Improving Corrective Maintenance Effort Prediction: An Empirical Study

Andrea De Lucia*, Aldo Persico°, Eugenio Pompella°, Silvio Stefanucci*
delucia@unisannio.it, aldo.persico@eds.com, eugenio.pompella@eds.com, stefanucci@unisannio.it

* Faculty of Engineering - University of Sannio
Palazzo Bosco Lucarelli, Piazza Roma - 82100 Benevento, Italy

° EDS Italia Software S.p.A.
Viale Edison - Loc. Lo Uttaro - 81100 Caserta, Italy

Abstract

This paper reports on an empirical study aiming at improving the cost prediction model currently used in a major software enterprise. We used a multiple regression model and the data collected from two corrective maintenance projects. The improvement of the model performances is achieved by taking into account different corrective maintenance task typologies, each affecting the effort in a different way.

1 Introduction

Planning software maintenance work is a key factor for a successful maintenance project. Planning involves estimating of size, effort, duration, staffing and costs in order to guarantee the control of the maintenance process and reduce the risks and the inefficiencies related with the maintenance work. Indeed, early estimates and accurate evaluation permit to significantly reduce project risks and can be useful for predicting maintenance costs, comparing productivity and costs among different projects, learning the process performance and parameters, and so on.
Software project costs are essentially human resource costs and this entails that the effort (man-days needed for system maintenance) should be maintained under severe control.
In the area of software maintenance there are few studies on the accuracy of effort prediction models [Jor95, CTA98, JSK00].
This paper presents an empirical study about improving the estimation and prediction of corrective maintenance effort in a large software organization.
To obtain a good estimation model the following are necessary:

- an understanding of the process to infer and identify the characteristics that are important for the estimation;
- the availability of a set of measures for the identified characteristics, that permits the construction of a variety of models and the selection of the more suitable.

The subject organization has a nontrivial experience on estimation processes and model definition and classification based on intervention type. Moreover, the results of the adopted estimation model must be considered flexible, because often they are discussed and evaluated together with the customer.
The most confident estimation model adopted by the organization is deeply based on the system size. This measure, if necessary, is translated in equivalent kLOC COBOL from FP and from other different programming languages using the conversion factors indicated by Caper Jones [Jon99]. Finally the prediction model is expressed by a linear or a quadratic equation, depending on a breakpoint value of the unique independent variable, which is the number of annual corrective maintenance tasks normalized on the software code size (in kLOC).
This model has been improved by considering the impact of different maintenance task types. Section 2 presents the data set used in our empirical study; section 3 discusses the empirical results and the improved cost estimation model. Concluding remarks are given in section 4.

2 Case study

To improve the cost estimation model adopted within the organization we have analyzed the data of two corrective maintenance projects.
Project 1 was conducted on a set of four telecommunication systems, based on a traditional platform, which manage the telephonic network information and the registration, configuration, and running of data and resources related to the new installed plants. In particular, the management of running data permits the activation of specific supplementary telephonic services.
Project 2 was conducted on two subsystems of an accounting system implementing a particular financial procedure which is in charge of funds assignment and

their distribution to the beneficiating institutions. It manages all the aspects involving with the accounting, such as introit management, expenditure management, annual financial closure, and so on.
The data set was composed of 41 observations, 28 corresponding to quarterly maintenance periods for the four systems of Project 1 and 13 corresponding to monthly maintenance periods for the two systems of Project 2. For each observation, the following data were available (see Table 1):

- size of the system to be maintained;
- effort spent in the maintainance period;
- number of maintainance tasks, distinguished in three categories:
  - type A: the maintenance task requires software source code modification;
  - type B: the maintenance task requires fixing of data misalignments through database queries;
  - type C: the maintenance task requires intervention not comprised in the previous categories, such user disoperation, problems out of contract, and so on.

The final set is composed of 41 observations, collected from the two projects, each with the metrics shown in Table 1 and Table 2.

Metric	Description.
NA	# of tasks requiring software modification
NB	# of tasks requiring fixing of data misalignment
NC	# of other tasks
SIZE	Size of the system to be maintained [kLOC]
EFFORT	Actual Effort [man-days]

Table 1: Collected metrics

Metric	Min	Max	Mean	Std.Dev.
NA	0	154	37,2927	34,9923
NB	0	1096	232,439	295,939
NC	21	980	206,268	214,398
SIZE	179,23	5277	2063,084	1682,884
EFFORT	55	750,4	297,931	175,926

Table 2: Descriptive statistics of the data set

The application of the model adopted within the organization to the data set produces good results, though non excellent for all the subsystems (see Table 3 and Table 4). The measure $PRED_{XX}$ [BG83, Jor95] represents the percentage of observation with relative error at most equals to XX.
It must not be surprising that a single metric model could have this good performances. The model is the result of many years of experience and has worked well when applied. Moreover, the used metric was choosen because it presents a strong linear correlation with the corrective maintainance effort. However, the model does not take into account the different types of maintenance tasks.

