ISSN: 2321-7782 (Online)
Volume 3, Issue 11, November 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Reusability Estimation of Object Oriented Software: A Systematic Review

Namrata Chopra[1]
Research Scholar
Department of Computer Science
Goel Institute of Technology & Management
Lucknow (U.P.) – India

Namrata Dhanda[2]
Associate Professor and head of CS/IT
Department of Computer Science
Goel Institute of Technology & Management
Lucknow (U.P.) – India

Abstract: ***One of the promises which object oriented programming holds is that it increases software reusability. In fact, software components designed in object oriented concept is easier to be reused than those designed in conventional programming language. But the state of the art software reusability in most Object oriented environments is still very limited. In object oriented programming, estimating reusability plays a key role in reducing cost and improving the software quality. Object oriented system helps in achieving the concept of reusability through different types of inheritance programs, which further facilitate in developing reusable software components. Object oriented metrics recognize the effectiveness of each reuse policy. Software reusability has significant impact on overall software quality. Quality of software increases as reuse of software modules increases. Although, software quality improvement cannot be understood unless it is calculated. This paper examines how object oriented system increases software reusability, what are the limitations of software reuse in current state, and how to improve object oriented software reusability.***

Keywords: Software reusability; Systematic software reuse; software development; Reusability Estimation

I. INTRODUCTION

Software reusability is considered as a significant technique of avoiding re-coding of work and improving the quality of the software development process [6]. The object-oriented principles have provided estimation tools and basic ideas, which can be recognized and applied in the software development. Reuse of existing software components increase the overall quality and efficiency of software development life cycle and maintenance process [12, 14]. Software reusability reduces the amount of efforts that need to be delivered from scratch and hence less testing time for new software. Industrial observers suggest that a reuse strategy could save up to 20% of development costs [9]. C++ templates are used to support the concept of reusability in object oriented programming. Object oriented software is a group of classes which are abstract data types, and templates are a way of making classes more abstract without actually knowing what data type will be controlled by the functions of the class [16, 19]. This raises questions about how general programming included in the structure of templates in the code can be measured to identify effectiveness of this reuse strategy. The measurement of reuse would assist software developers to examine current levels of reuse and provide insight in developing quality software that is easily reused.

II. SOFTWARE REUSABILITY

The IEEE defines software reusability as 'The ability of a software system or component to perform its desired function under its stated condition to evaluate the quality of software [4, 23].' Reusability is one of the most important attribute of software quality for delivering high quality software. It is also an important factor to quality estimation of object oriented software at an early phase of software development life cycle [1- 3]. Developing software system from existing classes i.e. Reuse offers many advantages:

- Development cost is reduced.
- Reliability is increased.
- Less time to market.
- Low maintenance cost.

Reusability is an important characteristic of a high-quality software classes. Software reusability is a characteristic that refers to the anticipated reuse prospective of a software system [21, 22, 30, 31]. Reusability criteria constantly support developer for quality oriented software design at an early stage of software development life cycle [24-29]. To design and develop a high quality and effective product, reusability plays an important role for estimation of software quality [32, 33]. Estimating reusability factor early in the development process may greatly reduce the overall cost and enhance the customer satisfaction.

III. Review of Related Work

Several works has been done in this area. Some of them are as follows:

In 2006, Parvinder S. Sandhu & Hardeep Singh [7] proposed reusability evaluation model for estimating the reusability of software components. The authors analyzed the CK metrics and eliminate the inconsistencies and developed the framework of metrics to evaluate the reusability. Also they proposed Neuro fuzzy Inference Engine can be used to estimate the reusability. Author proposed an algorithm in which the inputs can be given to Neuro fuzzy Inference Engine in the form of structural attributes and expected output can be achieved in terms of reusability. The following refined CK metric suit are used in this model are: Tuned Weighted Methods per Class (TWMC), Lack of Tuned Degree of Inheritance (LTDIT), Lack of Tuned Number of Children (LTNOC), Lack of Coupling between Objects (LCBO), and Lack of Cohesion in Methods (LCOM).

