

Quantifying Reusability of Object Oriented Design: A Testability Perspective

Mahfuzul Huda¹, Yagya Dutt Sharma Arya¹, Mahmoodul Hasan Khan²

¹Department of Computer Science & Engineering, Invertis University, Bareilly, India

²Department of Computer Science and Engineering, IET, Lucknow, India

Email: mahfuzul@iul.ac.in

Received 16 March 2015; accepted 13 April 2015; published 14 April 2015

Copyright © 2015 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

The quality factor of class diagram is critical because it has a significant influence on overall quality of the product, delivered finally. Testability analysis, when done early in the software creation process, is a criterion of critical importance to software quality. Reusability is an important quality factor to testability. Its early measurement in object oriented software especially at design phase, allows a design to be reapplied to a new problem without much extra effort. This research paper proposes a research framework for quantification process and does an extensive review on reusability of object oriented software. A metrics based model “Reusability Quantification of Object Oriented Design” has been proposed by establishing the relationship among design properties and reusability and justifying the correlation with the help of statistical measures. Also, “Reusability Quantification Model” is empirically validated and contextual significance of the study shows the high correlation for model acceptance. This research paper facilitates to software developers and designer, the inclusion of reusability quantification model to access and quantify software reusability for quality product.

Keywords

Reusability, Testability, Object Oriented Design, Design Metrics, Object Oriented Software, Software Quality Model, Software Testing, Effort

1. Introduction

Significant efforts have been made in the field of software measurement for improving the quality of product and most of them pursue the goal of the quality of final product, *i.e.* later phase in system development life cycle [1]. In software engineering it is widely accepted that the quality of finally delivered product is highly dependent

on the early decision in the development process [2]. An accurate evaluation of software quality depends on testability estimation, which in turn depends on the factor that can affect testability. Reusability is strongly associated to testability and continuously plays an important role to deliver best quality and high class software within fixed time and given budget [3]. Reusability factor is not new; Mxllroy defines “reusability factor” has been behind many software developments for product’s quality. Reusability plays an important role in software quality assessment at design time and also acts as the basis to find testability indices for industry project’s ranking. The basic criterion for evaluating testability factor is reusability. If a component is not reusable then the whole concept of component based software development fails. The definitions of reusability from various researchers are summarized in **Table 1**.

Software design is a method in software engineering that produces a model of the system and allows system designers/developers to rotate your computational model into a workable algorithm that resolve the problems by the help of qualitative and quantitative assessment [10]. ISO 9126 quality model defines: “level of quality and performance of design achieves expected functionalities by reusability factor [11]”. A reusability criterion constantly supports designer for improved software design at an early stage of software development process. Design phase has direct impact on the product testing cost and effort and plays the backbone role of any product. To design and develop a good quality product, reusability plays a key role for assessment of software testability. Estimating reusability factor early in the development life cycle may greatly reduce the overall product development cost and enhance the customer satisfaction. The object oriented design principle is a suitable language for generalized productivity. The objective of this study is to produce a model that quantifies reusability at design instant of development life cycle. In this regard, research proposes a “Reusability Quantification Framework”, which is summarized in **Figure 1**.

This paper is structured in such a approach that it initially describes the reusability: an important factor to testability for quantification process and then lists and describes reusability quantification model with their correlation establishment and also highlights information for statistical significance of the developed model that are significant for further study. The developed model has been empirically validated; furthermore, the research paper concludes with software industry utility of model for project leveling in conclusion section.

2. Reusability: A Testability Perspective

Reusability is recognized as a very important factor to testability evaluation of object oriented software, which consent to the inclusion of changes in a design and gives an early focus to create testable software within development life cycle. The IEEE defines software reusability as the capability of a system or component to perform its required function under its specified condition that means “a component is reusable then the whole component of product design do not required to change [12]”. **Table 2** is denoted as the reusability is an important criterion in different quality model.

In general the overall objective of the reusability quantification is to deliver good quality software that is effective in function, easily approachable to user and permit the incorporation of changes in a product design and gives an early focus to deliver testable software within specified condition *i.e.* fixed time and bounded budget because delivering good quality product is no longer an advantage, but a essential factor. Objective of this study

Table 1. Researcher definition.

