Software Quality Assessment Approach using Analytical Hierarchical Model: Applied in SMEs

Abeer Hamed
Information System Department,
Faculty of Computer & Information Sciences,
Mansoura University, Egypt
Abeer.salah.hamed@gmail.com

Alaa Riad
Information System Department,
Faculty of Computer & Information Sciences,
Mansoura University, Egypt
amriad2014@gmail.com

Ramadan Mowad
Computer Science Department
Faculty of Computing and Information
Technology, Future University in Egypt, Egypt
Ramdan.mowad@fue.edu.eg

Hazem El-Bakry
Information System Department
Faculty of Computer & Information Sciences,
Mansoura University, Egypt
elbakry@mans.edu.eg

Abstract: *Continuous software quality evolution is crucial albeit challenging due to tight budgets and timelines. In fact, neglecting software quality in favour of customer satisfaction; either by reducing internal or external defects or by adding technical features shall place much pressure on project managers and stakeholders to find a balance between project budget and schedule. The increasing demand for high-quality software among stakeholders emphasizes the need for comprehensive approaches that can evaluate and rank quality metrics to maximize their benefits. In this research paper, we introduce an assessment approach developed using key quality metrics integrated from Cost Of Software Quality (COSQ) metrics, reliability metrics, and cost of defects metrics based on quantitative multicriteria decision analysis capabilities of the Analytic Hierarchy Process (AHP) model applied with a real case study for the software development team in Small and Medium Enterprises (SMEs) since most of them do not follow standards nor adopt well-structured quality methods. The evaluation is established based on the integrated AHP assessment approach as a practical solution for organizations seeking a critical examination of the relative significance of software metrics in terms of their utility in enhancing business performance. As a result, an evaluation of these quality metrics has been conducted to highlight the percentage of weight for each quality metric based on interviews with subject matters experts. The ranking of software quality metrics greatly helps stakeholders in selecting the most appropriate attribute for evaluating the developed software to identify high-impact quality initiatives and measure their effectiveness in the Software Development Life Cycle (SDLC).*

Keywords: *Analytical hierarchical process, cost of defects metrics, cost of software quality metrics, reliability metrics, small and medium enterprises, software quality*.

Received November 29, 2023; accepted February 19, 2024
https://doi.org/10.34028/iajit/21/2/10

1. Introduction

In the field of software engineering, software quality is a vital component and a highly significant concern for all stakeholders involved in each organization. Its importance continues to escalate rapidly in response to growing customer demand [5, 11]. It refers to the extent to which the software aligns with the predetermined customer requirements, adheres to the agreed budget and timeline, and demonstrates efficient functionality and effective deliverables [24]. If the software quality is inadequate or fails to meet the intended requirements, it can lead to the overall failure of the development endeavor. To mitigate the factors contributing to inadequate software products, it is essential to adhere to software quality parameters that encompass crucial attributes of exemplary software [11].

Customers have strong expectations for software that demonstrates superior quality and reliability, while also holding the belief that software development companies will adhere to the highest standards throughout the software development process. In order to meetcustomer satisfaction, expand their market share, reduce software costs, and enhance the reliability of the software product, it is imperative to attain a high level of software quality [22, 24, 25]. Assessing the quality of software hinges significantly on the consideration of reliability as a pivotal factor [5, 6]. Consequently, organizations aiming to ensure the effectiveness and quality of their software before deploying it to end-users consider the measurement of software reliability to be of critical importance [11, 25].

Software metrics are utilized to estimate software products and the software development process. They are consistently valuable in effectively managing and controlling the software development process, hence ultimately enhance software quality. These metrics should possess certain attributes, such as simplicity, clarity, ease of understanding, definability, reasonability, robustness, validity in addition to objective orientation [11, 24].

Balancing software quality with budget and time constraints is an ongoing challenge for organizations. Software quality is often neglected in the evolution and adaptation of software. The importance of quality in software engineering is widely recognized today to ensure customer satisfaction.

Stakeholders often find themselves uncertain about a reliable measure of software quality due to the diverse interpretations and perspectives surrounding the concept. The aim of this study is to fulfil this need by developing an Analytic Hierarchy Process (AHP) based assessment approach to identify high-impact quality improvement projects tailored for Small and Medium-sized Enterprise (SME) resource constraints and priorities.

Throughout this paper, our goal shall be to study and investigate the proposed integrated assessment approach with 3 layers (COSQ, reliability, cost of defects) using an AHP model provided with case study for one of SME in software field. Section 2 sheds light on the related literature review and background. Section 3 focuses on a proposed assessment quality approach. Section 4 presents the case study and the research results. Finally, section 5 presents the conclusion and future work.

2. Literature Review

Actually, Quality is deemed the primary concern in software development and is consistently demanded by customers [24]. Assessing and prioritizing software quality improvement initiatives require consideration of key attributes such as the cost of poor-quality metrics, reliability metrics, as well as cost of defects metrics. This literature review section discusses software quality metrics in different studies for evaluating these critical quality characteristics and techniques for decision-making from practitioners' sources. Rathi *et al*. [25] have addressed the fundamental elements of software reliability, the different approaches employed to enhance it, and the associated challenges. This paper provides valuable insights into the field of software reliability and serves as a relevant resource for further research in the area.

Colakoglu *et al*. [6] have presented a comprehensive analysis of software product quality metrics. The study systematically maps existing literature to identify and categorize various metrics used to assess software product quality. The study likewise provides an overview of the current state of research in this domain, highlighting the metrics commonly utilized and the areas where further research is needed. However, the study has some limitations, as the applicability and practicality of deriving different metrics have not been assessed.

Bajjouk *et al*. [4] have conducted a study evaluating different software testing strategies for improving reliability and quality. They have analyzed data from three software companies, comparing the effectiveness of unit, integration, and system testing approaches. Metrics like the number of detected defects, test coverage, and cost were measured. The findings have highlighted opportunities to integrate various test levels for continuous quality improvement.

Belinda *et al*. [5] have presented a study applying the AHP to evaluate key software quality attributes. They have developed a quality metrics assessment model with sample product alternatives rated and priorities determined through AHP calculations. The proposed integrated approach provides an objective means to quantitatively analyze quality metrics trade-offs. While the AHP has proven to be a valuable tool for resolving intricate decision-making challenges across diverse domains, its utilization within the software quality assurance industry remains relatively limited.

The literature review reveals that there is a significant gap in research in this area, indicating a lack of substantial progress or exploration in this direction. Existing approaches lack integration and prioritization capabilities and need to optimize improvement efforts. This study addresses these gaps through an AHP enhanced assessment model to conduct a comprehensive analysis of software quality and reliability metrics.

