International Journal of Mathematical Archive-2(5), May - 2011, Page: 635-641
Available online through www.ijma.info ISSN 2229 – 5046

AN EMPIRICAL VALIDATION OF QUALITY TESTING MODEL FOR OBJECT ORIENTED DESIGN

***Mazhar Khaliq[1], Riyaz A. Khan[2] and M. H. Khan[3]**

[1]Faculty of Applied Sciences, Integral University, Lucknow, India

[2]Faculty of Applied Sciences, Integral University, Lucknow, India

[3]Department of Computer Science and Engineering, I.E.T., GBTU, Lucknow, India

[1]khaliqmazhar@rediffmail.com, [2]riyazakhan68@yahoo.co.in, [3]mhkhan.ietfaculty@yahoo.com

(Received on: 14-04-11; Accepted on: 26-04-11)

ABSTRACT

This paper describes an improved Quality Testing Model for Object Oriented Design (QTMOOD) for the assessment of high-level design quality attributes in object-oriented design. The proposed model relates design properties such as encapsulation, inheritance, polymorphism, coupling and cohesion to a reduced set of attributes such as efficiency, complexity, understandability and reusability from SATC's identified set of quality attributes. The relationship between quality attributes and design properties are weighted in accordance with their respective influence and importance of design properties on quality attributes. The model is validated empirically. Empirical data is collected from various application domains, and then analyzed using QTMOOD to support the validation. The result indicates that the proposed model may be used to provide an overall quality assessment of object-oriented software in early phase of development life cycle. Early use of the model may help to ensure that the favorable internal design properties will lead to the development of quality end product. However, further the experiment is required on a large scale of data in order to generalize and standardize the model.

Keywords: *Object oriented quality, Quality models, Design properties, Quality attributes.*

---

1. INTRODUCTION:

An effort has been made to put forth the demand and dependency of quality software in our modern society. Dependency and requirements on computer software increases the difficulties and failures of software and the often-devastating effect that a software error can have in terms of life, financial loss, or time delays. So the demand of quality software continues to increase by the software industry. Spending resources to improve the quality of software product does this. Software quality is not an advantage but a necessary factor for software industry [19][20]. The demand has increased the size and complexity of computer software system during the past decades. The object-oriented technology is more popular in software development environment in recent years. More and more organizations are introducing object oriented methods and languages into their software development practices. Object oriented software methodology is a way of structuring software as a collection of discrete objects reflecting real-world entities and mapping them into design constructs to represent relationships and functionality efficiently. It reflects a natural view of the domain and handles inherent complexity better. The object-oriented technology is more powerful to design the software in order to provide the product of superior quality [15]. Object oriented software development is iterative with overlapping phases in the development process. While the basic set of objects, operations, attributes, and relationships are identified in analysis phases, the details of a class's methods, parameters, data declaration, relationships, and algorithm are resolved during design. A fundamental constraint of object oriented modeling and the design is the Object, which combines both data structure and behavior in a single entity [22].

The indication and anticipation of quality as early as possible in the software development life cycle is required because with each phase of SDLC, cost impact of modification and Improvements get significantly increased [9,19]. Tradition software measurement used to evaluate the product characteristics such as size, complexity and performance. But the quality is switched to rely on some fundamentally different properties like Encapsulation, Inheritance & Polymorphism, which differentiate the approach of object-oriented software development from conventional software development. This switching led to develop many measuring tools by various researchers and practitioners to measure the object-oriented properties. Most of the tools available for object-oriented software analysis, may normally be used in the later phase of system development life cycle and rely upon the information extracted on the operations performed in the software. This provides information too

---

Corresponding author: *Mazhar Khaliq[1]
E-mail: [1]khaliqmazhar@rediffmail.com

late to help in improving the software product prior to the completion of the product. It is also fine that a couple of object-oriented measuring tools altogether may be used to measure all aspects of object-oriented design [1, 10, 11].

