Web Based Application Maintenance Time Estimation Modeling using Bootstrap SEM

Somchai Prakancharoen
Department of Computer and Information Science Faculty of Applied Science
King Mongkut's University of Technology North Bangkok
spk@kmutnb.ac.th

***Abstract*— The objective of this research was to find out the components and their relationships which effect to Web based application maintenance time estimation. Maintenance time indicators were defined from literature review of software maintenance time estimation articles. The 17 maintenance time indicators were used to construct questionnaire. One hundred and forty of public and private sector completed IT software maintenance projects were collected. All of these time indicators which passed significance Factor analysis were then grouped in to four Factors (Application Attribute, Application Difficulty, Maintenance Team Attribute, Application Reliability and Modularity). These four Factors were used to find out their relationship with Structural Equation Modeling (SEM). The Estimation technique that suitable for SEM calculation was inspected by Bootstrap technique with 1,000 sample datasets. Maximum likelihood (ML) was the best technique. SEM under ML was used to construct the Factors and Dependence variable (Maintenance time) relationship of Software maintenance time model. The model was cross validation for accuracy, tested with 30 completed software maintenance projects. The result of cross validation was presented in Mean Magnitude of Relative Error (MMRE) about 42.55%.**

Keywords- Web based application software maintenance, Factor analysis, Structural equation modeling, Bootstrap

I. INTRODUCTION

Software Maintenance is effort consumption activity and may cause critical event if it cannot be delivered to user in a suitable time. If we can precisely estimate amount of software maintenance time then project planning of software maintenance could be easily defined.

The Objective of this research is to construct software maintenance time estimation model. This model was especially designed for the public and private sector of Thailand web based application during 2006-2008.

Methodology of research was conducted with many activities. First, Software maintenance metrics about maintenance time was gathered from many related research and even text books. The questionnaire was designed and test of reliability after that 140 completed software maintenance projects from both public and private software department were collected. Factor analysis by principle component analysis was used to deleted some non related indicators to maintenance time and group some related indicators to be factor or component. The technique "Structural equation modeling" was used to find out factors relationship. There are many techniques were use to analyze the model. Bootstrap technique was used to find out which was the best one. After that SEM equation use this technique to find the Factor relationship under Bootstrap approaches. The SEM equation was then cross validation with thirty completed software maintenance projects which resulted in Mean of Magnitude of Relative Error.

II. RELATED THEORY AND RESEARCH

A. Factor analysis [1]

Factor analysis is technique of reducing some unimportance indicators and grouping some related indicators to be new latent variable (factor or component). There are many method used to compose indicators to new component such as Principle component analysis: PCA, Maximum likelihood: MLE.

B. Structural Equation Modeling: SEM [2]

Structural equation modeling is a method of confirmatory factor analysis. It can analyze whether a user's purposed model of factor relation is good or not.

Estimation method of model calculation are Maximum Likelihood Estimation: ML, Generalized Least Squares: GLS, Asymptotically distribution free: ADF, Un weighted least square: ULS…etc. Each method gave difference model fitting.

Result model of SEM must be checked with Goodness of fit by some statistics such as Chi-square (χ^2) (ought to non significance), Goodness of Fit Index: GFI (exceed > 0.9), Adjusted Goodness of Fit Index: AGFI (exceed > 0.9), Root Mean Square Error of Approximation: RMSEA (lower than <= 0.06) and Hoelter's N (ought to exceed 75).

C. Software sizing

Donald J.Reifer [11] present that "Web Objects", source code sizing method, which has a similar concept to "Function Point". "Web object" defined the web based software sizing with more reasonable size estimation to web based application than Function point. The attribute of web object covered Internal logical files, External interface files, External inputs, External outputs, External inquiries, # multi-media files, # web building blocks, # scripts (animation, audio, video, visual, etc.) , # of links (xml, html and query language lines). This research choose web object in software sizing in this research.

D. Relevant research

Literature review from related papers [3, 4, 5, 6, 7, 8, 9, 10, 12, 13, 15] were studying then some commonly used indicators in software maintenance time estimation were collected.

E. Evaluation criterion

Cross validation of purposed model is an important activity to confirm of model reliability. The final best fitted model would be tested with completed software maintenance in actual time. Predicted time from the purposed model is then calculated and then analyzed MMRE value (Magnitude of Relative Error–MRE) [16] as equation (1).

$$MMRE = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{|ActualTime_i - PredictedTime_i|}{ActualTime_i}\right]\times 100 \quad (1)$$

F. Bootstrap [14]

Bootstrap is a technique which increase sample dataset from original dataset. These new sample datasets should have the same size with the original dataset. Each dataset was created by re sampling (in random) with replacement any case from the original dataset. Normally, five hundred of new datasets was the minimum amount that will be suitable for using to calculation of specific parameters. The method of how to estimate are the same as Structural equation model estimation (ML, GLS, SLS,ULS). Discrepancy value which produce by these methods should indicate the poor or better fit of the model distribution estimation. Normally, the small discrepancy value mean more fitted.

