International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-4, November 2019

Maintainability Assessment for Object Oriented Software System

Amit Sharma, Deepak Kumar Singh, Prabhat Kumar Vishwakarma

Abstract: ***In modern era, Maintainability of object oriented software system will be report to 70-75% for all these efforts extended by the resource and efforts used for the design phase in development life cycle. To make better or maintain the maintenance, the analyst design this phase early. For designing the software system the system furter classified into the number of object metrics and the maintenance of the object oriented system having fewer changes required for the sub system. In this work, the maintainability metrics are used to calculate or measure the localized location that are being utilized in designing.***

Keyword : object oriented, maintainability, software.

I. INTRODUCTION

Software development life cycle basically contains the four phases- system requirement and analysis, design, system implementation and its maintenance. In designing and object oriented system is defined into multiple systems and this system further classified into various systems. Under the section of implementation, lots of changes required for the system to be operational correctly and enlarge the same from 60 to 80% for all the programming works under the maintenance phase [pressman]. The promote or given higher rank maintenance cost has a major issue and concern for the developers or users in designing of an object oriented software system. To enhance this condition, the analyst must maintain the system to reduce about this situation. According to Rambagh, the maintainability of object oriented software system that totally depends on its design [1]. Maintainability matrix is needed to maintain the effect of alter and to reduce the high cost of the maintenance.

Manuscript published on November 30, 2019.
* Correspondence Author
Amit Sharma*, Ph.D. Scholar, Dr. APJ AKTU, Lucknow, UP, India. Email: amit.krsharma123@gmail.com
Dr. Deepak Kumar Singh, Director, Sachdeva Institute of Technology, Mathura, UP, India. Email: yadav.k.deepak@gmail.com
Dr. Prabhat kumar Vishwakarma, Professor, Department of CSE, IINTM, Janakpuri, Delhi, India. Email:prapoo2012@gmail.com

As per IEEE, "Maintenance means the modification of software product after development after the deployment for the correction of faults and to improve the performance or other attributes to adjust to a changed domain". Maintenance almost taken 60 to 80% of the software development cost [2]. This paper includes as –
Section 2: related work and literature survey off the research.
Section 3: describe the model for object oriented software model.
Section 4: define the metrics i.e. maintainability or object oriented in view of design level.
Section 5: Conclusion and its Future work.

II. LITERATURE REVIEW:

Although there have been number of paper address about the object oriented software system maintenance problem. Different techniques are used for guess the maintainability to reduce the effort and the developer cost on some basis of criteria and measurement. Bsili et. Al. was defined the special part of maintenance i.e fault proneness and detection[3]. Some model for accessing effort was also be presented to estimate the maintenance of object oriented system[4].Aggarwal . k, et al. proposed some fuzzy based methodology that described the 81 rules and validate these by multiple projects[5]. Muthana S. et. Al.in (2000) describe the prediction model for the estimation of maintainability of object oriented software system and also describe some model that define the impact rate and the error rate [6]. Kiewkanya M. et al. in (2004) defines that object oriented system for the comfort of maintenance for the best understanding of two factors i.e. modifiability and understandability. Also describe the three techniques are discriminant technique, weighted score level technique and weighted predicate level technology. Rizvi A. et al. in (2010) describe the MEMOOD model that describe the opportunity to improve the class level diagram [7]. Gautam C. et al. in (2011) characterize the MEMOOD model to determine the software maintainability in terms of complexity and the degree of complexity[8]. Abreu et al. describe the two vectors these are granularity and category. Also describe the categories of object oriented metrics like, complexity, design, size, quality, reuse and productivity [9]. M.Alshayeb et al. describe the iterative procedure of object oriented metrics. Also describe the afile process and long cycled framework and then the result would be effort and the source code that will be created, added, deleted and changed. The development of software system cannot be done for the long term iteration [10]. R.D.Neal et al. defines that some metrics are not validated and these metrics need not to be used for measurement [11].

R. Harrison et al. describe the tactful model that having some distinguishing limit esteems helps to the Chidamber and Kemerer measurements. This procedure is mostly for the eclipse version for the object oriented software system. This distinguished have mostly gives the hazards levels [12].L.H. Ethzkorn describe the threshold values that are too much precise about the multiple distribution parameters. It also describe the high level object oriented quality models [13].
H. Lieu. Et al. describes the gap between the quality measurement of object oriented software system at the time of development phase and the change in the development i.e. reuse of the existing software like regression [14]. M. Subramanyam et al. describes metrics design that can help to describe that how we can improve the software quality to enhance the feature of the system [15]. Rachel Harrison et al. proposed some properties of metrics object oriented design (MOOD) and showed that these are used to calculate the complete assessment for the software system [16].Goldberg et al. proposed the linear model where the variables are using conventional and functional based object oriented software system [17]. C. Shyam et al.suggest to improve and calculate the modularization quality for the software system that provide the higher degree of dependencies for the good quality of the modularisation [18].