Literature reveals that there is no known comprehensive and complete model or framework exists for evaluating the overall quality of designs based on object oriented approach using its internal design properties. So it is required to find a quantifiable way to relate measurable object oriented characteristics to high level desirable software quality attributes in order to provide a significant and improved measurement of object oriented software [1]. However, the better framework is provided by the earlier models, from which the development of model can be proceeded. A slightly improved Quality Testing Model for Object Oriented Design (QTMOOD) model has been worked out to cater to the persisting relevant demands for the assessment of high-level design quality attributes in object-oriented design. In this model, structural and behavioral design properties of classes; objects and their relationships are evaluated using a suit of object oriented design metrics. The proposed model relates design properties such as encapsulation, inheritance, polymorphism, coupling and cohesion to a set of identified high-level quality attributes such as efficiency, complexity, understandability and reusability with the help of five object oriented metrics. This model has the low-level design metrics well defined in terms of design characteristics, and quality is evaluated as an aggregation of the model's individual to high-level quality attribute [15].

The rest of the paper is arranged as follows: Section 2 describes a brief review regarding the existing software quality models. Section 3 presents the discussion about the proposed model. The description has been made for the validation of the model in Section 4. Finally, Section 5 gives some important conclusion about the proposed model.

2. SOFTWARE QUALITY MODELS:

The indirect models have been developed by various researchers and practitioners in order to quantify software quality. These models attempt to measure software product quality by using a set of quality attributes, characteristics, and a selected set of metrics [3]. Many of the proposed models for software quality assessment relies on the information extracted from implementation of software [1]. Hence, these models may be used in later phase of software development life cycle (SDLC). Software quality is needed and to be indicated as early as possible in the SDLC since with each iteration of cycle [19]. Thus, there is a need for a metrics based quality model for testing the quality at design phase of object oriented software development. It may ensure quality compliances at this stage to increase the reliability of the system as a whole as reliability itself is a byproduct of quality.

There are some quality testing models, which are currently popular and in use in software industry [14]: As a case in point, Factor-Criteria-Metrics (FCM) Model is generally accepted as a basis for software evaluation. This model is based on decomposing the attributes into a set of quality attributes which themselves can be decomposed into a set of criteria. These criterion values can be assessed from a set of software-related measurements, software metrics. It was found that the early models McCall model [16] and Boehm model [2], which were developed based on FCM model in the late of 1970s. These models are often criticized for lack of rationale in determining which factors should be included in the quality definition and which criteria should relate to some specific factors [12]. In recent years, ISO 9126 model [7] was proposed, which is a derivation of McCall's Model. The REBOOT (Reuse Based on Object Oriented Technology) model is also based on the FCM model, it may consider as a general quality model and reusability model [6]. SATC (Software Assurance Technology Center) has developed a software quality model, which can have a quantitative measure of software quality. Following the structure of ISO 9126 model, the SATC model defines a set of goals, which are related to software product and process attributes that may indicate the probability of success [21]. Geoff Dromey has proposed a quality model framework, in which a methodology is given for the development of quality models in a bottom-up fashion [5]. QMOOD (Quality Model for Object Oriented Design) was proposed as a hierarchical model, which can be used to assess object-oriented design quality. In this model, a hierarchy of levels is used to relate high-level, it makes difficult to assess quality attributes to the low level of details [1].

Most of the proposed models are empirically validated [3]. Some quality models are hierarchical models based on group of quality criteria, associated with group of metrics. All the models are vague in their definition of the lower level details and metrics needed for the quantitative assessment of software product quality. This shows the lack of specifics in these models, it offers little guidance to software developers, who need to produce quality product. The inability of these models to account for dependency among quality attributes was found to be another difficulty. Earlier models also failed to include ways of accounting for degree of influence of individual attributes. The attributes performance and reliability may be the most valuable attributes for a large organization with real time processing [18].

