Evaluating Software Quality Attributes using Analytic Hierarchy Process (AHP)

Botchway Ivy Belinda[1], Akinwonmi Akintoba Emmanuel[2], Nunoo Solomon[3], Alese Boniface Kayode[4]
Department of Computer Science and Engineering, University of Mines and Technology, UMaT, Tarkwa, Ghana[1]
Department of Cybersecurity, Federal University of Technology, Akure, FUTA, Akure, Nigeria[2, 4]
Department of Electrical and Electronic Engineering, University of Mines and Technology, UMaT, Tarkwa, Ghana[3]

Abstract—**The use of quality software is of importance to stakeholders and its demand is on the increase. This work focuses on meeting software quality from the user and developer's perspective. After a review of some existing software-quality models, twenty-four software quality attributes addressed by ten models such as the McCall's, Boehm's, ISO/IEC, FURPS, Dromey's, Kitchenham's, Ghezzi's, Georgiadou's, Jamwal's and Glibb's models were identified. We further categorized the twenty-four attributes into a group of eleven (11) main attributes and another group of thirteen (13) sub-attributes. Thereafter, questionnaires were administered to twenty experts from fields including Cybersecurity, Programming, Software Development and Software Engineering. Analytic Hierarchy Process (AHP) was applied to perform a multi-criteria decision-making assessment on the responses from the questionnaires to select the suitable software quality attribute for the development of the proposed quality model to meet both users and developer's software quality requirements. The results obtained from the assessment showed Maintainability to be the most important quality attribute followed by Security, Testability, Reliability, Efficiency, Usability, Portability, Reusability, Functionality, Availability and finally, Cost.**

Keywords—Analytic Hierarchy Process (AHP); software quality; quality attribute; quality model; sub-attributes

I. Introduction

Software quality is a paramount issue to all software stakeholders in a given establishment and its demand is increasing rapidly due to customer demand [1]. In the last few decades, the importance of the use of quality software has increased exponentially [2]. Software users see software as a tool to enable them to carry out their daily activities with ease, and hence, use it to perform sensitive tasks [3]. The use of less quality software can, directly and indirectly, endanger one's life [30] as well as causing huge loss to software users. As a result, many software quality models have been proposed to evaluate software quality, yet, none of these models has been widely accepted as the benchmark for assessing software quality. This is because these models do not address all the important software quality attributes that are of keen interest to stakeholders and are tailored towards meeting specific project's requirements. To address stakeholder requirements, custom software quality models have been proposed [4]. These custom made quality models offer different benefits to the software industry and research community and hence do not cover a wide scope of quality attributes.

This research presents an evaluation of software quality attributes using the Analytic Hierarchy Process (AHP). It was conducted based on a questionnaire given to stakeholders to assess the quality attributes they expect a software quality model to have. As a result, an evaluation of these quality attributes was made and represented with a graph to pictorially highlight the percentage of weight each quality attribute ranked. Ranking the software quality attributes will assist developers greatly in selecting the best quality attribute for evaluating developed software. Previous works have failed to rank quality attributes and have led to the proposal of numerous custom software quality models.

The rest of the paper is organized as follows: Section II discusses related work done on AHP, software quality models and quality attributes. Section III discusses software quality, quality models and attributes. The methodology used to select the software quality attributes is illustrated in Section IV. Section V presents the conclusion and discusses future works.

II. Related Works

Software quality models have been reviewed by numerous researchers in addressing software quality problems. The authors in [14] evaluated the quality of software in Enterprise Resource Planning (ERP) systems using the ISO 9126 model. They offered a comparison between existing quality models and identified the quality characteristics of ERP systems but they did not rank the main quality characteristics of the model.

In [15], research was conducted on an analytical and comparative study of software usability quality factors. They analysed ten famous quality models for developing a usability model that satisfies the demand of current business software and proposed an integrated improved usability model for assuring software quality. The new usability evaluation model was proposed from ten models of McCall, Boehm, Shackel, FURPS, Nielson, SUMI, ISO 9242-11, ISO 9126, and QUIM model.

A research was conducted by [16] on an approach for enhancing software quality based on ISO 9126 quality model. They were able to propose a new quality model for integrating some quality attributes in software development.

The authors in [13] also worked on the quality assessment of Commercial-Off-The-Shelf products by adopting the ISO 9126 quality model.

Another study was conducted by [3] on software quality attributes to enhance software quality assurance. The authors did this research because, in recent times, industries are giving more attention to software quality improvement. Therefore, they focused on meeting customer perspectives of software quality to propose a new model. The limitation of the research is that it did not address availability, testability and reusability problems.

Authors in [25] worked on extending Dromey's quality model to specify the security quality requirements in a software product. They conducted the research based on the increase in cybercrimes. The model was able to enhance the security requirement of software and trained people on how to develop secure software.

A study by researchers in [26] adapted the ISO/IEC 9126 quality model to evaluate Enterprise Resource Planning (ERP) systems. The model was proposed as a result of the urge in the increasing usage of ERP systems by organisations to get faster data transactions. The researchers proposed the model to have six (6) main software quality attributes including functionality, maintainability, reliability, efficiency, usability and portability. The limitation was that the model did not address some of the most important software quality attributes such as availability, testability and flexibility. It also did not rank the quality attributes.

