
Analytical model of information system development productivity in adaptive and perfective maintenance phases

Kai Huang and Xiaoying Kong*

Faculty of Engineering and Information Technology,
Centre for Real-time Information Networks,
University of Technology, Sydney,
Sydney, Australia
Email: kai.huang-1@student.uts.edu.au
Email: xiaoying.kong@uts.edu.au
*Corresponding author

Li Liu

Faculty of Engineering and Information Technologies,
The University of Sydney,
Sydney, Australia
Email: Li.liu@sydney.edu.au

Abstract: This paper presents an analytical model for information system (IS) maintenance productivity in adaptive and perfective phases. The modelling approach is from economic view point. The productivity model considers the economic value of the maintenance phase, pre-committed fixed cost and variable cost consumed in adaptive/perfective maintenance. Influence factors to 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 phases.

Keywords: analytical model; economic model; maintenance; productivity; IS maintenance management; software engineering; simulation.

Reference to this paper should be made as follows: Huang, K., Kong, X. and Liu, L. (2018) 'Analytical model of information system development productivity in adaptive and perfective maintenance phases', *Int. J. Industrial and Systems Engineering*, Vol. 29, No. 3, pp.338–351.

Biographical notes: Kai Huang is a System Engineer in Quantum Group. He holds two Master's in Information Technology from University of Technology, Sydney. He has many years of industrial experience in IT administration and post-sale technical support. His research interests include network engineering, IT security and infrastructure automation.

Xiaoying Kong is currently a Senior Lecturer in the Faculty of Engineering and Information Technology, University of Technology, Sydney. She received her BE and ME in Control Engineering from Beijing University of Aeronautics and Astronautics. She received her PhD in Mechatronic Engineering from the University of Sydney. She has many years of industrial experience in aeronautical engineering and commercial web development. Her research

interests include control theory, sensor technologies, data fusion, robotics, software engineering, and web technologies.

Li Liu is currently a Senior Lecturer in Project Management in the University of Sydney. He earned a Doctorate from UNSW (Australia) and a BE in Electrical Engineering. He has one decade of work experience in electrical/systems engineering, project management, research and consulting. His research interests include management of infrastructure projects, IT/IS project management and control theory. He has published in *IEEE Transactions on Engineering Management*, *Project Management Journal*, *Journal of Information Technology* and a number of other international journals and conferences.

This paper is a revised and expanded version of a paper entitled 'Analysis of development productivity in information system adaptive and perfective maintenance' presented at Third International Conference on Advances in Information Processing and Communication Technology – IPCT 2015, Rome, Italy, 10–11 December 2015.

1 Introduction

The software development lifecycle of an information system (IS) includes new IS application development, IS implementation, operation and maintenance phases. 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 (Boehm, 1981). 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 (Boeing, 1979; Lientz et al., 1981; Jorgensen, 1995; Banker and Slaughter, 1997; de Lucia et al., 2003; Nguyen et al., 2011). 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 (Boehm, 1981). 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) (Boehm, 1981; Boehm et al., 2000; Moazeni et al., 2014; Rosa et al., 2014), and function point analysis (FPA) (Albrecht and Gaffney, 1983; ISO/IEC, 2007; COSMIC, 2009). 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.

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 (Huang et al., 2014). 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 organised 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.

2 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.

2.1 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’ (IEEE, 1992). We can use the ratio of ‘an output primitive’ to ‘its corresponding primitive’ of the software for productivity equation (IEEE, 1992).

Boehm 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” (Boehm, 1981). The measure of DSI is equivalent to line of code (LOC) measurement. To extend the basic efforts/cost model to maintenance phase, Boehm 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 (Boehm, 1981).

In FPA approach, instead of using 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 (Albrecht and Gaffney, 1983; ISO/IEC, 2007; COSMIC, 2009).

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 (Basili et al., 1996; de Lucia et al., 2003).

Artificial neural networks approach uses back propagation to represent nonlinear regression models (Hertz et al., 1991). Neural networks approach could use a large

number of parameters to model complex relations between variables in development and maintenance efforts or productivity (Jorgensen, 1995).

