

International Journal of Computer Science and Mobile Computing



A Monthly Journal of Computer Science and Information Technology

ISSN 2320-088X

IJCSMC, Vol. 3, Issue. 3, March 2014, pg.778 – 786

RESEARCH ARTICLE

A NOVEL APPROACH TO ENHANCE THE MAINTAINABILITY OF OBJECT ORIENTED SOFTWARE ENGINEERING DURING COMPONENT BASED SOFTWARE ENGINEERING

¹Hardeep Singh, ²Aseem Kumar

¹Lovely Professional University, Phagwara, Punjab, India

²Lovely Professional University, Phagwara, Punjab, India

¹Hsingh891@yahoo.ca, ²aseem.16839@lpu.co.in

ABSTRACT: *Object oriented software engineering is a software design technique that is used in software design in object oriented programming. The object oriented design is used for developing the new modules in the software systems. It is used for applying the identified requirements. In the MOOD metrics the new project is developed by using various components and these factors helps in the maintainability of new project. If we talk about CBSE, i.e. component based software engineering it allows us reusability of existing software components into new software development. Here we are going to design a automation tool using genetic algorithm which will helps us to find the compatibility of components so that developer can properly integrates them and can enhance the maintainability of software project.*

KEYWORDS: *CBSE, Testing, components, object oriented metrics*

I. INTRODUCTION

The requirements of the software increases day by day, the maintenance of the software is very necessary for developing good quality software. For this purpose, software engineering is required. To develop good quality software, a developer needs to adopt the concepts and strategies of software engineering. The most important thing in development process is requirement gathering and needs of the customer. If developer wants to develop good

quality software, then the software must fulfill these needs. To develop software many new and fast technologies are comes into picture. Object oriented design is one of the technologies. It helps in developing the new modules in the software system. The object oriented design helps to identify the various requirements. [1] The object oriented design is module based architecture. It also helps to increase the quality of the design. The object oriented design contains two types of modules, these modules are:

- 1) Static model for object oriented design
- 2) Dynamic model for object oriented design

These models are the main content in object oriented design. The state of the object and behavior of the object affects the object oriented design. The objects are use in a class diagrams

A. Object Oriented Metrics

The term metrics is frequently used to mean a set of specific measurements taken on a particular process. The object oriented metrics are used to evaluate and predict the quality of software. These metrics are used as an early indicator of externally visible attributes. The externally visible attributes could not be measures until too late in the software development process. Metrics to be a set of standards against which one can measure the effectiveness of object oriented analysis techniques in the design of a system. Object oriented metrics can be applied to analyze source code as an indicator of quality attributes. The source code could be any object oriented language. [2] On the basis of their requirements, object oriented metrics can be classified into two categories. :

- 1) Project based metrics
- 2) Design based metrics

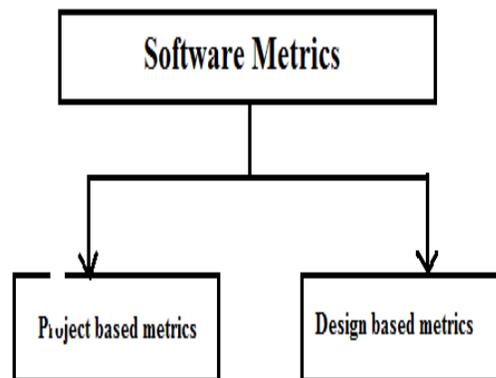


Fig 1: Software Metrics

For designing the object oriented metrics, UML is required. UML stands for unified modeling language. UML is the standard language. It is used to writing the software blueprints. It is used for the various software intensive systems.[3]. UML is very easy to understand and use.

B. MOOD Matrix

Mood matrix stands for maintainability of object oriented design. In the development of the software systems the object oriented developers are promises to reduce the maintenance effort. The object oriented development methods models the system components as the objects. These objects are helpful in allow the designer to separate the interface from the implementation. Earlier the maintainability can be defines in three different ways as,

