



Proposed Maintainability Model for Software Development: Design Issues

Anshul Mishra¹ and Ajay Kumar Yadav²

¹Institute of Environment and Management, Barabanki, 226021, UP, India

²United Institute of Management, Allahabad 211010, UP, India

Abstract: Maintainability is important phases of quality control in object oriented software development. Proposed maintenance model may greatly reduce overall maintenance effort and rework if we apply this model at design phase in development life cycle. The system state is decentralized among the objects and each object manages its own state information. This paper developed a multivariate process model provide efficient and effective support for object oriented software. Discuss about maintainability approach to automatic maintenance of object oriented software which is carried out at the time of software design. Basically some process model or framework use design level object oriented metrics, to quantify understandability and modifiability of class diagram. Finally we close with a discussion of where maintainability framework first in the present and future, object oriented software in design phase.

Keywords: Maintainability, Software Changeability, Attributes, Software Development Life Cycle, Preparation, Object Oriented Design.

I. Introduction

An integrated part of software is the maintenance one, which requires an accurate maintenance plan to be prepared during the software development [5]. The software maintainability models, which can last for 5-6 years after the development process, calls for an effective plan which can address the scope of software maintenance, the tailoring of the post delivery process the destination of who will provide maintenance, and an estimate of the life cycle cost [8]. IEEE standards glossary of software engineering as “the ease with which a software system or a component can be modified to correct faults, improve performances or other attributes or adapt to changed environment”. The software maintenance is an expensive and challenging task. Software maintenance takes more efforts than all other phase of software in design phase, but it has not been given as much importance as it deserved. It is admitted fact that approximately 65% to 85% efforts are spent on maintenance phase of software development in design phase. Developers develop software for improves the maintainability of class diagram of design phase and observe the impact analysis, implementation coverage criteria. Design the maintainability term according to ISO/ IEC 14764 software maintenance in software engineering is the modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the produce to a modified environment. The maintainers often rely on designers for an understanding of concepts and intentions in the system design and a poor relationship there will be also cause maintainability to suffer [10]. While many reasons are forwarded in an attempt to explain the spiraling cost of the software maintenance, a consensus has emerged that the maintainability of the software system is dependent on its object oriented design [11]. Proposed framework include concepts of modularity, understandability, changeability, testability, reusability and transferability from one development team to another. These do not take the form of critical issues at the code level. Poor concept of maintainability is typically the result of thousands of minor violation with best practices in documentation, complexity avoidance strategy and basic programming practices that make the differences between clean and easy to read code vs. unorganized and difficult to read code [6]. In this paper we will mainly focus on reducing maintenance cost through the framework of maintenance i.e. to develop framework to assess the quality characteristics of software affecting the software maintenance cost. In the case develop the software in design phase every module of software to improve the analyzability, changeability, stability and testability. Stability of software to measures the stability of object oriented software system by software change propagation analysis using a simulation way in software dependency network at feature level. The first Phase the product operation needs are translate into a set of operational requirement and high areas are highlighted. Transition phase is the second phase of the framework. Some of the attributes or sub attributes of maintainability task associated with this phase are developing a maintainability program plan that satisfied the next coming phase. The performance phase is the third phase of the per posed framework. Evaluating with the all proposal for changes in regard to their impact, change coverage, & test scenario on maintainability.

Participating in the development of appropriate controls for error, process variations, and other problem that may affect maintainability directly or indirectly [4].

II. Related Work

Maintainability is a word we all know in testing but sometimes have difficulty explaining. It is generally regarded as something we would like to have in our software but it's hard putting our finger on what maintainability accurate is. The ISO 9126 quality model defines maintainability as "the ease with which a software product can be modified to correct defects, modified to meet new requirements, modified to make future maintenance easier or adopted to a changed environment". Design for maintainability requires a product that is serviceable and supportable better yet if the design includes a durability features called reliability then you can have the best of all words [1, 7]. Maintainability is an important Character which is given by product design, it makes easy to be repaired for mechanical system. It has a specific effect on the maintenance cost mechanical system [2]. A number of researchers addressed software maintainability but in the context of conventional structured design. Following section systematically summaries some of the relevant important efforts made by researchers in this area.

In 1980 YAU, S.S and Colofello, J.S highlight that are depends upon the design Guideline. They used Algorithms (relates the measures for estimating) techniques one of the most important quality attribute of software maintainability is the stability of a program, which indicates the resistance to the potential ripple effect that the program would have when it is modified [3].

IN 1982, R.S .Arnold has given a great answer against the two types of questions:

1. What characterizes "healthy" Software maintenance?
2. What Characterizes "satisfactory" software maintenance?

They first need for objectively measurably maintenance performance critical in judging in 'adequacy' of maintenance and present a set of criteria for judging maintenance performance. Totally relate these questions arise first in software development in design phase.

