

Testability Measurement Framework: Design Phase Perspective

Abdullah¹, Dr. Reena Srivastava², Dr. M.H. Khan³

Research Scholar, School of Computer Application, BBD University, Lucknow, India ¹

Dean, School of Computer Application, BBD University, Lucknow, India ²

Associate Professor, Department of C.S. E., I.E.T., Lucknow, India ³

Abstract: Measuring testability early in the development life cycle considerably reduces the overall development cost, effort and rework. It is very expensive and error prone decision to correct the design to get better testability after the coding has started.

This paper proposes a testability measurement framework for object oriented software concerning at design phase. This framework correlates the testability factor with object oriented design properties and also correlates design properties with object oriented design metrics. No such framework has been presented in the literature that accurately measure testability of object oriented software taking design phase into consideration. The proposed framework reduces the gap between object oriented design properties, metrics and testability. This framework measures effort in measuring testability of object oriented software early at design phase and makes it possible to produce reliable end product within time and budget.

Keywords: Testability Measurement Framework, Testability Estimation, Software Testability, Software Quality, Design Phase, Object Oriented Design.

I. INTRODUCTION

Software is often the costliest item in a computer based application. Focus on software cost and quality is a part of engineering (analysis and design) and not of production. Moreover, cost of software is not dependent on volume of production. Like any product, software is designed based on some software specification; it is developed under formalized product reviews (quality assurance) and formalized testing procedures. It requires great amount of effort, time and money to design and develop high quality reliable and testable software. Effective software testing will significantly make a contribution to the delivery of superior quality software products, satisfied customers and will give exact and consistent results [4]. Software testing is a necessary and expensive activity of development life cycle that is used for determining whether a program has errors or not. Any technique that improves a software design at an early stage of software development life cycle can have highly beneficial impact on the final testing cost and effectiveness.

One of the key goals of software developers is to keep the testing and maintenance effort under control. For this control, software testability is one of the most important theories in design, and testing of software program and components [15]. Creating programs and components with high level testability constantly simplifies test operations; decreases test cost, and enhance software quality. A recent survey reveals that the cost incurred in testing often range from 40% to 80% of the entire cost involved in development [5]. One way to improve the testability of a software system is the design for testability, which can deal with dissimilar aspects of software including size, complexity, consistency, modularity, self-documentation and system structure [4]. Dino Esposito argued that

testability should be measured as a key attribute in order to promise the software quality [6]. Software testability analysis may be helpful to observe, and measure the excellence of software testing using an empirical analysis approach. According to software engineering ideology, if the course of action for development of any software is right, the chance of success of the software project is greatly increased. A successful project is one that satisfies the expectations of three major goals (1) low cost (2) completion on schedule (3) superior quality software [16].

II. MOTIVATION

The concerns that motivated us for the development of testability measurement framework are enumerated below.

1. Present testability models and metrics for object-oriented design (OOD) contain ambiguous details that limit applicability.
2. Unavailability of testability measurement framework for OOD quality assessment with lower-level metrics, well-defined in terms of design characteristics only.
3. Validation of quality characteristics values that whether they are in valid range or not.
4. Minimizing the cost and effort in measuring testability of OOD and producing what the customer wants.

III. SOFTWARE TESTABILITY MEASUREMENT

Testability is one of the important quality pointers. Its measurement, improves the test process. On the other hand, testability has always been indescribable thought and its truthful measurement or evaluation is a complex exercise [1, 2]. There is no understandable definition to

‘what phase of software are truly related to testability’ [2]. A number of progresses on the measurement of testability and design for testability have been reported in the literature [7, 8, 9, 10, 11]. Practitioners repeatedly support that testability should be considered early in the design phase. Our objective is to make available a complete framework to facilitate measuring and assessing testability of object oriented design in a realistic approach, with a center of attention on the analysis and design phase of object-oriented development.

Our main inspiration is that it is through the analysis and design stage that testability study can yield the highest payoff: design decisions can be made to get better testability before implementation starts. As soon as the design meets the testability requirements, it can be implemented and the constraints added for testability improvement of the design must be verified before testing.

