Received April 19, 2019, accepted May 24, 2019, date of publication June 5, 2019, date of current version July 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2920879

An Optimization Model for Software Quality Prediction With Case Study Analysis Using MATLAB

MEHDI GHEISARI[1,2], DEEPAK PANWAR[3], PRADEEP TOMAR[4], HARSHWARDHAN HARSH[5], XIAOBO ZHANG[1], ARUN SOLANKI[4], ANAND NAYYAR[6], AND JAFAR A. ALZUBI[7]

[1]Faculty of Automation, Guangdong University of Technology, Guangzhou, China
[2]Young Researchers and Elite Club at Parand, Islamic Azad University, Parand, Iran
[3]Department of Computer Science and Engineering, Amity University Rajasthan, Jaipur, India
[4]Department of Computer Science and Engineering, Gautam Buddha University, Greater Noida, India
[5]Department of Applied Sciences, Amity University Rajasthan, Jaipur, India
[6]Graduate School, Duy Tan University, Da Nang, Vietnam
[7]School of Engineering, Al-Balqa Applied University, Salt 19117, Jordan

Corresponding authors: Xiaobo Zhang (zxb_leng@gdut.edu.cn), Anand Nayyar (anandnayyar@duytan.edu.vn), and Jafar A. Alzubi (j.zubi@bau.edu.jo)

ABSTRACT Stakeholder satisfaction is the main motive of the software industry. Consumer and producer satisfaction means achieving a software-quality product. There are so many techniques, methods, and models are present for the software-quality prediction. Computational intelligence is also playing a crucial role in the prediction of quality characteristics. This paper gives a new optimal mathematical model for the prediction of the degree of stakeholder satisfaction (Q). Optimal models validate the real data using the relationship impacts of various quality attributes. It uses the equations of constraints. The optimal model gives the maximum and minimum values for Q. Constraints constitutes of software quality characteristics. In the given case study example, the idle value of Q is 30 but using optimal model it gives the maximum optimal value of Q = 22.788020075 for xLAB IT consulting services on their project In-Reg Molecule Registration using the MATLAB. It means if there is any change in the value of any software quality characteristics, then it will decrease the value of Q. It proves that the given result is an optimal solution.

INDEX TERMS ISO/IEC 9126, software quality, degree of customer satisfaction, reliability, reusability.

I. INTRODUCTION

History of Software starts with the name ''Alen Turing'' the father of Software. In 1935 Alen Turing proposed a theory about Software in an essay ''Computable numbers with an application to the Entscheidungsproblem'' and after that two new academic fields Computer Science and Software Engineering came into existence. Both the areas have a universal relation, and that was about the study of software and its creation. Computer Science emphasized the theoretical part, and the Software Engineering focused on the practical part of the software development. As the software engineering discipline dealing with the practical part of software development since 1935, therefore sufficient literature is available. ''Software engineering is the establishment and use of sound engineering principles to obtain economically software that is reliable and works efficiently on real machines''.

Software engineering consists of tools, methods, process and a quality focus. As the decades passing with each day the software development industry also growing with superior speed. High availability of new methodologies, tools, models, and concepts are giving the potential to this great industry. Software crisis is a huge problem for the current software development industry.

An increasing percentage of software crisis also motivates researchers to develop new methodologies like Component-Based Software Engineering (CBSE), Re-Engineering, Hybrid re-engineering, agile development, and Extream programming, etc. If the researcher talks about the development of a software system using a CBSD process than it is very efficient, cost-effective and useful for the production of quality software products. In this scenario quality prediction of CBS is necessary and important to analyze [1] Some

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Messina.

of them are bounded according to the limitations of COTS and some are constrained due to the vulnerabilities of OSS. In the case of COTS developer focuses on the interfaces but in OSS code is available to predict something more. But this is necessary for the researcher to focus at both the parts of software development while it is the case of COTS-based development or OSS based development and the third condition can also be there like the combination of COTS and OSS. Many factors exist, which influences the prediction or assessment process of the software quality attributes. It is hard to quantify software quality because it is a collection of various characteristics according to the different software quality assurance models like ISO/IEC – 9126. The complexity of the situation contributes to the fact that it is often not clear that up to what extent the different quality attributes influence the quality and in which context. Also, these factors do not only influence the quality but also have some influence on each other. Software quality assurance is a big problem in the field of software development. It acts like a combinatorial problem. So in that case, researches have to find some objective to optimize the problem [2]. When the problem is NP-Complete, then it is indeed a difficult task to solve it, in that case, the software quality assurance team uses different methods to give an optimized solution. Optimization is a way to find the best possible results under specific circumstances. In the case of Software Development Life Cycle optimization methods is used to minimize the effort with each phase to gain maximum profit.

Researchers can explain an effort in the form of function of certain decisions variables and optimization is a process which is used to find minima and maxima of a function. So this study used some optimization methods to show the relationships between different quality attributes like reusability. It is used to show the impact of quality attributes in three different manner – positive, negative and none. Optimization methods are useful to achieve the goals in the form of stakeholder's satisfaction or in the form of a higher degree of stakeholder satisfaction.

II. RELATED WORK

Quality is a vital property of an engineering product, which is very important to active up to an extent / particular level according to the stakeholder. So it is necessary to maintain the quality of the software products. There are so many quality assurance models which are present like McCall model, Boehm's model, ISO/IEC 9126, etc. These models help to effectively use the resources to make a quality product according to the requirements of the stakeholders.

A. SOFTWARE QUALITY PREDICTION OF CBSD

Software production organizations using CBSD process to produce economic software systems in the comparison of traditional methods. In CBSD, the products are built by integrating the different software components. A component can be thought of as a COTS or OSS and it has an extra property of reusability. In the case of COTS, interfaces provide this facility and for the OSS it is elementary to reuse due to the availability of the code. The developer can modify the code according to the new requirement.

Software development using COTS and OSS are economical and easy to understand regarding portability. This study talks about software quality attributes and sub-attributes for predicting the quality of CBS. Different attributes have different ways of assessing their values, like if we take an example of functionality according to the ISO/IEC 9126-2. Suitability works as a sub-attribute of functionality can be calculated with the help of some defined parameters like- metric name, metric functionality, input type, formula to calculate, type of scale, and the people involved in the calculation like developer and customer [3]. According to the ISO/IEC 9126, there are some attributes which are very important for quality assurance, and these attributes have different metrics and parameters to be calculated as shown in Table 1. Software quality attributes and sub-attributes are defined in the field of software quality assurance and prediction by a large number of researchers like McCall, Boehm's, Musa, Jorgenson and many more [4]. Software attributes like – maintainability, reliability, etc. are used for traditional and component-based software products. These attributes can be calculated with the numbers of software metrics

Literature shows that COTS and OSS are the essential parts for CBD. In the current scenario, the software production industry wants to generate more revenue to get more profit leading to the evolution of technologies. CBD plays an important role to make quality software in less time, but there are some limitations for both COTS and OSS in CBD.

- "A component is a language neutral, independently implemented a package of software services delivered in an encapsulated and replaceable container accessed via one or more given interface."
- "A software component is a physical package of executable software with a well-defined and published interface." •"A Software Component is a unit of composition with contractually specified interfaces and explicit context dependencies only. A software component can be deployed independently and is subject to third-party composition".

It is impossible to be ideal for anything in this world, so OSS also has some limitations like the quality of a product which is generated with the help of OSS is dependent upon code review and data testing. Sometimes it is inconvenient for the developer due to some reasons for example-

- Analysis of large projects.
- Lack of tools and methods for the development and quality assurance.

