Received: 23 June 2020; **Revised:** 11 Dec 2021; **Accepted:** 18 Dec 2021; **Published:** 26 Dec 2021
Journal Name [Online ISSN Coming Soon], Volume 2, Issue 1, Article 12, Pages 127-154, December 2021

Digital Object Identifier 10.1111/RpJC.2020.DOINumber

A Review on Models for Software Quality Enhancement from User's Perspective

Amna Ikram[1], Isma Masood [2], Tahira Sarfraz[3], and Tehmina Amjad[4]

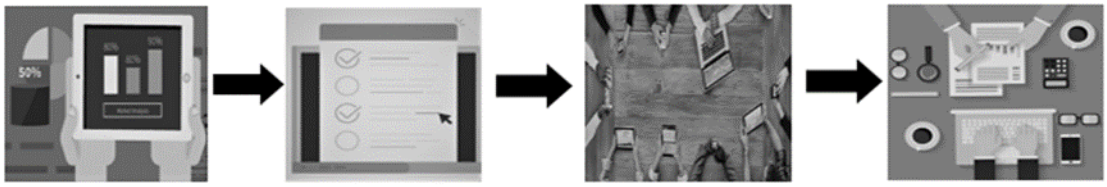
[1,2,3]Department of Computer Science, Shaheed Zulfikar Ali Bhutto Institute of Science and Technology, Islamabad, Pakistan
[4]Department of Computer Science and Software Engineering, International Islamic University, Islamabad, Pakistan
[2]Nanjing University of Science and Technology, Nanjing, China

ABSTRACT Nowadays industries are more determined to improve the quality of software products. The concentration of researchers is more towards internal quality enhancement. Less attention has been given to the quality's definition from the user's perspective. The user wants to meet his goal with satisfaction that depends upon the usability of the software product. The progress of software organizations relies on the satisfaction of the user. Our focus is to explore the software model of quality factors from the user's view. In this study, various existing software models that are proposed by different researchers for quality are reviewed and discussed. The comparative analysis of the quality attributes of these models is represented. We selected the benchmark of the ISO 9126 model for surveying different university students to rank the quality attitudes from the user's view. We have proposed a user's perspective quality model based on survey results. The proposed quality model is made up of quality factors with their respected sub factors (1) Functionality with accurateness, interoperability, and compliance (2) Reliability with recoverability, maturity, and fault tolerance (3) Usability with clarity, easy to operate and practicable (4) Efficiency with resource behavior and execution efficiency (5) Maintainability with simplicity, changeability and testability and (6) Portability with installability, coexistence, and replaceability. The future challenges related to our research area are mentioned at the end.

Keywords Supportability, Usability, Software Quality, Reliability, Functionality, Performance, McCall Model, Dromey Model, International Standard of Organization, Efficiency, Maintainability, Boehm Model, Portability, FURPS.

I. INTRODUCTION

Different studies have determined that 'Quality' can be defined differently according to the different perceptions of researchers. The software quality of products can be determined by measuring how well the software performs the required services of the user. The characteristics of quality are REL, USA, EFF, MAIN, and PRO, etc. The availability of functionality is not enough as that functionality should be usable and understandable by the user. Software Quality Assurance checks on the software quality. SQA manages all quality assurance activities of a company. The errors in the software product can be minimized through metrics and techniques like data mining techniques. The aim is to develop a highly maintainable and flexible software product in which quality is entirely attained. For some systems like Real-time and control systems, it is very important to fulfill all the quality parameters without any compromise at any state. Low quality may lead to software failure, unsatisfied clients, or financial loss. Quality identifies the user's needs by measuring the attributes through metrics. The existing quality models consist of quality attributes for software products. The motivation for this study is to perform a comparative analysis of the existing software models and select the best benchmark of quality factors to evaluate the quality based on the user's perspective. In this paper, we evaluate the quality factors to propose a new quality model based on the user's perspective. However, more studies have been done for the improvement of internal quality as compared to quality improvement according to the user's view [1]. Thus, the major goal of this study is to explore the quality attributes valuable for the users. As in the field of software development, it is essential to meet the user's needs to achieve their satisfaction. Nigussu Bitew Kassie et al. [1] performed an analysis on different exiting models of quality and proposed a quality model from the user's perspective. Software quality factors are well defined in their paper. Brijendra Singh et al. [2] presented a model consisting of REL, USA, EFF, MAIN, and PRO for direct specification and quality prediction of the software product.



Comparative Analysis of existing Quality models

Google form of Questionnaire for Software's evaluation

Users performing Evaluation of Softwares

Evaluation of Survey Results to propose model

FIGURE 1. Whole Process of Software Quality Evaluation

They proved through experiments that the developed program having tacit knowledge led to the level of quality. While Kazuhiro Esaki et al. [5] proposed a model ISO/IEC9126 consisting of FUN, REL, USA, EFF, MAIN, and PRO to evaluate the customer's satisfaction with the software quality. Dr. Deepshikha Jamwal et al. [7] analyzed the different models of quality i-e McC model, BHM model, DROM model, FURPS model, and ISO 9126 model. While Durgesh Samadhiya et al. [8] compared these software quality models to point out their key features. Several significant-quality factors show the quality level of software has been found in the literature. J. P. Miguel et al in [9] describes the main models to point out strengths and deficiencies. Also concluded that there are very general models such as FURPS, Boehm and Dromey discussed for achieving the quality and hence it becomes tough to choose them for some particular cases. Al-Qutaish and R.E in [10] performed an analytical and comparative analysis on the models that are McC model, BHM model, DROM model, FURPS model, and ISO 9126 model. After analysis, they concluded that ISO 9126 model covers the most quality factors. While A. Abran, et al in [23] evaluated and performed a survey of ISO usability standards plus evaluation methods such as ISO 9241-11 and 9126, along with this a proposal is also presented for integration into a consolidated model. Hence, the major contributions of the study are given below:

1. The study of the different exiting models includes the McC model, BHM model, DROM model, FURPS model, and ISO 9126 model to perform the comparative analysis of them based on quality factors.

2. We selected the benchmark of the ISO 9126 model as it supports all the approaches of quality for software process [8] for the questionnaire to explore the characteristics important to users. Each question will reflect one software quality. For this reason, we choose three software to evaluate their quality i-e. MS Word, MyEclipse, and Google Chrome. Its process is described in Fig 1.

3. Survey results helped in the evaluation of quality factors which are important from the user's view. After gathering and analysis of the survey's results, we identified the most critical quality factors and presented our software quality model based on the user's perspective.

4. This main goal is to explore the attributes from the user's perspective and help organizations to deliver high-quality software products.

5. Highlighting the future directions for this study.

Later, in this paper, section 2 defines the basic terminologies, section 3 presents the problem statement, section 4 performs a comparative analysis of existing models, section 5 describes the research methodology, section 6 presents the questionnaire's results, the result's discussion represented by section 7 and finally section 8 presents the future work and Section 9 presents the conclusion of the paper. Table 1 presents the abbreviations used in this study.

II. BASICS CONCEPTS AND TERMINOLOGIES

The important factors of software quality are discussed here.

1) *FUNCTIONALITY*

The basic services and their stated properties are defined in functionality. It refers to the aspects that the system does or should provide to the user.

2) *USABILITY*

The software is available for the user's needs at the required time.

3) *RELIABILITY*

The capability of a system to provide the user's required function without error.

4) *PORTABILITY*

The capability of software to run in different environments with different resources like various Operating Systems or hardware.

5) *MAINTAINABILITY*

The ability of software to change without causing any problem. There should be fewer dependencies in the system so that changing one module won't affect the rest of the system.

6) *EFFICIENCY*

Efficiency relates to the system's performance. The time required by the system to complete the tasks is measured by it.

7) *AVAILABILITY*

The ability of the system to work properly without failure when required by the user.

8) *UNDERSTANDABILITY*

The system should not only provide the functionality, but it should be understandable to the user.

III. PROBLEM STATEMENT

To the best of our knowledge, more research has been done for the improvement of internal quality as compared to quality improvement according to the user's perception [1]. This study aims to explore the quality attributes valuable for the users and form a user's perspective model. This would help the developers to provide high-quality products.

The research questions of our paper:

1) What are the quality attributes that are important to users?
2) Are they achievable or easy to implement to improve quality?
3) How can the attributes help users to satisfy them?

IV. COMPARATIVE ANALYSIS

The comparative analysis of exiting software quality models is shown in Table 2.

TABLE I

Sr. No	Abbreviations	Descriptions	Sr. No	Abbreviations	Descriptions
1	*SDLC*	Software development Life Cycle	6	*SA,A,N,SD,D*	Strongly Agree, Agree, Neutral, Strongly Disagree, Disagree
2	*ISO*	International Standard of Organization			
3	*SQA*	Software Quality Assurance	7	*MCS*	Mission Operation System
4	*AHP*	Analytic Hierarchy Process	8	*RAD*	Rapid application development
5	*SA*	Software Quality	9	*BBN*	Baysian Belief Nets

TABLE II
COMPARATIVE ANALYSIS OF EXISTING SOFTWARE QUALITY MODELS

Quality Attributes	McCall's model	Boehm Model	Dromey Model	FURPS Model	ISO 9126 Model
Functionality			☑	☑	☑
Efficiency	☑	☑	☑		☑
Usability	☑		☑	☑	☑
Performance				☑	
Reliability	☑	☑	☑	☑	☑
Portability	☑	☑	☑		☑
Maintainability	☑		☑		☑

V. RESEARCH METHODOLOGY

The research methodology of this study includes designing a questionnaire, managing a survey, and identification of quality factors from the survey results. This process is shown through a block diagram in Fig 2.

