

# Design Phase Estimation of Object and Aspect Oriented Software: A New Approach

Sumit Srivastava

Computer Science and Engineering  
Goel Institute of Technology & Management  
Lucknow (U.P.), India

Yogendra Pratap Singh

Assistant Professor,  
Department of Computer Science  
Goel Institute of Technology & Management  
Lucknow (U.P.), India

**Abstract**— Effective estimation is very necessary for efficient design of an object oriented software. Software developers need to read and understand source programs and other software artifacts. The increase in size and complexity of software considerably affects a number of quality attributes, especially effectiveness and testability. False analysis frequently leads to ambiguities, misunderstanding and hence to faulty development results. It is very tough to obtain an understandable view on all the probable factors that have positive impact on software testability. Researchers, practitioners and quality controllers have always argued that effectiveness should be measured as a key attribute in order to promise the quality software. Calculating effectiveness at source code level directs to late arrival of desired information. An exact measure of software quality fully depends on effectiveness estimation. This paper shows the results of a systematic literature review conducted to collect related evidence on effectiveness estimation of object oriented software. In this paper, our objective is to find the known complete and comprehensive software effectiveness estimation model and related framework for estimating the effectiveness of object oriented software at an initial stage of development life cycle.

**Keywords:** *Software Effectiveness; Quantification; Object Oriented Design Characteristics; Software Quality; Software testing, Aspect Oriented Design Characteristics.*

## I. INTRODUCTION

There are numerous approaches to make the system very much reliable. Among several available methods object oriented design is one of the important method to measure effectiveness and design effectiveness [1, 2]. Aspect oriented design signed itself as an essential approach for resolving mainly of the software problems. In an object oriented approach, the data is considered as the most significant element and it cannot move freely around the system [16]. Increase in the size of program, increases needless effort and complexity. Software complexity always increases with error handling functions. Software with high complexity generally produces software with faults [5]. High complexity always decreases effectiveness of software. Though, software faults vary noticeably with respect to their severity. A failure occurred by a fault may go ahead to a complete system crash or an incapability to open a file [12, 13].

Therefore, there is an urgent need to develop a model that can be applied to identify those classes that are prone to have serious faults. From the abovementioned discussion, it appears that reducing unwanted complexity early in the

development process leads to the development of high quality reliable end products. Effectiveness is an essential software quality factor that is useless if it is not available at an initial stage in the software development life cycle. It becomes more important in the case of object oriented design. This chapter illustrates the need and significance of effectiveness at design phase and build up a multivariate linear Effectiveness.

## II. SOFTWARE EFFECTIVENESS

Effectiveness has always been an indefinable concept. Its truthful measurement or assessment is a complex exercise for the reason that of the various potential factors influencing effectiveness. It has been find out from systematic literature review that area researchers, quality controllers and industry personnel had made significant efforts to estimate software effectiveness but at the source code level. Calculating effectiveness at source code level directs to late arrival of desired information. An exact measure of software quality fully depends on effectiveness estimation. This paper shows the results of a systematic literature review conducted to collect related evidence on effectiveness estimation of object oriented software. [3, 4, 7]. The majority of companies pay out over 70 percent budget on testing, maintenance of the software to manage the quality [5]. Effectiveness Estimation helps to examine the maintenance effort and easiness of software at design level [14]. The effectiveness 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 environment”[6,23].

Software identification processes normally focus on avoiding errors, detecting and correcting software faults that do occur, and predicting effectiveness after identification. It is well understood that delivering quality software is no longer an advantage but a necessary factor. Unluckily, the majority of the industries not only fails to deliver a quality product to their customers, but also do not understand the Estimation Model for Object-Oriented Design. Developed model estimates the effectiveness of class diagrams in respect of their Effectiveness, Effectiveness. Lastly the developed models have been validated using experimental tryout. appropriate quality attributes [20]. Software maintenance required for extra effort than any other software engineering activity [5]. The effectiveness of software is not possible directly, but with

the help of their internal characteristics Estimation [6]. Noticeably a definition of effectiveness wishes to be strongly link to the term maintenance. Effectiveness is the easiness or simplicity with which a software system can be maintained (using the definition of software maintenance above) and is a key characteristic of software [16, 18].

