Journal of Software Engineering and Applications, 2018, 11, 486-499
http://www.scirp.org/journal/jsea
ISSN Online: 1945-3124
ISSN Print: 1945-3116

Empirically Validated Software Efficiency Estimation Model: Product Operation Perspective

Mohammad Zubair Khan[1], Abdullah Alsaeedi[1], Mahfuzul Huda[2]

[1]Department of Computer Science, College of Computer Science and Engineering, Taibah University, Medina, KSA
[2]Department of Computer Science, College of Computing and Informatics, Saudi Electronic University, Riyadh, KSA
Email: zubair.762001@gmail.com, aasaeedi@taibahu.edu.sa, mah.huda@gmail.com

How to cite this paper: Khan, M.Z., Alsaeedi, A. and Huda, M. (2018) Empirically Validated Software Efficiency Estimation Model: Product Operation Perspective. *Journal of Software Engineering and Applications*, **11**, 486-499.
https://doi.org/10.4236/jsea.2018.1110029

Received: August 30, 2018
Accepted: October 23, 2018
Published: October 26, 2018

Abstract

The quality element of a design class is important as it has a substantial listed influence on overall required quality of the software. The complexity and lengthy codes of program severely affect several quality measures, especially efficiency of the final delivered software. This paper does an appropriate review on efficiency of object oriented software, and suggests some relevant information about class design level efficiency. Program efficiency is an important factor to software quality. Moreover, if software efficiency analysis has been done in the early steps of the software creation process then it assures as a criterion to software quality. This article proposes a comprehensive research idea for estimation of the efficiency index. Subsequently, it does an extremely methodical review on design efficiency of software. After that, a design metric based efficiency estimation model at the design time has been proposed by creating the link between design properties of the program and efficiency criteria. Furthermore, the correlation is mapped among all with the help of statistical values for selected design properties and efficiency criteria. In addition, efficiency estimation model is empirically validated with try-out data.

Keywords

Software Quality, Design Phase, Software Efficiency, Object Oriented Development, Software Performance, OOD Metrics

1. Introduction

Nowadays software and application programs are being used increasingly in an extensive variety of applications and areas. Therefore, their correct operations and proper working are very crucial and important for the growth and success of businesses. One of the major concerns of the software and application development industry is to produce and deliver high efficient quality software. Therefore, proper evaluation and estimation of software efficiency has always been a primely important and highly prioritized task for software industry professionals as highlighted [1] [2].

Designing a successful project required up-front, program ongoing financial costs, and indicated end-use needs. Moreover, to be effective in operation, efficient and well organized approaches to designing and developing large-scale program software are required and demanded to an entire software community. Subsequently, performance and utility—, every software development and making project are focused by very common internal factors and related criteria:

- It is must be required from them to provide appropriate product to the users and customers with all listed and required functionalities of program at a minimum condition.
- It is also required product to be delivered on schedule and within financial plan because of urgency and balancing the rate level to be first to demand in market in order to sustain in a competitive edge.
- Moreover, it is also expected from them to meet customer's expectations by delivering product's performance to end users.

Therefore, the performance and efficiency of the program has many influences on the criteria that directly or indirectly affect software and program quality. Enormous code size and complex design of software may lead to poor efficiency index and performance, and in turn series of event result in ineffectual testing and challenging the maintenance process, which may finally result in bad consequences given in program [3] [4]. It is an understood fact that defects of software structures and designs have a strong adverse effect on quality attributes and related criteria. Structuring and developing a high quality complex design endures to be an inefficiently initially defined requirement and process. Therefore, software and program design must be simple and minimally complex; this minimizes the required effort for the unforeseen future functional extensions. Moreover, the design is alienated into functionally separated and less interdependent modules of reasonable program size [5]. The main objective of this research work is to propose a complete model for efficiency estimation of the program performance that computes efficiency indices at the design time of the product development life cycle process. In view of this fact, research work proposes an "efficiency estimation framework", for object oriented program design in product operation perspective, which is well explained and summarized in **Figure 1**.

