Volume 4, Issue 11, November 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Maintainability Measurement Model of Object Oriented Design

Geeta Laxmi
M.Tech.Scholer
Department of Comp. Sc. & Engg
Integral Uni. Lucknow, India

Mrs. Kavita Agrawal
H.O.D
Department of Comp. Sc. & Engg
Integral Uni, Lucknow, India

Dr. Rizwan Beg
Professor
Department of Comp. Sc. & Engg
Integral Uni, Lucknow, India

***Abstract*— Measuring maintainability of object oriented design early in the development process, particularly at design phase; greatly reduce the overall development cost and effort, and in addition support the designers and developers for producing high quality maintainable and reliable software within time and budget. To design and deliver quality products inside time and financial plan maintainability plays a very important role. This paper shows the need and significance of maintainability at design phase and build up a multivariate linear model "Maintainability Measurement Model" for Object-Oriented Design. Developed model estimates the maintainability of class diagrams in respect of their analyzability, understandability and modifiability.**

In this paper an endeavor has been made to establish a link among object oriented design properties and maintainability factors analyzability, understandability and modifiability as a first contribution. Despite the fact that at the same time, in order to quantify class diagram's analyzability, understandability and modifiability the study further developed three additional multivariate models. Finally the proposed models have been validated using experimental tryout.

***Keywords*—Software maintainability, maintainability measurement, Software metrics, Object Oriented Design, Software Quality.**

I. INTRODUCTION

Software Engineering has turn into extremely essential discipline of study, practice and research. Everyone are working hard to decrease the problems and to meet the purpose of developing high-quality maintainable software that is delivered on time, within budget, and furthermore satisfies the requirements [1]. Software has become significant to expansion in almost all areas of human endeavour. The skill of programming only is no longer enough to make large programs. There are serious problems in the price, timeliness, maintenance and quality of many software products [2]. Software engineering has the purpose of solving these problems by producing feature- quality maintainable software within time, financial plan. To attain this goal, we encompass to center in a closely controlled way on both the quality of the product and on the process used to develop the product [4].

II. RESEARCH AREA BACKGROUND

Software is going away to be changed several times for different reasons while being developed and particularly after it has been delivered. Commonly the term maintenance is used when referring to those changes made to software products after they have been delivered Depending on the reasons for alteration and the wider organizational perspective, a variety of approaches to maintenance such as corrective or adaptive maintenance are or relatively should be applied[3,15]. Despite the truth that software maintenance is a costly and difficult task; it is not correctly managed and often unnoticed. One cause for this poor management is the lack of established measures for software maintainability [30, 9].

With all cries and dissatisfaction, discipline is improving and maturing day by day. New solutions are being provided in the niche areas and encouraging results are being observed. We do feel that within couple of years, situation is bound to improve and software engineering shall be a stable and mature discipline, reasons that the complexity of software tends to be high. Whereas any system with a high degree of complexity will be hard to reach a certain level of reliability, as an outcome [5, 6]. There is an imperative demand to put into practice software engineering concepts, strategy, practices to avoid deviation, and to improve the software development process in order to deliver good quality maintainable software in time and within budget [24, 26].

111. MOTIVATION

A software maintenance system is motivated by a number of factors:

- **To make available stability of service**: This entails setting up bugs, improving from failures, and willing to help changes in the operating system and hardware.
- **To maintain compulsory upgrades**: These are usually caused by changes in government regulations, and also by attempts to maintain a competitive edge over competitor products.
- **To support user requests for improvements**: It include enhancement of functionality ad better performance.
- **To make possible future maintenance job**: This usually involves code and database reorganization and updating documentation.