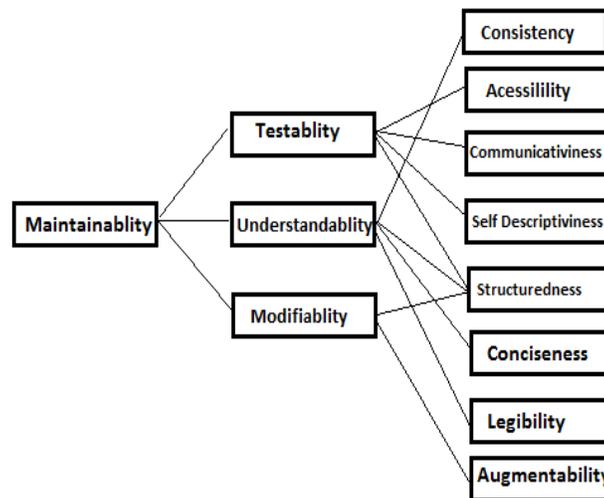


Fig2: Existing Maintainability Model

C. Testability

Testability is the quality of the software design. It helps in the automated testing. Testability is extrinsic property that helps to find out the various kind of bugs presented in the system. It also helps in reduces the bugs for effectiveness of the system. A testable product is used for the complete execution of the test scripts. When the testability is take place in the system, the customers reports the minimum number of defects. The testable products are easy and the cost to maintain product also less. Testability is an important aspect for the maintainability of software product.

D. Understandability

Understandability of the software system defines as the attributes of the software that uses the user efforts to recognize the various logical concepts. Understandability of the software system defines the user requirements, state of their tasks.

E. Modifiability

Modifiability is defined as the changes occur in the system to increase the performance of the system. Whenever there is the need to change the properties of the system, the developers change the many features according to the demand of the developer.

F. Component-Based Approach

This approach is used to revolutionize the development and maintenance of software systems. In this the distributed system approach is used. Many companies today claim to be doing component based development. All this development is comes under the distributed system.

G. Component Based Software engineering

Component based software engineering is the branch of software engineering that emphasizes the separation of concerns. Component based software engineering assembles the software products from pre existing smaller products. These products are known as the components. A component model generally defines a concept of components and rules for their design time composition and is usually accompanied by one or more component technologies, implementing support for composition and interoperation. [4] This practice aims to bring about an equally wide ranging degree of benefits in both the short-term and the long-term for the software itself and for organizations that sponsor such software. Software engineers regard components as part of the starting platform for service orientation.

II. LITERATURE REVIEW

Maintainability Estimation Model for Object-Oriented Software in Design Phase (MEMOOD)S. W. A. Rizvi and R. A. Khan [2010]: In this paper, author discussed about the development of a multivariate linear model ‘Maintainability Estimation Model for Object Oriented software in Design phase. Its increasing maintainability by adding two new factors understandability and modifiability into it. This paper shows the models, the models use design level object oriented metrics. These models use the multiple linear regression techniques [5]. The values of understandability, modifiability and maintainability are used in the software development process. These values help software designer to review the design and to take appropriate measures. The maintenance team uses the information to know that on what module to focus during maintenance.

A Maintainability Estimation Model and Tool, Alisara Hinceeranan and Wanchai Rivepiboon[2012]: In the given paper, author presents a multivariate linear regression to establish the maintainability estimation model. In this author add the two new factors, as: flexibility and extendibility. This tool helps in the software designer. It helps in improving the maintainability of class diagram during design phase. It also helps in reduces the increasing cost of

software maintenance phase [6]. The tool helps the software designer for evaluates maintainability of software system. These software development life cycles help a software designer to improves the maintainability of software system before delivering into the market for use.

Metrics Identification for Measuring Object Oriented Software Quality, Aman Kumar Sharma, Arvind Kalia, Hardeep Singh, [2012]: In this paper author discuss about the various quality metrics. In this paper author presents a review of quality metrics suites. The author used the various metrics like: MOOD, CK and Lorenz & Kid. In this paper process and product metrics are used, which are helpful in managing the activities, such as scheduling, costing, staffing and controlling. These metrics are also helpful for the different phases of the metric such as, analyzing, designing, coding, documentation and testing. [7] Software metrics is the measuring property which is used to measure the quality of a software object, which is related to software project of any size. The Object oriented metrics are useless if they are not mapped to software quality parameters.

Emergence of Component Based Software Engineering, Ardhendu Mandal, S. C. Pal,[2012]: In this paper author discuss about the need of component based software engineering. Now days with the help of Component Based Software Development (CBSD), the industry is moving into new direction. Most of the software that we used are not new, the software are built from the existing components. These days software systems are more complex as compared to those of early. [8] These complex, high quality software systems are built efficiently using component based approach in a shorter time. Component based systems are easier to assemble. Component Based Software Development aims to construct complex software systems by means of integrating reusable software components. This approach promises to alleviate the software crisis at great extents. The objective of this paper is to gain attention towards this new component based software development paradigm and to highlight the benefits and impact of the approach for making it a successful software development approach to the concerned community and industry.