3. A Proposed Approach

Software metrics serve as an approach to direct and assess software development activities, offering a quantitative basis, enabling the prediction of the software development process, and shedding light on how they influence the quality of the software and the specific stages in which they are applied during its development [11, 24].

This study focuses on addressing software quality from both the user and developer perspectives, employing an integrated approach aimed at enhancing software quality and bolstering revenue generation [5]. The proposed integrated assessment approach for identifying impactful software quality improvement initiatives in SMEs leverages the quantitative decision analysis capabilities of the AHP. AHP provides a structured means of synthesizing expert judgements into objective weights and priorities. To apply this method, the researchers have first developed a three-layer hierarchical model linking key COSQ, reliability, and the cost of defects metrics (see Figure 1). Each one of the three layers is critical to the approach and will be explained in more depth in the following sections.

The inclusion of metrics aims to capture critical success drivers for quality management in SME contexts. By eliciting pairwise comparisons to quantify each element's relative importance within this hierarchy, the AHP methodology enables prioritization of improvement options according to their estimated contribution to enhancing overall software excellence.

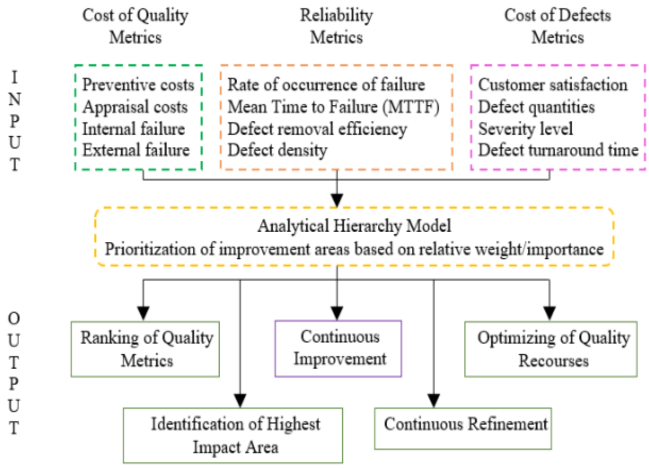

Figure 1. Software quality assessment integrated approach.

3.1. Layer 1: Cost of Quality Metrics

Software quality serves as a benchmark for evaluating software requirements and acts as a fundamental prerequisite for fulfilling user specifications [5, 27]. To enhance quality, organizations must take into consideration the costs associated with achieving quality, as continuous improvement programs aim not only to meet customer requirements but also to do so at the lowest possible cost [19]. Cost of quality (COQ) is a methodology that enables organizations to assess the allocation of resources towards activities that directly impact the quality of their products or services, encompassing both failures and deficiencies [14]. Our research has focused on using COQ metric to measure the effective software quality assurance during project of the Software Development Life Cycle (SDLC).

The concept COQ can serve as a foundation for allocating budgets to support the quality operation [20]. Moreover, the assessment of resources allocation towards activities aims at preventing poor quality, evaluating the quality of products or services, and addressing internal and external failures. The COQ is commonly defined as the combination of the non-conformance (cost of poor quality) and conformance costs (cost of good quality) [9, 19]. The Cost of non-conformance refers to the expenses associated with poor quality resulting from product and service failures, including activities like rework and return. On the other hand, the cost of conformance refers to planned activities, whereas failure costs are not part of planned activities, such as expenses related to inspection and quality appraisal [20, 28].

Furthermore, the COQ consists of four distinct categories, namely prevention cost, appraisal cost, internal failure, and external failure [9, 19]. The equation provided demonstrates the calculation of the COQ [24]. The Quality Cost=Conformance Cost+Non-conformance cost where Cost Conformance=Appraisal Cost+Prevention Cost and Cost of Non-Conformance=Cost of Internal Failure+Cost of External Failure.

- **Prevention Costs**

It refers to the proactive measures that shall be taken in order to prevent the occurrence of defects and imperfections, and thus decreasing the total number of errors, and thereby reducing the total cost associated with errors. To ensure quality the preventive cost is incurred and certain actions are undertaken in order to investigate, prevent, or mitigate the risk of defects [20]. These prevention costs are linked to the design, the implementation, and the maintenance of the quality management system [17, 19]. Preventive costs are allocated prior to the actual development of the product [28]. Examples of prevention costs include activities such as new product review, quality planning, process capability evaluations, quality improvement team meetings, quality improvement projects and training [20].

- **Appraisal Costs**

Appraisal costs do not contribute to the reduction of the total number of errors. Instead, their purpose is to detect errors prior to the delivery of the product to the customer [19]. These costs represent the direct expenses incurred for measuring quality that is defined as the degree to which the product or service aligns with customer expectations. The cost associated with evaluating quality requirements encompasses a range of activities, such as verification and control conducted throughout various stages of the development life cycle. These activities ensure that quality standards are met and customer expectations are fulfilled [17, 20]. However, appraisal is an expensive and unreliable approach to achieving quality. While these activities are necessary within a comprehensive quality program, it is highly essential to transit towards more preventive methods to minimize the costs associated with failures and poor quality.

- **Internal Failure**

It refers to the costs incurred by an organization due to defects occurring at any stage of the development life cycle before the product is delivered to the customer. These costs includes expenses associated with scrap, reworking, retesting, re-inspection, or redesign [17, 20].

- **External Failure**

It refers to the cost incurred after delivery to a customer or user due to defects occurred in various aspects which hold the greatest significance as they are identified by the customer and are likely to have a direct impact on their level of satisfaction. These costs will exceed internal failure costs, whether identified by programmers or testers.

These may include expenses related to warranty claims, product replacement, consequential losses, and the assessment of penalties incurred [17, 28]. The most effective strategy to reduce external failure costs is to be eliminated and focus on improving the other three costs of quality, i.e., the prevention costs, appraisal costs, and internal failure costs.

3.2. Layer 2: Reliability Metrics

Software reliability holds significant importance within the domain of software quality [26, 32]. The evaluation of software quality involves the quantification of program faults as well as recognizing the crucial significance of reliability, which is an essential aspect that cannot be ignored and is too complex to be measured accurately [12, 21]. According to American National Standards Institute (ANSI), software reliability is defined as the probability of a software system or component successfully performing its intended function within the specified operating conditions for the designated time duration [23, 25]. The occurrence of failures or the presence of faults in the system gives rise to the unreliability of any product. In the context of software, its unreliability primarily emanates from bugs or design faults. These factors play a substantial role in contributing to the overall unreliability of the software [12, 24].The reliability of software is highly dependent on the quality attributes and measurements associated with it. Measuring reliability entails quantifying the probability of a software system operating continuously and without failures for an indefinite duration, unless intentionally modified [1, 11, 27].