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 primitives 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 (Liu et al., 2015). In this research, we will apply dynamic analytical modelling approach to analyse the IS software maintenance economic model.

Owing 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 (Basili et al., 1996; IEEE, 2010). 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.

2.2 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 (Swanson, 1976; Boehm, 1981; IEEE, 2010). The first failure is processing failure, the bugs of the software system. The existence of processing failure causes wrong operation of the system. It may break down, give wrong output or be unstable. The second failure is performance failure. It happens when the system cannot satisfy the specified performance limitation. For example, in the banking system, one transaction should be completed in a certain time; otherwise the transaction will be cancelled. If the delay is caused by slow program other than business reasons, performance failure happens. The third failure is implementation failure. For example, inconsistencies or incompleteness in the detailed design, derived from the functional specifications, may be present. Implementation failure erodes the quality of software system. Corrective maintenance deals with above three types of failure. If system failures do not exist, corrective maintenance will not be carried out. So the effort devoted into corrective maintenance is highly related to the quality of development (Swanson, 1976; Boehm, 1981).

Adaptive maintenance is often caused by system environmental change (Swanson, 1976; Boehm, 1981; IEEE, 2010). Environmental change includes change in data environment and change in processing environment. For example, a change of the data structure in database, a change in the classification code system associated with a particular data element. Examples of processing environmental change include the installation of new system hardware, or the transfer of the system to a new platform.

Adaptive maintenance deals with these two environmental changes. It enhances the portability of the system and avoids the system invalidation in rapid changing environment. It may lengthen the lifetime of the system (Swanson, 1976; Boehm, 1981).

Perfective maintenance is the activity to enhance the system (Swanson, 1976; Boehm, 1981; IEEE, 2010). It includes the software enhancement of efficiency, performance or maintainability. Program inefficiency can be caused by poorly designed algorithms, non-optimised embedded multi-layers loop statement. Solving this kind of problem will accelerate the system operation. Examples of performance enhancement include: adding new function, improvement of the interactivity with users, improvement of readability and understanding of the running system function result. Enhancement of maintainability can be implemented by adding more accurate program comments, using professional maintenance software tools, and changing the programming language to object-oriented language. Perfective maintenance aims to make the system a more perfect design implementation. It improves the customer experience, maintainer experience, and the system operation. Perfective maintenance is carried out based on the comparison of improvement value and costs input into the maintenance (Swanson, 1976; Boehm, 1981; IEEE, 2010).

Owing 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% (Basili et al., 1996)

In the literature, influences on system development and maintenance have been identified (Boehm, 1981; de Barros Sampaio et al., 2010). 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.

Table 1 Influence factors in maintenance process

<i>Group</i>	<i>Influence factors</i>
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

Source: Huang et al. (2014)

3 Analytical economic model for adaptive/perfective maintenance productivity

3.1 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 (Huang et al., 2014). The productivity metric can be presented as follows (Liu et al., 2015; Huang et al., 2014):

$$Productivity_Eco = \frac{S_{MT} \times T}{K_{MT} + C_{MT} \times T} \quad (1)$$

where $Productivity_Eco$ is the metric of productivity of the IS system maintenance process, S_{MT} is the economic value of the maintenance in a unit time, K_{MT} is the pre-committed fixed cost, C_{MT} 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) (Chen, 2005):

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

where $N(d_1)$ and $N(d_2)$ are the cumulative probability distribution functions for a standardised normal random variable. d_1 and d_2 are calculated as below (Chen, 2005):

$$d_1 = \frac{\ln(S_{MT}/K_{MT}) + (r + \sigma^2/2)T}{\sigma\sqrt{T}} \quad (3)$$

$$d_2 = \frac{\ln(S_{MT}/K_{MT}) + (r - \sigma^2/2)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T}$$

where r is the discount rate of return, σ is the uncertainty rate.

Equations (1) (2) and (3) are the analytical 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.

3.2 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.

