ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

Testability Estimation of Object Oriented Software

Shweta Sivastava[1], Dr. Majhar Khaliq[2]

M.Tech. Scholar, Department of Computer Science, R.K.G.IT, Ghaziabad, India[1]

Assistant Professor, Department of Computer Science, K.M.C.A.F.U, Lucknow, India[2]

Abstract: Software developers and quality controllers 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. Regardless of the fact that software testability is very important and one of the most considerable components of the software development life cycle, it is weakly managed. This is mostly due to the lack of its proper supervision and control. The study highlights the importance of testability in general and as a factor of software quality. Two most important contributions are made in the paper. A relation between object oriented testability factors and properties has been established as a major contribution. Finally, a model has been proposed for estimating testability of object oriented software using design metrics.

Keywords: Software Quality, Software Design, Software Testability, Complexity, Effectiveness

I. INTRODUCTION

Our modern society is becoming increasingly dependent on software, due to this the size and complexity of computer software has grown at a very high peak. Dependency and requirements on computer software increases the difficulties and failures of software and the often-devastating effect that a software error can have in terms of life, financial loss, or time delays. So the demand of quality software continues to increase. Software quality is not an advantage but a necessary factor for software industry [1] [3].Software design is the more creative and highly significant phase in software development process. Software design can play the role to control and improve the software quality.

The quality of software design affects the quality of final product. Considering the fact that inefficiencies in software design account for the maximum errors thereby contributing to testing and maintenance costs, it is only wise to isolate the errors as early as possible in the design phase to eliminate ripple costs [4].

Assessment of software product quality throughout the development life cycle is very important to manage and improve the software quality. 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 [4].

Thus, there appeared to be need for identifying the design properties, which may be used in early stage of development to give good indication of software quality. In order to provide the significant assessment of software product quality, it is required 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 a bit effort has been made in this paper.

II. SOFTWARE TESTABILITY

The most common definition of Testability is ease of performing testing. 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 [6]. Software testability is a key aspect to allow the detection of difficult to uncover defects in software. Software testability supports the testing process and facilitates the creation of better quality software. Software testability is affected by many different factors, including the required validity, the process and tools used the representation of the requirements, and so on [7]. One reason is that there are many potential factors that can affect testability. Software testability is an external software attribute that evaluates the complexity and the effort required for software testing [8].

III. TESTABILITY FACTORS

A truthful measure of software quality fully depends on testability measurement, which in turn depends on the factors that have an effect on software testability. Hence, it appears extremely desirable and important to recognize the factors that facilitate testability in order to get the accurate and reliable measure of software testability [9]. Despite the fact that, getting a universally accepted set of testability factors is only probable [10, 11]. Testability quality criteria are the characteristics which help to identify the testability factors. Criteria present a more complete, actual definition of factors as well as criteria common among factors assist to show the interrelationship between factors. The criteria of the testability factor are the characteristics of the software product or development cycle by which the factor can be judged or recognized. An endeavor has been made to collect a set of testability factors that can affect software testability. However,

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

without any loss of generality, it comes into view to include the factors namely, modifiability, simplicity, understandability, flexibility, traceability, complexity, self descriptiveness and modularity as testability factors [12, 103, 14].

IV. DESIGN PROPERTIES

In order to design the software by an object oriented approach, the three fundamental properties are significantly being used i.e. encapsulation, inheritance and polymorphism. Encapsulation is the mechanism to hide the internal specification of an object and shows only the external interface. Information hiding is the process of hiding all the information about the module unless it is specifically declared publicly". Information hiding provides rise to encapsulation in object oriented language [14, 19].Inheritance is an approach where an object acquires the characteristics from another object by sharing of attributes and operations among classes through their hierarchical relationship [15].The new classes of objects that inherit much of their behavior from previously defined classes. Inheritance is a form of reuse that enable a process of development to define objects incrementally by reusing previously defined objects as the basis for new objects [16,18].

Polymorphism is an important concept that has a capability to build a flexible system. Polymorphism means, the ability to have several forms, which is to carry out different processing steps by the operations having same messages. Polymorphism allows the implementation of given operations, which are dependent on the object that contains the operations; an operation can be implemented in different ways in different classes [18, 19].

The two more, most important design properties may be included, that have been generally used in designing of the software that is coupling and Cohesion. Coupling is the process to interact or communicate between two objects by passing messages. It refers to the degree of association from one object to another. It shows the relationship or interdependency between modules.