Proposed framework for efficiency estimation is structured in such a way that it primarily describes the efficiency as an important quality factor of any software program. Then lists and describes program's design properties and related design metrics for the object oriented paradigm. For efficiency estimation of program, process tries to develop a model to find efficiency indices. Subsequently, check and compare all calculated measures to commercial programs and product design. For statistical significance of gained values and better acceptability of developed model, research is showing and highlights all information accordingly.

Nomenclature

A OOD: Object-oriented design

B OO: Object-oriented

C OOP: Object-oriented programming

2. Software Design Efficiency: A Quality Factor

The Institute of Electrical and Electronics Engineers (IEEE) defines software efficiency as the performance and intuitive capability of a program to must perform its designated and expected functions under its stated condition. A software is efficient then the whole component and module of a product design do expected operation within a specified condition [6] [7]. Efficiency is well recognized as an important attribute to quality index evaluation of software, which gives privilege to the inclusion and insertion of performance. In a product and layout design, it notifies an early emphasis to developed good quality and high performance software in the early phase and process of the development life cycle [8]. The given table in this manuscript is tries to explain, the efficiency as an important measure and quality element in different standard program quality model [9] [10] [11].

In general, the measure of efficiency estimation of a program by developed model is to deliver a good quality and high efficient software that is effective in function. Moreover, it also gives an attention about performance at early step in the program or module development cycle. Additionally, supports to deliver required software within specified condition by the users that is mapped time and confined estimate. Because high performance and on time delivery of a good quality software is product's essential factor that finally increases their project success rate.

3. Related Work

Several past studies defined software efficiency as an important factor of the quality. In this segment, study discusses the contents of the efficiency in given some well-known software quality models like, McCall quality model, Boehm quality model, FURPS quality model, Dromey quality model, ISO 9000 Quality-Model and ISO/IEC 25010 Quality Model. All experts in their quality model and work defined it, as the measures to which the software product design provides appropriate product performance, related to the mentioned amount of resources used, under customized or stated conditions.

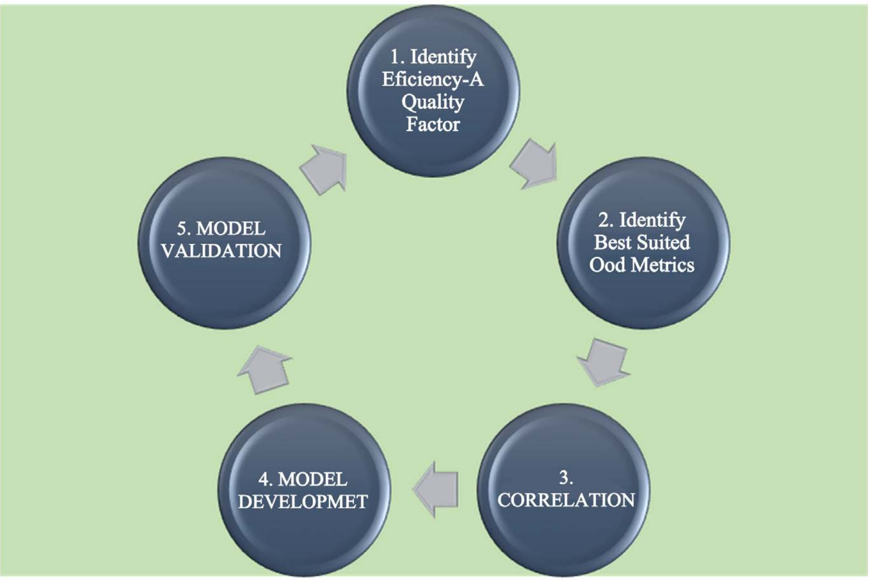
1. Identify Efficiency-A Quality Factor
2. Identify Best Suited Ood Metrics
3. CORRELATION
4. MODEL DEVELOPMET
5. MODEL VALIDATION

Figure 1. Research framework.

