International Journal of Advances in Computer Science & Its Applications
Volume 6 : Issue 1 [ISSN 2250-3765]
Publication Date : 18 April, 2016

Analysis of Development Productivity in Information System Adaptive and Perfective Maintenance

Kai Huang, Xiaoying Kong, Li Liu

***Abstract*—This paper presents a development productivity model for Information System (IS) adaptive/perfective maintenance process. The modelling approach is from economic perspective. The productivity model considers the economic value of the maintenance activity, pre-committed fixed cost and variable cost consumed in adaptive/perfective maintenance. Factors that influence the productivity are analysed using simulation. The simulation provides a tool for IS project managers to tune the project parameters to obtain the optimal productivity in adaptive/perfective maintenance.**

***Keywords*—economic model, maintenance, productivity, software maintenance management, simulation**

I. Introduction

The lifecycle of an Information System (IS) includes new application development, implementation and maintenance. After a new application has been developed and implemented, the system is handed over to users for operation. On user side, during the operation, software modification activities continue while leaving the IS primary functions intact [1]. This modification process is the IS software maintenance phase.

One of IS project managers' tasks is to estimate the efforts in IS development/maintenance and ensures the development/maintenance efforts are conducted productively. One common approach to estimating efforts is to analyse the structure and components of the efforts, which involves quantifying the structure and elements. Research shows that the efforts in new IS development phase and maintenance phase are different [2-7]. As a result, the structure and estimation models for new development efforts and maintenance efforts are very different. Efforts estimation models have been widely used in new software development phase [1]. Modelling of maintenance efforts is a relatively under explored area of research.

The efforts estimation modelling approaches in literatures can be classified into two types: static cost model and dynamic analytical model.

Static cost models analyse the development efforts using direct software system components and the efforts directly consumed in these components. Examples of these types of models include Constructive Cost Model (COCOMO) [1,8], and Function Point Analysis [9-11]. In these models, time parameters are not included. All parameters are unchanged over time. The other type of model is dynamic analytical model. This type of model considers economic value changes in IS development. The input and output values of model change over time. Future values are discounted to get present value. Uncertainty of project values is considered. Dynamic analytical model analyses the dynamic relationships among these economic values of project, uncertainty and project time [12].

In this paper, we use dynamic analytical model to analyse the IS maintenance efforts. The focus is on the productivity of IS maintenance. There are three types of maintenance: corrective, adaptive, and perfective. Productivity model for corrective maintenance was explored in literature [13]. Considering the lack of research on adaptive and perfective maintenance, this paper will model the development efforts in adaptive and perfective processes using dynamic analytical modelling approach.

This paper is organized as follows. Next section will review the related work and propose the economic model for adaptive/perfective maintenance. Then productivity values are analysed, and influence factors are analysed through simulation. Finally, conclusions are drawn and future research directions suggested.

II. Related work and three types of maintenance

In this section, literature on system development efforts modelling approaches is reviewed first before focusing on the types of maintenance and influence factors to maintenance.

A. *Related work*

To understand the development efforts, productivity has been a major metric to measure the effectiveness of system development effort. IEEE standard defines productivity as "the ratio of work product to work effort" [14]. We can use the ratio of "an output primitive" to "its corresponding primitive" of the software for productivity equation [14].

Beohm defined the software development productivity as the ratio of "delivered source instruction" (DSI) and "the number of man-months (MM) estimated for the software development phase of the lifecycle" [1]. The measure of DSI is equivalent to Line of Code (LOC) measurement. To extend the basic efforts/cost model to maintenance phase,

Kai Huang
University of Technology, Sydney
Australia

Xiaoying Kong
University of Technology, Sydney
Australia
Xiaoying.kong@uts.edu.au

Li Liu
The University of Sydney
Australia

Beohm identified influence factors that affect the efforts, cost or productivity, proposed a factor Annual Change of Traffic (ACT) to represent annual maintenance size ratio. ACT is applied to the basic efforts MM as the estimate efforts [1].

In Function Point Analysis (FPA) approach, instead of using line of code (LOC) to measure the output primitive, FPA identifies outputs, inquiries, inputs, internal files, and external interfaces from functional user requirements. Total function points are calculated using these five types of points with assigned complexity [9,11,14].