Coupling may assess the number of collaboration between classes or the number of messages passed between objects .Cohesion is the process to measures the degree of connectivity among the elements of a single class or object [21]. It refers to the degree, to which the no. of method in a class are related to each other. The internal consistency occurs within the parts of the design, and it is focused on data that is encapsulated within an object and how the methods communicate with data to provide well bounded behavior [10].

V. MODELS DEVELOPMENT

In order to create a model for testability, multiple linear regression procedure has been used. In support of this to developing TEM^{OOD}, the study has developed two models for Complexity and Effectiveness.

$$Y=a_0+ a\mathbf{1}\ x1+ a\mathbf{2}\ x2+ a\mathbf{3}x3\ldots\ldots a\mathbf{n}\ xn \quad (1)$$

Where
Y is dependent variable, x1, x2, x3……xn are independent variables
a**1,** a**2, ……a**n are the coefficients and a_0 is the intercept.

VI. COMPLEXITY MODEL

In order to establish a Complexity estimation model for object oriented class diagram, metrics listed in [20] will play the role of independent variables at the same time as Complexity will be taken as dependent variable. The data used for developing Complexity model is taken from [20]. As per the mapping, Metrics 'MFA, DCC, CAM' are selected from [20] as independent variable to develop the Complexity estimation model. Using SPSS, values of coefficient are calculated and Complexity model is originated as below.

Complexity =2.192 -.896× Inheritance -.096× Coupling -.078× Cohesion (2)

TABLE I Coefficients

Model		Unstandardized Coefficients		Standardized Coefficients
		B	Std. Error	Beta
1	(Constant)	2.192	.592	
	Inheritance	-.896	.456	-.804
	Coupling	-.096	.059	-.671
	Cohesion	-.078	.184	-.186
a. Dependent Variable: Complexity				

TABLE II Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.927[a]	.859	.437	.16844
a. Predictors: (Constant), Cohesion, Inheritance, Coupling				

VII. EFFECTIVENESS MODEL

In order to establish an Effectiveness estimation model of object oriented class diagram, metrics listed in [20] will play the role of independent variables at the same time as Effectiveness will be taken as dependent variable. The data used for developing Effectiveness model is taken from [20]. As per the mapping, Metrics 'DAM, MFA, NOP' are selected from [20] as independent variable to develop the Effectiveness estimation model. Using SPSS, values of coefficient are calculated and Effectiveness model is originated as below.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

Effectiveness = .276+ 4.925× Encapsulation + 5.359× Inheritance + .167× Polymorphism (3)

TABLE III Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients
		B	Std. Error	Beta
1	(Constant)	.276	2.155	
	Encapsulation	4.925	1.733	.629
	Inheritance	5.359	1.649	.794
	Polymorphism	.167	.420	.097
a. Dependent Variable: Effectiveness				

TABLE IV Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.950[a]	.903	.756	.71916
a. Predictors: (Constant), Polymorphism, Encapsulation, Inheritance				

In addition the consideration of R2 (Coefficient of Determination) and adjusted R2 in the Table above, is too very positive. As, it refers to the percentage of the whole variance in Extendibility by all the three metrics (independent variables) participating in the model (3).

VIII. TESTABILITY ESTIMATION MODEL

The generic quality models [2] [5] [17] have been taken as a base to develop the Testability Model for Object Oriented Design. In order to set up a model for Testability estimation, a multiple linear regression method has been used to get the coefficients. Applying this method, Complexity Model (2), Effectiveness Model (3) have been developed respectively.

Taking into account the testability major contributors Complexity Model (2), Effectiveness Model (3), develop a model for Testability estimation. The data used for developing model given in equation (4) has been taken from [20].

Testability =3.276+.587× Complexity +.547× Effectiveness (4)

TABLE IV Coefficients[a]

Coefficients[a]				
Model		Unstandardized Coefficients		Standardized Coefficients
		B	Std. Error	Beta
1	(Constant)	3.276	1.188	
	Complexity	.587	1.093	.152
	Effectiveness	.547	.183	.847
a. Dependent Variable: Testability				

TABLE VI Model Summary

Model Summary				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.954[a]	.911	.822	.64585
a. Predictors: (Constant), Effectiveness, Complexity				

In addition the consideration of R2 (Coefficient of Determination) and adjusted R2 in the Table above, is too very positive. As, it refers to the percentage of the whole variance in Extendibility by all the three metrics (independent variables) participating in the model (4).

IX. VALIDATION

Empirical validation is an important phase of proposed research. Empirical validation is the approved approach and practice to justify the model acceptance. Observance view of this truth, practical validation of the testability Estimation model has been performed using sample tryouts. In order to validate developed testability Estimation model the data has been taken from [20].During experiments, testability value of the projects has been calculated using the developed model, followed by the calculation of testability rating.