### III. EFFECTIVENESS MANAGEMENT

Key aspects of effectiveness management include:

- o Corporate level participation
- o Integral piece of product identification not parallel
- o Effectiveness procedures incorporated into design process
- o Built into programme plan and create a effectiveness plan
- o Ownership of the effectiveness plan inside the design team

### IV. OBJECT AND ASPECT ORIENTED DESIGN CONSTRUCTS

Object oriented design and development are well-known conceptions in today's software development environment. Object oriented design supports number of design properties such as coupling, cohesion, inheritance and encapsulation. Object oriented system consider object as the primary agent involved in a computation process. It requires more significant effort at the early phase in the software development life cycle to recognized objects, classes, and the relationships among them. Object oriented programming is a basic knowledge that supports quality objectives [13, 15]. The necessity to deal with the effectiveness of software design is the essential issue that influenced the overall development cost and quality. A good object oriented design needs design procedures and practices that must be used in development cycle [17]. Their violation will ultimately have a strong impact on the quality attributes. Object oriented principles direct the designers what to hold up and what to keep away from. A number of measures have been defined so far to measure object oriented design. There are several important themes of object orientation that are known to be the basis of internal quality of object oriented design and support in the perspective of estimation [18,19]. These themes significantly include inheritance, encapsulation, cohesion and coupling.

### VI. OBJECT AND ASPECT ORIENTED DESIGN METRICS

The most central aim of metric selection is to pick such metrics which are statistically important and must be applicable. Studies have been conducted and found that there exists powerful relation among Object Oriented software metrics and its effectiveness. Software metrics offer an effortless and inexpensive way to identify and correct probable reasons for low software quality according to the effectiveness sub -factor as this will be supposed by the programmers. Set up Estimation programs and design metric standards will support in preventing failures before the maintenance process and decrease the essential effort during that phase. Internal metrics are extremely associated with the programmers' view of effectiveness [9-12]. However, unhappiness with internal quality standards may not necessarily outcome in low rank of effectiveness although it is generally expected. In that case, it is likeable that, regardless of what internal Estimations designate, the concluding judge for the effectiveness of the delivered software is the programmer [19, 21, 22].

### VII. MODELS DEVELOPMENT

Estimation of class diagram's Effectiveness is prerequisite for the accurate effectiveness Estimation. For this reason prior to developing EEMOOD, the study has developed models for Effectiveness. In order to set up the models subsequent multivariate linear model (1) has selected.

$$Y = \mu + \beta_1 * X_1 + \beta_2 * X_2 + \dots + \beta_n * X_n + \epsilon \quad (1)$$

Where

- Y is dependent variables.
- X1, X2... Xn are independent variables.
- $\beta_1, \beta_2 \dots \beta_n$  are the coefficients.
- $\epsilon$  is error term
- $\mu$  is the intercept.

#### 4.5.1 Effectiveness Estimation Model

In order to set up an Effectiveness estimation model of object oriented class diagram, metrics listed in [8] will play the role of independent variables while Effectiveness will be taken as dependent variable. The data used for developing Effectiveness model is taken from [10].

SDMetric software has been used to calculate the values of independent variables. Firstly, class diagram has been converted into XMI file and using sdmetric, xmi file has been used to calculate the values of independent variables.

The correlation among Effectiveness Factors and Object Oriented Characteristics has been established as depicted in equation2. Using SPSS, values of coefficient are calculated and Effectiveness model is originated as below.

SDMetrics V2.33 (Demo)

Project Views Help

Metric Data Tables Histograms Kiviati diagrams

Select element type Class

Sort by No sort and No sort Highlight nothing

Name	NumAttr	NumOps	NumPubOps	Setters	Getters	Nesting	IFImpl	NOC	NumDesc	NumAnc	DIT	CLD	OpsInh	AttrInh	Dep_Out	Dep_In	NumAssEl_ssc	NumAssEl_sb	NumAssEl_nsb	EC_Attr	IC_A	
Root.Component View.Entity	1	1	1	0	1	0	0	3	5	0	0	2	0	0	0	0	0	0	0	0	2	0
Root.Component View.Car	2	Demo	1	0	1	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	3	1
Root.Component View.Wife	4	0	0	0	0	0	0	0	0	2	2	0	3	3	0	0	0	0	0	0	Demo	3
Root.Component View.Husband	3	0	0	0	0	0	0	0	0	Demo	2	0	3	3	0	0	0	0	0	0	1	2
Root.Component View.House	4	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	Demo	0	0	0	2	4
Root.Component View.Person	2	2	Demo	0	2	0	0	2	2	1	1	1	1	1	0	0	0	0	0	0	1	0