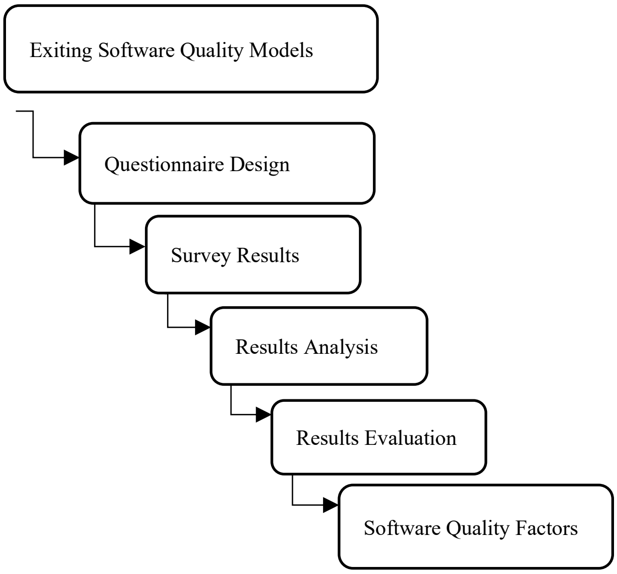


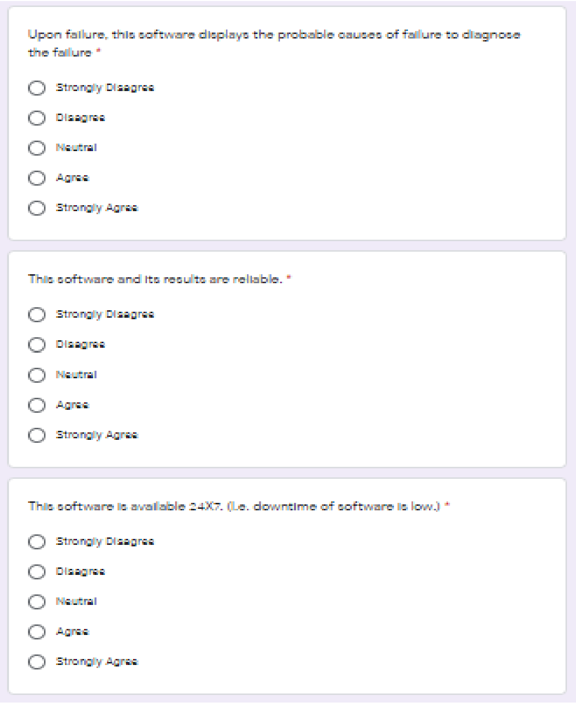
FIGURE 2. Identification of Software Quality Factors

A. DATA COLLECTION

The google form was designed which is a web-based process to collect views extensively. The designed questionnaire contained questions for the three software's that we have selected for this study including Google Chrome, MS Word, and MyEclipse. We have selected Google Chrome as it is the most commonly used tool of several internet users. University students use MS Word to perform most of the tasks like assignments. MyEclipse is universally used by professional programmers and students. Although selected software's are different from each other they are commonly known platforms. We also wanted to perform the quality evaluation of different software on the basis of attributes. This selection would give more understanding about different quality attributes.

Fig 3 shows the questionnaire. The available choices for the participants were 'strongly disagree (SD)', 'disagree (D)', 'neutral (N)', 'agree (A)', and 'strongly agree (SA)'. The details of the participants are shown in Table 3. We have found open-source datasets of quality attributes from different sites like Kaggle, Github, or Promise software engineering repository. But our focus of the study was to enhance the software product quality from the user's perspective by using the best existing software model of quality. So we selected the quality factors of the ISO 9126 model for the questionnaire as it supports all the perspectives of software quality [7] [8]. Each question will reflect one quality attribute.

Sex	Male	23.7%
	Female	76.3%
Age Group	19-30 Yrs	94.7%
	31-45 Yrs	5.3%
	46-60 Yrs	0%
	Above 60	0%
Academic/ Professional Status	Ph.D. student	0%
	Graduate Student	65.8%
	Undergraduate Student	8.1%
	Faculty/Staff	5%
	Other	21.1%
User Type	Novice User	44.7%
	Moderator	34.2%
	Power	21.1%


Upon failure, this software displays the probable causes of failure to diagnose the failure *

○ Strongly Disagree
○ Disagree
○ Neutral
○ Agree
○ Strongly Agree

This software and its results are reliable. *

○ Strongly Disagree
○ Disagree
○ Neutral
○ Agree
○ Strongly Agree

This software is available 24X7. (i.e. downtime of software is low.) *

○ Strongly Disagree
○ Disagree
○ Neutral
○ Agree
○ Strongly Agree

FIGURE 3. Identification of Software Quality Factors

VI. RESULTS

Table 4 shows the abbreviation of the quality attributes that have been used in the paper, as we have selected ISO 9126 benchmark so these quality characteristics were chosen to evaluate and analyze in this study. Table 5 shows the result of the software quality factors and percentages of MS Word software that have been acquired by the questionnaire. MS Word software was selected as it is mostly used by all types of users. The table shows six quality attributes including FUN, REL, USA, EFF, MAIN, and PRO while the rows show the percentages of 'SA, A, N, SD, D'. Results were analyzed and percentages were obtained after collecting data from different users.

Table 6 shows the result of the software quality factors and percentages of MyEclipse software that have been acquired by the questionnaire. We have selected MyEclipse software as it is widely used by computer science students as it is a very productive tool and has time-saving features. This software provides Java developers to produce efficient code and create quality achieving applications by using the latest Java Enterprise Edition technologies. The table shows six quality attributes including FUN, REL, USA, EFF, MAIN, and PRO while the rows show the percentages of 'SA, A, N, SD, D'. Results were analyzed and percentages were obtained after collecting data from different users.

Table 7 shows the result of the software quality factors and percentages of Google Chrome as it is the fastest web browser and is used by everyone these days. It is very basic, simply designed, and easy to use. The second reason for selecting Google Chrome is that the participants do not need more information on this before trying out the designed questionnaire. The table shows six quality attributes are FUN, REL, USA, EFF, MAIN, and PRO while the rows show the percentages of 'SA, A, N, SD, D'.

Results were analyzed and percentages were obtained after collecting data from different users. For the identification of significant factors that may have an impact on the software quality, a statistical analysis was performed. Results showed that FUN, REL, USA, EFF, MAIN, and PRO were the attributes that affect more on the software quality and browser. A percentage of these quality attributes were built against software quality factors and with this percentage, the most critical attributes have been identified. By using these quality attributes, we present the model based on the user's perception in Fig 7.

TABLE IV

Abbreviation of QA

FUN	Functionality
REL	Reliability
USA	Usability
EFF	Efficiency
MAIN	Maintainability
PRO	Portability

TABLE V

Results of SQF & Percentage in MS Word

No	1	2	3	4	5	6
QA	FUN	REL	USA	EFF	MAIN	PRO
SA	19.75 %	14.25 %	30.25 %	13.15 %	17.15 %	17.1 %
A	68.4 %	60.55 %	63.15 %	52.65 %	56.0 %	55.25 %
N	11.85 %	20 %	5.3 %	22.4 %	18.45 %	17.15 %
D	0%	5.25 %	1.3 %	10.5 %	6.6 %	5.25 %
SD	0 %	0%	0%	1.3 %	1.8 %	5.25 %
Total	100%	100%	100%	100%	100%	100%

TABLE VI

Results of SQF & Percentage in MyEclipse

No	1	2	3	4	5	6
QA	FUN	REL	USA	EFF	MAIN	PRO
SA	10.55 %	10.4 %	14.03 %	14.45 %	14.5 %	11.85 %
A	57.9 %	57.9 %	51.73 %	53.95 %	55.3 %	52.65 %
N	30.25 %	25.15 %	30.7 %	27.2 %	23.65 %	32.9 %
D	1.3 %	6.55 %	3.5 %	3.95 %	5.25 %	1.3 %
SD	0%	0%	0%	0.5 %	1.3 %	1.3 %
Total	100%	100%	100%	100 %	100%	100%

TABLE VII

Results of SQF & Percentage in Google Chrome

No	1	2	3	4	5	6
QA	FUN	REL	USA	EFF	MAIN	PRO
SA	19.75 %	21.05 %	22.4 %	22.4 %	18.4 %	21.05 %
A	61.85 %	55.25 %	56.6 %	61.85 %	68.4 %	64.45 %
N	15.8 %	19.75 %	17.05 %	13.15 %	5.3 %	9.25 %
D	1.3 %	2.8 %	3.95 %	0%	7.9 %	3.95 %
SD	1.3 %	1.3 %	0%	2.6 %	0%	1.3 %
Total	100%	100%	100%	100%	100%	100%

VII. DISCUSSION

Fig 4 shows the quality rated by each participant about the MS Word software. In the last part of the question of MS Word, the participants were asked to rate MS Word software's quality overall without setting any specific attribute, out of all the participants 57.9% participants rate MS Word quality as high quality which means it is used widely around the globe, 23.7% of the participants rate the software as very high quality, 7.9% of the participants rate the software as low quality which is a very less ratio and shows that there are very few users who don't like the quality of the software. None (0%) of the participants rate the quality as very low quality whereas 10.5% of the participants state that the quality of MS Word software is average which means not very high nor very low.

Fig 5 shows the quality rated by each participant about the MyEclipse Software. After providing all questions related to quality attributes of the ISO 9126 model the last part was to ask the participants to rate MyEclipse software's quality overall without setting any specific attribute, out of all the MyEclipse quality as high-quality software that means it is mostly used by the programmers that code using Java Language, 18.4% of the participants rate the software as very high quality, 7.9% of the participants rate the software as low quality which is a very less ratio and shows that there are only 2 participants 44.7 % participants rate participants who don't like the quality of the software. 5.3% of the quality participants rated as very low quality whereas 26.3% of the participants state that the quality of MyEclipse software is average which means not very high nor very low.

