2014

Simple Empirical Software Effort Estimation Model

Rosa, Wilson; Madachy, Ray; Boehm, Barry; Clark, Brad

ESEM’14, September 18–19, 2014, Torino, Italy.
http://hdl.handle.net/10945/46165

Simple Empirical Software Effort Estimation Model

Wilson Rosa
IT Estimation Division
Naval Center for Cost Analysis
Wilson.rosa@navy.mil

Ray Madachy
Department of Systems Engineering
Naval Postgraduate School
rjmadach@nps.edu

Barry Boehm and Brad Clark
Center for Systems and Software
University of Southern California
boehm@usc.edu
brad@software-metrics.com

ABSTRACT

Context: An effort estimation model with more than 20 parameters is not very useful at early conceptual phase if you don't have a logical approach for specifying the input values.

Goal: This paper presents a simple approach for predicting software development effort.

Method: The regression model uses product size and application types to predict effort. Product size is measured in terms of the equivalent source lines of code. The analysis is based on empirical data collected from 317 very recent projects implemented within the United States Department of Defense over the course of nine years beginning in 2004.

Result: Statistical results showed that source lines of code and application type are significant contributors to development effort.

Conclusion: The equation is simpler and more viable to use for early estimates than traditional parametric cost models. The effect of product size on software effort shall be interpreted along with application domain.

Categories and Subject Descriptors

D.2.9 [**Software Engineering**]: Management – *cost estimation.*

General Terms

Management, Measurement, Design, Economics

Keywords

COCOMO, software cost estimation, application domain, SEER-SEM, operating environment, application type

1. INTRODUCTION

Defensible estimates are mostly needed at the early conceptual phase of a software-intensive system's definition and development. An estimation model with 20-30 input parameters is not very helpful if you don't have a defensible approach for specifying the input values for such key parameters as the software's complexity, database size, platform volatility, required schedule compression, tool coverage, or your proposed project personnel experience. The purpose of this study is to provide a simple software effort model for early estimates. It examines the direct effect of product size and application type on cost. The analysis framework and variables used in this study builds on casual relationships established in past studies [3, 4, 5, 7, 8, 9, 12, 13, 15].

ESEM'14, September 18–19, 2014, Torino, Italy.

http://dx.doi.org/10.1145/2652524.2652558

2. RELATED WORK

Mainstream parametric cost models [2, 6, 10, 11] use source lines of code (SLOC) as measurement for predicting software effort. The main reason to using SLOC is that it allows practitioners to determine or at least estimate what parts of the system they will actually be developing. In contrast, function-point or use-case-based estimates are made without a way of determining which functions are going to be provided by Commercial-Off-The-Shelf products, cloud services, or other non-developed items, causing serious overestimates. While controversy exists over whether or not SLOC is a good measure, consistent use of this metric provides for meaningful statistical results [7,8,9,12].

3. RESEARCH METHOD

3.1 Population and Sample

The sample was identified as 317 software projects involving 12 application types, 7 operating environments, and 74 different software developers. These projects were completed during the time period from 2004 to 2012. The breakout according to application type (vertical axis) and operating environment (horizontal axis) are shown in Table 1.

Table 1 Software Dataset

	GSF	GV	SVU	MVM	AVM	AVU	OM	
TUL	1	0	0	0	5	2	0	8
PLN	20	0	0	0	0	0	0	20
IIS	11	2	0	0	1	0	0	14
SCI	10	1	0	1	6	0	1	19
SYS	13	3	0	3	6	0	0	25
TEL	22	2	0	22	1	0	0	47
TST	6	0	0	4	1	0	0	11
RTE	21	3	0	5	20	3	5	57
MP	16	0	0	3	9	1	5	34
VC	0	14	0	0	9	1	3	27
VP	0	0	1	1	9	2	5	18
SCP	14	1	1	3	3	9	6	37
	134	26	2	42	70	18	25	**317**

GSF = ground fixed; GV = ground vehicle; SVU = space vehicle unmanned; MVM = maritime vehicle manned; AVM = aerial vehicle manned; AVU = aerial vehicle unmanned; OM = ordnance and missile; TUL = tool; PLN = mission planning; IIS = intelligence and information systems; SCI = scientific; SYS = system; TEL = telecommunications; TST = test software; RTE = real-time embedded; MP = mission processing; VC = vehicle control; VP = vehicle payload; SCP = sensor control and signal processing