It is worth noting that each organization tends to employ its own unique set of reliability metrics. Therefore, there is a pressing necessity to establish a standardized framework for reliability metrics in accordance with international standards [27].

To measure software reliability, reliability metrics are employed, providing a quantitative expression of the software product's reliability. The selection of a specific metric depends on the nature of the system to which it is applied. The quality of software is influenced by various factors, including the software reliability model and software quality metrics [24]. Reliability metrics are derived from formulas based on failure frequency and associated data [27]. Common reliability metrics include:

- **Rate of Occurrence of Failures (ROCOF)**

It refers to the frequency of failures within a specific time frame. It represents the quantity of unforeseen incidents that transpire during a designated operational period. ROCOF for a software product is determined by dividing the total number of observed failures by the duration of the observation period. A ROCOF value of 0.02 indicates the likelihood of two failures transpiring for every 100 units of operating time. Furthermore, ROCOF is also recognized as a metric for measuring failure potency [16, 24, 25].

- **Mean Time to Failure (MTTF)**

It refers to the duration between successive system breakdowns or average over a large number of failures. A value of 100 signifies that the expected occurrence of one failure can be projected for every 100 instances [16, 24, 25]. The selection of time units is entirely reliant on the system and can be specified in various ways, including the number of transactions [12]. To calculate MTTF, the failure data for 'n' number of failures can be utilized as empirical evidence. Let the failures occur at specific time interval denoted by t1, t2, ..., tn. The MTTF can be determined using the Equation (1) presented below [10, 25]. It is crucial to emphasize that the time measurements solely take into account the runtime aspect [16].

- **Mean Time to Failure (MTTF)**

It refers to the duration between successive system breakdowns or the average over a large number of failures. A value of 100 signifies that the expected occurrence of one failure can be projected for every 100 instances [16, 24, 25]. The selection of time units is entirely reliant on the system and can be specified in various ways, including the number of transactions [12]. To calculate MTTF, the failure data for 'n' number of failures can be utilized as empirical evidence. Let the failures occur at specific time interval denoted by t1, t2, ..., tn. The MTTF can be determined using the Equation (1) presented below [25].

$$\sum_{i=1}^{n} \frac{t_{i+1} - t_i}{(n-1)} \tag{1}$$

- **Mean Time to Repair (MTTR)**

The process of rectifying a fault after its occurrence requires a certain duration. MTTR assesses the average time required to identify and correct faults that have resulted in a failure [25].

- **Probability of Failure on Demand (POFOD)**

It refers to the possibility of a transaction request failing occurring in response to a service request. It encompasses the cumulative count of system flaws, considering different system inputs. POFOD becomes relevant when a facility appeal is made, indicating the potential for system failure. A POFOD value of 0.1 implies that one out of every ten service appeals may result in failure. This metric holds particular relevance for security systems that intermittently require the provision of services [24, 25].

- **Defect Removal**

The metrics for eliminating defects involve identifying and addressing them prior to delivering the product to the customer. To achieve effective elimination of these defects, formal inspection and formal testing processes are conducted, aiming to attain a high level of defect removal efficiency [24]. A quality product is achieved through high defect removal efficiency, which yields a significant impact [24].

- **Defect Density**

It indicates the ratio of defects in relation to the size of the software [11, 24, 27]. The presence of many delivering defects leads to low product quality, resulting in a diminished impact on software quality [24].

- **Availability**

It refers to the possibility of a system to be working at a given time. It takes into consideration the duration required for system repairs and restarts. For instance, an availability value of 0.995 indicates that the system is potentially accessible for 995 out of every 1000-time units. Availability is a measure of the system's usability, accounting for both planned and unplanned downtime. In practical terms, if a system experiences an average of four hours of downtime per 100 operating hours, its availability would be 96% [24, 25, 32]. This metric holds particular significance for systems where continuous uptime is essential. In these scenarios, the duration of repair and restart time carries significance, and any interruption in service during this period cannot be disregarded [16].

Reliability encompasses not only the accuracy but also the precision characteristics of the software application. Using these reliability metrics, we can eliminate any error or fault in the software process that is how it improves the reliability of the software product [24, 31].

3.3. Layer 3: Cost Of Software Quality

- **Customer Satisfaction**: it serves as a crucial determinant of success for various companies, as it represents the extent to which the product fulfils the requirements and expectations of the customers, ultimately paving the way to achieving high-quality products [11, 24]. The European Customer Satisfaction Index (ECSI) model defines seven hypothetical variables as shown in the equation below, is a widely used framework for measuring and analyzing customer satisfaction [24].

Customer Satisfaction Index-ϵ

$$\epsilon_j = \frac{\sum_{i=1}^{n} v_{ij} . x_{ij}}{10 \sum_{i=1}^{n} v_{ij}} \quad (2)$$

where ϵ_j- the satisfaction index for customer j,

- i iterates from 1 to n, where n is the number of satisfaction variables.
- j iterates over each individual customer surveyed.
- v_{ij}- weight of variable i according to its influence on customer j satisfaction.
- x_{ij}- the rating customer j gave to variable i.
- Number 10 is connected with used scale (1-10).

The variable v_{ij} is determined as a covariance between the value x_{ij} and y_{ij} for each of the examined customers, where y_{ij} is the sum of all measurable variables for ***j***-customer [24]. Equation (2) quantifies the outcomes of customer satisfaction. It relies on the evaluations made by customers regarding the product and its associated services. So the organizations employ this approach to formulate their strategic development plans [24]. The ECSI model aids in assessing product quality by evaluating customer satisfaction, thereby exerting a significant impact on software quality [24].

- **Defect Quantities**: software faults, which encompass bugs, errors, or defects observed across all components of the system development life cycle (including requirements, design, code, and testing documents), play a significant role in influencing the timely delivery of software. Deviations from specifications are commonly referred to as defects, which are described as significant unforeseen incidents that arise during testing necessitate subsequent investigation and correction. These defects, including secondary defects known as “bad fixes,” are identified and addressed when there is a disparity between anticipated and actual test results [24, 25].
- **Defects Severity**: it refers to the extent of defects impact on the end user’s business, directly influencing the software quality. Software testing plays a crucial role in determining the severity level of defects as high defect severity refers to low product quality [24].

$$Bussiness\ Impact = Effect\ on\ the\ End\ Use * Time\ of\ Occurance \quad (3)$$

$$Defect\ Severity\ Index = \sum (Severity\ Index * No.of\ Valid\ Defects) \quad (4)$$

- **Service Time/Response Time/Defect Turnaround Time**: the response time or turnaround time for fixing defects varies based on their severity level. Equation (4) is employed to measure the turnaround time accurately [24].

$$Defect\ Turnaround\ time = \frac{Actual\ time\ fix\ defects}{Planned\ time\ fix\ defects} \quad (5)$$

The presence of a large number of delivering defects leads to low product quality, resulting in a diminished impact on software quality [24].