Software quality attributes for the CBS are the attributes which are related to the COTS and OSS quality prediction. Some of them are bounded according to the limitations of COTS, and some are constrained due to the vulnerabilities of OSS. In the case of COTS developer focuses on the interfaces but in OSS code is available to predict something more. But this is necessary for the researcher to focus at

TABLE 1. Software quality metrics of different characteristics.

S.No	Name of the attribute	Name of the subattribute	Purpose	Method of application	Formula	Scale	Target
1.	**Functionality**	Suitability (adequacy)	Check the adequacy of functions	No. of function performing by no. of the function evaluated	Z=1-X/Y X= no. of functions with problem Y= no of functions evaluated	Absolute	Developer
		Suitability (Completeness)	Completeness of implementation according to the requirement specification	Black Box Testing according to the requirements and then calculate the number of missed functions	Z=1-X/B X= Missing functions Y= Functions according to requirements	Absolute	Developer SQA
		Accuracy	Check the difference between the actual and expected output	Check input vs. output test cases results	Z=X/T X= Total test cases encountered differ T= test cases execution time	Ratio	Developer
		Interoperability	Change in the interface for specific data	Test all the interface function values with the help of output	Z=X/Y X=numbers of different formats approved Y= total number of format exchanged	Absolute	Developer
		Security	Access Control intensity	Total number of illegal operation in comparison to requirement specification	Z=X/Y X= Total number of illegal operations Y= Total types of illegal operations		
		Maturity	No. of failures during trial	Calculate the no. of detected failures	Z= X_1/X_2 X1= Detected failures X2= test cases	Absolute	Developer and tester
2.	**Reliability**	Recoverability	Availability of the system for a specific time	Test system availability with respect to some parameters like repair time	Z=T/A T= Total System Downtime A= Number of breakdowns	Absolute	User maintainer
		Fault Tolerance	Breakdown caused due to the software until system restart	Total numbers of Breakdown / Total numbers of failures	Z= 1-X/Y X= Total breakdowns Y= Total failures	Absolute	User
		Undeerstandability	How many functions are understood	Use questionnaires for the user	Z= X/Y X= Functions understood Y= Total functions	Absolute	User Maintainer

TABLE 1. ***(Continued.)*** **Software quality metrics of different characteristics.**

3.	**Usability**	Learnability	How much time a user spent to learn the functions of software system	Test the User	T= Mean time to use the functions of the system	Ratio	User
		Operability	Errors can be handled by the user	Test the user with respect to change in behavior	T= X-Y X= Time taken to complete the correction Y= Start time checking of correction	Ratio	User
		Attractiveness	Check the attractiveness of user interface	Questionnaire tool	A particular questioner calculation method	Absolute	User
4.	**Efficiency**	Time Behaviour	Time is taken by a task to complete	Measure the time taken by a task for the completion of an operation	T= Total time taken to get the result	Ratio	User, Developer, Maintainer
		Resource Utilization	Check the inefficiency due to Input/output device	Execution of large no of concurrent task	Z=X/Y X= Occupied time by input and output device Y= A particular specified time	Absolute	Developer
5.	**Maintainability**	Testability	Check the operational testing perform by user	Observation of the user behavior	Z=X/Y X= No. of cases used the built-in test function Y= No. of cases of test	Absolute	Developer Maintainer
		Analysability	Check whether the user is capable of finding the operation which caused failure	Observation of the maintainer and user behavior	Z=X/Y X=No. of data records Y= No. of data records planned	Absolute	Developer
		Changeability	Problem solution in a specified time	Handled interaction of customer and supplier	Time Calculation for request and response of maintenance	Ratio	User Maintainer
		Stability	Are the user is capable of handling the system after maintained	Calculation of failures by user and maintainer to operate the system before and after the maintenance	Total number of cases at the user site and maintainer site	Ratio	User Maintainer
		Adaptability	Adaptation of data structure	Observation of user and maintainer when user is adopting the software	Z=X/Y X=Operable data Y= Expected to be operable data	Absolute	Developer Maintainer

TABLE 1. ***(Continued.)*** **Software quality metrics of different characteristics.**

6.	**Portability**	Installability	Check the installation process is easy to user	Check the behavior of a user during the time of installation	Z=X/Y X= Total no. of cases succeed by the user Y= Total no. of attempts	Absolute	Developer
		Co-existence	Checking for any constraint	Use different system concurrently	Z=X/T X= Total no. of failures T= Total time for concurrent process	Ratio	Developer
		Replacability	Check the consistency of new components	Check the behaviour of user	$Z=1-X_1/X_2$ X_1= New functions inconsistent X_2= Total no. of new functions	Absolute	Developer

both the parts of software development while it is the case of COTS-based development or OSS based development and the third condition can also be there like the combination of COTS and OSS. So it is essential for the researchers to make some methods, models, and tools for the development organization through which these types of situations can be handled easily. This study and review of literature provide some good and appropriate solutions to overcome these type of problems like how can be a researcher would be able to predict the quality of such type of software product which has been produced with the help of COTS and OSS. There are so many quality attributes given by the different software quality assurance models like reliability, maintainability, usability etc. but in the case of CBS there is an attribute which is very important to be measured and this is Reusability because CBS is based on the concept of reusability of code or we can say that it is dependent upon the use of legacy software's.

Hackbarth *et al.* [5] stated that maintenance is a crucial and active quality attribute for software development. The base of the calculation and prediction of this attribute depends upon some parameters like- corrections of errors, optimization of the process, dropping the extra capabilities, rise in capabilities, etc. In the case of CBSD maintainability act as a variable which remains stationary and unexecuted. It includes lots of changes to the product.

Mahmood *et al.* [6] alongside the advancement of different technologies like object-oriented, CBSD is successively extended to the open source software and COTS. Due to this reason, CBSD becomes the influential field of the Software development industry. The demand for high quality needs analysis and exploration of different quality attributes like reliability, reusability, etc. for the CBSD.

Arora *et al.* [7] Quality of service sparkle as an important factor in the development of distributed systems. QOS can improve if the knowledge about a component is precisely available, but the nature of COTS is black-box, so it does not provide the suitable details for the prediction of quality to improve the QoS.

Gill and Tomar [8] in today's software development industry CBSD playing an essential role to make the sector a profitable place. When an organization uses this technique than the development process moves towards to achieve a higher degree of effectiveness and efficiency, in the view of hybrid design, the architecture and the procedure of the product turn out to be commonly strengthening, and the exterior, interior and process quality of the product are guaranteed.

Boehm *et al.* [9] find the relations between the different characteristics and reusability. It seems the capability of a learning object as a traceable, usual, and modular and standardized characteristics. It also explains the importance of different reusable parameter and also suggests the use of quality metrics.

Seffah [10] *et al.* stats that another important quality attribute "Usability" has an excellent existence for the quality of services. The author also discusses the cost-effective nature of the methods used for usability assessment and prediction. The high cost always has a negative impact on stakeholder satisfaction.

Khoshgoftaar [15] Classifying which software modules, during the software development process, are likely to be defective is an effective technique for refining software quality. Such a tactic allows a more attentive software quality & reliability enhancement endeavor.

Smidts [16] Researcher use a specific set of ranking for various quality attributes. The overall calculated value of each measure for each ranking condition was elicited through an expert estimation and then combined into a single score using multi-attribute utility concept. Use of this calculation in each software development phase can predict a more accurate quantitative prediction of software reliability.