3.2 Instrumentation

Data were collected by means of a questionnaire containing over 20 items. The data collection questionnaire used in the study was obtained from an existing one; *Software Resource Data Report (SRDR)* questionnaire [14]. The source questionnaire entitled "SRDR Sample Formats" can be downloaded from the Defense Cost Analysis Resource Center (DCARC) website:

http://dcarc.cape.osd.mil/Files/Policy/2011-SRDRFinal.pdf
http://dcarc.cape.osd.mil/Files/Policy/Final_Developer_Report.xlsx

The questionnaire collected data on product size, effort, schedule and product attributes like required reliability, software process maturity, etc.

3.3 Data Normalization

The objective of data normalization is to improve data consistency, so that comparisons and projections are more valid. The dataset in this study was normalized using three steps:

3.3.1 Converting to Equivalent Size

Since the dataset captures project size by type (new, modified, reused, auto generated), it was possible to adjust the raw size to be its equivalent in new code using the COCOMO Reuse Model [1, 2]. The adjustment is based on the additional effort it takes to modify the code for inclusion in the product taking into account the amount of design, code and testing that was changed. Once adjusted it is called Equivalent Source Lines of Code (ESLOC).

3.3.2 Converting to Logical Count

Several projects were reported in either Physical or Non-Commented Source Statements. Those projects were converted into Logical SLOC using empirical factors from recent studies [7,12].

3.3.3 Data Grouping

To reduce variation and ensure valid comparisons, the 34 SEER-SEM application domains [6] were stratified into 12 general complexity zones called Application Types [7,12]. The application domains to application types mapping are shown in Table 2 below.

Table 2Application Type Taxonomy

Application Type	Symbol	SEER-SEM Application Domain(s)
Test	TST	Diagnostics Testing Software
Software Tools	TUL	Business Analysis Tool, CAD, Software Development Tools
Intelligence & Information Systems	IIS	Database, Data Mining, Data Warehousing, Financial Transactions, GUI, MIS, Multimedia, Relational/Object-Oriented Database, Transaction Processing, Internet Server Applet, *Report Generation,* Office Automation
Mission Planning	PLN	Mission Planning & Analysis
Mission Processing	MP	Command/Control
Real Time Embedded	RTE	Embedded Electronics/Appliance, GUI (cockpit displays), Robotics
Scientific Systems	SCI	Expert System, Math & Complex Algorithms, Simulation, Graphics
Sensor Control and Signal Processing	SCP	Radar, Signal Processing
System Software	SYS	Device Driver, System & Device Utilities, Operating System
Telecommunica tions	TEL	Communications, Message Switching
Vehicle Control	VC	Flight Systems (Controls), Executive
Vehicle Payload	VP	Flight Systems (Payload)

3.4 Effort Model and Variable Selection

The regression equation is based on the COCOMO Post-Architecture model [1, 2] without the effort multipliers. Dummy variables were added [16] to account for the impact of application types. The variables are described in Table 3.

Table 3 Variables in the Study

Variable	Type	Definition
Person-Month	Dependent	Software engineering effort (in Person-Month)
Thousand Equivalent Source Lines of Code (KESLOC)	Independent	The COCOMO method is used to make new and adapted (modified, reuse, generated) code equivalent so they can be rolled up into an aggregate size estimate.
Application Type	Dummy	The treatment of 12 (r) application types required the addition of 11 *(r-1)* dummy variables (D) as denoted below: D1 = 1 if PLN, 0 if TUL or

Variable	Type	Definition
		otherwise D2 = 1 if IIS, 0 otherwise D3 = 1 if SCI, 0 otherwise D4 = 1 if SYS, 0 otherwise D5 = 1 if TEL, 0 otherwise D6= 1 if TST, 0 otherwise D7= 1 if RTE, 0 otherwise D8= 1 if MP, 0 otherwise D9= 1 if VC, 0 otherwise D10= 1 if VP, 0 otherwise D11= 1 if SCP, 0 otherwise

3.5 Model Validity Measures

The measures for validating the model are described in Table 4.