A Maintainability Estimation Model and Metrics for Object-Oriented Design (MOOD) Kiranjit Kaur, Sami Anand[2013]: In this paper author discuss about the maintainability of MOOD metrics. Maintainability of any software in design phase helps a software designer to improve the maintainability of software before deliver to a customer. The maintainability is ease with which a software system or component can be modified to correct faults, improve performance. [9] 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. Object Oriented development use to reduce the maintenance effort that not based on reliable experimentation. In this paper, author discuss about the multivariate linear model. 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.

III. PROPOSED WORK

In earlier time maintainability of object oriented design matrices contain Testability, Understandability, and Modifiability. These three factors are used to help in maintains of the software projects. The present work is about increasing the maintainability factors of the MOOD metrics. These factors help to increase the functionality of the software system. With the help of these factors the reliability, portability and maintainability of the software system become easy.

In earlier time maintainability of object oriented design matrices contain Testability, Understandability, and Modifiability. These three factors are used to help in maintains of the software projects

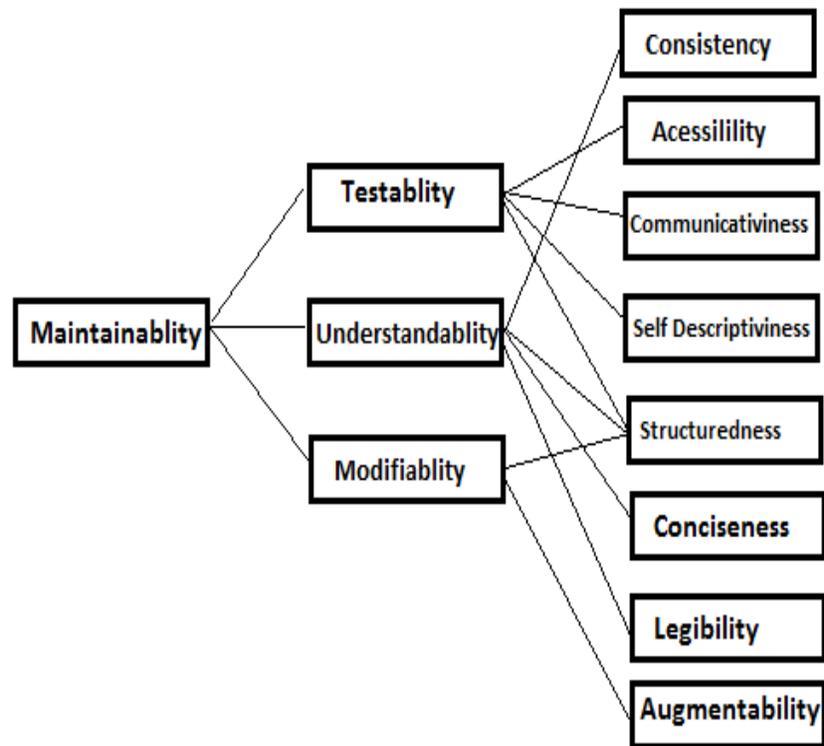


Fig 3: Earlier model od MOOD matrices

But as per our base paper two new components are integrated in it. These two new factors are reliability and portability. These factors enhance maintains of maintainability of object oriented design matrices up to higher level.

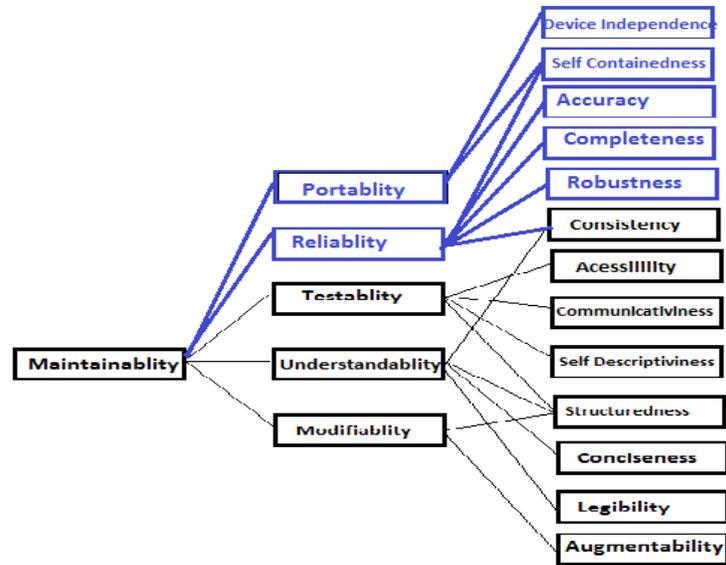


Fig 4: Present model of MOOD matrices

According to base paper the effect of these factors on maintains of software projects is given below:

