J. King Saud Univ., Vol. 19, *Comp. & Info. Sci*., pp. 1-16, Riyadh (1427H./2007)

An Empirical Validation of Object Oriented Design Quality Metrics

R. A. Khan[1], K. Mustafa[2] and S. I. Ahson[3]

[1]Department of IT, BBA University (A Central University), Lucknow, UP, India
[2]CCSIT, King Faisal University, KSA
[3]Department of Computer Science, JMI New Delhi, India
[1]khanraees@yahoo.com, [2]kmustafa@kfu.edu.sa, [3]drsiahson@yahoo.com

(Received 11 March 2006; accepted for publication 13 December 2006)

Abstract. This paper describes an integrated single class based metric called Weighted Class Complexity (WCC) for object oriented design. The metric is discussed from measurement theory viewpoint, taking into account the recognized object oriented features which the metrics was intended to measure encapsulation, inheritance, coupling and polymorphism, and the quality factors efficiency, complexity, understandability, reusability and maintainability/testability. Empirical data, collected from eight different application domains, is then analyzed using WCC metric to support this theoretical validation. The result shows that the proposed metric could be used to provide an overall quality assessment of object oriented software system in early stage of development life cycle, which may be helpful to the developer to fix problems, remove irregularities and non-conformance to standards and eliminate unwanted complexities in the early development cycle. Early use of this metric may help to ensure that the analysis and design have favorable internal properties that will lead to the development of quality end product. WCC may significantly help in reducing rework during and after implementation to design effective test plans. However, further studies are needed before these results can be generalized.

Keywords: Object oriented quality metrics, Quality models, Design characteristics, Quality attributes.

1. Introduction

The importance of quality software is no longer an advantage but a necessary factor as software error can have in terms of life, financial loss, or time delays. No doubt that the software quality can make or break a company. Unfortunately, most companies not only fail to deliver a quality product to their customers, but also fail to understand the attributes of a quality product. Traditional software metrics used to evaluate the product characteristics such as size, complexity, performance and quality is switched to rely on some fundamentally different attributes like encapsulation, inheritance and polymorphism, which are inherent in object-orientation. This switching led to the definition of many metrics proposed by various researchers and practitioners to measure the object oriented attributes. Most of the metrics, available for object oriented software analysis, may normally be used in later phase of system development life cycle and rely upon information extracted on the operationalization of software [9]. Such metrics provide the indication of quality too late to improve the product, prior to the completion of product. It is also true that a *couple* of object oriented metrics altogether may be used to measure all the aspects of object oriented design [7]. Thus, there appeared to be a need for developing a *single integrated object oriented metric, encompassing all the object oriented design constructs*, which may be used in an early stage of development to give good indication of software quality. It is strongly felt to be more productive and constructive.

The indication and anticipation of quality as early as possible in System Development Life Cycle (SDLC) is necessary because with each iteration of SDLC, cost impact of modification and improvement will significantly increase [8]. The ability to cover all aspects of quality factors and the design characteristics, to represent different aspects of the system under measurement, to get the same value for the same systems for the different people at different time, to be used with the minimum number of metrics, to have an empirical validation and ability of failure free operation are identified as the essential features of the desired object oriented metrics. In addition to these essential features, it is also felt that the desired metrics should possess some desired features that are identified as its ability to quantify various attributers of reuse, to reduce the rework after the implementation, to reduce the testing and maintenance cost, to ensure that analysis and design have favorable internal properties that will lead to the development of quality product, to reflect the level of maintainability, to improve the estimation process to achieve better resource allocation, to measure the psychological complexity factors that affect the ability of a programmer to create, modify and comprehend software and the end user to effectively use the software, to be implemented easily, to be interpreted easily and its ability to enhance predictability.

The rest of the paper is organized as follows: Section 2 describes the proposed metrics that has been theoretically validated in Section 3. Experimental validation for the metrics has been discussed in Section 4. Finally, Section 5 presents some important conclusions about the proposed metrics.

2. WCC: A Class Based Metrics

Many of the metrics proposed by different researchers and practitioners for object oriented software analysis relies on the information extracted from the implementation of software [9]. Hence, these metrics may be used in later phase of software development life cycle (SDLC). Software quality is required to be indicated as early as possible in the SDLC since with each iteration of cycle, cost impact of modification and improvement significantly increases. Thus, there is a need for object oriented metrics that may be used in code and design phase and 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.