IV. SOFTWARE MAINTAINABILITY

The key word of "maintainability" appeared in the classification of maintenance. It is also programmed as the first key attribute of good designed software by Sommerville [7] in the opening to his book. Maintainability is one of the most significant characteristics of software quality. The mass of software companies splurge 60 to 70 percent of resources for correcting, adopting and maintaining the existing software [8]. The majority of companies pay out over 70 percent budget on testing, maintenance of the software to manage the quality [10, 16, 17].Maintainability measurement helps to examine the maintenance effort and easiness of software at design level. The maintainability definition according to IEEE glossary of Software Engineering is "the ease with which a software system or component can be modified to correct faults, get better performance or other attributes, or adapt to a change background". Software maintenance required additional attempt than any other software engineering activity [28]. The maintainability of software is not possible directly, but with the help of their internal characteristics measurement. Noticeably a definition of maintainability wishes to be strongly link to the term maintenance [23, 25, 27].

V. ROLE OF DESIGN PHASE

Measuring software maintainability early hours in the development life cycle, mainly at the design phase, may facilitate designers to incorporate necessary improvement and corrections for improving maintainability of the final software. If we can calculate the maintainability at the early stages of the development life cycle, the price of the software can be economical [12, 13]. Maintainability uniqueness point out the product's ability to be modifiable, maintainable, and updateable. Early measurement of maintainability facilitates to make use of its attributes more proficiently to control/improve the quality of software [18, 19]. The design phase measurement of software maintainability is additional practicable for software development and maintenance cost-effectively.

Maintainability is one of the most perfect quality indicators [11]. A truthful measurement of maintainability is a sign of enhanced designing, improve quality product and low maintenance cost. Maintainability is a quality factor; its measurement or evaluation can be used to analyse the amount of effort required for maintaining the software and help out to allocating required resources [20, 21].

TABLE I

Study/Author	Design Phase
J Voas et al. (1992)	Design Phase
JMC & Srinivas (1996)	Design Phase
Jungmayr (2002)	Design Phase
S. Mouchawrab (2005)	During Analysis and Design Phase
Jerry et al. (2005)	Design Phase
Kolb et al. (2006)	Design Phase
D.Esposito (2008)	Design Phase
Nazir et al. (2009)	Early in the development Process
Nazir et al. (2010)	Design Phase
Nazir et al. (2010)	Design Phase
Nazir &Khan (2012)	Design Phase

Maintainability Measurement at Design Phase Consideration by Various Experts [22]

VI. OOD CHARACTERISTICS

Object oriented technology have turn into the most accepted and recognizable concept in software industry. Object oriented notion is now broadly used by software industry. Despite the truth that technology is not grown-up enough from testing point of view [27, 28, 29], almost everybody speak about it, approximately everyone state to be doing it and nearly everyone says that it is superior than conventional function oriented design. For the reason that most of the centre of the object oriented approach to software development has been on analysis and design phase, only a small research studies have been faithful to explore the concept of maintainability in object oriented system.

Object oriented ideology direct the designers what to carry and what to stay away from. Numerous measures have been defined so far to estimate object oriented design. There are various important themes of object orientation that are identified to be the foundation of internal quality of object oriented design and support in the perspective of measurement. These themes significantly take account of cohesion, coupling, inheritance, and encapsulation [31, 32].

Cohesion refers to the internal Maintainability within the parts of the design. A class is cohesive when its parts are extremely correlated. It should be complicated to divide a cohesive class. Cohesion can be used to recognize the badly designed classes. Coupling indicates the association or interdependency among modules. For illustration, object A is coupled to object B if and only if A send a message to B that means the number of association between classes or the number of messages passed between objects. Coupling is a measure of interconnecting between modules in a software organization. Inheritance is the sharing of attributes and operations amongst classes based on a hierarchical relationship. It is a method whereby one object acquires uniqueness from one, or more other objects. Inheritance occurs in all levels of a class hierarchy.