IV. TESTABILITY MEASUREMENT AT DESIGN PHASE

A full measurement can only be achieved after a product is complete [12]. Although software testability is most noticeably significant during testing, but paying consideration to testability early in the development process can potentially enhanced testing efficiency, reduce cost and appreciably improves testing phase effectiveness [7]. There is a universal conformity between industry professionals and academicians to incorporate testability within the development life cycle in order to deliver secure, safe and reliable software inside time and budget [13].

Bohem found that among 60% to 70% of all the faults detected in large level projects was either specification or design faults [14]. A decision to modify the design in order to recover testability after coding has started may be extremely expensive and error-prone [4]. The target of increasing the testability of object oriented design is not only to detect defects but more significantly, to detect defects as quickly as they are occurred. This results in reducing the cost and time to fix the bug and producing superior quality software each build of the release.

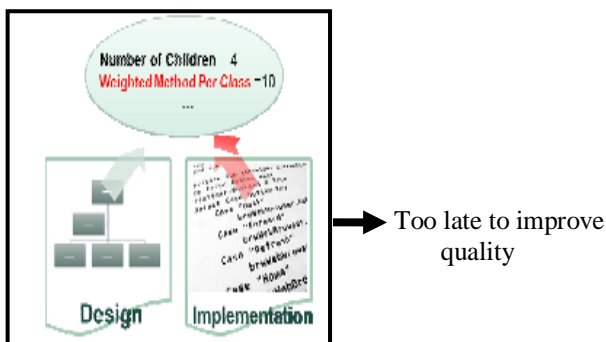


Fig. 1.

V. TESTABILITY MEASUREMENT FRAMEWORK

Testability measurement framework for object oriented design considers that measurement is a tool for measuring the quality, effectiveness of an estimation activity. Software testability measurement process involves the both, quantification and prediction with the help of the

software testability model [1]. This framework form a roadmap to industry personnel and researchers to assess, and preferably, measure software testability at design phase of development life cycle.

(a) PREMISES

A framework is a theoretical explanation of a complex process. It provides a truthful support for future research. The framework for Testability measurement of an object oriented design has the following assumptions:

- ➔ Testability of an object oriented design is affected by numerous factors; it is not easy to obtain a clear view on all the potential factors that have positive impact on testability measurement.
- ➔ Object oriented design quality attributes is affected by various object oriented design properties.
- ➔ The framework measures the testability with the help of design properties and design level testability factors.
- ➔ One can chose any of the testability factor to measure the testability of object oriented design.

(b) THE FRAMEWORK

The prescriptive framework as depicted in figure (2) has been proposed to measure testability of object oriented software at design phase. The proposed framework for testability measurement consists of following seven phases

1. RECOGNITION OF TESTABILITY FACTOR

A perfect measure of software quality totally depends on testability measurement. Testability is a high level factor to software quality. In order to measure testability, its straight measures are to be recognized. In this phase, all testability factors are to be documented. Then after design phase factors will also be finalized keeping in view their impact on the overall testability.

2. OO SOFTWARE CHARACTERIZATION

Object oriented design overcomes the negative aspect of procedure oriented programming. The flaws encounters in the procedural approach are uninvolvement through object oriented approach. Object oriented design treats data as an important element in the program development and does not permit it to move freely within the system. The prominent object oriented design properties are: Inheritance, Coupling, Cohesion and Encapsulation. Dissimilar software characteristics have their impact on testability and quality as well. Object oriented software characteristics that have affirmative impact on testability measurement will be recognized in this phase. The involvement of each property to boost the design will also be analyzed

3. METRIC SELECTION

Software metrics are measures that are used to quantify different attributes of the software products, software development resources and software development process in making their classes testable. Metric selection is important step in measuring testability of object oriented design. In the lack of any testability metric suited at design phase, a group of testability metrics is to be proposed that may not provide the purpose.