In order to establish a relationship between design constructs and attributes of quality, the influence of design constructs and quality attributes are being examined with respect to SATC's attributes [5, 6]. It was observed that each design constructs affects certain quality attributes. This is being depicted in Fig. 1 [5, 8].

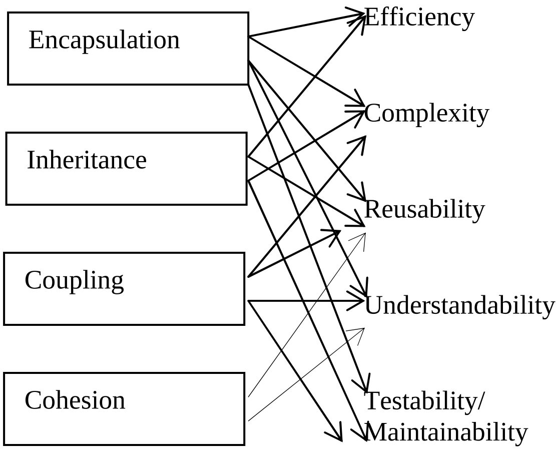

Fig. 1. Design constructs affecting quality attributes.

The survey result on the development of object oriented metrics depicts that all metrics have relevance with respects to a class [1, 2, 4, 7, 8]. This motivated effort towards developing a single class based metrics, Weighted Class Complexity (WCC), which would give a cumulative measure of encapsulation, coupling, cohesion, and inheritance aspects of object oriented design and would thereby give an indication of 'quality' of a class in terms of complexity. This single metric when averaged would enable computing the average complexity of software and finally the quality.

The complexity in this context has more of physiological meaning rather than complexity as a quality attribute. This WCC should take into account most of the design constructs, i.e. WCC should be coupled of an encapsulation factor, an inheritance factor, a coupling factor and a cohesion factor.

2.1. Metrics formulation

It is evident from Fig. 1 that the encapsulation affects all five-quality factors, and may be considered as a key construct to improve the quality of software [5, 11, 12, 13]. This means better encapsulation should cause a decrease in WCC. Again, the more number of external links (coupling), the lower is the flexibility of software and greater the complexity. So, the increase in coupling factor should cause an increase in WCC. For cohesion, we know that the higher the cohesion, the better the design and therefore a measure of increasing cohesion should cause a decrease in WCC, or vice versa. Inheritance is a factor that has a two-fold effect. While the increased use of inheritance increases reusability, it also means greater design complexity and difficulty in implementation and maintenance.

After considering all these effects, an empirically and intuitively persuasive metric is being formulated by relating measurable design characteristics together with the quality contributors as follows:

$$\text{WCC} = (\text{RFC} * \text{Level}) + \text{LCOM} \qquad (1)$$

where RFC (Response For Class) is based on the formulation for orthogonal software given below as [6]:

$$\text{RFC} = \text{WMC} + \text{CBO}$$

LCOM is the Lack of Cohesion Metrics [15-17], WMC is Weighted Method Per Class Metrics and CBO is the Coupling Between Objects Metrics.

3. Theoretical Validation

RFC is measuring the coupling in addition to encapsulation. As the deeper the class is embedded in hierarchy, the greater would be the number of inherited methods and hence greater the design complexity. This gives an idea to consider ‘level’ and find the product of RFC and level (RFC * Level), which shows the additional effort of implementing the class with RFC calculated at the particular level. So that it gives an indication of inheritance and coupling in addition to encapsulation. An increase in this factor would increase the complexity measure of WCC. The addition of LCOM indicates the cohesion of a class. A higher cohesion (lower LCOM) indicates a good design. So, adding LCOM implies that if cohesion is low, LCOM will be high, therefore WCC should be increased for low cohesion [3].