Regression model has been frequently used to establish the relationship between efforts, cost, productivity and related influence factors. Linear model, linear multivariable model, and exponential regression model have been used to analyse development and maintenance efforts or productivity. The coefficients, factor parameters are established by parameter estimation approach or none/semi parametric estimation approaches [6,15].

Artificial neural networks approach uses back propagation to represent nonlinear regression models [16]. Neural networks approach could use a large number of parameters to model complex relations between variables in development and maintenance efforts or productivity [4].

These approaches are the major approaches in the type of static cost model. Static models do not include the dynamic time factors in efforts/cost modelling. From economic perspective, the values of these input and output primitive are related to present value, future value, discount rate, and uncertainty of the values over time. To consider these economic factors, dynamic analytical modelling approach uses analytical equations to model the relationship among economic values, cost/efforts, discount rate, and uncertainty level of values with the function of project or activity time. These analytical equations also form a simulation tool to view the dynamic relationships virtually when model parameters and influence factors change over project time[12]. In this research, we will apply dynamic analytical modelling approach to analyse the IS software maintenance economic model.

Due to different characteristics in software maintenance activities, maintenance process can be classified into three major types: corrective maintenance, adaptive maintenance, and perfective maintenance. Research shows that the efforts and productivity in different types are different [15,17]. To further understand the economic model of productivity in IS software maintenance process, we understand the different types of maintenance first, and then present the economic model of productivity for each type.

B. *Types of maintenance and influences factors*

The input cost and output value of different maintenance type is dramatically different, so the maintenance productivity of all maintenance types is not the same. Based on the motivation what causes maintenance, maintenance activities can be divided into three types: corrective, adaptive, and perfective maintenance.

The most common maintenance activity is corrective maintenance. Three kinds of failure are solved in corrective maintenance [1,17,18]. The first failure is processing failure, the bugs of the software system. The second failure is performance failure. The third failure is implementation failure [1,18].

Adaptive maintenance is often caused by system environmental change[1,17,18]. Environmental change includes change in data environment and change in processing environment. **Perfective maintenance** is the activity to enhance the system[1,17,18]. It includes the software enhancement of efficiency, performance or maintainability. Due to the different characteristics in maintenance type, the effort devoted to each type is different. The maintenance efforts in different maintenance types are statistically distributed as: corrective: 14%, adaptive: 5%, perfective: 61%, and others: 20% [15].

In literature, influences to system development and maintenance have been identified[1,19]. Boehm grouped the factors into product attributes, computer attributes, personnel attributes, and project attributes. For the maintenance process, the influence factors can be classified into two groups: technical factors and non-technical factors as in Table 1 [13].

TABLE I. INFLUENCE FACTORS IN MAINTENANCE PROCESS

Group	Influence Factors	
Technical factors	Quality of system original program Code reusability Modern programming practice Quality of system documentation Adequacy of system design specification Maintenance personnel experience	Database size Maintenance personnel abilities on maintenance Maintenance personnel abilities on configuration management Maintenance tool availability Software system complexity
Non-Technical factors	User demands for enhancements, extents Software reliability request Usage time	Maintenance personnel stability Environmental change

Influences to corrective maintenance productivity model have been studied in previous research. In this paper, we will focus on adaptive perfective maintenance.

III. Economic Model For Information System Adaptive / Perfective Maintenance

A. *Productivity model for IS maintenance*

To analyse the economic values, efforts or productivity in IS software adaptive/perfective maintenance, we review the economic model for general maintenance process first. We use the productivity metric to model the effectiveness of the maintenance process in economic perspective.

In maintenance process, "productivity" measures the outputs, inputs of maintenance activities, and their ratio in economic values. In dynamic analytical modelling approach, the output of a maintenance activity is the total economic value of this activity. The inputs of the maintenance are the total cost of pre-committed fixed cost plus the variable cost consumed to this activity. Example of fixed cost includes the cost for educating maintenance developers. Variable cost includes salaries paid for system developers in this activity only [13]. The productivity metric can be presented as follows [12,13]:

$$\mathrm{Pr}\,oductivity\_Eco = \frac{S_{MT} \times T}{K_{MT} + C_{MT} \times T} \qquad (1)$$

where *Productivity_Eco* is the metric of productivity of the IS system maintenance process,

SMT is the economic value of the maintenance in a unit time, *KMT* is the pre-committed fixed cost, *CMT* is the variable cost in a unit time, and *T* is the total time of the maintenance activity.

