ISSN 2229-5518

An Empirical Study on Testability Measurement of Object Oriented Software

M.A. Khanum, Arun Mani Tripathi[2]
Associate Professor, Department of Computer Application , Integral University, Lucknow,India [1]
M.tech student, C.S.E., Integral University, Lucknow, [2]

Abstract: Testability has always been a difficult exercise and its accurate estimation or assessment a complex exercise. Most of the studies quantify testability or more specifically the attributes that have influence on software testability but at the implementation level. A decision to transform the design in order to improve software testability after implementation has started may be very error prone and expensive. While estimating testability early in the software development life cycle may significantly reduce the overall development cost. In this paper Testability has been recognized as a major factor to object oriented software quality and importance is being drawn to measure design testability, near the beginning in the software development process. Study identified effectiveness and flexibility are the major factors that affect overall testability at design phase. The identified testability factors are to be correlated with the object oriented design characteristics. After successfully establishing the relationship, study developed Testability Measurement Model for Object Oriented Design. Subsequently testability model has been empirically validated and contextual interpretation has been performed using try out software projects.

Keywords: Testability, Flexibility, Effectiveness, Testability Model

I. Introduction

Software systems are fastest emerging trend in the real world and the probable possibilities concerning the functions and features provided by a precise application is generating remarkable interest amongst a huge number of users around the world [1-3]. As the importance grows, so does the request for software application. In view of the fact that development of large software system involves a number of activities which are need to be properly coordinated to fulfill expected requirements [4][5]. Therefore in current scenario the importance for developing good quality software is not longer an advantage but an essential factor. Software testing is one the most fundamental method to correct the system faults and also help in improving the software quality directly [6].

Assessment of software product quality throughout the development life cycle is very important to manage and improve the software quality [7-10]. Assessment of software quality after the completion of development work is no longer an advantage but it is more important to monitor and manage the quality of software when it is under development [11][12]. Thus, there is a need for identifying the design properties, which may be used in early stage of development to give good indication of software quality [13]. In order to provide the significant assessment of software product quality, it is important to identify a set of high-level desirable quality attributes, and to find a way to relate the design properties to quality attributes, significantly. In this regard an effort has been made in this paper.

II. Software testability

The common definition of testability is ease of performing testing. Object oriented design testability is an external quality attribute that estimates the complexity and the required effort for software testing. Software testability is a main aspect to permit the detection of difficulties to uncover defects in software. The IEEE Standard Glossary defines testability as the degree to which a system or component facilitates the establishment of test criteria and performance of tests to determine whether those criteria have been met [14][16].Software testability is a non-functional requirements and important to the testing team member, who are involved in user acceptance testing. Non functional requirements are generally a quality requirement and may make the user more satisfied and pleased[15]. Testability is one of the significant concept in software design and testing software program and its components. Creating programs and components with greater level testability constantly simplify the test process; reduce cost, and increases software quality [17].

III. Factor Identification

Software testability is a major factor to software quality. In order to estimate testability, its direct measures are to be recognized [18][19]. In this study, the commonly accepted group of factors for testability are identified. Object oriented design level factors will also be examined keeping in view their overall impact on the software testability. Most of the mechanisms available for testability estimation of object oriented software, may normally be used in later phases of system development life cycle and rely upon information extracted on the operationalization of software [20]. Such methods provide an indication of testability and hence quality, but too late to improve the product, prior to its completion. The characteristics of testable software like adequate complexity, low coupling and good separation of concerns make it easier for reviewers to understand the software artifacts under review [21][23]. Plenty of work has been carried out in describing the need and importance of incorporating software testability since early 90s. A number of methods of measuring testability have been proposed[24]. Unfortunately, the significant achievements made by the researchers in the area have not been widely accepted and are not adopted in practice by industry [22]. It has been found that there is a conflict in considering the factors while estimating software testability in general and at design level exclusively.

IV. Software Characterization

In this section study will be identified different design characteristics that have their impacts on software testability measurement and product quality as well. The involvement of each properties help to improve the software design will also be analyzed. The consolidated charting for the object oriented design properties considered by different researchers is shown in following table. This study shows that abstraction is least important factor in while cohesion, coupling, encapsulation and inheritance play key role in testability measurement.