	Relative Error [%]	
Subsystem	**Ave.**	**Max**
1-1	13,44	31,04
1-2	19,67	32,26
1-3	126,17	144,39
1-4	33,13	50,87
2-1	24,57	66,66
2-2	65,81	76,20

Table 3: Organization model relative error

$PRED_{25}$	$PRED_{50}$
43,9	58,5

Table 4: Organization model prediction performances

3 Improving the effort prediction model

Our observation was that the effort required to perform a maintenance task of type A might be sensibly different than the effort required to perform task of type B or C. For this reason we decided to use the number of tasks of the different types to improve the model and take into account the difficulty and the effort needed for the different maintenance task types.
In particular we used a multivariate linear regression model of the type:

$$Effort = b_0 + b_1\, NA + b_2\, NB + b_3\, NC + b_4\, Size$$

where NA, NB, NC, and Size are defined as in the previous section. The presence of the size of the system being maintained is consistent with the findings of [NV97]. Certainly, it would be very interesting to also consider the size of the maintenance tasks, but this metric is too fine grained and was not available.
Indeed, the main problem encountered was the poorness of the data set. Indeed, although we had 41 points (maintenance periods) in the data set, they only referred to two maintenance projects and to 6 different systems. This also means that the size of the different versions of the same subsystem only slightly differs over the different maintenance periods.
Another consideration is that the maintenance projects analyzed presented a very different distribution of the maintenance task types. In fact, project 2 has practically a null percentage of tasks of type B and has a percentage of tasks of type A (average 35%) clearly greater than project 1, where tasks of types B and C account up to 80% of the total number of tasks.
These characteristics can affect the analysis results because the data set is not well distributed, but this problem can only be overcome by increasing the size of the sample data set.

Another interesting point is whether an intercept term b_0 should be included in the model. Such a term would suggest the existence of an effort type not directly related to the variables being included in the model and/or a constant deviation term to balance the model error. However, from the *p-value* for the significance test of the regression coefficient b_0 we have statistical evidence to state that an intercept value is not needed. Thus, the intercept term b_0 was not considered in the model.

The correlation matrix (Table 5) shows the absence of strong correlations between the independent variables. The highest correlation value (0,6959) is between the number of tasks of type B and C, while the values of the correlations with A indicate independence among the variables. A potential reason for this result is that the types B and C are more recurrent than type A and generally have similar resolution effort because they do not require modifications to the software code.

Also the correlations with the dependent variable (Effort) are congruent. The strong correlation with the Size (0,7394) can be explained by the fact that, generally, the growth of the system size is generally accompanied by a growth of the effort spent for its maintenance [LB85].

A point of interest is the value for the correlation between the effort and the number of maintenance tasks of type A (0,4468). We expected a higher value, because this is the task type requiring more effort, as it involves changes to software code. A potential reason for this can be the limited size of the sample data set. Table 6 shows the model parameters. RMSE is the root of the mean square error. It is an estimate of the standard deviation of the residuals; AdjR^2 is a version of R^2 which seeks to remove the distortion due to a small sample size.

To assess the prediction error on future observations, a leave-one-out cross-validation of the model [Mey86] was performed and the PRESS statistics [SO91] were calculated. The results are shown in Table 7. PRESS (PREdiction Sum of Square) is the sum of squared prediction errors:

$$PRESS = \sum_i (\hat{y}_{i^*} - y_i)^2$$

SPR is the sum of the absolute value of prediction errors:

$$SPR = \sum_i \left| \hat{y}_{i^*} - y_i \right|$$

Finally, $R^2_{predict}$ is an R^2-like statistic reflecting the model prediction ability:

$$R^2_{predict} = 1 - \frac{PRESS}{\sum_i (y_i - \bar{y}_i)^2}$$

It is evident that the values for the PRED measures are very promising: the model predicts about the 63% of the cases within a relative error less than 25% ($PRED_{25}$) and about 88% of the cases with a relative error less than 50% ($PRED_{50}$). This demonstrates an improvement of the model with respect to the figures shown in tables 3 and 4. Also, the relative mean error is 22,75% and can be considered very good. Indeed, as outlined in [VMP91], an average error of 100% can be considered "good" and an average error of 32% "outstanding".

Due to the excellent values for the PRED measures we assessed more efficiently the predictive capability of the model by performing a leave-more-out cross-validation. We decided to discard from the learning data set a prediction set composed of all the observations related to a single subsystem. In this way, we can be more confident about the model performances, because in this way the model has no knowledge about the characteristics of the subsystem used as prediction set. The results are shown in Table 8.