The following equation represents the relationship among these economic values in equation (1)[20]:

$$C_{MT} = S_{MT} N(d_1) - K_{MT} e^{-rT} N(d_2) \qquad (2)$$

where *N*(*d*1) and *N*(*d*2) are the cumulative probability distribution functions for a standardized normal random variable. *d*1 and *d*2 are calculated as below [20]:

$$\begin{aligned} d_1 &= \frac{\ln(S_{MT}/K_{MT}) + (r + \sigma^2/2)T}{\sigma\sqrt{T}} \\ d_2 &= \frac{\ln(S_{MT}/K_{MT}) + (r - \sigma^2/2)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T} \end{aligned} \qquad (3)$$

where *r* is the discount rate of return, σ is the uncertainty rate. Equations (1) (2) and (3) are the economic model for maintenance productivity. This general model can be used in three types of maintenance. To analyse the influences in this economic model, we separate types of maintenance due to the different influences. The influence analysis for corrective maintenance has been explored in literature, in this paper we focus on the analysis for adaptive/perfective maintenance.

B. *Analysis of productivity values of adaptive/perfective maintenance*

To analyse the productivity using economic model to adaptive/perfective maintenance process, the input, output values and influence factors to these values will be identified and analysed in the following sections.

1) Maintenance output – the economic values of adaptive/perfective maintenance

Adaptive maintenance and perfective maintenance are similar from the perspective of modelling the economic productivity, but they are different from corrective maintenance. There are two reasons. Firstly the corrective maintenance is consecutive, while adaptive and perfective maintenance are not consecutive. Corrective maintenance is a daily routine work. Adaptive and perfective maintenance concentrate in a certain period. Only when environment changes, the customer has new demands, or system problems accumulate to certain degree, adaptive and perfective maintenance are necessary. Secondly the corrective maintenance only decreases the depreciation speed of the original system, while adaptive and perfective maintenance can keep or increase the value of the original system.

We suppose the original value of the system at the time development just finishes is *SDev*, the value of the system depreciates based on the following exponential accelerated depreciation model, the value of the system is as below:

$$S_{No\_MT}(T_{bef}, \alpha_{No\_MT}) = S_{Dev} \times \alpha^{T_{bef}}{}_{No\_MT} \qquad 0 < \alpha_{NO\_MT} < 1 \ (4)$$

where *SNo_MT* is the economic value of the system in a unit time without maintenance; *αNo_MT* is the depreciation rate without maintenance; and *Tbef* is system running time before adaptive/perfective maintenance.

With adaptive / perfective maintenance, new value is added in, and the depreciation rate may change too. Usually the depreciation rate becomes higher because the system is better, but if adaptive maintenance is not successful, it may be lower. We assume the new added value is *SAdd_val*, the depreciation rate of the system with adaptive/perfective maintenance is *αAdaPec_MT* , and the value of the system with maintenance is as below:

$$\begin{aligned} & S_{WithAdaPec\_MT}(S_{Dev}, S_{Add\_Val}, \alpha_{AdaPec\_MT}, \alpha_{No\_MT}, T_{bef}, T_{MT}) \\ & = (S_{Dev} \times \alpha_{No\_MT}^{T_{bef}} + S_{Add\_Val}) \times \alpha_{Corr\_MT}^{T_{MT}} \qquad 0 < \alpha_{AdaPec\_MT} < 1 \end{aligned} \qquad (5)$$

where *αAdaPec_MT* is the depreciation rate with adaptive/perfective maintenance, *SWithAdaPec_MT* is the unit value of the system with adaptive perfective maintenance in a unit time; and *TMT* is the system maintenance time in adaptive/perfective maintenance.So the economic value of the adaptive / perfective maintenance is:

$$\begin{aligned} & S_{MT}(S_{Dev}, S_{Add\_Val}, \alpha_{AdaPec\_MT}, \alpha_{No\_MT}, T_{bef}, T_{MT}) \\ & = S_{WithAdaPec\_MT} - S_{No\_MT} \\ & = (S_{Dev} \times \alpha_{No\_MT}^{T_{bef}} + S_{Add\_Val}) \times \alpha_{AdaPec\_MT}^{T_{MT}} - S_{Dev} \times \alpha_{No\_MT}^{T_{bef}+T_{MT}} \end{aligned} \qquad (6)$$

2) Maintenance input - the cost of the adaptive/perfective maintenance

Adaptive maintenance may cause to buy new devices and software. Perfective maintenance need to buy maintenance tools too. The most cost is the payment to maintainers. The cost is divided into two parts – pre-committed fixed cost and variable cost.

