Volume 3, Issue 10, October 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

An Empirical Validation of Testability Estimation Model

Mohd Nazir*, **Mustafa K.**
JMI, New Delhi, India

Raees A. Khan
DIT, BBAU, Lucknow, India

Abstract— This paper is the extension of our paper that talks about testability estimation model [12]. In fact, practical usefulness and the greater acceptability of any new measure cannot be established without its empirical validation. The paper presents empirical validation of the model, which is carried out using sample tryouts of industrial projects. Moreover, it briefly describes the utility and significance of the model.

Keywords— Software Quality, Testability Quantification, Object Oriented Design, Understandability, Complexity.

I. Introduction

Software non-performance and failures are found to be very expensive. Errors in the software product are so common and so detrimental that they may even cost to the national economies. Hence, therefore, delivering quality software is an inevitable aspect in today's world. Unfortunately, software industry lack to deliver the quality product and even some time important quality attributes are neglected by the practitioners. Several researchers and practitioner argue that highly testable system enhances reliability and quality of the system. Testability is realized as one of the important indicators of quality. It gives the insights that are valuable during design, coding, testing and also for quality assurance. Moreover, its quantification can be vital for stakeholders and the development process. The significance of testability concept is twofold - first, it can guide the tester in deciding where to focus during testing; secondly, knowledge about what makes some programs more testable than others that can guide the developer to incorporate design-for-test features to a larger extent. Quantitative data on testability may be of immediate use in the software development process. The manager can use data/information to plan and monitor testing activities while tester may use it to determine on what code to focus. Software developer may use testability metrics to review the code.

Testability is still an elusive concept; its measurement has always been a difficult exercise [4]. Moreover, it is hard and often a complicated task to conspicuously visualize the potential factors that may affect testability and their dominant degree under different contexts [6]. There is little consensus among researchers and practitioners on 'what aspects are actually related to software testability'. Hence, it appears quite conclusive from the available literature [5][7-11] that there is a conflict among practitioners in considering the factors while estimating testability in general and exclusively at design level. Thus, commonly accepted set of testability factors is proposed; considering these as the major contributors, testability estimation model is developed [12]. The paper is organized as follows. Section 1 highlights the significance of testability quantification. Section 2 briefly describes the model-TEM^{OOD}. In section 3, authors' claims are validated. Section 4 presents the empirical validation of the model, followed by its significance. Finally, section 5 concludes the paper.

II. Model Description

In most of the studies, Researchers examine the impacts OO software characteristics and successfully established their relationships with quality factors. In this paper, we examine and assessed their impact on the particular aspect/attribute i.e. testability and by associatively and congruence perspective, concluded on identifying testability factors affected by object characteristics. It is observed that each of these characteristics either has positive or negative impact on testability factors. Understandability model [1] and Complexity model [2] forms the strong basis for development of testability quantification model. Relative significance of these is weighted proportionally. In order to establish the model, multiple linear regression technique is used and the model for testability quantification is thus formulated as follows:

$$\text{Testability} = \alpha_0 + \beta_1 * \text{Understandability} + \beta_2 * \text{Complexity} \qquad (1)$$

Where β_1, β_2, are the coefficients of respective variables 'Understandability and Complexity' while α_0 is an intercept.
The data used for developing model is taken from [13] which consist of six industrial projects with around 10 to 20 classes. The values of variables 'Understandability and Complexity' are used in MATLAB and using the values of Coupling Metrics (CPM), Cohesion Metrics (COM), and Inheritance Metrics (INM) from the data set to compute the coefficients and thus, testability quantification model is formulated as given below.

$$\text{Testability} = -483.65 + 300.92 * \text{Understandability} - .86 * \text{Complexity} \qquad (2)$$

III. Validating the Claim

The applications [3] are used for validating testability quantification model in Eq. (2). We labelled the applications as System A, System B, System C and System D. All the systems are commercial software implemented in C++ and the number of classes associated with them is 6, 5, 6 and 10 respectively. The correlations summary of between testability factors and the testability is given in the table I below.