Fenton and Neil [17] Lots of organizations hunger to calculate the number of defects (faults) in software systems, before the deployment of the software product, to scale the likely delivered quality work. To help in this process, there are several software metrics, and statistical models have been developed, with respectively an extensive literature. Researchers provide a critical review of the related literature and the state-of-the-art. Most of the software quality assurance models use the size and complexity metrics to predict defects and also are based on testing data, or take a multivariate attitude.

Researchers talked about the user of CBSD. This process of software development has a significant impact on the software industry positively to earn more money with a high degree of customer satisfaction. Reliability of these products acts as an important issue to be focused. The author also discussed the relation between the impacts of the failure of the components on each other.

Software quality attributes like portability, reusability, etc. are essential to analyze and to increase productivity. Any organization like Microsoft and Apple were also at their root level some decade's age, but these companies are continuously working for the quality of service to produce, to execute and to deploy the software products according to the market and customers' requirements. Software quality prediction and practical assessment of software products concerning the customer requirement makes it possible to achieve high levels of CMM for any organization. ISO/IEC 9126 quality assurance model is trendy in current conditions of software industry but according to the literature survey [11], if any researcher use this model for the quality prediction of CBS than one more quality attribute should also be included in it and which is Reusability because it is the base of CBS or we can say that it is the foundation of CBSD. Software quality prediction is a measure of the fulfillment criteria of customer requirements in the form of different quality attributes [12]. Measurement of attributes is dependent on some essential characteristics like the metric name, which is useful to match the metric with its exact value. Features like the purpose of the metric (ISO/IEC 9126-2), method of application, measurement formula, and most influential interpretation of the measured value of that particular matric as shown in Table 1. But the software quality means it is to meet some performance goal according to the customer requirements. The researcher defined the degree of customer satisfaction in the form of a variable Q. Metrics-based quality estimation models comprise the various attributes that provide a quality-based ordering of program modules and those that deliver a quantitative prediction of a quality feature for the program modules.

Degree of customer satisfaction measure how well the customer expectations meets the actual output of a product or a service provided by the software development organization to the customer. Fulfillment of expectations is dependent upon quality attributes, and it signifies that the quality attributes should be optimized concerning the customer requirement specifications.

B. PREVIOUS MODEL

We have an existing model for software quality prediction. It calculates the different quality attributes mathematically to analyze the degree of stakeholder satisfaction. The model is like that-

$$Q = \sum_{i=1}^{i=n} \sum_{j=1}^{j=m} K_{ij} W_{ij} \tag{1}$$

where (2),

$$Q = \textit{quality measure degree in stakeholder satisfaction}$$

$$Q = \sum_{i=1}^{i=7} \sum_{i=1}^{j=5} K_{ij} W_{ij} \tag{2}$$

$$K_{ij} = R_i S_j$$

$$R_i = \textit{priority given to the main attributes}$$

$$S_j = \textit{priority given to the sub} - \textit{atrribute}$$

$$\{\textit{before applying SFL the normalization value for } S_j \textit{ is } 0.1 \textit{ and the scale of } S_j \textit{ is between } (0-1)\}$$

W_{ij} = calculated value of quality attributes according to formulas defined in ISO/IEC 9126.

After taking equation number for modified ISO/IEC 9126. We get (3), as shown at the bottom of this page. Some attributes have five sub-attributes, but some have less than five, so in that case, the value of extra sub-attribute should be equal to zero. So it does not have any effect for Q.

The previously given model was only about the ideal condition and relationships between the different quality attributes. It was depended upon the positive coefficient. Now the new proposed model is the extended version, in which we are optimizing the quality coefficient with the help of relationships of quality characteristics and sub-characteristics according to the available literature and software industry. According to the standard optimization function, our proposed model is an optimized model with the help of proper analysis of constraints and non-negative constraints. The proposed model has some equations which show the different relationship

$$Q = R\,(S_1W_{11} + S_2W_{12} + S_3W_{13} + S_4W_{14} + S_5W_5)] + [R_2\,(S_1W_{21} + S_2W_{22} + S_3W_{23} + S_4W_{24} + S_5W_{25})] + \cdots\cdots + [R_7\,(S_1W_{71} + S_1W_{72} + S_1W_{73} + S_1W_{74} + S_1W_{75})] \tag{3}$$

impacts of the characteristics on each another. Characteristics have positive, negative and no effect on each other. Equations show the behavior of constraints and non-negative constraints and also talks about the maximum and minimum values of the various variables. According to the previous model, software quality means the degree of stakeholder's satisfaction. It constitutes of many quality attributes defined by researchers through different quality assurance models like as – McCall quality model, ISO/IEC 9126, etc. Proposed work has its dependency on the ISO/IEC 9126 quality assurance model. It consists of six main attributes, as already mentioned. But ISO/IEC 9126 does not make any transparency in case of reusability attribute and its sub-attributes. When a developer wants to go for the quality prediction for CBD than it is necessary to consider the reusability factor. So the previously proposed mathematical model has its dependency on seven quality for quality prediction of CBS [13]. In the previous model researcher defined Q as a quality degree measure in stakeholder satisfaction.

C. STANDARD OPTIMIZATION

The proposed model is about the optimization of previously given mathematical model and according to mathematics if a researcher going to optimize a model than there are some essential factors which have great significance. The equation's for the constraints, non-negative constraints, limits of the different variables are the important parameters to make an optimized mathematical model.

An optimization process or mathematical programming are according to the various problems. Researcher or mathematician cannot solve all the current issues with a single model [14]. Optimization provides the conditions through which the maximum and minimum value of any function can be counted.

An optimization can be stated as follows-

$$Find\ A = \begin{Bmatrix} a_1 \\ a_2 \\ . \\ . \\ . \\ a_n \end{Bmatrix} which\ minimize\ f(A)$$

Subject to

$$\begin{aligned} b_j(A) &\leq 0 \quad j = 1, 2 \ldots\ldots\ldots\ldots x \\ c_j(A) &= 0 \quad j = 1, 2 \ldots\ldots\ldots\ldots y \end{aligned} \tag{4}$$

where A is an n-dimensional vector. Which is called a design vector. Function f (A) is known as the objective function, b_j and c_j are known as the constraints of inequality and equality respectively. Variables n, x and y need not be related in any way. So the equation no.3 and 4 shows the constrained optimization problem.

III. PROPOSED MODEL FOR SOFTWARE QUALITY PREDICTION

This study proposed a model to optimize the degree of stakeholder satisfaction. Every optimized problem has three parts, and these are as follows

- Objective function
- Constraints
- Non-Negative Constraints

According to the equation' 1 we use

$$Max\ Q = \sum_{i=1}^{i=n} \sum_{j=1}^{j=m} K_{ij} W_{ij} \tag{5}$$

As an objective function of the proposed model.

Since the study objective is to maximize the degree of stakeholder satisfaction, it means it works as a best choice optimization objective function.

A. CONSTRAINTS EQUATIONS

After examining the literature, we have founded the relating and conflating quality attributes as discussed with the help of tables 2, 3 and 4. Table 2 is discussing the relationship between the various quality attributes given by the Boehm's in his research. Boehm's explained the influence status of the quality attributes in the form of a positive and negative manner. which is very effective to make some real and empirical relationship between the quality attributes. The influenced value is stated by negative, positive and no effects status between different qualities attributes [18]–[22]. In the capacity of "No Effects" status, it is also possible that the relationships have not been found or established yet. The relations are the collective opinions in industry and also in literature. But some basic understandings need to be placed like-

- Quality depends upon stakeholders needs and how the product can fulfill those needs or requirements, and due to this reason, the importance of particular quality attributes are influenced.
- Second, the different customer has different interest and according to which the prioritization values may vary from one attributes to another.