Table 4 Model Validity Measures

Measure	Symbol	Description
Coefficient of Determination	R^2	The Coefficient of Determination shows how much variation in dependent variable is explained by the regression equation.
Coefficient of Variation	CV	Percentage expression of the standard error compared to the mean of dependent variable. A relative measure allowing direct comparison among models.
Variance Inflation Factor	VIF	Indicates whether multicollinearity (correlation among predictors) was present in a multi-regression analysis. A VIF lower than 10, indicates no multicollinearity.
Measure of Magnitude	F-test	The value of the F test is the square of the equivalent t test; the bigger it is, the smaller the probability that the difference could occur by chance.
P-value	α	Level of statistical significance established through the coefficient alpha ($p \leq \alpha$).
Model Validation		A cross-validation technique called "Jackknife" was used to test the accuracy of the predictor models. The idea is to hold out one observation at a time, use the remaining n-1 observations to estimate the model parameters. Use these parameters to calculate the predicted value for the hold out observation. If the resultant R2 value is significantly lower, then there is a change of overfitting in the model.

4. EFFORT MODEL RESULT

The regression equation was developed using the **CO$TAT** statistical analysis package of the ACEIT [16] application suite.

The resulting model, Eq. (1), predicts software development effort (in person months) as a function of product size and application type. Application type is the dummy variable. The treatment of the 12 application types (r) required the inclusion of 11 (*r-1*) dummy variables (D1…D11).

Eq. (1) is applicable for project size ranging between 1 and 842 KESLOC, 12 different application types (Table 2), and different business sectors (military, government and commercial).

$$PM = \mathbf{(2.047 \times KESLOC^{0.9288})} \times (2.209^{D1}) \times (1.917^{D2}) \times (3.068^{D3}) \times (3.072^{D4}) \times (3.434^{D5}) \times (4.521^{D6}) \times (4.801^{D7}) \times (4.935^{D8}) \times (5.903^{D9}) \times (7.434^{D10}) \times (10.72^{D11}) \qquad \textit{Eq. (1)}$$

Where:

PM	=	Engineering Labor in Person Months
KESLOC	=	Product size in thousand Equivalent Source Lines of Code
D1	=	1 if PLN, 0 if TUL or other application
D2	=	1 if IIS, 0 if other application type
D3	=	1 if SCI, 0 if other application type
D4	=	1 if SYS, 0 if other application type
D5	=	1 if TEL, 0 if other application type
D6	=	1 if TST, 0 if other application type
D7	=	1 if RTE, 0 if other application type
D8	=	1 if MP, 0 if other application type
D9	=	1 if VC, 0 if other application type
D10	=	1 if VP, 0 if other application type
D11	=	1 if SCP, 0 if other application type

Table 5 displays the accuracy of the model. The result shows that the effect of KESLOC (*P* value is 0.0000) on effort is "highly" significant, when treated along with application type. The R^2 value shows that 89% percent of the variation of software development effort has been explained by the regression.

The model's accuracy was also examined using a cross-validation model with an R^2 value of 87%. This number is slightly lower (even after adjusting for degrees of freedom) than the R^2 of 89%. Thus, the slight deterioration in fit from 89 to 87% does not pose a threat to external validity.

Table 5 Model Accuracy

Measure	Symbol	Result
Coefficient of Determination	R^2	89%
Coefficient of Determination (Cross-Validation Model)	R^2	87%
Coefficient of Variation	CV	33%
Measure of Magnitude	F-test	214
P-Value	α	0.0000
Sample Size	N	317
Mean (Effort)	Mean	431

Table 6 shows the model's statistical significance. The result indicates no multicollinearity present in the regression model as the VIF for all variables is lower than 10. Thus, there is no need to remove predictors from the model. The model is valid as all p-values are below .05.