Fig 6 shows the quality rated by each participant about the Google Chrome browser. After providing all questions related to quality attributes of the ISO 9126 model the last part was to ask the participants to rate browser Google Chrome quality overall without setting any specific attribute, out of all the participants 42.1 % participants rate Google Chrome's quality as the high quality that means it is the browser that is mostly by the participants, 28.9% of the participants rate the browser as very high quality, 7.9% of the participants rate the software as low quality which is a very less ratio and shows that there are only 3 participants who don't like the quality of the browser. 5.3% of the participants rated as very low quality whereas 15.8% of the participants state that the quality of the Google Chrome browser is average which means not very high nor very low. The user's perspective model is presented in Fig 7 is obtained from all the results of the evaluation.

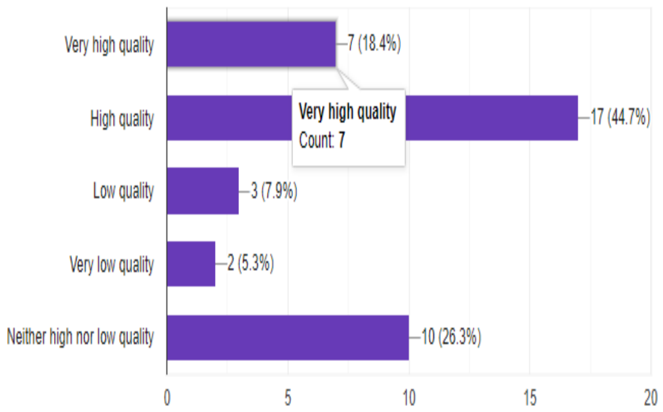


FIGURE 5. Quality Rated by Users for MyEclipse

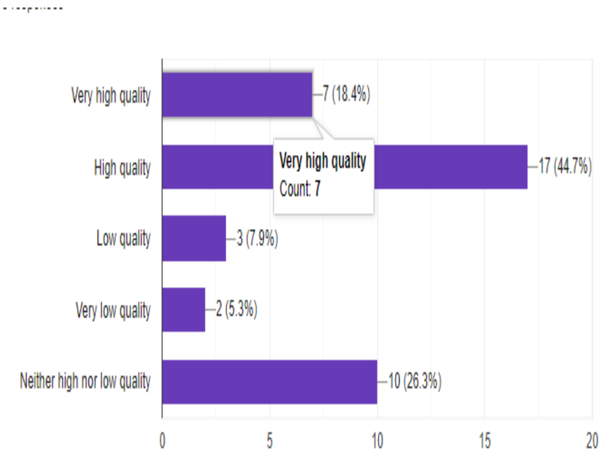


FIGURE 6. Quality Rated by Users for Google Chrome

We proposed a user's perspective model to enhance the product quality by analyzing the survey results. This quality model consists of quality factors with their respected sub factors. The value of quality attributes was calculated in percentage and starting from high to low values (1) 79.4% of Functionality with accurateness, interoperability, and compliance (2) 78.3% of Usability with clarity, easy to operate and practicable (3) 76.5% of Maintainability with simplicity, changeability, and testability (4) 74.1% of Portability with installability, coexistence, and replaceability (5) 73.1% of Reliability with maturity, fault tolerance and recoverability and (6) 72.8% of Efficiency with resource behavior and execution efficiency. These sub-factors were retrieved from result data analysis.

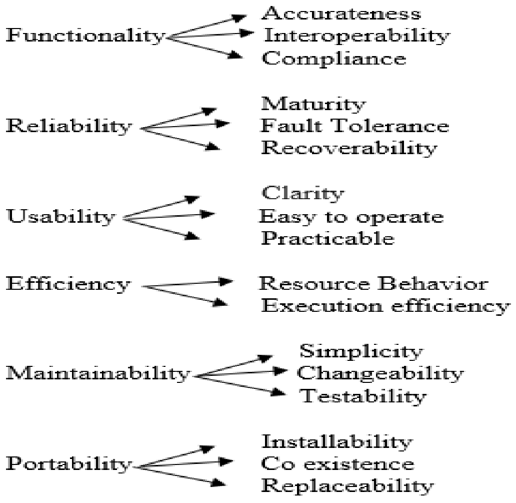


FIGURE 7. User's Perceptive Software Quality Model (Most Critical)

In 2004, A. Abran, et al in [23] stated there has been a vast change in the development of software applications with time. The distributed application growth especially for technical staff and developers resulted in having no direct access to software systems for the end-users. The usability of software is not a luxury it is a basic need of software to meet so that the user can be satisfied. To meet the usability of the software and implement it, there should be basic knowledge available for software systems and if not available then it becomes a very challenging task in designing software systems. Usability ISO 9241-11 model was proposed by a certain group of Human-Computer Interaction experts whereas the usability

ISO 9126 model was proposed by Software Engineering experts. This means that usability models that are currently used in the industry are not mature. Some weakness in usability ISO 9126 has some weakness such as at the detail level of measure the architecture is unclear, overlapping of some concepts, lack of standard of quality requirement, ambiguous measures, and some quality model descriptions are static. Some of the weaknesses in usability ISO 9241-11 are that it identifies usability strictly from the only process perspective, hence tackles only a single point of view, learnability characteristic is not as tackled as it is recommended and also does not tackle the security aspect. The two characteristics learnability and security are added as a baseline. The enhanced model is referred to as the Consolidated Usability Model, the characteristics of the usability model include effectiveness, efficiency, learnability, and security (Table 8).

In 2005, H. Al-Kilidar et al, in [28] states ISO/IEC 9126 can be utilized to access the nature and design reports and documents of good quality. The experience of the ISO/IEC 9126 model's observer indicated that the model is clear enough to become appropriate or sufficient for each software product. In this manner, it isn't sufficiently explicit to be valuable for any specific plan item. The problems with ISO/IEC 9126 are that the completeness and coverage of functional implementation, overlapping of concept's definition that may cause many metrics counting, and reliability and maintainability recognized as quality attributes by standards. However, it may not be considered while designing structure as the standard overlooks different attributes that may be significant in design products. For example, modularity and validity as simple counts are not enough for the evaluation of structure quality. Sometimes measures need data that can't be accessed by the designers, for example, functional understandability. Sometimes measures need non-accessible counting items from the design document. No rules or techniques are characterized for gathering the metrics into a general assessment and evaluation. Table 9 represents the summary of this research paper.

TABLE VIII
Summary of ISO Usability Model

Ref. No.	[23]
Year	2004
Main Idea	Evaluation and survey of ISO usability standards and evaluation methods such as ISO 9241-11 and 9126, along with this proposal is also presented for integration into a consolidated model.
Quality Models	ISO 9241-11 and 9126.
Quality Factors	Usability.
Findings	An enhanced usability model is presented that includes both product-related and process-related usability characteristics and uses ISO 9241-11 as a baseline and then integrates ISO 9126 for the enhanced model.
Limitations	In this paper some characteristics are still missing that are important to evaluate the usability in ISO models such a satisfaction.

TABLE IX
Summary of ISO Usability Model

Ref. No.	[28]
Year	2005
Main Idea	Utilization of ISO/IEC 9126 to document software design through an experiment.
Quality Models	ISO/IEC 9126.
Quality Factors	Usability.
Findings	In the current organization, ISO/IEC 9126 neglects to accomplish any of the mentioned objectives.

Limitations	The problems with ISO/IEC 9126 that are presented in the paper are some idea definitions are ambiguous. For example functional consistence and some idea definitions overlap.

In 2007, I. Heitlager, et al in [24], to check the maintainability of the software and the effort needed to it is linked to the technical quality of code for the software system. The ISO 9126 model doesn't have a related set of sequences for analyzing and estimating the maintainability of the code and on another side, a Maintainability Index (MI) has also been suggested that calculates that specific number that states maintainability of the system. In this research paper, the ISO model's characteristic maintainability has been discussed and various problems with the maintainability index have been identified along with the number of requirements that should be met by the maintainability model have been discussed. A new model of maintainability is sketched to solve these problems. Source code properties such as volume only map with analyzability, complexity per unit property maps with changeability and testability, duplication property maps with analyzability and changeability, unit size property maps with analyzability and testability, and the unit testing property maps with analyzability, stability, and testability. The future work suggested by this paper is this approach should be compared with rating schemas captured by the Baysian Belief Nets (BBN). The details are given in Table 10.

TABLE X

SUMMARY OF ISO MODEL FOR MAINTAINABILITY

Ref. No.	[24]
Year	2007
Main Idea	Maintainability Index (MI) has been suggested that calculates that specific number that states maintainability of the system.
Quality Models	ISO 9126 model.
Quality Factors	Sub-factors of Maintainability i.e. Analyzability, Changeability, Testability, and Stability.
Findings	The source code of the software must meet the maintainability sub-characteristics such as analyzability, changeability, stability, and testability.
Limitations	The identified maturity index limitations are regarding analysis of root-cause, language-independent, ease of computation, explainability, understandability, and control.

In 2009, Che-Wei CHANG et al. [4] the importance of the quality of software products is increasing quickly in the computerized era. According to this paper, quality can be measured from the user's perception that how much software is providing the services that they need. They used expert decision-making algorithm-based method for the quality evaluation in the quantitative model which was the analytic hierarchy process (AHP). Multi-attribute decision problems can be structured with it. They allow the twenty-one decision-makers to identify the best quality software provided by the four firms. They established the Pair-wise comparison matrix and determine Consistency. On the basis of high-level inconsistency in making judgments, they exclude some decision-makers. Then find the eigenvectors (weighs) of the level of the model to determine quality. These levels are criteria and sub-criteria of the model included quality factors and sun factor given in Table 11. This model evaluates the quality of software plus more suitable for academic and government purposes.