According to the literature survey, it is also hard to know that the granularity of the relations between the attributes exists. It also comes into account that some research is defining the relationship casually not explicitly and because of which there is a need of an empirical mathematical functional model to be applied therefor it creates the need to proposed a mathematical model.

After the literature survey about the software characteristics and sub-characteristics, this study makes the author capable of finding some relevant relationships between the different qualities attributes as shown in table 6. These relationships help to execute an objective function optimally. The defined relationships make the constraints for the objective function through which researchers are capable of elaborating all the three parts for an optimize problem, or we can say that

TABLE 2. Software quality attributes relationships referred by Boehm's model.

S.no	Software Quality Attributes			Influenced Status according to Boehm's
	Attribute - 1	**vs**	**Attribute - 2**	
1.	Reusability		Portability	Positive
2.	Performance		Portability	Negative
3.	Usability		Portability	Negative
4.	Usability		Interoperability	Positive
5.	Portability		Cost	Negative

TABLE 3. Software quality attributes relationships referred by mccall model.

S.no	Software Quality Attributes			Influenced Status according to McCall
	Attribute - 1	**Vs**	**Attribute - 2**	
1.	Reliability	**Vs**	Reusability	Negative
2.	Reliability	**Vs**	Testability	Positive
3.	Reliability	**Vs**	Correctness	Positive
4.	Reliability	**Vs**	Flexibility	Negative
5.	Usability	**Vs**	Maintainability	Positive
6.	Usability	**Vs**	Testability	Positive
7.	Usability	**Vs**	Efficiency	Negative
8.	Usability	**Vs**	Flexibility	Positive
9.	Usability	**Vs**	Integrity	Positive
10.	Reusability	**Vs**	Efficiency	Negative
11.	Integrity	**Vs**	Reusability	Negative
12.	Reusability	**Vs**	Portability	Positive
13.	Maintainability	**Vs**	Reusability	Positive
14.	Portability	**Vs**	Maintainability	Positive
15.	Reusability	**Vs**	Testability	Positive

the equations for inequality and equality. A number of constraints observed through the study of different software quality models given by the researchers. We are merging perspective of the various relations between them. Some attributes positively affect each other, some in a negative or some are neutral. To design the constraints firstly, research count the total no of quality attributes available and have some relationships according to the literature, as shown in Table 5. After that, the counting of sub-attributes is also, as shown in Table 5.

Based on the available literature. These are the following constraints. According to Table 6. Now designed constraints for the relation R_1 and R_6 is –

For R_1 vs. $R_6(+)$:

Reliability effect Usability positively. It means when reliability increase, then Usability undoubtedly increase.

We get,

$$\Rightarrow R_1 \propto R_6$$
$$\Rightarrow R_1 = \beta R_6 + Y \begin{cases} \beta = \text{Proportionality Cofficient} \\ Y = \text{Linear Constant} \end{cases} \tag{6}$$

TABLE 4. Software quality attributes relationships according to real industrial data.

S.no	Company	Software Quality Attributes			Influenced Status according to McCall
		Attribute - 1	**Vs**	**Attribute - 2**	
1.	A	Reliability	**Vs**	Usability	Positive
		Efficiency	**Vs**	Flexibility	Negative
2.	B	Reliability	**Vs**	Usability	Positive
		Reliability	**Vs**	Maintainability	Negative
		Usability	**Vs**	Efficiency	Negative
3.	C	Reliability	**Vs**	Maintainability	Negative
		Reliability	**Vs**	Efficiency	Negative
4.	D	Usability	**Vs**	Efficiency	Negative
		Usability	**Vs**	Correctness	Negative
5.	E	Maintainability	**Vs**	Portability	Negative
		Efficiency	**Vs**	Portability	Negative

TABLE 5. Attributes and their sub-attributes.

S.No	ATTRIBUTE	No of Sub-Attributes
R_1	Reliability	4
R_2	Efficiency	3
R_3	Functionality	5
R_4	Maintainability	5
R_5	Portability	5
R_6	Usability	5
R_7	Reusability	3

TABLE 6. Software quality attributes relationships for optimization model.

S.no	Software Quality Attributes			Influenced Status for Optimization Model
	Attribute - 1	**vs**	**Attribute - 2**	
1.	Reliability		Usability	Positive
2.	Portability		Reliability	Negative
3.	Reusability		Reliability	Negative
4.	Reliability		Maintainability	Positive
5.	Usability		Efficiency	Negative

So that this equation may be written as

$$\Rightarrow R_1 - \beta R_6 = Y \tag{7}$$

Expanding equation no 6.

We get,

$$\Rightarrow (R_1S_1W_{11} + R_1S_2W_{12} + R_1S_3W_{13} + R_1S_4W_{14}) - \beta\,(R_6S_1W_{61} + R_6S_2W_{62} + R_6S_3W_{63} + R_6S_4W_{64} + R_6S_5W_{65}) = Y \tag{8}$$

For R_1 vs. R_5(-):

Reliability effect Portability negatively. It means when reliability increase, then portability adversely increase.

We get,

$$\Rightarrow R_1 \propto -R_5$$
$$\Rightarrow R_1 = -\alpha R_5 + X \begin{cases} \alpha = \text{Proportionality Cofficient} \\ X = \text{Linear Constant} \end{cases} \tag{9}$$

So that this equation may be written as

$$\Rightarrow R_1 + \alpha R_5 = X \tag{10}$$

Expanding equation no 9.

We get,

$$\Rightarrow (R_1S_1W_{11} + R_1S_2W_{12} + R_1S_3W_{13} + R_1S_4W_{14}) + \alpha\,(R_5S_1W_{51} + R_5S_2W_{52} + R_5S_3W_{53} + R_5S_4W_{54} + R_5S_5W_{55}) = X \tag{11}$$

For R_1 vs. R_7(-):

Reliability effect Reusability negatively. It means when reliability increase, then reusability adversely increase.

We get,

$$\Rightarrow R_1 \propto -R_7$$
$$\Rightarrow R_1 = -\gamma R_7 + Z \begin{cases} \gamma = \text{Proportionality Cofficient} \\ Z = \text{Linear Constant} \end{cases} \tag{12}$$

So that this equation may be written as

$$\Rightarrow R_1 + \gamma R_7 = Z \tag{13}$$

Expanding equation no 12.

We get,

$$\Rightarrow (R_1S_1W_{11} + R_1S_2W_{12} + R_1S_3W_{13} + R_1S_4W_{14}) + \gamma\,(R_7S_1W_{71} + R_7S_2W_{72} + R_7S_3W_{73}) = Z \tag{14}$$

For R_1 vs. R_4(+):

Reliability effect Maintainability positively. It means when reliability increase, then maintainability undoubtedly increase.

We get,

$$\Rightarrow R_1 \propto R_4$$
$$\Rightarrow R_1 = \delta R_4 + P \begin{cases} \delta = \text{Proportionality Cofficient} \\ P = \text{Linear Constant} \end{cases} \tag{15}$$

So that this equation may be written as

$$\Rightarrow R_1 - \delta R_4 = P \tag{16}$$

Expanding equation no 15.

We get,

$$\Rightarrow (R_1S_1W_{11} + R_1S_2W_{12} + R_1S_3W_{13} + R_1S_4W_{14}) - \delta\,(R_4S_1W_{41} + R_4S_2W_{42} + R_4S_3W_{43} + R_4S_4W_{44} + R_4S_5W_{45}) = P^{\cdots} \tag{17}$$

For R_2 vs. R_6(-):

Efficiency effect Usability negatively. It means when efficiency increase, then usability adversely increase.