3.1. McCall's Quality Assurance Model Area

In this model, program efficiency shows a level of product's performance to describe a process that uses the least inputs to create the maximum output. Subsequently, the McCall's quality model of software has focus on three main quality of software given as follow. First, Product Transition: That is used for interoperability and adaptability nature of the program, when migrating to new environment or new platform. Second, Product Revision: The module and program ability to undergo required changes whenever the basic requirement of users have been changed. Third is Product Operations: It is an operational characteristic of the program. In proposed model efficiency taken as the key part of product operations along with Correctness, Reliability, Integrity and Usability.

Subsequently, model contains eleven software quality factors with twenty-three related quality criteria. Where the used quality factors of the program describe different types of software properties and quality criteria of that module are valuable characteristics to one or more than one quality factors [12]. **Table 1**, is represented as the efficiency quality factor of the program and related criteria of program efficiency in the McCall software quality assurance model.

3.2. Boehm Quality Model

Barry W. Boehm, American software engineer produces Boehm quality model. His developed model for the program quality, attempts to hypothetically evaluation of the quality of whole software. Boehm defines "if efficiency index estimated at the software development time it will be helped to enhance the quality." Further in his claims, he pointed, the product high level factor efficiency; addresses and explains three classification; accountability, device efficiency, accessibility [13] [14]. **Table 2**, is presented as the efficiency quality factor and efficiency quality criteria of Boehm quality model.

Table 1. McCall's quality assurance model.

Quality factor	Quality criteria
Efficiency of Product	Concision Operability Execution Performance Resource utilization

Table 2. The efficiency quality factor and its quality criteria in Boehm quality model.

Factor	Criteria
Efficiency of the product	Accountability Design efficiency Accessibility Resource utilization

3.3. Dromey's Quality Model

Dromey's program quality model proposed a theoretical framework to evaluate quality at requirement, design step and implementation step. He claims internal product design depends on efficiency measures along with maintainability and reliability. In his proposed model efficiency showing as an important criteria of the high level internal design properties for the implementation of the product quality model [15]. **Table 3** presented efficiency as a quality criterion along with maintainability and reliability of internal product design model in Dromey's quality model factors. Further, in his study efficiency of the design supports resource utilization and accountability.

3.4. FURPS Quality Model

FURPS software quality model formerly presented by quality expert Grady R.B and extended by IBM Software Rational into FURPS+. Where the plus "+" indicates, the desired requirements as the form of design characteristics and constraints, requirements, implementation and execution in the form of interface requirement and physical requirement. Model claims about efficiency, how well is the software doing the work it is supposed to do? In this model efficiency showing as the measure of product's performance along with some others criteria as follows response time, throughput, resource usage, recovery time, availability, accuracy [5]. **Table 4** presented FURPS's quality model where efficiency selected as the criteria of the product performance.

3.5. ISO/IEC 25010 Quality Model

The ISO 25010 is the widely used quality standard model in present time. ISO 25010 uses efficiency as one of the main quality factor along with nine others quality factors namely as operability, functional, suitability, reliability, security, performance, compatibility, maintainability, and portability. Moreover, 28 quality criteria are arranged in this model for all ten-quality factors given in [16] [17]

[18]. Table 5 presented efficiency as the quality factor and its related quality criteria given in ISO/IEC 2510 quality model.

3.6. Critical Findings on Efficiency

From study on the above quality models and ISO/IEC 25010 standard, research could define, efficiency as an important factor of the product performance and would be well defined by the listed criteria like reliability, resource utilization, time behavior, accuracy given in Table 6. It is advisable that quality criteria are the internal characteristics, which properly define the modules and programs quality factors and associated properties. The high-level software efficiency characteristic addresses five important classifications; Execution, Accountability, Design efficiency, Performance of the product and Resource utilization. Table 7 is denoted as a comparison of the efficiency's criteria in four quality model and ISO/IEC 25010.

Table 3. Efficiency showing as quality criteria in Dromey quality model.

Factor	Criteria
Internal Product Design	Maintainability Efficiency Reliability

Table 4. Efficiency as a quality measure of the software in FURPS quality model.