3. QTMOOD - Quality Testing Model for Object Oriented Design [15]:

An overall quality has been assessed as an aggregation of a set of quality attributes, the significance of all attributes to the overall quality may not be comparable, therefore the influence of an individual attribute needs to be changed by its weighting factor. There is no common set of quality attributes that completely represents the overall quality. Thereby the frameworks provided by earlier models might be considered and utilized to evaluate and define a set of metrics, relationships and weights in the context of the model [15].

The Dromey's generic quality model has been extended to develop the Quality Testing Model for Object Oriented Design, The Dromey's model deals with three basic principles including the product properties that influence quality, a small set of high level quality attributes, and defining the relation between them to build the model of software product quality [5]. So based on these principle elements the following phases are considered in order to develop the QTMOOD:

- Identification of a set of high-level design quality attributes.
- Selection of product design properties that influences quality.
- Identification of object oriented design metrics.
- Relationship among these elements.

As Dromey suggested that a small set of high-level quality attributes has to be identified first to formulate a model for software product quality. And second thing, it is better to adopt an approach to employ a model that places only a single level of high-level quality attributes [5]. Keeping these approaches in mind, researcher has studied the set of quality attributes of SATC [21], and identified four quality attributes from the set of SATC such as efficiency, complexity, understandability and reusability. These attributes may be used as a common and viable set of software quality attributes, this set may predict the best level of quality of entire software. The adaptation of the object-oriented paradigm is expected to help produce better and cheaper software. The main structural mechanisms of this paradigm, namely, inheritance, encapsulation, and polymorphism, are the keys to foster reuse and achieve easier maintainability. It is feasible to identify the set of quality factors and the object oriented design characteristic at the design phase. Moreover, the design properties such as coupling and cohesion may be used to play the role about good. Therefore, the researcher has identified inheritance, encapsulation, polymorphism, coupling and cohesion as quality carrying product characteristics, which may be used at the design phase in order to develop a model, which can help to estimate the quality of object oriented software [15]. The five objected oriented metrics WMC(Weighted methods per class), DIT(Depth of inheritance), RFC(Response for a class), CBO(Coupling between object classes) and LCOM(Lack of cohesion in methods), proposed by Chidamber and Kemerer [4], have been identified that may be used to estimate the overall quality of software. These proposed metrics caters all the object oriented design characteristics and covers all the quality factors. There after, design metrics may be used to relate the knowledge about good design to the characteristic structural system properties. Therefore, these metrics may be used at the design level to estimate the quality of object-oriented software [15]. Finally, In order to establish a base relationship between quality attributes and design properties, the literature reveals that each of quality attributes is being affected by certain design properties [13]. The relationship among the elements can be drawn by examining the respective influence of design constructs on quality attributes with respect to the reduced set of attributes from the SATC's identified set of attributes. By the extensive reviewed information on object-oriented development, the relationship has been made in order to relate product design properties to quality attributes [13], which is depicted in Figure 1. In order to test the software quality of object oriented design in terms of improving efficiency, decreasing complexity, increasing understandability and increasing reusability, the QTMOOD model [15] has been developed in the light of foregoing descriptions and exploration.

4. VALIDATION:

Here, it is required to assess that how well the proposed model QTMOOD is able to predict the `overall quality' of' an object oriented software design. The internal characteristic of a design varies significantly with the objective and domain of the design. These characteristics influence the quality attributes and, therefore, the overall quality. So validation of the predictability of model requires the set of object oriented designs with the same set of requirements for evaluation and validation, as a limitation of this model. The assessment of the overall quality of designs determined by given model needed to agree with the generally accepted requirements or characteristics of the overall quality designs as perceived by analysts, developers, and customers. Keeping these objectives in mind, overall software quality evaluated by QTMOOD has been tested through Empirical Validation having a Designing an Experiment, Pre-Tryout, Try-Out and Contextual Finding.