The ratio of the above maintenance output and input will be used to calculate the productivity metric. There will be factors that influence the productivity. Influence analysis will be presented in the following sections.

IV. Influence Analysis for Adaptive / Perfective Maintenance Using Simulation

To analyse the influences to the productivity model, we use simulation based on the analytical equations of the economic model. In simulation, we will observe how the productivity changes when we change the parameters in the productivity economic model. We select a number of influence factors to analyse and present in this paper. The following parameters will be set up for analysis in the simulation:

S_{MT}: economic value of adaptive/perfective maintenance in a unit time (unit: dollar)

K_{MT}: pre-committed fixed cost of adaptive/perfective maintenance (unit: dollar)

C_{MT}: variable cost of adaptive/perfective maintenance in a unit time (unit: dollar)

T_{MT}: IS system maintenance time (unit: month)

Productivity_Eco: Economic productivity for adaptive/perfective maintenance process

r: discount rate of return

σ: rate of uncertainty

The selected influences factors include: code reusability F_{Code_reus} and modern programming practice F_{Mode_prac}.

A. Analysis of the basic economic productivity of adaptive/perfective maintenance without influence factors

To understand different influences to the economic productivity in adaptive/perfective maintenance activities, firstly we analyse the basic economic productivity without influence of factors. We set up a simulation using the economic productivity model and set values as follows: S_{Dev}=10000, S_{Add_Val}=2000, α_{No_MT}=0.8, α_{AdaPec_MT} =0.85, K_{MT}=1000, T_{bef}=6, r=0.03, σ=0.25. T_{MT} changes from 1 to 19 with the interval of 1. Fig. 1 shows how *Productivity_Eco* changes with T_{MT}. The simulation results indicate that for adaptive/perfective maintenance without influence of factors, the economic productivity at first inclines then declines with time.

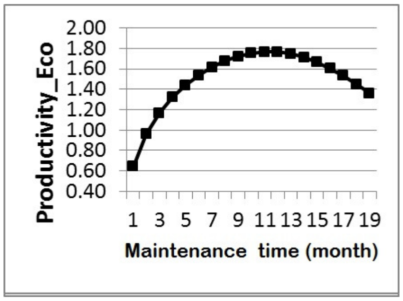


Figure 1. Productivity changes with time T_{MT} in basic adaptive/perfective maintenance without influences of factors

B. Analysis of influence factor - code reusability F_{Code_reus}

Code reusability is important for adaptive maintenance. It can shorten the transfer period from one platform to another, decrease the maintenance cost, and make the system more standard and higher quality. Working with higher code reusability IS system will save maintainers' effort. Code reusability F_{Code_reus} influences the economic productivity through 2 parameters of productivity model in adaptive / perfective maintenance: K_{MT} and T_{MT}. High code reusability decreases the pre-committed cost K_{MT}. It also shortens the maintenance time T_{MT}. We use the regression analysis approach to analyse the relationship between F_{Code_reus} and K_{MT} as follows:

$$K_{MT} = \beta_{KC0} + \beta_{KC1} \times F_{Code\_reus} + \varepsilon_{KC} \qquad \beta_{KC1} < 0 \qquad (7)$$

$$T_{MT} = \beta_{TC0} + \beta_{TC1} \times F_{Code\_reus} + \varepsilon_{TC} \qquad \beta_{TC1} < 0 \qquad (8)$$

where β_{KC0} and β_{TC0} are constant, β_{KC1} and β_{TC1} are the scaling factors, ε_{KC} and ε_{TC} are the error terms in the regression models respectively.