Factor	Criteria
Performance of the Product	Speed Efficiency Response Time Throughput Resource Usage Recovery Time Design Accuracy

Table 5. Efficiency and its measures in ISO/IEC 25010 quality model.

Factor	Criteria
Performance Efficiency of the product	Time behavior Resource utilization Design Accountability

Table 6. The definition of efficiency's finalized criteria.

Efficiency's Criteria	Definition
Execution	Execution supports to carrying out of an outline, design, usage, request, or strategy…
Performance	Procedures that may decide customer fulfillment
Accountability	Accountability of a plan is a measure that depicts the fundamental components of principled

Continued

Design efficiency	Design is a measure of how assets/inputs (reserves, ability, time, and so forth.) are changed over into outputs.
Resource utilization	Were the resources and input changed over to output in an opportune and financially well informed way?

Table 7. Efficiency's criteria in four quality model and ISO/IEC 25010.

Efficiency's Criteria	McCall	Boehm	Dromey	FURPS	ISO/IEC 25010
Time behavior	√	√			√
Operability	√	√			
Execution	√	√		√	√
Accountability		√	√	√	√
Design efficiency		√	√	√	√
Accessibility		√			
Internal Product Design			√		
Performance of the product	√		√	√	√
Concision					√
Resource utilization	√	√	√		√
Capacity					√

4. Object Oriented Design Properties Factor

In software designing and developments process, object-oriented concept is becoming highly preferred and ideal approach for commercial software design and development process. This paradigm allows for producing quality-oriented software product with high reliability and lesser product maintenance costs [19] [20]. Classes in a program of the object-oriented software development system provide a complete structuring standard that gives good performance in a design to be separated into proper designed small modules. One of the foremost and prime benefit of having object orientation concept in the development process of a program is its support for software maintainability.

Object oriented principles direct the program's developers and the software designers what to evade and what to care. Numerous guidelines have been mentioned in this approach or we can say techniques so far to design and evaluate object oriented concepts. These thoughts meaningfully comprise program's design artifacts viz. inheritance, polymorphism, encapsulation, coupling and cohesion. Encapsulation properties is a technique to use for understand data abstraction and design details hiding from modules. Moreover, it skins inner description of objects. Coupling applies more to classes as a major aspect of a design that decreased support and change costs. Cohesion metrics measure bolsters how well the strategies for a class are related with each other. Polymorphism supports easily assess and enhance the nature of the software product.

Object Oriented Design Metrics

In software design and development process, a metric is the measurement of a particular characteristic of an object oriented properties that useful for program's desired performance as well as efficiency. Several conceptual research work in the object oriented design metrics field were completed in past years [21]. Most of the design or implementation metrics are recognized by researchers and practitioners on highly practices, claim and acceptance given in [22] [23] [24] [25]. This ran to the explanation of five novel metrics, Design level Encapsulation Metrics (ENM), Design level Coupling Metrics (DCC), Design level Cohesion Metrics (COM), Design level Inheritance Metrics (MFA), and Design level Polymorphism (POL) [26], which might be evaluated from design level information only. A significant group of design metrics used for casing all the product design level characteristics. Chosen metrics have been given and explained in Table 8.

5. Mapping between Design Properties and Efficiency's Criteria

McCall's quality model for product operation has been considered as a basis to develop the efficiency estimation model for software at the design time of the development process. In this module, study is establishing the relationship and linking between program or software design properties and its efficiency's criteria as shown in the following Figure 2, which involves following steps:

1) Identification of object oriented design properties.
2) Identification of object oriented design Metrics.
3) A means of linking of design properties to design Metrics.

6. Model Development for Software Design Efficiency Factor

For development of the software efficiency mode, research uses the concept of multiple linear correlation technique. Used multivariate linear correlation model between dependent and independent values is given as. For development of the software efficiency mode, research uses the concept of multiple linear correlation technique and the used multivariate linear correlation model between dependent and independent values is given as follows.