4. Analytic Hierarchy Process (AHP)

AHP is a powerful technique utilized to tackle complex problems and facilitate multicriteria decision making. AHP aids decision-makers in identifying the most suitable decision aligned with their objectives and understanding of a given problem [3, 13]. AHP is a valuable tool for method selection that has been formulated by Saaty (1980) at the Wharton School of Business [15]. It is rooted in principles from mathematics and psychology and developed by Saaty which has been employed to conduct a multi-criteria decision-making assessment based on interview responses [3, 5]. AHP encompasses three fundamental principles: the hierarchy framework, priority analysis, and consistency verification [15]. The initial stage of the AHP involves the formulation of the decision problem in the structure of a hierarchy framework. After constructing the hierarchy framework, AHP utilizes the computation of criteria weights through a pairwise comparison matrix, aided by a scale of relative importance, as illustrated in Table 1. This method allows the derivation of ratio scales from paired comparisons and utilizes a specific scale to convert subjective judgments into objective judgments, effectively resolving qualitative problems through quantitative analysis. The weights presented as W_{ij} represent the relative importance of the i^{th} element compared to the j^{th} element, based on the scale. If W_{ij} is greater than one, it indicates that the j^{th} element is more significant than the i^{th} element, and vice versa. The highest level of importance in this scale is assigned a value of 9 for extremely significant elements, with the numerical value decreasing as the level of importance diminishes. The simplicity of this approach has resulted in its extensive adoption across diverse domains [3]. Finally, the calculation of weights is determined based on pairwise comparison inputs, where the dominant right EigenVector (EV) of a positive reciprocal decision matrix is identified [7, 29].

Table 1. The relative scale of comparison Saaty.

Scale of importance	Degree of preference	Explanation
1	Equal importance	Two requirements are of equal value
3	Moderate importance	Experience slightly favors one requirement over another
5	Strong importance	Experience strongly favors one requirement over another
7	Very strong importance	A requirement is strongly favored, and its dominance is demonstrated in practice
9	Extreme importance	The evidence favoring one over another is of the highest possible order of affirmation
2,4,6,8	Intermediate values	When compromise is needed
1/3,1/5,1/7,1/9	Values for inverse comparison	

The consistency of the pairwise comparison matrix is evaluated by multiplying the criteria weight by the pair-wise comparison matrix. The resulting matrix is then summed, weighed by the criteria weight, and divided by the same criteria weight.

$$w_i = \frac{1}{n} \sum_{j=1}^{n} \frac{a_{ij}}{\sum_i^n a_{ij}} \, , i, j = 1,2, \dots .n \tag{6}$$

where w_i represents the criteria weight or priority assigned to a specific criterion (i^{th} criterion) in the AHP model.

- n- the size of the matrix.
- $\sum(a_{ij})$: The sum of all elements in the i^{th} row of the matrix.
- a_{ij}: The element in the i^{th} row and j^{th} column of the matrix.

The overall sum is used to calculate *(λmax)* and the consistency index (*CI*) is then determined using the Equation (7).

$$CI = (\lambda max - n)/(n - 1) \tag{7}$$

where *n* is the number of criteria (matrix size).

Saaty provides the calculated random consistency index (*RI*) corresponding to the matrix criteria size (n) that being compared. Using CI and RI, the Consistency Ratio (*CR*) is computed as [18, 30]:

$$CR = CI / RI \tag{8}$$

In *order* to validate the consistency of the pairwise comparison table *CR* should satisfy the condition of $CR <= 0.1$ where *CR* is the average consistency index [2]. In this case the comparisons are deemed sufficiently consistent [3, 5, 13].

In our research, AHP is employed to conduct a multi-criteria decision-making technique and rank software quality improvement initiatives that fulfill the software quality requirements of both end users and developers.

5. Case Study

Following a similar approach adopted in previous software studies that has employed action research and contextual methodologies, we have actively been engaged with selected software company operating in the SME sector and collaborated closely with their diverse business development teams to demonstrate the application of the integrated quality assessment approach through this research [8]. The selected project revolves around the development of an attendance web Attendance Portal (AP) that enables employees to submit vacation requests, track their vacation balance, and undergo vacation approval cycles [8]. To gain a comprehensive understanding of the project's status, we have conducted a face-to-face interview with the project's subject matter experts who are directly involved in the creation and continuous updates of the selected project. The participants holding roles such as project managers, developers and quality assurance engineers were interviewed [8]. Thematic analysis was employed to analyze the qualitative interview data in order to derive the pairwise comparison judgments for priority weights.

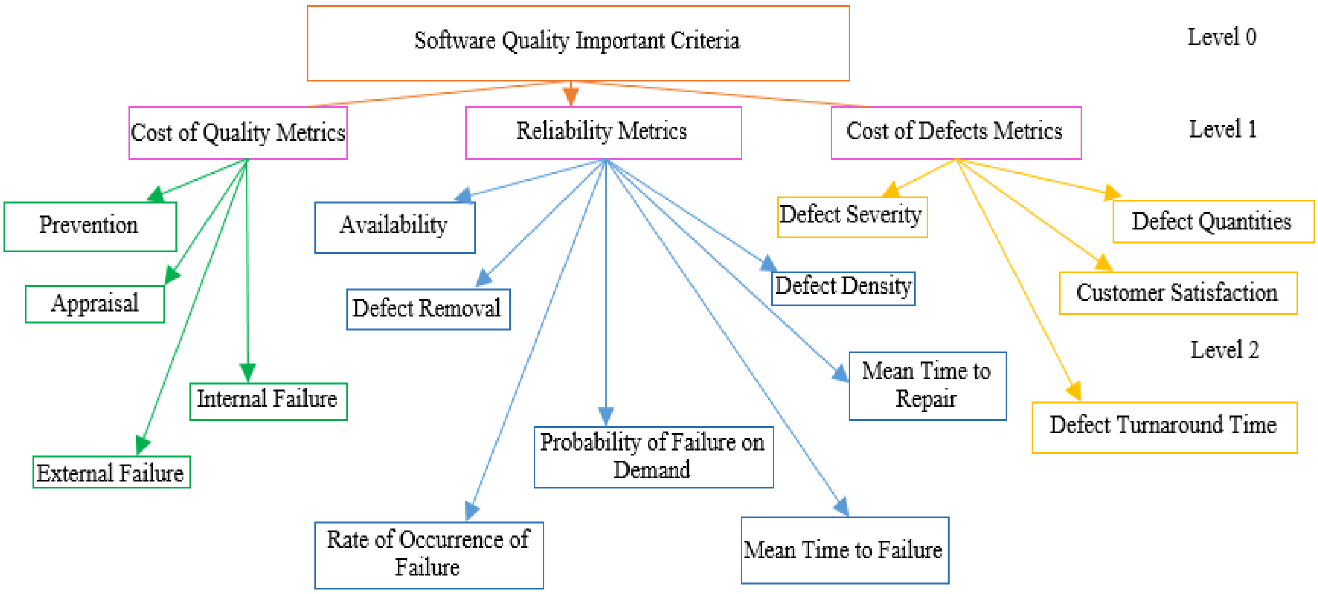

Figure 2. Hierarchical structure of selected software quality criteria.

