iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** IV **Month of publication:** April 2017

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

Volume 5 Issue IV, April 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

A Study on Factors Affecting Maintainability and Maintainability Models

Deepa N[1], P. V. Indu Bhanu[2], C. S. Kausthubhi[3], M Sai Sriya[4]
[1,2,3,4] *School of Information Technology & Engineering, VIT University*

Abstract: ***In the day-to-day life, Quality of software is essential. To achieve this quality, the main factor associated is Maintainability. So it is important to measure the maintainability of the system. Maintainability is defined as the process of changes made in the software very easily. To adapt with the changes in the environment, some of the functionalities are added and removed from the system to achieve the customer needs. While making the changes, some of the modules of the system may change and features also may change. The cost of maintenance is high. This paper gives the detailed description of important factors through which the maintainability is affected. To assess the maintainability of the system, these factors have a major role. The main purpose of the paper is to identify the various factors from various categories that effect the maintenance.***
Keywords*: *Software maintenance, software maintainability, factors affecting maintainability, maintenance process, software quality, maintainability models.

I. INTRODUCTION

We already know that huge expenses are made during the development and maintenance of the software to meet the user's needs. The expenses during the development include identifying the bugs, fixing the bugs, installing the new software, modifying the existing software. The expenses during the maintenance include changes to be made, the errors caused due to the changes, correcting the errors. The expenses may become huge while trying to search the error in the wrong place. So the primary concern while making changes in the developed system is, the cost for modifying it. While maintaining the system, we make use of resources in a large amount. So, to produce a software which can be maintained easily, may have chances of reducing the expenses.
Design is responsible for affecting the maintenance of the system. And also, architecture principles affect the maintenance of the system. The system which is not effectively designed for the purpose of maintenance will not be helpful for modifying the functionalities in the system later. So if the system is defined effectively which helps in the maintenance that may reduce the effort of evolution and maintenance i.e. nothing but designing the system with increased ability to maintain, which further helps us to make more modifications with the help of similar resources, or make the same modifications with limited resources.
The two terms i.e. Software maintenance and maintainability are almost related to each other. That is, Maintainability is defined as the ability to perform system's maintenance very easily. Whereas, Software maintenance is nothing but the activities involved in modifying the system which is already developed and given to the customers.
The quality of the software is generally defined as the total number of features in the system that would meet the requirements specified by the user. The quality of the software system is also affected by the maintenance of the system.
The various characteristics of the system are:

A. Functionality

The feature that would meet the user needs.

B. Reliability

The system should perform effectively and efficiently without any failure

C. Maintainability

The ability to maintain a software.

D. Portability

The system should run on any platform.

E. Robustness

The system should perform efficiently

F. Throughput

To improve the performance of the system.

These features are further divided into low level criteria of the quality or sub- features that are used as the software system's attributes. This paper's key objective is to identify the various partitions with the factors that affect the maintenance of the software. The factors influencing the maintenance of the system can be termed as Cost Drivers. Cost Drivers are defined as the factors which serve as indicators that derive various costs that are used while maintaining the system. This paper contains various effects on the maintenance of the system produced by many factors. This paper also describes various models in software maintenance.

II. SOFTWARE MAINTAINABILITY

There are numerous definitions available for the software maintainability. Maintainability is an aspect of the quality of software which reflects the following work to be performed [4]:

A. Removal of error
B. Increasing the Features or characteristic
C. Remove the features that are not use
D. Software revision

According to IEEE Glossary Terminology [5], maintainability is explained as "the ease of modification of the system i.e. modification such as fault correction, performance improvement or other features, or adapting to a new environment".

The time required for the maintenance of the system in the life cycle of the software development is 65%-75% of whole development time. Large amount of time is needed during this maintenance stage and it also affects the software cost. The more the effort you put in maintaining the software at the time of its life cycle of development, will reduce the cost of the software.

III. SOFTWARE MAINTENANCE

During the life time of the software, it has to undergo large amount of changes. We have numerous reasons for the occurrence of these changes. The reasons for the change in software are[4]:

A. Change in the user needs
B. Errors in the design of the software
C. Environment change
D. Hardware change makes a change in the software

The important stage in the life cycle of the software is maintenance of the software. The main idea is to make the software to be up to date and to have complete functionalities. The changes in the user needs on the system will make the products of the software to be changed by removing the existing functionalities and adding new functionalities. There are four types of software. They are, namely, perfective, corrective, adaptive and preventive maintenance [5].