In 2007, Parvinder Singh. & Hardeep Singh [8] proposed Quantitative Investigation of impact of the factors involvement towards estimating the reusability of software modules which helps to evaluate the quality of the components. They used Taguchi approach in analyzing the significance of different attributes in choosing the reusability level of a particular software component. They inferred from the results that the software complexity is the most significant factor in deciding the better level reusability of a function based software component and in case of object oriented, coupling and complexity collectively play the significant role in high reusability. The developed metrics for function oriented model are: Complexity Using Mc Cabe's Measure, Regularity Metric, Cyclometric Metric, Reuse Frequency Metric, Halstead Software Science Indicator and Coupling Metric. The proposed metrics for object oriented paradigm are: Tuned Weighted Methods per Class (TWMC), Lack of Tuned Degree of Inheritance (LTDIT), Lack of Tuned Number of Children (LTNOC), Lack of Coupling between Objects (LCBO) and Lack of Cohesion in Methods (LCOM).

In 2008, GUI and Paul D. Scott [10] proposed new measure of coupling and cohesion to assess the reusability of components. They shown that the new measures proposed by them was consistently superior at the time of measuring the component reusability. They used five metrics for coupling are : Weighted Transitive Coupling (WTCoup), Coupling Factor (CF), Classes Between Objects (CBO) , Response For Class (RFC), Data Abstraction coupling (DAC) and five metrics for cohesion are : Weighted Transitive Cohesion (WTCoh), RLCOM, Lack of Cohesion in Methods (LCOM), Variant Lack of Cohesion in Methods (LCOM3) and Tight Class Cohesion (TCC). Two approaches were used to estimate the performance of the various measurements in predicting software reusability. They are Linear Regression and Rank Correlation. They clearly demonstrated that their proposed metrics Weighted Transitive Coupling (WTCoup) and Weighted Transitive Cohesion (WTCoh) for cohesion and coupling are good predictors for evaluation the reusability of the component.

In 2009, Parvinder S. Sandhu, Harpreet Kaur and Amanpreet Singh [11] proposed reusability evaluation system for object oriented software components. They proposed software metrics and quality of the software components were inferred by

different neural network based approaches. The proposed metrics for object oriented paradigm are: Tuned Weighted Methods per Class (TWMC), Lack of Tuned Degree of Inheritance (LTDIT), Lack of Tuned Number of Children (LTNOC), Lack of Coupling between Objects (LCBO) and Lack of Cohesion in Methods (LCOM). The different neural network approaches are used for the modelling of the reusability data. The following neural networks algorithms were experimented by the author. They are

- Batch-Gradient-Descent with momentum
- Variable Learning Rate training with momentum
- Variable Learning Rate without momentum
- Resilient Back propagation
- Fletcher-Reeves version of the conjugate
- Polak-Ribiere Update version of the conjugate
- Gradient Powell-Beale Restarts version of the conjugate
- Gradient Quasi-Newton BFGS Algorithm
- Scaled Conjugate Gradient
- Quasi-Newton One Step Secant Algorithm
- Levenberg-Marquardt Algorithm
- Generalized Regression Neural Networks

In 2010, Sonia Manhas, & Nirvair Neeru [13] proposed reusability evaluation model for estimating reusability of software components. They proposed structural attributes and software metrics to evaluate the reusability of the component by experimenting with five different Neural Network based approaches by taking the metric value as input. The estimated reusability value enables to recognize the good quality components automatically. The proposed metrics are: Complexity Using McCabe's Measure, Regularity Metric, Cyclometric Metric, Reuse Frequency Metric, Halstead Software Science Indicator and Coupling Metric... The different neural network approaches are used for the modelling of the reusability data. The following five neural networks algorithms were experimented they author. They are

- Variable-Learning Rate training with momentum
- Variable Learning Rate
- Batch Gradient Descent with momentum
- Resilient Back propagation
- Batch Gradient Descent

In 2011, Fazal Amin, & Alan Oxley [15] proposed reusability attribute model for estimating reusability of software components. They proposed six attributes related to the reusability of software components. The proposed model is derived using the GQM approach and this helps to understand the factors to measure the software quality. Also they provide the metrics which are used to assess the reusability. The metrics which are used for reusability estimation are: Size Metrics, Coupling metrics, Cohesion metrics and Variability Metrics. The metric used for measuring the size is Lines of Code (LOC) and Number of Method (NOM). The metric used for measuring the coupling is coupling between Object Classes (CBO) and metric used for measuring the cohesion is Lack of Cohesion in Methods (LCOM). The variability metrics are based on the theory and

mechanism of inheritance. The metrics used to measure variability are Depth of Inheritance Tree (DIT) and Number of Children (NOC).

