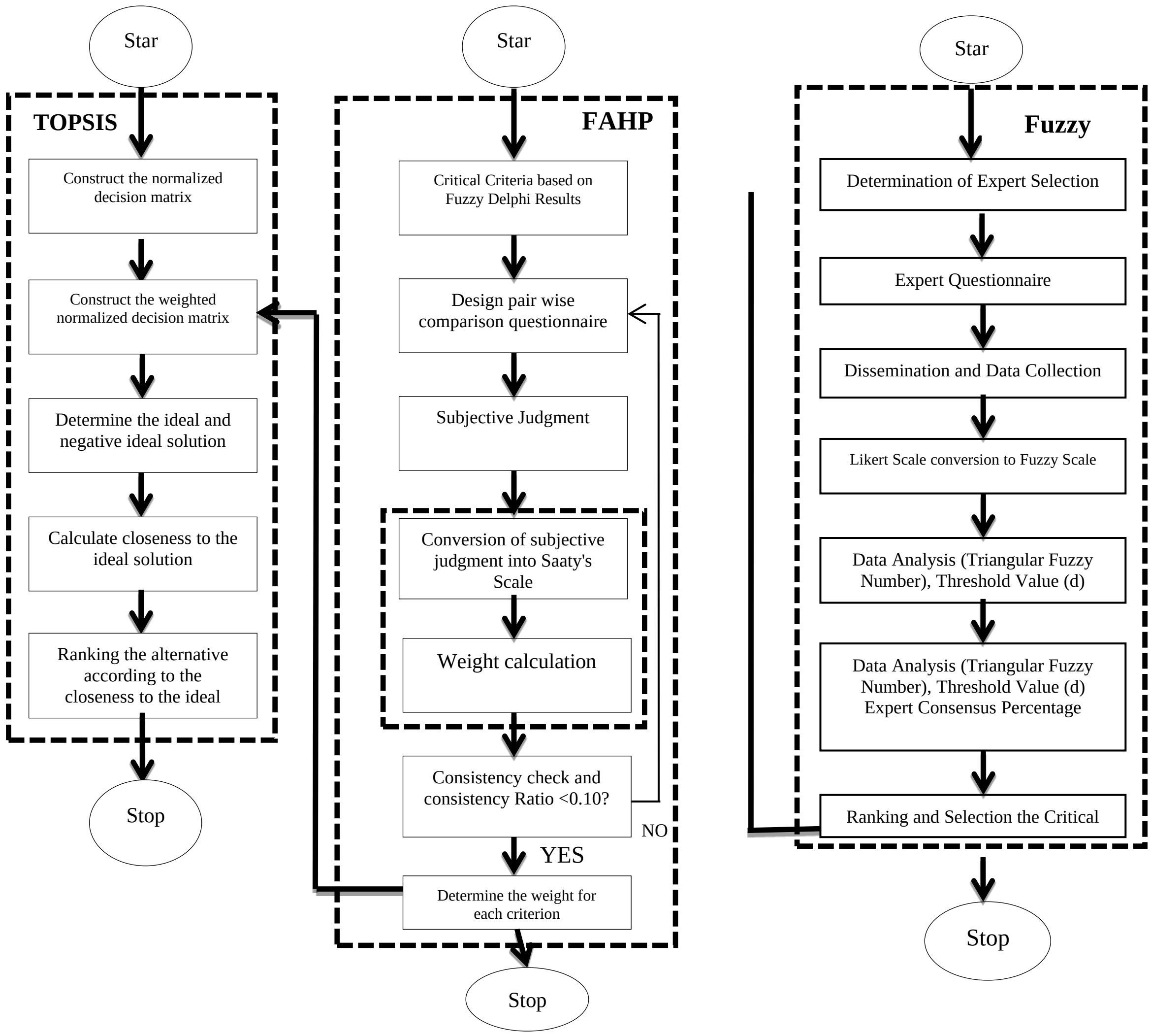

Figure 2. Integrated fuzzy delphi, FAHP, and TOPSIS

4.1. Fuzzy measurement

4.1.1. Data analysis using the fuzzy delphi method

A delphi-fuzzy type of method was proposed by Ishikawa, wherein there was application of the integration of the method of delphi with a technique that was fuzzy. The key aim with the delphi method was the achievement of a consensus that is based upon the views or votes given by experts for the topic of discussion. The use of proper sequential questionnaires is able to reduce the amount of redundant responses from the panel of experts or responders [40]. That technique was employed in achieving 'iteration', 'anonymity', 'statistical group response' and 'controlled feedback'. Within this, a panel of five to nine experts is supposedly able to reach consensus. The method operates extremely well, with a high level of accuracy, for scenarios of multi-participants, multi-principle and with 1 to many different objectives. The method calls for lots of repitition whilst the opinions are taken from an expert panel. That process is repeated till a final consensus is reached.

The fuzziness problem can be solved by integration of the fuzzy system within the delphi method. That integration involves use of semantic variables to solve the issue of fuzziness. The principal for working through that method is as follows. Firstly, a group of experts is provided with the questionnaire and requested to provide their responses on the topic. Then, those responses from the experts are used in preparing a second questionnaire set [41]. This second questionnaire set is then provided to the same group or a new group of experts and there is then collection of the new responses. Those responses are then used in preparing a third questionnaire and so on with the process repeated until convergence of the output to an optimum solution. The utilisation of fuzzy sets has greater consistency in solving the issue of the questionnaire and minimising error levels with high accuracy with fuzzy numbers. There is illustration of the fuzzy delphi method process within the sections that follow [42].

a. Fuzzy delphi method step no.1

With the assumption that invites have been extended to K experts to determine importance of criteria of evaluation and the rating for alternatives with regard to a number of criteria with the use of linguistic variables. Within this step, expert feedback with regard to the importance of items is represented by the linguistic options. For instance, it is expected that the expert selects the option that best reflects her/his opinion with regard to software quality factor, ranging from 'not at all important' to 'very important'.

b. Fuzzy delphi method step no.2

Linguistic variables are converted into triangular-fuzzy numbers. A 5-point Likert scale was chosen for this study to represent the feedback of the experts as can be seen within Table 2. Based upon Table 2, the values that the experts presented were converted to triple fuzzy values.