Summing up all these impacts, it is now clear that WCC is directly proportional to RFC * level, to LCOM and/or inversely proportional to Cohesion i.e.:

$$\text{WCC} \propto \text{RFC} * \text{level}$$

$$\text{WCC} \propto \text{LCOM}$$

$$\text{WCC} \propto \frac{1}{\text{Cohesion}}$$

After combining all these equations, a single class metric WCC may be defined as:

$$\text{WCC} \propto (\text{RFC}* \text{Level}) + \text{LCOM}$$

4. Experimental Validation

This section assesses how well the metrics-WCC 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, the validation of the predictability of metrics requires the set of object oriented designs with the 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 said metrics WCC needed to agree with the generally accepted requirements or characteristics of overall quality designs as perceived by analysts, developers and customers. Keeping these objectives in mind, overall software quality produced by WCC has been validated in three phases. Phase 1, *Design of Viable Experiment,* describes the selection of model validation suit. Assessments of projects overall quality and the evaluation of project design using WCC has been discussed in Phase 2 under the heading *Pre-Tryout.* A large sample of data has been used to validate the proposed metric as per experimental design and statistical analysis and interpretation of data gathered through the tryout has been interpreted in Phase 3.

4.1. Design a viable experiment

The two applications used in this empirical study to validate the integrated object-oriented metric set are industrial strength software developed by software industry based in Delhi, India. Name of the Projects, Class Diagram, Analyzer used and actual source data is being concealed, as per wishes of the company’s management. We have assured the authenticity of source data, to the best possible extent [10]. We labeled the applications as: System A and System B as shown in Table 1. System A was commercial software implemented in C++ and consisted 11 classes. System B was also implemented in C++ and consisted 8 classes. The industry professionals themselves have used full-scale code analysis system for estimating the quality of these systems [10]. They have rated the quality of both the software systems as ***'Low'***. These quality ratings by industry professionals have been taken as the benchmark value for the projects under study.

Table 1. Applications used in empirical study

Project	Classes	Quality
System A	11	Low
System B	8	Low

4.2. Perform pre-tryout

A group of software developers was assigned to study the quality of the two projects: System A & System B in the validation suite. All the developers had 8 to 12 years of experience in commercial software development, had knowledge of the object oriented paradigm, and had developed software using C++. The study was done over a period of one month. All the participants analyzed each project's design and used the metrics WCC to assign the quality to these software Systems A and B.

The descriptive statistics and the correlations between the metrics for each system are given in Tables 2-5. The values in bold are the mean value of the integrated metric set. The descriptive statistics for System A and B are included on 11 C++ and 8 C++ classes respectively.

Table 6 summarizes the results of the correlation analysis for the integrated metric set over the two software systems. The column lists the correlation values for each pair of metrics in the integrated metric set and the rows list the system. In the table Metric 1 X Metric 2 = correlation between Metric 1 and Metric 2.

Examining Table 6 shows that for System A, all of the metrics are highly correlated with each other, with WCC and (FRC*Level) being the most significantly correlated. This suggests low quality code because WCC increases due to the increase in FRC and not due to the increase in LCOM. The same is true for System B.

Table 2. Descriptive statistics for System A

	Min.	Max.	Mean	Std. Deviation
WCC	1	6	**3.18**	1.99
RFC	2	5	**3**	.83
Level	0	1	**.55**	.50
LCOM	1	3	**1.55**	.82
CBO	0	3	1.00	1.09
WMC	1	3	2.45	.68

Table 3. Correlation analysis for System A

	WCC	RFC	LCOM	RFC*Level
WCC	1	-.009	**.85**	**.91**
RFC		1	.07	.03
LCOM			1	**.31**
RFC*level				1

Table 4. Descriptive statistics for system B

	Min.	Max.	Mean	Std. Deviation
WCC	1	8	**2**	2.44
RFC	2	12	**3.25**	3.28
Level	0	1	.13	.35
LCOM	1	2	1.13	.35
CBO	0	6	1.50	1.85
WMC	1	7	3.75	2.25

Table 5. Correlation analysis for System B

	WCC	RFC	LCOM	RFC*Level
WCC	1	.27	**.74**	**.97**
RFC		1	.01	.22
LCOM			1	.01
RFC*level				1

Table 6. Correlation analysis summary

	WCCXLCOM	WCCX (RFC* Level)	LCOMX (RFC* Level)
System A	**.85**	**.91**	**.31**
System B	**.74**	**.97**	**.01**

4.3. Perform tryout

A large sample of data has been used to validate the proposed metric as per experimental design and statistical analysis and the interpretation of data gathered through the tryout has been interpreted. For this, another set of six projects were used in the same industry. We labeled the applications as: System A, System B, System C, System D, System E and System F. All these systems were commercial software implemented in C++ and consisted approximately 10-20 classes. The industry professionals themselves have used full-scale code analysis system for estimating the quality of these systems. Table 7 summarizes the quality ranking of these software systems given by industry professionals. These quality rankings have been considered as the benchmark for the projects under study.

