Nvert Analysis of Input Variables for Software Maintenance Cost Estimation Models

Chamkaur Singh[1], Dr. Neeraj Sharma[2], Dr. Narender Kumar [3], Mr. Jaswinder Singh[4]
[1]*Research scholar of I.K. Gujral Punjab Technical University, Jalandhar, Punjab, India*
[2]*Professor, GianJyoti Group of Institutions, Mohali, India*
[3]*Assistant Professor, HNB GarhwalUniversity,SrinagarGarhwalUttarakhand, India*
[4]*Assistant Professor, Chandigarh Group of Colleges, Landran, Punjab, India*
(E-mail: dhillon.chamkaur@gmail.com)

Abstract

Estimation of effort, schedule and cost for software projects is a difficult endeavor because it involves making predictions about the future by using historic project data and extrapolating it to prospect future values. To estimate software maintenance cost we have to study the factors which affect maintenance cost. In this paper we have discussed various technical and non-technical factors affecting the maintenance cost. According to various technical andnon-technical factors, input variables for maintenance cost estimation models have been decided. Further in this study we classify the maintenance cost estimation models are classified according to their input variables. So we tried to conclude that according to the input variables we have to decide the maintenance cost estimation model for accurate estimation of maintenance cost.

Introduction

Software maintenance is a very broad activity that includes improvements in capabilities, error correction, optimization, and removal of obsolete capabilities [1]. Because change is predictable, mechanisms must be developed for controlling, evaluation and making changes. So any work done to amend the software after it is in use is considered to be maintenance work. The purpose is to maintain the value of software over the period. The value can be improved by expanding the customer base, becoming easier to use, meeting additional requirements, more efficient and using newer technology

Software should be upgraded according the up-gradation in the technology or there may be internal issues in software that requires maintenance. 75% of total cost will be maintenance cost. There are many factors on which software maintenance cost depends. Maintenance doesn't mean just bug fixing. Actually, it is an effort to run the software as the user wants to run it. There are four major types of maintenance activities[10]:

1. Corrective:Modification performed in software after the delivery to correct the error and problems occurred in a program is known as corrective maintenance.

2. Adaptive:To cope with the changing environment, the changes made in the software come under adaptive maintenance activity.
3. Preventive:Potential faults are detected and corrected under the preventive maintenance. Preventive maintenance is required to face future challenges.
4. Perfective:Perfective maintenance is performed after getting the user feedback to improve the performance of a system. The goal of this activity is to satisfy the user.

FACTORS AFFECTING SOFTWARE MAINTENANCE COST

A. Non-Technical factors

1. **Developer Experience**:-Software maintenance depends upon the experience of a developer team. If the developer team is experienced then the system requirement can be defined more accurately and will decrease adaptive maintenance. If the demand is from new application are then it will increase adaptive maintenance workload[6].
2. **Team stability**:-If the team is stable then it is easy to handle maintenance activities. For a new engineer it would be time consuming to first the system and then perform any kind of changes in the existing system.[4]
3. **Time factor**: - Age of a system has direct link with maintenance effort and cost. As the system grows with passes of times, number of activities increases with age of software[3].Continuous maintenance activities gradually degrade the structure of the system as well as increase the complexity of the system. As the software gets older, due to changes in technology and increase in workload, more will be the maintenance cost.
4. **Environment factor**:-Software depends on external environment like rules, laws,workflow etc. So the change in external environment will increase the need of software maintenance. Adaptive maintenance will be more with change in external environment [7].
5. **Supporting environment**:-If there is change in hardware, operating system, network then software should be changed accordingly. The change in technology and the introduction of new hardware components will increase the maintenance cost.
6. **User requirements**:-As the user starts using the software his understandability increases. Then the user's expectations regarding the performance and functionality of the software increases. Therefore, perfection of the system needs maintenance in the existing system.

Non- Technical factors affecting maintenance cost	
Developer Experience	Developer's experience is inversely proportional to maintenance cost
Team stability	More stability lesser maintenance cost
Time factor	Increase in time factor, increases maintenance cost
Environment factor	Change in policies, rules and regulation can increase maintenance cost
Supporting environment	Change in technology increases maintenance cost
User requirements	User's expectations with passes of time increases maintenance cost

B. Technical factors

1.**Complexity of software**- if the structure of the software is complicated then it needs more time to read and understand the system. So to do any change takes more time which will increase work load and hence increase the maintenance cost [10].

2. **Capabilities of developer**- Programmer's experience also affect the maintenance cost. Good programming skills reduce the maintenance cost because of ease to read and understand the system. Similarly if testers use latest and advance methods then errors could be removed at the advance stages. So effective testing will reduce maintenance cost.

3. **Proper documentation**- if detailed information of the software system is available then it will help in understanding the system. Documentation should be systematic, clear and complete. Documentation helps in correcting errors and also to adapt a new technology or any change in the environment [17].

4. **Configuration change adaptability**- configuration management plays a vital role in managing software maintenance cost. Version control management, system integration and change control are main components of configuration control management [12].

5. **Improved programming techniques** - if the system is developed by using latest software engineering techniques and structured development methods then it is easy to maintain use modern programming techniques, which will reduced maintenance cost.