3.2.1 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 S_{Dev} , the value of the system depreciates based on the following exponential accelerated depreciation model, the value of the system is as below:

$$S_{No_MT}(T_{bef}, \alpha_{No_MT}) = S_{Dev} \times \alpha_{No_MT}^{T_{bef}} \quad 0 < \alpha_{No_MT} < 1 \quad (4)$$

where S_{No_MT} is the economic value of the system in a unit time without maintenance; α_{No_MT} is the depreciation rate without maintenance; and T_{bef} 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 S_{Add_val} , the depreciation rate of the system with adaptive/perfective maintenance is α_{AdaPec_MT} , and the value of the system with maintenance is as below:

$$\begin{aligned} S_{WithAdaPec_MT}(S_{Dev}, S_{Add_Val}, \alpha_{AdaPec_MT}, \alpha_{No_MT}, T_{bef}, T_{MT}) \\ = (S_{Dev} \times \alpha_{No_MT}^{T_{bef}} + S_{Add_Val}) \times \alpha_{AdaPec_MT}^{T_{MT}} \quad 0 < \alpha_{AdaPec_MT} < 1 \end{aligned} \quad (5)$$

where α_{AdaPec_MT} is the depreciation rate with adaptive/perfective maintenance, $S_{WithAdaPec_MT}$ is the unit value of the system with adaptive/perfective maintenance in a unit time; and T_{MT} is the system maintenance time in adaptive/perfective maintenance.

So the economic value of the adaptive/perfective maintenance is:

$$\begin{aligned} S_{MT}(S_{Dev}, S_{Add_Val}, \alpha_{AdaPec_MT}, \alpha_{No_MT}, T_{bef}, T_{MT}) \\ = S_{WithAdaPec_MT} - S_{No_MT} \\ = (S_{Dev} \times \alpha_{No_MT}^{T_{bef}} + S_{Add_Val}) \times \alpha_{AdaPec_MT}^{T_{MT}} - S_{Dev} \times \alpha_{No_MT}^{T_{bef} + T_{MT}} \end{aligned} \quad (6)$$

3.2.2 Maintenance input – the cost of the adaptive/perfective maintenance

Adaptive maintenance may cause to buy new devices and software. Perfective maintenance needs 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.

4 Influence analysis for adaptive and 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:

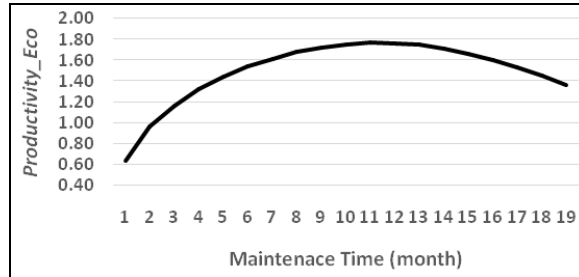
S_{MT}	economic value of adaptive/perfective maintenance in a unit time (unit: dollar)
K_{MT}	pre-committed fixed cost of adaptive/perfective maintenance (unit: dollar)
C_{MT}	variable cost of adaptive/perfective maintenance in a unit time (unit: dollar)
T_{MT}	IS system maintenance time (unit: month)
$Productivity_Eco$	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} ; modern programming practice F_{Mode_prac} ; user demands for enhancements F_{user_dema} and environmental change F_{Envi_chan} , and quality of system documentation F_{Qual_docu} .

4.1 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: $S_{Dev} = 10,000$, $S_{Add_Val} = 2,000$, $\alpha_{No_MT} = 0.8$, $\alpha_{AdaPec_MT} = 0.85$, $K_{MT} = 1,000$, $T_{bef} = 6$, $r = 0.03$, $\sigma = 0.25$. T_{MT} changes from 1 to 19 with the interval of 1. Figure 1 shows how $Productivity_Eco$ changes with T_{MT} .

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



The simulation results indicate that for adaptive/perfective maintenance without influence of factors, the economic productivity at first inclines then declines with time.