Table I
Correlation Analysis Summary

	Testability X Understandability	Testability X Complexity
System A	.71	.99
System B	.94	.99
System C	.95	.99
System D	.81	.84

Results in the summary show that for all the System, both factors 'Understandability and Complexity' are highly correlated with Testability. It strongly indicates the higher significance of considering both the factors for making a prediction of testability in design phase. In order to further justify the claim, Hypothesis testing is performed to test the significance of r (Correlation Coefficient) by using the following formula:

$$t_r = \frac{r\sqrt{N-2}}{\sqrt{1-r^2}}$$

With *N-2* degree of freedom, a coefficient of correlation is judged as statistically significant when the *t* value equals or exceeds the *t* critical value in the *t* distribution table.

$H_{0(T^U)}$: Testability and Understandability are not highly correlated.

$H_{1(T^U)}$: Testability and Understandability are highly correlated.

Table II
Correlation Coefficient Test for Testability and Understandability

	System A	System B	System C	System D
Testability ^ Understandability	.71	.94	.95	.81
t_r	2.01	4.77	6.08	3.90
$t_{r\text{-Critical Value}}$	2.447	2.571	2.447	2.228
$t_r > t_{r\text{-Critical Value}}$	X	√	√	√
$H_{0(T^U)}$	Accept	Reject	Reject	Reject

$H_{0(T^C)}$: Testability and Complexity are not highly correlated.

$H_{1(T^C)}$: Testability and Complexity are highly correlated.

Table III
Correlation Coefficient Test for Testability and Complexity

	System A	System B	System C	System D
Testability ^ Complexity	.99	.99	.99	.84
t_r	14.03	12.15	14.03	4.37
$t_{r\text{-Critical Value}}$	2.447	2.571	2.447	2.228
$t_r>2.44$	√	√	√	√
$H_{0(T^C)}$	Reject	Reject	Reject	Reject

Using two-tailed test at the .05 level with different degrees of freedom, it is evident from the tables II and III that null hypothesis is rejected (except for System A of Testability and Understandability). Hence, therefore, author's claim of correlating Testability with Understandability and Complexity at design phase is statistically justified.

IV. Empirical Validation

This section assesses how well the model-TEM^{OOD} is able to predict 'Testability', hence the quality of an object oriented software design. The internal characteristic of a design varies significantly with scope or boundary of domain and its context. These characteristics positively or negatively influence these factors of testability and hence, testability and maintainability. As a result, the overall quality gets affected. Testability estimated by the model-TEM^{OOD} is validated using tryout data, followed by statistical analysis and interpretation. In order to validate TEM^{OOD} , data set [3] is used; the known testability rating for the given projects (P_1-P_{10}) is shown in table IV.

Table IV
Given Testability Rating

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
Testability	5	9	1	4	8	2	6	10	3	7

Using the same set of data, the Understandability model [1] and Complexity Model [2] are implemented in order to compute Testability of each of the projects using developed model-TEM^{OOD} [12]. Computed testability using the model is then ranked on the basis of the results that is given in table V.

Table V
Computed Testability Rating

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
Testability Rating	7	10	1	2	8	3	9	6	4	5

Table VI summarizes the Known Testability Ranking and Computed Testability Ranking using the model-TEM^{OOD} and their rank correlations. Speraman's Rank Correlation coefficient r_s is used to test the significance of correlation between calculated values of testability using the model-TEM^{OOD} and known testability ranking. r_s is computed using the formula given as under:

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

Where'd' is the difference between the ranks of calculated values of testability using model and the known value, and n is the number of Projects (n=10) used in the experiment.

Table VI: Computed Ranking, Known Ranking and their Correlations

Projects → Testability Ranking ↓	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
Computed Ranking	7	10	1	2	8	3	9	6	4	5
Known Ranking	5	9	1	4	8	2	6	10	3	7
$\sum d^2$	4	1	0	4	0	1	9	16	1	4
r_s	.97	.99	1	.97	1	.99	.94	.89	.99	.97
r_s >.781	√	√	√	√	√	√	√	√	√	√

The correlation values between Testability using the model-TEM^{OOD} and known value are shown in the table above. Pairs of these values with correlation values r_s above [±.781] show that correlation is acceptable with high degree of confidence, that is at the 99%. It is obvious from the values in the table that testability of all the Projects computed using proposed model are highly correlated with the known value. In order to further assure the validity of the model, a $\chi 2$-test is also performed to reject the null hypothesis, against given qualitative testability ranking of the projects as shown in table VII, stated as follows:

H_0=Testability estimate obtained using the model-TEM^{OOD} is not comparable to the given estimate.