In [27], the authors presented a software quality model for academic information systems. Their objective was to guide academic institutions that are in the process of building their E-learning systems to evaluate and choose the appropriate software attributes that are essential to the success of the entire system. The researchers identified the key attributes for Information System's Software Quality (ISSQ) from the users' perspective to measure the quality of the E-learning system. The proposed model consisted of six (6) standard attributes with their sub-attributes. This was achieved based on the ISO/IEC 9126 model. The limitation was that the proposed model failed to evaluate the importance of quality attributes.

These researchers have shown how the use of software quality models is gaining much importance in the development of software. Nevertheless, they have not ranked the quality attributes and hence, it is difficult to know the weight of each attribute to ease decision making. This research work employs the use of the Analytic Hierarchy Process, a multi-criteria decision-making tool to evaluate software quality attributes and rank them.

Analytic Hierarchy Process (AHP) has been applied by several researchers to enhance group decisions. The researchers in [17] applied this technique to evaluate and select Commercial-off-the-shelf (COTS) components. They found AHP to be useful in making trade-offs between tangible and intangible factors in calculating the weight of COTS components. Applying these weights as coefficients of an objective function in the proposed model helped to determine the best component under constraints such as budgetary constraint, compatibility among components and system reliability. Their findings have validated AHP to be an effective and flexible tool.

AHP was applied by [4] to produce an integrated framework that applies statistical analysis to generate software quality models tailored to stakeholder specifications. They found AHP to be quite accessible and conducive for decision-making that requires the reduction of decisions complexity in pair-wise matrices.

AHP was also applied by [18] in evaluating the reliability of object-oriented software systems. They took the ISO/IEC 9126 model as the base model for the evaluation. Their results showed AHP to be useful for making decisions for the hierarchical structure of the model.

Authors in [28] applied the Analytic Hierarchy Process to develop an algorithm for evaluating software functionality. The research was due to the increase in the number of sub-attributes of software functionality quality attribute. They wanted to know the most important sub-attribute that has a great impact on software products. The AHP technique was seen to be a useful tool for the decision-making process.

In [29], the AHP technique was used to perform a risk assessment of software quality. The authors were able to construct an index system of software quality risk assessment by calculating the weight and order of risk factors. With the use of AHP, they were able to categorise risk factors into demand risk, technology risk, process risk and management risk.

The authors in [19] applied AHP to analyse software reliability. They reported that although software reliability is an important quality attribute, different stakeholders have a variety of views in that regard. Hence, they applied AHP which is designed to manage human assessment subjectively.

The Analytic Hierarchy Process has been seen to effectively aid researchers in solving complex decision-making problems in various fields but its rate of application in the software quality assurance industry is minimal. Most software quality attributes used for software quality assurance have not been ranked, hence, it is difficult to note the important attributes to use to evaluate software projects. In this research, AHP is used to rank quality attributes by using the value of their criteria weights. The higher the criteria weight, the higher its importance in evaluating software quality.

III. Software Quality

Software quality is a benchmark for measuring software requirements and the prerequisite to meet the user's specifications. Software quality involves user requirements, system design, documentation, and all the requirements needed for the development of professionally acceptable software [5]. It strictly follows the software development life cycle and evaluates and improves software performance [5]. Software quality can be enforced using software quality models.

A. Software Quality Models

Different software quality models have been proposed by researchers such as McCall [6], Boehm [7], Jamwal [8], Grady [9], Dromey [10] and ISO/IEC [11] among others as shown in Table I. These quality models contain quality attributes that may be used to ascertain the quality of a software product by determining how the software executes its code or how the software architecture is structured and organized with the system's requirements [12]. All the quality models have software quality attributes and sub-attributes used for the measurement of software quality [13]. Quality attributes and sub-attributes are used to characterize products and can be measured. They usually end with the word "lity". According to the ISO 9126 standard, a software quality model is expected to have the following attributes: Functionality, Reliability, Usability, Efficiency, Maintainability and Portability.

TABLE I. SOFTWARE QUALITY MODELS AND THEIR SUB-ATTRIBUTES

Quality Models Quality Attributes	**McCall**	**Boehm**	**FURPS**	**Dromey**	**ISO-9126**	**Glibb**	**Kitchenham and Pickard**	**Ghezzi *et al.***	**Georgiadou**	**Jamwal and Jamwal**	**Proposed Quality Model**
Maintainability	/	/		/	/		/	/	/		/
Flexibility	/	/						/			
Testability	/	/									/
Correctness	/									/	
Reliability	/	/	/	/	/		/	/	/	/	/
Efficiency	/	/		/				/	/		/
Usability	/	/	/	/	/	/	/	/	/	/	/
Portability	/	/		/	/			/	/	/	/
Reusability	/			/				/			/
Interoperability	/										
Understandability		/									
Functionality			/	/	/				/		/
Performance			/							/	
Supportability			/								
Availability						/					/
Adaptability						/					
Accuracy								/			
Robustness									/	/	
Extensibility			/								
Security									/		/
Cost										/	/
Integrity	/										
Confidentiality											
Non-Repudiation											

B. *Software Quality Attributes*

Software quality attributes are used to measure customer fulfilment of a product for other similar products. They are also used by software developers to develop quality software. These attributes include correctness, reliability, portability, efficiency, maintainability, supportability, functionality, usability, availability, among others. The software development life cycle ensures that implementing quality attributes in software development may result in the production of a well-engineered software product and is to be enforced throughout the development, implementation, and deployment phases of the software [5].

