A Maintainability Estimation Model and Metrics for Object-Oriented Design (MOOD)

Kiranjit Kaur
M.tech, research work, Software Engineering.

Sami Anand Asst. Prof.(L.P.U.)
M.tech ,member of IEEE&IEEE Computer Society,
research work in area of Network Security and Software Engineering..

Abstract— ***Maintainability is that phase which may help the designer for improving the quality of the software system before deliver to a customer. This paper uses a multivariate linear model, which estimates the maintainability of a class diagram in the term of reliability, portability that are the sub-characteristics of maintainability which are use to evaluate maintainability estimation model of class diagram and create the Maintainability Estimation Metrics for a maintainability estimation of class diagram. These metrics help a software designer for the purpose of improving the maintainability of a class diagram in the design phase, which are helpful in feature to reduce the increasing high cost of software maintenance phase***

Index Terms — **UML Class Diagram, Software Maintainability, Software Maintainability Model, Software Metrics.**

I .INTRODUCTION

A good development method is one that requires less effort to maintain the software. It reduces the increasing high cost of software maintenance phase. Maintainability of any software in design phase may be helping a software designer to improve the maintainability of software before deliver to a customer [1], [2]. The maintainability is defined by "IEEE standard glossary of Software Engineering as "the ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment". Software maintains is one of the important part that are expensive in cost and challenging in nature, which it is not properly managed and often ignored [3].

UML (Unified Modeling Language) has been proposed as a standard language for expressing object-oriented software designs which used in the development of any software system.UML provides the structural and behavioral aspects of software systems. Class diagram is one of the important parts of any software which play an important role in the design phase of object-oriented software early estimation of their maintainability may help designers to incorporate required enhancements and corrections for the purpose of improve the maintainability of the final software to be delivered in future [4].

11.RELATED WORK

"Maintainability Estimation Model for Object-Oriented software in design phase (MEMOOD)" estimation the maintainability of UML class diagram in term of understandability and modifiability and developed a multivariate linear model [4]. Object –Oriented process are used as a solution to software development problems. Object –Oriented development use to reduce the maintenance effort that not based on reliable experimentation [5]. Presents a multivariate linear regression for establish the maintainability estimation model and develop the Maintainability Estimation Tool for a maintainability estimation of class diagram. It takes two quality factor flexibility and *extendibility* [*6]*. It effort to maintain a software system is related to technical quality of the source code of that system. It defines the Maintainability Index limitations and problems [7]. The Halstead complexity is used for measuring maintainability. It shows the results that confirmed partially our assumptions that need to be evaluated with future uses [8]. The types of models are used that give us a vocabulary and a tool that allow us to discuss how to maintain software so as not to make it deteriorate. Verifying and valid verification measurements are used [9]. Study on the empirical evidence using some object –oriented metrics that can effectively predict maintainability of software systems. These metrics are such as size, inheritance, cohesion and coupling [10]. It presented a concern-oriented framework which supports the instantiation and comparison of concern measures. In this paper there is a rich body of ideas regarding the way to address concern measurement [11]. When more and more attentions are focused on the quality of the software, it's reasonable to believe that the *software* complexity metrics will be sit on its right place that is the main purpose of a survey on metrics of software complexity [12]. Measure the software metrics and Reliability that try to define how software is reliable and easy to maintain, which free from errors, faults and failure [13].

III. PROPESAL WORK

Scope of study: A new Extended Maintenance Model is to be design which calculates Maintainability of UML class diagrams of Object Oriented Software in the terms of reliability and portability based upon MOOD metrics. On the basis of following factors we will create our Extended Maintenance Model. These new added factors can decrease the maintain cost of the software project. In portability we can reuse the software component. As we know that the software project takes more cost for debugging as compare to new building of project So if we use reusability factor in maintenance model. We can replace the component which contains bugs instead of debugging it. It may reduce the maintenance cost of our project. The other factor is reliability. It increases the reliability of software project during its maintenance. It may increase the accuracy, completeness and robustness of our software project.

Portability: The effort required to transfer a program from one environment to another. To executable version of software in a new environment is a act of porting. It's like software component but in portability the component change its environment. A software system is a collection of software units. We use the term software unit to indicate an application program, a system program, or a component of a program. The term environment refers to the complete range of elements in an installation that interact with the ported software. This includes a processor and operating system and it may also involve I/O devices, networks. It is important to remember that porting is an alternative to redevelopment. One alternative starts with an existing implementation, the other starts with a specification. When a new implementation is the goal, a critical choice must be made as to which approach will be most cost effective. The portability of a software unit is only meaningful with respect to one or more environments. We are particularly interested in portability for a class of environments, since we do not expect to know all likely target environments when the software is developed.

Reliability: The system's ability not to fail/ the extent to which the system fails. It's is most commonly measured by the Mean Time between Failures. Changes that reduce reliability should be avoided. The software verification process aims to detect and remove defects and make it more reliable. Changes that introduce defects into the software make it less reliable. Tests should be used to verify that no defects have been introduced by the change after it has been implemented. Reliability can be reduced by reusing components that have not been developed to the same standards as the rest of the software. The effect of a modification on software reliability can be estimated indirectly by measuring its effect on the complexity of the software. This effect can be measured when the change is designed.

