

Testability Quantification Framework of Object Oriented Software: A New Perspective

Mahfuzul Huda¹, Y.D.S.Arya², M. H. Khan³

Research Scholar, Department of Computer Sc. & Engineering, Invertis University, Bareilly, India¹
Pro -Vice Chancellor, Invertis University Bareilly, India²

Associate Professor, Department of Computer Science and Engineering at IET Lucknow, India³

Abstract: The quality of any object oriented design is critical because it has a great influence on overall quality of finally delivered software product. Testability quantification early in the software development process is a criterion of crucial important to software development team. Testability has always been an indefinable concept its correct measurement, quantification or evaluation is a difficult task because of its potential factors. Testability analysis of object oriented software at an initial stage of software development process has been identified as a key factor for high quality product. A best suited object oriented design (OOD) properties and its associated metrics are helpful if applied in the early stage of development process. This paper describes an improved testability quantification framework using the identified set of potential factors, OOD properties and OOD metrics for software products at early phase, exclusively at design time. The proposed framework relates OOD properties to high level quality attributes/constructs using appropriate information to develop quality product and it may used to benchmark software products according to their key attribute. The objective of this research work to encourage researchers and developers to provide a framework to access and quantify software testability at early stage of development life cycle.

Keywords: Testability, Object Oriented Design, Testability Framework, Object Oriented Software, Software Quality Model, Software Testing, Effort Estimation

I. INTRODUCTION

This Software development processes mainly focus on reducing errors, finding and correcting software faults that does occur, and support to deliver high quality software within specified time and specified resources. However, the description of different characteristics that contribute to the concept of 'quality' is not enough on its own in order to assure quality in practice. Software quality is still vague terminology that means different things to different people, the way we measure quality depends on the viewpoint we take [1, 2]. It is well understood that delivering quality software is no longer an advantage but is a necessary factor. Unfortunately, most of the industries not only fail to deliver a quality product to their users, but also do not recognize the significant quality factors [3, 4, and 5]. Software testing is an important discipline of software engineering, that maintain the process of checking software to ensure that it behaves as per the specified requirements and conditions i.e. fixed time and bounded budget. Current software practice places a strong emphasis on module testing, to the extent that the amount of test components produced on a project can exceed the amount of actual application program required. Developing programs and components with good testability always reduced test operation this shows the importance of testability as a feature of software [6, 7, and 8]. Testability is a quality factor; its measurement or evaluation can be used to predict the amount of effort required for testing and help allocating required resources. There is no clear definition to 'what aspects of software are actually related to testability' [9, 10]. Thus, testability is one of the important quality factors in development

process where in software engineer can identified completeness, correctness and quality of a certain piece of software in significant amount of time and software development effort.

However, testability has always been an elusive concept and its correct measurement or quantification is a tough exercise. From the literature survey it has been observed that, most of the studies measure testability or precisely the attributes that have impact on testability at the source code stage. It has been also inferred from the literature on testability factors that there is an acute need of proposing a commonly accepted set of the factors affecting software testability [2]. Estimating testability at a later stage leads to the late arrival of desired information, leading to late decisions about changes in design [11,12]. This simply increases cost and rework. Therefore, early evaluation of testability in the development process may enhance quality and reduce testing efforts and costs. . Despite the fact that software testability has always been challenging task and its quantification is a complex exercise, hence an appropriate framework is required to perform testing activities properly and effectively. Software testability always supports the testing process and facilitates the creation of better quality software within given time and budget [13, 14, and 15].

II. TESTABILITY: A QUALITY PERSPECTIVE

A most common definition of Software Testability is ease of facilitating testing process. Testability is an external software quality attribute that evaluates the complexity and the effort required for software testing. Software

testability is a key aspect to allow the detection of difficult to uncover defects in software. Defined testability as 'the effort with which faults will manifest themselves as failure when the software undergoes the testing processes. According to this definition, a typical software with high testability exposes its faults easily. Binder in 1994 defined software testability as the relative ease and expanses of revealing software faults [1]. Bach in 1999 defined testability is anything that makes software easier to test by making it easier to design and execute tests [16].

The IEEE glossary of software engineering 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]. Testability is a non-functional requirement important to the testing team members and the users who are involved in user acceptance testing. Non functional requirements are mostly quality requirements and may make the customer satisfied and happy. Hence testability is a distinct quality characteristic of software that makes it easier to test and one of the important concepts in software creation at initial stage of software development process exclusively at design time of software program and components. Testability has been identified as a distinct characteristic of software quality. Building programs and components with high level testability always simplifies test process, reduces total test cost and hence increases software quality [3, 5]. Researchers proposed different characteristics of software quality ; many of them consider testability as an important part of their quality characteristics [19,20,21,22]. In ISO 1926 quality model (ISO 1991); testability holds a prominent place as part of the reusability characteristic of software quality [4].