In 2012, Ajay Kumar [18] proposed a reusability model for categorization of the reusability of software modules using support vector machine. The metrics used for identification of reusable software modules are Complexity Using McCabe's Measure, Regularity Metric, Cyclometric Metric, Reuse Frequency Metric, Halstead Software Science Indicator and Coupling Metric.

IV. Comparative Analysis of the Existing Models

In this section, we present comparisons among some reusability estimation model based on some factors and reuse metrics. The results of the comparisons are presented in the below table, Table I.

TABLE I

Comparisons of Reusability Estimation Model and Related Approach

Year	Authors	Models	Method
2005	Richard W. & Selby [5]	Evaluation software reuse empirically by mining software repositories	Goal Question Metric (GQM)
2006	Parvinder S. & Sandhu [7]	Reusability Evaluation Model	Neuro-Fuzzy Inference System
2007	Parvinder S. & Sandhu [8]	Quantitative Investigation model	Taguchi Approach
2008	Gui Gui & Paul D. Scott [10]	Evaluation of software component reusability model	Linear Regression and Rank Correlation
2009	Parvinder S. & Sandhu, Harpreet Kaur [11]	Reusability Evaluation system	Neural Network Approaches
2010	Sonia Manhas & Rajeev Vashisht [13]	Reusability Evaluation Model	Neural Network Algorithms
2011	Nasib S. Gill & Sunil Sikka [17]	Inheritance hierarchy Based model	Metrics based approach
2011	Fazal-e-Amin & Ahmad Kamil Mahmood [15]	Reusability Attribute Model	Goal Question Metric (GQM)
2012	Ajay Kumar [18]	Reusability classification model	SVM classifier
2014	Neha Goyal & Deepali Gupta [20]	Reusability Calculation	CK Metric

V. Critical Observations

After successful completion of the literature survey some important observations can be enumerated as follows.

» If we estimate the software reusability at an early phase in the software development life cycle may improve the software quality and as well as client satisfaction, and reduce overall cost, time and effort of rework.

» In order to reducing effort in measuring reusability of object oriented design we need to identify a minimal set of reusability criteria for object oriented design, which have direct impact on reusability measurement.

» Object oriented software characteristics must be identified and then the software reusability criteria significant at the design phase of development life cycle should be finalized.

» Further, reusability metrics must be selected at the design phase because metric selection is an important step in reusability estimation of objects oriented design.

VI. CONCLUSION

A systematic literature review of the related works in the field of software reusability estimation model is performed and the results of the systematic literature review are discussed in the form of critical observations in this paper. Related work classification scheme serves as a framework for future research to differentiate between different reusability estimation to test more models in practice. We have consolidated array of existing software reusability model based on the metrics and approach for qualifying the components for reuse. The comparative analysis of the existing models classifies the models discussed by this study. Although our review has explored the all existing reusability estimation model, in conclusion further studies are needed to develop a systematic reusability estimation model that are consistent and reliable, in current object oriented development environment.