To analyse the influence of F_{Code_reus} to productivity, in simulation we set: S_{Dev}=10000, S_{Add_Val}=2000, α_{No_MT} =0.6, α_{AdaPec_MT} =0.8, T_{bef}=6, r=0.03, σ=0.25, and β_{KC1}=-400, β_{KC0}=1400, β_{TC1}=-10, and β_{TC0}=16.

The value of code reusability F_{Code_reus} varies from 0.6 to 1.5 with the interval of 0.05. The bigger the value is, the higher the code reusability is. We observe the change of the economic productivity *Productivity_Eco* with the change of F_{Code_reus}. The simulation results are illustrated in Fig. 2.

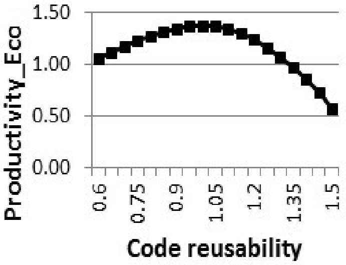


Figure 2. *Productivity_Eco* changes with code reusability F_{Code_reus}.

High code reusability can increase the unit value of adaptive/perfective maintenance. With code reusability rises from low to standard level or average level, the productivity goes up to the peak. When code reusability continues to rise from average to very high, the productivity bends down and declines. So from productivity viewpoint, we shouldn't pursuit to very high code reusability. To keep high code reusability in maintenance, the maintainers should be very careful not to destroy the code block complement. Maintainers may put more effort on keeping reusability rather than system efficiency.

C. Analysis of the influence factor - modern programming practice F_{Mode_prac}

Modern programming practice follows standard development and maintenance process framework, uses modularized and hieratical programming design. The system design structure is clear and easily adjusted. Applying modern programming practice in maintenance task can shorten maintainers' time and effort on read/understand the present software program. Modern programming practice may need more pre-committed cost on more detailed programming design, better maintenance task management and more reasonable decomposition of maintenance work, and maintainers training in advance. So F_{Mode_prac} is positively correlated with K_{MT} and negatively correlated with T_{MT}. We use the following regression model to analyse their relationship:

$$K_{MT} = \beta_{KM\,0} + \beta_{KM\,1} \times F_{Mode\_prac} + \varepsilon_{KM} \qquad \beta_{KM\,1} < 0 \quad (9)$$

$$T_{MT} = \beta_{TM0} + \beta_{TM1} \times F_{Mode\_prac} + \varepsilon_{TM} \qquad \beta_{TM1} < 0 \qquad (10)$$

where β_{KM0} and β_{TM0} are constant, β_{KM1} and β_{TM1} are the scaling factors, and ε_{KM} and ε_{TM} are the error terms in the regression models respectively.

In simulation we set: S_{Dev}=10000, S_{Add_Val}=2000, α_{No_MT} =0.6, α_{AdaPec_MT} =0.8, T_{bef}=6, r=0.03, σ=0.25, and β_{KM1}=400, β_{KM0}=600, β_{TM1}=-10, and β_{TM0}=16. Fig. 3 shows that the result is similar to that of code reusability. The simulation result shows that applying too much modern programming practice will decrease the productivity. It is not suitable to pursuit extreme modern programming practice.

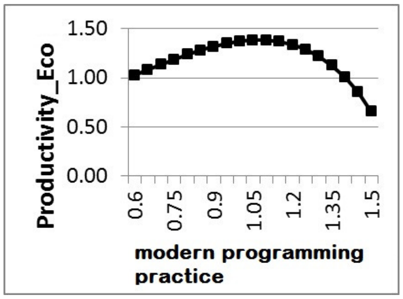


Figure 3. Productivity changes with modern programming practice

V. Conclusion and Further Research

In this research, we present an information system maintenance productivity model for adaptive and perfective types of maintenance. The modelling is based on dynamic analytical modelling approach. A selected number of influence factors are analysed using simulation approach for its impacts on productivity. These factors include code reusability, modern programming practice, user demands for enhancement and environmental change for adaptive/perfective maintenance. For each influence factor, simulation is designed to observe the changes of economic productivity when changing the parameters in economic model. Simulation results illustrate that for each influence factor, an optimal productivity value exists in adaptive/perfective maintenance activity. Therefore this simulation approach provides a tool for IS project managers for tuning the parameters to obtain the optimal productivity in economic perspective.