III. Research Framework

A. Independence variables collection.

Literature review from II.D were considered and summary that 17 indicators were mostly referenced in software maintenance time estimation as illustrated in table I.

TABLE I. MOST COMMON REFERENCED INDICATORS

Indicator	Description	Data Type
W_O	Web Object	0-∞
App_Req	Application Requirement	1-5
App_Reli	Application Reliability	1-5
App_Comp	Application Complexity	1-5
App_Modu	Application Modularity	1-5
MT-Cap	Maintenance Team Capability	1-5
MT_Exp	Maintenance Team Experience	1-5
MT_Coh	Maintenance Team Cohesive	1-5
MT_Stab	Maintenance Team Stability	1-5
MT_App-Exp	Maintenance Team Application Experience	1-5
App_Plt_Diff	Application Platform Difficulty	1-5
App_Lang_Diff	Application Language Difficulty	1-5
App_Aging	Application Aging	1-5
App_Lang_Old	Application Language Oldie	1-5
App_Rel_T_Org	Application Related to Organization	1-5
CMM_Lev	Capability Maturity Model	1-5
MT_T	Maintenance Team Tool	1-5

(level 1-5 is Likert rating scale : 1 =very low, 2 =low, 3 =nominal, 4 =high, 5 =very high)

B. Data collection

Indicators from III.A were designed to be a form fill-in questionnaire. This questionnaires were then submitted to target sample source (thirty five public and private sector software departments). One hundred and forty completed software maintenance projects were sent back. Data from questionnaire was prepared and cleaning before statistical processing. Some Indicator (Such as MT_Time, W_O) have more skew ness than criterion (not exceed +/- 1) thus Log_{10} transformation for both indicator (represent with L_MT_Time, L_W_O) would reduce their skew ness to become a normal distribution.

C. Factor Analysis

One hundred and ten completed software projects were (training case) then transformed to be standardized value for preventing from sizing domination of each indicator. PCA method was then used to factor extraction. KMO value was "0.775", Bartlett's test of Sphericity was non significant *(α=0.001)* and cumulative variance explained = "71.081 %" (appropriate large). This could conclude that factor analysis met good criterion. Variamax rotation method could depict more clarity factor and their indicators as presenting in table II.

TABLE II. FACTOR AND THEIR COMPOSED INDICATORS

Factor	Indicator
App_Attribute	zApp-Req, zApp_Aging, zApp_Lang_Old, zApp_Rel_T_Org, zCMM_Lev, zL_W_O
App_Diffculty	zApp_Comp, zApp_Plt_Diff, zApp_Lang-diff
App_Reliability_Modularity	zApp_Reli, zApp_Modu
MT_Attribute	zMT_Cap, zMT_Exp, zMT_App_Exp, zMT_Coh, zMT_Stab, zMT_T

D. Structural equation modeling (SEM)

- Four extracted factors and their indicator were constructed to be a purposed model subject to concept of researcher as represented in figure 1. Latent variable maintenance time (MT_Time) was a target latent variable which composed of one manifest variable zL_MT_Time. MT_Time was endogenous latent variable while another factor were exogenous latent variables (dominator factor).

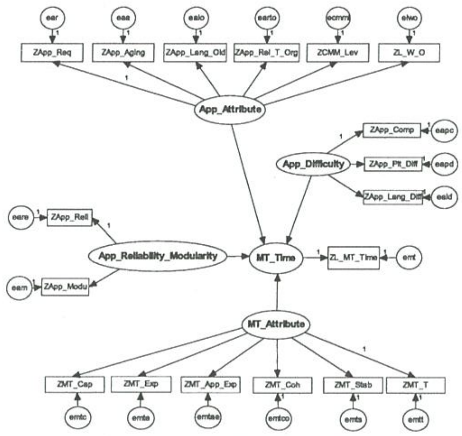

Figure 1. Researcher's purposed model

- One thousand bootstrap datasets were generated and used to search for the best estimation methods. The discrepancy between the population moments and the sample moments were presented in the table III. The result of bootstrap calculation show that ML(small value of discrepancy-χ^2) was the best estimation methods in this research domain.

TABLE III. DISCREPANCY VALUE (C)

		Population discrepancy		
		C-ML	C-SLS	C_ULS
Sample discrepancy	C-ML	478.246(0.965)	765.988(1.664)	752.491(1.634)
	C-SLS	499.963(0.922)	724.652(1.415)	711.883(1.390)
	C_ULS	499.142(0.858)	725.059(1.409)	712.283(1.385)

- SEM with ML estimation and Bootstrap calculation (ML-1,000 datasets) for all remaining significance factors and Indicators were then considered modify their relationship by adding, cutting. Goodness of fit indices were immediately observed whether their statistics were closely to the best criterion. After many trials were tried out, the met criterion model (goodness of fit) was illustrated in figure 2. The passed statistics criterion was shown in table IV.

TABLE IV. IMPORTANT GOODNESS OF FIT STATISTICS

Statistics	Value
Chi_square (χ^2)	23.337 (p=0.055)
GFI, AGFI	0.956, 0.887
RMSEA	0.077
HOELTER 's N	140(0.01)

Chi_square (χ^2), GFI, AGFI, RMSEA and Hoelter's N were all passed lower criterion of best fit model.