In 2009, J. P. Correia, et al in [27] depending upon the standard of ISO/IEC 9126, a group of Software development has designed a practical approach to measure the real quality of the software product. This specific model of quality presents other levels that fall below in hierarchy defined and explained by ISO/IEC 9126 that contains software system characteristics like unit complexity, volume, and others. The mapping characteristics of the system and IEC/ISO 9126 properties and sub-characteristics are explained in a binary form: this property moreover affects the characteristics. This specific mapping is done by mutual agreement among three experts, casually depending on the experience obtained by assessment of software quality. The survey-based experiment performed in this paper to study mapping between the system software properties and the main quality characteristics. The hierarchy procedure is used in form of a formally organized method that causes the reputation of system software quality characteristics and system properties. The results of the experiments are analyzed by setting two objectives first is to verify the native binary mapping and the second is to improve the mapping stage. Finally, the result after the experiments proves duplication property maps with stability and changeability, unit size property maps with analyzability, unit interfacing results with a decrease in weight, and the unit testing quality property maps with stability and testability, inward coupling maps with stability, outward coupling results with a decrease in weighing and exception handling only handles stability. Refer to Table 12 for the detail of the results.

TABLE XI
Summary of Evaluation of Quality

Ref. No.	[4]
Year	2009
Main Idea	Control and measure the software product quality.
Quality Models	They proposed a quantitative model.
Quality Factors	It includes Functionality, Reliability, Usability, Efficiency, Maintainability, Portability, Suitability, Accuracy, Security, Maturity, Fault tolerance, Recoverability, Stability, Interoperability, Understandability, Learnability, Operability, Time behavior, Attractiveness, Coexistence, resource behavior, Analyzability, Changeability, Testability, Adaptability, Installability, and Replaceability.
Findings	This model performs the quality evaluation of software more effectively from the user's perspective.
Limitations	This study is limited to the analytic hierarchy process.

In 2010, Jamwal et al. [7] Quality is now considered to be a key element in software development. Quality contains the attributes and important characteristics of a product that satisfies the given requirements. McC model defines quality from three perspectives i.e. Product operation, Product revision, and Product transition. ISO 9126 model assesses the software quality in form of internal, external software characteristics and their relationship to attributes. BHM model defines the general utility containing different quality characteristics which define several criteria hierarchy structures. DROM model is the framework of three quality models i-e Requirement, Design, and Implementation. FURPS model is made up of FUNC, USA, REL, PER, and SUPP. FURPS model fails to determine the product's Portability. Sub factors of software evolvability are not addressed explicitly in the McC model and DROM model. McC model, BHM model, DROM model, and FURPS model covered no Architectural integrity. McC model does not explicitly include analyzability but is partially addressed in the BHM model, DROM model. In the McC model, DROM model, and FURPS model, domain-specific factors are not addressed explicitly. The characteristics in the BHM model, DROM model don't define the ability to examine the effect of change stimulus at the level of software architecture. ISO 9126-1 model covers all the perspectives of quality as mentioned in the details in Table 13. In 2010, Durgesh

Samadhiya et al. [8] there are several quality models which are proposed in software engineering. The well-known McC quality model focused on the quality attributes which reflect the user's and developer's needs. BHM model defines the general utility of characteristics which defines several criteria in hierarchy structure. DROM model focus on the association characteristics. FURPS model is made up of FUNC, USA, REL, PER, and SUPP. ISO 9126 model assesses the software quality in form of internal, external software characteristics and their relationship to attributes. ISO 9126 model and McC model support quality attribute more than BHM, FURPS, and DROM models. "Reliability" is the common attribute in models. FUNC, REL, USA, EFF, MAIN, and PRO are the most common quality attributes. They manufacture a full image for software quality in the McC model. It is tough to specify the quality requirement by using it. FURPS model can't determine the product's Portability. BHM model provides basic help from top to bottom in a hierarchy to software quality but this help is temporary. DROM model identifies quality evaluation which varies from product to product. ISO 9126 quality model supports all the perspectives of software quality. Table 14 represents its details.

TABLE XII

Summary of a Study for Mapping System Properties to ISO 9126 Maintainability

Ref. No.	[27]
Year	2009
Main Idea	A Survey is performed to map system properties to ISO/IEC 9126 for maintainability.
Quality Models	ISO/IEC 9126.
Quality Factors	Maintainability.
Findings	Finally the result after the experiments proves an increase in the mapping stage and shows that volume only maps with analyzability and complexity per unit property maps with changeability.

TABLE XIII

SUMMARY OF ANALYSIS OF SOFTWARE QUALITY MODELS

Ref. No.	[7]
Year	2010
Main Idea	They analyzed the different software quality models consisting of characteristics that reflect the product quality.
Quality Models	McCall's model, Boehm's model, Dromey's model, FURPS model, and ISO 9126 model.
Quality Factors	Functionality, Reliability, Usability, Performance, Supportability, Maintainability, Efficiency, Correctness, Testability, Flexibility, Interoperability, Reusability, Understandability, Integrity, Modifiability, Human Engineering, and Portability.
Findings	ISO 9126-1 quality model is more useful than others as it has been built based on an international consensus and agreement from all the country members of the ISO organization.
Limitations	They performed an analysis of the software quality models on the specific criteria's including Software evolvability, Architectural integrity, analyzability, domain-specific attributes, and change stimulus.

In 2010, Al-Qutaish and R.E in [10] state that software quality is very essential and sometimes very critical to achieve in various types of organizations. For some systems like Real-time and control systems, it is very important to fulfill all the quality parameters and no compromise is acceptable as it may lead to capital loss, permanent injury, or can also cause losing a human life and mission failure. The different quality models are proposed in software engineering. They compare the different quality models consisting of different attributes to find the key features. Choosing the best quality model according to the need of the software is quite a challenging task. In this paper, the quality model is set for discussion. From these factors, only reliability is common in all models. According to understanding, this paper has not covered all the quality attributes that are important to achieve in all quality models (Refer Table 15). In 2011, B. Tomar and Dr.V. M. Thakarein [13], Quality consists of all important properties and attributes of an activity that links to achieving the satisfaction of a given requirement. The quality attribute of a product can be a feature that meets all the requirements of the customer and thus provides product satisfaction. It keeps away from the deficiencies and can be used for fitness factors. In this paper, the model that is discussed includes Georgiadou. This paper aims to introduce the reader, novice with various quality attributes. The result will also help support the researcher in the future software quality model and manage more and more studies on techniques and methods that are used by researchers and the software industry. This paper has not covered all the quality attributes that are important to achieve in all quality models. See Table 16 for the key points of this research paper.

TABLE XIV
Summary of Analysis of Software Quality Models

Ref. No.	[8]
Year	2010
Main Idea	They compare the quality model consisting of different attributes to find the key features.
Quality Models	McCall's model, Boehm's model, Dromey's model, FURPS model, and ISO 9126 model.
Quality Factors	Functionality, Reliability, Usability, Performance, Supportability, Maintainability, Efficiency, Correctness, Testability, Flexibility, Interoperability, Reusability, Understandability, Integrity, Modifiability, Human Engineering, and Portability.
Findings	ISO 9126 quality model is good as it supports all the approaches of quality for software process.
Limitations	They compared the software quality models on the basis of their quality attributes.

In 2011, P. Sudhaman [15], Nowadays as software have become more pervasive, so there has been an increase in growing concerns of software quality in the IT industry. The different quality models help to standardize the software quality. But still, no involving consent relation exists between software quality model and process for accessing the quality of software and hence remain an open issue. This research paper aims to compute characteristics of the quality of the software-defined by ISO/IEC 9126 by using the software metrics that are being proposed in this paper. The metrics that are proposed in the paper are testing, number of the undetected functions during software system, the total number of the variations that are obtained in the results to the expected result for the set of a given input, a number of the satisfied requirement, interaction found between the number of systems and time that is spent to analyze design documentation, adaptability, the number of platforms for which software applicable for and installability.

TABLE XV
Summary of Quality Models in Software Engineering

Ref. No.	[10]
Year	2010
Main Idea	An analytical and comparative analysis is performed on the quality model and metrics.
Quality Models	McCall's model, Boehm's model, Dromey's model, FURPS model, and ISO 9126 model.
Quality Factors	Testability, efficiency, Flexibility, Functionality Maintainability, Portability, Correctness, Performance, Reliability, Integrity, Efficiency, Usability, Reusability, Human Engineering, Modifiability, Interoperability, Understandability, and Supportability.
Findings	From this paper this can be gathered that only one quality model that is McCall quality model covers most of the factors.
Limitations	In this paper the metrics of McCall's, Dromey's, Boehm's and FURPS quality models have been defined ambiguously that may confuse.

TABLE XVI
Summary of Study of Software Quality Models

Ref. No.	[13]
Year	2011
Main Idea	A systematic study is performed on the quality model.

Quality Models	Boehm quality model, FURPS, ISO 9000, McCall quality model, Dromey's quality model, CMM, and Georgiadou.
Quality Factors	Functionality, Security, Usability, Consistency, Reliability, Response Time, Testability, Recoverability, Predictability, Maintainability, Accuracy, Performance, Efficiency, Availability, Accuracy, Recovery time, Extensibility, Adaptability, Throughput, and Compatibility.
Findings	As a result only the McCall quality model achieved most of the quality attributes but there are still, some factors that are not achieved.
Limitations	As in this paper a systematic study is performed for the quality model being used so they have not covered all the quality attributes such as resource usage and supportability that are important to achieve in all quality models.

In this database different quality attributes values for specific software product will be saved and will play important role in a software project. Table 17 represents its details.

TABLE XVII

SUMMARY OF EVALUATION OF SOFTWARE QUALITY MODELS

Ref. No.	[15]
Year	2011
Main Idea	To perform the quality evaluation by using software metrics for a software system product.
Quality Models	ISO/IEC 9126 model.
Quality Factors	Number of attempts and the total number of attempts (succeeded/failed), several faults that are tolerated in software, frequency of the usage of software, required time, time is taken to perform a specific operation on the software, resource available, size of software modified.
Findings	These metrics have proven to perform well to measure the quality of the software.
Limitations	In this paper it is also proposed for future work to develop a database "software quality database SQD".