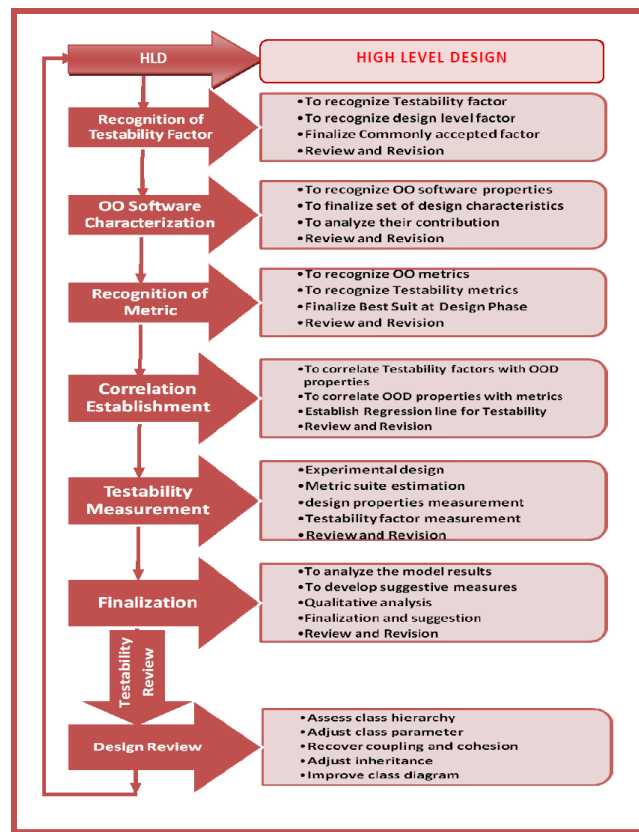


Fig.2. Testability Measurement Framework: Design Phase Perspective

4. CORRELATION ESTABLISHMENT

This is the most important step of the above framework, Correlation is a statistical relation among two or more variables; coefficient of correlation (simple relation) method was used for measuring the degree of correlation between variables. The range of correlation 'r' exist between ± 1 , positive value of 'r' indicates positive correlation between the two variables and negative value of 'r' indicates negative correlation between the two variables. The value of 'r' vary to +1 indicate perfect positive correlation and vary to -1 designate perfect negative correlation, and $r=0$ (shows no correlation) between the two variables. Correlation phase in framework correlated the identified testability factors with OO design properties. A regression line will be established to measure testability factors in terms of design characteristics with the help of design metrics [4].

5. TESTABILITY MEASUREMENT

Established hierarchical multiple regression analysis will be used to measure testability factors using design metrics value. Subsequent to a model is developed to evaluate software testability in design phase. Moreover, the association of Testability with these factors has been tested and justified with the help of statistical procedures.

6. FINALIZATION

On the basis of the testability measurement conclusions, an evaluation of testability factors is performed. A related result will be discussed and used for review and revision

of the identified design. This phase will assist in benchmarking software products according to their testability.

7. DESIGN REVIEW

On the basis of the results achieved from the finalization phase, the particular design is to be reviewed and revised to obtain better level of testability. Design characteristics are to be sincerely examined and may be adjusted as a result in order to achieve the index value.

8. REVIEW AND REVISION

Review and revision is common in all phase of the testability measurement framework. In this course of action each phase of the framework are reviewed and revised. Absolute suggestions and improvements are included in this step. The changes occurred at any stage may implement during the review and revision at any phase of the framework.

VI. FRAMEWORK SIGNIFICANCE

The framework has the following significance which is as follows:

➔ Proposed framework make possible to reduce effort and cost in measuring testability of object oriented design.

➔ It may assist to find the effect of object oriented design properties over the software testability measurement.

- It may help out to advancement in development techniques and methodologies of object oriented design over the estimation of software testability.
- It may assist to make up high quality reliable and testable software.
- It may help to decrease rework and expenditure of software testing and maintenance.
- It may facilitate to producing software that satisfies specifications.

VII. CONCLUSION

A testability measurement framework for object oriented design; design phase perspective has been developed in this study. The proposed framework correlates the object oriented design properties with testability factors. The framework bridges the gap between object oriented design characteristics, metrics and testability. Framework helps putting testability benchmarking of software projects. The proposed framework is standard in nature, and may be used by industry experts to measure testability in order to make design decisions near the beginning in the development life cycle. Strong hypothetical basis presented by the study to supports the claim of the framework's usability and reliability to measure testability of object oriented software at design phase. Framework's implementation is in steps forward, and will come out as our forthcoming work.