References

1. Z.M. Zhang, Y.T. Zhuang, Y.H. Pan, "Object-Oriented Software Reverse Engineering," J. of Computer Research and Development, vol.40, no.7, pp.1062–1068, 2003.
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. William Frakes, Carol Terry, "Software Reuse : Metrics and Models", ACM Computing Surveys, Vol.28, No.2, pp.416-435, Jun 1996.
4. IEEE Press, "IEEE Standard Glossary of Software Engineering Technology,"ANSI/IEEE Standard 610.12-1990, 1990.
5. Richard W. Selby, "Enabling Reuse-Based Software Development of Large-Scale Systems," IEEE Transaction of Software Engineering, Vol. 31, No. 6, PP. 495-510, Jun 2005.
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. Parvinder Singh Sandhu and Hardeep Singh, "Automatic Reusability Appraisal of Software Components using Neuro-Fuzzy Approach", International Journal Of Information Technology, vol. 3, no. 3, 2006, pp. 209-214.
8. Parvinder S. Sandhu Pavel Blecharz and Hardeep Singh, "A Taguchi Approach to Investigate Impact of Factors for Reusability of Software Components", World Academy of Science, Engineering and Technology, pp.135-140, Sep 2007.
9. 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
10. Gui Gui and Paul D. Scott, "New coupling and cohesion Metrics for Evaluation of Software Component Reusability", in Proc. ICYCS, 2008, pp.1181-1186.
11. Parvinder S. Sandhu, Harpreet Kaur and Amanpreet Singh, "Modeling of Reusability of Object Oriented Software System", World Academy of Science, Engineering and Technology Issue. 30, pp. 162-165, Aug 2009.
12. 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.
13. Sonia Manhas, Rajeev Vashisht, Parvinder S. Sandhu and Nirvair Neeru," Reusability Evaluation Model for Procedure Based Software Systems", International Journal of Computer and Electrical Engineering, Vol.2, No.6, pp. 1107-1110, Dec 2010.
14. 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.
15. Fazal-e-Amin, Ahmad Kamil Mahmood and Alan Oxley, "Reusability Assessment of Open Source Components for Software Product Lines", International Journal on New Computer Architectures and Their Applications (IJNCAA), 1(3), pp. 519-533, 2011.
16. 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
17. Nasib S. Gill and Sunil Sikka, "Inheritance Hierarchy Based Reuse & Reusability Metrics in OOSD", International Journal on Computer Science and Engineering (IJCSE), Vol. 3 No. 6, pp. 2300-2309, June 2011.
18. Ajay Kumar, "Measuring Software Reusability using SVM based Classifier Approach", International Journal of Information Technology and Knowledge Management.", Vol. 5, No. 1, pp.205-209, Jan 2012.
19. 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
20. Neha Goyal, Er. Deepali Gupta, "Reusability Calculation of Object Oriented Software Model by Analyzing CK Metric", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) , ISSN: 2278 – 1323 Volume 3 Issue 7, July 2014
21. P K Bhatia, "Research on software reuse methods based on the object-oriented components", Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference.", Changchun Vol. 5, No. 1, pp. 1857 – 1860 , 29-31 Dec. 2012.IEEE.
22. Peter Freeman, 1983. Reusable software engineering concepts and research directions. In Tutorial: Software Reusability, pp:10-23.
23. 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
24. Victor, R. Basili and Hans Dieter Rombach, 1988. Towards a comprehensive framework for reuse: A reuse-enabling software evolution environment. Technical Report CS-TR-2158, University of Maryland, December 1988.

25. Jack Cooper, 1994. Reuse-the business implications. In Marciniak [Mar94], pp: 1071-1077.

26. ISO, "ISO/IEC 9126-1: Software Engg.-Product Quality-Part-1: Quality Model", Geneva, Switzerland, 2001

27. 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.

28. McCall JA, Richards PK, Walters GF. Factors in software quality, RADC TR-77-369: 1977. (Rome: Rome Air Development Center)

29. 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

30. Dromey RG. Concerning the Chimera (software quality). IEEE Software. 1996;1:33-43.

31. Boehm BW, Brow JR, Lipow M, McLeod G, Merritt M. Characteristics of software quality. North Holland Publishing. Amsterdam, the Netherlands; 1978.

32. Grady, RB. Practical software metrics for project management and process improvement, Prentice Hall; 1992.

33. Khan R A & Mustafa K, A Model for Object Oriented Design Quality Assessment Proceedings, Integrated Design And Process Technology Symposium, , Kusadasi, Izmir, Turkey, June 28-July 2, 2004