4.2 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 three 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:

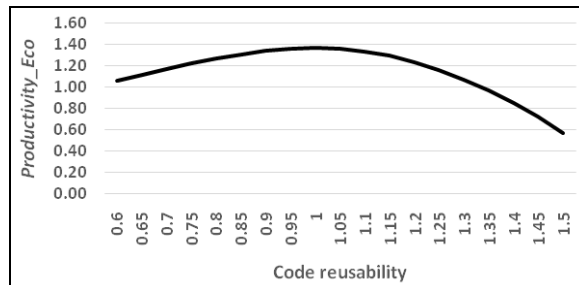
$$K_{MT} = \beta_{KC0} + \beta_{KC1} \times F_{Code_reus} + \varepsilon_{KC} \quad \beta_{KC1} < 0 \tag{7}$$

$$T_{MT} = \beta_{TC0} + \beta_{TC1} \times F_{Code_reus} + \varepsilon_{TC} \quad \beta_{TC1} < 0 \tag{8}$$

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: $S_{Dev} = 10,000$, $S_{Add_Val} = 2,000$, $\alpha_{No_MT} = 0.6$, $\alpha_{AdaPec_MT} = 0.8$, $T_{bef} = 6$, $r = 0.03$, $\sigma = 0.25$, and $\beta_{KC1} = -400$, $\beta_{KC0} = 1,400$, $\beta_{TC1} = -10$, and $\beta_{TC0} = 16$.

Figure 2 Productivity_Eco changes with code reusability F_{Code_reus}



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 Figure 2.

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 should not pursue 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.

4.3 Analysis of the influence factor – modern programming practice F_{Mode_prac}

Modern programming practice follows standard development and maintenance process framework, uses modularised 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:

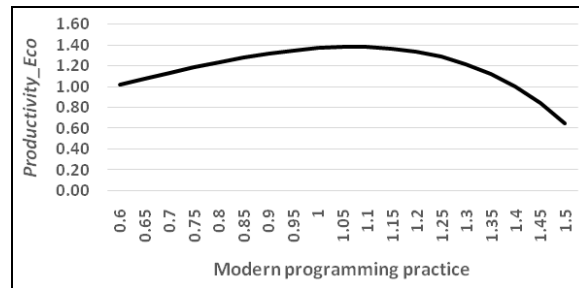
$$K_{MT} = \beta_{KM0} + \beta_{KM1} \times F_{Mode_prac} + \varepsilon_{KM} \quad \beta_{KM1} > 0 \quad (9)$$

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

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: $S_{Dev} = 10,000$, $S_{Add_Val} = 2,000$, $\alpha_{No_MT} = 0.6$, $\alpha_{AdaPec_MT} = 0.8$, $T_{bef} = 6$, $r = 0.03$, $\sigma = 0.25$, and $\beta_{KM1} = 400$, $\beta_{KM0} = 600$, $\beta_{TM1} = -10$, and $\beta_{TM0} = 16$. Figure 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 pursue extreme modern programming practice.

Figure 3 Productivity changes with modern programming practice



4.4 Analysis of the influence of user demands for enhancements F_{user_dema} and environmental change F_{Envi_chan}

User demands for enhancements and extentions are the main reason for perfective and adaptive maintenance. On the other hand, user demands always accompany environmental changes. When business environment changes, for example the organisation develops a new information channel, the IS software system should add a new module for that new channel. When software or hardware environment changes, for example with big data era, more information is stored in cloud disk, and users need to transfer part of data to cloud disk to save hardware cost. All these cause adaptive or perfective maintenance of the present system. Environmental change is always the source of user demands for enhancements and extents. So we can discuss these two factors together. We use user demands as the representative.

User demands affect maintenance time T_{MT} in the economic productivity model. Very frequent or brand new user demands need more maintenance time. Maintainers are busy in analysing with different user requirements, put great effort in transferring program and adding new functions. Frequent user demands also increase the pre-committed cost K_{MT} on purchasing software and hardware. Meanwhile, satisfying frequent demands means the new added value S_{Add_val} of the system increases quickly, and the depreciation rate α_{AdaPec_MT} becomes higher too. So user demands for enhancements and extents F_{user_dema} is positively correlated to S_{Add_val} , α_{AdaPec_MT} , K_{MT} and T_{MT} . We use the following regression model to analyse their correlations:

$$S_{Add_val} = \beta_{SU0} + \beta_{SU1} \times F_{User_dema} + \varepsilon_{SU} \quad \beta_{SU1} > 0 \tag{11}$$

