Maintainability Techniques for Software Development Approaches – A Systematic Survey

Gagan Tiwari
Mewar University, Chittorgarh, India

Arun Sharma
Krishna Institute of Engineering & Technology, Ghaziabad, India

ABSTRACT

Time, effort and money involved in maintaining software has always been considered greater than its development time. Also, its vagueness in prediction at early stage of development makes the process more complex. Researchers and developers are working on devising various techniques/algorithms for better prediction. Present paper conducts a detailed survey on these techniques and identified several factors or characteristics on which maintainability depends. These factors vary for different software development approaches like object-oriented, component-based, aspect-oriented and others due to the architectural difference.

Keywords: Maintainability, Soft Computing, Object Oriented, Component Oriented, Aspect Oriented.

1. INTRODUCTION

Software maintenance, in general, refers to the set of activities that are performed to keep a system operational, as software changes after the system has been deployed. Software maintenance begins as soon as a system has been released to users for the first time. Maintainability in a system should be considered as one of the most important quality aspect. A lot of research work has been carried out to measure the maintainability. To measure it, one need to identify the set of attributes / factors that bear on the efforts needed to make specified modifications. As we know that maintenance time of software is always greater than its development time, so it is important to measure the maintainability of software to reduce the maintenance operational time. However, it may be very difficult to measure these factors and combines them to get final value of maintainability.

The definition of maintainability as per IEEE is defined 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 is maintainability. Maintainability is a system that can evolve from its current state to its future desired state. It is termed as the most difficult and costliest activity due to it's inherently involvement in making predictions about the future.

There are four major categories of maintenance [1]:

- 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.

A. Maintainability Evaluation

There are no criteria or fixed formula to measure or evaluate the maintainability. To increase systems functionality and systems complexity, many researchers have given different terminology and methods to evaluate maintainability but there is an inconsistency and vagueness in the terminology and methods involved with maintenance of the software applications.

B. Literature Review for Maintainability techniques

Many researchers have proposed their theory and designed models to assess the maintainability for the software applications. We divided the existing literature into two sections. First section describes the traditional measures for maintainability and second section describes the soft computing based measures for maintainability.

1) *Traditional Maintainability Techniques:* hairuddin and Elizabeth [2] proposed a maintainability model and included factors, namely, Modularity, Readability, Programming Language, Standardization, Level of Validation and Testing, Complexity and Traceability for evaluating the maintainability for software systems.

Another model was proposed by Fioravanti and Nesi [3] for effort estimation/prediction of adaptive maintenance. It assumed that the system efforts for adaptive maintenance are typically spent for performing several operations like understanding, addition and/or deletion of facts and modifications/changes of other system code portions. This metric is applied on several software projects and validated for predicting adaptive maintenance. The results shows that the proposed model is better ranked with respect to other models.

Bandini *et al.*, [4] considered three independent factors, namely; design complexity, maintenance task and programmer's ability to predict the maintenance performance for object-oriented systems. To measure design complexity, interaction level, interface size and operation argument complexity were chosen. Perfective and corrective maintenance were selected to represent maintenance task. Correlation analysis was performed to conclude that the selected attributes were able to predict the maintenance efforts of the systems.

Another interesting work for measuring software maintenance project effort estimation has been done by Ahn *et al.* [5]. They proposed a software maintenance project effort estimation model, which is based on the function point measure and 10 maintenance productivity factors. These factors are classified into engineer's skills, technology characteristics, and maintenance environment. Survey method was used to validate the proposed model. To find the relation of actual efforts and

function points, regression analysis was performed, by taking the data from maintenance projects.

The assessment was made by Ardimento *et al.* [6] that if a component is difficult to understand then it will be difficult to maintain it and advised the trial usage of component before adopting it for the application.

Several problems were discussed by Kajko *et al*. [7] related with the maintenance of the software systems and proposed two maintainability models separately for product and process. Product aspects, consisted of common characteristics, variable characteristics, maturity level, traceability characteristics and others, while in process aspects the common tasks were to manage maintainability through out the whole software life - cycle.

2) *Soft Computing based Maintainability Techniques:* Artificial Neural Network (ANN) based approach is considered to be very helpful in estimating/predicting maintainability of the system. ANN is inspired from biological nerve system. Its architecture, its learning algorithm, and its activation functions characterize ANN. A set of data (input/output) is used for training the network. Once the network is trained, it sets a relationship between input and corresponding output. Now this network can be used to predict the output for any input set of data. Researchers considered different factors as inputs for training the network.

Artificial Neural Network based approach is adopted by Singh *et al.* [8] to predict the maintainability of the systems. They considered Readability of Source Code, Documentation Quality, Understanding of Software and Average Cyclomatic Complexity as independent variables to measure Maintainability, which is considered as dependent variable. These variables were used to train the Artificial Neural Network, by using Mat Lab. The results obtained were quite appreciable with the prediction quality of 91.42%.

Fuzzy model is used to measure the maintainability of the software system by Aggarwal *et al.* [9]. The authors considered four factors affecting maintainability, namely, average number of live variable, average life span of variables, average Cyclomatic complexity and the comments ratio. All inputs are classified into fuzzy sets viz. low, medium and high, while maintainability is classified as very good, good, average, poor and very poor. All the inputs and outputs are fuzzified and in total 81 rules are proposed for the model. Model is validated against the software projects developed by undergraduate engineering students. However, the model is not validated on real life complex projects.

ANN based approach is adopted by Aggarwal *et al.* [10] to predict the maintainability of the object -oriented systems. They considered principal components of eight Object Oriented metrics as independent variables. These include Lack of Cohesion, Number of Children, Depth of Inheritance and others. Maintainability was considered as dependant variable. Results obtained by training the network using back propagation algorithm show that independent variables chosen for the study were able to predict the maintenance efforts with a mean absolute relative error of 0.265.

ANN based approach is also used by Shukla and Mishra [11] to estimate software maintenance efforts. They chose 14 factors as cost drivers for their study and conducted the experiment by taking various options of number of hidden layers and number of hidden nodes. Input data selected for training was 60% of the total, while 20% each were used for validation and testing. Mean Relative Error obtained from the experiment was around 5%. Results concluded that neural network was able to successfully model the maintenance effort.

The ISO 9126 model is extended to add one sub-characteristic, Trackability under Maintainability and proposed a Fuzzy Logic based approach to predict the maintainability of the Component Based System by Grover *et al.* [12] [13]. Authors considered Interaction Complexity, Reusability, Testability, Understandability and Trackability as inputs to predict the maintainability as output. Inputs are designed by Fuzzy sets as Low, Medium and high, while maintainability as Very Low, Low, Medium, High and Very High. Total 243 rules were provided to fuzzy inference engine to get the output. The value of maintainability is measured then by using defuzzification process. The proposed model is then applied on a classroom-based project to evaluate its maintainability.

In this paper, the authors Ardil *et al*. [14] have explored the different predictor models to NASA's public domain defect data-set coded in Perl programming language. In this paper various machine learning algorithms including Fuzzy and Neuro-Fuzzy based techniques are explored and analysis is performed for the prediction of level of impact of faults in software modules. In this the author tries to find the best algorithm for classification of software component in different level of impact of faults. The author compares the different machine learning algorithms on the basis of the least value of MAE (Mean Absolute Error) and RMSE (Root Mean Square Error). The author uses Logistic Model Tree (LMT) and Simple Logistic algorithms for analyzing the performance of all the algorithms. The conclusion is based on the basis of comparisons.

For Aspect Oriented systems Kumar *et al*. [15] have chosen maintainability factor as important factor out of six main desirable characteristics of software quality they are: functionality, maintainability, usability, efficiency, reliability and portability as defined in ISO 9126. Out of four sub-attributes of maintainability – analyzability, changeability, stability and testability, authors have considered changeability as the most significant from the point of view of organizations. Aspect-Oriented Programming (AOP) approach is compared with Object Oriented Programming (OOP) approach for the analysis based on changeability sub-factor of maintainability factor of software quality. AspectJ is used for AOP. The essence is that a change is easily absorbable in AO system. In comparison to object oriented system aspect oriented system can observe more changes.

In this paper the authors Kaur *et al*. [16] tries to evaluate and compare the application of different soft computing techniques like Artificial Neural Network, Fuzzy Inference System and Adaptive Neuro Fuzzy Inference System for predication of software maintenance effort. The author uses OO paradigm with OO design metric for the input to ANN, FIS and ANFIS as predictors of maintenance effort. Authors concluded that Adaptive Neuro Fuzzy Inference System is best. And hence Soft Computing Techniques can be successfully used for prediction of software maintenance effort.

Riaz *et al*. [17] tries to present the outcomes of systematic review on maintainability predictions and metrics. Out of 710 reviews 15 were selected for systematic review. Usually models designed for maintainability predictions were algorithmic techniques based but which algorithm is to be applied to which maintenance type, this type of distinction is not available. The authors suggest designing models that are more reliable and robust.

In another work Riaz et al. [18] presents the need of predicting maintainability for relational database-driven software applications. This paper presents the current scenario of maintainability predictions for relational database-driven software applications. The observations suggest that predicting maintainability for such software applications is not measured by using formal metrics but it should be predicted.

Maintainability Index has also been used to increase software maintainability by Sandeep Sharawat [19]. He used Li-Henry data for predicting Maintainability Index to train neural networks. He uses different training algorithms out of which Trainlm algorithm was found to be suitable to simulate the network for predicting Maintainability Index.

The applications of Artificial Neural Network were explored by Dash et al [20] to evaluate maintainability of the object oriented software. To predict maintainability they uses Multi Layer Perceptron neural network model. Authors considered maintenance effort as dependent variable and principal components of object oriented metrics as the independent variable. The results obtained from the Multi Layer Perceptron model were compared with other models and they found Multi Layer Perceptron model to be more useful than others.

2. ANALYSIS

On the basis of above survey and detailed analysis of the work done by different researchers in predicting maintainability for various application development domains, following tables can be designed:

TABLE 1

TRADITIONAL MAINTAINABILITY TECHNIQUES

S.No	Authors	Domain	Factors used to predict maintainability
1.	Khairuddin and Elizabeth	Procedure Oriented	Modularity, Readability, Programming Language, Standardization, Level of Validation and Testing, Complexity and Traceability.
2.	Fioravanti and Nesi	Object Oriented	Understanding, Addition and/or Deletion of facts and Modifications/Changes.
3.	Bandini *et al.*	Object Oriented	Design Complexity: Interaction level, Interface size and Operation argument complexity, Maintenance Task: Perfective and Corrective maintenance Programmer's ability: Correlation analysis
4.	Ahn *et al*.	Procedure Oriented	Engineer's skills, Technology characteristics, and Maintenance environment
5.	Ardimento *et al.*	Component Oriented	Component understanding and maintaining it.
6.	Kajko *et al*.	Procedure Oriented	Variable characteristics, Maturity level, Traceability characteristics and other common characteristics.

TABLE 2

SOFT COMPUTING BASED MAINTAINABILITY TECHNIQUES

S.No.	Authors	Domain	Factors used to predict maintainability
1.	Singh *et al*.	Object Oriented	Readability of Source Code, Documentation Quality, Understanding of Software and Average Cyclomatic Complexity
2.	Aggarwal *et al.*	Object Oriented	Average number of live variable, Average life span of variables, Average Cyclomatic Complexity and the Comments Ratio
3.	Aggarwal *et al.*	Object Oriented	Lack of Cohesion, Number of Children, Depth of Inheritance and others.
4.	Grover *et al.*	Component Oriented	Interaction Complexity, Reusability, Testability, Understandability and Trackability.
5.	Ardil *et al*.	Object Based	Based on Comparison: Least value of Mean Absolute Error, Root Mean Square Error, Logistic Model Tree and Simple Logistic algorithms.
6.	Kumar *et al*.	Aspect Oriented	Maintainability - Analyzability, Changeability, Stability and Testability.
7.	Kaur *et al.*	Object Oriented	Comparing Artificial Neural Network, Fuzzy Inference System and Adaptive Neuro Fuzzy Inference System to predict software maintenance effort.
8.	Sandeep Sharawat	Object Oriented	Maintainability Index, Li-Henry data.
9.	Dash *et al*.	Component Oriented	Multi Layer Perceptron neural network model

3. CONCLUSION

Maintainability has been a very important but expensive activity in software development. Latest development approaches like component-based, aspect-oriented and others also have this as a major challenge. Present paper conducts a detailed survey on the work done by different researchers in this direction. However, from the work it emerged that factors considered for assessing the maintainability were different in these research work. Moreover the quality models like ISO 9126 cannot be used as it is in various specialized domains. Therefore there is a strong need to analyze the maintainability in depth to identify the characteristics and sub-characteristics on which maintainability depends. Also, due to the rapid emerge in bio-inspired and other computing techniques like ANN, Fuzzy approach, Genetic Algorithm and others,

prediction is more accurate than the existing analytical techniques. We are currently working on detailed study of maintainability for specialized domains and on applying these techniques for better prediction on maintenance cost and efforts.

4. REFERENCES

[1]. Lientz, B. P., Swanson, E. B., 2000. Software Maintenance Management, Addison - Wesley Reading, MA.

[2]. Khairuddin, H., Elizabeth, K., 1996. A Software Maintainability Attributes Model, Malaysian Journal of Computer Science, Vol. 9, Issue 2, pp: 92-97

[3]. Fioravanti, F., Nesi, P., 2001. Estimation and Prediction Metrics for Adaptive Maintenance Effort of Object - Oriented Systems, IEEE Transactions on Software Engineering, Vol. 27, Issue 12, pp: 1062–1084.

[4]. Bandini, S., Paoli, F. D., Manzoni, S., Mereghetti, P., 2002. A support system to COTS based software development for business services , Proceedings of the 14th International Conference on Software Engineering and Know ledge Engineering, Ischia, Italy, Vol. 27, pp: 307–314.

[5]. Ahn, Y., Suh, J., Kim, S., Kim, H., 2003. The Software Maintenance Project Effort Estimation Model Based on Function Points, Journal of Software Maintenance: Research and Practice, Vol. 15, Issue 2, pp: 71-85.

[6]. Ardimento, P., Bianchi, A., Visaggio, G., 2004. Maintenance-oriented Selection of Software Components, Proceedings of 8th European Conference on Software Maintenance and Reengineering, pp: 115 –124.

[7]. Kajko-Mattsson, M., Canfora, G., Chorean, D., van Deursen, A., Ihme, T., Lehmna, M., Reiger, R., Engel, T., Wernke, J., 2006. A Model of Maintainability – Suggestion for Future Research, Proceedings of International Multi-Conference in Computer Science & Computer Engineering (SERP'06), pp: 436-441.

[8]. Singh, Y., Kaur, A., Sangwan, O. P., 2004. Neural Model for Software Maintainability, Proceedings of International Conference on ICT in Education and Development (AISECT 2004), pp: 1-11.

[9]. Aggarwal, K. K., Singh, Y., Chandra, P., Puri, M., 2005. Measurement of Software Maintainability Using a Fuzzy Model, Journal of Computer Sciences, Vol. 1, Issue 4, pp: 538-542.

[10]. Aggarwal, K. K., Singh, Y., Kaur, A., Malhotra, R., 2006. Application of Artificial Neural Network for Predicting Maintainability using Object -Oriented Metrics, Transactions on Engineering, Computing and Technology, Vol. 15, pp: 285-289.

[11]. Shukla, R, Mishra, A. K., 2008. Estimating Software Maintenance Effort - A Neural Network Approach, Proceedings of the 1st conference on India Software Engineering Conference, Hyderabad, India, pp: 107-112.

[12]. Grover, P. S., Kumar, R., Sharma, A., 2007. Few Useful Considerations for Maintaining Software Components and Component -Based Systems. ACM SIGSOFT Software Engineering Notes, Vol. 32, Issue 4, pp: 1-5.

[13]. Sharma, A., Kumar, R., Grover, P. S., 2009. Reusability Assessment for Software Components – a Neural Network Based Approach, Accepted for publication in International IEEE Conference (IACT' 09) to be held at Thapar University, Patiala from 26-28 March, 2009.

[14]. Ardil, Ebru., Ucar, Erdem., Sandhu, Parvinder. S., 2009. Software Maintenance Severity Prediction with Soft Computing Approach. International Journal of Computer Science and Engineering 3:2 2009.

[15]. Kumar, Avadhesh., Kumar, Rajesh., Grover, P. S. An Evaluation of Maintainability of Aspect – Oriented Systems: a Practical Approach. International Journal of Computer Science and Security, Volume (1): Issue (2).

[16]. Kaur, Arvinder., Kaur, Kamaldeep., Malhotra, Ruchika, 2010. Soft Computing Approaches for prediction of Software Maintenance Effort. International Journal of Computer Applications (0975 – 8887), Vol. 1 – No. 16.

[17]. Riaz, M., Mendes, E., Tempero, E. D.: A Systematic Review of Software Maintainability Prediction and Metrics. In: ESEM 2009, 2009, pp. 367-377.

[18]. Riaz, M., Mendes, E., Tempero, E. D.: Towards Predicting Maintainability for Relational Database-Driven Software Applications: Extended Evidence from Software Practitioners. International Journal for Software Engineering and its Applications. Vol. 5 No. 2, April 2011.

[19]. Sharawat, Sandeep.: Software Maintainability Prediction Using Neural Networks. International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622. Vol. 2, Issue 2, Mar-Apr 2012, pp. 750-755.

[20]. Dash, Yajnaseni., Dubey, Sanjay. Kumar., Rana, Ajay.: Maintainability Prediction of Object Oriented Software System by Using Artificial Neural Network Approach. International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Voloume-2, Issue-2, May 2012.