TableVII Computed Ranking, Actual Ranking and their Relation

Projects ↓	Testability Ranking		$\sum d^2$	r_s	$r_s >$
	Computed Rank	Known Rank			
p1	1	6	25	0.85	✓
p2	7	7	0	1.00	✓
p3	8	9	1	0.99	✓
p4	9	10	1	0.99	✓
p5	10	8	4	0.98	✓
p6	3	2	1	0.99	✓
p7	2	3	1	0.99	✓
p8	5	4	1	0.99	✓
p9	4	5	1	0.99	✓
p10	6	1	25	0.85	✓

Charles Speraman's Coefficient of Correlation (rank relation) r_s was used to check the significance of correlation between calculated rank of testability using model and it's 'Expert's rank'[2]. Rank correlation is the process of determining the degree of correlation between

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

two variables. The 'r_s' was calculated using the method given as under: Speraman's Coefficient 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=28) used in the experiment.

The correlation values between testability through model and known ranking are shown in table (VII) above. 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% in eight cases out of ten. Therefore we can conclude without any loss of generality that testability Estimation model measures are really reliable and significant and applicable.

X. CONCLUSION

Software testability is very important and one of the most considerable components of the software development. The lack of testability aspect often leads to false analysis that may in turn lead to ambiguities, misunderstanding and hence to faulty development results. Form the correlation values it is clear that both Complexity and Effectiveness are strongly correlated with testability. It plays an important role as far as the issue of delivering quality software is concerned. Therefore the testability model has been validated theoretically as well as empirically using experimental try-out. Result shows that the values of 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.

REFERENCES

[1]. Binder, Robert V. "Design for testability in object-oriented systems." Communications of the ACM 37.9 (1994): 87-101.
[2]. 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
[3]. Poston, Robin, Jignya Patel, and Jasbir Dhaliwal. "A software testing assessment to manage project testability." (2012).
[4]. ISO, "International standard ISO/IEC 9126. Information technology: Software product evaluation: Quality characteristics and guidelines for their use." 1991.
[5]. 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
[6]. IEEE Press, "IEEE Standard Glossary of Software Engineering Technology,"ANSI/IEEE Standard 610.12-1990, 1990.
[7]. Mahfuzul Huda, Dr.Y.D.S.Arya, and Dr. M. H. Khan. "Evaluating Effectiveness Factor of Object Oriented Design: A Testability Perspective." International Journal of Software Engineering & Applications (IJSEA), Vol.6, No.1, January 2015, DOI:10.5121/ijsea.2015.6104
[8]. J Voas and Miller , "Improving the software development process using testability research", Proceedings of the 3rd international symposium on software Reliability Engineering , p. 114--121, October, 1992, RTP, NC, Publisher: IEEE Computer Society.
[9]. Lee, Ming-Chang. "Software Quality Factors and Software Quality Metrics to Enhance Software Quality Assurance." British Journal of Applied Science & Technology 4.21 (2014).
[10]. 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.
[11]. Mahfuzul Huda, Dr.Y.D.S.Arya, and Dr. M. H. Khan. "Testability 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/IJARCCE.2015.4168
[12]. Jianping, Fu, Liu Bin, and Lu Minyan. "Present and future of software testability analysis."International Conference on Computer Application and System Modeling (ICCASM 2010). Vol. 15.2010.
[13]. 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
[14]. Jungmayr S.: Testability during Design, Softwaretechnik-Trends, Proceedings of the GI Working Group Test, Analysis and Verification of Software, pp. 10-11, Potsdam. (2002).
[15]. James, Bach.: Heuristics of software Testability (1999)
[16]. 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
[17]. Dromey, R.G.: A Model for Software Product Quality. IEEE Transaction on Software Engineering 21(2), 146–162 (1995).
[18]. J. Gao, S. Ming-Chih, "A Component Testability Model for Verification and Measurement", In a Proceedings of the 29th Annual International Computer Software and Applications Conference, pages 211–218. IEEE Computer Society (2005).
[19]. Mahfuzul Huda, Dr.Y.D.S.Arya, and Dr.M. H. Khan. "Measuring Testability of Object Oriented Design: A Systematic Review." International Journal of Scientific Engineering and Technology, Vol. 3, Issue 10, pp: 1313-1319 Oct, 2014
[20]. Jagdish Bansiya, "A Hierarchical Model for Object Oriented Design Quality Assessment", IEEE Transaction of Software Engineering, Volume 28, No. 1, January 2002, and pp: 4-17
[21]. K.K. Aggarwal, Yogesh Singh. New Age International, Jan 1, 2005 - Software engineering.