TABLE I
OBJECT ORIENTED DESIGN CONSTRUCTS CONTRIBUTING IN TESTABILITY MEASUREMENT A CRITICAL LOOK

Design Parameters / **Author/Study**	Cohesion	Coupling	Encapsulation	Inheritance	Abstraction
MC Gregor et al. (1996)			√	√	
Bruce & Shi(1998)		√		√	
B.Pettichord(2002)		√			
Baidry et al.(2002)		√			
M Bruntik (2004)				√	
S.Mouchawrab (2005)	√	√		√	
I.Ahson et al.(2007)	√	√	√	√	
Nazir et al.(2005)	√	√	√	√	
Lee et al.((2012)	√	√	√	√	√
Khan et al. (2012)	√	√	√	√	
Nikfard & Babak (2013)		√	√	√	

V. Correlation Establishment

Correlation establishment is an important step between testability and its major identified factors. In this step the recognized software testability factors are to be correlated with the Object Oriented design properties. A regression line will be established to quantify testability factors in terms of design characteristics with the help of design metrics. Here ENM shows encapsulation metrics, INM shows inheritance metrics, CPM shows cohesion metrics and COM shows coupling metrics value in following figure.

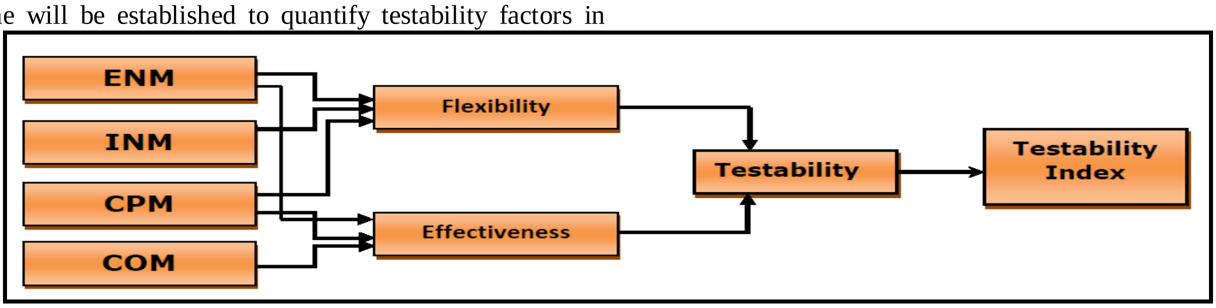

Fig. 1. Correlation Establishment

VI. Model Development

Quantification of design diagram Effectiveness and Functionality is precondition for the testability estimation model. For that reason before developing testability model, the research paper has developed 2 models for Effectiveness and Functionality. In order to develop all the models given below multivariate linear model is applied.

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

VI .A) Flexibility Estimation Model

Flexibility is strongly related to testability and regularly plays a key role to deliver high class, best quality testable software within time and given budget [6] [19]. It is one of the most important notions in design for testing of software programs and components [11][12].

$$\textbf{Flexibility} = -2.004 + .029 \times \text{Coupling} + 14.091 \times \text{Inheritance} + 3.964 \times \text{Encapsulation} \quad (2)$$

TABLE III **Coefficients**[a]

Model		Unstandardized Coefficients B	Unstandardized Coefficients Std. Error	Standardized Beta	t	Sig.
1	(Constant)	-2.004	3.836		-.522	.654
	Coupling	.029	.231	.039	.126	.911
	Inheritance	14.091	4.678	1.180	3.012	.095
	Encapsulation	3.964	3.174	.492	1.249	.338

a. Dependent Variable: Flexibility

TABLE IIIII **Model Summary**

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Change Statistics R Square	F	df1	df2	Sig. F
1	.920[a]	.847	.618	.77456	.847	43.699	3	2	.220

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

TABLE IVV **ANOVA**[b]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	6.657	3	2.219	43.699	.220[a]
	Residual	1.200	2	.600		
	Total	7.857	5			

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

b. Dependent Variable: Flexibility

VI. B) Effectiveness Estimation Model

Effectiveness is strongly related to testability and regularly plays a key role to deliver high class, best quality testable software within time and given budget [13] [15]. It is one of the most important notions in design for testing of software programs and components [11][17].

$$\textbf{Effectiveness} = -9.440 - 4.161 \times \text{Encapsulation} + 1.506 \times \text{Coupling} + 26.392 \times \text{Cohesion} \quad (3)$$

TABLE V **Coefficients**[a]

Model		Unstandardized B	Unstandardized Std. Error	Standardized Beta	t	Sig.
1	(Constant)	-9.440	9.297		-1.015	.417
	Encapsulation	-4.161	16.480	-.134	-.252	.824
	Coupling	1.506	.671	1.787	2.245	.154
	Cohesion	26.392	26.581	1.036	.993	.425

a. Dependent Variable: Effectiveness

TABLE VI **Model Summary**

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Change Statistics				
					R Square	F Change	df1	df2	Sig. F
1	.946[a]	.895	.738	1.22687	.895	35.696	3	2	.153