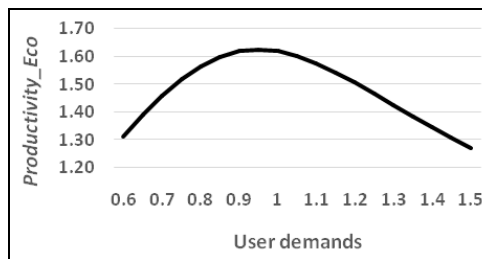
$$\alpha_{AdaPec_MT} = \beta_{\alpha U0} + \beta_{\alpha U1} \times F_{User_dema} + \varepsilon_{\alpha U} \quad \beta_{\alpha U1} > 0 \tag{12}$$

$$K_{MT} = \beta_{KU0} + \beta_{KU1} \times F_{User_dema} + \varepsilon_{KU} \quad \beta_{KU1} > 0 \tag{13}$$

$$T_{MT} = \beta_{TU0} + \beta_{TU1} \times F_{User_dema} + \varepsilon_{TU} \quad \beta_{TM1} > 0 \tag{14}$$

In simulation we set parameters and coefficients as follows: $S_{Dev} = 10,000$, $\alpha_{No_MT} = 0.6$, $r = 0.03$, $\sigma = 0.25$ and $\beta_{SU1} = 4,000$, $\beta_{SU0} = 0$, $\beta_{\alpha U1} = 0.2$, $\beta_{\alpha U0} = 0.6$, $\beta_{KU1} = 1,000$, $\beta_{KU0} = 0$, $\beta_{TU1} = 4$, $\beta_{TU0} = 8$. The value of user demands factor F_{user_dema} varies from 0.6 to 1.5 with the interval of 0.05. The bigger the value is, the bigger the user demands are. We observe the change of $Productivity_Eco$ with the change of F_{user_dema} . The results are illustrated in Figure 4.

Figure 4 Productivity changes with user demands for enhancements F_{user_dema}

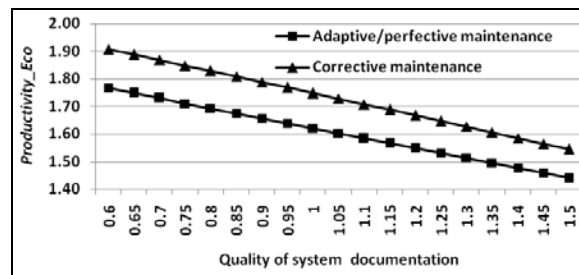


When user demand rises from small to middle level, the productivity goes up to the peak. When user demand continues to rise from middle to very big, the productivity bends down and declines. This indicates that to obtain optimal productivity, users would better not raise too many demands in short time. Arranging demands evenly will yield optimal economic productivity.

4.5 Influence of the factor – quality of system documentation F_{Qual_docu}

The quality of system documentation influences the economic productivity of adaptive/perfective maintenance in the same way for corrective maintenance. High documentation quality can reduce maintainers' effort on preparation work for maintenance. So it is negatively correlated with the pre-committed fixed cost of maintenance K_{MT} . We can compare the difference between the influences of F_{Qual_docu} on corrective and adaptive/perfective maintenance. We set the new added value $S_{Add_val} = 4,000$, system running time before adaptive/perfective maintenance $T_{bef} = 0$. The economic productivity results of the model with factor F_{Qual_docu} are given in Figure 5.

Figure 5 Productivity_Eco changes with quality of system documentation F_{Qual_docu}



From Figure 5 we can find the two lines of productivity are almost parallel. The distance value between the two productivity lines reflects the added value of adaptive/perfective maintenance. With value added, the productivity of adaptive/perfective maintenance is lower. S_{MT} of adaptive/perfective maintenance is larger than that of corrective maintenance, but C_{MT} of the adaptive/perfective maintenance is larger too and grows more quickly, consequently causing the productivity of adaptive/perfective maintenance to become lower.

5 Conclusions

In this paper, we present an IS 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, and quality of system documentation. 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 phases. Therefore this simulation approach provides a tool for IS project managers for tuning the parameters to obtain the optimal productivity in economic viewpoint. This could help formulating strategies to improve maintenance productivity.