III. OBJECT ORIENTED SOFTWARE SYSTEM MODEL:

It is proposed by Coad and Yourdon as a object-oriented software requirement analysis and design model [19] and can be confluence along the Courtois for the decomposition of the software system model [20]. These components are designed ans defined by the graphical model can be explained in Figure-1.

1.1 object oriented model:

Class: class is a gathering of objects and similar behavior by the set of methods.
Object: an object is an instance that can be used to save the methods and the states that can effect the state.
Method: depends on the object and declared within the class.
Attribute: describe the properties of the class.
Link: a connection between the two classes and the objects that they required as per need.
Message connection: It means a connection between the methods that has been described.
Instance connection: it means the connection between the method and object will be created at a moment as they required.
Inheritance: It means the relationship between the classes and the objects require the characteristics of each other.

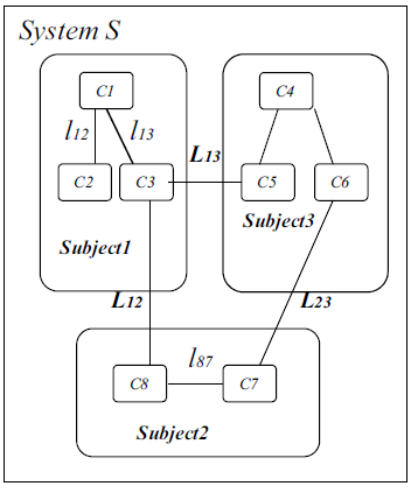

Figure-1
(object oriented decomposition)

IV. OBJECT ORIENTED AND MAINTAINABILITY METRICS

There some object oriented and maintainability metrics that can be described as-

1.2 object oriented maintainability metrics:

The six new object oriented metrics are may be discussed as: All of the six maintainability metrics may be considered as-

(i) Weight Method per Class (WMC): These metric will be considered as the total number of methods used in a class. Also contain the number of methods and also maintain the required time and effort used to preserve in class.
(ii) Response for a Class (RFC): It is combination of complexity of class and the methods that are used within the class. Also help to calculate the testability and understandability.
(iii) Lack of Cohesion of Method (LCOM): It is a collection of inter related method of class that contained the methods of input variable and attributes. It also help to calculate the reusability and the efficiency.
(iv) Depth of Inheritance Tree (DIT): It is a relationship between the class in which the operators and variable are predefined. It also help to calculate the depth of tree and the inherited properties of a class used to calculate the reusability and testability.
(v) Number of Children (NOC): It is used to calculate the number of children's used in the subclass for the reuse and also help to calculate the reusability and the testability.
(vi) SATC preferred some criteria for the object oriented metrics are-
(i) used to reduction in the complexity of an architecture.
(ii) Increase the design and testing hierarchy.
(iii) also improve the capacity of psychological complexity.

1.3 Maintainability Metric

Maintenance contained only major three characteristics. These actions are-
Action 1: to analyze and isolate of the systems affected by their change.

Retrieval Number: D5401118419/2019©BEIESP
DOI:10.35940/ijrte.D5401.118419
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Action 2: For one and all affected chunks, all actions wants inter related change.
Action 3: Integration all the system and perform the regression tests.
Maximum number of the maintainability models does not contain the activities, I and we assume the effort used and this is independent and not to be examined. Most of this action was performed in the action 2 and action 3 as they required.
To calculate the effort the general formula will be treated as-

$$Effort(Change) = Effort\ (Classes) + Effort(Integrartion)$$

Where,
Effort(change)= total effort used to implement a change,
Effort(classes)= sum of appropriate efforts used to change in all affected class,
Effort (Integration)= total efforts for integration and regression testing.
decomposition maintainability for a change *ch* (*maintainability*$_{ch}$) will be-

$$maintainability_{ch} = 1 - \frac{Effort_{ch} - IdEffort_{ch}}{WstEffort_{ch} - IdEffort_{ch}}$$

Where,
Effort$_{ch}$ = efforts used to implement the change *ch* during decomposition,
IdEffort$_{ch}$ *= efforts used to implement the same change on ideal decomposition,*
WstEffort$_{ch}$= efforts used to implement the same change on corresponding to the worst decomposition.
The above definitions of ideal decomposition towards to the worst decomposition for a change, swings to-

$$Maintainability_{Ch} = 1 - \frac{\sum_{SSj\ of\ the\ subsystem} |SSj| - r}{|S| - r}$$

where |*SSj*|= size of the sub system(Number of subclasses in a class),
|*S*| = numbers of classes in the software system, and
r = number of affected classes change.
The whole derivation can be found on the paper [10].
Maintainability range metrics will be lie between 0-1. Maintainability will be 1 if achieved and if all the *r* affected classes are only will be the one subsystem. Inversely, maintainability will be if all system and subsystem affected.
For an amount of *n* number of changes, maintainability will be-

$$Maintainability = \frac{\sum_{ch=1}^{n} Maintainability_{ch}}{n}$$

V. CONCLUSION AND FUTURE SCOPE

In this paper, an object oriented software maintainability model is described and the role of maintainability metrics are described in view for the calculation of the maintainability I terms of their good quality of metrics also helps us to calculate the maintainability of the object oriented software system and plays a vital role for the development of good quality software system.