The proposed model as described in Figure 2 integrates key metrics from three prevalent quality assessment approaches: COSQ, reliability, and cost of defects metrics. Specifically, the selection criteria will encompass a comprehensive set of 15 metrics derived from these domains. From COSQ models, the metrics include (prevention costs, appraisal costs, internal failure costs and external failure costs). Reliability metrics incorporated are (ROCOF, MTTF, MTTR, defect removal efficiency, defect density and availability). Finally, from cost of defects metrics, the criteria cover (customer satisfaction, defect quantities, severity level, and defect turnaround time).

The implementation is done using the GNU Octave software tool, which facilitates seamless calculation and analysis in support of the decision-making process. Setting up a hierarchy with these fifteen metrics is an input criterion into the AHP comparison matrix, which ranks possible improvement initiatives based on how they are thought to affect things. It is then used to get information about how much each product costs, what problems it has, and how reliable it is right now.

Table 2. Calculated pairwise comparison matrix.

	Prevention costs	Appraisal costs	Internal failure	External failure	ROCOF	MTTF	MTTR	POFOD	Defect removal	Defect density	Availability	Customer satisfaction	Defect quantities	Defect severity	Defect turnover time
Prevention costs	1.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	7.00	3.00	5.00	5.00	3.00	5.00	7.00
Appraisal costs	0.33	1.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	5.00	3.00	5.00	5.00	5.00
Internal failure	0.33	0.33	1.00	0.33	0.33	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	5.00	5.00
External failure	0.33	0.33	3.00	1.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	5.00	7.00
ROCOF	0.33	0.33	3.00	0.33	1.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	5.00
MTTF	0.33	0.33	0.33	0.33	0.33	1.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	5.00
MTTR	0.33	0.33	0.33	0.33	0.33	0.33	1.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
POFOD	0.33	0.33	0.33	0.33	0.33	0.33	0.33	1.00	3.00	3.00	3.00	3.00	3.00	3.00	3.00
Defect removal	0.14	0.33	0.33	0.33	0.33	0.33	0.33	0.33	1.00	0.33	3.00	3.00	3.00	3.00	3.00
Defect density	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	3.00	1.00	3.00	3.00	3.00	3.00	3.00
Availability	0.20	0.20	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	1.00	3.00	3.00	3.00	3.00
Customer satisfaction	0.20	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	1.00	3.00	3.00	3.00
Defect quantities	0.33	0.20	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	1.00	3.00	3.00
Defect severity	0.20	0.20	0.20	0.20	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	1.00	3.00
Defect turnover time	0.14	0.20	0.20	0.14	0.20	0.20	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	1.00
Sum	**4.89**	**7.80**	**16.07**	**10.68**	**13.53**	**18.87**	**21.67**	**24.33**	**33.67**	**27.00**	**36.33**	**37.00**	**39.67**	**48.33**	**59.00**

First, the pairwise comparison judgements from interviews with participants to share their thoughts for prioritizations of the fifteen criteria into a reciprocal matrix, which initiates by evaluating the relative importance between two selected items. Participants are required to compare and assess each criteria using the relative scale pairwise comparison, as depicted in Table 2 to rank potential improvement initiatives based on their assessed impact. For example, if we consider the comparison between prevention costs and appraisal costs, and the participants assign a value of 3 to indicate that prevention costs have moderate or essential importance relative to Appraisal Costs then a=3. Reciprocals are automatically assigned to each pairwise comparison.

The normalization process of the pairwise comparison matrix has resulted in the creation of the normalized pairwise comparison matrix, as shown in Table 3. It is obtained by dividing each criteria value in

Table 2 by the sum of the column. The criteria weight is calculated by taking the average of the rows (n).

For instance, the calculations for the criteria weight of the first column are as follows:

1. $\sum_i^n a_{ij}$ hence, (1+1/3+1/3+……+1/7) = 4.89.
2. $\frac{a_{ij}}{\sum_i^n a_{ij}}$ hence, 1/ 4.89= 0.20.
3. $\sum_{j=1}^n \frac{a_{ij}}{\sum_i^n a_{ij}}$ hence, (0.20+0.07+….+0.03) =0.1726.

Table 3. Calculated normalized pairwise matrix.

Metrics/Criteria	Criteria weight	Criteria weight %
Prevention costs	0.1726	17.26 %
Appraisal costs	0.13598	13.59 %
Internal failure	0.085167	8.51 %
External failure	0.11303	11.3 %
ROCOF	0.093991	9.39 %
MTTF	0.072575	7.25 %
MTTR	0.061806	6.18 %
POFOD	0.054275	5.43 %
Defect removal	0.039305	10.73%
Defect density	0.04751	3.93 %
Availability	0.033955	3.39 %
Customer satisfaction	0.030268	3.02 %
Defect quantities	0.026385	2.6 %
Defect severity	0.019301	1.93 %
Defect turnover time	0.013857	1.38 %

It is observed from Table 3 that the criteria weight of preventive costs is 17.26%, appraisal costs is 13.59%, internal failure is 8.51%, external failure is 11.3%, ROCOF is 9.39%, MTTF is 7.25%, MTTR is 6.18%, POFOD is 5.43%, defect removal is 10.73%, defect density is 3.93%, availability is 3.39%, customer satisfaction is 3.02%, defect quantities is 2.6%, defect severity is 1.93% and defect turnover is 1.38%. It seems that the Prevention is seen to have the highest weight 17.26 % while defect turnover time is seen to have the lowest weight 1.38%. Figure 3 presents a graphical representation showcasing the relative weights assigned to the software quality metrics.