6. **Size of database**- Database maintenance is major components of software maintenance cost of software system. Bigger the size of database, more difficult to handle it. Maintenance workload increase with the increase in size and complexity of database.

Technical factors affecting maintenance cost	
Complexity of software	More the complexity of software system more will be maintenance cost
Capabilities of developer	Good programming skills reduce the maintenance cost
Proper documentation	Proper documentation can decrease the maintenance cost
Configuration change adaptability	Change in configuration increases maintenance cost
Improved programming techniques	Modern programming techniques, will reduced maintenance cost.
Size of database	Bigger the size of database, more difficult to handle it.

MAINTENANCE COST ESTIMATION MODELS

Many software maintenance cost estimation models have been developed and used to estimate the maintenance costs. These models are based on various maintenance activities like, corrections of errors, functional enhancements, technical evolution, and reengineering. They can be categories into three types:

i) Phase-level maintenance estimation models.
ii) Release-level maintenance estimation models.
iii) Task-level maintenance estimation models.

PHASE-LEVEL MODELS

These models are based on maintenance work for fixed period of time or for a certain phase of software. The usual maintenance work includes all those activities, which are performed during the use of software system after it's delivered. It involves corrections of faults, functional changes and enhancements, and technical improvements etc. The maintenance models like COCOMO, SLIM, PRICE-S, SEER-SEM, and Knowledge Plan are comes under this category. In these models, maintenance costs are usually depends on estimates produced when the cost of a new system is estimated. Thus, the size of the system is a main cost driver to estimate the maintenance effort. Most of these models use additional cost drivers that are specific to software maintenance. For example, software understanding (SU) and the level of unfamiliarity of the programmer (UNFM), two drivers for calculating sizing, is used by COCOMO. SEER-SEM uses change in size over time and maintenance rigidity as parameters for maintenance cost calculations [12].

It is very difficult to calculate maintenance cost at early stage because software system is based on many assumptions. Also there is no study published to check estimation accuracy of these models. Except COCOMO all the above mention models are proprietary and their details have not been fully available. These models consider adaption and reuse work same as in case of new software development but this is not always true and not validated empirically.

RELEASE-LEVEL MODELS

There is a set of maintenance cost estimation models that works on micro level rather than calculating maintenance cost for whole system. These models estimate maintenance cost of planned maintenance activities.This approach based on the data from the past releases and examines the changes to estimate the cost for the next release.

Regression Model discussed by Basili et al[19]., which involves maintenance activities like corrections of errors and enhancement. The model uses SLOC, a variable, which is measured as the sum of modified, add and deleted coding lines including blanks and comments.

Linear regression models introduced by Ramil and Lehman [11] also consider size as input variable cor cost estimation. This model considers maintenance activities like error correction, technical changes and functional enhancement etc. Basically this model depends on maintenance activities involved in growth of system.

More than one model can be calibrated to improve prediction of software maintenance cost. One such model is proposed by Caivano et al. This is based on reverse engineering concept input variables used by this model are SLOC, cyclomatic complexity number of changes to modules and changed modules. Calibration of models improves the accuracy of model. Many studies have been conducted and proved that model calibration improves prediction cost.

Maintenance activities like error correction, minor change, user's change requests, adding, modifying, deleting, improving software functionality and technical enhancement etc. , these types of activities are consider same as new development in which the size can be measured in source line of code . but researcher's suggests that these task types have different characteristics and they requires an appropriate estimation model.

Maintenance Function Point (MFP) model proposed by Niessink and van Vliet, uses the FPA procedure for enhancement to determine the functional point count. This model basically considers the functions affected by the changes. Another study conducted by Abran et al. suggested that functional size and maintenance cost has positive correlation. So functional size should be main input variable for maintenance cost estimation model. However author suggested that other factors like complexity of system should also be considered [19].

TASK-LEVEL MODELS

The task level models predict cost of maintenance activities that comes from change requests or error reports. These models used to estimate cost for small maintenance activities ranging from few hours to few days.

Sneed proposed a maintenance cost estimation model called Softcalc [6]. This model use Source Line of Code, data point, object points and functional points as input variables. Complexity of the system and project influence factor in considered in calculation maintenance cost.

Another way to calculate maintenance cost of software system is to divide change request into levels of difficulty and resources requirement. Briand and Basili developed a model that uses four major steps to estimate software maintenance cost.

- identifying predictable metrics,
- identifying significant predictable metrics,
- generating a classification function,
- Validating the model.

The inputvariables are divided into five intervals as described in table

Interval	Effort
1	Less than one hour
2	Between one hour and one day
3	Between one day and one week
4	Between one week and one month
5	above one month

According yo Briand and Basili' Models inputs can be predicted correctly only after the implementation of change request.[Basili 1997].

Basili et al. proposed a classification model that takes functional calls, number of exception and declaration statements as a input variables. This model is very useful tool because these input variables can be collected from component version.

Jorgensen tested eleven different models to estimate the cost of maintenance activities.[Jorgensen 1995]. All of the models use size as main input variables and four other factors such as change, mode, cause and confidence of maintainer.