REFERENCES

- [1] S. Mouchawrab, L. C. Briand, and Y. Labiche, "A measurement framework for object-oriented software testability", *Info. And Software Technology*, Volume 47, Issue 15, Pages 979-997, December 2005.
- [2] 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
- [3] Lee, Ming-Chang. "Software Quality Factors and Software Quality Metrics to Enhance Software Quality Assurance." *British Journal of Applied Science & Technology* 4.21 (2014).
- [4] 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.
- [5] A. Zaidman et. al, "On how developers test open source software systems", Technical Report TUD-SERG-2007- 012, Delft University of Technology, Software Engineering Research Group, 2007.
- [6] DinoEsposito, "Design Your Classes for Testability", 2008. URL:<http://dotnetslackers.com/articles/nnet/Design-Your-Classes-for-Testability.aspx>
- [7] Improving the Testability of Object-oriented Software during Testing and Debugging Processes, Sujata Khatri, R.S. Chhillar, V.B.Singh, *International Journal of Computer Applications* (0975 – 8887) Volume 35– No.11, December 2011.
- [8] S. Jungmayr, "Design for Testability", *CONQUEST* 2002, pp. 57-64.
- [9] Pettichord, B. Design for Testability. In *Proc. of Pacific Northwest Software Quality Conference*, 2002.
- [10] Jimenez, G., Taj, S., and Weaver, J. Design for Testability. In *Proceedings of the 9th Annual NCHIA Conference*, 2005.
- [11] Jungmayr, S. Testability Measurement and Software Dependencies. In *Proceedings of the 12th International Workshop on Software Measurement*, pp. 179–202, October 2002.
- [12] 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.
- [13] Testability Estimation Model (TEMOOD): M. Nazir & R.A.Khan, Lecture Notes of the Institute for Computer Sciences, Social- Informatics and Telecommunications Engineering, Vol. 85, Meghanathan, Natarajan; Chaki, Nabendu; Nagamalai, Dhinakaran (Eds.), Volume 85, Part 3, LNCS, Springer-Verlag, 2012, pp 178-187, (ISBN 978-3-642-27307-0). January 2012.
- [14] Nasib Singh Gill. "Software Engineering" Khanna Book Publishing Company (P) LTD., ISBN 818732517-8, page no.07, 2002.
- [15] Tiwari, Rajeev, and Noopur Goel. "Reuse: reducing test effort." *ACM SIGSOFT Software Engineering Notes* 38, no. 2: 1-11, 2013.
- [16] Jibitesh Mishra, Ashok Mohanty "Software Engineering. Pearson, 2012. www.pearsoned.co.in/jibiteshmishra

BIOGRAPHIES



Abdullah received the MCA degree from Uttar Pradesh Technical University, Lucknow, in 2006. He is currently working as an Associate Professor, in the Department of Computer Application, at Institute of Environment and Management, Lucknow. His research interests Include Software testability, Software Quality Estimation. He has written various books and study materials for (North Orissa University) Orissa, (Suresh Gyan Vihar University, Jaipur) Rajasthan, (Bharati Vidyapeeth University, Pune) Maharashtra, NAAC RE-ACCREDITED "A" Grade University.



Dr. Reena Srivastava is currently working as Dean, School of Computer Applications at BBD University, Lucknow. She received her Ph.D. degree from MNNIT Allahabad, India. Her research area includes Multi-Relational Classification, Privacy Preserving Data Mining and Software Engineering.



Dr. M. H. Khan, Associate Professor, Department of Computer Science and Engineering at IET Lucknow UP. Obtained his MCA degree from Aligarh Muslim University (Central University) in 1991 .Later he did his PhD from Lucknow University. He has around 25 years rich teaching experience at UG and PG level. His area of research is Software Engineering. Dr. Khan published numerous articles, several papers in the National and International Journals and conference proceedings.