SDMetrics V2.33 (Demo)

Project Views Help

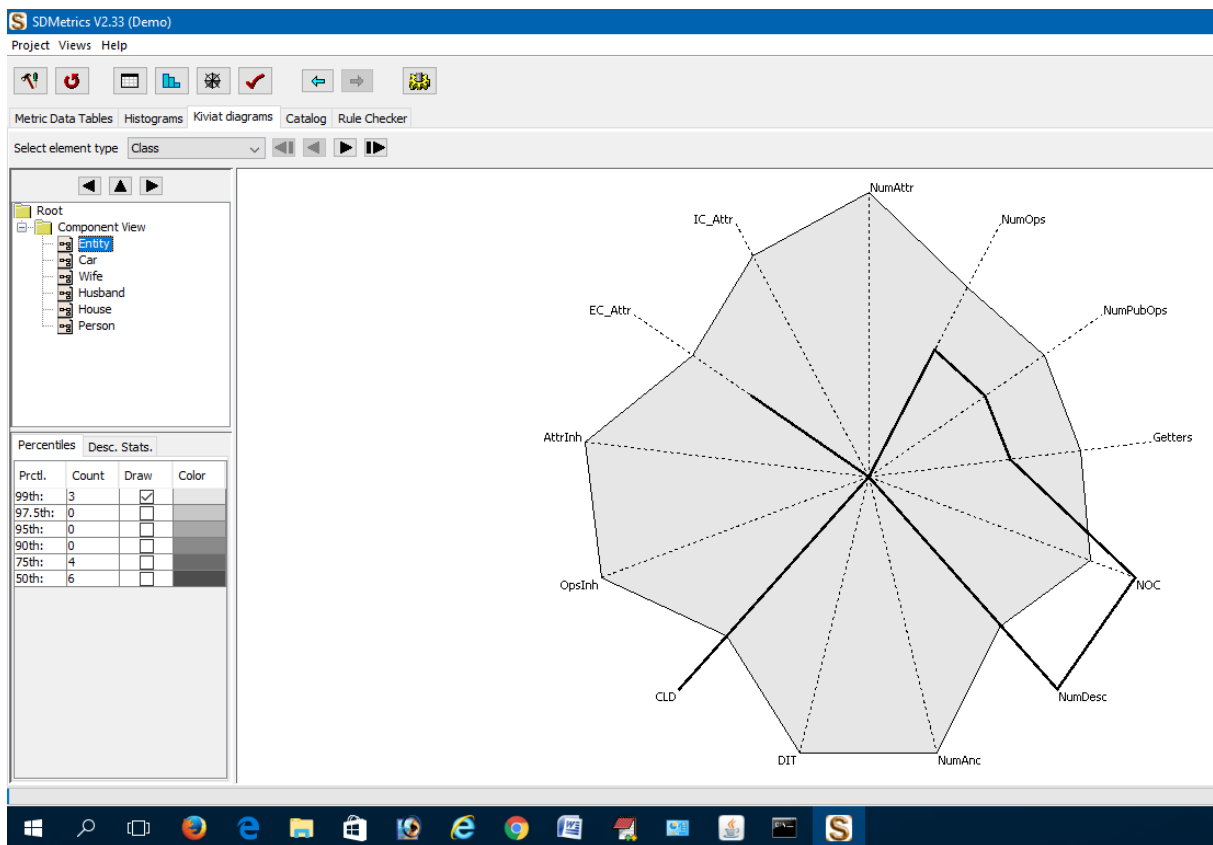
Metric Data Tables Histograms Kiviati diagrams Catalog

