IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

MAINTAINABILITY MODEL WITH CLASS INHERITANCE HIERARCHIES

[1] Hanumantha Rao Battu, [2] Satya Prasad Raavi

[1] Research Scholar, Department of CSE,
Acharya Nagarjuna University, Guntur, India
[2] Professor, Department of CSE,
Acharya Nagarjuna University, Guntur, India

Abstract: Measurement is the basic fundamental thing not only for software engineering but also for any engineering discipline in the real world nature. Measuring the software maintainability in appropriate way improves the quality of the software system in significant manner. Previously so many researchers worked on the object-oriented software system maintainability parameter to find the proper measures in terms of various estimated models. The class inheritance hierarchy of the object- oriented system plays major role in measuring the maintainability.

The main object of this research paper is to find out the estimated model for object-oriented software maintainability. Frederick T. Sheldon et.al contributed work regarding the metrics for maintainability has been extended towards maintainability model measurement in good manner. The researcher used two metrics for maintainability model are Understandability and Modifiability, which are the most important factors of the maintainability. This research may be useful for future researchers in terms of measuring the maintainability model with the help of metrics instead of the models of sub factors. The maintainability model establishment with metrics is the new research proposal which does not exist in the previous researchers' work.

Index Terms **- Object oriented, software Maintainability, OO Software Metrics, Maintainability model, Understandability metrics, Modifiability metrics, class inheritance.**

I. INTRODUCTION

The poor management in the software [1] occurs in any software system. Because there is no proven measures for the maintainability parameter. The maintenance phase of the software development life cycle (SDLC) carries the 50-70% portion among the total work [2], [3]. Most of the researchers measured the Object-Oriented (OO) software system maintainability because the majority of the software applications follow the OO nature. Measurement is the basic concept for any type of discipline existed in the real world not only in the software system. The maintainability is one of the important parameter of the software system quality [4].Measuring the maintainability leads to improve the quality of the software system.

The maintainability of the software system can easily adapt to the changed environment, to improve the attributes like performance, modified to correct faults of any component or software system [5]. Several maintainability studies performed by different authors [6], [7] to identify the importance of the maintainability in software systems. The class inheritance hierarchy is used to measure the maintainability in good manner. The maintainability depends on the several factors [8] like understandability, analyzability, testability, modifiability ….etc. understandability and modifiability are the most important factors which plays crucial role among the many number of maintainability dependent factors in judging the maintainability of the software system. The measurements of the understandability and modifiability factors are done mostly with estimated models, but Frederick T. Sheldon et.al. [9] developed metrics for the individual factors. The model estimation with metrics obtain accurate measure rather than with models of the factors. So, the researchers' main focus is to propose the maintainability estimated model accurately with the help of metrics of understandability and modifiability metrics.

The structure of the paper is in the following manner. The literature survey regarding various estimated maintainability models by different authors are placed in Section-II. Section-III deals with the appropriate metric selection for the maintainability model prediction. The developed data table with the values of maintainability, modifiability and understandability are placed in Section-IV. Section-V shows the estimated maintainability model and its statistical significance. The results and proposed maintainability model validation is given in Section-VI. Section-VII and Section-VIII illustrates the conclusion and the future scope of this research work.

II. LITERATURE SURVEY

The ultimate goal of any researcher to measure the software maintainability is to release the more qualitative software systems in future. Due to non functional behavior of the software quality, quality parameters and those parameter's factors also measured in terms of various models rather than metrics. The model development gives absolute but not accurate measures towards parameter or factors measurements. The main intention of this research is to develop the maintainability parameter model with developed metrics of understandability and modifiability factors.

In the maintainability metrics measurement chidember-kerner developed CK metrics [10],[11] namely Depth of Inheritance (DIT) and Number of Childs(NOC) based on the inheritance property of the OO class diagram hierarchies. Bailet al.[12] and Tang et al.[13] conducted various studies to validate the CK metrics and identified those metrics as quality indicators in OO software systems. Balasubramanian[14],Li[15] validated CK metrics and identified some deficiencies in the approach of CK metrics and proposed some new metrics for OO system maintainability also. Li[15] proposed two new metrics based on the class inheritance property namely Number of Ancestor Classes(NAC) and Number of Descendant Classes(NDC). Lake and cook [16] also developed Number of Parents(NOP) and Number of Descendants(NOD) metrics to measure the OO software maintainability in appropriate manner. Henderson-Sellers[17] proposed new maintainability metric named Average Depth of Inheritance(AID) is the average complexity of DIT metric value. The developed maintainability metrics only considered the inheritance property of the OO software system, but the maintainability depends on the several factors like Understandability, Analyzability, Modifiability …etc . In this connection Frederick T. Sheldon et.al. [9] proposed two metrics named one as Average Understandability (AU) and another one is Average Modifiability (AM) to give more support to the OO software maintainability parameter.