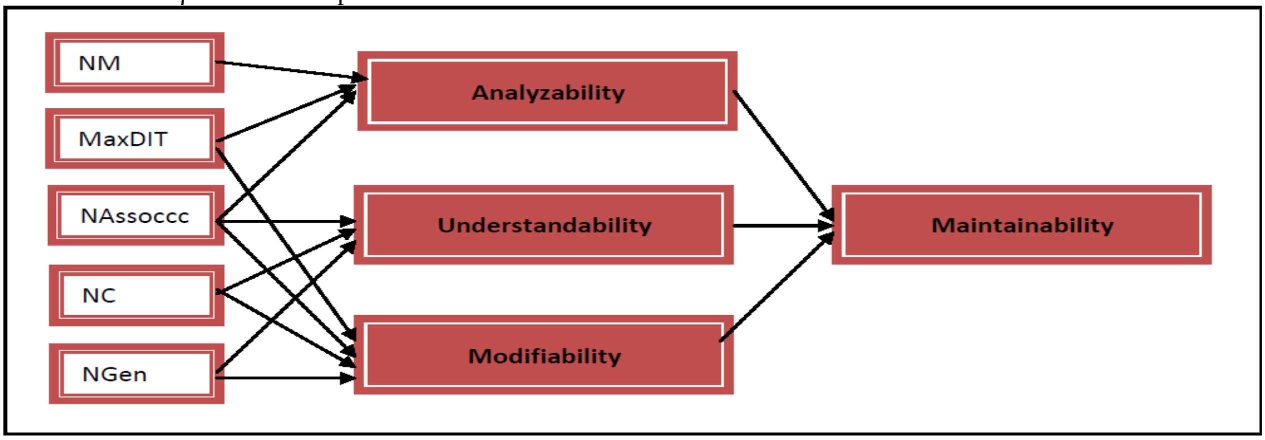
VII. MODELS DEVELOPMENT

Measurement of class diagram's analyzability, understandability and modifiability is precondition for the accurate maintainability measurement model. Therefore before developing MMMOOD, the study has developed three models for Analyzability, Understandability and Modifiability. In order to set up all the three models following multivariate linear model (1) has selected.

$$Y = \mu + \beta_1 * X_1 + \beta_2 * X_2 + \ldots\ldots + \beta_n * X_n + \varepsilon \qquad (1)$$

Where

- Y is dependent variables. X1, X2… Xn are independent variables. ß1, ß 2… ß n are the coefficients. € is error term and µ is the intercept.



The above Fig. 1 describes the measurement process of the maintainability model

A. Analyzability Measurement Model

In order to create a multivariate model for analyzability of class diagram, metrics listed in [33], will play the role of independent variables whereas analyzability will be in use as dependent variable. The data used for developing analyzability model is taken from [33] that have been composed during the controlled experiment. It comprises a set of 28 class diagrams (denoted as D0 to D27) and the metrics value of each diagram. In addition, the mean value of the expert's rating of analyzability of these diagrams is also known and termed as 'Known Value' in this research paper.
The correlation in the middle of maintainability factors and object oriented characteristics has been established as depicted in Figure 1. As per the mapping, Metrics 'NM, Max DIT, N Assoc' are selected from [33] as independent variable to build up the analyzabiliy measurement model. Using SPSS, values of coefficient are calculated and analyzability model is formulated as given below

Analyzability=1.619 - .016 x NM + .377 x NAssoc + .671 MaxDIT

Moreover the evaluation of R2 (Coefficient of Determination) and adjusted R2 in the Table 1, is also very encouraged. As, it refers to the percentage or proportion of the complete difference in Analyzability by all the three metrics (independent variables) participating in the model (2).

Table 1- Model Summary

R	R Square	Adjusted R Square	Std. Error of the Estimate
.988[a]	.977	.954	.38788
a. Predictors: (Constant), MaxDIT, NAssoc, NM			

ANOVA Table shows the result of the ANOVA test. In this Table, I get F ratio of **42.046** with (**3, 3**) degree of freedom. Obtained value is bigger than the critical value of F is **5.13** for the **0.05** significance level

Table 2-ANOVA[b]						
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	18.977	3	6.326	42.046	.006[a]
	Residual	.451	3	.150		
	Total	19.429	6			
a. Predictors: (Constant), MaxDIT, NAssoc, NM						
b. Dependent Variable: Analyzability						

Statistical importance of Independent Variables

It can be noticed from the end column of Table 3, that all of the three metrics participating in the model is statistically significant at a significance level of 0.05 (equivalent to a confidence level of 95%).