C. *Software Quality Attributes Descriptions*

This section itemizes and describes some software quality attributes.

- Correctness: Correctness refers to the capability of software to meet its required results.
- Usability: Usability is the ease of use and learnability of software by customers.
- Efficiency: Efficiency is the ability of software to perform well, given that tasks are completed faster while using fewer resources and saving computer power with great performance.
- Reliability: Reliability refers to the probability of software operating in a given environment within a specified period to perform well without encountering a breakdown.
- Accuracy: Accuracy refers to the degree to which a software product provides the right results during usage without encountering an error.

- Robustness: Robustness refers to the ability of a software product to cope with any form of error it may encounter during operation.
- Functionality: Functionality is the ability of software to perform the tasks for which it was intended.
- Performance: Performance refers to the total effectiveness of a software product.
- Availability: Availability refers to the degree to which a software product is operational and easily accessible when needed for usage.
- Maintainability: The ease with which software can be modified to correct faults or improve performance.
- Flexibility: Flexibility is the ability of software to adapt to possible future changes in its requirements.
- Portability: The measure of the ease of transferring software from one computing environment to the other.
- Reusability: Reusability is the use of existing tested and validated loosely coupled components in the development of software applications.
- Testability: Testability is the ease with which the correctness of software can be verified.
- Understandability: The capability of a software product to enable the user to understand whether it is suitable and its usability for specific tasks and conditions for use.
- Interoperability: Interoperability is the ease with which software is used with other software applications.

IV. Analytic Hierarchy Process (AHP)

AHP is a method of multi-criteria evaluation that organizes and simplifies the decision-making process. It was originally developed by Thomas L. Saaty [20] to provide measures of judgement consistency; to derive priorities among criteria and alternatives, and to simplify the rating of preferences among decision criteria using pair-wise comparisons [21]. The AHP decision-making tool is robust and flexible in dealing with complex decision problems. It uses a multi-level hierarchical structure of objective or goal, criteria or attributes, and alternatives.

AHP is based on mathematics and psychology [22]. It helps decision-makers to find a decision that best suits their goal and their understanding of a given problem. It is a method to derive ratio scales from paired comparisons [23] and is based on a certain scale that changes subjective judgements into objective judgement and solves qualitative problems with quantitative analysis. It is simple and hence has seen its application in many fields.

A. *Assessment of Quality Attributes*

The research uses an Analytical Hierarchy Process (AHP) to perform a multi-criteria decision-making assessment to select a suitable software quality attribute for the development of the quality model. The selection will be made from eleven attributes (Maintainability M(s), Testability T(s), Reliability R(s), Efficiency E(s), Usability U(s), Portability P(s), Reusability Re(s), Cost Co(s), Functionality Fn(s), Security S(s) and Availability A(s)) and three alternatives ("Mostly addressed", "Doubles up as a Sub-attribute", "Has Sub-attributes"). This information will be used to develop a hierarchical structure with the goal at the top level, the attributes at the second level, and the alternatives at the third level as shown in Fig. 1.

The hierarchical structure obtained was synthesized to determine the relative importance of each attribute to the goal. This is done using a pair-wise comparison matrix with the help of a scale of relative importance as shown in Table II.

The quality attributes used for the judgement matrix are shown in Table III. It consists of eleven (11) main attributes and thirteen (13) sub-attributes. The AHP technique was only applied to the eleven (11) main attributes.

B. *Selection of Appropriate Software Quality Attributes using the Analytic Hierarchy Process (AHP)*

The judgement matrix was determined by twenty (20) experts' decisions, based on related research. The implementation was done using MATLAB/Simulink Software R2020b. The software allows for easy calculation and analysis for the decision-making process. It also helps in constructing the model and drawing analysis.

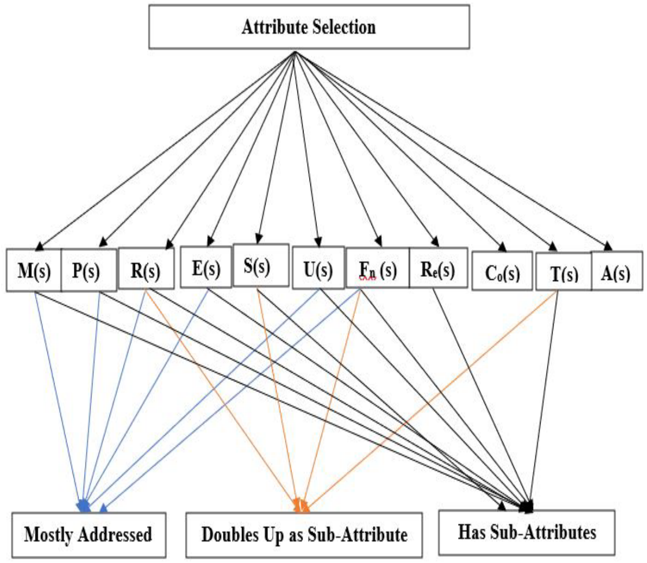

Fig. 1. Hierarchical Structure of Software Quality Attributes.