TABLE 1: effect of factors on MOOD matrices

FACTOR	Sub-Factor	Effect on maintains
Testability	Accessibility	2.1%
	Communicativeness	8.7%
	Self Descriptiveness	9.32%
Understandability	Consistency	10.09%
	Conciseness	4.21%
	Legibility	9.27%
Modifiability	Structrendness	10.51%
	Augmentability	1.04%
Portability	Device independence	7.87%
	Self containedness	11.11%
Reliability	Completeness	7.34%

	Robustness	3.25%
	Accuracy	5.21%

So these are various factors on which work is done till now and the effect of these factors is on maintains of software projects is shown in the table. But still we can improve maintains of software projects.

If we talk about CBSE model, means component based software engineering, it allows us to reuse the previous existing components into new projects. In other hand we can use old existing components into new building projects. For example if there is a one XYZ software making company. This company had done lots of projects in different languages. It has all software project components like authentication component, data access component, login component, file edit component etc. now suppose this company is working on a new project and it needs a login component for new project, so according to CBSE it can its old existing login component into new project.

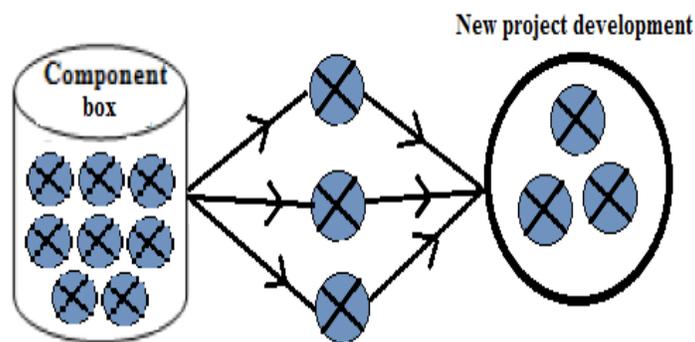


Fig 5: Component based software engineering model

Now these various factors Testability, Understandability, Modifiability, Portability, Reliability are also considered while using component reusability according to CBSE.

For example if we talk about Portability, it means we are going to use one component with other project of different platform, as example we can say that we are going to use one component which is build in C++ with the whole project which is build into java. So how this part effects the maintains of project?, it can be compatible with other components or not? If its compatible then it can enhance the maintains.

- 1) Testability: if there is a lack in testing after reusability of existing components then it may effects on maintains of software projects.
- 2) Understandability: if developer does not properly understands the user requirements and he/she uses some old existing components into new project but the component which they used is not

perfect for user and it does not compatible with all other components then it will decrease the maintains of software project.

- 3) Modifiability: if the modification is required into software project and instead of modification developer uses old existing component which is very complex then it also creates lack in maintains.
- 4) Portability: suppose if we are going to use one component which is build in C++ with the whole project which is build into java and developer converts both components into same script but still both are not working properly then they will decrease maintains.
- 5) Reliability: it deals with reusability of components, means we can reuse components by checking their compatibility if they are not compatible to each other then it's difficult to maintain the project.

So working on these factors to know the effects of Testability, Understandability, Modifiability, Portability, Reliability on maintains as during component based software engineering is our main object of thesis.

REFERENCES

1. Booch, G: "Object-Oriented Analysis and Design with Applications", 2nd ed., Benjamin Cummings, 1994
2. El Emam, K., "A Methodology for Validating Software Product Metrics," National Research Council of Canada, Ottawa, Ontario, Canada NCR/ERC-1076, June 2000 June 2000.
3. Roger S. Pressman: "Software Engineering", Fifth edition, ISBN 0077096770
4. Inheritance Metrics For Object Oriented Design, Kumar Rajnish1, Arbind Kumar Choudhary [2010]S. W. A. Rizvi and R. A. Khan [2010]
5. Maintainability Estimation Model for Object- Oriented Software in Design Phase (MEMOOD),pp. 26-32
6. Alisara Hinceeranan and Wanchai Rivepiboon [2012] A Maintainability Estimation Model and Tool ,pp.-143-146
7. Metrics Identification for Measuring Object Oriented Software Quality,Aman Kumar Sharma, Arvind Kalia, Hardeep Singh,[2012]
8. Ardhendu Mandal, S. C. Pal, Emergence of Component Based Software Engineering, Volume 2, Issue 3, March 2012 ISSN: 2277 128X
9. Kiranjit Kaur Sami Anand, A Maintainability Estimation Model and Metrics for Object-Oriented Design (MOOD), ISSN: 2278 – 1323 International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2, No 5, May 2013