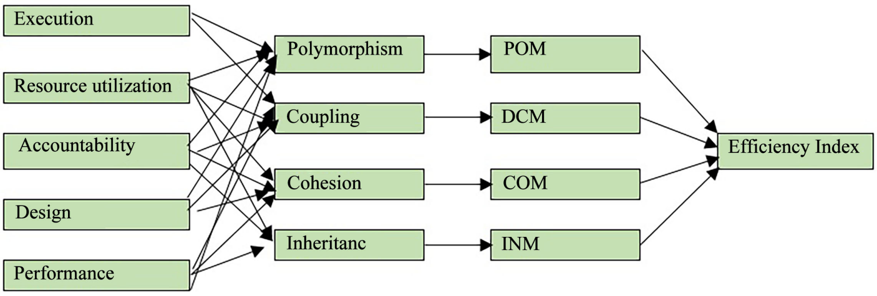

Figure 2. Mapping between design properties and efficiency's criteria.

Table 8. Selected metrics description.

Metrics	Description	Definition
ENM	Encapsulation Metric	Greater encapsulation in design increases reliability and reduces complexity of design.
DCM	Design Coupling Metric.	It applies more to classes as a major aspect of a design that decreased support and change costs.
COM	Cohesion Metric	Cohesion metrics measure bolsters how well the strategies for a class are related with each other.
POM	Polymorphism Metric	The metric takes data originating from the primary phases of the advancement procedure giving engineers the chance to early assess and enhance the nature of the software product.
INM	Inheritance Metric	The inheritance metrics give us data about the inheritance tree of the framework.

$$Y = a_0 + a_1 x_1 + a_2 x_2 + a_2 x_2 \cdots a_n x_n \tag{1}$$

Moreover, when put the values of design properties in the above equation, the equation appears like

$$\text{Efficiency} = a_0 + \text{Coupling}_1 x_1 + \text{Coupling}_2 x_2 + \text{inheritance}_3 x_3 + \text{polymorphism}_4 x_4$$

The Model development data is taken from [27] that have been collected through the class diagrams. It includes a set of twenty class diagrams (denoted as Project 1 to Project 20) and the metrics value of every class diagram. A group of ten software experts having 10 to 12 years' experience in commercial software development have devolved software-using C++. Furthermore, the class diagram's mean values rating of efficiency given by the quality experts are also used in this research paper. The relationship among efficiency criteria's viz. Reliability, Resource utilization, Time behaviour, Accuracy, Performance and Object Oriented design properties have been established, as illustrated in Figure 1. Using Statistical Package for the Social Sciences (SPSS) and matrix laboratory (MATLAB), values of model's coefficient for dependent and independent variables are calculated and finally, efficiency model is formulated as given below.

$$\begin{aligned} \text{Efficiency} = 7.442 - 0.459 * \text{coupling} - 0.040 * \text{cohesion} \\ - 0.677 * \text{inheritance} + 1.128 * \text{polymorphism} \end{aligned} \tag{2}$$

Table 9 shows the coefficient parts for Efficiency estimation quality model. The Unstandardized coefficients component of the Table 8 gives us the coefficient values that study require in order to develop above efficiency Equation (2). The experimental assessment of efficiency is very hopeful to get efficiency index of object oriented software (Figure 3).

Efficiency Model Summary

The Model Summary Table 10 result is most helpful when performing dependent and independent multiple regressions approach. Here, capital R is the determinant's coefficient that tells how powerfully the multiple selected independent variables are associated to the proposed dependent variable. Moreover, R

Square is a model's supportive value as it gives the very important coefficient of selected determination. Subsequently, adjusted R-square provides prospects to adjust number of independents in this model. In addition, the standard error of the estimate is a value of the model accuracy of predictions made with a regression model.