TABLE II. The Scale of Comparison [24]

Scale of Importance	Degree of Preference
1	Equal Importance
3	Moderate Importance
5	Strong Importance
7	Very Strong Importance
9	Extreme Importance
2,4,6,8	Intermediate Values
1/3, 1/5, 1/7, 1/9	Values for Inverse Comparison

TABLE III. QUALITY ATTRIBUTES AND THEIR SUB-ATTRIBUTES

Attribute	**Sub-attribute**
Maintainability M(s)	Flexibility
	Extensibility
	Supportability
Usability U(s)	Understandability
Reliability R(s)	Robustness
	Accuracy
Testability T(s)	
Functionality $F_n(s)$	Correctness
	Interoperability
Availability A(s)	
Reusability $R_e(s)$	
Cost $C_o(s)$	
Efficiency E(s)	Performance
Portability P(s)	Adaptability
Security S(s)	Integrity
	Confidentiality
	Non-Repudiation

C. Quality Attribute Selection Judgement Matrices

A geometric mean of the scores from the questionnaire was found and represented in Table IV for effective criteria and pair-wise comparison.

Table V shows the normalised pair-wise comparison matrix while Table VI shows the consistency matrix.

Twenty experts were given questionnaires to fill for the multi-criteria decision process. The geometric mean of these questionnaires was found by multiplying the values for each of the attributes in Table IV and setting it to the 1/nth power. The sum of each attribute was finally calculated. The geometric mean of the scores was found by

$$\left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}} = \sqrt[n]{x_1 x_2 \cdots x_n} \tag{1}$$

Where n is the number of terms that are being multiplied.

The normalised pair-wise comparison matrix was found in Table V by diving each of the values for the attributes in Table IV by the sum. To calculate the criteria weight, an average of the rows is found.

It is seen from Table V that the criteria weight of Maintainability is 17.37%, Testability is 13.02%, Usability is 7.22%, Functionality is 6.22%, Cost is 4.73%, Portability is 7.13%, Availability is 5.99%, Reusability is 6.86%, Security is 13.61%, Reliability is 10.35% and Efficiency is 7.49%. Maintainability is seen to have the highest weight while Cost is seen to have the lowest weight.

To check for the cost of the expert's evaluation, the consistency of the pair-wise comparison matrix is calculated. This is done by multiplying the criteria weight by the pair-wise comparison matrix which is not normalised in Table IV as shown in Table VI. The weighted sum of the new matrix is found and then divided by the criteria weight. The overall sum is found for the calculation of $\lambda_{\mathbf{max}}$ and Consistency Ratio (CR). The value of the consistency ratio must be less than 0.1 to make the judgement matrix acceptable.

$$\lambda_{\max} = \frac{134.04}{11} = 12.186 \tag{2}$$

$$\mathrm{CI} = \frac{\lambda_{\max} - \mathrm{n}}{\mathrm{n} - 1} = \frac{12.186 - 11}{10} = 0.119 \tag{3}$$

$$\mathrm{CR} = \frac{\mathrm{CI}}{\mathrm{RI}} = \frac{0.119}{1.51} = 0.079 \tag{4}$$

The selection judgement matrix shows consistency since the value of the Consistency Ratio (CR) is 0.079 which is less than 0.1.

TABLE IV. GEOMETRIC MEAN OF THE FILLED QUESTIONNAIRE

Attributes	**M(s)**	**T(s)**	**R(s)**	**P(s)**	**A(s)**	**E(s)**	**Fn(s)**	**Re(s)**	**S(s)**	**U(s)**	**Co(s)**
M(s)	1.00	2.05	2.44	2.98	2.95	2.75	2.95	2.18	1.55	1.59	2.80
T(s)	0.49	1.00	2.04	2.30	2.99	2.02	2.85	1.90	1.20	1.31	2.10
R(s)	0.41	0.49	1.00	1.00	1.95	1.72	2.69	1.50	1.28	2.10	2.59
P(s)	0.34	0.43	1.00	1.00	1.20	0.75	0.98	1.29	1.05	1.03	1.68
A(s)	0.34	0.33	0.51	0.51	1.00	1.02	1.01	0.67	0.20	2.54	1.54
E(s)	0.36	0.50	0.58	1.33	0.98	1.00	1.02	1.59	0.28	2.85	1.50
Fn(s)	0.34	0.35	0.37	1.02	0.99	0.98	1.00	2.65	0.32	0.55	1.85
Re(s)	0.46	0.53	0.67	0.78	1.49	0.63	3.57	1.00	0.23	0.60	1.55
S(s)	0.65	0.83	0.78	0.95	5.00	3.57	3.13	4.35	1.00	0.87	1.90
U(s)	0.63	0.76	0.48	0.97	0.39	0.35	1.82	1.67	1.15	1.00	1.00
Co(s)	0.36	0.48	0.39	0.60	0.65	0.67	0.54	0.65	0.53	1.00	1.00
SUM	**5.36**	**7.75**	**10.26**	**13.44**	**19.60**	**15.46**	**21.56**	**19.44**	**8.79**	**15.44**	**19.51**

TABLE V. NORMALISED PAIR-WISE COMPARISON MATRIX

Attributes	Criteria Weight	Criteria Weight (%)
M(s)	0.1737	17.37
T(s)	0.1302	13.02
R(s)	0.1035	10.35
P(s)	0.0713	7.13
A(s)	0.0599	5.99
E(s)	0.0749	7.49
Fn(s)	0.0622	6.22
Re(s)	0.0686	6.86
S(s)	0.1361	13.61
U(s)	0.0722	7.22
Co(s)	0.0473	4.73