Show: Metrics

Sort by No sort and No sort

Name	Domain	Category	Description
NumAttr	class	Size	The number of attributes in the class.
NumOps	class	Size	The number of operations in a class.
NumPubOps	class	Size	The number of public operations in a class.
Setters	class	Size	The number of operations with a name starting with 'set'.
Getters	class	Size	The number of operations with a name starting with 'get', 'is', or 'has'.
Nesting	class		The nesting level of the class (for inner classes).
IFImpl	class	Inheritance	The number of interfaces the class implements.
NOC	class	Inheritance	The number of children of the class (UML Generalization).
NumDesc	class	Inheritance	The number of descendants of the class (UML Generalization).
NumAnc	class	Inheritance	The number of ancestors of the class.
DIT	class	Inheritance	The depth of the class in the inheritance hierarchy.
CLD	class	Inheritance	Class to leaf depth.
OpsInh	class	Inheritance	The number of inherited operations.
AttrInh	class	Inheritance	The number of inherited attributes.
Dep_Out	class	Coupling (import)	The number of elements on which this class depends.
Dep_In	class	Coupling (export)	The number of elements that depend on this class.
NumAssEl_ssc	class	Coupling	The number of associated elements in the same scope (namespace) as the class.
NumAssEl_sb	class	Coupling	The number of associated elements in the same scope branch as the class.
NumAssEl_nsb	class	Coupling	The number of associated elements not in the same scope branch as the class.
EC_Attr	class	Coupling (export)	The number of times the class is externally used as attribute type.
IC_Attr	class	Coupling (import)	The number of attributes in the class having another class or interface as their type.
EC_Par	class	Coupling (export)	The number of times the class is externally used as parameter type.
IC_Par	class	Coupling (import)	The number of parameters in the class having another class or interface as their type.
Connectors	class	Complexity	The number of connectors owned by the class.
InstSpec	class		The number of instance specification where the class is a classifier.
LLInst	class		The number of lifelines that represent a property of which this class is the type.
MsgSent	class	Coupling (import)	The number of messages sent.
MsgRecv	class	Coupling (export)	The number of messages received.

Name: Assoc Domain: diagram



First, values from different criteria's has been calculated for every independent variables and then avg of these values has been taken as the value of an independent variable.

	A	B	C	D	E
1	Effectiveness	Encapsulation	Inheritance	Coupling	Hierarchies
2	9.53388	0.88	0.7	0.62	1
3	7.18464	0.98	0.38	1.19	2
4	10.70366	0.78	0.51	1.78	1
5	11.11615	0.99	0.72	0.93	1
6	8.1138	1	0.47	1.1	2
7	2.41206	0.66	0.47	4.64	29

Table 2. Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Change Statistics				
					R Square Change	F Change	df1	df2	Sig.
1	.999 <sup>a</sup>	.996	.988	.34864	.996	99.540	4	1	.075

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

Table 3. ANOVA<sup>b</sup>

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	48.645	4	12.161	99.540	.075 <sup>a</sup>
	Residual	.122	1	.122		
	Total	48.767	5			

a. Predictors: (Constant), Hierarchies, Inheritance, Encapsulation, Coupling  
 b. Dependent Variable: Effectiveness

Table 1. Coefficients<sup>a</sup>

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-4.079	4.229		0.965	.511
	Encapsulation	4.599	2.966	0.228	1.566	.362
	Inheritance	12.000	1.982	0.492	6.053	.104
	Coupling	2.698	0.738	1.310	3.659	.170
	Hierarchies	-0.507	0.068	-1.827	-7.391	.086

a. Dependent Variable: Effectiveness

Effectiveness Estimation Model Output and Result Summary

The Coefficients part of the output gives us the values that we need in order to write the regression equation (4). The Standardized Beta Coefficients give a measure of the contribution of each variable to the Effectiveness model. A big value designates that a unit change in this predictor variable has a large effect on the criterion variable. The t and Sig (p) values give a rough indication of the impact of each predictor variable – a big absolute T value and small p value suggests that a predictor variable is having a large impact on the criterion variable. The experimental

evaluation of Effectiveness is very encouraging to obtain effectiveness index of software design for low cost testing and maintenance.

Table 4.1: Coefficients for Effectiveness Estimation Model

The descriptive statistics of the output gives the mean, standard deviation, and observation count (N) for each of the dependent and independent variables and is shown in Table 4.2.

Table 4.2: Descriptive Statistics for Effectiveness Estimation Model

	Mean	Std. Deviation	N
Effectiveness	8.1773	3.1646	6
Encapsulation	0.8816	0.1531	6
Inheritance	0.5416	0.1281	6
Coupling	1.7100	1.5144	6
Hierarchies	6.0000	11.2783	6

4.6 Empirical Validation

Empirical validation is a vital phase of proposed research. Empirical validation is the standard approach to justify the model approval. Taking view of this truth, practical

validation of the effectiveness model has been performed using sample tryouts. In order to validate developed effectiveness model the data has been taken from [10].