We perform Adaptive maintenance to make the software compatible with the new environment such as hardware, database and operating systems. We perform Corrective maintenance to rectify the errors in the different phases of the software development life cycle. If the faults which are present at the time of delivery and the faults that are present in the software at the time of maintenance is allowed to be removed, then such software system is either called as corrective or repairable. We perform preventive maintenance in order to make the product work, to increase its life time. We perform perfective maintenance to increase the level of performance and the speed and changing the functionalities.

The important stage in the life cycle of the software is maintenance of the software. The main idea is to make the software to be up to date and to have complete functionalities. The changes in the user needs on the system will make the products of the software to be changed by removing the existing functionalities and adding new functionalities. There are four types of software. They are, namely, perfective, corrective, adaptive and preventive maintenance [5].

We perform Adaptive maintenance to make the software compatible with the new environment such as hardware, database and operating systems. We perform Corrective maintenance to rectify the errors in the different phases of the software development life cycle. If the faults which are present at the time of delivery and the faults that are present in the software at the time of maintenance is allowed to be removed, then such software system is either called as corrective or repairable. We perform preventive maintenance

in order to make the product work, to increase its life time. We perform perfective maintenance to increase the level of performance and the speed and changing the functionalities.

IV. MAINTAINABILITY MODELS

There are numerous models that have been proposed. Some of the models consider only matrix of the design level whereas some may concentrate on the design and coding. These models are[4]

A. Fuzzy model
B. Bohem's quality model
C. ISO 9126 software quality model
D. McCall's model
E. Memood model

These type of models use various methodologies and functions to know whether the maintainability is possible or not. The features of the software are considered for the measurement of maintainability that are indeed called as factors that affect the maintainability. We have models to evaluate maintainability like:

F. Deterioration and Maintainability Model[6]

By name we can understand this a model which consider two important aspects i.e. maintainability and the other is deterioration. This model uses three dimensions:

1) Maintainability
2) Effort
3) Change

This model helps us to maintain the software product without being deteriorated.

This model distinguishes between the approaches to implement the change. The approaches are:

4) Fast hac
5) Control update

Fast hack approach is to make a change in the system with little effort, but here the system maintainability will be decremented time to time where as in the other approach we may apply minimum effort without deteriorating the system.

G. Maintainability Estimation Model for Object-Oriented Software in Design Phase

1) MEMOOD [7]:
This model deals with three important factors maintainability, understandability and modifiability[7].
This is to estimate the maintainability based on the other factors.
In this model we will identify the class diagrams of a software products maintainability based on how well they are understood and how well they can be modified.
This model uses linear regression method [7] and Pearson correlation coefficient method [7] to perform evaluation.

2) A methodology based on data mining is used to evaluate the maintainability [10].This is based on quality standard ISO/IEC 9126.this method uses k-means clustering algorithm. The process to evaluate will take place in three steps.
First step is to extract the needed metrics and elements which is stored in an xml file of relational database which further can be used to perform evaluation or analysis as per the need.
Second step is to perform the Analytic Hierarchy Process[AHP] for the metrics which are identified.
Third and last step is to apply k-means clustering algorithm on the values which have been derived based on this the maintainability behavior can be identified.

H. Enterprise Architecture Models[11]

This model is also to evaluate maintainability in the system. This model provides a framework for evaluation. This uses diagrammatic representations of the software product or system.

1) Framework has been created[8] which is used to measure the maintenance. It is based on two important concepts: Product and Process
2) A practical model[9] which assumes that the maintenance of system is closely related to the source code by which the system is

developed and the effort applied to develop that.

In this model various metrics related to the software has been identified and they have given rating, numbering then they are correlated to check there effect with the maintenance of the system.