Name of Researcher	Definition
Freeman [4]	“Reuse is the use of any kind of information which a developer may need in the software creation process”.
Basili and Rombach [5]	“Software reuse is the use of everything related with software with a software project, including knowledge”.
Tracz [6]	“Reusability is the use of software specification that was designed for reuse”.
Braun [7]	“Use of existing software components in a new perspective, either in the same system or in another system”.
Cooper [8]	“Reuse is the ability of a previously developed software component to be used again or used repeatedly in part or in it’s entirely, with or without modification”.
Krueger [9]	“Reuse is the process of developing software systems from existing software rather than building them from scratch”.

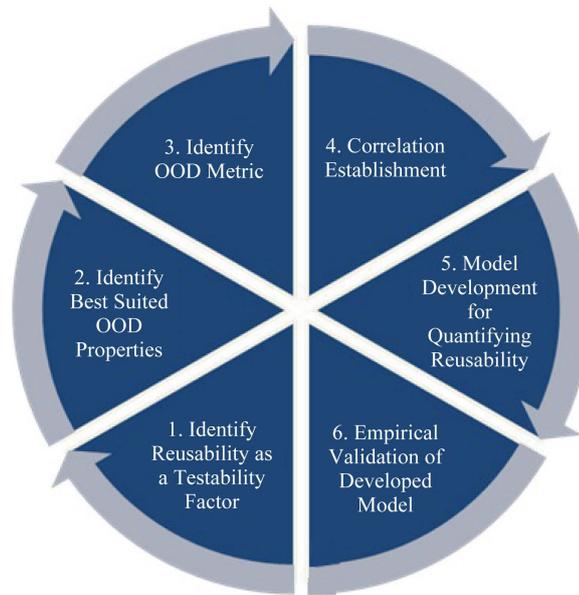


Figure 1. Research framework for reusability quantification process.

Table 2. Reusability quality criteria in different quality model.

Quality Model	Definition
McCall quality model [13]	This quality model has three quality of software product: according to this model “Reusability quality factor” show the quality of product as a form of product transition that means adaptability to new environment with the help of generality and self documentation quality criteria.
Dromey quality model [14]	Dromey proposed a framework for evaluate requirement design and implementation phase. The high level software product properties for the implementation include ‘reusability as the quality criteria of Dromey quality mode.
Boehm quality model [15]	Boehm quality model used reusability as the quality factor to accomplish the completeness criteria to automatics and qualitative evaluate the quality of software.
FURPS model [16]	FURPS model extended by IBM Rational software used “reusability as the quality factor” in FURPS QUALITY MODE to accomplish the consistency and documentation quality criteria.

to evaluate software reusability by using the concept of software quality assessment during the initial stage in development life cycle. Here research is required to develop a structured approach to make sure that software is reusable, effective and high quality.

3. Object Oriented Design Properties

In software development environments, object oriented designing and development is becoming very preferred paradigm in industrial software development environments for large-scale system design. This technology offers support to deliver final software product with higher quality and lower maintenance costs [17]. Classes in object oriented design system provide an excellent structuring principle that allows a structure to be divided into well designed units which may then be implemented separately. One of the major advantage of having object orientation is its support for software reusability, which may be achieved either through the simple reuse of a class in a library or via inheritance among relationship. Object oriented principles direct the designers what to avoid and what to support. Several procedures have been defined in this method so far to estimate object oriented design. There are several essential qualities of object orientation that are recognized to be the basis of internal qualities of object oriented design that support in the context of measurement. These ideas significantly include design properties *i.e.* encapsulation, inheritance, coupling, and cohesion. Encapsulation is a mechanism to realize data abstraction and information hiding. It hides internal specification of an object. Inheritance is the distribution of

operations and attributes among classes. Coupling indicates the relationship or interdependency between modules. Cohesion refers to the internal reliability inside the components of the design. A class is cohesive when its components are greatly interrelated and it should be difficult to divide a cohesive class. Cohesion can be used to recognize the defectively designed classes. **Table 3** summarize as the design properties and their advantages.

4. Object Oriented Design Metrics

Several research works in the object oriented metrics arena were done in recent years [3]. Most of the metric is accepted by practitioners on “heavy usages and popularity” and by academic experts on empirical (post development) validation. This led to the definition of four new metrics, Encapsulation Metrics (ENM), Coupling Metrics (DCC) and Cohesion Metrics (COM), Inheritance Metrics (MFA) which could be calculated from design information only. A suite of object-oriented metrics, covering all the design attributes, has been listed and described in **Table 4**.