TABLE VI. CONSISTENCY OF PAIR-WISE COMPARISON MATRIX

Attributes	Criteria Weight (CW)	Weighted Sum Value (WSV)	WSV/CW
M(s)	0.1737	2.079816	11.973
T(s)	0.1302	1.585515	12.178
R(s)	0.1035	1.27414	12.308
P(s)	0.0713	0.864021	12.116
A(s)	0.0599	0.720668	12.029
E(s)	0.0749	0.903899	12.057
Fn(s)	0.0622	0.763373	12.274
Re(s)	0.0686	0.847724	12.359
S(s)	0.1361	1.718038	12.623
U(s)	0.0722	0.880394	12.196
Co(s)	0.0473	0.564336	11.929
SUM			**134.04**
$\lambda_{\max}$ = 12.186 CR = 0.079			

It can be seen from Table V that Maintainability M(s) has the highest weight which is 17.37% while Cost Co(s) has the lowest weight of 4.73%. Fig. 2 shows a graphical representation of the weights of the software quality attributes.

D. Alternative Selection Judgement Matrices

The alternatives which are “Mostly addressed”, “Doubles up as Sub-attributes”, and “Has Sub-attributes” were also analysed for Maintainability M(s) as shown in Table VII.

Table VII shows that Maintainability has a higher weight of 74% for being mostly addressed and a lower weight of 11% for doubling up as a sub-attribute.

The alternatives were also analysed for Testability T(s) as shown in Table VIII.

Table VIII shows that Testability has a higher weight of 67% for being mostly addressed and a lower weight of 10% for doubling up as a sub-attribute.

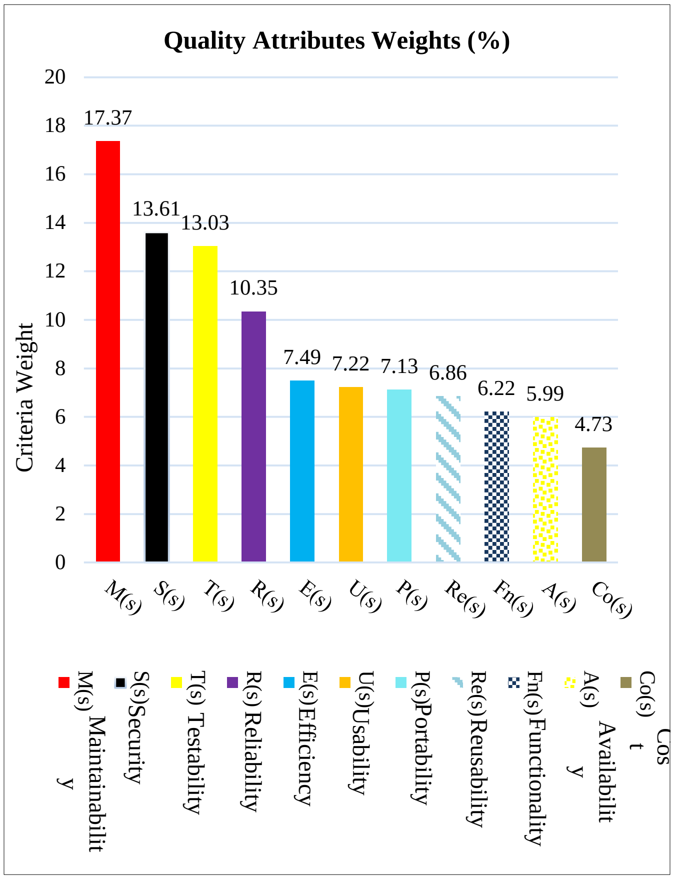

Fig. 2. Weights of the Software Quality Attributes.

TABLE VII. THE WEIGHT OF ALTERNATIVES FOR M(S)

M(s)	Has Sub-Attributes	Doubles Up as Sub-attribute	Mostly Addressed	Criteria Weight
Has Sub-Attributes	1.00	1.00	1/4	0.15
Doubles Up as Sub-attribute	1.00	1.00	1/9	0.11
Mostly Addressed	4.00	9.00	1.00	0.74
$\lambda_{\max}$ = 3.0749			CR = 0.0646	

TABLE VIII. THE WEIGHT OF ALTERNATIVES FOR T(S)

T(s)	Has Sub-Attributes	Doubles Up as Sub-attribute	Mostly Addressed	Criteria Weight
Has Sub-Attributes	1.00	3.00	1/4	0.23
Doubles Up as Sub-attribute	1/3	1.00	1/5	0.10
Mostly Addressed	4.00	5.00	1.00	0.67
$\lambda_{\max}$ = 3.0869			CR = 0.07496	

The alternatives were also analysed for Reliability R(s) as shown in Table IX.

Table IX shows that Reliability has a higher weight of 78% for being mostly addressed and a lower weight of 8% for doubling up as a sub-attribute.

The alternatives were also analysed for Efficiency E(s) as shown in Table X.

Table X shows that Efficiency has a higher weight of 62% for being mostly addressed and a lower weight of 10% for doubling up as a sub-attribute.

The alternatives were also analysed for Usability U(s) as shown in Table XI.

Table XI shows that Usability weights 80% for being mostly addressed and a weight of 8% for doubling up as a sub-attribute.

The alternatives were also analysed for Portability P(s) as shown in Table XII.