7. Empirical Validation of the Developed Model

The No matter, how influential a hypothetical outcome may be, it has to be scientifically authenticated if it is working to be of any real-world use. This is factual in all Industrial disciplines, including Software Engineering. Consequently, in adding to the hypothetical or we can say hypothetical validation, an investigational tryout validation is similarly significant in order to make the claim higher adaptable. In assessment of this important truth, a rigorous investigational authentication process of the developed efficiency model given in equation no. 2 has been performed with the support of design level metrics specified in the data set. As per the association among object oriented features and efficiency key factors represented in Figure 1, and the efficiency Model, the given metrics values were used to compute the efficiency of design level diagrams. Summary of the values calculated by the developed model against the given values of efficiency by the commercial software developers, are given in Table 11.

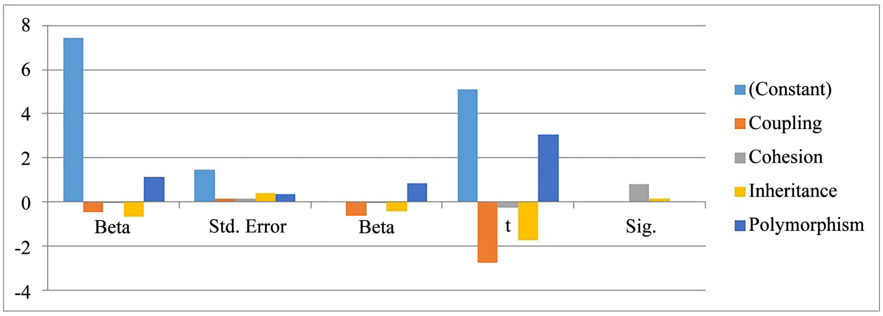

Figure 3. Graphical representation of model's coefficients.

Table 9. Coefficients for efficiency evaluation model.

Coefficients[a]						
Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	7.442	1.456		5.112	0.004
	Coupling	−0.459	0.168	−0.644	−2.738	0.041
	Cohesion	−0.040	0.165	−0.049	−0.242	0.818
	Inheritance	−0.677	0.393	−0.402	−1.721	0.146
	Polymorphism	1.128	0.370	0.866	3.050	0.028

a. Dependent Variable: Efficiency.

Table 10. Efficiency estimation model summary.

Model	R	Adjusted R Square	R Square	Std. Error of the Estimate
1	0.911[a]	0.693	0.829	0.31066

a. Predictors: (Constant), Polymorphism, Cohesion, Inheritance, Coupling.

Table 11. Efficiency measures estimated by developed model.

Object Oriented Projects	Efficiency Index by Developed Model
Project 1	6.24
Project 2	6.91
Project 3	4.31
Project 4	7.75
Project 5	7.90
Project 6	4.56
Project 7	3.92
Project 8	2.40
Project 9	5.54
Project 10	2.69

Efficiency measures for ten projects given by the experts for common selected design metrics are as follows: Project 1: 9.30, Project 2: 8.5, Project 3: 7.4, Project 4: 8.6, Project 5: 9.6, Project 6: 7.9, Project 7: 8.3, Project 8: 6.9, Project 9: 7, Project 10: 6.8.

Spearman's Rank Correlation coefficient r_s was used to test the efficiency model's significance of correlation between calculated values by developed model of efficiency and given indices for the program efficiency as given values. The Correlation coefficient "r_s" was calculated using the below given formula: under: Spearman's Coefficient of Correlation (r_s)

$$r_s = 1 - \frac{6\sum d^2}{n\left(n^2 - 1\right)} \quad -1.0 \le r_s \le +1.0 \tag{3}$$

where, small "d" is the difference between calculated efficiency rank and expert's known rank of efficiency quality factor and "n" is the number of total software projects used in the experimentation. The validation data used for model validation is taken from [27] that have been collected through the class diagrams. It includes a set of twenty class diagrams (denoted as Project 1 to Project 20) and the metrics value of every class diagram. A group of ten software experts having 10 to 12 years' experience in commercial software development have devolved software-using C++ (**Figure 4**).