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

TABLE VII **ANOVA**[b]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	25.720	3	8.573	35.696	.153[a]
	Residual	3.010	2	1.505		
	Total	28.731	5			

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

b. Dependent Variable: Effectiveness

VII. TESTABILITY ESTIMATION MODEL

The generic quality models [12] [16][19] have been considered as a basis to develop the Testability Model.Before developing the model for Testability, it is important to make sure the appropriate association among Testability, Flexibility and Effectiveness of class diagrams. Table below, shows the relationship values among them. From the correlation values it is clear that both Flexibility and Effectiveness are strongly correlated with Testability.

TABLE VIII **Correlations**

	Testability	Flexibility	Effectiveness
Testability	1.000	.985	.992
Flexibility	.985	1.000	.961
Effectiveness	.992	.961	1.000

TABLE IX **Coefficients**[a]

Model		Unstandardized Coefficients		Standardized	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	4.227	.173		24.366	.002
	Flexibility	.260	.071	.419	3.657	.067
	Effectiveness	.534	.104	.589	5.143	.036

a. Dependent Variable: testability

$$\textbf{Testability} = 4.227 + .260 \times \textbf{Flexibility} + .534 \times \textbf{Effectiveness} \quad (4)$$

TABLE X **Model Summary**

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.999[a]	.998	.996	.09744

a. Predictors: (Constant), Effectiveness, Flexibility

TABLE XI **ANOVA**[b]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	9.338	2	4.669	491.812	.002[a]

	Residual	.019	2	.009		
	Total	9.357	4			
a. Predictors: (Constant), Effectiveness, Flexibility						
b. Dependent Variable: testability						

VIII. Empirical Validation of Developed Model

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 Testability model has been performed using sample tryouts. In order to validate developed Testability model the data has been taken from [11].

TABLE XII : Computed Ranking, Actual Ranking and their Relation

Projects ↓	Testability Ranking		$\sum d^2$	r_s	$r_s > \pm.781$
	Computed Rank	[illegible]			
P1	1	6	25	0.85	✓
P2	5	7	4	0.98	✓
P3	7	9	4	0.98	✓
P4	9	10	1	0.99	✓
P5	8	8	0	1.00	✓
P6	2	2	0	1.00	✓
P7	4	3	1	0.99	✓
P8	6	4	4	0.98	✓
P9	10	5	25	0.85	✓
P10	3	1	4	0.98	✓

- **r_s above±.781** means significant results.

Speraman's Coefficient of Correlation r_s was used to check the significance of correlation among calculated values of testability using model and it's 'Known Values'. The 'r_s' was estimated using the method given as under: Speraman's Coefficient[1] of Correlation

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

'd' = difference between 'Calculated ranking' and 'Known ranking' of testability.

'n' = number of projects (n=10) used in the experiment.

The correlation values between testability through model and known ranking are shown in table XII. Pairs of these values with correlation values r_s 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 testability Estimation model measures are really reliable and significant and applicable.

IX. CONCLUSION

The model developed in the paper will address testability at design stage of software development life cycle. It may also help to positioning testability benchmarking of software projects. From the correlation values it is clear that both Flexibility and Effectiveness are strongly correlated with testability. It plays a significant role as far as the issues of delivering quality software are concerned. Therefore, the software testability model has been validated theoretically as well as mathematically using experimental try-out. Outcome result shows that the values of software testability computed through model are highly correlated with the 'known values'. The applied validation on the testability estimation model concludes that developed model is highly consistent, acceptable and reliable. The model is developed, may be used by software industry practitioners and quality controllers to measure testability in order to make good design decisions near the beginning in the software development life cycle.

References

[1] IEEE Press, "IEEE Standard Glossary of Software Engineering Technology,"ANSI/IEEE Standard 610.12-1990, 1990.

[2] ISO, "International standard ISO/IEC 9126. Information technology: Software product evaluation: Quality characteristics and guidelines for their use." 1991.

[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] Zhao, L.: A New Approach for Software Testability Analysis. In: Proceeding of the 28th International Conference on Software Engineering, Shanghai, pp. 985–988 (2006)

[5] Huda, M., Arya, Y.D.S. and Khan, M.H. (2014) Measuring Testability of Object Oriented Design: A Systematic Review. *International Journal of Scientific Engineering and Technology* (*IJSET*), **3**, 1313-1319.