5. Correlation Establishment

The correlation establishment among Reusability, OOD Properties and Metrics, describes the quantification process of reusability model in order to establish a multivariate linear model for reusability and OOD constructs. The values of these design metrics can be identified by class diagram metrics. Identified metrics will take part in the role of independent variables while reusability will be taken as dependent variable. Evaluation of reusability is very helpful to get testability index of software design for high quality product as early as possible in development life cycle. After rigorously evaluation of existing literature on the topic correlation between reusability and design metrics have been established, shown in **Figure 2** [11] [12] [18] [19]. To establish a correlation between object oriented design properties and reusability, the influence of design metrics are being examined with respect to developed reusability quantification model. It was observed that every Object Oriented Design metrics affect quality factor. Identified independent variables, namely Data Access Metrics (DAM), Measure of functional Abstraction (MFA), Direct Class Coupling (DCC), the values of dependent variable “Y” can be found out by using the “Reusability Quantification Model of Object Oriented Design”.

6. Reusability Quantification Model Development

It is clear from in-depth literature survey that reusability is not a new word; rather it has been in conversation among the engineering professionals at various forum, but there is no commonly accepted complete and com-

Table 3. Object oriented design properties and advantages.

Property	Definition	Advantages
Encapsulation	Defined as a kind of data abstraction and information hiding.	Reduces complexity Reusability
Inheritance	A measure of the “is-a” relationship among classes.	Eliminates redundant code Reusability
Coupling	Show the interdependency of an object to other objects in a system design.	Good understandability Reusability
Cohesion	Refers to the internal consistency within the components of design.	Reusability

Table 4. Metrics description.

Metrics	Name	Description
ENM	Encapsulation Metric	This metric count of all the methods defined in a class and hold a single construct behavior so that OOD property prevent access to attribute declaration by protecting the internal structure of the objects.
DCC	Coupling Metric	This metric shows the property of interdependency among objects in a design.
COM	Cohesion Metric	This metric computes the relatedness among methods and attribute in a class.
MFA	Inheritance Metric	This metric count “is-a” relationship between classes. This association is correlated to the level of class in an inheritance structure.

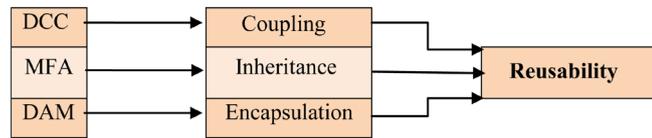


Figure 2. Correlation among reusability, OOD properties and metrics.

prehensive model or framework available to estimating the reusability index of the software product at design phase in development life cycle, that motivate to develop “Reusability Quantification Model of Object Oriented Design”, using object oriented principle based on its inner design property in development life cycle. This model used above mentioned design metrics to illustrate a range of measurement for software project. In order to create a model for reusability factor, following multiple regression technique has been selected.

$$\text{Reusability} = \alpha_0 \pm \alpha_1 * \text{Coupling} \pm \alpha_2 * \text{Inheritance} \pm \alpha_3 * \text{Encapsulation} \quad (1)$$

Using “SPSS” math work software values of all design metrics, intercept, and coefficient of the respective metrics are calculated. On the basis of this technique, the multiple regression reusability model has been developed in Equation (2). The developed multiple regression model takes the following form

$$\text{Reusability} = -37.111 + 3.973 * \text{Coupling} + 32.500 * \text{Inheritance} + 20.709 * \text{Encapsulation} \quad (2)$$

Statistical Significance of Reusability Quantification Model

The descriptive statistics of the output table gives the valuable record of statistics that are mean, standard deviation and number of software projects selected for each of the dependent variable and independent variable.

ANOVA examination for dependent variable (reusability) gives the result of F (frequency ratio) with df (degree of freedom). ANOVA output **Table 5** shows frequency ratio of 19.077 with (3, 4) degree of freedom by experimental tryout at the confidence interval level of 95%. The critical values of F is 9.12 [20]. Study obtained F ratio at (3, 4) degree of freedom is 19.077 that is larger than this, so result conclude that F-ratio is likely to occur by chance with a $P < 0.05$ level of significance.

The coefficients part and statistical significance of design metrics and constant for reusability model gives the values, that need in order to write the reusability model and prove that all the three identified design metrics do the statistically significance role at the 95% confidence level. The standardized beta coefficient give a measure of the contribution of each independent variable to the reusability model and significance values give a rough suggestion of the impact of each predictor variable. The data used for developing Reusability model has been collected through large commercial object oriented system (**Table 6**).