Table 4.4: Effectiveness Ranking and their Relation

Projects □	Effectiveness Ranking		Σd <sup>2</sup>	r <sub>s</sub>	r <sub>s</sub> > ±.781
	Computed RRARank Rank	Known Actual Rank			
P1	1	5	16	0.903	□
P2	4	10	36	0.781	□
P3	2	6	16	0.903	□
P4	7	4	9	0.9454	□
P5	9	9	0	1.000	□
P6	10	8	4	0.903	□
P7	5	1	16	0.903	□
P8	6	2	16	0.903	□
P9	8	3	25	0.848	□
P10	3	7	16	0.903	□

Speraman’s Coefficient of Correlation *r<sub>s</sub>* was used to check the significance of correlation among calculated values of effectiveness using model and it’s ‘Known Values’. The

‘*r<sub>s</sub>*’ was estimated using the method given as under: Speraman’s Coefficient of Correlation

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

$$-1.0 \leq r_s \leq +1.0$$

‘d’ = difference between ‘Calculated ranking’ and ‘Known ranking’ of effectiveness.

‘n’ = number of projects used in the experiment.

The correlation values between effectiveness through model and known ranking are shown in table (4) above. Pairs of these values with correlation values *r<sub>s</sub>* above [±.781] are checked in table. The correlations are up to standard with high degree of confidence, i.e. up to 99%. Therefore we can conclude without any loss of generality that effectiveness Estimation model measures are really reliable and significant and applicable.

#### 4.7 CONCLUSION

The study has developed model to compute effectiveness, effectiveness of the class diagrams. Effectiveness model measures the effectiveness of class diagrams in terms of their design constructs. Effectiveness model have been developed using the method of multiple linear regressions. The study moreover validates the quantifying ability of effectiveness model. The applied validation on the effectiveness model concludes that proposed model is highly consistent, acceptable and considerable. The values of effectiveness are of instant use in the software development process. These values help software designers to review the design and take proper corrective measures, early in the development cycle, in order to control or at least reduce future maintenance/testing.

#### REFERENCES

[1] Zainab Al-Rahamneh, Mohammad Reyalat, Alaa F. Sheta, Sulieman Bani-Ahmad, Saleh Al-Oqeili,, “A New Software Effectiveness Growth Model: Genetic-Programming-Based Approach”, Journal of Software Engineering and Applications, 2011, Vol: 4, PP:476-481 doi:10.4236/jsea.2011.48054..

[2] Natasha Sharygina , James C. Browne, and Robert P. Kurshan, “A Formal Object-Oriented Analysis for Software Effectiveness: Design for Verification”, 2011, pp:1-15

[3] 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, pages 3086-3090, August 2013.

[4] Nikolaos Tsantalis, Alexander Chatzigeorgiou, “Predicting the Probability of Change in Object-Oriented Systems”, IEEE Transactions on Software Engineering, VOL. 31, NO. 7, July 2012, pp: 601-614.

[5] Stephanie Gaudan, Gilles Motet and Guillaume Auriol, “A new structural complexity metrics applied to Object Oriented design quality assessment”, [http://www.lesia.insa-toulouse.fr/~motet/papers/2007\\_ISSRE\\_GMA.pdf](http://www.lesia.insa-toulouse.fr/~motet/papers/2007_ISSRE_GMA.pdf).

[6] Mahfuzul Huda, Dr.Y.D.S.Arya, and Dr.M. H. Khan. “Measuring Reusability of Object Oriented Design: A Systematic Review.” International Journal of Scientific Engineering and Technology, Vol. 3, Issue 10, pp: 1313-1319 Oct, 2014.

[7] Everaldo E. Mills, “Software Metrics”, SEI Curriculum Module SEI- CM-12-1.1, Software Engineering Institute, Dec 1988, pp: 1-43.

[8] Haifeng Li Minyan Lu Quieting Li, “Software Effectiveness Metrics Selecting Method Based on Analytic Hierarchy Process”, Sixth International Conference on Quality Software, 2006. QSIC 2006. 27-28 Oct. 2006, pp: 337 – 346, ISSN: 1550-6002, ISBN: 0-7695-2718-3.

[9] Offutt, J. and R. Alexander, (2013): A fault Model for Subtype Inheritance and Polymorphism. In 12th International Symposium, Software Effectiveness Engineering, Nov 27-30, 2013, IEEE, pp. 84- 93.