[6] Abdullah, Dr, Reena Srivastava, and M. H. Khan. "Testability Measurement Framework: Design Phase Perspective."International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 11, Pages 8573-8576 November 2014.

[7] Dromey, R.G.: A Model for Software Product Quality. IEEE Transaction on Software Engineering 21(2), 146–162 (1995).

[8] Huda, M., Arya, Y.D.S. and Khan, M.H. (2015) Quantifying Reusability of Object Oriented Design: A Testability Perspective. *Journal of Software Engineering and Applications*, **8**, 175-183. http://dx.doi.org/10.4236/jsea.2015.84018

[9] Abdullah, Dr, Reena Srivastava, and M. H. Khan. “Modifiability: A Key Factor To Testability”, International Journal of Advanced Information Science and Technology, Vol. 26, No.26, Pages 62-71 June 2014.

[10] Gao, J., Shih, M.-C.: A Component Testability Model for Verification and Measurement. In: Proc. of the 29th Annual International Computer Software and Applications Conference, pp. 211–218. IEEE Comp. Society (2005)

[11] Bansiya, Jagdish, and Carl G. Davis. "A hierarchical model for object-oriented design quality assessment." Software Engineering, IEEE Transactions on 28.1 (2002): 4-17.

[12] Abdullah, Dr, M. H. Khan, and Reena Srivastava. “Testability Measurement Model for Object Oriented Design (TMM^{OOD}).” International Journal of Computer Science & Information Technology (IJCSIT) Vol. 7, No 1, February 2015, DOI: 10.5121/ijcsit.2015.7115.

[13] Huda, M., Arya, Y.D.S. and Khan, M.H. (2015) Testability Quantification Framework of Object Oriented Software: A New Perspective. *International Journal of Advanced Research in Computer and Communication Engineering*,**4**,298-302. http://dx.doi.org/10.17148/IJARCCE.2015.4168

[14] Fu, J.P. and Lu, M.Y. (2009) Request-Oriented Method of Software Testability Measurement. *Proceedings of the ITCS* 2009 *International Conference on Information Technology and Computer Science*, Kiev, 25-26 July 2009, 77-80.

[15] Huda, M., Arya, Y.D.S. and Khan, M.H. (2015) Evaluating Effectiveness Factor of Object Oriented Design: A Testability Perspective. *International Journal of Software Engineering & Applications*(*IJSEA*),**6**,4149. http://dx.doi.org/10.5121/ijsea.2015.6104

[16] Lee, Ming-Chang. "Software Quality Factors and Software Quality Metrics to Enhance Software Quality Assurance." British Journal of Applied Science & Technology 4.21 (2014).

[17] Huda, M., Arya, Y.D.S. and Khan, M.H. (2015) Metric Based Testability Estimation Model for Object Oriented Design: Quality Perspective. *Journal of Software Engineering and Applications*, **8**, 234-243. http://dx.doi.org/10.4236/jsea.2015.84024

[18] Badri, M. and Toure, F. (2012) Empirical Analysis of Object-Oriented Design Metrics for Predicting Unit Testing Effort of Classes. *Journal of Software Engineering and Applications*, **5**, 513-526.http://dx.doi.org/10.4236/jsea.2012.57060

[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] Mouchawrab, S., Briand, L.C. and Labiche, Y. (2005) A Measurement Framework for Object-Oriented Software Testability. *Information and Software Technology*, **47**, 979-997. http://dx.doi.org/10.1016/j.infsof.2005.09.003

[21] Jungmayr, S. (2002) Testability during Design, Softwaretechnik-Trends. *Proceedings of the GI Working Group Test, Analysis and Verification of Software*, Potsdam, 20-21 June 2002, 10-11.

[22] Bruntink, M. and Van Deursen, A. (2004) Predicting Class Testability Using Object-Oriented Metrics. *Proceedings of the Fourth IEEE International Workshop on Source Code Analysis and Manipulation*, Chicago, 15-16 September 2004,136-145.

[23] Amin, A. and Moradi, S. (2013) A Hybrid Evaluation Framework of CMM and COBIT for Improving the Software Development Quality.

[24] Zheng, W.Q. and Bundell, G. (2008) Contract-Based Software Component Testing with UML Models. *International Symposium on Computer Science and Its Applications* (*CSA* ’08), 978-0-7695, 13-15 October 2008, 83-102.