In 2012, K. Mordal et al in [14], for an organization that wants to keep control of the software project and their systems it is important to measure the software quality. As there is a various metric of software quality that is assessable to compute the aspect of quality that varies than due to this these metrics is explained at the lower level for individual parts of the system such as classes, function, and methods although the programmer wants to see the full design of the whole system. The issues have been identified including the composing metrics needed for the various ranges, the need for an aggregate of quality assessments of the components, and the need to identify the wrong result that should be corrected. A quality model is represented for concrete projects that are defined empirically in large companies to gather requirements. In this paper, the Squale model is also presented which is a software quality model that is developed with the collaboration of huge organizations so that the requirements of the software can be met. The result presents that Squale meets all of the important requirements that have been recognized with implementing decomposability as an exception. As this issue in practice happens very rare so might be unlikely to occur. Every researcher defined the term quality according to their perception in papers. See Table 18 for more details. In 2012, Brijendra Singh et al. [2] defined the quality of software products on the basis of basic components. A model is proposed in this study consists of basic components taken from quality attributes for user satisfaction. They took two programs written in different languages and used the developed software product as input for creating the understandability of the model. The developed program having tacit knowledge led to the level of quality (See Table 19).

In 2012, S. Fahmy et al, in [29] Electronic Book or digital book has become another medium in training. A significant thought of digital book for a system of education is to achieve the quality of the software that are dispatched to the software. The word 'e-Book' is vague and has been utilized conversely with other, smaller laptops and tablets to give some examples it is contended that to procure the most ideal result from the digital book, the quality of the software system utilized ought to be of 'satisfactory.

From the research, efforts are made to look into and a correlation is made to distinguish top quality 'attributes' that ought to be utilized in the assessment of the electronic book. In this paper addition of the ISO 9126 model for evaluation is proposed, ordering five quality attributes for electronic books. ISO 9126 is the most relevant and good model that suits the electronic book and is chosen for evaluation. The general ISO 9126 needs further examination, analysis, and mapping of attributes is necessary already it can be completed adjusted to the electronic book. Functionality is picked as it is substance to the main area (education) also reliability is picked as it highlights the data representation and substance [10]. Efficiency is selected as it refers to the capacity of the software item to give usable function to accomplish its point [20]. Usability is incorporated as it is a significant feature particularly in client applications [11]. Maintainability is not selected from this specific model as it tends to be just assessed by the developer. Refer to Table 20 for more information.

TABLE XVIII

SUMMARY OF SOFTWARE QUALITY METRIC AGGREGATION

Ref. No.	[14]
Year	2012
Main Idea	The practical issues with quality metrics have been identified with the existing methods for aggregation that are used in real projects.
Quality Models	Squale model.
Findings	The result shows that Squale satisfies all and most of the requirements that have been identified with performing decomposability being a notable exception.
Limitations	Limitation presented by this study is that to notice that the experiment that is performed are artificial or not in the sense that the distribution of results of quality for individual components is limited whereas in real life it could increase much more.

TABLE XIX

SUMMARY OF QUALITY MODEL FOR SOFTWARE Quality

Ref. No.	[2]
Year	2012
Main Idea	Achieve the quality level of the software product that satisfies the user.
Quality Models	They proposed a model of quality attributes.
Quality Factors	It includes Reliability, Usability, Efficiency, Maintainability, and Portability.
Findings	Direct specification and prediction of quality of software product focused by this proposed model. Also found that the program has tacit knowledge leads to quality.
Limitations	How tacit knowledge played the role in leading quality is not defined.

TABLE XX

SUMMARY OF EVALUATING QUALITY ATTRIBUTES OF MODELS FOR E-BOOK

Ref**. No.**	[29]
Year	2012
Main Idea	This paper is keen on deciding a proper model for assessing and evaluating the nature or quality of software in the electronic book.
Quality Models	Boehm, McCall, FURPS, Bayesian, ISO 9216, and Dromey.
Quality Factors	Reliability, Functionality, Efficiency, Usability, Portability, and Maintainability.

Findings	Refinement is also performed to make sure a systematic approach is selected for the evaluation of quality more deeply that including the characteristics identified.
Limitations	Maintainability is not selected from this model.

In 2013, Kazuhiro Esaki et al. [5] Quality of the product is now the main concentration for a developer. The Profitability of the product depends upon the sale which leads to customer satisfaction. The customer's satisfaction depends upon the product's quality and its price. It is important to control the needs of the quality of the client for the system to observe its suitability. The definition of its quantitative measure and inherent attributes should be provided. They used ISO 9126 model to observe the customer satisfaction for the quality of Laptop Personal Computers (LPCs). They classified the negative comments collected from the online website of Laptop Personal Computers and found the importance of those characteristics for user's satisfaction. Inherent attributes of target products are also collected to perform correlation analysis as the customer's satisfaction depends on it. They verified the validity of prediction models of quantitatively predicting total customer satisfaction by using the inherent attributes of target LPCs by multiple-regression analysis. This paper calculates the significance of customer requirements for each quality characteristic of the model. Table 21 represents the key features of this research paper.

In 2013, RK Pensionwar et al in [12], the quality of the software is mostly the common objective of many software organizations that results in helping deliver software products that fulfill all requirements and customer expectations. Nowadays in research, Software quality and software metrics are highly in demand and very important research areas. Quality models of software are the most important place from which the data can be improved and the quality of different projects can be increased. This performance increase percentage can be in form of satisfaction of customer, quality and less cost. There are various models of quality that are proposed in the field of software engineering. They compare the different quality models consisting of different attributes to find the key features. This paper discusses some of the quality models and checks the type of improvement that models need to meet the level of quality. The discussed quality models were used in most of the papers. The quality attributes are used for comparison. The quality policy of companies, the framework of quality, improved quality methods like PSP, CMMI, TSP, and SIX SIGMA were also studied along with models. The result only some quality attributes were achieved and along with this it also results that the companies lack in implementing quality policies in some projects for defining high-quality products. Table 22 contains the essential information of this research paper.

In 2013, A. Abran et al in [25] say that Portability is considered as a non-functional requirement at the software system level. The ISO model and IEEE standard provide several concepts that play an important role in describing different types of portability views and requirements at the hardware level and system level. They presented the results of portability analysis of the system for portability requirements defined in various standards from a different point of view that is several function concepts of portability. As the structure of the reference framework depends on the model of software-FUR and data needed to measure the functional size is easily available. The measurement result for a certain application would be distinct as the standard-based reference model is generic. The proposed measurement model and specification are surely independent of language and type of software. NFR derived and important details and with the capacity to take all the size that is functional of portability into the account for the functional point-based model for software estimation. See Table 23 for more details of this research paper.

TABLE XXI

SUMMARY OF QUANTITATIVE MEASUREMENT OF QUALITY

Ref. No.	[5]

Year	2013
Main Idea	Measure the customer’s satisfaction.
Quality Models	They proposed an ISO/IEC9126 model.
Quality Factors	Functionality, Reliability, Usability, Efficiency, Maintainability, and Portability.
Findings	The model predicts the extent of customer satisfaction of the target products from the inherent attributes of a product.
Limitations	Use of inherent attributes of target products of the system must be applied in the design phase of development.

TABLE XXII

SUMMARY OF SYSTEMATIC STUDY OF SOFTWARE QUALITY MODELS

Ref. No.	[12]
Year	2013
Main Idea	A systematic study is performed on the software quality model.
Quality Models	McCall’s quality model, FURPS quality model, and CMM.
Quality Factors	Flexibility, Maintainability, Correctness, Testability, Reliability, Efficiency, Usability, Integrity, Reusability, Portability, Interoperability, Supportability, Modifiability, Understandability, Performance, and Functionality.
Findings	Results show that the companies lack in implementing quality policies in some projects for defining high-quality products.
Limitations	As this paper was systematically studied for the quality model being used in companies, so this paper should have also analyzed ISO 9126 model, as this is a standard mostly adopted by most of the companies to improve their quality.

In 2014, Miguel et al in [9] stated that as Software products are increasing day by day and these products are used in all daily activities of human life and as a result measuring, analyzing, and evaluating the quality of the software product has gained importance and becomes a very difficult task in a software organization. Different techniques are proposed to solve this quality critical task. Many quality models are suggested in software engineering. They compare the different quality models consisting of different attributes to find the key features. This paper aims to present the description of main quality models and present the deficiencies and strengths. Tailored models of quality are Alvaro Model, Bertoa model, GEQUAMO, Rawashdeh Model. The quality attributes are focused that are proposed factors on which the paper focus and discussed. This paper presented that the models discussed for achieving the software quality are very general such as FURPS and therefore it becomes difficult to use and apply to very specific cases. Refer to Table 24 for this literature.

In 2014, Suman et al in [11], Software Quality is considered a key element in today’s era and is the most important attribute to achieve in Software Engineering. The objective of software engineering is to develop software that is highly maintainable, flexible, and in which quality is fully achieved. For some systems like Real-time and control systems, it is very important to fulfill all the quality parameters and no compromise is acceptable in any state. Any software that lacks quality aspect may result with failure of software, client un-satisfaction, losing a life of human and also capital loss. Quality consists of many characteristics. This is the reason why quality is captured in a model that describes these characteristics and their relationship. These models show users' points of view regarding the quality of the software. Different companies use various quality models depending on the requirement.

TABLE XXIII

SUMMARY OF REFERENCE FRAMEWORK FOR SYSTEM PORTABILITY

Ref. No.	[25]
Year	2013

Main Idea This paper presents portability concepts and characteristics into a framework for standard-based reference of the system portability requirements.

Quality Models ISO 9126, ISO 2382-1, and ISO 24765.

Quality Factors Portability.

.

Findings Portability-NFR provides an integrated review of the system portability requirement to the system engineer, a methodology to analyze this portability NFR on the basis of the ISO model standards reference framework.

Limitations To perform research on the model of estimation using the size of the software derived from system non-functional requirements as input should be investigated for the initial development and maintenance phase.