Table 7. Quality ranking of systems

Projects	Classes	Quality Ranking
System A	11	*Low*
System B	9	*Low*
System C	12	*Low*
System D	19	*High*
System E	18	*High*
System F	15	*Low*

In order to investigate the correlations and relationships between the object oriented metric WCC and software quality, a correlation and a multiple linear regression analysis has been conducted for six projects. Definitions and discussions on the terminologies used in the correlation and regression analysis are providing in the following section. A multiple linear regression model for application in the context may be defined as follows [9]:

$$Y= a + b_1 X_1+ b_2 X_2+\ldots\ldots\ldots..+ b_n X_n \qquad (2)$$

The various components of the regression model along with other statistical terminologies used are listed here.

Independent Variable (X_i's): 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. In the regression model (2), the *X_i's* denote the independent variables.

Dependent Variable (Y): The dependent variable in an experiment is the variable that the investigator measures to determine the effect of the independent variable. In the regression Eq. (2), variable Y denotes the dependent variable.

Coefficients (a_i's): The estimated multiple linear regression coefficient measures 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. In the regression model (2), a_i's represent the coefficient terms.

Linear Correlation Coefficient (r): The linear correlation coefficient expresses quantitatively the magnitude and direction of the linear relationship between two variables. The sign of the coefficient tells us whether the relationship is positive or negative. The numerical part of the correlation coefficient describes the magnitude of the correlation. The higher the number, the greater the correlation.

The descriptive statistics and the correlations between the metrics for each system are given in Tables 8-19. The values in bold are the mean value of the integrated metric set. Table 20 summarizes the results of the correlation analysis for the integrated metric set over the two software systems. The column lists the correlation values for each pair of metrics in the integrated metric set and the rows list the system. In the table Metric 1 X Metric 2 = correlation between Metric 1 and Metric 2.

Examining Table 20 shows that for all the systems, all of the metrics are highly correlated with each other, with WCC and (FRC*Level) being the most significantly correlated.

The multiple linear regression model listed in Eq. (2) was fitted to the minimal set of the metric and shown in Eq. (3) for System A, B, C, D, E and F respectively and the results are given in Table 21.

$$WCC = a + b_{RFC*Level}\,(RFC*Level) + b_{LCOM}\,LCOM \qquad (3)$$

The standardized beta weight (β_i's) and raw score beta weight (b_i's) has been calculated and shown in Table 21.

The computed standardized beta weights (β_i's) in Table 21 for all the systems show that the RFC*Level component has the most significant contribution on WCC. It is also evident from the raw score beta weights (b_i's). LCOM component also has a considerable significant contribution on WCC which is depicted through both the beta weight (β_i's) and raw score beta weight (b_i's). Examining the F ratio in Table 21, it is clear that the regression shown in Eq. (3) is significant at .01 level of significance for the Systems A, D, E and F and at .05 for the Systems B and C.

Table 8. Descriptive statistics for System A

	Min.	Max.	Mean	Std. Deviation
WCC	1	10	**4.9**	2.36
RFC	4	8	**6**	1.29
Level	0	1	**.58**	.49
LCOM	1	3	**1.66**	.76
CBO	1	3	2.08	.76
WMC	2	6	3.91	1.14

Table 9. Descriptive statistics for System B

	Min.	Max.	Mean	Std. Deviation
WCC	1	10	**5.11**	3.21
RFC	2	8	**4.77**	1.92
Level	0	1	**.66**	.50
LCOM	1	2	**1.33**	3.40
CBO	1	3	1.88	4.83
WMC	1	5	2.88	7.49