Table 3-Coefficients[a]				
		Unstandardized Coefficients		Standardized Coefficients
Model		B	Std. Error	Beta
1	(Constant)	1.619	.365	
	NM	-.016	.018	-.303
	NAssoc	.377	.50	1.065
	MaxDIT	.671	.222	.546

B. Modifiability Measurement Model

The data used for developing modifiability model is taken from [33]. The association in the middle of maintainability factors modifiability and object oriented characteristics has been established as depicted in Figure 1. As per the mapping, Metrics 'NC, N Assoc', NGen, Max DIT are selected from [33] as independent variable to build up the modifiability measurement model. Using SPSS, values of coefficient are calculated and modifiability model is formulated as given below. After establishing a model for analyzability the next task is to build a similar model for modifiability also.
Applying the similar technique of multiple regressions on the available data resulted into the following Modifiability model (3).

Modifiability=1.088+.283 $_X$ NC - .079 $_X$ NAssoc -.067 $_X$ NGen+.170 $_X$ MaxDIT

Table 4- Model Summary				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.980[a]	.960	.879	.62499
a. Predictors: (Constant), MaxDIT, NAssoc, NGen, NC				

ANOVA Table highlights the result of the ANOVA examination.

Table 5- ANOVA[b]						
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	18.647	4	4.662	11.935	.079[a]
	Residual	.781	2	.391		
	Total	19.429	6			
a. Predictors: (Constant), MaxDIT, NAssoc, NGen, NC						
b. Dependent Variable: Modifiability						

Statistical importance of Independent Variables

As long as statistical impact and consequence of individual independent variables in the modifiability model (2) is concern.

Table 6- Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients
		B	Std. Error	Beta
1	(Constant)	1.088	.923	
	NC	.283	.601	1.341
	NAssoc	-.079	.822	-.204
	NGen	-.067	.165	-.316
	MaxDIT	.170	.947	.122

a. Dependent Variable: Modifiability

C. Understandability Measurement Model

The data used for developing understandability model is taken from [33]. The relationship among maintainability factors and object oriented characteristics has been established as depicted in Figure 1.Applying the same practice of multiple regressions on the available data resulted into the following understandability model (3).

Understandability =1.139 - .201 x NC+ .571 x NAssoc -.273 x NGen

Table 7- Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.973[a]	.947	.788	.82344

a. Predictors: (Constant), NGen, NAssoc, NC

ANOVA Table highlights the result of the ANOVA examination.

Table 8- ANOVA[b]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	12.122	3	4.041	5.959	.290[a]
	Residual	.678	1	.678		
	Total	12.800	4			

a. Predictors: (Constant), NGen, NAssoc, NC

b. Dependent Variable: Understandability

VIII. MAINTAINABILITY MEASUREMENT MODEL (MMMOOD)

Previously than establishing the model for maintainability, it is necessary to make positive the proper correlation amongst maintainability, analyzability, understandability and modifiability of class diagrams.

Maintainability= 2.168 - . 313 x Analyzability - .938 x Understandability + 1.043 x Modifiability

The assessment of R2 (Coefficient of Determination) is nearer to 1, and adjusted R2 in the Table below, is too very hopeful.

Table 9- Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.879[a]	.773	.091	.48908

a. Predictors: (Constant), Modifiability, Understandability, Analyzability

ANOVA Table (Table 8) highlights the result of the ANOVA examination

Table 10- ANOVA[b]						
Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	.813	3	.271	1.133	.583[a]
	Residual	.239	1	.239		
	Total	1.052	4			
a. Predictors: (Constant), Modifiability, Understandability, Analyzability						
b. Dependent Variable: Maintainability						

Statistical importance of Independent Variables

As long as statistical importance and significance of individual independent variables of maintainability model (4) is concern. It can be observed from the last column of Table, that the three factors participating in the model is statistically significant at a significance level of 0.05 (equivalent to a confidence level of 95%).