In this research paper, a comparison study of quality models is presented in which different concepts of quality characteristics of software are reviewed, analyzed and different software quality models that are used in the different organizations are also discussed. Software quality models represent quantitatively and a fixed structure of quality. There are different quality models are used for comparison. These models have been analyzed on some quality factors. According to my understanding, this paper does not provide any information about data sets and the research approach that is used for evaluation as that is very important for analyzing software quality models. Table 25 represents the details of this paper.

TABLE XXIV

SUMMARY OF EVALUATION OF SOFTWARE QUALITY MODEL

Ref. No.	[9]
Year	2014

Main Idea To review the quality models for evaluation of software production.

Quality Models McCall's model, Boehm's model, FURPS Model, Dromey's model, ISO 9126, and ISO 25010.

Quality Factors Accuracy, Adaptability, Human engineering, Analyzability, Attractiveness, Stability, Correctness, Changeability, Flexibility, Efficiency, Functionality, Usability, Reliability, Interoperability, Modifiability, Installability, Integrity, Maintainability, Suitability, Performance, Maturity, Operability, Portability, Resource utilization, Reusability, Performance, Transferability, Testability, and Understandability.

.

Findings Tailored models of quality are presented that play important role in achieving the quality of software.

Limitations In this paper no new user perspective from real industry is discussed which plays important role in achieving software quality.

In 2014, Anas Bassam AL-Badareen et al. [22] there are several models are proposed for software quality but they proposed the model of quality from the user's perspective. User satisfaction is necessary for quality. This study discussed the characteristics, sub characteristics, and proposed model from the user's view. This would help the developers to develop the software based on the user's satisfaction. This study discussed the attributes of the software product that affect the satisfaction and emotions of the user (see Table 26). In 2016, Cigdem Altin Gumussoy [17] the purpose is to provide a usability guideline to the software developer for the construction of software design of bank. They highlighted the complex points in the software design of the bank to prevent it from *a disaster. They analyzed usability problems from thrice software development projects of banking. That problem is faced by users during using it. The experts have then grouped the problems are then grouped based on severity. Each of the criteria of usability from various heuristics is scaled.*

TABLE XXV

SUMMARY OF COMPARATIVE STUDY OF QUALITY MODELS

Ref. No.	[11]
Year	2014

Main Idea Comparison of different quality models is performed to achieve the quality of software systems.

Quality Models Boehm's quality model, McCall's model, Ghezzi model, FURPS /FURPSISO Standard quality model, Capability Maturity Model, Dromey's quality model, Bansiya's, QMOOD model, Aspect-Oriented Software quality model, IEEE Model, SQuaRE's model. UML Conceptual Model, and S. K Model.

Quality Factors Accuracy, Availability, Reliability, Correctness, Efficiency, Flexibility, Ambiguity, Functionality, Human Engineering, Integrity, Interoperability, Security, Feasibility, Maintainability, Modifiability, Performance, Portability, Process Maturity, Security, Testability, Reusability, Robustness, Scalability, Understandability, and Supportability.

Findings A complete comparison chart is formed that represents the quality models which cover most of the quality factors.

Limitations Most of the quality models compared are only focusing on aspects, not the overall model which should not be included in this study such as the Aspect-Oriented software quality model.

TABLE XXVI

SUMMARY OF USER'S PERSPECTIVE MODEL

Ref. No.	[22]
Year	2014

Main Idea To satisfy the user.

Quality Models They proposed the User's Perspective Model.

Quality Factors Functionality, Reliability, Performance, Usability, Suitability, Portability, Training, Accuracy, Compliance, Integrity, Efficiency, Fault Recovery, Maturity, Coverage, Interface factors, Operation Supportability, Coexistence, Adaptability, and Setup facility.

Findings This model will help to achieve the user's satisfaction level.

Limitations All the defined characteristics belong to only users.

In 2016, César Coelho et al. [19] Software portability is the capability of the software to run in a different environment. This study proposed a NanoSat Mission Operation framework for software portability. This framework gives an on-board software framework for the space segment. It consists of high-level components, layers and is based on the Consultative Committee for Space Data Systems Mission operation which monitors and controls the

TABLE XXVII

SUMMARY OF USABILITY GUIDELINE FOR SOFTWARE DEVELOPMENT

Ref. No.	[17]
Year	2016

Main Idea To satisfy the user.

Quality Models They use heuristics and their corresponding criteria.

Quality Factors Usability.

Findings *This method reflected that the usability problems found in the previous projects don't match any of the heuristics criteria and the developed project was according to the customer's satisfaction.*

Limitations *This method can be used only early during the phase of design (at an early stage).*

Software applications of nanosatellites. It also interacts with platform peripherals. This framework helps the app developer to keep complete focus on the software's business logic. The peripherals can be accessed through the common interfaces that exist between the various nanosatellites. There is no demand to modify the code of an app on different platforms. During the phase of development, the app's testing and debugging take place. These steps take place on a simple OPS-SAT software simulator. There is a possibility of reusability of an app. This flexible design of the framework leads towards the ground by giving all the building blocks for the complete solution (See *Table 28).* In 2016, F.CM Reith et al in [26], Reliability plays an important role in achieving the quality factors of the software. In this paper, a review is performed on the factors that affect the reliability characteristics of the ISO model of the Glasgow Coma Scale. The review displays multiple potential factors so that the observers and researchers be aware of the possible confounding effect. All the factors have been analyzed along with their reliability estimates. The possible effects of experience are easily modifiable. These factors have a deficiency of concrete findings are discussed that link to deficiency in the application. The two factors that influence are modifiable that including education experience and stimulus type. These partly factors of modifiability are the main focus so that reliability can be improved. The presentation of an unvarying strategy should be used to increase reliability and declines the effect of the observer's features. Refer to Table 29 for details. In 2016, Freddy Paz et al. [20] Usability of the software is commonly defined as that software that is easy to use. The usability evaluation methods consist of various tasks which collect the data for the interaction between the software and the user. It evaluates how the activities of software participate to achieve the specified goal. They selected the recent publications by setting the selection criteria from the databases named SCOPUS, ACM Digital Library, and ISI. They listed the details of evaluation methods used in the papers. They listed all the software domains using those methods. They represented the commonly used usability evaluation method used for the listed software domain. They performed mapping which shows how many times the most popular usability evaluation methods were used in different platforms. This paper retrieved the commonly used methods of usability evaluation (Table 30).

TABLE XXVIII

SUMMARY OF NSMOF FRAMEWORK FOR SOFTWARE PROBABILITY

Ref. No.	[19]
Year	2016
Main Idea	To run the application in different environments.
Quality Models	NanoSat Mission Operation framework.
Quality Factors	Portability.
Findings	There is no need to change the code of an app while changing the platforms.
Limitations	*They considered only one environmental factor.*

TABLE XXIX

SUMMARY OF FACTORS THAT INFLUENCE SYSTEM'S RELIABILITY

Ref. No.	[26]
Year	2016
Main Idea	This paper presents the factors that influence the reliability factor.
Quality Models	ISO 9126 model.
Quality Factors	Reliability.

Findings This paper found that evidence to support the effects that influence education and training, type of stimulus to access GCS, level of consciousness affecting the reliability of the GCS. The observer's profession does not affect reliability.

Limitations This paper strongly encourages repeated application and training of the standardized method of GCS assessment, for the less experienced and experienced observers.

7

In 2016, Deepak Gupta et al. [21] Usability has now become the major attribute to achieve and it should assure during all the phases of software development. According to them, several evaluation methods and models were proposed for usability but there is a lack of defining datasets for evaluation. They performed a comparative analysis of methods of evaluation techniques i-e Inspection, Testing, and inquiry on the basis of some criteria. They generated the Software development lifecycle dataset based on subfactors of effectiveness, efficiency, and satisfaction for the SDLC models. The models were the Build & Fix model, Waterfall model, Evolutionary model, RAD model, and Spiral model. The questionnaire contains the sub-factors to scale them whether they exist or not in the model. Another dataset named live auction was produced by giving a questionnaire to participants for the evaluation of web applications on the basis of the importance of each subfactor. The qualitative evaluation techniques focused on the ''what'' kind of queries related to issues of usability. This study created the two datasets for the evaluation process of usability. It presented the comparison of the various evaluation techniques of software. Refer to Table 31 for further information.

TABLE XXX

SUMMARY OF USABILITY EVALUATION METHODS FOR SOFTWARE ENGINEERING

Ref. No.	[20]
Year	2016

Main Idea Performed the systematic review for analyzing and evaluating the existing usability evaluation Techniques.

Technique Systematic Mapping Review of Usability Evaluation Methods.

Quality Factors Usability.

Findings This systematic review determines the currently using trends in methods of usability evaluation for the platforms of software development.

Limitations *They considered only recent publications on usability evaluation methods.*

In 2018, Mengmeng Zhu et al. [18] *Software reliability is one of the main attributes in software development projects. Reliability is provided for the user's convenience by providing their functionality according to their need.* They considered one of the top ten environmental factors, Percentage of Reused Modules as a random variable. It has a random impact on the rate of detection of software faults. This study proposed a non-homogeneous Poisson process (NHPP) model of software reliability with some assumptions including detection of suspected faults. A fault occurs due to the randomness caused by these environmental factors. Real data compiled from various industries. Then applied to get the distribution or randomness of environmental factors. They represented the martingale framework includes the Brownian motion and white noise process for the detection of a fault in the software process. It shows the effect caused by that the randomness of factors. Gamma distribution is employed to model the randomness of PoRM. They performed experiments on the two open-source datasets to validate the proposed software reliability model. OSS applications are significantly affected by the randomness caused by the environmental factors. They also compared the performance of the proposed software reliability model with existing software reliability models. Table 32 represents the details of this paper.

TABLE XXXI

SUMMARY OF USABILITY DATASETS FOR EVALUATION

Ref. No.	[21]
Year	2016
Main Idea	They generated datasets on the basis of subfactors.
Technique	Usability Evaluation Techniques.
Quality Factors	Usability.
Findings	Software development lifecycle and live auction datasets are found.
Limitations	Only the sub-factors of effectiveness, efficiency, and satisfaction are considered.