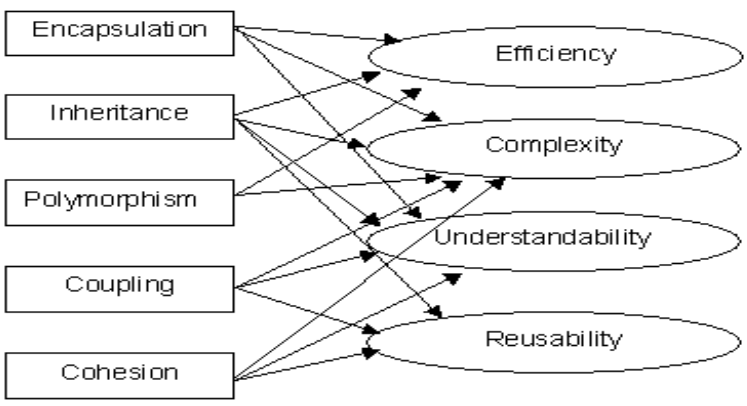

Figure 1: Design constructs and Quality Attributes Relationship

Based upon the relationship shown in Figure 1 between design constructs and quality attributes, the relative significance of individual design properties that influence a quality attribute is weighted proportionally. A multiple linear regression (MR) may be used to get the coefficients [8,17]. The multiple linear regressions establish a relation between dependent variables and multiple independent variables. Thus the MR equation may get the form as follows:

$$Y=\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots\ldots\ldots\ldots + \beta_n X_n \quad (1)$$

Where the $\mathbf{X_S}$ are the independent (predicator or regressor) variables, **Y** is the dependent (response) variable, and β is the regression coefficient, the average amount of dependent increase when the independents are held constant.

The various components of the regression model along with other statistical terminologies used are listed here.
Independent Variable (X_i's): In the regression equation (1), the X_i's denote the independent variables. The independent variable in an experiment is the variable that is systematically manipulated by the investigator. In most experiments, the investigator is interested in determining the effect that one variable; has on one or more of the other variables.

Dependent Variable (Y): In the regression equation (1) variable Y denotes the dependent variable. The dependent variable in an experiment is the variable that the investigator measures to determine the effect of the independent variable.

Coefficients (β_j's): In the regression equation (1), β_j's represent the coefficient terms. The estimated multiple linear regression coefficient measure the respective independent variable's contribution on the variable. The larger the absolute coefficient values, the larger (positive or negative according to the sign) the impact of the independent variable on the dependent variable.

As quality attributes depend upon one or more number of design attributes, as depicted in Figure 1, component wise effect may be speculated and respective component weightages (CWs) may be fixed using regression equation. Thereby, the CWs of individual design attributes have to be calculated in terms of the regression coefficient β. According to the relationship, as shown in Figure 1, and MR equation, the computation formulas may take the form as shown in Table 1.

4.1 An Empirical Validation:

This section assesses how well the model is able to predict the 'overall quality' of an object oriented software design. The internal characteristic of a design varies significantly with the objective and domain of the design. These characteristics influence the quality attributes and, therefore, the overall quality. So validation of the predictability of the QTMOOD requires the set of object oriented designs with same set of requirements for the evaluation and validation, as a limitation of this model. The assessment of the overall quality of designs determined by the QTMOOD needed to agree with the generally accepted requirement or characteristics of overall quality designs as perceived by analysts, developers and customers. Keeping these objectives in mind, overall software quality produced by the QTMMOD may be validated in three phases. Phase 1, Design of the Experiment, describes the section of model validation suit. Having Pre-Tryout as Phase 2. Evaluator's assessments of projects overall quality, and evaluation of the project design using QTMOOD has been discussed in Phase 3 under the heading Tryout. The Statistical Analysis in Phase 4 includes the rank correlation coefficient test for result comparison; and results comparison for significance of the rank correlation coefficient.