REFERENCES:

1. Rombach Dieter., "Design Measurement: Some Lessons Learned", *IEEE Software*, March 1990, pp. 17-25.
2. IEEE, IEEE Standard: 1219-1993_IEEE Standard for Software Maintenance, INSPEC Accession No. 4493167, IEEE Computer Society, 1993.
3. Basili, V R, L C, Briand and W L, Melo , 'A validation of object-oriented design metrics as quality indicators', *IEEE Transactions on Software Engineering*, October 1996, pp751-761.
4. S. Bandini, F. D. Paoli, S. Manzoni, and P. Mereghetti, "A support system to COTS based software development for business services", In Proceedings of the 14th International Conference on Software Engineering and Know ledge Engineering, Vol. 27, pp. 307–314, 2002.
5. K. K. Aggarwal, Y. Singh, P. Chandra, and M. Puri, "Measurement of Software Maintainability Using a Fuzzy Model", Journal of Computer Sciences, Vol. 1(4), pp. 538-542, 2005.
6. Muthanna S., Kontigiannis K., Ponnambalam K. and Stacey B.,"A Maintainability Model for industrial Software System Using DesignLevel Metrics", IEEE Computer Society, 2000, pp 248-256.
7. Rizvi S.W.A. and Khan R.A., "Maintainability Estimation Model for Object-Oriented Software in Design Phase (MEMOOD)",Journal of Computing, Volume 2, Issue 4, April 2010,
8. Gautam C., kang S.S., "Comparison and Implementation of Compound MEMOOD MODEL and MEMOOD MODEL", International journal of computer science and information technologies, pp 2394-2398, 2011.
9. B. F. Abreu: "Design metrics for OO software system", ECOOP 95, Quantitative Methods Workshop, 1995.
10. M. Alshayeb and Li.W.,"An empirical validation of object-oriented metrics in two different iteration software processes", IEEE transactionod Software Engineering, Vol-29, no-.11, Nov 2003.
11. R. D. Neal, "The Measurement Theory Validation of Proposed Object-Oriented Software Metrics", Dissertation, Virginia Commonwealth University, (1996).
12. R. Harrison, Samaraweera, L.G. Dobie and Lewis, P.H: Comparing Programming Paradigms: An Evaluation of Functional and Object-Oriented Programs, Software Eng. J., vol. 11, pp. 247-254, July 1996.
13. H. Etzkorn, W. E. Hughes, C. G. Davis, "Automated Reusability Quality Analysis of OO Legacy Software", Information and Software Technology, 43, (5), (2001), 295–308
14. H.Lilu, K.Zhou and S.Yang: "Quality metrics of OOD for Software development and Re-development", First Asia-Pacific Conference on Quality Software, August 2002.
15. M.Subramanyam and R.Krishnan: "Emphirical Analysis of CK metrics for OOD complexity:Implication for software defect", IEEE transaction on software engineering, 2003.
16. R.Harrison, S.J.Counsell and R.V.Nithi: "An evaluation of the MOOD set of OOSM", IEEE Transaction on Software Engineering, vol.24 no.6 , pp.491-496, June 1998. JürgenWüst, "SD METRICS TOOL", in der Lache 17, 67308.
17. A. Goldberg, Robson, D., "Smalltalk-80: the language and its implementation", Reading, MA: Addison Wesley, 1983.
18. A. Shyam and C. F. Kemerer, "Towards a Metrics Suite for Object Oriented Design", Proceeding on Object Oriented Programming Systems, Languages and Applications Conference (OOPSLA"91), ACM, Vol. 26, Issue 11, Nov 1991, pp. 197-211.
19. Coad, P.; Yourdon, E.: "Object-Oriented Analysis.", *Prentice Hall*, 1991.
20. Kiran, G.A., Haripriya, Jalote, P. "Effect of Object Orientation on Maintainability of Software", *In ICSM97*, Bari, Italy,, October 1998, pp. 114-121.
21. Tagoug, N.,"Information Systems Decomposition: Exploratory Study of Influencing Factors", *PhD Dissertation,University of Montreal, Montreal, Canada, 1998, 247p.*