TABLE XXXII

SUMMARY FOR THE DETECTION OF FAULT IN SOFTWARE DEVELOPMENT

Ref. No.	[18]
Year	2018
Main Idea	Environmental factors that affect the reliability during the SDLC.
Quality Models	Non-homogeneous Poisson process (NHPP) model.
Quality Factors	Reliability.
Findings	Applications are significantly affected by the randomness caused by the environmental factors.
Limitations	*They considered only one environmental factor.*

In 2019, Fernández Pérez et al. [16] the term 'Usability' is defined in many ways in previous researches but best defined in ISO models. The assessment of usability is defined in ISO25010 as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use". The criteria of the software evaluation process are complex to consider. So it requires to have a link between software evaluation outcomes and the authentic usability of the product. They used soft computing for the evaluation of usability includes fuzzy logic and fuzzy linguistic modeling. The proposed Fuzzy Cognitive Mapping for interdependence between considered criteria. Essential criteria are the sub-criteria determined by usability requirements. Usability model represented in a graph having the nodes of evaluation criteria and subcriteria at different levels of the model. Also contains vertical and horizontal links. Horizontal links give the interdependence between criteria. Each criterion at different levels has a weight represented by a linguistic variable. For each pair of sibling criteria (nodes), FCM is found. Thus the various adjacency matrix appears and is gathered. Then matrix of interdependence between criteria is determined. On the basis of this study, a mobile app was developed to assess the usability of mobile applications. This method and app of this proposal in a controlled environment prove that it is an operative, reliable and precise solution (See Table 33).

TABLE XXXIII

SUMMARY OF FCM FOR USABILITY ASSESSMENT

Ref. No.	[16]
Year	2019

Main Idea	Evaluation of Software Usability.
Technique	Fuzzy Cognitive Mapping.
Quality Factors	Usability.
Findings	This method allows evaluating the interrelations between considered criteria, the essential criteria, and data independence to get a global index of usability.
Limitations	This method is limited to a controlled environment.

In 2019, T. Kroeger and N. Davidson in [30] states that a vast increase in software engineering associations is characterizing and executing forms as a way to maintain, guide, and switch the execution of the project. A statement baselining this process methodology to business improvement for the quality of the process will affect the cost, time to lounge the product in the market, and the quality. This research ISO 9126 explains the result of the attempts to examine the question and propose a perspective-based quality model of software engineering procedure obtained from the expressed experience of software designing specialists. The proposed model suggests that specialists see the obtained quality of the procedure regarding the four quality attributes of usability, suitability, evolvability and manageability appropriateness, convenience, reasonability. The paper additionally proposes how information on these quality attributes, the factors of environment, properties, and their connections can be useful to back up the software procedure engineering activities. As indicated by the model, stakeholders judge in general the nature of a software designing process regarding four particular quality characteristics: usability, suitability, evolvability, and manageability. Moreover, such decisions are impacted by properties identified with the architecture, semantics, and representation procedure of software engineering. Researchers are constantly putting their efforts to ensure quality in the field of software development. See Table 34 for more details.

In 2020, Nigussu Bitew Kassie et al. [1] focused on software quality, according to the user's perspective. Software quality assurance (SQA) manages all the activities related to quality assurance in the company. Quality can be measured by the user's perspective of how well the software product performs to meet their needs. They represented qualities and relationships among them through different models for the enhancement of attributes according to the user's point of view. Software quality increases the efficiency of software. It reduces cost, time and gains a high rate of customer satisfaction. The factors of enhancing Software quality fall into the category of non-functional requirements. Software quality metrics are used to measure the quantity to which a process possesses an attribute. Product metrics, process metrics, and project metrics are the categories of quality metrics. They proposed a user's perspective quality model through a survey to help the software companies to provide a software product of high quality. Refer to Table 35 for more details. H. Ghandorh et al in [31], States that over recent years, it has been preferred that existing software should be portable and flexible with a variety of new computing environments. Even now, software engineers also agree that portability is a very important quality attribute of software quality and can be easily achievable. Mostly this attribute is acquired by a very famous Ad-Hoc Technique which helps in making software portable. The problem that is addressed in this paper is that many software product managers attain the portability attribute by using a subjective approach and there are very few people that use a unified measuring approach to attain software portability. In this paper, this issue is addressed and an overview is presented that consists of the proposed metric of measurement of portability. In this paper, the methodology used is SLR (systematic literature review) to extract and analyze the data. The research question for this study was about which methods help measure portability. After analyzing the results should that there Software Architecture Analysis Method (SAAM) helps in attaining portability and the results also suggest that there are different designs of software and hardware systems that are a huge problem for standardizing the ways which help in measuring software portability. Refer to Table 36 for more information about this paper.

TABLE XXXIV

SUMMARY OF PERSPECTIVE BASED MODEL OF QUALITY IN SOFTWARE ENGINEERING

Ref. No.	[30]
Year	2019
Main Idea	This paper presents a perspective-based model of quality for software engineering processes.
Quality Models	ISO 9216 model.
Quality Factors	Usability, Suitability, Evolvability, and Manageability.
Findings	The perfect model gives both an expressed structure that is used for encouraging the debate of software designing process issues and also presents the technique that will help to improve the activity of the software engineering process.
Limitations	In this paper, other quality attributes like reusability should also have been analyzed.

TABLE XXXV

SUMMARY OF PERSPECTIVE BASED MODEL OF QUALITY IN SOFTWARE ENGINEERING

Ref. No.	[1]
Year	2020
Main Idea	To achieve the quality level of a software product from the user's perspective.
Quality Models	They proposed a model of the user's perspective.
Quality Factors	It includes Functionality, Reliability, Usability, Efficiency, Maintainability, Portability, Understandability, Interoperability, Operability, and Aesthetic.
Findings	Found the user's point of view about the quality.
Limitations	How the final proposed model of user's perspective is found from a survey.

TABLE XXXVI

SUMMARY OF SOFTWARE PROBABILITY MEASUREMENT

Ref. No.	[31]
Year	2020
Main Idea	To perform a systematic literature review for software portability measurement.
Method	Software Architecture Analysis Method (SAAM).
Quality Factors	Portability.
Findings	From this paper this can be gathered there are different designs of software and hardware systems that are a huge problem for Standardizing the ways which help in measuring software portability.
Limitations	In this paper there is no discussion about current metrics were used in a specific domain to measure portability.

The above-mentioned research questions are answered here.

RQ1: What are the quality attributes that are important to users?

Several quality attributes are important to users which include functionality, usability, reliability, performance, supportability, maintainability, flexibility, testability, efficiency, portability, and Interoperability, etc.

RQ2: Are they achievable or easy to implement to improve quality?

The quality attributes i-e Functionality, Efficiency, Usability, Reliability, Portability, and Maintainability can be achieved by following the below guideline.

1) Functionality (Are the required functions available/workable in software).
2) Efficiency (How efficient is the software).
3) Usability (Is the software easy to use).
4) Reliability (How reliable the software or long-lasting).
5) Portability (Can the software be transferred from the current environment to another).
6) Maintainability (How easy is it to modify the software).

RQ3: How the attributes can help users to satisfy them?

The above-mentioned attributes are the basis of achieving the quality level in software which leads to user satisfaction.

VIII. FUTURE CHALLENGES

The software quality assurance of software products through models and techniques highlighted the following future challenges.

1) TACIT KNOWLEDGE

It is observed that there is a need for more research on how tacit knowledge contributes to software quality as we reviewed in the literature that programs having tacit knowledge leads to quality level [2]. Tacit knowledge is the knowledge that cannot be transferred through teaching but required personal experience.

2) NEW SOFTWARE QUALITY MODEL

The analysis of the existing models with their benefits and limitations [7] would lead to building a new software quality model by avoiding their mentioned limitations.

3) ENVIRONMENTAL FACTOR LEADING TO FAULT BEHAVIOUR

The fault that occurs due to the randomness caused by an environmental factor is detected by a non-homogeneous Poisson process (NHPP) model of software reliability [18]. Future direction pointing toward the study of two or more environmental factors and correlation could also be considered between these environmental factors.

4) USER'S PERSPECTIVE MODEL

The different user perspective models that are proposed previously can be implemented in some software organizations to meet the quality level of users.

5) LACK OF QUALITY MATRICES

Software quality metrics are used to measure the quantity to which a process possesses an attribute. It has been observed that no research study is conducted that analyzes quality matrices along with quality models. As to evaluate quality models for given quality attributes, more value will be added if metrics are also analyzed in the future. [10]

6) LACK OF ASPECT-ORIENTED MODEL

There are very few studies that are available on aspect-oriented quality models of software as we know that researchers mostly used other existing quality models for their research. This is an important area for future researchers. [11]

7) LACK OF QA ADDRESSED

After analysis of the literature review, some of the factors that have not been covered are learnability, time behavior, conformance, compliance, self-descriptive, and co-existence. All these factors have their importance and can play an important role in achieving the quality of the software hence there is a need to evaluate quality models for these factors. [13]

8) CREATION OF SOFTWARE QUALITY ATTRIBUTE DATABASE

Software quality database can be created in the future that contains all the identified quality attributes/ quality factors to be used by the user suitable for their quality projects. [15]

IX. CONCLUSION

In this era of software engineering, achieving the quality of the software system or product is the most important and challenging task as user satisfaction relies on the quality of the product. In this paper, we studied the existing software quality models and presented the comparative analysis of quality attributes of these models. We also focus to explore the software model of the quality attribute from the user's perspective. After presenting the comparative analysis, we came to know that the ISO 9126 quality model is used as a standard in most organizations to achieve the quality of the software so we have selected this as a benchmark for this research paper. Then depending on ISO 9126 benchmark we have performed a survey from different university students to rank the quality attitudes from the user's perspective. For this purpose, we choose three softwares including MS Word, MyEclipse, and Google Chrome. Students evaluated their quality level on the basis of quality attributes of the ISO 9126 model. Then, after gathering the response of the participants from the survey we have identified the most critical quality factors, and based on this we have presented our software quality model according to the user's perspective. Finally, we believe that our quality model according to the user's perspective will flourish and innovative research areas will be beneficial to highlight future enhancement. We suggest that more research should be done in the future that covers some of the other quality factors that are also very important to address such as learnability, time behavior, conformance, compliance, and self-descriptive.