Table 2. Linguistic variables of the agreement

Linguistic variables	Fuzzy scale		
Very important	0.6	0.8	1
Important	0.4	0.6	0.8
Neutral	0.2	0.4	0.6
Not important	0	0.2	0.4
Not important at all	0	0	0.2

c. Fuzzy delphi method step no.3

For each of the experts, the vertex method is used for computing distance been average of (r_ij)̃ and (r_ij^k)̃ plus distance between average for (w_j)̃ and (w_j^k)̃, wherein k equals1, k. According to Chen [43], computation of distance between 2 fuzzy numbers, n ̃ equating to n1, n2, n3 and m ̃ equating to m_1, m_2, m_3, is done through in (2):

$$d(\tilde{m},\tilde{n}) = \sqrt{\frac{1}{3}[(m_1 - n_1)^2 + (m_2 - n_2)^2 + (m_3 - n_3)^2]} \quad (2)$$

d. Fuzzy delphi method step no.4

According to Cheng and Lin [44] noted that the evaluation of the expert and the average distance are equal or lower than the threshold of 0.2. Then that shows that there has been achievement of consensus. In addition, if between criteria weight (n) and the score of alternatives (m×n) the percentage for consensus is greater than 75% then go to step no.5. Otherwise, the steps are to be repeated [45].

e. Fuzzy delphi method step no.5

Fuzzy evaluations are aggregated in (3). where ~rij denote fuzzy rating of i alternative at j criteria, w~j denote fuzzy weighting of j criteria. Aggregate the fuzzy evaluations by $A_i=[x_{ij}].[wj]$, i=1,…m, j=1,…n,i.e.

$$\tilde{A} = \begin{bmatrix} \widetilde{A_1} \\ \vdots \\ \widetilde{A_m} \end{bmatrix} \text{where} \widetilde{A_\iota} = \widetilde{r_{\iota 1}} \otimes \widetilde{W_1} \oplus \widetilde{r_{\iota 2}} \otimes \widetilde{W_2} \oplus \ldots \oplus \widetilde{r_{\iota n}} \otimes \widetilde{W_n} \text{ I} = 1, \ldots, m \quad (3)$$

f. Fuzzy delphi method step no.6

For each of the alternative options, there was defuzzification of the fuzzy evaluation ((A_i)˜ equating to a_i1, a_i2 and a_i3) through.

$$a_i = \frac{1}{4}\,(a_{i1} + 2a_{i2} + a_{i3}) \quad (4)$$

There can be determination of ranking order for the alternative options in accordance with the a_i values.

4.2. Measurement of weight through use of FAHP

The popular method of AHP can be used for setting weights within MCDM [46]. That method is based upon paired comparisons for the production of ratio scales. There is measurement of the ratio scales through main eigenvectors, while there is use of eigen value for the calculation of the index of consistency. Each perspective is assigned a weight when AHP is used. There is a rating of each factor of quality for each of the models of quality that are put forward for evaluation. Following that, there is utilisation of AHP for derivation of ratio scales from the pairwise comparisons. Three participants were selected for completion of the AHP who were engaged within a team for software quality assurance and that have more than 5 years experience.

Three copies of the pairwise comparisons, thus totalling 6 comparisons among all of the perspectives, were presented to participants and their perspectives on them gathered from their responses. There was creation of a relative scale of 1 to 9 in order to measure differences within participant preferences regarding the perspectives. Each of the software quality professionals critically analysed those perspectives based upon their experience and knowledge. Subsequently there was creation of a reciprocal matrix based upon the pairwise comparisons. Finally, there was computation of the eigenvector so that relative ranking for the perspectives could be provided. It was requested of the 3 evaluators that they complete comparisons for the 7 criteria. As a result, AHP can be considered a most useful form of process of decision-making involving multi-criteria. Whilst AHP has considerable popularity, it is not able to deal with the imprecision and uncertainty that are associated with the perception of decision makers. So those pitfalls can be overcome, fuzzy is combined with AHP since fuzzy set theory has the capability for representation of fuzzy and vague values. FAHP that is developed on from AHP has been used widely in the solving of complicated problems in decision making [47]–[49].

4.3. Identification of rank through use of TOPSIS

Lastly, it is recommended that the TOPSIS method is used since it is adopted extensively for factor ranking and several alternatives can be ranked with selection of the appropriate one [50]. As can be seen in Figure 3, there is ranking of available alternative scores into descending order, while there is prioritisation of the factors that are most urgent based upon TOPSIS. Aggregate scores merely offer an idea of which models of quality have greater urgency than others. The quality models are the alternatives for which scores are calculated by TOPSIS. Following that, there is selection of the best alternative. The technique gives an indication that the option that is appropriate offers a shortest distance geometrically to a positive ideal solution and the longest distance geometrically to a negative ideal solution. The steps involved in the process are illustrated within the following sections.

4.3.1. Construction of the normalised DM

The initial step attempts the transformation of various attribute dimensions to non-dimensional attributes that allow comparison between attributes. There is then normalisation of $(\mathrm{x_{ij}})_{\mathrm{m*n}}$(the matrix) from the $(\mathrm{x_{ij}})_{\mathrm{m*n}}$ into matrix R equating to $(\mathrm{r_{ij}})_{\mathrm{m*n}}$ through adoption of a normalisation method as (5):

$$r_{ij} = x_{ij} \Big/ \sqrt{\textstyle\sum_{i=1}^{m} x_{ij}^2} \text{ Error! No text of specified style in document.} \quad (5)$$

R (the new matrix) results as (6):

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix} \quad (6)$$

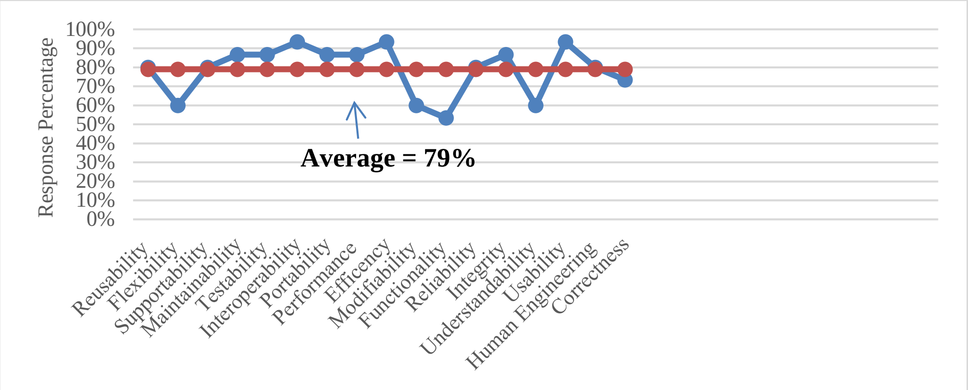

Figure 3. Fuzzy delphi consensus average

4.3.2. Construction of the normalised weighted DM

Introduced into the normalised DM is a set of weightings (w equating to $w = w_1$ and w_2 and w_3 and w_j, andw_n) that has been calculated from use of the AHP method. So that the weighted matrix can be constructed, there ought to be multiplication of each of the columns from R (the normalised DM) with w_j (the associated weight). The calculation then of the resulting matrix can be done through multiplication of each of the columns from R (the normalised DM) with w_j (its associated weight) and the resulting new matrix (V) is as (7):

$$\sum_{j=1}^{m} w_j = 1 \quad (7)$$

That process results in the new form of matrix, matrix V:

$$V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ v_{m1} & v_{m2} & \cdots & v_{mn} \end{bmatrix} = \begin{bmatrix} w_1 r_{11} & w_2 r_{12} & \cdots & w_n r_{1n} \\ w_1 r_{21} & w_2 r_{22} & \cdots & w_n r_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ w_1 r_{m1} & w_2 r_{m2} & \cdots & w_n r_{mn} \end{bmatrix} \quad (8)$$