V. FACTORS AFFECTING MAINTAINABILITY

It is very important for the client of the software to measure the maintainability. Different types of attributes are predicted for the maintainability of system by different researchers in the literature. This paper provides an overview about the different categories that influence the maintainability of the software. The important factors which affect the maintainability of the system that are proposed by various researchers are:

A. The important quality factor according to ISO9126 is maintainability. It is best if approached at the design and programming level of the software. The main factors which affect the maintainability are stability, modifiability, analysability and testability. The factors which affect the code, documentation, tools required for development are focused by maintainability. These areas are supported as to increase productivity and quality while maintenance and to avoid servicing. Now lets discuss about the factors[4]:

TABLE I Factors That Affect Code, Documentation, Tools

Factor	**Description**
Analysability	It involves the need to recognize errors are failure causes and to recognize the modification of the parts of the software.
Modifiability	The steps taken to make changes in the system to make the implementation simpler.
Stability	It involves with the risk that come out from the results of unexpected modifications.
Testability	It validates the changes. It is the measurement of how the test criteria and the components of the system are done.

B. A methodology is proposed which depends on six attributes which is proposed by David E.Peercy. These attributes affects maintainability. These attributes are descriptiveness, simplicity, instrumentation, modularity , consistency and expandability[2].

TABLE II Factors Affecting Based On a Methodology Of David E.Peercy[2]

Factor	**Description**
Modularity	It gives the description of the logical division of the software into numerous modules, parts and components. It is easy to understand and make changes in the software if it is composed of independent components.
Descriptiveness	It describes the explanations of the design. It gives the information about assumptions, components, objectives etc..
Consistency	It indicates the association of the products and the terminology, symbology and notations that are common.
Simplicity	There should be no complexity in the implementation techniques, languages used and the organization. It uses the concepts of the similarity
Expandability	It describes that the changes in the physical information such as data storage, time of executions, computational procedures can be achieved easily.
Instrumentation	It helps in the improvement of testing.

C. A maintainability model of the software is proposed by the researchers to improvise the software quality so that the

maintenance of the software becomes effective. This model proposes some factors affecting the maintenance such as[4]:

TABLE III Factors Affecting Maintenance Based on Maintainability Model

Factor	Description
Readability	The extent of the code to which the user can understand.
Standardisation	The standards or guidelines which are available while writing a code.
Modularity	Decomposition of the system into numerous modules according to their functionalities to implement the abstractions of the data.
Programming Language	The code developed by a programmer in a language which can be reused.
Level of validation and testing	More time spending on validation and testing gives less errors and reduces the cost of maintenance.
Complexity	The difficulty in maintaining the code of the system should be put to a certain level.
Traceability	It traces the components to the requirements.

D. A fuzzy model is proposed by the researchers to measure the maintainability. This includes four factors which provides measurement of maintainability[3]:

TABLE IV Factors Affecting Maintenance Based on Fuzzy Model

Factor	Description
Average number of live variables	A live variable is defined as a variable which is referenced frequently. It is defined as the average of total live variables to the total executable statements. If the average number of live variables is more it is difficult to maintain the system.
Average live variable span	The time between two successive references of a variable. It is defined as the average of averages of size of span of all modules to the total modules.
Comments ratio	It is defined as sum of number of comment lines used and total LOC. If it is lower it is better for the readability which indeed is better for the maintainability.
Average cyclomatic complexity	It is average of cyclomatic complexities of all components.

E. To measure the software structure properties many researches are complexity measures of the maintainability by HalStead. Maintainability index is the combination of these properties. Anda tried to measure the maintainability by some factors[1]:

TABLE V Factors Affecting Maintenance Based on Anda[1]

Factor	Description
Design for the problem domain	The problem should be able to justify the reason for the selection of the design. With the help of design patterns, the process for the maintenance will be simpler. The developers must be best competent to prove that a better design is a better solution.
Appropriate comments	Comments must be in an understandable format so that it can help in improving the codes readability.
Naming conventions	If the standard conventions are used for writing the names of methods, classes, etc. will helpful in understanding.