REFERENCES

[1] Nigussu Bitew Kassie and Jagannath Singh, "A study on software quality factors and metrics to enhance software quality assurance", *Int. Productivity and Quality Management*, vol. 29, no. 1, 2020.

[2] Brijendra Singh and Suresh Prasad Kannojia, "A Model for Software Product Quality Prediction", *Journal of Software Engineering and Applications*, vol. 5, pp.395-401, Jun. 2012.

[3] Naheed Azeem and Shazia Usmani, "Defect Prediction Leads to High-Quality Product", *Journal of Software Engineering and Applications*, vol. 4, pp.639-645, Nov. 2011.

[4] Che-Wei CHANG1, Der-Juinn HORNG, and Hung-Lung LIN, "A Quantity Model for Controlling and Measuring Software Quality Based on the Expert Decision-Making Algorithm", *Intelligent Information Management*, vol. 1, pp. 81- 88, Nov. 2009.

[5] Kazuhiro Esaki, "Prediction Models for Total Customer Satisfaction Based on the ISO/IEC9126 System Quality Model", *American Journal of operations Research*, vol. 3, pp.393-401, Jul. 2013.

[6] B. W. Boehm, J. R. Brown and M. Lipow, "Quantitative Evaluation of Software Quality", *ICE '76: Proceedings of the 2nd international conference on Software engineering*, 1976.

[7] Dr. Deepshikha Jamwal, "Analysis of Software Quality Models for Organizations", *International Journal of Latest Trends in Computing,* vol.1, issue 2, Dec. 2010.

[8] Durgesh Samadhiya and Su-Hua Wang, "Quality Models: Role and Value in Software Engineering", *2nd International Conference on Software Technology and Engineering (ICSTE)*, 2010.

[9] J. P. Miguel, D. Mauricio, and G. Rodríguez, "A Review of Software Quality Models for the Evaluation of Software Products*", International Journal* of *Software Engineering & Applications (IJSEA)*, vol.5, no.6, Nov. 2014.

[10] R. E. Al-Qutaish, "Quality Models In Software Engineering Literature: An Analytical and Comparative Study", *Journal of American Science, Marsland Press, Michigan, USA*, vol. 6, no. 3, pp. 166-175, 2010.

[11] Suman, M.W. and Rohtak, M.D.U. "A comparative study of software quality models", *International Journal of Computer Science and Information Technologies*, vol. 5, no. 4, pp.5634–5638, 2014.

[12] R.K Pensionwar, A. Mishra and L. Singh, "A Systematic Study of Software Quality – The Objective of Many Organization", *International Journal of Engineering Research & Technology (IJERT),* vol. 2, issue 5, May. 2013.

[13] A. B. Tomar and Dr.V. M. Thakare "A Systematic Study of Software Quality Models", *International Journal of Software Engineering & Applications (IJSEA),* vol.2, no.4, Oct. 2011.

[14] K. Mordal, N. Anquetil, J. Laval, A. Serebrenik,B. Vasilescu and Stéphane Ducasse, "Software quality metrics aggregati on in industry", *Journal of Software: Evolution and Process,* 2012.

[15] P. Sudhaman "Evaluating software quality using software metrics for a software product", *Int. J. Information Systems and Change Management,* vol. 5, no. 2, 2011.

[16] Yamilis Fernández Pérez, Carlos Cruz Corona and Ailyn Febles Estrada, "Fuzzy Cognitive Maps for Evaluating Software Usability", *Uncertainty Management with fuzzy and rough sets,* Vol. 377, pp.141-155, Jan. 2019.

[17] Cigdem Altin Gumussoy, "Usability guideline for banking software design", *Computers in human behavior,* vol.62, pp. 277-285, 2016.

[18] Mengmeng Zhu and Hoang Pham, "A software reliability model incorporating martingale process with gamma-distributed environmental factors", *Stochastic Modeling and optimization*, Jun. 2018.

[19] César Coelho, Dr. Mario Merri, and Dr. Mehran Sarkarati, "NanoSat MO Framework: Achieving On-board Software Portability", *SpaceOps Conference*, May. 2016.

[20] Freddy Paz and José Antonio Pow-Sang, "A systematic mapping review of usability evaluation methods for software development process", *International Journal of Software Engineering and Its Applications*, vol. 10, no. 1, pp. 165-178, 2016.

[21] Deepak Gupta and Ashish Khanna, "Software usability datasets", *International Journal of Pure and Applied Mathematics*, vol. 117, no. 15, 2017.

[22] Anas Bassam AL-Badareen, Mohd Hasan Selamat, Marzanah A. Jabar, Jamilah Din, and Sherzod Turaev, "User's perspective of Software quality", *Researches in Software Engineering, Parallel and Distributed Systems,* May. 2014.

[23] A. Abran, A. Khelifi, W. Suryn, and A. Seffah, "Consolidating the ISO Usability Models", *Article*, Jan. 2004.

[24] 22I. Heitlager, T. Kuipers and Joost Visser, "A Practical Model for Measuring Maintainability", *Sixth International Conference on the Quality of Information and Communications Technology*, Sep. 2007.

[25] 23A, Abran, K. T. Al-Sarayreh, J. J. Cuadrado-Gallego, "A standards-based reference framework for system portability requirements*", Journal of computer Standards & Interfaces*, Jan. 2013.

[26] F.CM Reith, A. Synnot, R. D. V Brande, R. L. Gruen, and A. IR Maas, "Factors Influencing the Reliability of the Glasgow Coma Scale: A Systematic Review", *Belgium*, Dec. 2016.

[27] J. P. Correia, Y. Kanellopoulos and J. Visser "A Survey-based Study of the Mapping of System Properties to ISO/IEC 9126 Maintainability Characteristics", *IEEE International Conference on Software Maintenance*, Sep. 2009.

[28] H. Al-Kilidar, K. Cox and B. Kitchenham, "The Use and Usefulness of the ISO/IEC 9126 Quality Standard", *Conference: Empirical software Engineering*, Dec.2005.

[29] S. Fahmy, N. Haslinda, W. Roslina, and Z. Fariha, "Evaluating the Quality of Software in e-Book Using the ISO 9126 Model", *International Journal of Control and Automation*, Jun. 2012.

[30] T. Kroeger and N. Davidson, "A Perspective-based Model of Quality for Software Engineering Processes", *Australian Software Engineering conference*, 2009.

[31] H. Ghandorh, A. Noorwali, A. B. Nassif, L. F. Capretz and R. Eagleson, "A Systematic Literature Review for Software Portability Measurement:Preliminary Results," *9th International Conference on Software and Computer Applications*, Malaysia, 2020.

[32] I. Padayachee, P. Kotze and A. van der Merwe, "ISO 9126 External Systems Quality Characteristics, Sub-Characteristics and Domain-specific criteria for Evaluating E-Learning Systems", *In Proceedings of SACLA Conference*, 2010.

[33] R. E. Al-Qutaish, "Quality Models In Software Engineering Literature: An Analytical and Comparative Study", *of American Science, Marsland Press, Michigan, USA*, vol. 6, no. 3, pp. 166-175, 2010.

[34] A. Rawasshdeh and B. Matalkal, "A New Software Quality Model for Evaluating COTS Components," *Journal of Computer,* vol 2, no.4, pp.173-381, Apr. 2006.

[35] Garvin, D.A, "What does 'product quality really mean?" *Sloan Management Review*, Vol. 26, No. 1, p.25, 1984.

[36] M. Ortega, "Construction of a Systematic Quality Model for Evaluating a Software Product," *Software Quality Journal*, pp. 24, 2003.

[37] Julien Siebert, Lisa Joecke, Jens Heidrich, Adam Trendowicz, Koji Nakamichi, Kyoko Ohashi, Isao Namba, Rieko Yamamoto, and Mikio Aoyama, "Construction of a quality model for machine learning systems," *Software Quality Journal,* Apr. 2021.

[38] T. Galli, F. Chiclana, F. Siewe, "On the Use of Quality Models to Address Distinct Quality Views," *Article of Applied System Innovation,* Jul. 2021.

[39] T M. Mousaei, "Review on Role of Quality Assurance in Waterfall and Agile Software Development," *Journal of Software Engineering & Intelligent Systems,* vol. 5, Aug. 2020.

[40] Chuan Yue, "A projection - based approach to software quality evaluation from the users'perspectives," *International Journal of Machine Learning and Cybernetics,* 2019.

[41] H. Ghandorh, A. Noorwali, A. B. Nassif, L. F. Capretz and R. Eagleson, "A Systematic Literature Review for Software Portability Measurement: Preliminary Results," *9th International Conference on Software Applications,* Malaysia, 2020.

[41] Dr.Jagannath Singh and Nigussu Bitew Kassie, "User's Perspective of Software Quality," *IEEE Conference,* 2018.

[42] Ayat Mohammad and Hamed Fawareh, "Software Quality Model in the Presence of Culture Quality Factors," May. 2021.

[43] P. Kokol, "Software quality: A Historical and Synthetic Content Analysis," *Computer Engineering Journal,* Jun. 2021.

[44] Shivani Yadav and Bal Kishan, "Analysis and Assessment of Existing Software Quality Models to Predict the Reliability of Component-Based Software," Jun. 2020.

[45] G. L. Saini, Deepak Panwar, Sandeep Kumar, and Vijander Singh, "A systematic literature review and comparative study of Mathematical Sciences and Cryptography, May 2020.

[46] Deepak Gupta and Ashish Khanna, "Software usability datasets", *International Journal of Pure and Applied Mathematics,* vol. 117, no. 15, 2017.