4.3.3. Determination of the ideal and negative ideal solutions

Ideal alternative (A^*) and negative ideal alternative (A^-) may be calculated using the following equations.

$$A^* = \left\{\left(\left(\max_i v_{ij} \middle| j \in J\right), \left(\min_i v_{ij} \middle| j \in J^-\right) \middle| i = 1,2,\ldots,m\right)\right\} = \{v_1^*, v_2^*, \ldots, v_j^*, \cdots v_n^*\} \quad (9)$$

$$A^- = \left\{\left(\left(\min_i v_{ij} \middle| j \in J\right), \left(\max_i v_{ij} \middle| j \in J^-\right) \middle| i = 1,2,\ldots,m\right)\right\} = \{v_1^-, v_2^-, \ldots, v_j^-, \cdots v_n^-\} \quad (10)$$

J refers to the {i equating to 1 and 2..m} subset which presents perspectives (for our study of modifiability, usability and others) whilst the complement set for J is J^- and may be notated as J^c.

4.3.4. Calculation of the separation measurement based upon Euclidean distance

There is application of separation measurement through determination of distance between A* (ideal vector) and each of the alternatives within V through utilisation of Euclidean distance using as (11):

$$S_{i^*} = \sqrt{\sum_{j=1}^{n}\left(v_{ij} - v_j^*\right)^2}, i = (1,2,\cdots m) \quad (11)$$

Likewise, separation measurements for each of the alternatives within V from A^- (negative ideal) is calculated through as (12):

$$S_{i^-} = \sqrt{\sum_{j=1}^{n}\left(v_{ij} - v_j^-\right)^2}, i = (1,2,m) \quad (12)$$

From this step, the outcomes are S_{i^-} and S_{i^*} for each of the alternatives. Those values are in reference to distance between each of the alternatives and the 2 vectors which are negative ideal and ideal.

4.3.5. Calculation of closeness to ideal solution

Closeness of each of the alternatives (A_i) to the ideal solution (A^*) can be computed using (13):

$$C_{i^*} = S_{i^-}/(S_{i^-} + S_{i^*}), 0 < C_{i^*} < 1 , i = (1, 2, \cdots m)$$

Evidently, C_{i^*} equates to 1 when and only when A_i equates to A^*. Like wise, C_{i^*} equates to 0 when and only when A_i equates to A^-.

4.3.6. Ranking of the alternatives

Lastly, there can be ranking of the alternatives based upon their value with advanced stage within the ranking going to the alternative that has highest value and vice versa. Proper citation of other works should be made to avoid plagiarism. When referring to a reference item, please use the reference number as in [16] or [17] for multiple references. The use of ”ref [18]” should be employed for any reference citation at the beginning of sentence. For any reference with more than 3 or more authors, only the first author is to be written followed by et al. (e.g. in [19]). Examples of reference items of different categories shown in the References section. Each item in the references section should be typed using 8 pt font size [20]–[25].

5. RESULTS

There are two sections to the results with the first relating to data analysis using fuzzy delphi. Fuzzy method was used to analyse the critical software quality factor needed within Information Technology (IT) companies. Furthermore, the MCDM results as follows:

- The data analysis using fuzzy delphi

The fuzzy-delphi method was proposed for this study in order to analyse the critical software quality factor that the IT companies. Initially, the opinions of experts were collected and then there was utilisation of the fuzzy method in order to achieve a consensus amongst the opinions and to move from bias so that an optimal decision could be reached regarding quality factor [51]. Expert consensus is sought by the fuzzy delphi method through use of the format of a questionnaire that uses a likert scale. As Table 3 shows, there was conversion of the linguistic variables into fuzzy scale. There was conversion of the scores to fuzzy numbers that were derived through use of a mathematical formula for fuzzy delphi.

Table 3. Linguistic variables and the associated fuzzy scale

Linguistic variables	Fuzzy scale			Likert scale
Very important	0.6	0.8	1	5
Important	0.4	0.6	0.8	4
Neutral	0.2	0.4	0.6	3
Not important	0	0.2	0.4	2
Not important at all	0	0	0.2	1

The consensus of the response for the survey overall and for each of the parameters of quality is shown within Figure 3. As the figure shows, 71% was overall average for group consensus; as such that iteration was accepted. If consensus of the average group has a value over 75%, then it is considered that consensus was achieved. On the other hand, if that is lower than 75% then another survey iteration is needed as there had not been achievement of consensus [51], [52]. Following that there was calculation of the value of threshold (d) equating to the distance between the view of the expert and average fuzzy view. The basis for the decisions is the d value; if d is lower than the 0.2 threshold then there is acceptance of that parameter as there was achievement of expert consensus, however if the value of d is over 0.2 then there is rejection of the parameter [52]. All of the seventeen parameters (factors of quality) are shown within Figure 4.

The findings revealed rejection of ten dimensions and acceptance of seven dimensions from the total number of parameters. Based upon the results of fuzzy delphi, the critical quality factor dimensions that can be considered most important may be summarised with d (threshold value) and percentages of agreement; see Table 4. All of the dimensions with a value that was less than or equal to the threshold of 0.2 and that were greater than or equal to 75% in agreement were than considered acceptable. There was rejection of all other dimensions. To summarise, it can be concluded from this study that reliability, usability, portability, maintainability, efficiency, modifiability and interoperability at the acceptable factors that are most important. Those factors, then, constitute the basic attributed for models of software quality to assure the quality of software within the assurance phase.