Table VII: Qualitative Testability Ranking

Project	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
Testability	High	High	High	High	High	High	High	High	High	High

The Chi-Square Test observations for all the ten systems are listed in table VIII by using the formula given below that may be applicable for small samples, as frequencies of cells are fewer than 10. The assumptions made for model values are *low* for less than or equal to 1760 and *high* for greater than 1760 and the degree of freedom may be calculated by using the formula df= (row-1) (column-1).

$$\chi 2 = \frac{N[|AD - BC| - N/2]^2)}{(A+B)(C+D)(A+C)(B+D)} \qquad (3)$$

In Eq. (3), A, B, C, and D are being replaced by 8_A, 2_B, 2_C, 8_D respectively. The computed value of $\chi 2$ is greater than the critical value for 1 degree of freedom at .05 level of significance, which is 3.84. Hence, the null hypothesis is rejected and it leads to the inference that testability estimation model (TEM^{OOD}) *gives quite comparable result regarding testability for all the systems to the given estimate of testability*.

Table VIII : $\chi 2$ Test Observation

	High	Low	Total
TEM^{OOD}	8_A	2_B	10
Industry Rating	2_C	8_D	10
Total	10	10	20
$\chi 2$ = 5.0			

V. CONCLUSIONS

The paper highlighted the importance of 'Understandability and Complexity' as factor or major contributor to testability. It has been proven that testability has strong correlation with Understandability and Complexity. These factors are obviously relevant and significant contributors in the context of testability while estimating it at design level. The model- TEM^{OOD} has been validated theoretically as well as empirically using experimental try-out. However, the model is validated on a small data set and it is to be done further on live industrial projects for better acceptability and utility.

REFERENCES

[1] Nazir M., Khan R. A., and Mustafa K., "A Metrics Based Model for Understandability Quantification", International Journal of Computing, Volume 2, Issue 4, April, 2010.

[2] Nazir M., Khan R. A., and Mustafa K. "Complexity Quantification Model: A Metric-Based Approach", 4th IEEE International Conference on Advanced Computing & Communication Technologies, Panipat, October 30, 2010.

[3] Genero M., 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", Proceedings of ADIS 2001, Workshop on decision support in Software Engineering, 2001.

[4] S. Mouchawrab, L. C. Briand, and Y. Labiche, "A measurement framework for object-oriented software testability", Information and Software Technology, Volume 47, Issue 15, pp. 979-997, Dec. 2005.

[5] Bruce W.N.Lo and Haifeng Shi, "A preliminary testability model for object-oriented software", in Proc. International Conf. on Software Engineering, Education, Practice, IEEE 1998, pp. 330-337.

[6] L. Zhao, "A new approach for software testability analysis", International Conference on Software Engineering, Proceeding of the 28th international conference on Software Engineering, Shanghai, 2006, pp. 985–988.

[7] R.V. Binder, "Design for testability in object-oriented systems". Communications of the ACM. Vol. 37(9), pp. 87-101, 1994.

[8] J. Gao and Ming-Chih Shih, "A component testability model for verification and measurement", In Proc. of the 29th Annual International Computer Software and Applications Conference, IEEE CS 2005, pp 211–218.

[9] Bach James, "Heuristics of software testability", 1999.

[10] Jungmayr S., "Testability during Design, Software technik-Trends.", Proceedings of the GI Working Group Test, Analysis and Verification of Software, Potsdam, pp. 10-11, 2002.

[11] E. Mulo, "Design for Testability in Software Systems", Master's Thesis, 2007.

[12] Nazir M., Khan R. A, Testability Estimation Model ($TEM^{OOD)}$, In Advances in Computer Science and Technology , LNCS, 2012, Volume 86, pp 178-187.

[13] Khan, R.A., Mustafa, K.: Metric based Testability Model for Object Oriented Design (MTMOOD). SIGSOFT Software Engineering Notes 34(2), March 2009.