4.1.1 Designing the Experiment:
A medium size C++ project is used in the validation of QTMOOD, using overall quality or QL measure. This project was selected because several designs of the project that addresses the same set of requirements that is easily available. The size of these projects, reflected by the number of classes in different ranges, made them an ideal candidate for both QTMOOD and the assessment of company quality executives. A suite of 7 projects was undertaken for the study and different development teams of the same company developed these.

4.1.2 Pre – tryout:
The medium size C++ projects were used to fit the regression line by acquiring real data from commercial projects of a software company based in India. Name of the projects and actual source data is being concealed, as per the wishes of company's management. These projects include the number of classes in the range 10-20. Using these data, the component weightage (CW) calculated for encapsulation, inheritance and polymorphism β_1, β_2 and β_3 respectively to define the quality attributes (efficiency) relationship with design properties (encapsulation, inheritance and polymorphism) shown in Table 1. Table 1 summarizes the remaining so calculated computation formulae for quality attributes with component weightage. Thereby, equations relate the quality attributes with the design properties, as given in Table 1.

Table 1: Computational Formulas for Quality Attributes

Quality Attributes	Index Computation Equations
Efficiency	0.24* Encapsulation + 0.26* Inheritance + 0.25* Polymorphism
Complexity	0.2* Encapsulation + 2.5* Inheritance + 0.25* Polymorphism+ 1.5* Coupling + 0.5* Cohesion
Understandability	0.30* Encapsulation + 1.2* Inheritance + 0.35* Coupling + 0.28* Cohesion
Reusability	0.5* Inheritance + 0.25* Coupling + 0.55* Cohesion

Table 2: Design Property Metric Values

Metrics	1	2	3	4	5	6	7
Encapsulation	2.85	2.87	3.92	3.35	4.5	4.25	3.95
Inheritance	.35	.55	.25	.32	.42	.45	.28
Polymorphism	1.90	1.92	2.0	1.85	1.95	1.87	2.5
Coupling	1.85	1.87	2.05	1.8	2.5	2.65	2.75
Cohesion	1.25	1.32	1.57	1.26	1.35	1.27	1.42

Table 3: Design Quality Attribute Indices

Quality	1	2	3	4	5	6	7
Efficiency	1.25	1.31	1.51	1.35	1.68	1.60	1.64
Complexity	5.32	5.89	5.76	5.26	6.86	7.05	6.95
Understandability	2.27	2.54	2.63	2.37	3.10	3.10	2.89
Reusability	1.32	1.47	1.50	1.30	1.58	1.59	1.61
QL	**10.16 (1)**	**11.21 (3)**	**11.40 (4)**	**10.28 (2)**	**13.22 (6)**	**13.34 (7)**	**13.09 (5)**

Table 4: Comparative Quality Rankings

Project No.	QTMOOD Ranking	Evaluators Ranking									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	1	1	2	1	3	2	3	1	1	2	2
2	3	4	3	3	4	4	2	3	5	4	3
3	4	3	5	5	2	3	4	3	4	3	5
4	2	2	1	2	1	1	1	2	2	1	1
5	6	5	5	6	5	5	6	5	6	5	6
6	7	6	6	7	7	6	7	7	7	6	6
7	5	5	4	4	6	7	5	4	3	6	4

4.1.3 Tryout:

A group of some independent evaluators have been studied the quality of the projects in the validation suite. All the evaluators analyzed each project's design and assigned the score on a ordinal scale to one or more of the four quality attributes i.e. Efficiency, Complexity, Understandability and Reusability by using their own traditional tool. These scores for all four high-level quality attributes were summed separately to give the evaluator's QL for project. The projects were ranked from 1 to 7 based on decreasing QL scores by each evaluator as shown in Table 4. The values of Table 3 have been used to put QTMOOD Ranking in Table 4. The 7 projects of the validation suite are evaluated using QTMOOD. The metrics values calculated for the five metrics used to assess the design properties are shown in Table 2. The values for four high-level quality attributes are computed using the equations in Table 1 and shown in Table 3. The values of the 4 quality attribute measurements for each project in Table 3 are summed to provide the QL value for the project, which was used as a basis for ranking the project designs (1 through 7, 1 for highest value of QL and 7 for lowest value of QL) as shown in the column labeled QTMOOD Ranking of Table 4.