Table XII shows that Portability weighs 56% for being mostly addressed and a weight of 9% for doubling up as a sub-attribute.

The alternatives were also analysed for Reusability Re(s) as shown in Table XIII.

Table XIII shows that Reusability has a weight of 41% for having sub-attributes and a weight of 26% for doubling up as a sub-attribute.

The alternatives were also analysed for Functionality Fn(s) as shown in Table XIV.

Table XIV shows that Functionality's weightiness for being mostly addressed is 57% and 10% for doubling up as a sub-attribute.

TABLE IX. THE WEIGHT OF ALTERNATIVES FOR R(S)

R(s)	**Mostly Addressed**	**Has Sub-Attributes**	**Doubles Up as Sub-attribute**	**Criteria Weight**
Mostly Addressed	1.00	7.00	8.00	0.78
Has Sub-Attributes	1/7	1.00	2.00	0.14
Doubles Up as Sub-attribute	1/8	1/2	1.00	0.08
$\lambda_{\textbf{max}}$ = 3.035		CR = 0.0304		

TABLE X. THE WEIGHT OF ALTERNATIVES FOR E(S)

E(s)	**Has Sub-Attributes**	**Doubles Up as Sub-attribute**	**Mostly Addressed**	**Criteria Weight**
Has Sub-Attributes	1.00	4.00	1/3	0.28
Doubles Up as Sub-attribute	1/4	1.00	1/5	0.10
Mostly Addressed	3.00	5.00	1.00	0.62
$\lambda_{\textbf{max}}$ = 3.0867		CR = 0.0747		

TABLE XI. THE WEIGHT OF ALTERNATIVES FOR U(S)

U(s)	**Mostly Addressed**	**Has Sub-Attributes**	**Doubles Up as Sub-attribute**	**Criteria Weight**
Mostly Addressed	1.00	9.00	8.00	0.80
Has Sub-Attributes	1/9	1.00	2.00	0.12
Doubles Up as Sub-attribute	1/8	1/2	1.00	0.08
$\lambda_{\textbf{max}}$ = 3.075		CR = 0.0649		

TABLE XII. THE WEIGHT OF ALTERNATIVES FOR P(S)

P(s)	**Has Sub-Attributes**	**Doubles Up as Sub-attribute**	**Mostly Addressed**	**Criteria Weight**
Has Sub-Attributes	1.00	5.00	1/2	0.35
Doubles Up as Sub-attribute	1/5	1.00	1/5	0.09
Mostly Addressed	2.00	5.00	1.00	0.56
$\lambda_{\textbf{max}}$ = 3.054		CR = 0.0465		

TABLE XIII. THE WEIGHT OF ALTERNATIVES FOR R_E (S)

Re(s)	**Mostly Addressed**	**Doubles Up as Sub-attribute**	**Has Sub-Attributes**	**Criteria Weight**
Mostly Addressed	1.00	1.00	1.00	0.33
Doubles Up as Sub-attribute	1.00	1.00	1/2	0.26
Has Sub-Attributes	1.00	2.00	1.00	0.41
$\lambda_{\textbf{max}}$ = 3.054		CR = 0.0463		

TABLE XIV. THE WEIGHT OF ALTERNATIVES FOR F_N (S)

Fn(s)	**Has Sub-Attributes**	**Doubles Up as Sub-attribute**	**Mostly Addressed**	**Criteria Weight**
Has Sub-Attributes	1.00	4.00	1/2	0.33
Doubles Up as Sub-attribute	1/4	1.00	1/5	0.10
Mostly Addressed	2.00	5.00	1.00	0.57
$\lambda_{\textbf{max}}$ = 3.0247		CR = 0.0213		

The alternatives were also analysed for Availability A(s) as shown in Table XV.

Table XV shows that Availability has a weight of 60% for doubling up as a sub-attribute and a lower weight of 17% for being most addressed.

The alternatives were also analysed for Cost Co(s) as shown in Table XVI.

Table XVI shows that Cost has a weight of 77% for doubling up as a sub-attribute and a weight of 11% for having sub-attributes.

The alternatives were also analysed for Security S(s) as shown in Table XVII.

Table XVII shows that Security has a weight of 56% for doubling up as a sub-attribute and a weight of 9% for having sub-attributes.

The overall weights for the software quality attribute selection are summarised in Table XVIII.

TABLE XV. THE WEIGHT OF ALTERNATIVES FOR A(S)

A(s)	**Mostly Addressed**	**Has Sub-Attributes**	**Doubles Up as Sub-attribute**	**Criteria Weight**
Mostly Addressed	1.00	1.00	1/5	0.17
Has Sub-Attributes	1.00	1.00	1/2	0.23
Doubles Up as Sub-attribute	5.00	2.00	1.00	0.60
$\lambda_{\max}$ = 3.0951		CR = 0.08196		

TABLE XVI. THE WEIGHT OF ALTERNATIVES FOR C_O (S)

Co(s)	**Has Sub-Attributes**	**Mostly Addressed**	**Doubles Up as Sub-attribute**	**Criteria Weight**
Has Sub-Attributes	1.00	1.00	1/9	0.11
Mostly Addressed	1.00	1.00	1/5	0.13
Doubles Up as Sub-attribute	9.00	5.00	1.00	0.77
$\lambda_{\max}$ = 3.0389		CR = 0.0336		