[10] SDMetric.com.

- [11] Abdullah, Dr. M. H. Khan, and Reena Srivastava. "Testability Estimation Model for Object Oriented Design (TMMOOD)." *International Journal of Computer Science & Information Technology (IJCSIT)* Vol. 7, No 1, February 2015, DOI: 10.5121/ijcsit.2015.7115.
- [12] F. Li., T.Yi, "Apply Page Rank Algorithm to Measuring Relationship's Complexity", IEEE, DOI 10.1109/PACIA.2008.309, pp. 914-917, 2008, ISBN: 9780769534909.
- [13] Abdullah, Dr. Reena Srivastava, and M. H. Khan. "Testability Estimation Framework: Design Phase Perspective." *International Journal of Advanced Research in Computer and Communication Engineering* Vol. 3, Issue 11, Pages 8573-8576 November 2014
- [14] Yong Cao Qingxin Zhu. Improved metrics for encapsulation based on information hiding. DOI: 10.1109/ICYCS.2008.76, The 9th International Conference for Young Computer Scientists, IEEE computer society 2008, p: 742-724.
- [15] Sch aril N., Black Andrew P., Ducasse S. Object oriented Encapsulation for Dynamically Typed Languages. OOPSLA 2004, ACM, pp: 130-139.
- [16] Abdullah, Dr. Reena Srivastava, and M. H. Khan. "Modifiability: A Key Factor to Effectiveness ", *International Journal of Advanced Information Science and Technology*, Vol.26, No.26, Pages 62-71 June 2014.
- [17] Usha Chhillar, Shuchita Bhasin, "A New Weighted Composite Complexity Measure for Object-Oriented Systems", *International Journal of Information and Communication Technology Research* Volume 1 No. 3, July 2011, pp: 101-108, ISSN-2223-4985.
- [18] Dromey, R.G.: A Model for Software Product Quality. *IEEE Transaction on Software Engineering* 21(2), 146-162 (1995).
- [19] Abdullah, Dr. M. H. Khan, and Reena Srivastava. "Flexibility: A Key Factor to Testability", *International Journal of Software Engineering & Applications (IJSEA)*, Vol.6, No.1, January 2015. DOI: 10.5121/ijsea.2015.6108.
- [20] Fiondella, L.; Gokhale, S.S., "Software effectiveness model with bathtub- shaped fault detection rate" *Effectiveness and Maintainability Symposium (RAMS), 2011 Proceedings - Annual* , 24-27 Jan. 2011, pp: 1 - 6, ISBN: 978-1-4244-8857-5.
- [21] Mahfuzul Huda, Dr.Y.D.S.Arya, and Dr. M. H. Khan. "Quality Quantification Framework of Object Oriented Software: A New Perspective." *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 4, Issue 1, Jan, 2015, DOI: 10.17148/IJARCC.2015.4168.
- [22] Mohan, K.K.; Verma, A.K.; Srividya, A., "Software effectiveness estimation through black box and white box testing at prototype level ", 2nd International Conference on Effectiveness, Safety and Hazard (ICRESH), 14-16 Dec. 2010, pp: 517 - 522, ISBN: 978-1-4244-8344-0.
- [23] J.H. Hayes and L Zhao, "Effectiveness 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.
- [24] R. Pressman, "Software Engg: A Practitioner's Approach", Sixth Ed. Mcgraw- Hill, 2005.
- [25] H. Zuse, "A Framework of Software Estimation" Walter de Gruyter, 1998.
- [26] ISO/IEC 9126-4:2004, "Software Engg. Product Quality- Quality in Use Metrics", ISO/IEC 2004.
- [27] 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.
- [28] Illingworth, V., (Ed.) (1983). *Dictionary of Computing*. Oxford, Oxford University Press.
- [29] Sommerville, I. (1992). *Software Engineering*. 4th ed. New York, Addison- Wesley.
- [30] McClure, C. (1992). *The three Rs of software automation: re- engineering, repository, reuseability* New Jersey, Prentice-Hall.
- [31] M. Dagpinar and J. Jahnke, "Predicting Effectiveness with Object- Oriented Metrics – an Empirical Comparison," *Proc. 5th Working Conference on Reverse Engineering (WCRE'03)*, 13 - 17 Nov. 2003, pp. 155 - 164, 2003.
- [32] 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.
- [33] <http://msdn.microsoft.com/en-us/library/ee658094.aspx>
- [34] IEEE Std. 65.12-1990. *Standard Glossary of Software Engineering Terminology*, IEEE Computer Society Press, Los Alamitos, CA, 1993.
- [35] 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
- [36] S. Muthanna, K. Kontogiannis, K. Ponnambalam and B. Stacey, "A Effectiveness Model for Industrial Software Systems Using Design Level Metrics", In *Working Conference on Reverse Engineering (WCRE'00)*, 2000
- [37] Hayes, J. Huffman, Mohamed, N., Gao, T. The Observe-Mine-Adopt Model: An agile way to enhance software effectiveness. *Journal of Software Maintenance and Evolution: Research and Practice*, Volume 15, Issue 5, Pages 297 – 323, October 2003.
- [38] G. DiLucca, A. Fasolino, P. Tramontana, and C. Visaggio. Towards the definition of an effectiveness model for web applications. In *Proceeding of the 8th European Conference on Software Maintenance and Reengineering*, pages 279–287. IEEE Computer Society Press, 2004.
- [39] Hayes J.H. and Zaho L (2005), "Effectiveness Prediction a Regression Analysis of Measures of Evolving Systems", *Proc.21st IEEE International Conference on Software Maintenance*, 26-29 Sept.2005, pp.601-604.
- [40] C.V. Kotten, A.R. Gray, "An application of Bayesian network for predicting o object-oriented software effectiveness", *Information and Software Technology Journal*, vol: 48, no: 1, pp 59-67, Jan2006.
- [41] Y. Zhou and H. Leung, "Predicting object-oriented software effectiveness using multivariate adaptive regression splines" , *Journal of Systems and Software*, vol. 80, no. 8, pp. 1349-1361, 2007
- [42] MO. Elish and KO. Elish, "Application of TreeNet in Predicting Object-Oriented Software Effectiveness: A Comparative Study", *European Conference on Software Maintenance and Reengineering*, pp 1534-5351, March 2009, DOI 5.159/CSMR.2009.57
- [43] Alisara Hincheeranan and Wanchai Rivepipoon, "A usability Estimation Model and Tool." *International Journal of Computer and Communication Engineering*, Vol. 1, No. 2, July 2012.
- [44] 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