We get,

$$\Rightarrow R_2 \propto -R_6$$
$$\Rightarrow R_2 = -\phi R_6 + U \begin{cases} \phi = \text{Proportionality Cofficient} \\ U = \text{Linear Constant} \end{cases} \tag{18}$$

So that this equation may be written as

$$\Rightarrow R_2 + \phi R_6 = U \tag{19}$$

Expanding equation no 18.

We get,

$$\Rightarrow (R_2S_1W_{21} + R_2S_2W_{22} + R_2S_3W_{23}) + \phi\,(R_6S_1W_{61} + R_6S_2W_{62} + R_6S_3W_{63} + R_6S_4W_{64} + R_6S_5W_{65}) = U \tag{20}$$

B. IN-EQUALITY CONSTRAINTS

It is one of the three important aspects of an objective function. The proposed model has in-equality constraints because it depends upon the minimum and maximum value of the variables, or it defines a limit of any variable. Like proposed have seven software quality attributes and thirty sub-attributes. And none of the sub-attribute could exceed by 1.

$$\Rightarrow (W_{11}, W_{12}, \ldots\ldots\ldots\ldots W_{73}) \leq 1 \tag{21}$$

then, we get

$\Rightarrow \left(K_{ij}W_{ij}\right) \leq 1$ for all thirty sub-attributes

According to this the main attributes obeying the following inequalities-

$$\Rightarrow (R_1 \leq 4),\quad (R_2 \leq 3),\quad (R_3 \leq 5),\quad \times (R_4 \leq 5),\quad (R_5 \leq 5),\quad (R_6 \leq 5),\quad (R_7 \leq 3)$$

After expending R_1, R_2, R_3, R_4, R_5, R_6 and R_7 we get,

$$\Rightarrow (R_1S_1W_{11} + R_1S_2W_{12} + R_1S_3W_{13} + R_1S_4W_{14}) \leq 4$$
$$\Rightarrow (R_2S_1W_{21} + R_2S_2W_{22} + R_2S_3W_{23}) \leq 3$$
$$\Rightarrow (R_3S_1W_{31} + R_3S_2W_{32} + R_3S_3W_{33} + R_3S_4W_{34} + R_3S_4W_{35}) \leq 5$$
$$\Rightarrow (R_4S_1W_{41} + R_4S_2W_{42} + R_4S_3W_{43} + R_4S_4W_{44} + R_4S_5W_{45}) \leq 5$$
$$\Rightarrow (R_5S_1W_{51} + R_5S_2W_{52} + R_5S_3W_{53} + R_5S_4W_{54} + R_5S_5W_{55}) \leq 5$$
$$\Rightarrow (R_6S_1W_{61} + R_6S_2W_{62} + R_6S_3W_{63} + R_6S_4W_{64} + R_6S_5W_{65}) \leq 5$$
$$\Rightarrow (R_7S_1W_{71} + R_7S_2W_{72} + R_7S_3W_{73}) \leq 3$$

Since none of the sub-attributes could be negative values-

$$\Rightarrow \left(K_{ij}W_{ij} \geq 0\right)$$

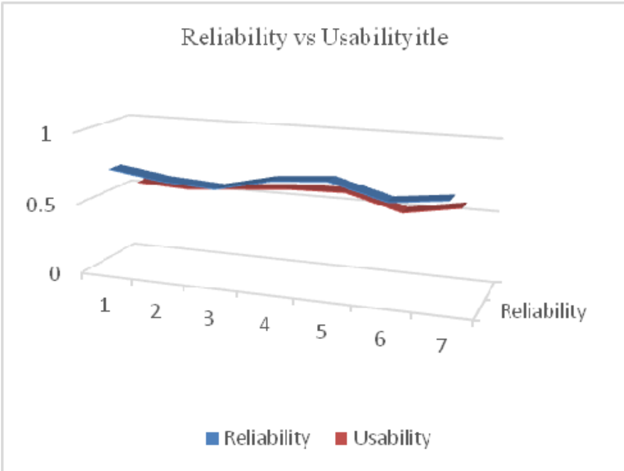

FIGURE 1. Shows relationship between reliability and usability.

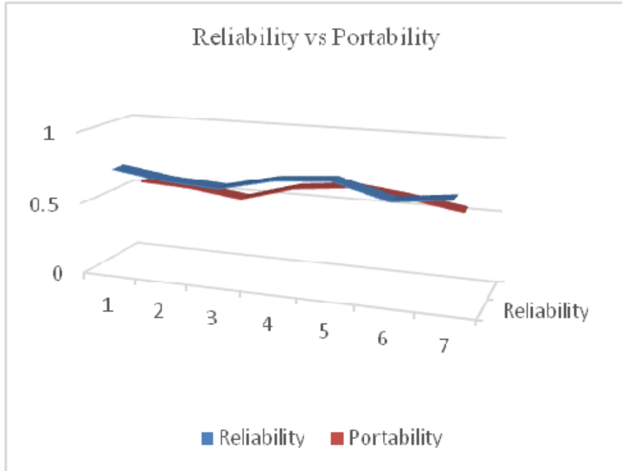

FIGURE 2. Shows relationship between reliability and portability.

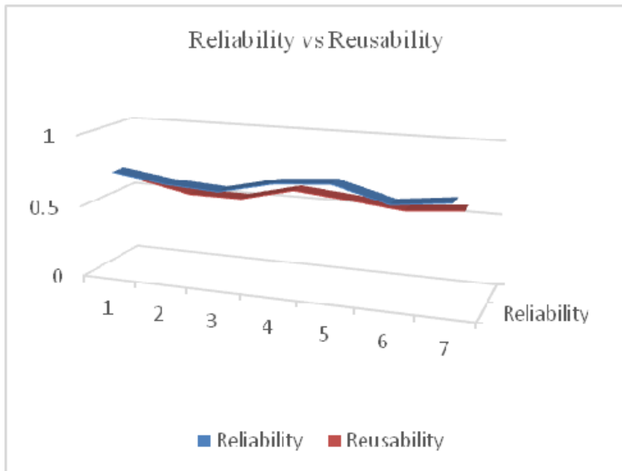

FIGURE 3. Shows relationship between reliability and reusability.

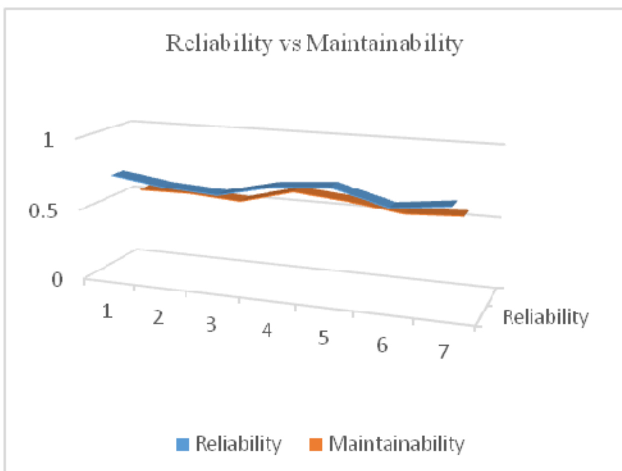

FIGURE 4. Shows relationship between reliability and maintainability.

IV. EXPERIMENTATION AND EVALUATION OF THE OPTIMIZATION MODEL-CASE STUDY

This section has two parts. The first part of the evaluation shows the relationship impact of various quality characteristics on each other. This part of the section applying some mathematical methods like co-correlation coefficient on real data while the second part discussed the MATLAB results for the calculation of the degree of stakeholder satisfaction.