TABLE XVII. THE WEIGHT OF ALTERNATIVES FOR S(S)

S(s)	**Mostly Addressed**	**Has Sub-Attributes**	**Doubles Up as Sub-attribute**	**Criteria Weight**
Mostly Addressed	1.00	2.00	1/2	0.35
Has Sub-Attributes	1/2	1.00	1/9	0.09
Doubles Up as Sub-attribute	2.00	9.00	1.00	0.56
$\lambda_{\max}$ = 3.0745		CR = 0.0642		

TABLE XVIII. THE WEIGHTS FOR SOFTWARE QUALITY ATTRIBUTE SELECTION

Element	**Weight**
Alternatives	
Mostly Addressed	0.520
Doubles up as Sub- attributes	0.221
Has Sub-attributes	0.259
Criteria or Attributes	
Maintainability	0.1737
Testability	0.1302
Reliability	0.1035
Efficiency	0.749
Usability	0.722
Portability	0.713
Reusability	0.686
Security	0.1361
Functionality	0.622
Availability	0.599
Cost	0.473
Combined Consistency: 0.057	

The results in Table XVIII show that “Mostly Addressed” is the highest-ranking software quality alternative with 52% and “Has Sub- attribute” is the lowest ranking alternative with 22%. The result also shows Maintainability as the highest-ranking software quality attribute with 17%. Table XVIII also shows that the overall analysis is consistent since the value of CR is 0.057 which is less than 0.1.

V. RESULTS AND DISCUSSIONS

The software quality attributes have been evaluated and according to Table IV, Maintainability is seen to weigh 17.37%, Testability has a percentage of 13.02%, Reliability has a percentage of 10.35, Efficiency has a percentage of 7.49, Usability has a percentage of 7.22, Portability has a percentage of 7.13, Reusability has a percentage of 6.86, Functionality has a percentage of 6.22, Security weighs 13.61, Availability has a percentage of 5.99 and Cost has a percentage of 4.73. This was pictorially represented in Fig. 2.

Tables VII, VIII, …, XVII has shown that Maintainability has a higher weight of 74% for being mostly addressed and a lower weight of 11% for doubling up as a sub-attribute. Testability weight of 67% for being mostly addressed and a weight of 10% for doubling up as a sub-attribute. Reliability has 78% for being mostly addressed and a weight of 8% for doubling up as a sub-attribute. Reliability is seen to also have 78% for being mostly addressed and a weighs 8% for doubling up as a sub-attribute. 62% was the weight of Efficiency for being mostly addressed and 10% for doubling up as a sub-attribute. Usability’s weightiness for being mostly addressed is 80% and 8% for doubling up as a sub-attribute. Portability also has a weight of 56% for being mostly addressed and a weight of 9% for doubling up as a sub-attribute. 41% was the weight of Reusability for having sub-attributes and 26% for doubling up as a sub-attribute. Functionality also has a weight of 57%

for being mostly addressed and a weight of 10% for doubling up as a sub-attribute. 60% was also the weight of Availability for doubling up as a sub-attribute and 17% for being mostly addressed. Cost has a weight of 77% for doubling up as a sub-attribute and a weight of 11% for having sub-attributes. Finally, Security is seen to have 56% for doubling up as a sub-attribute and 9% for having sub-attributes.

VI. Conclusion and Future Work

The paper uses a multi-criteria decision-making analysis based on the expert's evaluation and the use of the Analytic Hierarchy Process (AHP) to rank software quality attributes. A hierarchical model is presented for the AHP process. The results show the criteria weight of Maintainability to be 17.37%, Testability to be 13.02%, Reliability to be 10.35%, Efficiency to be 7.49%, Usability to be 7.22%, Portability to be 7.13%, Reusability to be 6.86%, Security to be 13.61%, Functionality to be 6.22%, Availability to be 5.99% and Cost Co(s) to be 4.73%. Maintainability is therefore the most important quality attribute followed by Security, Testability, Reliability, Efficiency, Usability, Portability, Reusability, Functionality, Availability and Cost.

The future work will include the integration of AHP with Linear Programming (LP) to select the most important software quality attribute among several attributes. The criteria weights produced from the AHP technique will serve as function coefficients in LP to build a linear model. Sensitivity analysis will also be performed to check changes in criteria weight and effect on the attributes.

References

[1] I. Gambo, A. Soriyan, and P. Achimugu, (2011) “Software Architecture Performance Quality Model : Qualitative Approach”, ARPN Journal of Systems and Software, Vol. 1, No. 1, pp. 28-33.

[2] S. Slaughter, D. E. Harter, and S. A Slaughter, (2000), “Process Maturity and Software Quality : A Field Study”, International Conference on Information Systems, pp. 407 - 411.

[3] N. B. Kassie, J. and Singh, (2020) “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.

[4] M. G. Siavvas, K. C. Chatzidimitriou, and A. L. Symeonidis, (2017), “QATCH - An adaptive framework for software product quality assessment”, Expert Systems With Applications, 43 pp.

[5] M. K. Sharma, (2017), “A study of SDLC to develop well-engineered software”, International Journal of Advanced Research in Computer Science, Vol. 8, No. 3, pp. 520–523.

[6] J. A. Mccall, and P. K. Richards, (1977) “Factors in Software Quality: Concept and Definitions of Software Quality”, Final Technical Report, Vol. 1, 168 pp.