In the following table we have summarized some software maintenance cost estimation models and their input variables.

Model/ Study	Maintenance Task	Input Metrics	Source
COCOMO	Major enhancement	SLOC, SU, UNFM,DM, CM, IM	Nguyen, 2010
SLIM	Corrective,Perfective,Adaptive	SLOC added and modified	Nguyen, 2010
SEER-SEM	Corrective,Perfective,Adaptive	SLOC,FP(Functional points), Annual change rate	Nguyen, 2010
PRICE-S	Major enhancement	SLOC, FP,UCCP of reuse code	Nguyen, 2010
Study by Sneed	Corrective,Perfective,Adaptive, Preventive	SLOC, FP, Objective points, Number of Errors, Complexity Index etc.	Sneed, 2004
Study by De Lucia and others	Corrective	SLOC, Number of tasks	De Lucia *et al.,* 2005
Study by Vu Nguyen	Functional enhancement, Error correction	SLOC of adaptive, reused, new module etc.	Nguyen, 2010
Study by Ren and others	Corrective	Weight of each technical and non-technical factor	Ren et al, 2011
Study by Marounek	Regular maintenance	Expert judgment	Marounek, 2012
Study by Ebert	adds, modifes, or deletes tasks or functions	Functional size change	Ebert et al. 2014
Study by Lai and others	Regular maintenance	user stories and development documents inspection activity	Lai et al. 2015

CONCLUSION

In this paper we have discussed various factors which affect the software maintenance cost estimation. we categories the software maintenance cost estimation model according the maintenance activities like, corrections of errors, functional enhancements, technical evolution, and reengineering. They can be categories into three types: Phase-level maintenance estimation models, Release-level maintenance estimation models, Task-level maintenance estimation models. So we conclude that according to maintenance activities we have decide the input variables for software maintenance cost estimation model then according to those input variables we have to choose cost estimation model to predict an accurate maintenance cost of a software system.

REFERENCES

1. Roger S. Pressman, Software Engineering: A Practitioner's Approach Seventh Edition, McGraw-Hill Higher Education, 2010
2. Ren YC, "Research on Software Cost Estimation and Its ExpertSystem," Doctor's degree of Liaoning Technical University, 2008.
3. Barry W. Boehm. Software Engineering Economics[M]. PearsonEducation. 1981.
4. Zhu JD, Qi CH, Hou XM, et al., "Estimating software cost based on grey theory," Computer Engineering and Applications, vol. 46, no.16, pp. 71-73, 2010.
5. Lai, S. (2015)."A Maintainability Enhancement Procedure For Reducing Agile Software Development Risk", International Journal of Software Engineering & Applications (IJSEA), Vol.6, No.4, pp. 29 – 40.
6. Sneed, H.M. (2004), "A cost model for software maintenance and evolution", Proceedings of IEEE 20th International Conference on Software Maintenance, Chicago, USA pp. 264 – 273.
7. Tang Q, " Approach to Improve the Software Cost EstimationAccuracy," Science Technology and Engineering, vol. 6, no. 4, pp.455-461, 2006.
8. Li SM, HeM, Yang D, et al., " Software Cost Estimation Methodand Application," Journal of Software, vol. 18, no. 4, pp. 775-795,2007
9. Accurately Estimate Your Software Maintenance Costs. Available at "http://www.galorath.com/index.php/software_maintenance_cost"
10. Banker, R. D., Datar, S. M., &Kemerer, C. F. (1986). Factors affecting software maintenance productivity : an exploratory study 1, 160–175.
11. Lehman's laws of software evolution. Available at "http://en.wikipedia.org/wiki/Lehman%s_laws_of_software_evolution"
12. Nguyen V., Boehm B.W., Danphitsanuphan P. (2010), "A Controlled Experiment in Assessing and Estimating Software Maintenance Tasks", APSEC Special Issue, Information and Software Technology Journal, 2010
13. T.N.Sharma, Anil Bhardwaj, Anita Sharma,(2011). A Comparative study of COCOMO II and Putnam models of Software Cost Estimation.(November)
14. Pragya Siddhi, Varun Kumar Rajpoot,(2012). A Cost Estimation of Maintenance Phase for Component Based Software.(May-June)
15. C.V.S.R SYAVASYA, (2006). An Approach to Find Maintenance Costs Using Cost Drivers of Cocomo Intermediate Model, (December).
16. Sharma, T. N., Bhardwaj, A., &Kherwa, G. R. (2012). Statistical Analysis of various models of Software Cost Estimation, 2(3), 683–685.
17. YongchangRen ; Tao Xing ; Xiaoji Chen ; Xuguang Chai, (2011). Research on Software Maintenance Cost of Influence Factor Analysis and Estimation Method. Intelligent Systems and Applications (ISA), IEEE conference publications. (May)

18. Marounek P. "Simplified Approach to effort estimation in software maintenance", University of economic, Prague, Faculty of information and statistics, Journal of systems integration, 2012: 51-63.
19. Vu Nguyen, (2010). Improved size and effort estimation models for software maintenance.(December)