In 1998, Lionel Briand highlights software maintenance processes that are depend upon artifacts of software module. To measure or identifying their specific problem and need, they propose a qualitative, inductive method for characterizing and evaluating software maintenance process. This method allow for a set of concrete steps to be taken for maintenance quality and productivity improvements [9].

In 2000 Stephen R.schach presents a process for software construction that recognizes maintenance. In order to improve main ability, the requirements frequently change during development of Software design phase [12]. The data structure, propagation graph, maintenance methodology during the development of software. Requirement changes the process treats development as a special case of software maintenance.

In 2001 Guy Gable and Taizan Chan focus on seeks to stimulate research on large application package software maintainability by presenting a tentative framework for future research in the area, partially instating that frame work with examples from the papers in this special issue at the journal of software maintenance and evolution, highlight other specific areas of research need and neglect and posing along list of related research questions [6]. In 2009 YU FENG Ding focus on maintainability design criteria measure index used in product maintenance analysis. It uses a large scale of maintainability design support tool through the unified modeling language in case diagram.

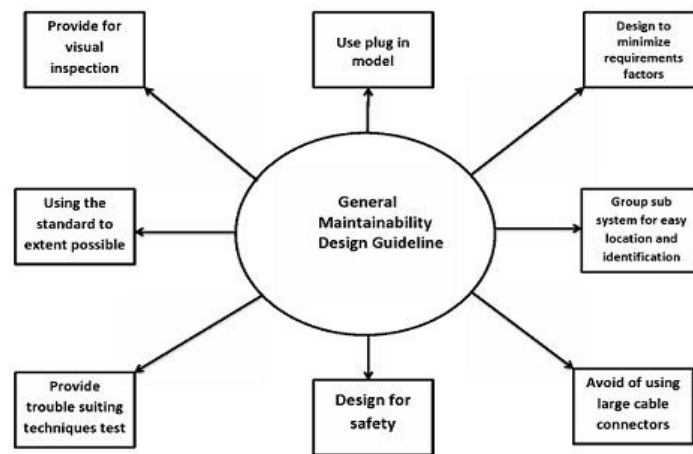
A Survey of maintainability estimates of object oriented design shows the maximum efforts focus at the later stage of software development life cycle. On the other hand produce our framework may be reduce the efforts and cost of software development in design phase.

III. MAINTAINABILITY MEASURE

Some of the important general design guidelines that maintainability professional have developed are shown in figure 1 [14]. Maintainability criteria are composed of in trinsic criteria and contextual criteria.

There are four major categories of maintenance [18].

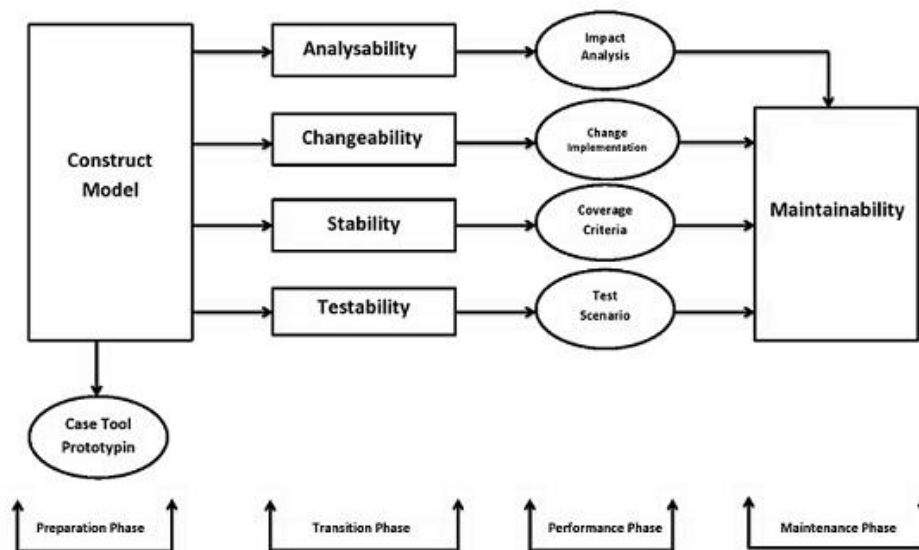
- Corrective maintenance: Reactive modification of a software product performed after delivery to correct discovered problems.
- Adaptive maintenance: Modification of a software product performed after delivery to keep a software product usable in a changed or changing environment.
- Perfective maintenance: Modification of a software product after delivery to improve performance or maintainability
- Preventive maintenance: Concerns activities aiming on increasing software maintainability and prevent problems in the future.



General Maintainability Design Guideline
Figure 1 [14]

IV. Proposed Framework

In this section, we discuss the different activities (relate maintainability activity) in this proposed framework. Display the 4 phase, every phase performs a set of task during the development of software. This type of Framework may greatly reduce the efforts and cost starting phase to ending phase.