Spearman's Rank Correlation Coefficient r_s = 0.84242 is calculated for the proposed model is more than the threshold value for 10 projects. Results show that the calculated values of efficiency quality factor using developed model

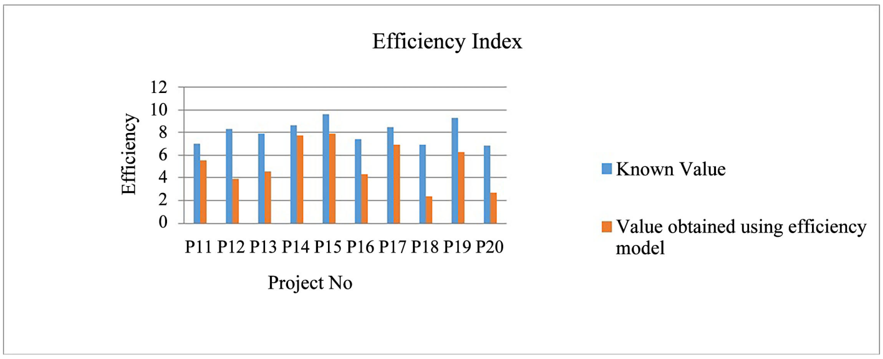

Figure 4. Graphical representation of index comparisons.

(given in Equation (2)) are highly correlated with the given values. Thus, the correlation between variables in the proposed model is acceptable with the high degree of confidence level, *i.e.* at the 0.05. The correlation is up to standard with high level of relation confidence *i.e.* at the 95%. Therefore, without any further loss of generality, we can conclude that presented efficiency model (Equation (2)) estimates and indices are reliable and valid in the context. However, the research study needs a larger set of experimental tryout on developed efficiency model for better model acceptability and utility.

8. Conclusion and Future Work

Model's equation in this research work shows the significance of efficiency and its correlations with design characteristics namely, encapsulation, coupling, inheritance and design polymorphism. Further, study developed an efficiency estimation model with the support of multiple linear regression method on object-oriented design properties. Arithmetical results confirm that efficiency model is extremely significant and up to standard. Software efficiency is dynamic and one of the substantial components of the program and system development nowadays. Above described five quality attributes, have many highlighted possessions in shared, counting low level module coupling, modularity and high level module cohesion. Design efficiency is directly increased reusability, flexibility, changeability, maintainability and scalability and denotes to system elements' capability to design for numerous type software product, which is used by the software design and development. Efficiency composed with quality attributes permits a technology to be moved to another software project with fewer design and development cost and time, as well as improved performance and reliability of the software. The realistic validation on the efficiency model will be completed in future on live industrial software projects for improved and broad acceptability and high-level utility.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Huda, M., Arya, Y.D.S. and Khan, M.H. (2015) Quantifying Reusability of Object Oriented Design: A Testability Perspective. *Journal of Software Engineering and Applications*, **8**, 175-183. https://doi.org/10.4236/jsea.2015.84018

[2] Abdullah, Dr, Srivastava, R. and Khan, M.H. (2014) Modifiability: A Key Factor to Testability. *International Journal of Advanced Information Science and Technology*, **26**, 62-71.

[3] Drown, D.J., Khoshgoftaar, T.M. and Seiya, N. (2009) Evaluation Any Sampling and Software Quality Model of High Assurance Systems. *IEEE Transaction on Systems, Mean and Cybernetics, Part A: Systems and Human*, **39**, 1097-1107. https://doi.org/10.1109/TSMCA.2009.2020804

[4] Abdullah, Khan, M.H. and Srivastava, R. (2015) Flexibility: A Key Factor to Testability. *International Journal of Software Engineering & Applications* (*IJSEA*), **6**, 89-99. https://doi.org/10.5121/ijsea.2015.6108

[5] Grady, R.B. (1992) Practical Software Metrics for Project Management and Process Improvement. Prentice Hall, Englewood Cliffs.