Table 6 Model Validity

Variable	VIF	T-Stat	P-value
Intercept	NA	3.9	0.0001
KESLOC	1.2	47.3	0.0000
D1	3.3	4.0	0.0000
D2	2.6	3.1	0.0023
D3	3.2	5.6	0.0000
D4	3.8	5.8	0.0000
D5	6.0	6.7	0.0000
D6	2.3	6.8	0.0000
D7	6.7	8.7	0.0000
D8	4.7	8.5	0.0000
D9	4.0	9.2	0.0000
D10	3.1	9.8	0.0000
D11	5.0	12.7	0.0000

5. CONCLUSION

This study introduced a simple effort estimation model for predicting early phase software development projects using data from 317 very recent projects. Results showed that SLOC and application type are significant contributors to development effort (*P* value is 0.0000) (Table 5). The model explains 89% of the variation in software development effort. Thus, the effect of product size on software effort shall be interpreted along with application domain.

Although the model in this study is not highly precise, it has the advantage of providing information on its relative accuracy. The model may also be applicable to non-military sectors, to the extent that their practices are similar to those in the military.

A future study will investigate fixed costs for small projects in the dataset to determine whether additional model form improvements are necessary. Future work will also examine the impact of other cost drivers such as personnel capability, process maturity and requirements volatility.

6. REFERENCES

[1] Boehm B., Software Engineering Economics. Englewood Cliffs, NJ, Prentice-Hall, 1981

[2] Boehm B., Abts C., Brown W., Chulani S., Clark B., Horowitz E., Madachy R., Reifer D., Steece B., Software Cost Estimation with COCOMO II, Prentice-Hall, 2000

[3] Boehm, B., 2000, "Safe and simple software cost analysis," *Software, IEEE*, 17(5), pp. 14–17.

[4] Clark, B., Devnani-Chulani, S., and Boehm, B., 1998, "Calibrating the COCOMO II Post-Architecture model," *Proc. Int'l Conf. Software Eng. (ICSE '98)*, pp. 477–480.

[5] Clark, B. K., 1997, "The Effects of Software Process Maturity on Software Development Effort," Technical Report No. AAT 9816016, University of Southern California, CA.

[6] Galorath, 2001, *SEER-SEM™ User's Manual*, Galorath, Inc., El Segundo, CA, Chaps.1-4, 6-6, 15-6.

[7] Jones C., McGarry J., Dean J., Rosa W., Madachy, R., Boehm B., Clark B., Tan T., 2013, "Software Cost Metrics Manual" http://softwarecost.org/index.php?title=Main_Page, Site accessed on 27 February 2014.

[8] Madachy, R.; Boehm, B.; Clark, B.; Tan, T.; Rosa, W. "US DoD Application Domain Empirical Software Cost Analysis", *Empirical Software Engineering and Measurement (ESEM), 2011 International Symposium on,* On page(s): 392 - 395

[9] Lipkin, I., 2011, "Test Software Development Project Productivity Model", Dissertation Thesis, Univerity of Toledo (Toledo, OH, 2011).

[10] PRICE, 1998, "Your Guide to PRICE S: Estimating Cost and Schedule of Software Development and Support", PRICE Systems, L.L.C, Mount Laurel: NJ, pp. 1–452.

[11] QSM, Inc., 2001, *User's Guide to SLIM-Estimate 5.0*, 2nd Edition, QSM, Inc., McLean,VA, pp. 1–360.

[12] Rosa W., Boehm B., Clark B., Madachy R., Dean J., 2013, "Domain-Driven Software Cost and Schedule Estimating Models: Using Software Resource Data Reports (SRDRs)", 2013 International Cost Estimating and Analysis Association (ICEAA), Professional Development & Training Workshop

[13] Rosa W., Packard T., Krupanand A., James Bilbro J., Hodal M.: COTS integration and estimation for ERP. Journal of Systems and Software 86(2): 538-550 (2013)

[14] Software Resource Data Report, 2011, http://dcarc.cape.osd.mil/Files/Policy/2011-SRDRFinal.pdf

[15] Tan, T, "Domain-Based Effort Distribution Model for Software Cost Estimation," PhD Dissertation, Computer Science Department, University of Southern California, June 20

[16] TECOLOTE, Inc., "Automated Cost Estimating Integrated Tools: CO$TAT", http://ww.aceit.com/Pages/Products/ProductPage.aspx?id=f638a6d8-60e9-414a-9970-7fed249b9d25, Site accessed on 1 July 2014.