Table 10. Descriptive statistics for System C

	Min.	Max.	Mean	Std. Deviation
WCC	1	8	**4.66**	2.64
RFC	5	8	**6**	1.34
Level	0	1	**.58**	.51
LCOM	1	3	**1.66**	.77
CBO	1	3	2.08	.79
WMC	2	6	3.91	1.16

Table 11. Descriptive statistics for System D

	Min.	Max.	Mean	Std. Deviation
WCC	1	11	**3.21**	3.10
RFC	2	11	**6.63**	2.73
Level	0	1	**.42**	.50
LCOM	1	3	**1.68**	.82
CBO	1	6	2.52	1.61
WMC	1	9	4.10	2.30

Table 12. Descriptive statistics for System E

	Min.	Max.	Mean	Std. Deviation
WCC	1	8	**3.11**	2.78
RFC	5	11	**6.94**	1.83
Level	0	1	**.33**	.48
LCOM	1	2	**1.33**	.48
CBO	1	5	2.72	1.40
WMC	2	9	4.22	1.92

Table 13. Descriptive statistics for System F

	Min.	Max.	Mean	Std. Deviation
WCC	1	10	**3.40**	3.2
RFC	2	8	**5.13**	1.92
Level	0	1	**.40**	.48
LCOM	1	2	**1.46**	.49
CBO	1	5	2.26	1.48
WMC	1	6	2.86	1.58

Table 14. Correlation analysis for System A

	WCC	RFC	LCOM	RFC*Level
WCC	1	.80	**.02**	**.88**
RFC		1	-.18	.83
LCOM			1	**-.06**
RFC*level				1

Table 15. Correlation analysis for System B

	WCC	RFC	LCOM	RFC*Level
WCC	1	.89	**.28**	**.98**
RFC		1	.08	.90
LCOM			1	**.13**
RFC*level				1

Table 16. Correlation analysis for System C

	WCC	RFC	LCOM	RFC*Level
WCC	1	-.53	**.24**	**.59**
RFC		1	.08	-.11
LCOM			1	**.01**
RFC*level				1

Table 17. Correlation analysis for System D

	WCC	RFC	LCOM	RFC*Level
WCC	1	.06	**.57**	**.46**
RFC		1	-.05	.39
LCOM			1	**.23**
RFC*level				1

Table 18. Correlation analysis for System E

	WCC	RFC	LCOM	RFC*Level
WCC	1	-.63	**.45**	**.99**
RFC		1	-.31	.99
LCOM			1	**.29**
RFC*level				1

Table 19. Correlation analysis for System F

	WCC	RFC	LCOM	RFC*Level
WCC	1	.58	**.01**	**.25**
RFC		1	.40	.25
LCOM			1	**.67**
RFC*level				1

Table 20. Correlation analysis summary

	WCCXLCOM	WCCX (RFC* Level)	LCOMX (RFC* Level)
System A	**.02**	**.88**	-.06
System B	**.28**	**.98**	**.13**
System C	**.24**	**.59**	**.01**
System D	**.57**	**.46**	**.23**
System E	**.45**	**.99**	**.29**
System F	**.01**	**.25**	**.67**

Table 21. Regression analysis summary

Estimated Parameters	Systems					
	A	B	C	D	E	F
$\beta_{RFC*Level}$	.88	.95	.58	.35	.93	.43
β_{LCOM}	.07	.16	.23	.49	.17	.07
$b_{RFC*Level}$	.70	.77	.58	.38	.75	.44
b_{LCOM}	.21	.15	.78	1.85	.98	.45
a	1.94	2.68	1.87	-1.17	.35	1.50
F ratio	16.5	6.11	4.18	10.4	9.11	4.21

Table 22. χ^2 test observations

	High	Low	Total
WCC	8_A	2_B	10
Industry Rating	2_C	8_D	10
Total	10	10	20
			Value of χ^2 is 5.0

4.4. Analysis and interpretation

Examining Table 20 shows that for all the systems, all of the metrics are highly correlated with each other, with WCC and (FRC*Level) being the most significantly correlated. In order to further assure, χ^2 test has been used for testing the null hypothesis stated as follows:

H_0: Quality estimates obtained through WCC are not significantly comparable/close to those obtained from industrial quality experts.
H_a: Quality estimates obtained through WCC are significantly comparable/close to those obtained from industrial quality experts.