4.1.4 Statistical Analysis:

The Spearman's rank correlation, $\mathbf{r_s}$ may be used to test the significance of correlation between QTMOOD's design-based quality assessment and the evaluator's implementation based assessment of the project. It provides a nonparametric significance test that works well with ranked data without precise proportional scaling and can be used to detect relationships other than the linear one. For the level of significance of α=95%, the threshold value calculated for n=7 projects was ±0.745.

For two independent evaluations X_1 and X_2 of n items that are to be correlated, the values of X_1 and X_2 are ranked from '1 to n' according to their relative size within the evaluations. For each X_1 and X_2 pair in relative rank, differences in ranks 'd' are computed. The sum of all squired ds denoted by $\mathbf{\Sigma d^2}$ is used to compute $\mathbf{r_s}$ as follows:

$$\mathbf{r_s} = 1 - (6\Sigma d^2/n(n^2-1)) \quad ; \quad -1.0 \leq r_s \leq +1.0 \qquad (2)$$

where, r_s is the coefficient of Rank Correlation, n is the number of paired observation, and d is the difference between the ranks for each pair of observations.

4.2 Contextual Finding:

In order to comparing the results for the significance of $\mathbf{r_s}$ the correlation values between QTMOOD determined

design quality ranking of 7 projects and the 10 evaluators assessments of overall quality of these projects based on design and implementation shown in Table 4. Pairs of QTMOOD – Evaluator with correlation values $\mathbf{r_s}$ above (±0.745) are checked in Table 5. It is evident that the quality indicators evaluated using QTMOOD are highly correlated with the measurements done by human evaluators, the correlation values shown in Table 5. Therefore, inference can be given about the reliable testing of quality of object-oriented software. And that QTMOOD has been able to appropriately evaluate the quality of the projects under study.

Table 5: QTMOOD and Evaluators Ranking Correlation Summary

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Σd^2	4	6	2	12	10	6	3	8	7	5
r_s	0.92	0.89	0.96	0.78	0.82	0.89	0.94	0.85	0.87	0.91
r_s**>0.745**	√	√	√	√	√	√	√	√	√	√

5. CONCLUSION:

The proposed quality-testing model, QTMOOD, for the assessment of high-level design quality attributes in object oriented design has been developed and validation is required by using structural and functional information from object oriented software. The model's ability to estimate overall design quality from design information in order to test the quality, at least for the projects under consideration, has been demonstrated. Several functionally equivalent project's quality estimate computed by the model has been found to be significantly correlated, through the reported statistical analysis, with the assessment of overall project quality characteristics determined by independent evaluators. Quality indicators evaluated using QTMOOD are highly correlated with the measurements done by human evaluators. It is apparent that the model can be used effectively in monitoring and hence controlling the quality of software in design phase in terms of improving efficiency, decreasing complexity, increasing understandability and increasing reusability. QTMOOD may be able to discover the flaws in the software design at early stage of SDLC. This model is able to appropriately test the quality of software projects under study and development at design phase. However, experiments may be done on a large scale in order to generalize and standardize the model.

REFERENCES:

[1] Bansiya Jagdish, "A Hierarchical Model for object-oriented Design Quality Assessment", IEEE Transaction on software engineering, Vol.28, No.1, January 2002.

[2] Boehm, B. W., Brown, J. R., Kaspar, H., Lipow, M., Macleod, G. J., and Merritt, M. J., "Characteristics of Software Quality", Amsterdam: North-Holland, 1978.