The Reusability Quantification Model summary (**Table 7**) of the output is most useful when performing multiple regressions. In this table “R” is the multiple correlation coefficient that used to know how strongly multiple independent variable are related to dependent variable. “R square” gives supportive coefficient of determination.

7. Empirical Validation of Developed Model

Empirical validation of work proves that how significant developed model, where metrics and model are able to quantify the reusability index of object oriented design in design stage. This validation is an essential phase of research to evaluate the developed model for appropriate execution and high level acceptability. It is also the fine approach and practice for claiming the model acceptance. To justify claiming for acceptance of developed model, an experimental validation of the proposed reusability quantification model at design phase has been carried out using tryout data from different versions of two famous windows application frameworks, object windows library (OWL) of four publicly released versions OWL 4.0, OWL 4.5, OWL 5.0, OWL 5.2 and Microsoft foundation class (MFC) of five publicly released versions MFC 1.0, MFC 2.0, MFC 3.0, MFC 4.0, MFC 5.0 [21]. In order to validate developed model, the value of metrics are available by using above data set for following projects in **Table 8**.

8. Statistical Analysis of Reusability Quantification

Charles Spearman’s Rank Correlation Coefficient (r_s) was used to test the significance of correlation between

Table 5. Examination of ANOVA for reusability model.

Model (1)	Sum of Squares	Df	Mean Square	F	Sig.
Regression	51197.757	3	17065.919	19.077	0.008 ^a
Residual	3578.411	4	894.603		
Total	54776.169	7			

a. Predictors: (Constant), Inheritance, encapsulation, Coupling; b. Dependent Variable: Reusability.

Table 6. Coefficients of independent variables and constant for reusability model.

Model Equation No. 2	Unstandardized Coefficients		Standardized Coefficients	t	Sig.
	B	Std. Error	Beta		
(Constant)	-37.111	77.201		-0.481	0.656
Coupling	3.973	72.513	0.009	0.055	0.959
Inheritance	32.500	7.395	1.017	4.395	0.012
Encapsulation	20.709	84.225	0.056	0.246	0.818

Dependent Variable: Reusability.

Table 7. Model summary for reusability model.

Model	R	R Square	Adjusted R Square	Std. Error of the estimation	Change Statistics				
					R Square Change	F Change	df1	df2	Sig. F Change
1	0.967 ^a	0.935	0.886	29.90991	0.935	19.077	3	4	0.008

a. Predictors: (Constant), Encapsulation, Coupling, Inheritance.

Table 8. Known and calculated reusability ranking and project acceptance under charles spearman's rank correlation coefficient.

Projects	Calculated Reusability Index	Calculated Reusability Rating	Known Reusability Index	Known Reusability Rating	d ²	r _s	r _s > 0.6563
P1	162.73716	11	195.92	14	9	0.983929	√
P2	218.71919	12	204.37	15	9	0.983929	√
P3	267.08929	14	56.84	9	25	0.955357	√
P4	278.69338	15	81.94	10	25	0.955357	√
P5	265.89075	13	11.29	7	36	0.935714	√
P6	36.3647	7	157.71	12	25	0.955357	√
P7	48.33769	8	168.81	13	25	0.955357	√
P8	127.42131	9	54.62	8	1	0.998214	√
P9	127.78604	10	82.53	11	1	0.998214	√
P10	0.461	1	10.50	5	16	0.971429	√
P11	17.4928	4	10.50	6	4	0.992857	√
P12	14.1456	3	8.00	2	1	0.998214	√
P13	34.9219	5	8.70	3	4	0.992857	√
P14	34.9219	6	7.20	1	25	0.955357	√
P15	12.9665	2	9.00	4	4	0.992857	√

calculated index values of reusability by *Reusability Quantification Model of Object Oriented Design* and its known index values given by expert shown in **Figure 3**. The “ r_s ” was calculated using the following formula

$$r_s = 1 - \frac{6\sum d^2}{n(n^2 - 1)} \quad -1.0 \leq r_s \leq +1.0$$

where

r_s is coefficient of Rank Correlation,

d is the difference between calculated index values and known values of reusability,

n is the number of software projects for experiment, (In this research $n = 15$ software projects)

\sum is notification symbol, significance “The Sum”.