WCC's values of all the six projects have been tested using the Chi-Square Test (χ^2). The χ^2 test applies only to discrete data, counted rather than measured values and hence becomes readily applicable in our context. The χ^2 test is not a measurer of the degree of relationship. It is merely used to estimate the likelihood that some factor other than chance (sampling error) accounts for the apparent relationship. Because the null hypothesis states that there is no relationship (the variables are independent), the test merely evaluates the probability that the observed relationship results from the chance. As in other tests of statistical significance, it is assumed that the sample observations have been randomly selected.

The Chi-Square observations for all the 10 systems are listed in Table 22 by using Eq. (4) [14], applicable for small samples as frequencies of cells are fewer than 10. The assumptions made for WCC values are low for less than or equal to four and high for greater than four and the degree of freedom may be calculated by using the formula df=(row-1)(column-1).

$$\chi^2 = \frac{N[\,|\,AD - BC\,|\, - N/2]^2}{(A + B)(C + D)(A + C)(B + D)} \tag{4}$$

In Eq. (4), A, B, C and D are being replaced by 8_A, 2_B, 2_C and 8_D respectively. The computed value of χ^2 is greater than the critical value of χ^2 for 1 degree of freedom at .05 level of significance, which are 3.84. The test indicates that there is a significant relationship between the WCC value and industry rating for quality of all the systems at the .05 level of significance. Hence, the null hypothesis is rejected and it leads to the inference that ***'WCC gives the same result regarding quality for all the systems as it was obtained by using full-scale code analyzer, by the organization'.***

A critical examination of the results obtained from the tryout leads to the following implications and future scopes:

- WCC gives the same result regarding quality for all the systems as it was obtained by using full-scale code analyzer.
- It will help to evaluate the quality of software and provide the cost estimates of a software project that facilitate the estimation and planning of new activities in an early stage of development life cycle.
- It may be further extended to discover the underlying errors in the software design at the early stage of software development life cycle leading to reduce effort on quality assurance and the avoidance of unnecessary overhead.

5. Conclusions

All metrics available eventually conduct measures taking class as a basis, whereas the proposed object oriented metric is a single class based metric. It caters to all the aspects of object-oriented design, i.e. encapsulation, coupling and cohesion. The metric may be used to indicate the software quality in the early stage of SDLC to monitor the cost impact of modification and improvement. Much cannot be said about the value of this metric, before it is used on a large-scale basis and critically examined. But, this metric has a certain impact and value in terms of integral effect and reduction in effort for estimating the quality and reliability of object oriented software. This eventually leads to evaluate reusability and testability/maintainability of software. Early use of these metrics may be helpful to the developer to fix problems, remove irregularities and non-conformance to standards and eliminate unwanted complexities in the early development cycle. It may be used to ensure that the analysis and design have favorable internal properties that will lead to the development of quality end product. WCC may significantly help in reducing rework during and after implementation to design effective test plans.

References

[1] Khan, R. A. and Mustafa, K. “Quality Estimation of Object Oriented Code in Design Phase”. *Developer IQ*, 4, No. 2 (2004).

[2] Abreu, F. Brito and Carpuca, Rogerio “Candidate Metrics for Object Oriented Software within a Taxonomy Framework”. *Proceeding of AQUIS’93,* Venice, Italy, October 1993; selected for reprint in the *Journal of Systems and Software*, 23, No. 1 (1994), 87-96.

[3] Letha, E.; Carl, D. and Wei, L. “A Statistical Comparison of Various Definitions of the LCOM Metrics”. *Technical Report TR-UAH-CS-1997-02, Computer Science Dept., Alabama in Huntsville,* 1997. http://www.cs.uah/tech-reports/TR-UAH-CS-1997-02.pdf

[4] Aline, L. B. “Formal Definition of Object Oriented Design Metrics”. *MS Thesis, Vrije Universiteit Brussel*, Belgium, 2002.

[5] Khan, R. A. and Mustafa, K. “A Review of SATC Research on OO Metrics”. *Proceedings, National Conference on Software Engineering Principles and Practices, SEPP-04,* 2004.