Hayes and Zhao[18] developed two levels of maintainability to the estimated models of OO software maintainability, one is easy to maintain and another one is not easy to maintain. Hayes and Zhao[18] focused on two levels of maintainability only and not considered the factors of maintainability parameter. Kiewkanya et al.[19] developed the models for understandability and modifiability factors of maintainability by including huge amount of metrics in every model prediction. Genero et al.[20] developed data set and different estimated models[21],[22],[23] for the maintainability depended factors i.e., Understandability, Analyzability and Modifiability based on the OO software system properties. Nazir et. al [24] and Rizvi and Khan[25] developed maintainability and its sub factors models with inclusion of the more number of metrics in every model. Syma Kumar et al.[26],[27],[28],[29] proposed various models for maintainability and its factors with less number of metrics involved in every model.

The above said all the maintainability models are developed based on the estimated models of its factors not with the metrics of the factors. If any model is developed with another models rather than metrics may produce the inaccurate results. In the process of producing more accurate results of maintainability models there is a need to include the metrics in the models estimation not the other estimated models.

III. METRICS SELECTION

Many of the applications in industrial and our regular life also follow the hierarchy feature. This hierarchy follows in the OO software systems as the class inheritance hierarchy levels not only in the banking sector and student management system but also more number of applications. Daly [30] stated that the modification of any system with three levels of inheritance is very much easier than the system with no inheritance. When the user is able to understand in better manner it is easy to do the modification of the system. Hence the inheritance levels from 1 to 3 will helps the user in understanding the given OO system in appropriate manner and that may easy to apply the modifications of the system.

The directed acyclic graph (DAG) with no loops [31] based OO class diagram is used in the development of maintainability metrics by Frederick T. Sheldon et.al. [9].The two important factors of maintainability i.e., understandability and modifiability are preferred in the metrics prediction. The researchers used the 3 levels of inheritance hierarchy for better maintenance of the system. The predecessors of the given class are considered for the understandability metric prediction. The successors of the given class are preferred in the modifiability metric prediction. The proposed metrics are evaluated with the well known weyker's properties [32] also. The below given types class diagrams are used in their research.

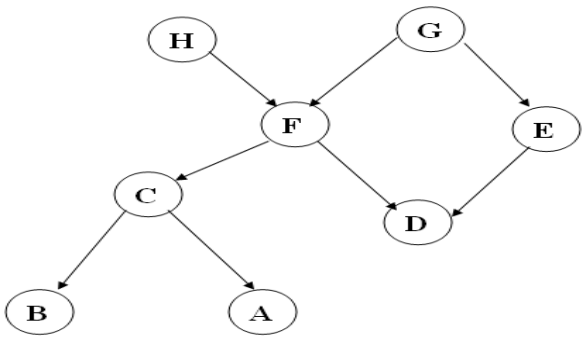

figure: class inheritance hierarchy

IV. MAINTAINABILITY DATASET

In the maintainability data set estimation process the researchers used the developed metrics of Frederick T. Sheldon et.al. [9]. The researchers considered more than 25 class based inheritance hierarchy diagrams and apply the understandability and modifiability metrics on the above stated diagrams individually. The used 25 class diagrams are similar to figure-1 and followed the inheritance levels from 1 to 3.

As per Daly's experiment [30] if any modifications on the given system can be applied easily for the 3 level inheritance hierarchy system rather than no inheritance system, which means maintainability of the inheritance based system is easier than no inheritance based system. The used 25 diagrams in this research also contain the level inheritances from 1 to 3. In this research work maintainability levels are divided into three types i.e., first one is low, second one is medium and last one is high level. Here the low level of maintainability had applied for the 3 levels of inheritance hierarchy based class diagrams. The medium level applied for the 2 levels of and high level applied for the single level based OO class based inheritance hierarchy diagrams. The calculated understandability, modifiability factor values and maintainability levels are shown in the following Table-1. In table-1 number of classes taken for the individual class diagram also specified individually.

Table-1: maintainability dataset

Class diagram	Number of classes	Understandability	Modifiability	Maintainability
1	2	1	1.12	1
2	3	2	2.5	2
3	3	1.66	2	1
4	4	2.5	3.31	3
5	4	3.5	4	1
6	4	2.25	2.88	2
7	4	2.25	2.88	2
8	5	2.25	2.65	1
9	5	2.2	2.8	2
10	5	2	2.5	2
11	5	2.4	3.1	2
12	5	2.6	3.4	2
13	5	2.2	2.8	2
14	5	2	2.6	2
15	5	2.6	3.4	2
16	5	2.6	3.6	2
17	5	2.2	2.8	2
18	5	2.8	3.7	3
19	5	2.6	3.4	3
20	5	2.4	3.1	2
21	7	2.43	3.14	2
22	6	2.33	3	2
23	6	2.5	3.25	2
24	7	2.43	3.14	2
25	8	3.25	4.38	3

V. MAINTAINABILITY MODEL ESTIMATION

The regression mechanism [33] gives the average relationship between the two or more variables. In this research maintainability parameter (variable) depends on the understandability and modifiability factors (variables). In this regression process SPSS package is used and applied the linear regression technique. Multivariate regression is used to estimate the maintainability model among the two types of linear regression.

The multivariate linear regression is as following.

$$A = 1+a*B1+b*B2+c*B3+d*B4+---------- \rightarrow (1)$$

In the above equation A is the dependant variable where as B1,B2,B3,B4…etc are the independent variables. a, b, c,d ..etc are the coefficient values of the independent variables. The Table-2 showing understandability and modifiability metric values as independent variables and maintainability levels are taken as dependant variable for the regression process. In the estimation of maintainability model 18 (72%) samples used for the model estimation and the remaining 7(28%) samples are utilized for validation purpose of all the total 25 class diagrams.

Table-2: Statistical Significance of Maintainability Model

	B	S.Error	t	Sig.
Constant	0.810	0.637	1.271	0.223
U	-2.000	1.265	-1.582	0.135
M	1.931	0.822	2.348	0.033

The statistical significance of the maintainability models are identified for the prediction of the maintainability model and also the constant, coefficients of the understandability (U) and modifiability (M) independent variables are shown in the table-2 .The Maintainability had denoted as Ma for model equation purpose. Based on the statistical significance the proposed maintainability model is in the following form of equation -2.

$$Ma = 0.810 - 2.0 * U + 1.931 * M \quad \text{---} \rightarrow (2)$$

The table-2 indicates variables and other information of the model are statistically significant with the 95% level of confidence.

To give more support to the proposed maintainability model Anova values are also calculated and placed in Table-3.

Table-3: Anova for Maintainability Model

	Sum of Squares	df	Mean Square	F	Sig.
Regression	1.933	2	0.967	14.335	.000
Residual	1.011	15	0.067		
Total	2.944	17			

The two-way Anova is used in the model estimation of the multivariate linear regression. Anova table gives the depth of freedom (df) and other useful information with 0.01 level of significance.

VI. RESULTS AND VALIDATION OF THE PROPOSED MODEL

The calculated values of maintainability(C-Ma) for the proposed maintainability model validation are given as the results in Table-4. The Actual maintainability (A-Ma) values are also placed in the same table. For this validation and comparing purpose 28%(7) samples of Table-1 data set is used which does not utilized in the prediction of maintainability model.

Table-4: Comparison between actual and calculated Maintainability

	Class diagrams						
	1	2	3	4	5	6	7
A-Ma	1	1	2	2	3	3	3
C-Ma	0.97	1.53	1.64	2.18	2.18	2.2	2.35

The comparison between the actual and calculated maintainability values are placed in the above table. The resultant calculated values are very much nearer to the actual values of the maintainability. Pearson's 2-tailed correlation is performed between the actual and calculated values of maintainability and the resultant correlation values are placed in table-5.

Table-5: Pearson Correlation between the actual and proposed values of Maintainability

	A-Ma	C-Ma
A-Ma	1	0.87
C-Ma	0.87	1

The correlation values of actual and calculated values of maintainability are taken with 0.01 level of significance.