The result showed the limitations of the degree achieved for a particular project or product. To evaluate this model experimentally. We have collected the real data from xLAB IT Consulting Services on their project ''In-Reg Molecule Registration'' to optimize the degree of stakeholder satisfaction factor. The base of the calculated values of different quality attributes is ISO/IEC-9126 software quality assurance model. Optimization model has different parametric equations related to the different software characteristics. The equations have different nature (equality and inequality). Data is based on six different quality attributes and sub-attributes.

A. RELATIONSHIP BETWEEN QUALITY ATTRIBUTES

This study used the statistical analysis method to evaluate the relationship status between various quality characteristics. This work used the Correlation coefficient method to find the nature of the impact of the quality attributes on each other. For the calculation of correlation coefficient researcher has been used the real data of various quality attributes from the industry. The values of quality attributes like reliability, usability, portability, reusability, maintainability, and efficiency are shown in tables 7,8,9,10,11 and 12. respectively. There are five relationship sets present on which the researcher applied the statistical method and after that, the impact of quality attributes shown graphically.

Relation 2:-Reliability vs Portability

According to the constraints and the statistical calculation for the following relationships between the quality attributes using the real data this work get the different impact values for the attribute relationships-

Relation 1: Reliability vs. Usability

After the statistical calculation, the Figure1. shows the relationship between reliability and usability and the value

TABLE 7. Values of reliability and its sub-attributes.

S.No	Sub-attributes			Cumulative Reliability
	Maturity	Recoverability	Fault -Tolerance	
1.	.3241	.3012	.1072	.7325
2.	.2997	.2541	.1258	.6796
3.	.3295	.3006	.0276	.6577
4.	.3116	.2654	.1657	.7427
5.	.3320	.3297	.1102	.7719
6.	.2999	.2844	.0855	.6698
7.	.3881	.2214	.1047	.7142

TABLE 8. Values of usability and its sub-attributes.

S.No	Sub-attributes			Cumulative Usability
	Understandability	Learnability	Operability	
1.	.2215	.2010	.1355	.5580
2.	.2651	.2416	.0348	.5415
3.	.2841	.2553	.0322	.5716
4.	.3110	.2103	.0805	.6018
5.	.2941	.2323	.0844	.6108
6.	.2712	.1997	.0342	.5051
7.	.2541	.2111	.1055	.5707

TABLE 9. Values of portability and its sub-attributes.

S.No	Sub-attributes			Cumulative Portability
	Adaptability	Instability	Co-existence	
1.	.2664	.2177	.0764	.5605
2.	.2746	.1997	.0689	.5432
3.	.2035	.2004	.0927	.4966
4.	.3200	.2449	.0375	.6024
5.	.3007	.2657	.0758	.6422
6.	.2887	.2449	.0655	.5991
7.	.2766	.1887	.0739	.5392

TABLE 10. Values of reusability and its sub-attributes.

S.No	Sub-attributes			Cumulative Reusability
	Correctness	Extensibility	Modularity	
1.	.3115	.2235	.0599	0.5949
2.	.2418	.1674	.1100	.5192
3.	.2559	.1547	.1077	.5183
4.	.2664	.2589	.0847	.6100
5.	.2695	.1649	.1408	.5752
6.	.2335	.2166	.0858	.5359
7.	.2418	.2330	.0964	.5712

TABLE 11. Values of maintainability and its sub-attributes.

S.No	Sub-attributes			Cumulative Maintainability
	Testability	Analysability	Stability	
1.	.2963	.2110	.0390	.5463
2.	.2271	.2003	.1237	.5511
3.	.1997	.1995	.1301	.5253
4.	.3165	.2555	.0529	.6249
5.	.2651	.2110	.1140	.5901
6.	.2777	.2335	.0244	.5365
7.	.2899	.1966	.0702	.5567

of Correlation coefficient is

$$\beta = 0.7356$$

which shows a positive correlation.

Relation 2: Reliability vs Portability

Figure 2. Shows the relationship between reliability and portability and the value of Correlation coefficient is

$$\alpha = 0.7021$$

which shows a positive correlation.

Relation 3: Reliability vs. Reusability

Figure.3 Shows the relationship between reliability and reusability and the value of Correlation coefficient is

$$\gamma = 0.8401$$

which shows a positive correlation.

Relation 4: Reliability vs Maintainability

Figure 4. Shows the relationship between reliability and maintainability and the value of Correlation coefficient

TABLE 12. Values of efficiency and its sub-attributes.

S.No	Sub-attributes		Cumulative Efficiency
	Time Behaviour	Resource Utilisation	
1.	.3119	.1343	.4462
2.	.3401	.1301	.4702
3.	.3621	.1022	.4643
4.	.3220	.1009	.4729
5.	.3321	.1636	.4957
6.	.3021	.0876	.3897
7.	.3157	.1376	.4533

TABLE 13. Optimal values of quality characteristics using MATLAB.

S.No	Variables	$\sum_{i=n}^{j=m} RiSjWij$	Cumulative Value of attributes	Maximum Cumulative value of attribute
1.	x_1			
2.	x_2	2.5202997880	Reliability	4
3.	x_3			
4.	x_4			
5.	x_5			
6.	x_6	2.9629625580	Efficiency	3
7.	x_7			
8.	x_8			
9.	x_9			
10.	x_{10}		Functionality	5
11.	x_{11}	4.999997590		
12.	x_{12}			
13.	x_{13}			
14.	x_{14}			
15.	x_{15}		Maintainability	5
16.	x_{16}	3.2889205150		
17.	x_{17}			
18.	x_{18}			
19.	x_{19}			
20.	x_{20}		Portability	5
21.	x_{21}	3.589659290		
22.	x_{22}			
23.	x_{23}			
24.	x_{24}			
25.	x_{25}		Usability	5
26.	x_{26}	3.426182420		
27.	x_{27}			
28.	x_{28}			
29.	x_{29}	2.9999997120	Reusability	3
30.	x_{30}			

is

$$\delta = 0.7663$$

which shows a positive correlation.

Relation 5: Efficiency vs. Usability

Figure 5. Shows the relationship between efficiency and usability and the value of Correlation coefficient is

$$\varnothing = 0.8648$$

which shows a positive correlation.

B. OPTIMIZATION MODEL RESULTS FOR Q USING MATLAB

Now the researcher using MATLAB to solve this optimization problem. For this, we use the optimization toolbox in MATLAB. It provides widely used algorithms for standard and large-scale optimization. These algorithms solve constrained and unconstrained continuous and discrete problems. The toolkit includes functions for linear programming, quadratic programming, binary integer

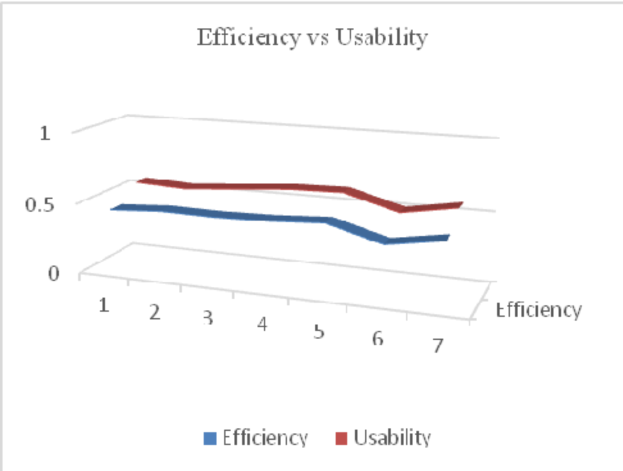

FIGURE 5. Shows relationship between efficiency and usability.

programming, nonlinear optimization, nonlinear least squares, systems of nonlinear equations, and multi-objective optimization.