	NA	NB	NC	SIZE	EFFORT
NA	1,0000	0,2291	0,0086	0,1217	0,4468
NB	0,2291	1,0000	0,6959	0,2857	0,4791
NC	0,0086	0,6959	1,0000	0,2653	0,5933
SIZE	0,1217	0,2857	0,2652	1,0000	0,7394
EFFORT	0,4468	0,4791	0,5933	0,7394	1,0000

Table 5: Metrics correlation matrix

Var.	b_i (Coeff.)	p-value	R^2	Adj R^2	RMSE
NA	2,311812	>10E-07	0,9683	0,9658	64,60
NB	0,1319257	0,013334			
NC	0,2607937	0,000430			
SIZE	5,944772E-02	>10E-07			

Table 6: Model parameters

$R^2_{predict}$	PRESS	SPR	Ave. Rel. Error	$PRED_{25}$	$PRED_{50}$
0,8328	206942,6	2127,42	22,75	63,41	87,80

Table 7: Model predictive performances

	Subsystems					
	1-1	**1-2**	**1-3**	**1-4**	**2-1**	**2-2**
Average Rel. Error	16,63	22,76	30,11	21,85	46,41	38,11
Max Rel. Error	49,79	61,37	85,01	35,45	112,27	56,56
$PRED_{25}$	71,43	71,43	57,14	57,14	14,29	33,33
$PRED_{50}$	100,00	85,71	85,71	100,00	71,43	50,00

Table 8: Subsystem prediction relative error

We can easily note that the relative error for the maintenance project 1 is still relatively low, while it is higher for the maintenance project 2: probably, this is due to the fact that the data set includes a higher number of observations from the maintenance project 1. This is also confirmed by the fact that the prediction relative error on

the observations belonging to the maintenance project 2 decreases if the prediction set is composed of observations of both projects (randomly selected), rather than of all the observations of a single system.

4 Conclusion

In this paper we have presented an empirical study from the experience of a major software enterprise in improving corrective maintenance effort prediction.

A data set obtained from two different corrective maintenance projects was used as case study to experimentally validate and compare model performances through linear regression models.

This data set was not very good, because of the limited metric set and the strong difference between the projects, evidenced by the absence of maintenance tasks of type B in project 2.

It would have been interesting to also consider the effort of the single maintenance tasks or, at least, the effort of all the tasks of the same type. In this way, we could have been more confident in our hypothesis of considering different maintenance task types in the costs estimation model. Nevertheless, the resulting costs estimation model shows a good predictability.

Various combination of the available metrics were also explored to look for better estimation models; in particular, considering the lack of tasks of type B in the second project and the correlation between tasks of type B and C, we evaluated a model grouping NB and NC. The results were slightly worst than the model distinguishing between the two maintenance task types, and then acceptable.

The final model is easy to use and understand, because the number of maintenance tasks and their typology in a specific time range are reasonably predictable, where a reasonable maintenance experience on the system to be maintained is available.

Acknowledgements

We would like to thank Prof. Aniello Cimitile for the stimulating discussions and suggestions.

The work described in this paper is supported by the project "Virtual Software Factory", funded by Ministero dell'Università e della Ricerca Scientifica e Tecnologica (MURST) and jointly carried out by EDS Italia Software, University of Sannio, University of Naples "Federico II", and University of Bari.

References

[Boe81] B. W. Boehm, *Software Engineering Economics*, Prentice-Hall Inc., Englewood Cliffs, N.J., 1981.

[BG83] E. Bradley and G. Gong, "A Leisurely Look at the Bootstrap, the Jack-Knife and Cross-Validation", *Amer. Statistician*, vol. 37, no. 1, 1983, pp. 836-48.

[CTA98] F. Calzolari, P. Tonella and G. Antoniol, "Dynamic Model for Maintenance and Testing Effort", *Proceedings of International Conference on Software Maintenance*, Bethesda, MA, USA, 1998, pp. 104-112.

[Jon99] C. Jones, "Mass-Updates and Software Project Management", http://www.spr.com/Resources/Books_Articles/Mass_Updates/mass_updates.htm, 1999.

[Jor95] M. Jorgensen, "Experience With the Accuracy of Software Maintenance Task Effort Prediction Models", *IEEE Transactions on Software Engineering*, vol. 21, n. 8, 1995, pp. 674-681.

[JSK00] M. Jorgensen, D. Sjoberg, and G. Kirkeboen, "The Prediction Ability of Experienced Software Maintainers", *Proceedings of 4th European Conference on Software Maintenance and Reengineering*, Zurich, SW, 2000, pp. 93-99.

[LB85] M. Lehman and L. Belady, *Program Evolution: Processes of Software Change*, Academic Press, Austin, 1985.

[Mey86] R.H. Meyers, *Classical and Modern Regression with Applications*, Duxbury Press, Boston, 1986.

[NV97] F. Niessink and H. van Vliet, "Predicting Maintenance Effort with Function Point", *Proceedings of International Conference on Software Maintenance*, Bari, Italy, 1997, IEEE Press, pp. 32-39.

[NV98] F. Niessink and H. van Vliet, "Two Case Study in Measuring Maintenance Effort", *Proceedings of International Conference on Software Maintenance*, Bethesda, Maryland, USA, 1998, IEEE Press, pp. 76-85.

[SO91] A. Stuart and J. K. Ord, *Kendall's Advanced Theory of Statistics*, 5th Edition, vol. 2, London, Edward Arnold, 1991.

[VMP91] S. Vicinanza, T. Mukhopadhyay, and M. Prietula, "Software Effort Estimation: an Exploration Study of Export Performance", *Information System Research*, vol. 2, no. 4, 1991, pp. 243-262.