Table 11- Coefficients[a]				
Model		Unstandardized Coefficients		Standardized Coefficients
		B	Std. Error	Beta
1	(Constant)	2.168	.599	
	Analyzability	-.313	1.156	-1.220
	Understandability	-.938	.745	-4.172
	Modifiability	1.043	1.543	4.896
a. Dependent Variable: Maintainability				

IX. EMPIRICAL VALIDATION

This part of study paying attention how the above designed model is competent to measure the maintainability of object oriented design at design phase of development life cycle. The empirical validation is an indispensable phase of planned research to validate maintainability measurement model for high and improved level acceptability. Empirical validation is the approved approach and practice to speak the model acceptance. Keeping vision of this truth, practical validation of the maintainability measurement model has been performed with sample tryouts. In order to validate proposed measurement model the value of metrics is available (Genero et al., 2001), data sets, taken from [33].

A. Validating Analyzability Model

The known and calculated analyzability values for the given 21 Projects (class diagrams) is shown in Table.

Projects	Analyzability Known	Analyzability Calculated
P1	1.804	2
P2	2.133	2
P3	2.558	2
P4	1.66	2
P5	2.181	2
P6	1.868	1
P7	2.558	3
P8	2.82	3
P9	2.053	3
P10	2.628	3
P11	2.829	3
P12	2.861	3
P13	2.545	2
P14	1.915	3
P15	4.394	5

P16	6.515	6
P17	4.247	5
P18	4.515	6
P19	7.566	5
P20	4.579	5
P21	4.471	4

Computed Ranking, Actual Ranking and Their Relation

Projects	Calculated Ranking	Known Ranking	**$\sum d^2$**	***r_s***	**r_s >.5487**
P1	2	2	0	1	✓
P2	6	3	9	0.9941	✓
P3	9	4	25	0.9837	✓
P4	1	5	16	0.9896	✓
P5	7	6	1	0.9993	✓
P6	3	1	4	0.997402597	✓
P7	10	8	4	0.9974	✓
P8	12	9	9	0.9941	✓
P9	5	10	25	0.9837	✓
P10	11	11	0	1	✓
P11	13	12	1	0.9993	✓
P12	14	13	1	0.9993	✓
P13	8	7	1	0.9993	✓
P14	4	14	100	0.9350	✓
P15	16	16	0	1	✓
P16	20	21	1	0.9993	✓
P17	15	17	4	0.9974	✓
P18	18	20	4	0.9974	✓
P19	21	18	9	0.9941	✓
P20	19	19	0	1	✓
P21	17	15	4	0.9974	✓

r_s >.5487 means significant results.

B. Validating Modifiability Model

The known and calculated modifiability values for the given 21 Projects (class diagrams) is shown in Table.

Projects	Modifiability Known	Modifiability Calculated
P1	1.858	2
P2	2.141	2
P3	1.7	2
P4	2.424	2
P5	1.779	2
P6	1.575	1
P7	1.983	3
P8	2.664	3
P9	1.858	2
P10	3.379	3

P11	2.868	3
P12	3.151	3
P13	2.575	2
P14	6.245	6
P15	3.592	3
P16	5.352	5
P17	6.68	6
P18	5.397	5
P19	6.319	5
P20	5.646	6
P21	5.028	4

Computed Ranking, Actual Ranking and Their Relation

Projects	Calculated Ranking	Known Ranking	**$\sum d^2$**	r_s	**r_s >.548**
P1	4	2	4	0.9974	✓
P2	7	3	16	0.9896	✓
P3	2	4	4	0.9974	✓
P4	8	5	9	0.9941	✓
P5	3	6	9	0.9941	✓
P6	1	1	0	1	✓
P7	6	9	9	0.9941	✓
P8	10	10	0	1	✓
P9	5	7	4	0.9974	✓
P10	13	11	4	0.9974	✓
P11	11	12	1	0.9993	✓
P12	12	13	1	0.9993	✓
P13	9	8	1	0.9993	✓
P14	19	19	0	1	✓
P15	14	14	0	1	✓
P16	16	16	0	1	✓
P17	21	20	1	0.9993	✓
P18	17	17	0	1	✓
P19	20	18	4	0.9974	✓
P20	18	21	9	0.9941	✓
P21	15	15	0	1	✓

r_s >.5487 means significant results.