Objective Function:

$$\text{Min} \quad Q = -\left(\sum_{i=1}^{30} x_i\right)$$

Equality Constraints:

$$(x_1 + x_2 + x_3 + x_4) - .7356 \times (x_{23} + x_{24} + x_{25} + x_{26} + x_{27}) = 0$$

$$(x_1 + x_2 + x_3 + x_4) + .7021 \times (x_{18} + x_{19} + x_{20} + x_{21} + x_{22}) = 0$$

$$(x_1 + x_2 + x_3 + x_4) + .8401\,(x_{28} + x_{29} + x_{30}) = 0$$

$$(x_1 + x_2 + x_3 + x_4) - .7663 \times (x_{13} + x_{14} + x_{15} + x_{16} + x_{17}) = 0$$

$$(x_5 + x_6 + x_7) + .8648\,(x_{23} + x_{24} + x_{25} + x_{26} + x_{27}) = 0$$

Inequality Constraints:

$$(x_1 + x_2 + x_3 + x_4) \leq 4$$

$$(x_5 + x_6 + x_7) \leq 3$$

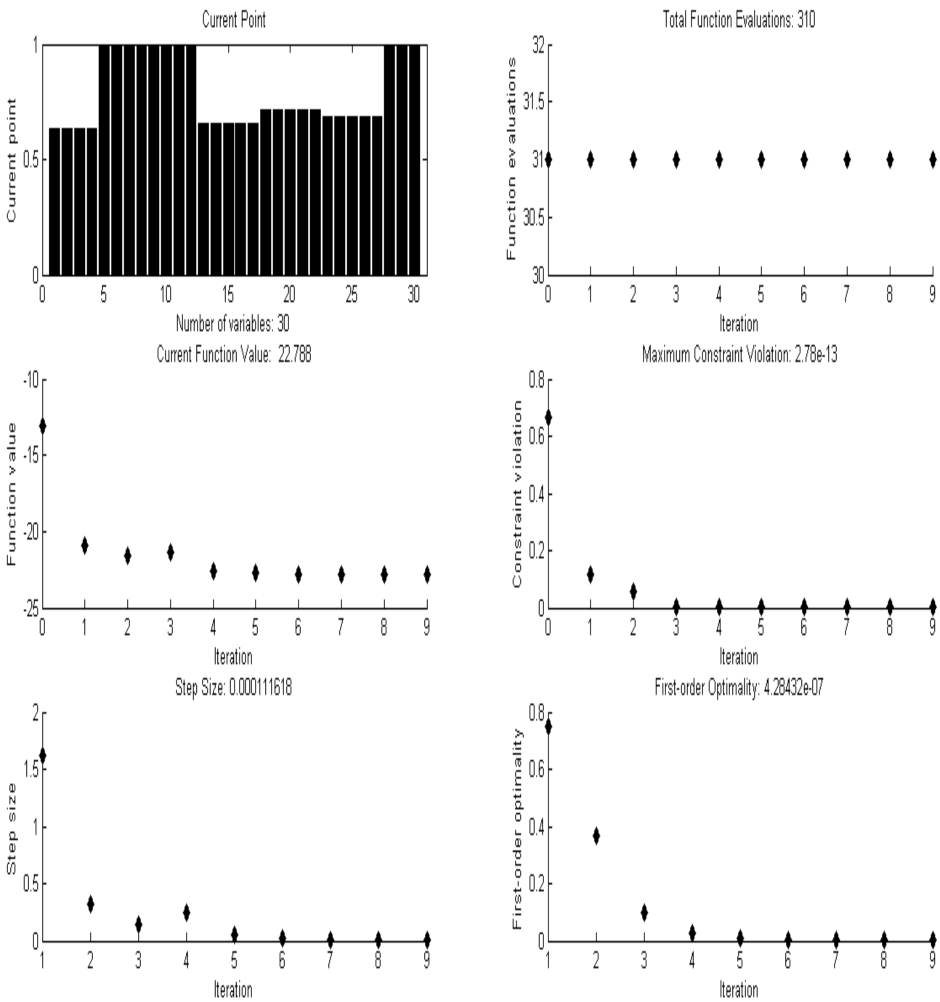

FIGURE 6. MATLAB results for the objective function.

$$(x_8 + x_9 + x_{10} + x_{11} + x_{12}) \leq 5$$
$$(x_{13} + x_{14} + x_{15} + x_{16} + x_{17}) \leq 5$$
$$(x_{18} + x_{19} + x_{20} + x_{21} + x_{22}) \leq 5$$
$$(x_{23} + x_{24} + x_{25} + x_{26} + x_{27}) \leq 5$$
$$(x_{28} + x_{29} + x_{30}) \leq 3$$

Bounds:

$$0 \leq (x_1, x_2, \ldots, x_{30}) \leq 1$$

Result Analysis Using MATLAB:

This study used them to find optimal solutions, perform trade-off analyses of different quality characteristics. Here the study used fmincon solver and interior point algorithm. This work also creates two m-files named optproblem.m and optconst.m, including optimization problem and constraints. And then call the functions in the tool, starting the first solution [0] in vector form, run the tool. To get the following output. Optimization running and given-

- Objective function value: 22.78802200759221, as shown in Figure. 6.
- Local minimum found the satisfied value of the constraints.

MATLAB provides us the optimal value of this objective function. From this, it is easily observed that the quality of this project practically cannot be increased by more than 22.78. Theoretically, it is possible up to 30. MATLAB also provides the optimal value of parameters. The optimal amount of parameters is also provided by MATLAB, as shown in Table 13. If there is any change in the cumulative value of an attribute, then it affects the total value of Q (degree of stakeholder satisfaction) negatively. It means the value of Q will decrease. This work gives a value of Q = 22.788020075; it is the maximum value of the project taken for the experiment. If there is any change in any attribute, then it will decrease the overall cost of Q. This proves that the proposed model is an optimal model for the software quality assurance.

V. CONCLUSION

This study performs a thorough discussion of the results obtained after evaluation of a given mathematical model for the prediction of the degree of stakeholder satisfaction. The results obtained are always an indicator of achievements. These results are also showing that the evaluation of different quality characteristics and proposed optimal model for software quality prediction is successful and can be used further. MATLAB validates evaluation of mathematical model using real data. The idle value of the degree of stakeholder is 30, but after using the optimal model the research is capable of executing an optimal value.

REFERENCES

[1] C. Jones and O. Bonsignour, *The Economics of Software Quality*. Boston, MA, USA: Addison-Wesley, 2011.

[2] N. Fenton and J. Bieman, *Software Metrics: A Rigorous and Practical Approach*. Boca Raton, FL, USA: CRC Press, 2014.

[3] D. Galin, *Software Quality Assurance: From Theory to Implementation*. London, U.K.: Pearson, 2004.

[4] A. Gillies, *Software Quality: Theory and Management*. Morrisville, NC, USA: Lulu Press, 2011.

[5] R. Hackbarth, A. Mockus, J. Palframan, and R. Sethi, "Improving software quality as customers perceive it," *IEEE Softw.*, vol. 33, no. 4, pp. 40–45, Jul./Aug. 2016.

[6] S. Mahmood, R. Lai, Y. S. Kim, J. H. Kim, S. C. Park, and H. S. Oh, "A survey of component based system quality assurance and assessment," *Inf. Softw. Technol.*, vol. 47, no. 10, pp. 693–707, Jul. 2005.