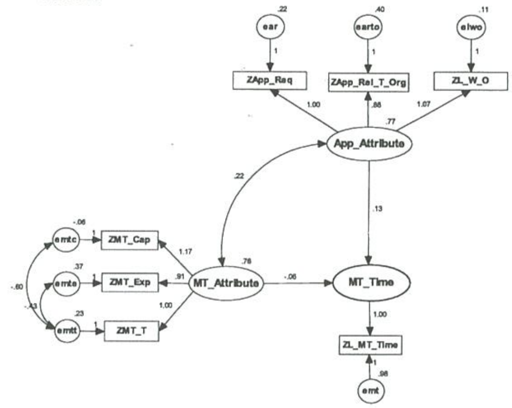

Figure 2. Final Model-best fitted

Final model in figure 2 illustrated significant factor and their indicators with MT_Time. Their relation ship quantity and direction were shown as well. MT_Time could be calculated from equation (unstandardized) (2)

$$MT\_Time = 0.13 * App\_Attribute - 0.06 * MT\_Attribute \quad (2)$$

$$ZL\_MT\_Time = MT\_Time + 0.98 * emt \quad (3)$$

$$ZMT\_Time = \text{anti Log } 10\ (ZL\_MT\_Time) \quad (4)$$

$$Time = ZMT\_Time * \sigma_{time} + \overline{time} \quad (5)$$

while "emt" estimation value was equal to "0.979", σ_{time} = "6.039" and $\overline{time}$ = "5.584"

- Cross validation.

  Software maintenance time predicted value from equation (2) were then compared to thirty known software maintenance time of completed software maintenance project. MMRE of SEM equation (2) as presented in table V.

TABLE V. MMRE VALUE OF MT_Time

Item	Equation (2)
MMRE	42.55%
Accuracy	57.45%

IV. CONCLUSION AND SUGGESTION

A. Conclusion

- App_Attribute factor was positively influencing with coefficient value = "+0.13" to Maintenance time. Thus, if App_Attribute increase in one unit the value of MT_Time would be also increased in 0.13 unit. App_Attribute factor composed of Log10_Z_Web_object,Application_Requirement, and App_Rel_T_Org indicators. MT_Attribute factor was negatively influencing to MT_Time with coefficient value = "-0.06". MT_Attribute was composed of Maintenance Team Capability, Maintenance Team Experience and Maintenance Team Tool. The more value of MT_Attribute would more decrease value of MT_Time in "0.06" unit. It was very reasonable fact that high capability, experience and having maintenance tools were indicators which decrease software maintenance time

- Relation between factor MT_Attributre and App_Attribute was bi-directional with coefficient value = "+0.22".
- Maintenance time estimation in equation (2) could be used to predict software maintenance time with accuracy percentage 57.45%.

B. Suggestion for further research

This model illustrated the interaction between two exogeneous latent variables (MT_Attributre and App_Attribute) which have bi-direction relationship or Non linear SEM event. It ought to solve and delete this kind of relation with some technique such as Product indicator approaches or Distribution-analytic approaches[18].

References

[1] R.J. Rummel. Factor analysis . Hawaii University : USA. 2002.
[2] Garson David. Structural equation modeling. North-Carolina state university: USA. 2007.
[3] Crosby Philip. Practical software engineering. University of Calgary:USA, 2005.
[4] Mukhija Arun. Estimating software maintenance. Institute of informatik. Zurich University: SWISS, 2008.
[5] Banker D. Rajiv et El. Factor affecting software maintenance. Carnegie mellon university: USA. 1988.
[6] Stark E. George. Measurement to manage software maintenance. Mitre corporation. Colorado: USA.1997.
[7] Mukhija Arun and Glinz M.. Estimating software maintenance. Requirement research group. Zurich university: SWISS. 2003.
[8] Mustafa K.. Quality metric development framework. Al husain bin talal university: JORDAN. 2005.
[9] Cristine W. Thackaberry. Estimating metrics for course ware maintenance effort. Washington state university: USA. 1998.
[10] Krishnan S. Mayuram. The role of team factors in software cost and quality. University of Michigan : USA. 1998.
[11] Riefer J. Donald. Web object counting convention. Reifer institute. CA: USA. 2006.
[12] David F. Rico. Software process improvement. J.Ross Publishing: USA. 2007.
[13] DACS. Software Maintenance Metrics. : USA. 2006.
[14] Trevor Hastie. Generalized additive models.Stanford University, USA,1995.
[15] Pankatt b. Hatt, Inflencing Factors In Outsource software maintenance, ACM SIGSOFT,2006.
[16] Martin Shepperd, Software project effort using analogies, IEEE Vol 23,1997.
[17] Chernick, Michael R. (1999). *Bootstrap Methods, A practitioner's guide*. Wiley Series in Probability and Statistics. ISBN 0471349127.
[18] Schermelleh Karin. Nonlinear SEM is Partial least squares and alternatives. Goethe university. Germany. 2009.