It is obvious from forgoing discussion that testability has always been an elusive concept and its correct measurement or evaluation is a difficult exercise because various potential factors have affect on software testability measurement. Testability is one of the most important quality indicators [2]. Most of the studies measure testability or more precisely the attributes that have impact on testability but at the source code level. However, testability estimation at the source code level is a good indicator of effort estimation; it leads to the late arrival of information in the development process. Evaluating testability at later phase of development process after coding has been started may be very expensive and error-prone. But if testability is evaluated earlier in the development process, before coding starts, it may greatly reduce the overall cost, time and rework. As a result it can accelerate the software development life cycle.

III. TESTABILITY QUANTIFICATION IN DESIGN PHASE

Software strives to build high quality products in a cost effective manner. Object orientation designing and development is becoming very preferred paradigm in industrial software development environments for large-stage may have highly beneficial impact on the testing

cost, its efficiency and effectiveness. Testability has been described as the ability to perform cost effective testing 'and hence it is becoming increasingly important to a project's success. Several experts pointed out the importance of testability and design for testability .especially in the context of large systems. Testability estimation early in design phase is highly emphasized in the study; hence, considered important for the delivery of quality software. Our goal is to provide a comprehensive framework to help measuring and quantifying testability in an empirical approach with a focus on the analysis and design stages of object-oriented development. Our main motivation is that it is during the analysis and design stages that testability analysis can yield the highest payoff: design decisions can be made to improve testability before implementation starts. When the design meets the testability requirements at early stage in creation process, it can be implemented and the quality criteria added for testability improvement of the design must be verified before testing.

Any technique that improves a software design at an early stage can have high impact on the final testing cost and its efficiency. Although software testability is most relevant during testing, but paying attention to testability early in the development process can potentially enhanced testing and significantly improves testing phase effectiveness. There is a common agreement among industry professionals and academicians to integrate testability with the development life cycle in order to deliver secure, effective, safe and reliable software within given time and budget.

IV. TESTABILITY QUANTIFICATION FRAMEWORK

A comprehensive survey and analysis of relevant literature reveals feasibility of developing a testability quantification framework at design phase. Taking into access the need and significance of estimating testability in design time, a prescriptive 'testability quantification framework' is hereby proposed, as depicted in figure 1 .The essential and desirable features have been identified and a commonly accepted set of testability factors have also proposed that can directly or indirectly affect testability and make it easy to calculate testability index values for project ranking. The proposed framework consists eight phases including a common phase of 'Design Review' and an input phase namely 'Generalize Object Oriented Design (HLD/LLD)' a brief description of the framework is given as follows.

A. Determine Testability factor

Software testability is an external software quality characteristic that evaluates /estimate the effort required for testing. In order to quantify testability, its direct quality criteria are to be identified. In this phase the minimal set of testability factors to be identified and best suited level quality criteria also be investigated keeping in view their importance on the overall testability index, and introduced the testability measures for software components by defining effectiveness and reusability notions. Thus testability measures are only applied to functional specification by examining input and output domain.