C. Validating Understandability Model

The known and calculated understandability values for the given 21 Projects (class diagrams) is shown in Table.

Projects	Understandability Known	Understandability Calculated
P1	1.308	1
P2	1.107	2
P3	0.906	2

P4	2.249	2
P5	0.705	2
P6	1.678	2
P7	2.048	2
P8	0.529	3
P9	1.107	2
P10	0.328	3
P11	-0.62	3
P12	0.899	3
P13	0.698	3
P14	-0.958	2
P15	-4.39	6
P16	-0.645	3
P17	-5.953	5
P18	-3.539	6
P19	-6.894	5
P20	-1.348	5
P21	0.343	6

Computed Ranking, Actual Ranking and Their Relation

Projects	Known Ranking	Calculated Ranking	$\sum d^2$	r_s	r_s >.548
P1	18	1	289	0.8123	✓
P2	17	2	225	0.8538	✓
P3	15	3	144	0.9064	✓
P4	21	4	289	0.8123	✓
P5	13	5	64	0.9584	✓
P6	19	6	169	0.8902	✓
P7	20	7	169	0.8902	✓
P8	11	10	1	0.9993	✓
P9	16	8	64	0.9584	✓
P10	9	11	4	0.9974	✓
P11	8	12	16	0.9896	✓
P12	14	13	1	0.9993	✓
P13	12	14	4	0.9974	✓
P14	6	9	9	0.9941	✓
P15	3	19	256	0.8337	✓
P16	7	15	64	0.9584	✓
P17	2	16	196	0.8727	✓
P18	4	20	256	0.8337	✓
P19	1	17	256	0.8337	✓
P20	5	18	169	0.8902	✓
P21	10	21	121	0.9214	✓

r_s >.5487 means significant results.

D. Validating Maintainability Model

The known and calculated maintainability values for the given 16 Projects (class diagrams) is shown in Table.

Projects	Maintainability Known	Maintainability Calculated
P1	1.021	1.96
P2	2.168	1.752
P3	2.168	0.501
P4	2.168	1.336
P5	2.168	0.92
P6	2.168	0.92
P7	2.67	0.92
P8	3.315	-0.018
P9	2.168	1.544
P10	3.315	1.128
P11	3.315	0.92
P12	3.315	1.128
P13	3.315	0.815
P14	2.168	1.233
P15	2.67	0.085
P16	4.462	4.15

Computed Ranking, Actual Ranking and Their Relation

Projects	Calculated Ranking	Known Ranking	$\sum d^2$	r_s	$r_s > 0.63$
P1	15	1	196	0.711	✓
P2	14	2	144	0.788	✓
P3	3	3	0	1	✓
P4	12	4	64	0.905	✓
P5	5	5	0	1	✓
P6	6	6	0	1	✓
P7	7	9	4	0.994	✓
P8	1	11	100	0.852	✓
P9	13	7	36	0.947	✓
P10	9	12	9	0.986	✓
P11	8	13	25	0.963	✓
P12	10	14	16	0.976	✓
P13	4	15	121	0.822	✓
P14	11	8	9	0.986	✓
P15	2	10	64	0.905	✓
P16	16	16	0	1	✓

$r_s > 0.6324$ means significant results.

Speraman's Coefficient of Correlation (***r_s***) was used to validate the impact of correlation among 'Known Values' and it's calculated values using developed model. Rank correlation is the procedure of determining the degree of corelation connecting two variables. The '***r_s***' was calculated through the formula given as under:

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

'd' = difference between 'Calculated Values' and 'Known Values' of model.

n = number of projects (class diagram) used in the experiment.