[7] B. W. Boehm, J. R. Brown, H. Kaspar, M. Lipow, G. McLeod, and M. Merritt, (1978) “Characteristics of Software Quality”, North-Holland Publishing Company, 169 pp.

[8] R. S. Jamwal, and D. Jamwal, (2009), “Issues and Factors For Evaluation of Software Quality Models”, Proceedings of the 3rd National Conference, pp. 1 - 6.

[9] R. B. Grady, (1992), Practical Software Metrics for Project Management and Process Improvement, Prentice Hall Inc., Englewood Cliffs, NJ, USA, 270 pp.

[10] G. R. Dromey, (1995), “A Model for Software Product Quality”, IEEE Transaction on Software Engineering, Vol. 21, No. 2.

[11] ISO/IEC 9126-1 (2001), “Software Engineering - Product Quality - Part 1: Quality Model”, Vol. 1, International Organization for Standardization: Geneva, 25 pp.

[12] O. Kononenko, O. Baysal, and M. W. Godfrey, (2016), “Code Review Quality : How Developers See It”, IEEE/ACM 38th IEEE International Conference on Software Engineering, pp. 1028 - 1038.

[13] S. Parthasarathy, C. Sridharan, T. Chandrakumar, and S. Sridevi, (2020) “Quality Assessment of Standard and Customized COTS Products”, International Journal of Information Technology Project Management, Vol. 11, No. 3, pp. 1–13.

[14] T. A. Alrawashdeh, M. Muhairat, and A. Althunibat, (2013) “Evaluating the Quality of Software in ERP Systems Using the ISO 9126 Model”, International Journal of Ambient Systems and Applications (IJASA), Vol. 1, No.1, 9 pp.

[15] A. Kabir, M. U. Rehman, and S. I. Majumdar, (2016) “An Analytical and Comparative Study of Software Usability Quality Factors”. 7th IEEE International Conference on Software Engineering and Service .

[16] B. B. Rohila, and A. G. Dinker, (2014) “An Approach for Enhance the Software Quality Based on Quality Model”, International Journal of Emerging Technology and Advanced Engineering, Vol. 4, No. 1, pp. 356–361.

[17] S. Verma, and M. K. Mehlawat, (2017) “Multi-criteria optimization model integrated with AHP for evaluation and selection of COTS components”, A Journal of Mathematical Programming and Operations Research, pp. 1–16.

[18] S. K. Dubey, and A. Mishra, (2014) “Fuzzy Qualitative Evaluation of Reliability of Object-Oriented Software System”, IEEE International Conference on Advances in Engineering & Technology Research (ICAETR), pp. 1–5.

[19] F. Febrero, M. A. Moraga, and C. Calero, (2017) “Software Reliability as User Perception Application of the Fuzzy Analytic Hierarchy Process to Software Reliability Analysis”, IEEE International Conference on Software Quality, Reliability and Security, pp. 1–8.

[20] T. L. Saaty, (1977), “A Scaling Method for Priorities in Hierarchical Structures”, Journal Mathematical Psychology, pp. 234–281.

[21] B. D. Khwanruthai, (2012) “How to do AHP analysis in Excel The Analytical Hierarchy Process - AHP”, 21pp.

[22] O. Taylan, A. O. Bafail, R. M. S. Abdulaal, and M. R. Kabil, (2014) “Construction Projects Selection And Risk Assessment By Fuzzy AHP And Fuzzy TOPSIS Methodologies”, Applied Soft Computing Journal, Elsevier, pp. 1–12.

[23] K. Teknomo, (2006) “Analytic Hierarchy Process (Ahp) Tutorial” http://people.revoledu.com/kardi/tutorial/ahp/, Accessed: 15 February 2020.

[24] T. L. Saaty, (2003), “Decision Making with the Analytic Hierarchy Process”, International Journal Services Sciences, Vol. 1, No. 1, pp. 83–98.

[25] S. Zafar, M. Mehboob, A. Naveed, and B. Malik, (2013), “Security quality model: an extension of Dromey’s model”, Software Quality Journal, Issue 1, Vol. 23, pp. 29-54.

[26] T. S. Alanazi, M. Akour, M. Anbar, and A. Alsadoun, (2019), “Enterprise Resource Planning Quality Model ERPQM”, 2019 First International Conference of Intelligent Computing and Engineering (ICOICE), IEEE, pp. 1–5.

[27] J. A. Al-Nawaiseh, Y. Helmy, and E. Khalil, (2020), “A New Software Quality Model For Academic Information Systems: Case Study E-Learning System”, International Journal of Scientific and Technology Research, Vol. 9, No. 1, pp. 271–282.

[28] S. Mahmudova, and Z. Jabrailova, (2020), “Development of an algorithm using the AHP method for selecting software according to its functionality”, Journal of Soft Computing, 8 pp.

[29] M. Yujun, L. Lingli, D. Shuaijun, and L. Guofeng, (2019), “AHP-Based Software Quality Risk Assessment Method for Information System”, Scientific Conference on Network, Power Systems and Computing (NPSC), pp. 189–193.

[30] M. Toapanta, J. Nazareno, R. Tingo, F. Mendoza, A. Orizaga, and E. Mafla, (2018), “Analysis of the Appropriate Security Models to Apply in a Distributed Architecture”. IOP Conference Series: Materials Science and Engineering, 5 pp.