Maintainability Process Framework

figure 2

A. Case tool prototyping:

Prototypes must be produced quickly and cheaply. The reason that the idea of prototyping so late is that tool for the efficient productions of prototype are difficult to produce and thus was not available is sufficient quality [13]. Propose framework case tool prototyping support to the construct model. Prototyping is an important technique to reduce the cost and risk involved in developing complex software system [15]. Prototyping tools involved build a big system, significant cost and risks are still involved these tools is an involves techniques for construct model (interactive design term) after developing software during design.

B. Transition Activity:

In quality Standard ISO 9126 in maintainability one out of six Quality characteristics. Over the years cumulative data have shown that maintenance is also a major cost concern Transition phase has four component analyzability, testability, stability and changeability. Main key in this phase changeability an important aspect of maintainability especially in environments were software changes are frequently required.

C. Impact Analysis:

Impact analysis very important tools factors to the software development of design level. The proposed method maintenance to start phase at design level where software changes are frequently required during processing the last stage. The best maintainability model is prime concern for developing quality software system. Impact analysis support to create the option for best maintainability model for software development.

D. Test Analysis:

Every test method target specific function of the system to be tested. The extent to which the targeted features are tested can be determined using test analysis [16]. Critical analysis of empirical research in software testing and will attempt to highlight and clarify the issues above in a structured and practical manner. Test cases can easily comprise of input data that is unexpected by the unit under test, something which may be hard to achieve during system testing. [16, 17]

V. Conclusion

In this paper we suggest a model or Framework for maintainability. The domain of maintainability is too huge to be done by the individual researchers or group of researchers. Researchers can not avoid the models. This framework may be support to reduce the efforts and cost in case of software development. This framework substantial benefits it term of increase productivity and reduce development time and cost. The choice of models also depends on aspect of the system under test and skills of users. Maintainability efforts allocations can be made easy by knowing complexity of cost, time and efforts. We will further explore this research.

References

- [1]. <http://www.barringer1.com/jul01prb.htm>.
- [2]. B.Abdullah , M.S.Yusoff and Z.M.Ripin, " Integration design for modularity and design for assembly to enhance for product maintainability" proceeding of 1st International Conference 7th AUN/SEED-net field wise seminar on manufacturing and material processing , pp-263-269, University Malaysia , 2006 .
- [3]. Yau, S.S. Department of Electrical Engineering and Computer Science, Northwest University Collofello J.S., 1980.
- [4]. L.Li and A.J.Offutt., "Algorithm Analysis of the Impact of Changes to Object-Oriented Software", In Proceedings of the International Conference on Software Maintenance (ICSM'96), Monterey, CA, pages 171-184, November 1996.
- [5]. A. Abran and H. Hguyenkim, "Measurement of the Maintenance Process from a Demand-Based Perspective" , Journal of Software Maintenance: Research and Practice , Vol. 5, no 2, 1993 [6].M. Lorenz, "Object-Oriented Software Development", A Practical Guide. Prentice Hall, NJ, 1993.
- [7]. IEEE Standard Glossary for Software Engineering, Terminology, ANSI/IEEE STD 610.12,1990.
- [8]. Ruchika Malhotra¹ and Anuradha Chug², "Software Maintainability Prediction using Machine Learning Algorithms", Software Engineering: An International Journal (SEIJ), Vol. 2, No. 2, SEPTEMBER 2012.
- [9]. L.Briand, "Qualitative Evaluation of Maintenance Organizations", process and products", Software Maintenance: Research and Practice.10, 249-278, 1998
- [10]. Han, J., "Designing for increased software maintainability", In: Proceedings International Conference on Software Maintenance. IEEE Computer Society Press, pp. 278–286 , 1991.
- [11]. W. Li, S. Henry, D. Kafura and R. Schulman, "Measuring Object-Oriented Design". In Journal of Object-Oriented Programming, Vol. 8, No. 4, pages 48-55, July-August 1995.
- [12]. Stephen R.schach,"A Maintenance Oriented Approach to Software Construction", Volume12, Issue1, pages 25-45, January, 2000.
- [13]. G.Pomberger, C.Doppler, "The Role of Prototyping in software Development", Laboratory for Software Engineering, Austria, 1992.
- [14]. M Pecht, "Product Reliability maintainability Supportability handbook" , CRC Press, BocaRaton Florida,PP.191-192,1995.
- [15]. Buchenau, M and J. F. Suri. "Symposium on designing interactive systems", In Conference Proceedings on Designing Interactive Systems: Processes, Practices, and Techniques, pp. 424-433. New York: ACM Press 2000.
- [16]. H. Zhu, P. Hall, J. May, Software unit test Coverage and Adequacy, ACM Computing Surveys29 (4) (1997) 366-427.
- [17]. Schwaber, C., Gilpin, M., Evaluating Automated Functional Testing Tools, Forrester Research, February 2005.
- [18]. Lientz, B. P., Swanson, E. B., " Software Maintenance Management", Addison - Wesley Reading, MA. 2000.