[6] Victor, L. and Charles, C. “Principal Components of Orthogonal Object-oriented Metrics (323-08-14)”. *White Paper Analyzing Results of NASA Object Oriented Data,* 2003.

[7] Xenos, M.; Stavrinoudis, D.; Zikouli, K. and Christodoulakis, D. “Object Oriented Metrics: A Survey”. *Proceedings of the FESMA 2000, Federation of European Software Measurement Associations,* Madrid, Spain, 2000.

[8] Khan, R. A.; Mustafa, K. and Yadava, S. “Quality Assessment of Object Oriented Code in Design Phase”. *Proceedings, QAI 4th Annual International Software Testing Conference,* Pune, India, 2004.

[9] Bansiya, J. “A Hierarchical Model for Object-oriented Design Quality Assessment”. *IEEE Transaction on Software Engineering*, 28, No. 1 (2002).

[10] Telesoft India Pvt. Ltd. “Unpublished Project Documentation for System A and B”. C- 56/14, Industrial Area, Sec-62, Noida (UP), India-201304 (2004).

[11] Hitz, M. and Montazeria. B. “Chidamber and Kemerer’s Metrics Suite: A Measurement Theory Perspective”. *IEEE Transactions on Software Engineering,* 22, No. 4 (1996), 267-271.

[12] Chidamber, S. R. and Kemerer, C. F. “A Metrics Suites for Object Oriented Design”. *IEEE Transaction on Software Engineering*, 20, No. 6 (1994), 476-493.

[13] Basili, B.; Briand, L. and Melo, W. L. “A Validation of Object Oriented Metrics as Quality Indicators”. *IEEE Trans. Software Engineering,* 22, No. 10 (1996), 751-761.

[14] Best, W. and Kahn, V. *Research in Education*. 6th ed., PHI, 1992.

[15] Henry, L. W.; Kafura, S. and Schulman, D. R. “Measuring Object-oriented Design”. *JOOP,* (1995), 48-55.

[16] Li, W. and Henry, S. “Maintenance Metrics for the Object-oriented Paradigm”. *Proceeding of the First International Software Metrics Symposium*, (1993), 52-60.

[17] Li, W. and Henry, S. “Object Oriented Metrics that Predict Maintenance”. *J. Systems and Software*, 23, No. 2 (1993), 111-122.

تصديق تجريبي لوسائل قياس جودة التصميم الشيئي

رائيس أ. خان[١]، و ك. مصطفى[٢]، و س. أي. أحسن[٣]

[١]قسم نظم المعلومات، جامعة بي بي أ، لكناو، الهند

[٢]جامعة الملك فيصل، المملكة العربية السعودية

[٣]قسم علوم الحاسب، جامعة ملية اسلامية، نيودلهي، الهند

(قدّم للنشر في ١١/٠٣/٢٠٠٦م؛ وقبل للنشر في ١٣/١٢/٢٠٠٦م)

ملخص البحث. يصف هذا البحث قياس مبني على درجة واحدة مدمجة تسمى درجة التعقيد الموزونة (WC)، للتصميم الشيئي. يدرس القياس من ناحية نظرية القياسات بالأخذ بعين الاعتبار الخواص الشيئية المعروفة والتي صمم القياس لتقييمها. تمت معالجة البيانات التجريبية، والتي جمعت من ثمانية مجالات تطبيقية مختلفة باستخدام قياس WCC لمساندة التصديق النظري. تشير النتائج إلى أن القياس المقترح يمكن استخدامه كمؤشر مجمل لجودة النظم البرمجية الشيئية في مراحل التطوير الأولي، لإزالة المشاكل، والتشوهات، وعدم المطابقة للمعايير، والتعقيدات الغير مرغوبة. يمكن استخدام هذا القياس من الحصول على تصميمات ذات خواص داخلية إيجابية، والتي ستؤدي إلى تطوير منتوج نهائي ذا جودة. يمكن WCC من التقليل من إعادة الأعمال خلال وبعد التطبيق لتصميم خطط ناجحة للاختبار الفعال. ولكن قبل تعميم النتائج يجب إجراء دراسات إضافية.