Comparison between calculated reusability rank correlation coefficient and threshold value (0.6563) at 95% confidence level is shown in following line graph **Figure 4**.

Study results have proved that the correlation is acceptable with high degree of confidence that is 95%. Therefore research is concluded without any loss of generality the “*Reusability Quantification Model of Object Oriented Design*” highly significant, because estimation values are more reliable and valid in the context.

9. Key Contributions and Findings

Developed “Reusability Quantification of Object Oriented Design” for object oriented design. The Model has been validated using the same set of try-out data. An empirical validation of the developed model is also performed using try-out data.

Some of the major findings are as given below:

1) Reusability has been identified as a key factor to software testability, addressed in design phase of object oriented software development to produce quality software.

2) The three metrics DAM (Data Access Metrics), MFA (Measure of functional Abstraction), DCC (Direct class Coupling) are identified for each of object oriented design constructs such as encapsulation, inheritance and coupling respectively.

3) Software design constructs are most appropriate and power full for controlling software quality factors in design phase.

4) A guideline produced by developed framework to be followed right from beginning of development so as to design class hierarchy as per the prescriptive reusability index for the project under development.

5) Developed Framework may be used to get reusability metric in design phase for reusability measuring.

6) Reusability indexing (RI) can be done using the proposed model “Reusability Quantification of Object Oriented Design”.

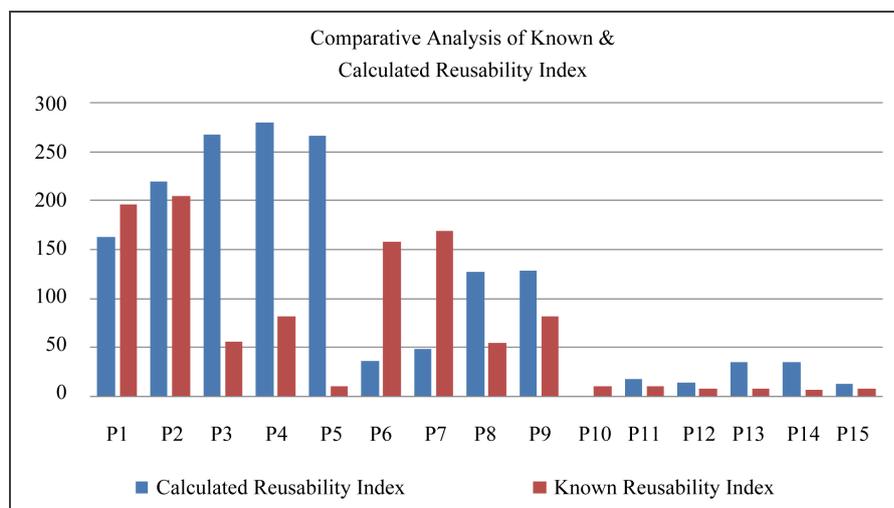


Figure 3. Line graph comparing between known and calculated reusability index for software projects.

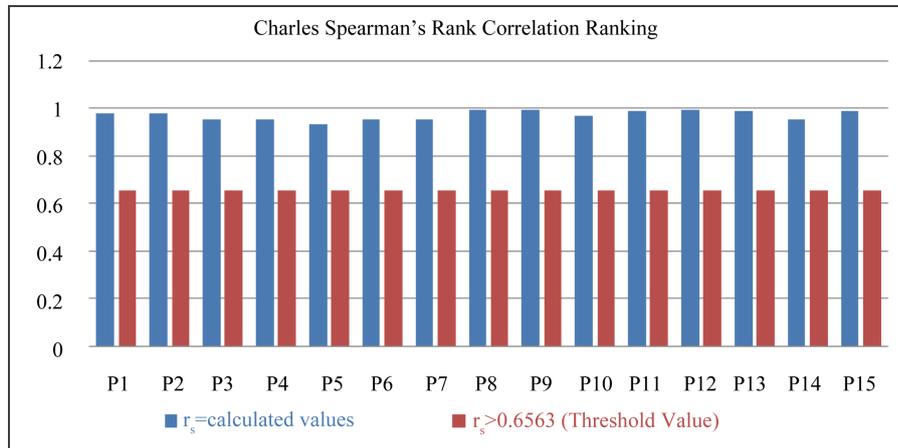


Figure 4. Line graph comparison between calculated correlation values and threshold value.

7) “Reusability Quantification of Object Oriented Design” provides a Reusability indexing (RI) benchmark for other researchers and designers.