Volume 5 Issue IV, April 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Objects and classes	The process of mapping classes between the various phases of software should be simple.
Use of components	The classes must be in an organized format with respect to their functionalities and the sequence of the code in which they function.
Technical platform	This is about the system which has developed without any documentation or developed with an ineffective document while changing its environment, then it will be a challenging task to make the changes in the system.
Simplicity	The very important and difficult issue is to identify the size and effort of the system. If it is a duplicate code then the situation becomes tough. Here it is difficult to identify a particular class because of presence of many classes. So with this, maintenance of the system is difficult.
Encapsulation	We knew that in java, we have the concept of dependencies where the object is created before its method is called, which causes problems of the maintenance.
Inheritance	Inheritance must be systematically used without affecting the systems maintenance. The concept of multiple inheritance have large amount of classes which is difficult to understand and will affect low maintenance.
Proprietary libraries	This factor will also affect the maintainability since the staff for maintaining the system must acquaint the system.
Layered architecture	It is very helpful to differentiate between various layers like persistence, presentation and business. The bottom layers should not depend on the top layers and the top layers should not completely dependent.

Besides these properties, various other factors also affect the maintainability. This factors include the talent of the employees who perform this task(maintenance), the tasks to be done on system for maintenance, the tools used in the maintenance, increase in the technology which includes improved standards and architectures, number of versions. There are different types of tasks in maintenance. If the modifications in the tasks are tiny, then it will be easy since it is easy to modify if it is restricted to the small region of the software. In this, the identification of the classes will be simpler. If we want to make a large change then it is difficult since it takes the many region of the system. During this large change, the system need to concentrate on various principles that are used.

VI. ADVANTAGES

A. If all the factors are considered properly then the maintenance of the system will be easier and simpler.
B. If the maintenance is performed effectively, then the cost will be less.

VII. DISADVANTAGES

In Real time, considering all the factors for the maintenance is difficult.