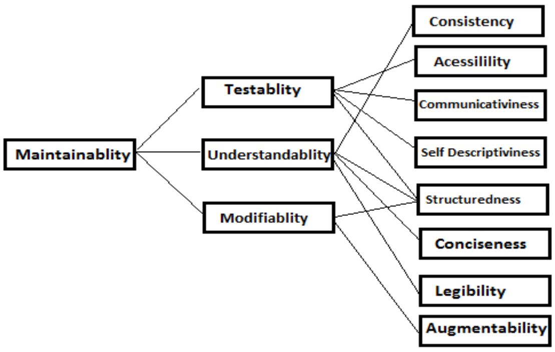

Fig 1: Existing Maintainability Model

Methodology: In the following, definitions will be given of a few fundamental concepts discussed in this paper. While the terms below will be used throughout the report, it must be understood that we can decrease the maintenance cost by adding reliability and portability in to maintenance model. Maintainability is a effort required to locate and fix a fault in the program within its operating environment. In the purposed methodology we add two new factors in to maintainability model. These two factors are portability and reliability. These factors may reduce the maintenance cost of the software project. Portability provides Device Independence and self contentedness to the software components and reliability provide self contentedness, accuracy, completeness and robustness.

In the purposed methodology we are going to add two new factors in to Maintenance model which reduces the maintenance cost of software project. We are going to develop a tool in MATLAB GUI which shows the maintainability process through different software components. This GUI will contain different components of software in which some components contain bugs. The working of GUI shows s that if we change the bugged components during maintenance then we can reduce the cost of maintenance because in software engineering we know that the cost of debugging is more then to create a new project. Now the reliability factor gives us the reliability of new added component. It will give estimation of the reliability of new added component in the existing environment. New added factor are Reliability and Portability, which increase the maintainability of the project by using some data set. Which increase the maintainability of the project.

1. The fixed data set is taken which is having MOOD metrics values

2. The inheritance approach is been used for adding the factor of reliability and portability in MOOD metrics
3. Reliability and portability are inherited from CK metrics
4. When both the features are added into MOOD metrics, software maintained will done more accurately in efficient way
5. The enhancement is made in MOOD metrics in terms of performance after adding feature of portability and reliability

Formula used for inheritance is as:

```
        function inheritance =
likelihood2class(likelihoods)

          [sample_n,class_n] = size(likelihoods);
                               maxs =
(likelihoods==repmat(max(likelihoods,[],2),[1,class_n])
                                   );
inheritance=zeros(sample_n,1);
                  for i=1:sample_n
      inheritance(i) = find(maxs(i,:),1);
```

V.REFERENCES

[1] Oman P. and Hagemeister J.,"Metrics for assessing a software system's maintainability," Software Maintenance, 1992, pp.337-334.

[2] Kiewkanya M., Jindasawant N. and Muenchaisri P., "A methodology for constructing maintainability model of object-oriented design," Quality Software QSIC2004 Proceeding Fourth International, 2004, pp.206-213.

[3] Dagpinder Melis and Jahnke Jens H.,"Predicting Maintainability with Object-Oriented Metric- an Empirical Comparison,"Proc.10th Working Conference on Reverse Engineering (WCRE'03), 13-17Nov.2003, pp.155-164, 2003.

[4] Rizvi S.W.A.And Khan R.A., "Maintainability Estimation Model for Object-Oriented Software in Sannella Design Phase (MEMOOD),"JOURNAL OF COMPUTING, VOLUME 2, ISSUE 4, APRIL2010.

[5] Hall John M., "The Maintainability of Object-Oriented Software"..

[6] Hincheeranan Alisara and Rivepiboon Wanchai , "A Maintainability Estimation Model and Tool," International Journal of Computer and Communication Engineering, Vol.1, No.2, july2012.

[7] Heitlager Ilja, Kuipers Tobias and Visser Joost, "A Practical Model for Measuring Maintainability,"-a preliminary report.

Data set (input)

Testability
.Accessibility 2.1%
.Communicativeness 8.7%
.Self Descriptiveness 9.32%

Understandability

.Consistency 10.09%

.Conciseness4.21%

.Legibility9.27%

Modifiability

.Structrendness10.51%

.Augmentability1.04%

Portability

.Device independence7.87%

.Self containdness11.11%

Reliability

.Complentness7.34%

.Robusrnesss3.25%

.Accuracy5.21%

[8] Gentzane Aldekoa, Salvador Trujillo, Goiuria Sagardui and Oscar Diaz "Experience Measuring Maintainability in Software Product Lines," XV Jornadas de ingenieria del Software y Bases de Datos JISBD 2006.

[9] Land Rikard , " Software Deterioration and Maintainability-A Model Proposal".

[10] Dagpinar Melis and Jahnke Jens H., "Predicting Maintainability with Object-Oriented Metrics-An Empirical Comparison," proceedings of the 10th Working Conferenceon Reverse Engineering (WDRE'03)2003 IEEE

[11] Figueiredo Eduardo, Anna Claudio Sant, Garcia Alessandro, Bartolomei Thiago T., Cazzola Walter and Marchetto Alessandro "On the Maintainability of Aspect-Oriented Software: A Concern-Oriented Measurement Framework".

[12] Yu Sheng, Zhou Shijie, "A Survey on Metric of Software Complexity".

[13] Rosenberg Dr.Linda, Hammer Ted and Shaw Jack, "Software Metrics and Reliability".

[14] Sharma Aman Kumar, Kalia Arvind and Singh Hardeep, "Metrics Identification for Measuring Object Oriented Software Quality," International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, issu-5, November 2012.