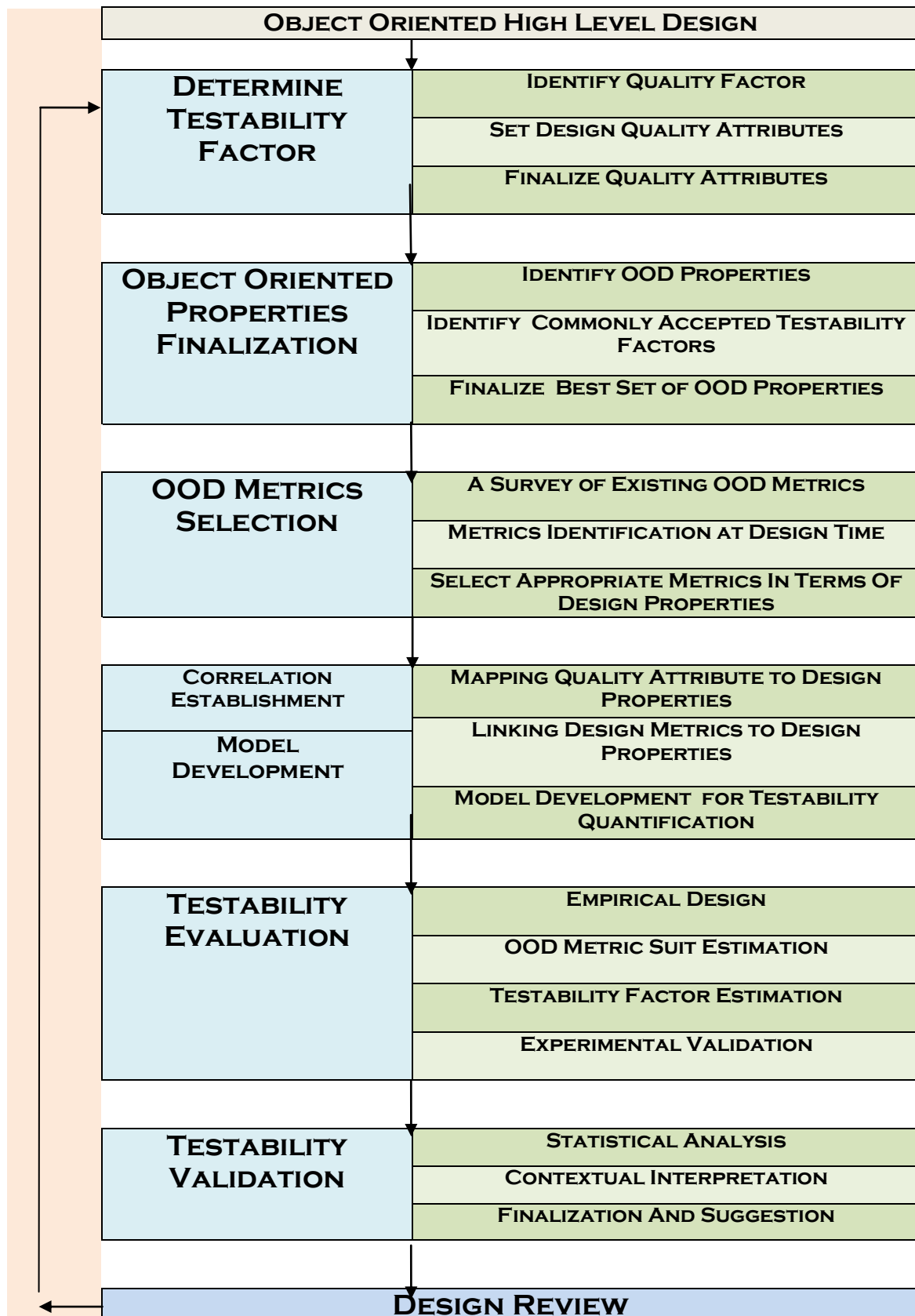


FIGURE 1: TESTABILITY QUANTIFICATION FRAMEWORK:A NEW PERSPECTIVE

B. Object Oriented Properties Finalization

Object oriented programming offers support to deliver software product with higher quality and lower testing cost. Major Object oriented programming advantage is its support for software reusability criteria design effectiveness. Which may be achieved through Object Oriented Programming properties such as inheritance, encapsulation, Polymorphism, Abstraction, Coupling, Cohesion etc.

C. OOD Metrics Selection

Object oriented programming metrics are popular calculation tool. Metrics determination is an important phase in testability estimation process [17, 18]. A suit of testability metrics is to be proposed for testability index quantification process with covering all the project's design attributes.

D. Correlation Establishment

This is a statistical relation among two or more variable for measure degree of correlation between variables. In order to establish a correlation between design constructs and testability a regression line will be established to quantify testability factors with design properties. A range of 0 to ± 1 was selected for the computed values of the testability factor. This is an important process in proposed framework.

E. Model Development

A standard quality model has been considered as a basis to develop 'metrics based quantification model for object oriented design' to quantify testability index values. Where the identified quality criteria are to be correlated with the object oriented design properties and object oriented design metrics.

F. Testability Evaluation

This phase assesses how well the proposed design metrics suit is able to evaluate the testability of OOS, the internal characteristics of the product's design influence the testability and other quality criteria and that varies significantly overall quality of the software product.

G. Testability Validation

On the basis of testability evaluation results a testability validation process performed by the help of statistical analysis and contextual interpretation to correlate values between calculated by proposed model and given by evaluators quantification procedures. This stage will help in ranking software products according to their class testability.

H. Design review

On the basis of product ranking and class testability index results obtained from above given phase in framework, the given design is to be reviewed to get better level of testability index values. Design criteria are to be critically examined and may be adjusted accordingly in order to get appropriate testability index value.

V. CONCLUSION

Testability has always been a complex concept and its accurate quantification is a complicated task. The developed framework for the quantification of testability in object oriented design during software development life cycle has the following significance in academic/industry.

- It may help to find out OOD properties effect over the software testability quantification process.
- It may assist developer to develop high quality testable software within specified condition.
- It may facilitate to providing software that satisfy required specification of individual.
- It may used by researchers to quantify testability early in development process for high quality software products.