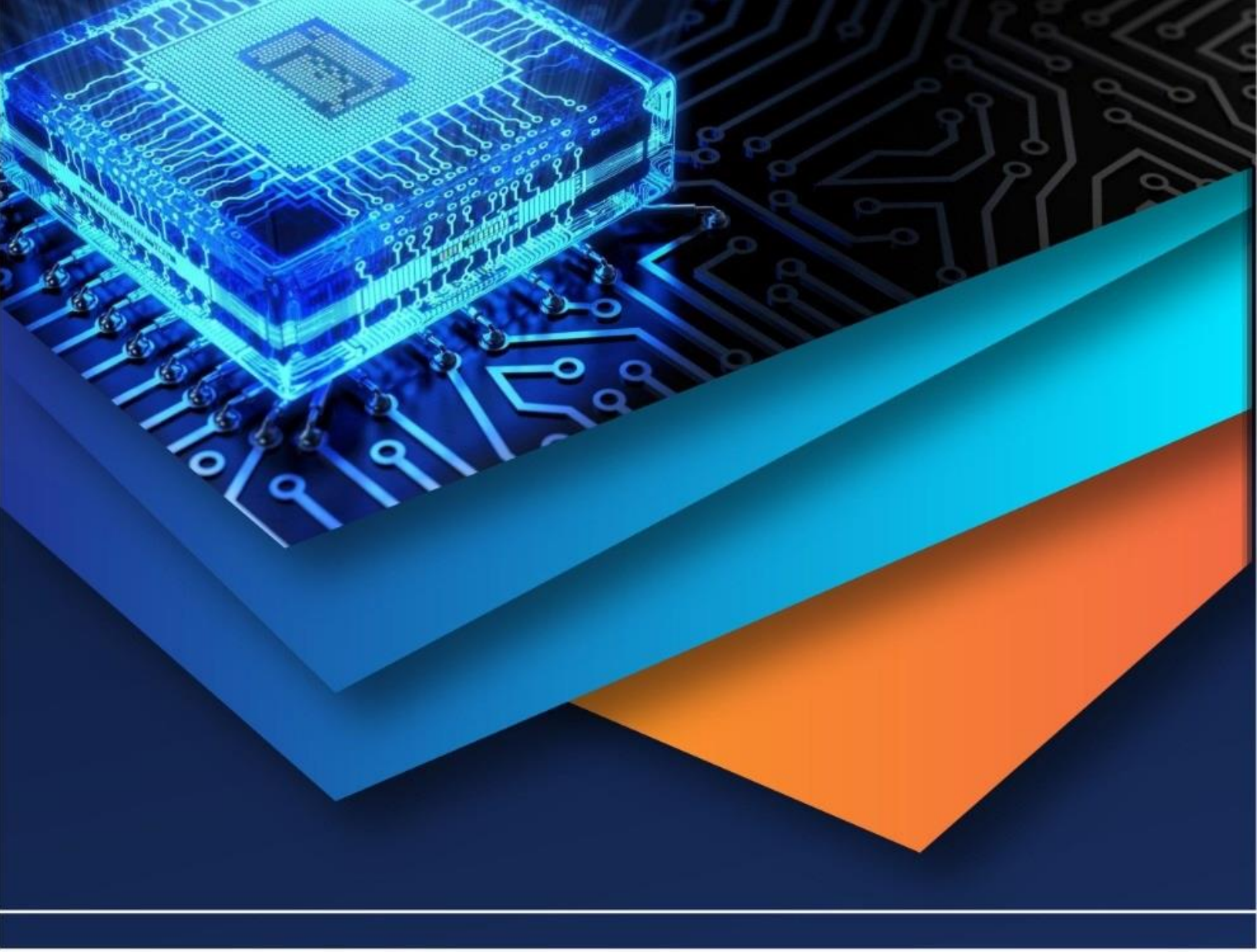
VIII. CONCLUSION

The ease of the modification of the system components or modules is software maintainability. In engineering, software maintainability is necessary to correct the defects, to meet the evolving requirements, to modify the product and cope with the changes in the environment. This also affects the time for the restoration of the system to its status of operation that results in a failure or operation removal. So, maintainability is necessary to identify, correct a fault.
Now on wards huge importance is to be given for the factors that affects the maintenance. Correct analysis and proper usage of factors that affect the maintainability, make the assessment of maintainability effective and efficient. If in the early stages of the

software development life cycle, the aspect of maintainability is given proper consideration, then software will be of high maintenance.

REFERENCES

[1] B.Anda,"Accessing software system maintainability using structural measures and expert assessments ", in IEEE International Conference Software Maintenance, 2007, pp.204-213.
[2] David E Peercy, "A Software Maintainability Evaluation Methodology ", IEEE Transaction on Software Engineering,Vol.SE-7,NO.
[3] K.K. Aggarwal, and Yogesh Singh, Pravin Chandra and Manimala Puri,"measurement and software maintainability using Fuzzy Model", Journal of Computer science 1(4):583-542,2005
[4] Balraj Kumar ," A survey of Key factors affecting Software maintainability ", 2012 International Conference on Computing Sciences.
[5] IEEE, IEEE Standard Glossary of Software Engineering Terminology, report IEEE Std, IEEE.
[6] L. Rickard, Software Deterioration and Maintainability-A Model Proposal, Malardalen University, In Proceedings of 2nd Conference on software Engineering Research & Practice, Sweden (SERPS), 2002
[7] M. J. Kiewkanya and N. Muenchaisri, A Methodology for Constructing Maintainability Model of Object-oriented design, Proc. 4thInternational Conference on Quality Software, pp. 206-213, 2004
[8] M. K. Mattson, A. V. Deursen, R. Reiger, G. Canfora, T. Ihme, T. Engel, D. Chiorean, M. M. Lehman and J. Wernke, A Model of Maintainability Suggestion for Future Research, In Proceedings of Software Engineering Research & Practice, pp. 436-441, 2006
[9] I. Heitlager, T. Kuipers and J. Visser, A Practical Model for Measuring Maintainability Software Improvement Group, Netherlands, 2007
[10] P. Antonellis, D. Antoniou, Y. Kanellopoulous, C. Makris, E. Theodoridis, C. Tjortjis and N. Tsirakis, A data mining methodology for evaluating maintainability acc. To ISO/IEC 9126 Software Engineering Product Quality Standard, Proc. 11th IEEE Conference on Software Maintenance & Reengineering (CSMR), 2007
[11] R. Lagerstorm, Analyzing System Maintainability using Enterprise Architecture Models, Royal Institute of Technology, Osquldas vag 12, 10044 Stockholm, Sweden, 2007



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY