A Systematic Review of Software Maintainability Prediction

Deeksha Datyal\(^1\), Aman Kaushik\(^2\)
\(^1\)\(^2\)Dept. of CSE/IT, Baddi University of Emerging Sciences and Technology, Baddi
dshagun.datyal@gmail.com, aman.kaushik@baddiuniv.ac.in

Abstract: This paper presents the systematic review on effective maintainability prediction that can help to improve the overall software quality. Software maintainability is a key quality attribute that enhances functionality to meet the evolving needs of the customers. Maintenance phase begins after the delivery of product. Tracking the maintenance behavior of the software product becomes a complicated task. This is the reason that predicting the cost and risk associated with maintainance after delivery is extremely difficult. As the importance of software maintenance is increasing prediction of maintainability of a software has become critical that is widely being acknowledged by the researchers and practitioners. Therefore the focus of this study is on measurement and prediction of quality attribute of maintainability.

Keywords: Software Maintenance, Software Maintainability, SDLC, Maintainability Prediction.

1. INTRODUCTION
As society becomes more and more dependent on software and demands that new, more capable software be provided on short cycles, the need for maintainable, reliable software continues to increase. Software maintainability can be defined as the ease with which a software system can be modified and is also an important software quality attribute. Also the maintenance process is associated with this quality attribute, which has long been known to represent the majority of the costs of a Software Development Life-Cycle (SDLC) i.e. more than the sixty percent of the total cost. Therefore, the maintainability of a software system can significantly impact software costs and it is important to be able to forecast a software system’s maintainability so to effectively manage costs. The changes in the software should be met as the customer provides new requirements which may arise due to change in the technology, introduction of new hardware or enhancement of the features provided etc. This process of changing the software which has been delivered is called software maintenance. The amount of time, resource and effort spent on software maintenance is much more than what is being spent on its development. Thus, producing software that is easy to maintain may potentially save large costs and efforts. One of the main approaches in controlling maintenance cost is to monitor software metrics during the development phase. Research into software maintainability prediction includes proposing and validating maintainability predictors based on measurable factors that have a bearing on the software maintenance activity.

Prediction, commonly known as estimation, is an important part of project planning. Estimates can be made for projects or processes as well as products. When these are made for projects, these are called effort estimates and the process is called effort estimation or software cost estimation. When estimates are made to a maintenance process, the means of obtaining such estimates is called maintenance cost prediction or maintenance project effort estimation. In addition, estimates of quality attributes give a measurable value of the quality of the attribute that a software product possesses.

Maintenance and maintainability can be differentiated as: Software maintenance is defined as “the process of modifying a software system or component after delivery to correct faults, improve performance or other attributes, or adapt to a changed environment”. Software maintainability is defined as “the qualitative indication of the ease with which the existing software system or component can be corrected, adapted modified or enhanced to correct faults, improve performance, or adapt to a changed environment”. From the definitions it is clear that maintenance is the process performed as part of the SDLC whereas maintainability is the quality attribute associated with the software product. These are two inherently different but interlocked concepts.

Maintenance vs. maintainability represent process vs. quality attribute and their predictions are called process cost prediction vs. quality attribute measurement, respectively.

A software maintainability prediction model enables organizations to predict the maintainability of their software systems, thus providing a means to better manage their maintenance resources in addition to adopting a defensive design. This can then help in reducing the maintenance effort and therefore, reducing the overall cost and time spent on a software project. Maintenance cost can be controlled by monitoring software metrics during the development phase.

Uses of accurate prediction of software maintainability:
- It helps project managers in comparing the productivity and costs among different projects.

217 BUEST, Baddi RIEECE -2016
• It provides managers with information for more effectively planning the use of valuable resources.
• It helps managers in taking important decision regarding staff allocation.
• It guides about maintenance process efficiency.
• It helps in keeping future maintenance effort under control.
• The threshold values of various metrics which drastically affect maintainability of software can be checked and kept under control so as to achieve least maintenance cost.

Need of maintainability prediction:
The prediction of the expected number of hours that system or device will be in an inoperative or “down state” while it is undergoing maintenance is of vital importance to the user & cause of the adverse effect that excessive downtime has on mission success. Therefore, once the operational requirements of a system are fixed, it is imperative that a technique be utilized to predict its maintainability in quantitative terms as early as possible during the design phase. The prediction should be updated continuously as the design progresses to assure a high probability of compliance with specified requirements. A significant advantage of using a maintainability prediction procedure is that it highlights for the designer, those areas of poor maintainability which justify product improvement, modification, or a change of design. Another useful feature of maintainability prediction is that it permits the user to make an early assessment of whether the predicted downtime, the quality, quantity of personnel, tools and test equipment are adequate and consistent with the needs of system operational requirements.

Elements of maintainability prediction: Each maintainability prediction technique utilizes procedures which are specifically designed to satisfy its method of application. However, all maintainability prediction methods are dependent upon at least two basic parameters:
(a) Failure rates of components at the specific assembly level of interest.
(b) Repair time required at the maintenance level involved.
There are many sources which record the failure rate of parts as a function of use and environment. This failure rate, is expressed as the number of failures per unit of time. A typical measure is “failures per 106 hours. “ The major advantage of using the failure rate in maintainability prediction calculations is that it provides an estimate of the relative frequency of failure of those components which are utilized in the design. Similarly, the relative frequency of failure of components at other maintainable levels can be determined by employing standard reliability prediction techniques using parts failure rates. Failure rates can also be utilized in applicable regression equations for calculating the maintenance action time. Another use of the failure rate is to weight the repair times for various categories of repair activity, in order to provide an estimate of its contribution, to the total maintenance time.

2. RELATED WORK
Khoshgaftaar et al. [1] predicted software quality by using the neural networks as a tool. They classified the modules as either fault prone or non-fault prone in a large telecommunications system. They had also made a comparison between the ANN model and a non-parametric discriminate model and found that the ANN model had better predictive accuracy than the other one. Fenton and Neil [2] estimated various software defect prediction models by using size and complexity metrics for predicting defects. They compared fault-proneness estimation models and summarized that software quality is a crucial prerequisite in the system development. Muthana et al. [3] used the polynomial regression to establish the relationship between design level metrics and the corresponding maintainability of Industrial software. The results have shown that predicted values using polynomial regression were quite close to actual values. Fioravanti and Nesi [4] presented a metric analysis to identify which metrics would be better ranked for its impact on the prediction of adaptive maintenance for object-oriented systems. The model and metrics proposed have been validated against real data by using MLR (Multi linear Regression Analysis) Model. The validation has identified that the several metrics can be profitably employed for the prediction of software maintainability. Dagpinar et al. [5] also based their study on empirical data to establish the relationship between software metrics and its maintainability however instead of design level metrics of structure languages, the metrics were replaced by object oriented metrics. They recorded significant impact of two metrics i.e. direct coupling metric and size metric on software maintainability while other parameters like cohesion, inheritance and indirect coupling were not considered significant by them.

Thwin and Quah [6] used neural networks to build software quality prediction models. They proposed that maintainability can be estimated with the help of fuzzy model. They also proved empirically that the integrated measure of maintenance obtained from this fuzzy model has strong correlation with maintainance. Zhou and Leung [7] have used multivariate adaptive regression splines (MARS) for predicting object-oriented software maintainability in 2007. They compared the prediction accuracy of proposed model with four other prevailing models: multivariate linear regression (MLR), support vector regression (SVR), artificial neural network (ANN), and regression tree (RT) and stated that MARS is best model to be used as far as maintainability of prediction is concerned. Hu and Zhong [8] proposed a model based on neural network to predict software module risk. The learning vector quantization network used in their study has predicted software quality. Arvindar et al. [9] predicted the software maintenance effort by application of diverse soft computing techniques. Two commercial software products were taken as dataset and they observed that soft computing techniques are
useful for the construction of accurate models to speculate the maintenance effort. In their analysis maintenance effort was chosen as dependent variable and eight OO metrics as independent variable.

Ratra et al. [10] compared early prediction of fault prone modules in software design and for this they have applied clustering and neural network techniques. The performance of the two methods were measured based on their accuracy, the mean absolute error and root mean square error values. Their result signified that the performance of neural network approach is much superior to clustering based approach.

Malhotra, Chug [11] aimed at assessing the efficiency of different prediction models for prediction maintainability of web based systems using Object Oriented metrics. There results show that the GMDH(Group Method of Data Handling Model) is very helpful model in prediction of software maintainability. Their results signifies that error rate for GMDH model is 35.5 % in comparison with F3LBPNN whose error rate is 45.7 % and GRNN with error rate 54.7 %. Hence, GMDH is found to be more accurate and precise for predicting maintainability of web based applications.

Ping [12] used Hidden Markov Model (HMM) to define health index of a product in literature and suggested that it works as a weight on the process of maintenance behavior over a period of time.

3. CONCLUSION

This paper reviews various works done in the field of software maintainability prediction. A proper analysis of various techniques used by researchers has been presented and it was found that machine learning algorithms eases the process of estimation of software maintainability. Early prediction of faults becomes quite useful in enhancing the software quality and a lot of work has also been done for developing mathematical models and their dependency with factors affecting maintainability. In future Hybrid concepts of feature selections like Principle Component Analysis and self-learning techniques like Multi-level Perceptron can be utilized for the problem and the performance can be compared with their traditional counterparts.

REFERENCES