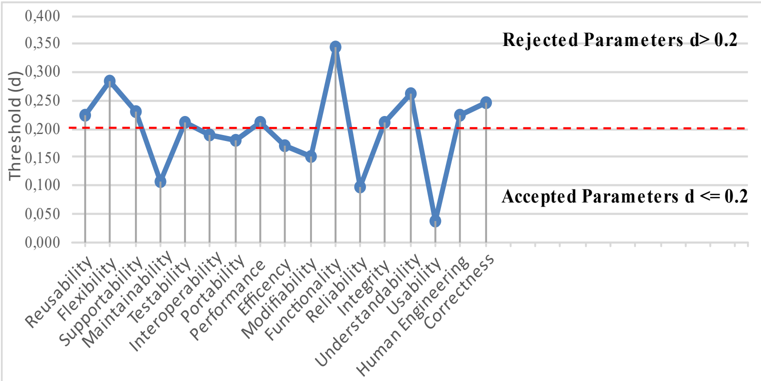

Figure 4. Rejected and accepted parameters based on threshold value (d)

Table 4. Software quality factors based on fuzzy delphi analysis

Software quality factors	Value	% of agreement
Usability	0.038	93
Reliability	0.098	80
Maintainability	0.106	87
Portability	0.179	87
Interoperability	0.190	92
Efficiency	0.171	94
Modifiability	0.152	60

5.1. Results for MCDM

Critical factors are selected from amongst the 3 groups based upon multi-perspectives. The results for discussion and evaluations are based upon 3 key steps. First, the decision matrix. Second, ranking of software quality factors. Third, the selection of the appropriate model of software quality.

5.1.1. Step one: the decision matrix

Within step one, there is a gathering of user perspectives and the relevant groups of factors. The 4 key perspectives are collected within one platform. Evaluation results for all of the perspectives were listed within the decision matrix; the calculation of mean score value for the perspectives for all of the factors is shown within Table 5.

Table 5. The decision matrix

Model	Usability	Reliability	Maintainability	Portability	Interoperability	Efficiency	Modifiability
McCall	1	1	1	1	1	1	0
Boehm	0	1	0	1	0	1	1
Dromey	1	1	1	1	0	1	0
FURPS	0	1	0	0	0	0	0
ISO	1	1	1	1	0	1	0

Table 6 shows that the quality assurance expert (the 1st evaluator) gave efficiency slight importance over other factors of quality, and modifiability was the highest weighting for the first evaluator at 37.4%. Lower weight, however, is interoperability at a value of 4%. The highest weight for the second evaluator is showing as being 43% for modifiability, however efficiency of that evaluator was a lower weight at 2%. The third evaluator showed portability as the highest factor of quality at 29%, though that evaluator had a lower value of 2% for interoperability. Furthermore, weight for each of the criteria of the quality for the models of quality for smart health application is calculated within Table 6, and those weights are employed in selecting an appropriate model of software quality.

In accordance to Table 7, the weighting of the 1st tester of the factor of software quality (multi-criteria) were as follows: the lowest value was c2 at 4.7%, then c4 at 5.3%, c6 at 5.7%, c3 at 12%, c5 at 13.1%, c1 at 21.8%, and then the highest value for c7 at 37.4%. In accordance with Tables 8 and 9, the weighting for the 2nd tester for the factor of software quality (multi-criteria) is as follows: the lowest value for c1 at 2%, then c6 at 4.2%, c2 at 5.9%, c4 at 8.6%, c3 at 13.5%, c5 at 22.6% and then the highest value for c7 at 43.2%. In accordance with Tables 10 and 11, the weighting for the factor of software quality (multi-criteria) for the 3rd tester was as follows: the lowest value for c2 at 2.7%, then c1 at 3.6%, c4 at 5.7%, c6 at 14.7%, c3 at 18.2%, c2 at 26%, and then the highest value for c5 at 29.2%.

Table 6. Comparison matrix of tester 1

Tester 1							
CR1	C1	C2	C3	C4	C5	C6	C7
	1.000	0.111	0.111	0.125	0.111	2.000	9.000
C1	1.000	0.125	0.125	0.143	0.125	3.000	9.000
	1.000	0.143	0.143	0.167	0.143	4.000	9.000
	7.000	1.000	7.000	1.000	2.000	0.250	4.000
C2	8.000	1.000	8.000	1.000	3.000	0.333	5.000
	9.000	1.000	9.000	1.000	4.000	0.500	6.000
	7.000	0.143	1.000	0.143	2.000	0.250	2.000
C3	8.000	0.125	1.000	0.125	3.000	0.333	3.000
	9.000	0.111	1.000	0.111	4.000	0.500	4.000
	6.000	1.000	7.000	1.000	2.000	0.167	4.000
C4	7.000	1.000	8.000	1.000	3.000	0.200	5.000
	8.000	1.000	9.000	1.000	4.000	0.250	6.000
	7.000	0.500	0.167	0.500	1.000	0.250	2.000
C5	8.000	0.333	0.333	0.333	1.000	0.333	3.000
	9.000	0.250	0.250	0.500	1.000	0.500	4.000
	0.250	2.000	2.000	4.000	2.000	1.000	2.000
C6	0.333	3.000	3.000	5.000	3.000	1.000	3.000
	0.500	4.000	4.000	6.000	4.000	1.000	4.000
	0.111	0.167	0.250	0.167	0.250	0.250	1.000
C7	0.111	0.200	0.333	0.200	0.333	0.333	1.000
	0.111	0.250	0.500	0.250	0.500	0.500	1.000

Table 7. Results of FAHP

Step 4 (geometric mean of fuzzy)				Step 5 (fuzzy weight)				Step 5-6 averaged weight criterion (Mi) and normalized weight criterion (Ni)			
CRI		r*i*		CRI		*Wi*		CRI	Mi	Ni	Rank
C1	1.782	2.010	2.284	C1	0.148	0.204	0.291	C1	0.214	0.218	2
C2	0.428	0.439	0.454	C2	0.036	0.045	0.058	C2	0.046	0.047	7
C3	0.893	1.150	1.285	C3	0.074	0.117	0.164	C3	0.118	0.120	4
C4	0.481	0.481	0.543	C4	0.040	0.049	0.069	C4	0.053	0.053	6
C5	0.891	1.190	1.515	C5	0.074	0.121	0.193	C5	0.129	0.131	3
C6	0.387	0.496	0.673	C6	0.032	0.050	0.086	C6	0.056	0.057	5
C7	2.737	3.471	4.137	C7	0.227	0.352	0.527	C7	0.369	0.374	1
Total	7.600	9.238	10.890					Total	0.985	1.000	
P (-1)	0.132	0.108	0.092								

Table 8. Comparison matrix of tester 2

Tester 2							
CR1	C1	C2	C3	C4	C5	C6	C7
	1.000	2.000	4.000	9.000	9.000	4.000	6.000
C1	1.000	3.000	5.000	9.000	9.000	5.000	7.000
	1.000	4.000	6.000	9.000	9.000	6.000	8.000
	0.250	1.000	4.000	1.000	7.000	0.250	4.000
C2	0.333	1.000	5.000	1.000	8.000	0.333	5.000
	0.500	1.000	6.000	1.000	9.000	0.500	6.000
	0.167	0.167	1.000	0.250	4.000	0.125	6.000
C3	0.200	0.200	1.000	0.333	5.000	0.143	7.000
	0.250	0.250	1.000	0.500	6.000	0.167	8.000
	0.111	1.000	2.000	1.000	4.000	0.167	4.000
C4	0.111	1.000	3.000	1.000	5.000	0.200	5.000
	0.111	1.000	4.000	1.000	6.000	0.250	6.000
	0.111	0.111	0.167	0.167	1.000	0.250	9.000
C5	0.111	0.125	0.200	0.200	1.000	0.333	9.000
	0.111	0.143	0.250	0.250	1.000	0.500	9.000
	0.167	2.000	6.000	4.000	2.000	1.000	4.000
C6	0.200	3.000	7.000	5.000	3.000	1.000	5.000
	0.250	4.000	8.000	6.000	4.000	1.000	6.000
	0.125	0.167	0.125	0.167	0.111	0.167	1.000
C7	0.143	0.200	0.143	0.200	0.111	0.200	1.000
	0.167	0.250	0.167	0.250	0.111	0.250	1.000

Table 9. Results of FAHP

Step 4 (geometric mean of fuzzy)				Step 5 (fuzzy weight)				Step 5-6 averaged weight criterion (Mi) and normalized weight criterion (Ni)			
CRI		ri		CRI		*Wi*		CRI	Mi	Ni	Rank
C1	0.195	0.218	0.252	C1	0.016	0.022	0.032	C1	0.023	0.020	7
C2	0.534	0.642	0.757	C2	0.044	0.065	0.096	C2	0.069	0.059	5
C3	1.219	1.472	1.739	C3	0.101	0.149	0.221	C3	0.157	0.135	3
C4	0.820	0.930	1.078	C4	0.068	0.094	0.137	C4	0.100	0.086	4
C5	2.166	2.494	2.784	C5	0.180	0.253	0.354	C5	0.262	0.226	2
C6	0.365	0.440	0.552	C6	0.030	0.045	0.070	C6	0.048	0.042	6
C7	4.137	4.757	5.344	C7	0.343	0.483	0.680	C7	0.502	0.432	1
Total	9.437	10.953	12.505					Total	1.162	1.000	
P (-1)	0.106	0.091	0.080								
INCR	0.083	0.101	0.127								

Table 10. comparison matrix of tester 3

	Tester 3						
CR1	C1	C2	C3	C4	C5	C6	C7
	1.000	0.125	7.000	6.000	7.000	6.000	2.000
C1	1.000	0.143	8.000	7.000	8.000	7.000	3.000
	1.000	0.167	9.000	8.000	9.000	8.000	4.000
	6.000	1.000	6.000	1.000	7.000	9.000	2.000
C2	7.000	1.000	7.000	1.000	8.000	9.000	3.000
	8.000	1.000	8.000	1.000	9.000	9.000	4.000
	0.143	0.125	1.000	0.250	2.000	0.250	2.000
C3	0.125	0.143	1.000	0.333	3.000	0.333	3.000
	0.111	0.167	1.000	0.500	4.000	0.500	4.000
	0.125	1.000	2.000	1.000	9.000	4.000	2.000
C4	0.143	1.000	3.000	1.000	9.000	5.000	3.000
	0.167	1.000	4.000	1.000	9.000	6.000	4.000
	0.143	0.111	0.250	0.111	1.000	0.250	2.000
C5	0.125	0.125	0.333	0.111	1.000	0.333	3.000
	0.111	0.143	0.500	0.111	1.000	0.500	4.000
	0.125	0.111	2.000	0.167	2.000	1.000	2.000
C6	0.143	0.111	3.000	0.200	3.000	1.000	3.000
	0.167	0.111	4.000	0.250	4.000	1.000	4.000
	0.250	0.250	0.250	0.250	0.250	0.250	1.000
C7	0.333	0.333	0.333	0.333	0.333	0.333	1.000
	0.500	0.500	0.500	0.500	0.500	0.500	1.000

Table 11. Results of FAHP

Step 4 (geometric mean of fuzzy)				Step 5 (fuzzy weight)				Step 5-6 Averaged weight criterion (Mi) and normalized weight criterion (Ni)			
CRI		ri		CRI		*Wi*		CRI	Mi	Ni	Rank
C1	0.335	0.357	0.390	C1	0.028	0.036	0.050	C1	0.038	0.036	6
C2	0.242	0.266	0.300	C2	0.020	0.027	0.038	C2	0.028	0.027	7
C3	1.399	1.777	2.246	C3	0.116	0.180	0.286	C3	0.194	0.182	3
C4	0.492	0.560	0.662	C4	0.041	0.057	0.084	C4	0.061	0.057	5
C5	2.387	2.901	3.451	C5	0.198	0.294	0.439	C5	0.311	0.292	1
C6	1.190	1.420	1.768	C6	0.099	0.144	0.225	C6	0.156	0.147	4
C7	1.811	2.564	3.281	C7	0.150	0.260	0.418	C7	0.276	0.260	2
Total	7.856	9.846	12.098					Total	1.064	1.000	
P (-1)	0.127	0.102	0.083								

5.1.2. Step two: the ranking of software quality factors

Included amongst models of software quality are the models of ISO 9126, FURPS, Dromey, Boehm, and McCall. The experiment, that was based upon the evaluation metric, was undertaken for FAHP-TOPSIS. Scores applying software quality factor weighting from the 1st tester through to the 3rd tester, which are represented, respectively, as W1, W2, and W3 are shown within 'Scores of different testers weighted' within Table 7, the Table 9 and within Table 11, which represents the ranking of software quality factors.

5.1.3. Step three: the selection of the appropriate model of software quality

Included amongst models of software quality are the models of ISO 9126, FURPS, Dromey, Boehm and McCall. The results showed the highest weighting being 0.39 for McCall followed by 0.34 for Boehm and then a lower weight of 0.14 for FURPS. This study concludes that McCall is the best software quality model.

6. DISCUSSION

There was utilisation of multi-perspective principles for the selection of critical factors of quality based upon high weight. The results, evaluation and discussion are based upon 3 key steps, i.e., i) selection of criteria, ii) decision matrix, and iii) selection of software quality model. The selection of crieteria step discovered the quality factors with the highest weight. The decision matrix integrate between software quality factors and software quality models in one uniform matrix.

6.1. Selection of criteria

The factors of software quality identified are shown in Table 1. The total of seventeen factors of software quality that belong to the software quality model were considered. The findings reveal that ten of the seventeen factors were rejected and seven accepted. In detail, modifiability is 60%, reliability is 80%, portability and maintainability are 87%, interoperability is 92% and efficiency is 94%. Figure 4 is a pictorial representation of the findings. In accordance with the work of [22], [25]. It was discovered that the quality factors of highest weight were usability, efficiency, portability, maintainability, reliability, modifiability, and interoperability. As such, the results of this study confirmed the aforementioned previous studies.

6.2. The decision matrix

The models of software quality and the factors of quality are collected within step 1, with the gathering of the 7 quality factors within a single platform. Evaluation results for each quality factor are listed within the decision matrix; herein there is calculation of the value of the mean score for quality factors for all of the models of software quality (see Table 12). The final results are presented within Table 12 for five quality models based upon seven factors of software quality. The matrix thus constructed (5×7) is representative of the decision matrix. The findings showed that modifiability is the quality factor that is most important with a figure of 37.4% based upon the three evaluators perspective. Modifiability is, in accordance with the work of [53], is the factor that is most important in relation to its impact upon the measurement of software quality. Usability and portability are considered the next most important factor of quality based upon the perspective of evaluators [54].

Table 12. Scores based on integrated (FAHP–TOPSIS) for quality model

QA model	S1+	S1-	W1	S1+	S1-	W2	S1+	S1-	W3	Average	Ranking
Boehm	0.374	0.1983	0.3465	0.432	0.2441	0.361	0.26	0.3209	0.5524	0.343244	2
McCall	0.1945	0.376	0.6591	0.2393	0.4346	0.6449	0.311	0.2717	0.4663	0.399711	1
Dromey	0.3963	0.1489	0.2731	0.4875	0.0921	0.1589	0.391	0.133	0.2538	0.2594	3
FURPS	0.4233	0.0	0.0	0.4962	0.0	0.0	0.413	0.0	0.0	0.148056	4
ISO9126	0.3963	0.1489	0.2731	0.4875	0.0921	0.1589	0.391	0.133	0.2538	0.2594	3

6.3. Selection of software quality model

Within step 2 there is presentation of the evaluation metrics values for the factors of software quality, i.e., interoperability, efficiency, maintainability, usability, portability, modifiability and reliability. The basis for the experiment are the evaluation metrics from integrated FAHP-TOPSIS. W1 and W2 and W3 are the categories for scores assigned to software quality factor weight from the three software quality professionals who served as evaluators; these are shown with Table 7, Table 9 and Table 11 within ‘Score with different tester (weighted)’. The consideration is that the model of Boehm has a rank of 3 with a value far from an ideal solution, and the model of McCall ranks at 1 with a value that is close to an ideal solution. The value for the model of McCall is close to 1 and so it is considered the best model of the three [22], [55].

7. CONTRIBUTIONS

Contribution to theory Initially, the study results are used for facilitating contributions to theory. Since there is a limited amount of literature dedicated to software quality, especially within applications of smart health care, the current study is an important knowledge source in relation to selection of software quality models. Contribution to method the data analysis shows that the study findings have considerable robustness and reliability in comparison to other analyses within previous studies. Two approaches are combined in this study, i.e., MCDM and fuzzy delphi, for the analysis and determination of appropriate models of quality based upon multiple criteria that may support the team leader of software quality assurance in the selection of the software quality model that is most appropriate, along with integration of valuable expert opinion from within the SQA field. Also, this study offers a new form of decision matrix resulting from consideration of multi-criteria.

8. LIMITATIONS

Given evaluator subjectivity, the research outcomes here cannot be sufficiently generalised. The outcomes of the study serve to support the team for assurance of software quality in the optimisation of their processes for selection of software quality model. The framework presented may be used extensively by the tester of software in order to discover the factors driving quality in relation to software products.

9. CONCLUSION

In selecting a model for software quality, a key aspect that ought to be borne in mind is that the needs of software products being developed ought to be satisfied by the model. The key aim for this paper has been the selection of an appropriate model of software quality through use of FAHP and TOPSIS. Decision makers utilised that approach in order to discover both ideal and non-ideal solutions. The selection of model as 'best model' is that which offers an ideal solution that is distant from being a 'negative ideal solution'. Fuzzy numbers and crisp numbers have been used in earlier models for selecting a best model within which lots of variation may occur. In a structured and simple manner, the method of TOPSIS and FAHP evaluates and selects a model that is appropriate. It was stated by the result that, when there was application of the TOPSIS method, satisfactory results were obtained and put forward with the selected model of software quality.

REFERENCES

[1] C. Free *et al.*, "The effectiveness of mobile-health technology-based health behaviour change or disease management interventions for health care consumers: a systematic review," *PLoS Medicine*, vol. 10, no. 1, pp. 1–45, 2013, doi: 10.1371/journal.pmed.1001362.
[2] T. Chen, "Assessing factors critical to smart technology applications to mobile health care- the fgm-fahp approach," *Health policy and technology*, vol. 9, no. 2, pp. 194–203, 2020, doi: 10.1016/j.hlpt.2020.02.005.
[3] V. V. -Jamsek, T. d. Jongh, I. G. -Urganci, R. Atun, and J. Car, "Mobile phone messaging for preventive health care," *Cochrane Database of Systematic Reviews*, no. 12, pp. 1–55, 2012, doi: 10.1002/14651858.CD007457.pub2.
[4] N. Nisha, M. Iqbal, A. Rifat, and S. Idrish, "Exploring the role of service quality and knowledge for mobile health services," *International Journal of E-Business Research (IJEBR)*, vol. 12, no. 2, pp. 45–64, 2016, doi: 10.4018/IJEBR.2016040104.
[5] W. Brown III, P.-Y. Yen, M. Rojas, and R. Schnall, "Assessment of the health IT usability evaluation model (health-ITUEM) for evaluating mobile health (mHealth) technology," *Journal of biomedical informatics*, vol. 46, no. 6, pp. 1080–1087, 2013, doi: 10.1016/j.jbi.2013.08.001.
[6] J. Goodman, L. Gorman, and D. Herrick, "Health information technology: benefits and problems," *National Center for Policy Analysis, Washington*, pp. 1–23, 2010.
[7] P. Keane, J. Anderson, S. McKemmish, and R. Manaszewicz, "Quality criteria models used to evaluate health websites," in *Proceedings of the 10th Asia Pacific health and Law Librarians Conference*, 2003, pp. 24–27.
[8] R. Shatnawi, "Synergies and conflicts among software quality attributes and bug fixes," *International Journal of Information Systems and Change Management*, vol. 9, no. 1, pp. 3–21, 2017.
[9] I. IEC, "9126-1 (2001) software engineering product quality-part 1: quality model," *International Organization for Standardization*, vol. 5, pp. 14–18, 2001.
[10] R. Kazman and L. Bass, "Toward deriving software architectures from quality attributes," 1994.
[11] H. Y. Abuaddous, A. M. Saleh, O. Enaizan, F. M. Ghabban, and A. B. A.-Badareen, "Automated user experience (UX) testing for mobile application: strengths and limitations," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 16, no. 4, pp. 30–45, 2022, doi: 10.3991/ijim.v16i04.26471.
[12] M. S. Hemayati and H. Rashidi, "Software quality models: a comprehensive review and analysis," *Journal of Electrical and Computer Engineering Innovations (JECEI)*, vol. 6, no. 1, pp. 59–76, 2017, doi: 10.22061/JECEI.2019.1076.
[13] B. Behkamal, M. Kahani, and M. K. Akbari, "Customizing ISO 9126 quality model for evaluation of B2B applications," *Information and Software Technology*, vol. 51, no. 3, pp. 599–609, 2009, doi: 10.1016/j.infsof.2008.08.001.
[14] M. F. Bertoa and A. Vallecillo, "Quality attributes for COTS components," *I+D Computación*, vol. 1, no. 2, pp. 128–144, 2002.
[15] A. Alvaro, E. S. d. Almeida, and S. R. d. L. Meira, "A software component quality framework," *ACM SIGSOFT Software Engineering Notes*, vol. 35, no. 1, pp. 1–18, 2010, doi: 10.1145/1668862.1668863.
[16] Y. F. Perez, C. C. Corona, and J. L. V. Galdeano, "A new model based on soft computing for evaluation and selection of software products," *IEEE Latin America Transactions*, vol. 16, no. 4, pp. 1186–1192, 2018, doi: 10.1109/TLA.2018.8362155.
[17] J. A. McCall, P. K. Richards, and G. F. Walters, "Factors in software quality. volume i. concepts and definitions of software quality," 1977.
[18] B. W. Boehm, J. R. Brown, and M. Lipow, "Quantitative evaluation of software quality," in *Proceedings of the 2nd international conference on Software engineering*, 1976, pp. 592–605.
[19] P. Kruchten, *The rational unified process: an introduction*. Boston, MA: Addison-Wesley Professional, 2004.
[20] R. G. Dromey, "A model for software product quality," *IEEE Transactions on Software Engineering*, vol. 21, no. 2, pp. 146–162, 1995, doi: 10.1109/32.345830.

[21] ISO, *Information technology-software product evaluation-quality characteristics and guide lines for their use*. Geneva, Switzerland: International Organization for Standardization, 1991.

[22] S. Kaur, S. K. Sehra, and S. S. Sehra, "A framework for software quality model selection using TOPSIS," in *2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, 2016, pp. 736–739, doi: 10.1109/RTEICT.2016.7807922.

[23] R. Kohli and S. K. Sehra, "Fuzzy multi criteria approach for selecting software quality model," *International Journal of Computer Applications*, vol. 98, no. 11, pp. 11–15, 2014, doi: 10.5120/17226-7548.

[24] M. Khan and M. D. Ansari, "Multi-criteria software quality model selection based on divergence measure and score function," *Journal of Intelligent & Fuzzy Systems*, vol. 38, no. 3, pp. 3179–3188, Mar. 2020, doi: 10.3233/JIFS-191153.

[25] M. Dabbagh and S. P. Lee, "A consistent approach for prioritizing system quality attributes," in *2013 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, Jul. 2013, pp. 317–322, doi: 10.1109/SNPD.2013.9.

[26] I. Gambo, A. Soriyan, and P. Achimugu, "Software architecture performance quality model: qualitative approach," *ARPN Journal of Systems and Software*, vol. 1, pp. 28–33, 2011.

[27] D. E. Harter and S. A. Slaughter, "Process maturity and software quality: a field study," *International Conference on Information Systems (ICIS) 2000 Proceedings*, vol. 37, pp. 407–411, 2000.

[28] N. B. Kassie and J. Singh, "A study on software quality factors and metrics to enhance software quality assurance," *International Journal of Productivity and Quality Management*, vol. 29, no. 1, pp. 24–44, 2020, doi: 10.1504/IJPQM.2020.104547.

[29] M. Toapanta, J. Nazareno, R. Tingo, F. Mendoza, A. Orizaga, and E. Mafla, "Analysis of the appropriate security models to apply in a distributed architecture," *IOP Conference Series: Materials Science and Engineering*, vol. 423, pp. 1–5, Nov. 2018, doi: 10.1088/1757-899X/423/1/012165.

[30] M. G. Siavvas, K. C. Chatzidimitriou, and A. L. Symeonidis, "QATCH - an adaptive framework for software product quality assessment," *Expert Systems with Applications*, vol. 86, pp. 350–366, Nov. 2017, doi: 10.1016/j.eswa.2017.05.060.

[31] A. A. Zaidan *et al.*, "A review on smartphone skin cancer diagnosis apps in evaluation and benchmarking: coherent taxonomy, open issues and recommendation pathway solution," *Health and Technology*, vol. 8, no. 4, pp. 223–238, Sep. 2018, doi: 10.1007/s12553-018-0223-9.

[32] O. Enaizan *et al.*, "Electronic medical record systems: decision support examination framework for individual, security and privacy concerns using multi-perspective analysis," *Health and Technology*, vol. 10, no. 3, pp. 795–822, May 2020, doi: 10.1007/s12553-018-0278-7.

[33] A. A. Zaidan, B. B. Zaidan, A. A.-Haiqi, M. L. M. Kiah, M. Hussain, and M. Abdulnabi, "Evaluation and selection of open-source EMR software packages based on integrated AHP and TOPSIS," *Journal of Biomedical Informatics*, vol. 53, pp. 390–404, Feb. 2015, doi: 10.1016/j.jbi.2014.11.012.

[34] A. A. Zaidan, B. B. Zaidan, M. Hussain, A. Haiqi, M. L. M. Kiah, and M. Abdulnabi, "Multi-criteria analysis for OS-EMR software selection problem: a comparative study," *Decision Support Systems*, vol. 78, pp. 15–27, Oct. 2015, doi: 10.1016/j.dss.2015.07.002.

[35] E. Triantaphyllou, B. Shu, S. N. Sanchez, and T. Ray, "Multi-criteria decision making: an operations research approach," *Electronics*, vol. 15, pp. 175–186, 1998.

[36] N. Kalid, A. A. Zaidan, B. B. Zaidan, O. H. Salman, M. Hashim, and H. Muzammil, "Based real time remote health monitoring systems: a review on patients prioritization and related 'big data' using body sensors information and communication technology," *Journal of Medical Systems*, vol. 42, no. 2, pp. 1–30, Feb. 2018, doi: 10.1007/s10916-017-0883-4.

[37] N. Malhotra and S. Pruthi, "An efficient software quality models for safety and resilience," *International Journal of Recent Technology and Engineering - IJRTE*, vol. 1, no. 3, pp. 66–70, 2012.

[38] O. H. Salman, A. A. Zaidan, B. B. Zaidan, Naserkalid, and M. Hashim, "Novel methodology for triage and prioritizing using 'big data' patients with chronic heart diseases through telemedicine environmental," *International Journal of Information Technology & Decision Making*, vol. 16, no. 05, pp. 1211–1245, Sep. 2017, doi: 10.1142/S0219622017500225.

[39] F. M. Jumaah, A. A. Zadain, B. B. Zaidan, A. K. Hamzah, and R. Bahbibi, "Decision-making solution based multi-measurement design parameter for optimization of GPS receiver tracking channels in static and dynamic real-time positioning multipath environment," *Measurement*, vol. 118, pp. 83–95, Mar. 2018, doi: 10.1016/j.measurement.2018.01.011.

[40] C. J. T. -Salinas, O. D. Troyer, M. J. Escalona, and M. Mejías, "A delphi-based expert judgment method applied to the validation of a mature agile framework for web development projects," *Information Technology and Management*, vol. 20, no. 1, pp. 9–40, Mar. 2019, doi: 10.1007/s10799-018-0290-7.

[41] M. Masdari and H. Khezri, "Service selection using fuzzy multi-criteria decision making: a comprehensive review," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 2803–2834, Feb. 2021, doi: 10.1007/s12652-020-02441-w.

[42] K. W. Khaw *et al.*, "Modelling and evaluating trust in mobile commerce: a hybrid three stage fuzzy delphi, structural equation modeling, and neural network approach," *International Journal of Human–Computer Interaction*, vol. 38, no. 16, pp. 1529–1545, Oct. 2022, doi: 10.1080/10447318.2021.2004700.

[43] C.-T. Chen, "Extensions of the TOPSIS for group decision-making under fuzzy environment," *Fuzzy Sets and Systems*, vol. 114, no. 1, pp. 1–9, Aug. 2000, doi: 10.1016/S0165-0114(97)00377-1.

[44] C.-H. Cheng and Y. Lin, "Evaluating the best main battle tank using fuzzy decision theory with linguistic criteria evaluation," *European Journal of Operational Research*, vol. 142, no. 1, pp. 174–186, Oct. 2002, doi: 10.1016/S0377-2217(01)00280-6.

[45] H. Chu and G. Hwang, "A delphi-based approach to developing expert systems with the cooperation of multiple experts," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2826–2840, May 2008, doi: 10.1016/j.eswa.2007.05.034.

[46] T. L. Saaty and M. S. Ozdemir, "Why the magic number seven plus or minus two," *Mathematical and Computer Modelling*, vol. 38, no. 3–4, pp. 233–244, Aug. 2003, doi: 10.1016/S0895-7177(03)90083-5.

[47] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965, doi: 10.1016/S0019-9958(65)90241-X.

[48] T. L. Saaty, *Decision making for leaders: the analytic hierarchy process for decisions in a complex world*. Pittsburgh, Pennsylvania: RWS publications, 2001.

[49] N. A. Azeez and C. V. d. Vyver, "Security and privacy issues in e-health cloud-based system: A comprehensive content analysis," *Egyptian Informatics Journal*, vol. 20, no. 2, pp. 97–108, Jul. 2019, doi: 10.1016/j.eij.2018.12.001.

[50] H. Ahmadi, M. S. Rad, M. Nilashi, O. Ibrahim, and A. Almaee, "Ranking the meso level critical factors of electronic medical records adoption using topsis method," *Health Informatics - An International Journal*, vol. 42, no. 4, pp. 82–93, 2016, doi: 10.12865/CHSJ.42.01.12.

[51] P.-L. Chang and C.-W. Hsu, "An assessment model for hydrogen fuel cell applications: fuzzy delphi approach," *International Journal of Social Science and Humanity*, vol. 1, no. 3, pp. 218–223, 2011, doi: 10.7763/ijssh.2011.v1.38.

[52] A. H. Ishak and M. R. Osman, "A systematic literature review on Islamic values applied in quality management context," *Journal of Business Ethics*, vol. 138, no. 1, pp. 103–112, Sep. 2016, doi: 10.1007/s10551-015-2619-z.
[53] Abdullah, K. M. H, and R. Srivastava, "Testability measurement model for object oriented design (TMMOOD)," *International Journal of Computer Science and Information Technology*, vol. 7, no. 1, pp. 153–163, 2015, doi: 10.5121/ijcsit.2015.7115.
[54] S. Khaddaj and G. Horgan, "The evaluation of software quality factors in very large information systems," *Electronic Journal of Information Systems Evaluation*, vol. 7, no. 1, pp. 43–48, 2004.
[55] A. B. A.-Badareen, M. H. Selamat, M. A. Jabar, J. Din, and S. Turaev, "Software quality models: a comparative study," in *Software Engineering and Computer Systems*, 2011, pp. 46–55, doi: 10.1007/978-3-642-22170-5_4.

BIOGRAPHIES OF AUTHORS

Ashraf Mousa Saleh is an Assistant Professor of Software Engineering and the Director of Information Technology and Communication at Amman Arab University (AAU). He received his BS in Computer Science from the Jordan University of Science and Technology (JUST), Jordan, a MSc in Information Technology from Utara University in Malaysia, and Ph.D in Software Engineering from USIM University in Malaysia. He can be contacted at email: asaleh@aau.edu.jo

Odai Enaizan is an Assistant Professor of Management Information System at University of Tabuk (UT). He received his BS in Management Information System from the Al-Balqa University, Jordan, a M.Sc in Information Technology from Tenaga University in Malaysia, and PhD in Management Information System from USIM University in Malaysia. He can be contacted at email: Aonaizan@ut.edu.sa.