8) For Industry project ranking, Reusability indexing (RI) is possible using the “Reusability Quantification of Object Oriented Design”. The developed model may be generalized and used by others researchers.

10. Conclusion

This paper has developed an efficient and accurate model for reusability quantification through object oriented design metrics using the technique of multiple regressions. The developed model proposed the possibilities to estimate overall reusability from designed information. Developed model has been demonstrated using functionally equivalent projects and the assessment of reusability index in OOD has been validated using structural and functional information from object oriented software. This paper also validates the quantifying ability of developed model. That validation study on this research work proves that proposed reusability quantification model is highly acceptable, more practical in nature and helps the software industry in project ranking.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor Prof. Dr. YDS Arya & Co-supervisor Associate Prof. Dr. M H Khan for the continuous support of my PhD study and research, motivation, enthusiasm. Their guidance helped me in all the time of research. Last but not the least; I would like to thank my parent for their patience, understanding and support that drive me to complete my study.

References

- [1] Binder, R.V. (1994) Design for Testability in Object-Oriented Systems. *Communications of the ACM*, **37**, 87-101. <http://dx.doi.org/10.1145/182987.184077>
- [2] 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.
- [3] 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 (IJSEA)*, **6**, 41-49. <http://dx.doi.org/10.5121/ijsea.2015.6104>
- [4] Freeman, P. (1983) Reusable Software Engineering Concepts and Research Directions. In Tutorial: Software Reusability, 10-23.
- [5] Basili, V.R. and Rombach, H.D. (1988) Towards a Comprehensive Framework for Reuse: A Reuse-Enabling Software Evolution Environment. Technical Report CS-TR-2158, University of Maryland, Maryland.
- [6] Tracz, W. (1995) Confessions of a Used Program Salesman: Institutionalizing Software Reuse. Addison-Wesley. <http://dx.doi.org/10.1145/211782.211785>
- [7] Braun, C.L. (1994) Reuse. In Marciniak, 1055-1069.

-
- [8] Cooper, J. (1994) Reuse-the Business Implications. In Marciniak, 1071-1077.
- [9] Krueger, C.W. (1992) Software Reuse. *ACM Computing Surveys*, **24**, 131-183.
- [10] 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. <http://dx.doi.org/10.17148/IJARCCCE.2015.4168>
- [11] ISO (2001) ISO/IEC 9126-1: Software Engineering—Product Quality—Part-1: Quality Model. Geneva.
- [12] IEEE Press (1990) IEEE Standard Glossary of Software Engineering Technology. ANSI/IEEE Standard 610.12-1990.
- [13] McCall, J.A., Richards, P.K. and Walters, G.F. (1977) Factors in Software Quality. RADC TR-77-369, Rome Air Development Center, Rome.
- [14] Dromey, R.G. (1996) Concerning the Chimera (Software Quality). *IEEE Software*, **13**, 33-43. <http://dx.doi.org/10.1109/52.476284>
- [15] Boehm, B.W., Brow, J.R., Lipow, M., McLeod, G. and Merritt, M. (1978) Characteristics of Software Quality. North Holland Publishing, Amsterdam.
- [16] Grady, R.B. (1992) Practical Software Metrics for Project Management and Process Improvement. Prentice Hall, Englewood Cliff.
- [17] Khan, R.A. and Mustafa, K. (2004) A Model for Object Oriented Design Quality Assessment. *Proceedings of the Integrated Design and Process Technology Symposium*, Izmir, 28 June-2 July 2004.
- [18] Fu, J.P., Liu, B. and Lu, M.Y. (2010) Present and Future of Software Testability Analysis. *Proceedings of the International Conference on Computer Application and System Modeling (ICCASM)*, Taiyuan, 22-24 October 2010, V15-279-V15-284.
- [19] Gao, J. and Ming-Chih, S. (2005) A Component Testability Model for Verification and Measurement. *Proceedings of the 29th Annual International Computer Software and Applications Conference*, Edinburgh, 26-28 July 2005, 211-218. <http://dx.doi.org/10.1109/COMPSAC.2005.17>
- [20] <http://homepages.wmich.edu/~hillenbr/619/AnovaTable.pdf>
- [21] Bansiya, J. (2002) A Hierarchical Model for Object Oriented Design Quality Assessment. *IEEE Transaction of Software Engineering*, **28**, 4-17.