VII. CONCLUSION

The main aim behind this research work is to propose the new maintainability model with the utilization of Frederick T. Sheldon et.al, developed metrics for understandability and modifiability factors of maintainability parameter. The previous maintainability models are based on the estimated models of understandability and modifiability not the metric values of them. In the prediction process of the maintainability the values of the understandability and modifiability are calculated for the 25 class diagram based inheritance hierarchy samples. The three levels of maintainability also identified and used in the model estimation for those 25 class diagrams. Based on the proposed maintainability model the maintainability values are calculate and compared with the 7 samples of actual maintainability values. The resultant values are nearer to the actual values and shows the good correlation values between them also.

VIII. FUTURE SCOPE

The main motto of this research work is to estimate the maintainability model with the metric values rather than models. It is the first step of calculating the maintainability model with the help of metric values. In future there is a rigorous research is required in the number of inheritance levels to maintain the OO software system in proper manner. The main aim researcher is to develop the more accurate understandability and modifiability metrics by applying much observation on the inheritance hierarchy based class diagrams instead of Frederick T. Sheldon et.al, developed metrics in future. The inclusion of careful observation in the metric calculation of the maintainability parameter factors leads to more accurate calculated values of maintainability.

REFERENCES

[1] M.Dagpinar and J.Jahnke Predicting Maintainability with Object-Oriented metrics –an Empirical Comparison,Proc. 10th Working Conference on Reverse Engineering (WCRE‘03), 13-17 Nov.2003, pp155-164, 2003.

[2] G. Parikh and N. Zvegintzov, Tutorial on Software Maintenance, IEEE CS Press, Los Alamitos, Calif., Order No.453, 1983.

[3] I. Sommerville, Software Engineering, Addison Wesley, 1996.

[4] ISO/IEC 9126. (1991) “Information technology - software product evaluation- quality characteristics and guidelines for their use”.

[5] IEEE (1990). “IEEE Standard Glossary ofSoftware Engineering Terminology,” IEEE Std. 610.12-1990. Institute of Electrical and Electronics Engineers.

[6] P. Antonellis, D. Antoniou, Y. Kanellopoulos, C. Makris, E. Theodoridis,C. Tjortjis, and N. Tsirakis, “A Data Mining Methodology for Evaluating Maintainability According to ISO/IEC-9126 Software Engineering Product Quality Standard,” Proc. 11th IEEE Conference on Software Maintenance and Reengineering (CSMR2007), 21 – 23 Mar.2007, Amsterdam, Netherlands, 2007.

[7] Ajay Rana , Soumi Ghosh and S K Dubey, “Comparative Study of Factors that Affects Maintainability, International Journal on Computer Science and Engineering”, Vol 3 (12), December 2011

[8] ISO/IEC 9126-1 (2001) Institute of Electrical and Electronics Engineers, Part1: Quality Model.

[9] Sheldon F.T.—Jerath K.—Chung H.(2002): Metrics for Maintainability of Class In-heritance Hierarchies. Journal of Software Maintenance 14, pp. 147–160.

[10]Chidamber, S.R.—Kemerer, C.F.: To wards A Metrics Suite for Object Oriented Design,OOPSLA’91,pp. 197-211,1991.

[11]Chidamber, S.R.—Kemerer, C.F.: A Metrics Suite for Object Oriented Design.IEEE Transactions on Software Engineering, Vol. 20, 1994, No. 6, pp. 476–493.

[12]Basili VR, Briand LC, Melo WL. A validation of object-oriented design metrics as quality indicators. Technical Report, University of Maryland, Department of Computer Science, College Park MD, 1995; 1–24. 11.

[13]Tang MH, Kao MH, Chen MH. An empirical study on object-oriented metrics. Proceedings 23rd Annual International Computer Software and Application Conference. IEEE Computer Society Press: Los Alamitos CA, 1999; 242–249.

[14]Balasubramanian NV. Object oriented metrics. Proceedings 3rd Asia–Pacific Software Engineering Conference (APSEC’96). IEEE Computer Society Press: Los Alamitos CA, 1996; 30–34.

[15] Li W. Another metric suite for object-oriented programming. The Journal of Systems and Software 1998; 44(2):155–162.