[3] Briand L. and Wi st J., "Empirical Studies of Quality Models in Object- Oriented Systems. Advances in Computers", Academic Press, Zelkowitz (ed.), 59, 97-166, 2002.

[4] Chidamber S. R. and Kemerer C. F., "A Metrics Suite for Object Oriented Design", IEEE Transaction on software engineering, 20 (6) June 1994, pp. 476-493.

[5] Dromey, R. G. "A Model for Software Product Qulity", IEEE Transaction on Software Engineering 21(2), Feb. 1995, p. 146-162.

[6] Even-Andre Karlsson. Chichester, "Software Reuse: A Holistic Approach -Measuring the Effect of Reuse Chapter", New York: Wiley, 1995: 113-180.

[7] Fenton, N. E., lizuka, Y., & Whitty, R. W. (eds). "Software Quality Assurance and Measurement: A Worldwide Perspective", London: International Thomson Computer Press, 1995.

[8] Fox J., "Applied Regression Analysis, Line Models, and Related Methods", AGE Publication, 1997.

[9] Khan R. A. & Mustafa K., "High Level Design Quality Assessment of Object Oriented Codes", proceedings of 2"[d] International Workshop on Verification and Validation of Enterprise Information System VVEIS Porto, Portugal, April 13, 2004.

[10] Khan R. A. & Mustafa K., "Quality Estimation of Object Oriented Code in Design Phase", Developer IQ, Software Technology Magazine, Techmedia, India, Vol. 4, No. 2, Feb. 2004.

[11] Khan R. A., Mustafa K., & Yadava S., "Quality Assessment of Object Oriented Code in Design Phase", Proceedings, QAI 4th Annual International Software Testing Conference, Pune, India Feb. 20-21, 2004.

[12] Kitchenham, B., Pfleeger S., "Software Quality-The Elusive Target", IEEE Software January 1996 12-21.

[13] Mazhar Khaliq, Riyaz A. Khan, M. H. Khan, "Significance of Design Properties in Object Oriented Software Product Quality Assessment",International Journal of Computing Science and Communication Technologies, Vol. 3, Issue 2, January 2011.

[14] Mazhar Khaliq, Riyaz A. Khan, M. H. Khan, "Software quality Measurement: A Revisit", Oriental Journal of Computer Science & Technology, vol.3 (1), 05-11, June 2010.

[15] Mazhar Khaliq, Riyaz A. Khan, M.H. Khan, "Quality Testing Model for Object Oriented Design: QTMOOD", Journal of Computer and Information Technology, vol. 2 (1), April 2011.

[16] McCall, J. A., Richards, P. G., and Walters, G. F. "Factors in Software Quality", Vols. I, II, and III (NTIS AD/A-049 014/015/055), Springfield: NTIS, 1977.

[17] Mohammad Alshayeb, "An Empirical Validatio of Object Oriented Metrics in Two Different Iterative Software Process", IEEE Transaction on Software Engineering, Vol. 29, No. 11, Nov. 2003.

[18] R A Khan, K Mustafa, "MQMOOD: Metric Based Model for object oriented Design Quality Assessment", Information Technology Journal 4, 2005.

[19] R A Khan, K Mustafa, S I Ahson, "An Empirical Validation of Object Oriented Design Quality Metrics", J. King Saud Univ. Vol. 19, Comp & Info. Sci., pp. 1-16, Riyadh.

[20] R A Khan, K Mustafa, S I Ahson, "Software Quality Concepts and Practices", Narosa Publishing House Pvt. Ltd., 2006.

[21] Rosenberg Linda, "Software Quality Metrics for Object Oriented System Environments", A report of SATC's research on 00 metrics" http://ourworld.compuserve.com/homepages/qualazur/$swmesu2.htm.

[22] Rumbaugh, J., Blaha, M., et. Al., "Object Oriented Modeling and Design", Prentice Hall, 1991.

****