References

- 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*, Vol. SE-9, No. 6, pp.639–648.
- Banker, R.D. and Slaughter, S.A. (1997) 'A field study of scale economies in software maintenance', *Management Science*, Vol. 43, No. 12, pp.1709–1725.
- Basili, V. et al. (1996) 'Understanding and predicting the process of software maintenance release', *Proceedings of the 18th International Conference on Software Engineering*, IEEE Computer Society, 25–29 March 1996, Berlin, Germany.
- Boehm, B.W. (1981) *Software Engineering Economics*, Prentice-Hall Advances in Computing Science and Technology Series, Prentice-Hall, Englewood Cliffs.
- Boehm, B.W., Horowitz, E., Madachy, R., Reifer, D., Clark, B.K., Steece, B., Brown, A.W., Chulani, S. and Abts, C. (2000) *Software Cost Estimation with COCOMOII*, Prentice Hall, Upper Saddle River, New Jersey, ISBN 0130266922.
- Boeing, C. (1979) *Software Cost Measuring and Reporting*, US Air Force-ASD Document D 180-22813-1.
- Chen, J. (2005) *The Physical Foundation of Economics, An Analytical Thermodynamic Theory*, World Scientific Publishing Co., Singapore.
- COSMIC (2009) *The COSMIC Functional Size Measurement Method Version 3.0.1 Measurement Manual*, Common Software Measurement International Consortium [online] <http://cosmic-sizing.org/publications/measurement-manual-3-0-1/> (accessed 3 August 2016).
- de Barros Sampaio, S.C. et al. (2010) 'A review of productivity factors and strategies on software development', *Fifth International Conference on Software Engineering Advances (ICSEA)*, IEEE.
- de Lucia, A., Pompella, E. and Stefanucci, S. (2003) 'Assessing the maintenance processes of a software organization: an empirical analysis of a large industrial project', *Journal of Systems and Software*, Vol. 65, No. 2, pp.87–103.
- Hertz, J., Krogh, A. and Palmer, R.G. (1991) *Introduction to the Theory of Neural Computation*, Addison-Wesley, Redwood City, CA.
- Huang, K., Kong, X. and Sandrasegaran, K. (2014) 'Analysis of the influence to productivity of software corrective maintenance using an economic model', *International Conference on Proceedings of Information Technology and Electronic Commerce (ICITEC)*, pp.117–121, IEEE.
- IEEE (1992) *IEEE Standard for Software Productivity Metrics 1045*, IEEE Standards Association, Piscataway, NJ, USA.
- IEEE (2010) *ISO/IEC/IEEE 24765:2010(E) Systems and Software Engineering – Vocabulary*, IEEE Standards Association, Piscataway, NJ, USA.
- ISO/IEC (2007) *ISO/IEC IS 14143-1: 2007 – Information Technology – Software Measurement – Functional Size Measurement – Part 1: Definition of Concepts*, February, International Organization on Standardization (ISO) and the International Electrotechnical Commission (IEC), New York.
- Jorgensen, M. (1995) 'Experience with the accuracy of software maintenance task effort prediction models', *IEEE Transactions on Software Engineering*, Vol. 21, No. 8, pp.674–81.
- Lientz, B.P. and Burton Swanson, E. (1981) 'Problems in application software maintenance', *Communications of the ACM*, Vol. 24, No. 11, pp.763–69.

- Liu, L., Kong, X. and 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*, Vol. 8, No. 1, pp.39–52.
- Moazeni, R., Link, D. and Boehm, B. (2014) 'COCOMO II parameters and IDPD: bilateral relevances', *Proceedings of the 2014 International Conference on Software and System Process*, pp.20–24, ACM.
- Nguyen, V., Boehm, B. and Danphitsanuphan, P. (2011) 'A controlled experiment in assessing and estimating software maintenance tasks', *Information and Software Technology*, Vol. 53, No. 6, pp.682–91.
- Rosa, W., Madachy, R., Boehm, B. and Clark, B. (2014) 'Simple empirical software effort estimation model', *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ACM, p.43.
- Swanson, E.B. (1976) 'The dimensions of maintenance', *Proceedings of the 2nd International Conference on Software Engineering*, 13 October, IEEE Computer Society Press, Los Alamitos, CA, USA, pp.492–497.