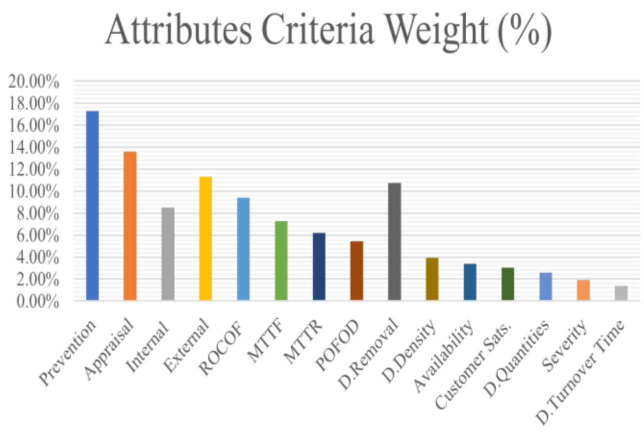

Figure 3. Software quality metrics weight.

To maintain consistency in the judgments made within the AHP a crucial step known as consistency verification is employed. Its purpose is to assess the level of consistency among the pairwise comparisons by calculating the consistency ratio. First calculate the eigenvalue (λmax) with the overall sum of the new matrix and then divided by the criteria weight as follows

1. The calculation of first row in the matrix is (0.1726+0.4079+….+0.0970)=3.0345.
2. Divide all the values of the Weighted Sum Value (WSV) by their Criteria Weight (CW) as displayed in Table 4.

Table 4. Calculated consistency of pairwise comparison matrix.

Metrics/Criteria	WSV	CW	λ WSV/CW)
Prevention costs	3.0345	0.1726	17.581
Appraisal costs	2.4548	0.13598	18.052
Internal failure	1.5211	0.085167	17.86
External failure	2.0451	0.11303	18.094
ROCOF	1.7155	0.093991	18.251
MTTF	1.2805	0.072575	17.644
MTTR	1.0808	0.061806	17.487
POFOD	0.93106	0.054275	17.155
Defect removal	0.6567	0.039305	16.708
Defect density	0.79986	0.04751	16.836
Availability	0.55432	0.033955	16.325
Customer satisfaction	0.48928	0.030268	16.165
Defect quantities	0.42121	0.026385	15.964
Defect severity	0.31558	0.019301	16.351
Defect turnover time	0.23647	0.013857	17.065
λ_{max}=17.169, CR=0.098066, CI=0.1549			

- $CI = \frac{\lambda_{max} - n}{n-1} = 0.1549$
- $CR = \frac{CI}{RI} = \frac{0.1549}{1.58} = 0.098066$

For example, for Prevention Costs=3.0345/0.1726= 17.581. Then calculate the average of these values to calculate(λ max),

(λ max)=(17.581+18.052+17.86+...+17.065)/15=17.169

The suitable value for the Random Index (RI) can be determined when dealing with a matrix size of fifteen. In this case, the value for RI is established as 1.58 [15].

The selection judgement matrix demonstrates consistency, as evidenced by the consistency ratio value of 0.098066 that is below the threshold of 0.1, which signifies that the judgements are deemed acceptable.

The resulting prioritizes rankings produced through AHP calculations and synthesis of subject matter experts' judgments; thus, it aims to guide strategic project selection and resource allocation decisions. These impact-based prioritizations equip quality teams to optimize limited resources by targeting top initiatives which are determined through the integrated assessment approach that help organizations maximize quality gains in a balanced manner considering reliability, economic as well as technical quality dimensions.

Additional benefits are recommendations guiding continuous improvement efforts and re-evaluation of metrics. Regular refinement of the iterative, data-driven solution ensures its enduring relevance over time. In summary, the proposed assessment approach based on AHP model delivers quantifiable actionable outputs and an evaluation structure empowering organizations to systematically strengthen software quality management through ongoing evidence-based refinements.

6. Limitation and Threats to Validity

In today's intensely competitive business landscape, the production of high-quality goods has emerged as a pivotal determinant of success. In this section, we examine potential factors that can present a threat to the accuracy of our research. While this preliminary study provides useful insights, some limitations must be acknowledged. As the proposed assessment approach is only evaluated through one real case study, further validation on different companies is still needed. We acknowledge that the study's limited sample size restricts its generality, and results from our small sample of the selected case study may not be representative of all software companies. Incorporating non-technical criteria like organizational culture, budget volatility, and political priorities may provide a more holistic view of feasibility and change management challenges. Further investigation is warranted to explore the acceptance and implementation of the approach across diverse commercial and industrial settings.

The proposed approach has solely relied on the AHP decision-making model. While AHP is a widely used and established model, in future research the integration of AHP and Linear Programming (LP) can be pursued to determine the most significant software quality attribute from a set of multiple attributes. Additionally, sensitivity analysis can be conducted to evaluate the impact of changes in criteria weights on the attributes and assess their implications. Therefore, we recommend that researchers consider this limitation and explore the use of additional decision-making models to enhance the comprehensiveness and validity of future studies.

7. Conclusions and Future Work

Software quality plays a pivotal role in the success of the information and technology industry. Accordingly, prior to market release, comprehensive software quality measurement is necessary to ensure that user requirements are met effectively. The increasing demand for high-quality software among stakeholders emphasizes the need for comprehensive approaches that can evaluate and rank quality metrics to maximize their benefits.

In this research paper, we introduce an assessment approach developed using key quality metrics integrated from COSQ metrics, reliability metrics, and cost of defects metrics based on quantitative multicriteria decision analysis capabilities of the AHP model.

The proposed approach develops a three-layer hierarchical model. The first layer is COSQ which provides a comprehensive understanding of the COQ of software products metrics developed through SDLC projects. The second layer is reliability metrics which hold crucial importance within the domain of software quality. The third layer is the cost of defects metrics. Key performance indicators-COQ, reliability, and cost of defects are synthesized into a hierarchical structure permitting systematic weighing of objectives to inform decision-making.

A traditional way to tackle reducing the COQ is to reduce the number of defects. However, simply aiming to find and fix all issues regardless of impact provides little optimization guidance under limited quality resources. Decisions must be made on where to focus effort first to achieve greatest returns. This study addresses the prioritization challenge by ranking defects metrics according to relative priority using AHP to facilitate absolute measurement of criteria, alternatives, and categories. The proposed integrated AHP model has been applied through a case study assessing fifteen software quality metrics across COSQ, reliability, and defects cost categories for a project team at a small-to-medium enterprise. Through stakeholders' interviews and preferences, attributes have been gathered and categorized to determine where to focus efforts for maximum quality returns.

These results provide several important benefits to organizations. First, the ranking of quality metrics according to impact guides strategic decision-making. A priority is established for each criterion based on its strength relative to the other criteria. The priorities assigned to the criteria indicate their relative importance in attaining the desired goal. The criterion with the highest priority would be deemed the most suitable option, and the ratios of the criterion priorities would indicate their relative strengths, as outlined by Saaty (1980).