[6] Huda, M., Arya, Y.D.S. and Khan, M.H. (2015) Metric Based Testability Estimation Model for Object Oriented Design: Quality Perspective. *Journal of Software Engineering and Applications*, **8**, 234-243. https://doi.org/10.4236/jsea.2015.84024

[7] Huda, M., Arya, Y.D.S. and Khan, M.H. (2015) Testability Quantification Framework of Object Oriented Software: A New Perspective. *International Journal of Advanced Research in Computer and Communication Engineering*, **4**, 298-302. https://doi.org/10.17148/IJARCCE.2015.4168

[8] McCable, T.J. (1976) A Complexity Measure. *IEEE Transaction on Software Engineering*, **SE-2**, 308-320. https://doi.org/10.1109/TSE.1976.233837

[9] Huda, M., Arya, Y.D.S. and Khan, M.H. (2014) Measuring Testability of Object Oriented Design: A Systematic Review. *International Journal of Scientific Engineering and Technology* (*IJSET*), **3**, 1313-1319.

[10] Vennila, G., Anitha, P., Karthik, R. and Krishnamoorthy, P. (2011) A Study of Evaluation Information to Determine the Software Quality Assurance. *International Journal of Research and Reviews in Software Engineering*, **1**, 1-8.

[11] Tomar, A.B. and Thakare, V.M. (2011) A Systematic Study of Software Quality Models. *International Journal of Software Engineering & Application*, **12**, 61-70. https://doi.org/10.5121/ijsea.2011.2406

[12] McCall, J.A., Richards, P.K. and Walters, G.F. (1977) Factors in Software Quality, RADC TR-77-369. Rome Air Development Center, Rome.

[13] Boehm, B.W., Brow, J.R., Lipow, M., McLeod, G. and Merritt, M. (1978) Characteristics of Software Quality. North Holland Publishing, Amsterdam.

[14] Boehm, B.W., Brown, J.R. and Lipow, M. (1976) Quantitative Evaluation of Software Quality. *Proceeding of the* 2nd *International Conference on Software Engineering*, San Francisco, 13 October 1976, 592-605.

[15] Dromey, R.G. (1996) Concerning the Chimera (Software Quality). *IEEE Software*, **1**, 3343.

[16] ISO (2005) 9001:2005, Quality Management System Fundamentals and Vocabulary.

[17] ISO (2001) 9001:2001, Quality Management System Requirements.

[18] ISO (2011) IEC25010: Software Engineering—System and Software Quality Requirement and Evaluation (SQuaRE)—System and Software Quality Model.

[19] Kemerer, C.F. (2008) An Empirical Validation of Software Code Estimation Models. *Communications of the ACM*, **30**, 416-429. https://doi.org/10.1145/22899.22906

[20] Basili, V.R. and Weiss, D.M. (1984) A Methodology for Collecting Valid Software Engineering Data. *IEEE Transactions on Software Engineering*, **SE-10**, 728-738. https://doi.org/10.1109/TSE.1984.5010301

[21] Pandian, C.R. (2004) Software Metrics—A Guide to Planning, Analysis, and Application. CRC Press Company, Boca Raton.

[22] Jacobson, I., Booch, G. and Rumbaugh, J. (1999) The Unified Software Development Process. Addison Wesley, Boston.

[23] Kemerer, C.F. and Porter, B.S. (1992) Improving the Reliability of Function Point Measurement: An Empirical Study. *IEEE Transactions on Software Engineering*, **18**, 1101-1024. https://doi.org/10.1109/32.177370

[24] Khatibi, V. and Jawawi, D.N.A. (2011) Software Cost Estimation Methods: A Review. *Journal of Emerging Trends in Computing and Information Sciences*, **2**, 21-29.

[25] Alshayeb, M. (2009) Empirical Investigation of Refactoring Effect on Software Quality. *Information and Software Technology*, **51**, 1319-1326. https://doi.org/10.1016/j.infsof.2009.04.002

[26] Huda, M., Arya, Y.D.S. and Khan, M.H. (2015) Evaluating Effectiveness Factor of Object Oriented Design: A Testability Perspective. *International Journal of Software Engineering & Applications*, **6**, 41-49. https://doi.org/10.5121/ijsea.2015.6104

[27] Dexter, M. (2016) Software India Pvt. Ltd.