X. CONCLUSION

This paper shows the significance of maintainability in general and as a key factor to software quality for producing high class software within time and budget.Study has developed four models to measure analyzability, understandability, modifiability and maintainability of the class diagrams. Maintainability model measure the maintainability of class diagrams in terms of their analyzability, understandability, modifiability. All the models have been developed with the help of multiple linear regressions. The paper furthermore validates the quantifying capability of developed models. Analyzability, understandability, modifiability is evidently highly appropriate and important in the perspective of software maintainability estimation. Statistical examination shows that maintainability model is statistically greatly significance and acceptable.

These models may assist software designer to assess the design and take appropriate decision to improve the design, early in the development life cycle, in order to control or reduce effort in measuring testability of object oriented design.Maintainability measurement model has been validated theoretically as well as empirically using experimental try-out. The applied validation on the maintainability model concludes that proposed model is highly consistent, acceptable and considerable.

REFERENCES

[1] J.H. Hayes and L Zhao, "Maintainability Prediction: a Regression Analysis of Measures of Evolving Systems," *Proc. 21st IEEE International Conference on Software Maintenance*, 26 - 29 Sept. 2005, pp. 601 - 604, 2005.

[2] R. Pressman, "Software Engg: A Practitioner's Approach", Sixth Ed. Mcgraw-Hill, 2005.

[3] H. Zuse, "A Framework of Software Measurement" Walter de Gruyter, 1998.

[4] ISO/IEC 9126-4:2004, "Software Engg. Product Quality-Quality in Use Metrics", ISO/IEC 2004.

[5] IEEE Computer Society. Software Engineering Standards Committee and IEEE-SA Standards Board. "IEEE Recommended Practice for Software Requirements Specifications." Institute of Electrical and Electronics Engineers, 1998.

[6] Illingworth, V., (Ed.) (1983). Dictionary of Computing. Oxford, Oxford University Press.

[7] Sommerville, I. (1992). Software Engineering. 4th ed. New York, Addison-Wesley.

[8] McClure, C. (1992). The three Rs of software automation: re-engineering, repository, reuseability New Jersey, Prentice-Hall.

[9] M. Dagpinar and J. Jahnke, "Predicting Maintainability with Object- Oriented Metrics – an Empirical Comparison," *Proc. 5th Working Conference on Reverse Engineering (WCRE'03),* 13 - 17 Nov. 2003, pp. 155 - 164, 2003.

[10] Abdullah, Dr, Reena Srivastava, and M. H. Khan. "Testability Estimation of Object Oriented Design: A Revisit." international Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 8, August 2013.

[11] J.H. Hayes and L Zhao, "Maintainability Prediction: a Regression Analysis of Measures of Evolving Systems," *Proc. 21st IEEE International Conference on Software Maintenance*, 26 - 29 Sept. 2005, pp. 601 - 604, 2005.

[12] IEEE Std. 65.12-1990. *Standard* Glossary of Software Engineering Terminology, IEEE Computer Society Press, Los Alamitos, CA, 1993.

[13] S.W.A. Rizvi and R.A. Khan, "A Critical Review on Software Maintainability Models," Proc. of the National Conference on Cutting Edge Computer and Electronics Technologies, 14 - 16 Feb. 2009, pp. 144 – 148,Pantnagar, India, 2009

[14] S. Muthanna, K. Kontogiannis, K. Ponnambalaml and B. Stacey, "A Maintainability Model for Industrial Software Systems Using Design Level Metrics", In Working Conference on Reverse Engineering (WCRE'00), 2000

[15] Hayes, J. Huffman, Mohamed, N., Gao, T. The Observe-Mine-Adopt Model: An agile way to enhance software maintainability. Journal of Software Maintenance and Evolution: Research and Practice, Volume 15, Issue 5, Pages 297 – 323, October 2003.