[16]Lake A, Cook C. Use of factor analysis to develop OOP software complexity metrics. Proceedings of the Annual Oregon Workshop on Software Metrics. Oregon Center for Advanced Technology: Portland OR, 1994. http://www.ocate.edu/ [8 December 2001].

[17]Henderson-Sellers B. Object Oriented Metrics: Measures of Complexity. Prentice Hall PTR: Englewood Cliffs NJ, 1996; 130–132

[18]Hayes J.H. and Zhao L. ―Maintainability Prediction: a Regression Analysis of Measures of Evolving Systems, Proc. 21st IEEE International Conference on Software Maintenance, 26 - 29 Sept. 2005, pp. 601 – 604, 2005.

[19]M. Kiewkanya, N. Jindasawat, and P. Muenchaisri, “A Methodology for Constructing Maintainability Model of Object-Oriented Design,” Proc. 4th International Conference on Quality Software, 8 - 9 Sept., 2004,pp. 206 - 213. IEEE Computer Society, 2004

[20]Genero M., Olivas J., PiattiniM. and Romero F. ―A Controlled Experiment for Corroborating the Usefulness of Class Diagram Metrics at the Early Phases of Object - Oriented Developments, Proc. of the ADIS2001, Workshop on Decision Support in Software Engineering, vol. 84, Spain,2001.

[21] Genero M.,Manso M.E., Piattini M. and Cantone,G. ―Building UML Class Diagram Maintainability Prediction Models Based on Early Metrics‖, 9 th International Symposium on Software Metrics (Metrics 2003), Proceedings IEEE Computer Society, pp. 263-275, 2003.

[22]Genero M., Olivas J., Romero F. and Piattini M. ―Assessing OO Conceptual Models Maintainabilit, In Olive et al. (Eds.), International Workshop on Conceptual Modeling Quality (IWCMQ‘02), Tampere, Finland, LNCS vol.2784, Springer-Verlag, pp. 288-299, 2003.

[23]Genero M., Manso E., Visaggio A., and Piattini M. ―Building Measure- Based Prediction Models for UML Class Diagram Maintainability, Journal of Empirical Software Engineering, vol. 12, no. 5, pp. 517 – 549, 2007

[24]Nazir M., Raees A. Khan and Mustafa K , ―A Metrics Based Model for Understandability Quantification ― JOURNAL OF COMPUTING, VOLUME 2, ISSUE 4, ISSN 2151-9617,2010

[25]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, ISSN 2151-9617 , 2010.

[26]D.N.V. Syma Kumar, R. Satya Prasad and Dr.R.R.L.Kantam “Maintainability of Object-Oriented Software Metrics Using Non-Linear Model “International Journal of Advanced Research in Computer Science Engineering and Information Technology Volume: 5 Issue: 3 20-Mar-2015,ISSN_NO: 2321-3337.

[27]D.N.V. Syma Kumar & R. Satya Prasad ―Maintainability Model with Object-Oriented Software Metrics International Journal of Engineering & Technology IJET-IJENS Vol:15 No:04 , August 2015.

[28] R.Satya Prasad and D.N.V.Syma Kumar ―Mainatanability of Object-Oriented Software Metrics with analyzability International Journal of Computer Science issues, Volume12,Issue3, May 2015.

[29]Dr.D.N.V.Syma Kumar Common Metrics for Object-Oriented Software Maintainability Model -in International journal of Scientific Technology & Research, ISSN :2277-8616, Vol.13 , No.10, October 2019

[30]Daly J, Brooks A, Miller J, Roper M, Wood M. Evaluating inheritance depth on the maintainability of object-oriented software. Empirical Software Engineering 1996; 1(2):109–132.

[31]Wang CC, Shih TK, Pai WC. An automatic approach to object-oriented software testing and metrics for C++ inheritance hierarchies. Proceedings International Conference on Automated Software Engineering (ASE'97). IEEE Computer Society Press: Los Alamitos CA, 1997; 934–938.

[32]Weyuker, E. J.: Evaluating Software Complexity Measures. IEEE Transactions on Software Engineering, Vol. 14, 1988, No. 9, pp. 1357–1365.

[33]Varalakshmi V., Suseela N., Gnana Sundar G., Ezhilarrasi S., Indrani B. (2004), Statistics Higher Secondary, Tamilnadu textbook Corporation, First Edition