References

[1] Boehm, B. W. 1981. "Software Engineering Economics." Prentice-Hall Advances in Computing Science and Technology Series, Englewood Cliffs: Prentice-Hall, 1981 1 (1981).

[2] Boeing, Co. 1979 "Software Cost Measuring and Reporting." U.S. Air Force-ASD Document D 180-22813-1 (1979).

[3] Lientz, Bennet P., and E. Burton Swanson. 1981. "Problems in Application Software Maintenance." Communications of the ACM 24,11 (1981): 763-69.

[4] Jorgensen, Magne. 1995 "Experience with the Accuracy of Software Maintenance Task Effort Prediction Models." Software Engineering, IEEE Transactions on 21,8 (1995): 674-81.

[5] Banker, Rajiv D., and Sandra A. Slaughter. 1997. "A Field Study of Scale Economies in Software Maintenance." Management Science 43,12 (1997): 1709-25.

[6] De Lucia, A., Pompella, E., & Stefanucci, S. 2003. Assessing the maintenance processes of a software organization: an empirical analysis of a large industrial project. Journal of Systems and Software, 65(2), 87-103.

[7] Nguyen, V., Boehm, B., and Danphitsanuphan, P. 2011. "A Controlled Experiment in Assessing and Estimating Software Maintenance Tasks." Information and software technology 53,6 (2011): 682-91.

[8] Boehm, B.W., Horowitz, E., Madachy, R., Reifer D., Clark B.K., Steece B., Brown A.W., Chulani S & Abts C., 2000 "Software Cost Estimation with COCOMOII", Prentice Hall, 2000, ISBN 0130266922

[9] Albrecht, A. J., and Gaffney J. J. (1983) "Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation." IEEE Transactions On Software Engineering SE-9,6 (1983):639-48.

[10] ISO/IEC, IS 14143-1: 2007 - - Information Technology - Software Measurement - Functional Size Measurement – Part 1: definition of concepts, International Organization on Standardization, February 2007

[11] COSMIC, "Measurement Manual, Version 3.0.1, Common Software Measurement International Consortium, 2009. URL: www.cosmicon.com

[12] Liu, L., Kong, X. & Chen, J. 2015. 'How Project Duration, Upfront Costs And Uncertainty Interact And Impact On Software Development Productivity? A Simulation Approach', International Journal of Agile Systems and Management.

[13] Huang, K., Kong, X., & Sandrasegaran, K. 2014. Analysis of the influence to productivity of software corrective maintenance using an economic model. In proceedings of Information Technology and Electronic Commerce (ICITEC), 2014 International Conference on (pp. 117-121). IEEE.

[14] IEEE 1992 IEEE Standard for Software Productivity Metrics 1045

[15] Basili, Victor, Lionel Briand, Steven Condon, Yong-Mi Kim, Walcélio L. Melo, and Jon D. Valett. 1996. "Understanding and Predicting the Process of Software Maintenance Release." 1996. IEEE Computer Society.

[16] Hertz, John, Anders Krogh, and Richard G. Palmer. 1991. "Introduction to the Theory of Neural Computation. 1991." Addison-Wesley, Redwood City, CA.

[17] IEEE 2010 ISO/IEC/IEEE 24765:2010(E) "Systems and software engineering - Vocabulary"

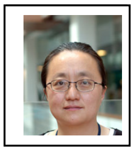
[18] Swanson, E. Burton. 1976. " The Dimensions of Maintenance." 1976. IEEE Computer Society Press.

[19] de, Barros Sampaio, Suzana Cândido, et al. 2010. A review of productivity factors and strategies on software development. Software Engineering Advances (ICSEA), 2010 Fifth International Conference on. IEEE, 2010.

[20] Chen, J. 2005 THE PHYSICAL FOUNDATION OF ECONOMICS, An Analytical Thermodynamic Theory, World Scientific Publishing Co. 2005.

About Author (s):

Kai Huang is a system engineer in Quantium Group. His research interests include network engineering, IT security and infrastructure automation.



Xiaoying Kong is currently a Senior Lecturer at University of Technology, Sydney. Her interests include software engineering and control theory.

Li Liu is Senior lecturer in project management at the University of Sydney. He has decade of working experience in systems engineering, project management, and business research.