[16] G. DiLucca, A. Fasolino, P. Tramontana, and C. Visaggio. Towards the definition of a maintainability model for web applications. In Proceeding of the 8th European Conference on Software Maintenance and Reengineering, pages 279– 287. IEEE Computer Society Press, 2004.

[17] Hayes J.H. and Zaho L (2005), "Maintainability Prediction a Regression Analysis of Measures of Evolving Systems", Proc.21st IEEE International Conference on Software Maintenance, 26-29 Sept.2005, pp.601-604.

[18] C.V. Koten, A.R. Gray, "An application of Bayesian network for predicting object-oriented software maintainability", *Information and Software Technology Journal,* vol: 48, no: 1, pp 59-67, Jan2006.

[19] Y. Zhou and H. Leung, "Predicting object-oriented software maintainability using multivariate adaptive regression splines" , Journal of Systems and Software, vol. 80, no. 8, pp. 1349-1361,2007

[20] MO. Elish and KO. Elish, "Application of TreeNet in Predicting Object-Oriented Software Maintainability: A Comparative Study", European Conference on Software Maintenance and Reengineering, pp 1534-5351, March 2009, DOI 5.159/CSMR.2009.57

[21] Alisara Hincheeranan and Wanchai Rivepiboon," A Maintainability Measurement Model and Tool." International Journal of Computer and Communication Engineering, Vol. 1, No. 2, July 2012.

[22] Abdullah, Dr. Reena Srivastava, and Dr.M. H. Khan. "Modifiability: A Key Factor To *Testability*." International Journal of Advanced Information Science and Technology (IJAIST) Vol.26, No.26, June 2014

[23] Johny Antony p & harsh dev , Estimating Maintainability of Software System Using Object-Oriented Metrics, International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR) ISSN 2249-6831 Vol. 3, Issue 2, Jun 2013, 283-294.

[24] P. Oman and J. Hagemeister, "Metrics for assessing a software system's maintainability," *Software Maintenance*, 1992, pp. 337 - 344.

[25] J. Bansiya and C. G. Davis, "A hierarchical model for object-oriented design quality assessment," *IEEE Transaction on software engineering*, vol. 28, pp. 4-17, 2002.

[26] M. Kiewkanya, N. Jindasawat, *et al*., "A methodology for constructing maintainability model of object-oriented design," *Quality Software QSIC 2004 Proceedings Fourth International*, 2004, pp. 206 - 213.

[27] T. Yi, F. Wu, *et al.*, "A comparison of metrics for UML class diagrams," *SIGSOFT Software. Eng. Notes* vol. 29, no. 5, pp. 1-6.

[28] S. R. Ragab and H. H. Ammar, "Object oriented design metrics and tools a survey," *Informatics and Systems (INFOS)*, 2010 The 7th International, pp. 1-7.

[29] S. W. A. Rizvi and R. A. Khan, "Maintainability Estimation Model for Object- Oriented Software in Design Phase (MEMOOD)," *COMPUTING*, vol. 2, no. 4, pp. 26-32.

[30] T. Yi, "Comparison Research of Two Typical UML-Class-Diagram Metrics: Experimental Software Engineering," *International Conference on Computer Application and System Modeling*, 2012, Taiyuan, vol. 12, pp. 86-90.

[31] S. Ghosh, S. K. Dubey, *et al.*, "Comparative Study of the Factors that Affect Maintainability," *International Journal on Computer Science and Engineering*, vol. 3, no. 12, pp. 3763-3769, 2011.

[32] S. Muthanna, K. Kontogiannis, K. Ponnambalaml and B. Stacey, "A Maintainability Model for Industrial Software Systems Using Design Level Metrics", In Working Conference on Reverse Engineering (WCRE'00), 2000

[33] M. Genero, J. Olivas, M. Piattini, and F. Romero, "A Controlled Experiment for Corroborating the Usefulness of Class Diagram Metrics at the Early Phases of Object-Oriented Developments," *Proc. of the ADIS 2001, Workshop on Decision Support in Software Engineering*, vol. 84. Spain, 2001.