The developed framework comprises eight steps namely Determine Testability factor, Object Oriented Properties Understanding, OOD Metrics Determination, Correlation Establishment, Model Development, Testability Evaluation, Testability Validation, along with additional common steps of design review. In the phase of 'design review', the OOD constructs are to be critically examined so that parameters can be adjusted accordingly to achieve the index value.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my supervisor Prof. Dr. YDS Arya & Co-supervisor Assoc. Prof. Dr. M H Khan for the continuous support of my PhD study and research, motivation, enthusiasm. Their guidance helped me in all the time of research. Last but not the least; I would like to thank my parent for their patience, understanding and support that drive me to complete my study

REFERENCES

1. Binder, Robert V. "Design for testability in object-oriented systems." *Communications of the ACM* 37.9 (1994): 87-101.
2. 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 (IJSET)*, Vol. 3, Issue 10, pp: 1313-1319 Oct, 2014
3. Sharma, M., & Mall, R. (2009). Automatic generation of test specifications for coverage of system state transitions. *Information and Software Technology*, 51(2), 418-432. doi:10.1016/j.infsof.2008.05.002.
4. ISO, "ISO/IEC 9126-1: Software Engg.-Product Quality-Part-1: Quality Model", Geneva, Switzerland, 2001
5. 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

6. IEEE Press, “IEEE Standard Glossary of Software Engineering Technology,” ANSI/IEEE Standard 610.12-1990, 1990.
7. Lee, Ming-Chang. "Software Quality Factors and Software Quality Metrics to Enhance Software Quality Assurance." *British Journal of Applied Science & Technology* 4.21 (2014).
8. Briand, L. C., Labiche, Y., & He, S. (2009). Automating regression test selection based on UML designs. *Information and Software Technology*, 51(1), 16–30. doi:10.1016/j.infsof.2008.09.010.
9. 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.
10. Singh, Y., & Saha, A. (2010). Improving the testability of object oriented software through software contracts. *ACM SIGSOFT Software Engineering Notes*, 35(1), 1. doi:10.1145/1668862.1668869.
11. Samar Mouchawrab ,Carleton University, Technical Report SCE-05-05 ,2005
12. 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, LNICST, Springer-Verlag, 2012, pp 178-187, (ISBN 978-3-642-27307-0). January 2012.
13. 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.
14. Jagdish Bansia, “A Hierarchical Model for Object Oriented Design Quality Assessment”, *IEEE Transaction of Software Engineering*, Volume 28, No. 1, January 2002, and pp: 4-17
15. Genero M., J. Olivias, 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.
16. Bach, James. "Heuristics of Software Testability" (1999).
17. Shyam R. Chidamber, Chris F. Kemerer, “Towards A Metrics Suit for Object Oriented Design”, OOPSLA, ACM, 1991, pp.197-211.
18. Kout, A., Toure, F., & Badri, M. (2011). An empirical analysis of a testability model for object-oriented programs. *ACM SIGSOFT Software Engineering Notes*, 36(4), 1. doi:10.1145/1988997.1989020.
19. McCall JA, Richards PK, Walters GF. Factors in software quality, RADC TR-77-369: 1977. (Rome: Rome Air Development Center)
20. Boehm BW, Brow JR, Lipow M, McLeod G, Merritt M. Characteristics of software quality. North Holland Publishing. Amsterdam, the Netherlands; 1978.
21. Dromey RG. Concerning the Chimera (software quality). *IEEE Software*. 1996;1:33-43.
22. Grady, RB. Practical software metrics for project management and process improvement, Prentice Hall; 1992.

BIOGRAPHIES



Mahfuzul Huda, he is pursuing PhD in the field of software engineering from Invertis University, Bareilly, India. He has more than 7 year of teaching & research experience. He is currently working in the area of Software Testability in Object Oriented Design.

He has also published & presented papers in refereed journals and conferences.



Dr. Y.D.S. Arya is currently working as **Pro - Vice Chancellor, Invertis University Bareilly, India**. **Dr. Arya** obtained M.Tech. Degree in Computer Science and Engineering from IIT Kanpur before his Ph.D. in Computer Science and Engineering. He

has more than 31 years of experience in teaching and programming. He has developed few software packages and has contributed in the development of Solaris Kernel 9.0. Apart from working at IIT Kanpur, he has worked with Sun Microsystems, San Francisco, Ca, USA and Fujitsu at Numazu Company in Japan.



Dr. M. H. Khan Associate Professor, Department of Computer Science and Engineering at IET Lucknow, India. Obtained his MCA degree from Aligarh Muslim University (Central University) in 1989 .Later he did his PhD from Lucknow University. He has

more than 26 years of experience in teaching and programming. His area of research is Software Engineering. **Dr. Khan** published numerous articles, several papers in refereed journals and conferences.