Application of the AHP model has yielded the following quantitative results ranking the attributes in order of priority: Preventive costs at 17.26%, appraisal costs at 13.59%, internal failure at 8.51%, external failure at 11.3%, ROCOF at 9.39%, MTTF at 7.25%, MTTR at 6.18%, POFOD at 5.43%, defect removal at 10.73%, defect density at 3.93%, availability at 3.39%, customer satisfaction at 3.02%, defect quantities at 2.6%, defect severity at 1.93% and defect turnover time at 1.38%. Preventive costs have emerged as the top priority, followed by appraisal costs and failure metrics.

Managers can optimize allocation of quality resources by focusing on higher-ranked metrics with the most potential for cost reduction and improvement which maximizes returns on quality investments. Second, regularly refining the model through re-evaluation ensures priority ratings keep pace with changing business needs and allow for continuous improvement. The iterative data-driven framework continually strengthens quality management.

Furthermore, prioritizing attributes in this evidence-based manner supports both optimizing resource investments and ongoing refinement of quality processes. With constrained budgets, impact-driven prioritization aids efficient continuous improvement.

In conclusion, the AHP based integrated quality assessment model presented in this study provides a flexible approach that is adaptable to varying organizational contexts. As the model elicits practitioner preferences through stakeholder interviews to gather pairwise comparisons of attributes and calculate prioritized weights, the specific results of quality metric prioritization are dependent on inputs from the target company's decision-makers. Their unique perspectives on quality metrics will influence the derived rankings. Therefore, applying the proposed approach based on AHP model does not produce a single fixed prioritization, but rather tailored, impact-based orderings customized to each individual organization's circumstances and targets. This flexibility means that if the same model is applied to different software development companies, the end ranking of metrics may reasonably differ based on variances in their quality philosophies and strategic considerations at a given point in time. By demonstrating applicability and benefits, the findings establish a foundation for future work further contributing to this vital domain.

Further studies are necessary to fully support and generalize the findings of this study. Large-sample field testing, utilizing the proposed quality assessment approach, is essentially required to validate its effectiveness. Additionally, it will be beneficial to conduct additional testing of the proposed approach at other software companies. This can facilitate a "what-if" analysis to assess how the final outcomes would be modified if the criteria weights have been different. Moreover, a fuzzy logic model is proposed to incorporate the weights of the criteria of the proposed quality assessment approach. Overall, this study presents a decision model that addresses the needs of constrained organizations seeking to maximize quality returns, but further research is essential for broader applicability and validation.

References

[1] Alakus T., Das R., and Turkoglu I., "An Overview of Quality Metrics Used in Estimating Software Faults," *in Proceedings of the International Artificial Intelligence and Data Processing Symposium*, Malatya, pp. 1-6, 2019. DOI:10.1109/IDAP.2019.8875925

[2] Al Nawaiseh A., *The Effect of Information Technology on Business and Marketing Intelligence Systems*, Springer, 2023. https://doi.org/10.1007/978-3-031-12382-5_20

[3] Asadabadi M., Chang E., and Saberi M., "Are MCDM Methods Useful? A Critical Review of Analytic Hierarchy Process and Analytic Network Process," *Cogent Engineering*, vol. 6, no. 1, pp. 1-11, 2019. https://doi.org/10.1080/23311916.2019.1623153

[4] Bajjouk M., Ramachandiran C., Rana M., and Chelliah S., "Software Testing for Reliability and Quality Improvement," *Journal of Applied Technology and Innovation*, vol. 5, no. 2, pp. 40-46, 2021. https://jati.sites.apiit.edu.my/files/2021/03/Volume5_Issue2_Paper7_2021.pdf

[5] Belinda B., Emmanuel A., Solomon N., and Kayode A., "Evaluating Software Quality Attributes Using Analytic Hierarchy Process," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 3, pp. 165-173, 2021. DOI:10.14569/IJACSA.2021.0120321

[6] Colakoglu F., Yazici A., and Mishra A., "Software Product Quality Metrics: A Systematic Mapping Study," *IEEE Access*, vol. 9, pp. 44647-44670, 2021. https://ieeexplore.ieee.org/document/9336003

[7] Goepel K., "Implementation of an Online Software Tool for the Analytic Hierarchy Process (AHP-OS)," *International Journal of the Analytic Hierarchy Process*, vol. 10, no. 3, pp. 469-487, 2018. https://doi.org/10.13033/ijahp.v10i3.590

[8] Hamed A., Elbakry H., Riad A., and Moawad R., "A Proposed Technical Debt Management Approach Applied on Software Projects in Egypt," *Journal of Internet Services and Information Security*, vol. 13, no. 3, pp. 156-177, 2023. DOI: 10.58346/JISIS.2023.I3.010

[9] Helmold M., *Virtual and Innovative Quality Management across the Value Chain*, Springer, 2023. https://doi.org/10.1007/978-3-031-30089-9_9

[10] Ismail E., Utelieva N., Balmaganbetova A., and Tursynbayeva S., "The Choice of Measures Reliability of the Software for Space Applications," *in Proceedings of the International Conference on Electrical, Communication, and Computer Engineering*, Istanbul, pp. 1-5, 2020. DOI:10.1109/ICECCE49384.2020.9179411

[11] Kassie N. and Singh J., "A Study on Software Quality Factors and Metrics to Enhance Software Quality Assurance," *International Journal of Productivity and Quality Management*, vol. 29, no. 1, pp. 24-44, 2020. DOI:10.1504/IJPQM.2019.10025496

[12] Kaur G. and Bahl K., "Software Reliability, Metrics, Reliability Improvement Using Agile Process," *International Journal of Innovative Science, Engineering and Technology*, vol. 1, no. 3, pp. 143-147, 2014. https://ijiset.com/v1s3/IJISET_V1_I3_24.pdf

[13] Khan A., Hussain I., and Zamir M., "Analytic Hierarchy Process-based Prioritization Framework for Vendor's Reliability Challenges in Global Software Development," *Journal of Software: Evolution and Process*, vol. 33, no. 3, pp. 1-22, 2021. https://doi.org/10.1002/smr.2310

[14] Krasner H., The Cost of Poor Quality Software in the US: A 2018 Report, Technical Report, 2018. https://www.it-cisq.org/the-cost-of-poor-quality-software-in-the-us-a-2018-report/

[15] Kyavars V. and Subramanian S., "Multi Criteria Decision Making to Select the Suitable Method for the Preparation of Nanocrystals Using an Analytic Hierarchy Process," *Indian Journal of Pharmaceutical Education and Research*, vol. 51, no. 2, pp. 249-259, 2017. https://ijper.org/sites/default/files/10.5530ijper.51.2.30.pdf