[7] A. Arora, V. S. Arunachalam, J. Asundi, and R. Fernandes, "The Indian software services industry," *Res. Policy*, vol. 30, no. 8, pp. 1267–1287, Oct. 2001.

[8] N. S. Gill and P. Tomar, "Modified development process of component-based software engineering," *ACM SIGSOFT Softw. Eng. Notes*, vol. 35, no. 2, pp. 1–6, Mar. 2010.

[9] B. W. Boehm, J. R. Brown, and M. Lipow, "Quantitative evaluation of software quality," in *Proc. 2nd Int. Conf. Softw. Eng.*, Oct. 1976, pp. 592–605.

[10] A. Seffah, M. Donyaee, R. B. Kline, and H. K. Padda, "Usability measurement and metrics: A consolidated model," *Softw. Qual. J.*, vol. 14, no. 2, pp. 159–178, Jun. 2006.

[11] M. M. M. Kia, J. A. Alzubi, M. Gheisari, X. Zhang, M. Rahimi, and Y. Qin, "A novel method for recognition of persian alphabet by using fuzzy neural network," *IEEE Access*, vol. 6, pp. 77265–77271, 2018.

[12] T. Foken and B. Wichura, "Tools for quality assessment of surface-based flux measurements," *Agricult. Forest Meteorol.*, vol. 78, nos. 1–2, pp. 83–105, 1996.

[13] M. Arif, G. Wang, and V. E. Balas, "Secure VANETs: Trusted communication scheme between vehicles and infrastructure based on fog computing," *Stud. Inform. Control*, vol. 27, no. 2, pp. 235–246, 2018.

[14] L. S. Pontryagin, *Mathematical Theory of Optimal Processes*. Evanston, IL, USA: Routledge, 2018.

[15] T. M. Khoshgoftaar and K. Gao, "Count models for software quality estimation," *IEEE Trans. Rel.*, vol. 56, no. 2, pp. 212–222, Jun. 2007.

[16] M. Li and C. S. Smidts, "A ranking of software engineering measures based on expert opinion," *IEEE Trans. Softw. Eng.*, vol. 29, no. 9, pp. 811–824, Sep. 2003.

[17] N. E. Fenton and M. Neil, "A critique of software defect prediction models," *IEEE Trans. Softw. Eng.*, vol. 25, no. 5, pp. 675–689, Sep./Oct. 1999.

MEHDI GHEISARI received the B.Sc. degree in software engineering from Islamic Azad University, where he was involved in the design and the implementation of automation. He is currently the Ph.D. degree in computer science with Guangzhou University. He is currently doing research on privacy-preserving in the IoT. Prior to that he was with the Islamic Azad University, where he is currently a Lecturer with the Department of Computer Science. There, he was involved in data storage of wireless sensor networks. He proposed an efficient method for data storage in WSN. With a strong background, his research interests include big data, deep learning, the IoT, and privacy. Furthermore, he has published several papers in several domains with his colleagues in highly ranked journals, including *Computers and Security*, *Computer and Electrical Engineering*, *Wireless Network*, the IEEE ACCESS, IJCS, and ETT, and in several ranked conferences.

DEEPAK PANWAR received the master's and Ph.D. degrees in computer science and engineering from Gautam Buddha University, India, in 2011 and 2018, respectively.

He is currently an Assistant Professor with Amity University Rajasthan, India. He has been published many research papers in reputed journals and conferences. His main research interests include the areas of software quality assurance and computational intelligence.

PRADEEP TOMAR received the Ph.D. degree from MDU, Rohtak, India.

He has been an Assistant Professor with the School of Information and Communication Technology, Gautam Buddha University, Greater Noida, India, since 2009. He has good teaching, research, and software development experience and vast administrative experience. He has a teaching experience of about 16 years for wide range of UG/PG level courses. He is also a member of various reputed organizations.

HARSHWARDHAN HARSH received the M.Sc. and Ph.D. degrees from M.D.S. University, Ajmer. He is currently an Associate Professor with the Department of Applied Mathematics, Amity School of Applied Sciences, Amity University Rajasthan. He has a teaching experience of about 16 years for a wide range of UG/PG degrees. He is a life member and office bearers of various academic/research societies/bodies.

XIAOBO ZHANG was born in Suizhou, China, in 1977. He received the master's degree in computer science and the D.Eng. degree in control theory and control engineering. He is currently with the Department of Automation, Guangdong University of Technology, where he is involved in the Internet of Things. He is also a Commissioner with the Guangdong University of Science and Technology. He is also involved in teaching, technology research and development, product design, and scientific theoretical research in the field of industrial control and information technology services. He has participated in more than 60 national and provincial research projects and corporate research projects of the National Natural Science Foundation, the Natural Science Foundation, Guangdong, Major Science and Technology Projects, Guangdong, and a combination of production, education, and research projects of the Ministry of Education, Guangdong. In recent years, there have been more than 30 papers published in academic journals with important influence at home and abroad, including three books as the chief editor. He has applied for more than 40 invention patents and obtained six software copyrights. His main research interests include hardware and software design for the Internet of Things (IoT) and big data, intelligent information processing and intelligent control (robot), and other industrial control systems.

ARUN SOLANKI received the M.Tech. degree from YMCA University, Faridabad, India, and the Ph.D. degree in computer science and engineering from Gautam Buddha University, Greater Noida, India, in 2014, where he has been with the Department of Computer Science, since 2009. His research interests include expert systems, machine learning, and search engines. He has published many research articles in SCI/Scopus-indexed international journals.

ANAND NAYYAR received the Ph.D. degree in computer science from Desh Bhagat University, in 2017, in the areas of wireless sensor networks. He is currently with the Graduate School, Duy Tan University, Da Nang, Vietnam. He has published over 250 research papers in various national and international conferences, and international journals (Scopus/SCI/SCIE/SSCI-indexed). He is a senior and a life member of more than 50+ associations, and an acting ACM distinguished speaker. He has authored/coauthored cum edited 25 books of computer science. He is currently with more than 400 international conferences as programme committee/advisory board/review board member. He is currently involved in the areas of wireless sensor networks, MANETS, swarm intelligence, cloud computing, the Internet of Things, blockchain, machine learning, deep learning, cyber security, network simulation, and wireless communications.

JAFAR A. ALZUBI received the B.Sc. degree (Hons) in electrical, electronics, and communications engineering from the University of Engineering and Technology, Lahore, Pakistan, in 2001, the M.Sc. (Hons.) degree in electrical and computer engineering from the New York Institute of Technology, New York, USA, in 2005, and the Ph.D. degree in advanced telecommunications engineering the Wireless Communications Research Laboratory, Swansea University, U.K., in 2012, where he joined, in 2008.

From 2005 to 2008, he became a full-time Lecturer with the School of Engineering, Al-Balqa Applied University, Jordan. He joined the Wireless Communications Research Laboratory, Swansea University, U.K., in 2008. He is currently an Associate Professor with the Computer Engineering Department, Al-Balqa Applied University, Jordan. He is currently a Visiting Associate Professor with Wake Forest University, NC, USA. He has published more than thirty papers in highly respected journals. He is the managing director of few projects supported by the European union. His research interests include elliptic curves cryptography and cryptosystems, and the classifications and detection of web scams, using algebraic–geometric theory in channel coding for wireless networks. As a part of his research, he designed the first regular and first irregular block turbo codes using algebraic–geometry codes and investigated their performance across various computer and wireless networks.