- [45] Johny Antony P & Harsh Dev , Estimating Effectiveness Of Software System Using Object-Oriented Metrics, International Journal of Computer Science Engineering and Information Technology Research (IJCSSEITR) ISSN 2249-6831 Vol. 3, Issue 2, Jun 2013, 283-294.
- [46] P. Oman and J. Hagemester, "Metrics for assessing a software system's effectiveness," *Software Maintenance*, 1992, pp. 337 - 344.
- [47] 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.
- [48] M. Kiewkanya, N. Jindasawat, *et al.*, "A methodology for constructing effectiveness model of object-oriented design," *Quality Software QSIC 2004 Proceedings Fourth International*, 2004, pp. 206 - 213.
- [49] 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.
- [50] 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.
- [51] 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.
- [52] 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.
- [53] S. Ghosh, S. K. Dubey, *et al.*, "Comparative Study of the Factors that Affect reliability," *International Journal on Computer Science and Engineering*, vol. 3, no. 12, pp. 3763-3769, 2011.
- [54] C. Gautam and S. S. Kang, "Comparison and implementation of compound memood model and memood model," *International Journal of Computer Science and Information Technologies* vol. 2, no. 5, pp. 2394-2398, 2011.
- [55] GPL (2011). StarUML. [Online]. Available: <http://staruml.sourceforge.net/en/index.php>.
- [56] Sparxsystems. (2012). XMI Import and Export. [Online]. Available: [http://www.sparxsystems.com/enterprise\\_architect\\_user\\_gui\\_de/projects\\_and\\_teams/importexport.html](http://www.sparxsystems.com/enterprise_architect_user_gui_de/projects_and_teams/importexport.html).
- [57] 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.
- [58] P. Antonellis, D. Antoniou, Y. Kanellopoulos, C. Makris, E. Theodoridis, C. Tjortjis, and N. Tsirakis, "A Data Mining Methodology for Evaluating Effectiveness According to ISO/IEC- 9126 Software Engineering Product Quality Standard," *Proc. 11th IEEE Conference on Software Maintenance and Reengineering* , 21 - 23 Mar.2011, Amsterdam, Netherlands, 2011.