[16] Mall R., *Fundamentals of Software Engineering*, PHI Learning Private Limited, 2018. https://books.google.jo/books?id=-JNuDwAAQBAJ

[17] Modhiya S. and Desai D., "A Review on Cost of Quality Methodology and Hidden Costs in Manufacturing Industries," *REST Journal on Emerging Trends in Modelling and Manufacturing*, vol. 2, no. 4, pp. 87-94, 2016. https://journals.indexcopernicus.com/search/article?articleId=1552642

[18] Mu E. and Pereyra-Rojas M., *SpringerBriefs in Operations Research*, Springer, 2017. https://doi.org/10.1007/978-3-319-33861-3_2

[19] Murumkar A., Teli S., Bhushi U., and Deshpande A., "Hidden Cost of Quality: A Review" *in Proceedings of the 11th ISDSI International Conference, at IIM*, Tiruchirappalli, pp. 1-20, 2017. https://www.researchgate.net/publication/321706247_Hidden_Cost_of_Quality_A_Review

[20] Murumkar A., Teli S., Jadhav S., Dharmadhikari S., and Nikam M., "Integrated Approach of Cost of Quality and Six Sigma," *in Proceedings of the Global Meet on Advances in Design, Materials and Thermal Engineering*, Navi Mumbai, pp. 30, 2018. https://docplayer.net/150025544-Global-meet-on-advances-in-design-materials-thermal-engineering.html

[21] Patel K., "Literature Review on Software Reliable Metrics," *International Journal of Engineering Research and Technology*, vol. 9, no. 5, pp. 407-410, 2021. https://www.ijert.org/literature-review-on-software-reliable-metrics

[22] Pillai N. and Hemamalini R., "Hybrid User Acceptance Test Procedure to Improve the Software Quality," *The International Arab Journal of Information Technology*, vol. 19, no. 6, pp. 956-964, 2022. https://www.iajit.org/portal/images/Year2022/No.6/20887.pdf

[23] Prabhu S. and Shashirekha H., "Effectiveness of Software Metrics on Reliability for Safety Critical Real-Time Software," *in Proceedings of the Congress on Intelligent Systems*, New Delhi, pp. 713-724, 2021. https://doi.org/10.1007/978-981-33-6981-8_56

[24] Rashid J., Mahmood T., and Nisar M., "A Study on Software Metrics and its Impact on Software Quality," *Technical Journal, University of Engineering and Technology Journal*, vol. 24, no. 1, pp. 1-14, 2019. https://api.semanticscholar.org/CorpusID:170078652

[25] Rathi G., Tiwari U., and Singh N., "Software Reliability: Elements, Approaches and Challenges," *in Proceedings of the International Conference on Advances in Computing, Communication and Materials*, Dehradun, pp. 1-5, 2022. DOI:10.1109/ICACCM56405.2022.10009422

[26] Sahu K. and Srivastava R., *Data Management, Analytics and Innovation*, Springer, 2019. https://doi.org/10.1007/978-981-13-1402-5_17

[27] Shahid M., "Software Quality Models and Reliability Measurement Metrics," *LC International Journal of STEM*, vol. 1, no. 2, pp. 49-54, 2020. file:///C:/Users/user/Downloads/LCJSTEM-019.pdf

[28] Shraim M., "A Simple Model for Identifying Costs of Quality," *in Proceedings of the ASEE Virtual Annual Conference Content Access*, Ohio, pp. 1-12, 2020. DOI:10.18260/1-2--34051

[29] Siavvas M., Chatzidimitriou K., and Symeonidis A., "QATCH-An Adaptive Framework for Software Product Quality Assessment," *Expert Systems with Applications*, vol. 86, pp. 350-366, 2017. https://doi.org/10.1016/j.eswa.2017.05.060

[30] Thakkar J., *Multi-Criteria Decision Making*, Springer, 2021. https://link.springer.com/book/10.1007/978-981-33-4745-8

[31] Tiwari U., Kumar S., and Matta P., "Execution-History Based Reliability Estimation for Component-based Software: Considering Reusability-Ratio and Interaction-Ratio," *International Journal of System Assurance Engineering and Management*, vol. 11, no. 5, pp. 1003-1019, 2020. https://link.springer.com/article/10.1007/s13198-020-01035-1

[32] Viswanathan G. and Prabhu J., "Survey of Methodologies for Quantifying Software Reliability," *International Journal of Internet Technology and Secured Transactions*, vol. 10, no. 5, pp. 565-575, 2020. https://doi.org/10.1504/IJITST.2020.109535

Abeer Hamed received the B.Sc. degree in computer and information system from Sadat Academy for Management sciences and the M.Sc. degree in Information system from Arab Academy for science, Technology and Maritime. Currently is a PhD student at Faculty of Computer and Information, Mansoura University, Egypt. Her research interests include Software Quality Assurance and Testing, Technical Debt, and software Process Improvement. Her email abeer.salah.hamed@gmail.com, ORCID. https://orcid.org/0000-0001-7595-0922

Alaa Riad is a professor at Faculty of Computers and Information, Mansoura University. He has received BS, MS, and PhD. in Electrical Engineering in 1982, 1988, and 1992 respectively, from Mansoura University, Faculty of Engineering, Egypt. Dr. Riad worked as the dean of the faculty of Computers and Information, Mansoura University. In addition, he worked as the chairman of information systems department, Faculty of Computers and Information. His research interests include intelligent information systems, mobile agents and networks, E-learning, and big data. His email amriad2000@mans.edu.eg

Ramadan Mowad received the B.Sc. degree in Electric Engineering and the M.Sc. degree in computer engineering from Military Technical College, and the Ph.D. degree in Software Engineering from the ENSAE College, France. He taught several courses in CS and CE in several institutions including the American University in Cairo, the Military Technical College, Cairo University and the Arab Academy for Science and Technology. He joined Future University in 2011 and currently working as Vice-Dean of FCIT. His research interests include software engineering and software quality assurance. His email Ramdan.mowad@fue.edu.eg, ORCID https://orcid.org/0000-0003-2349-8203

Hazem El-Bakry received the B.Sc. degree in electronics engineering and the M.Sc. degree in electrical commination engineering from the Faculty of Engineering, Mansoura University, Egypt, in 1992 and 1995, respectively, and the PHD degree from the University of Aizu, Japan in 2007. He is currently a full professor with the Faculty of Computer Science and Information Systems, Mansoura University, where he is also the Head of Information System Department. His research interests include Soft Computing, Image Processing, Neural Network, Biometrics and Pattern Recognition. His email: elbakry@mans.edu.eg, ORCID https://orcid.org/0000-0002-4798-0427