ISSN 2229-5518

Analysis of Software Cost Estimation using COCOMO II

T.N.Sharma

Abstract - COCOMO II is an objective cost model for planning and executing software projects. It is an important ingredient for managing software projects or software lines of business. A cost model provides a framework for communicating business decisions among the stake-holders of a software effort. COCOMO II supports contract negotiations, process improvement analysis, tool purchases, architecture changes, component make/buy tradeoffs and several other return-on-investment decisions with a credible basis of estimate. COCOMO II incorporates several field-tested improvements to both broaden its applicability and improve its estimating accuracy for modern software development approaches. COCOMO II includes two underlying information models. The first is a framework for describing a software project, including models for process, culture, stakeholders, methods, tools and the size/complexity of the software product. The second is an experience base that can be used to estimate the likely includes significant updates to COCOMO to improve its applicability to modern processes, methods, tools and technologies. It also includes a much larger, more pertinent database of modern precedents and improves the adaptability of the model so it can be optimized across a broad spectrum of domains and project circumstances. This paper presents cost estimation of various projects using COCOMO II. This article also presents statistical analysis for relevance of base COCOMO II model for effort estimation in present scenario.

Index Terms— Software Cost Estimation, COCOMO II, Scale Factors, Cost Drivers, Case Studies.

1 INTRODUCTION

Software cost estimation is a prediction of the cost of the resources that will be required to complete all of the work of the software project.

Software has a bad reputation about cost estimation. Large software projects have tended to have a very high frequency of schedule overruns, cost overruns, quality problems, and outright cancellations. Instead of it bad reputation, it is important to note that some large software projects are finished on time, stay within their budgets, and operate successfully when deployed.

Currently a new generation of software processes and products is changing the way organizations develop software. The new approaches – evolutionary, risk driven and collaborative software processes; fourth generation languages and application generators; commercial off the shelf (COTS) and reuse driven software approaches; fast track software development approaches; software process maturity initiatives – lead to significant benefit in terms of improved software quality and reduced software cost, risk and cycle time.

T.N.Sharma is undergoing his research work from Department of Statistics, University of Rajasthan, Jaipur (India). Mob.:9414248794. Email: tnsharma@rediffmail.com

COCOMO II model tailored to these new forms of software development, including rationales for the model decisions. The major new modeling capabilities of COCOMO II are a tailorable family of software sizing models, involving Object Points, Function Points, and Source Lines of Code; nonlinear models for software reuse and reengineering; an exponent-driver approach for modeling relative software diseconomies of scale; and several additions, deletions, and updates to previous COCOMO effort-multiplier cost drivers. This model is serving as a framework for an extensive current data collection and analysis effort to further refine and calibrate the model's estimation capabilities.

2. ESTIMATION EQUATIONS

In COCOMO II, the amount of effort in person-months, PM, is estimated by the formula:

The amount of calendar time, TDEV, it will take to develop the product is estimated by

TDEVNS = C x (PMNS)F

$$\text{where } F = D + 0.2 \times 0.01 \times \sum_{j=1}^{5} SFj$$

$$= D + 0.2 \times (E - B)$$

In COCOMO-II effort is expressed as person month(PM). COCOMO II treats the number of person-hours per month, PH/PM, as an adjustable factor with a nominal value of 152 hours/PM.

- The value of n is 16 for the Post-Architecture model effort multipliers, Emi, and 6 for the Early Design model, the number of SFi stands for exponential scale factors.
- The values of A, B, C, D, SF1 …, and SF5 for the Early Design model are the same as those for the Post-Architecture model.

Baseline Effort Constants:
A = 2.94; B = 0.91

$$PM = A \times Size^{E} \times \prod_{i=1}^{n} EM$$

$$\text{where } E = B + 0.01 \times \sum_{j=1}^{5} SF_j$$

Baseline Schedule Constants:

C = 3.67; D = 0.28

2.1 Scale Factors

The application size is exponent is aggregated of five scale factors that describe relative economies or diseconomies of scale that are encountered for software projects of dissimilar magnitude.

- Precedentedness(PREC)
- Development Flexibility (FLEX)
- Architecture / Risk Resolution (RESL)
- Team Cohesion (TEAM)
- Process Maturity (PMAT)

2.2 Cost Drivers/ Effort Multipliers

Cost drivers are characteristics of software development that influence effort in carrying out a certain project. Unlike the scale factors cost drivers are selected based on the rationale that they have a linear affect on effort. There are 17 effort multipliers that are utilized in the COCOMO II model to regulate the development effort.

- Required Software Reliability (RELY)
- Data Base Size (DATA)
- Developed for Reusability (RUSE)

- Documentation Match to Life-Cycle Needs (DOCU)
- Execution Time Constraint (TIME)
- Main Storage Constraint (STOR)
- Platform Volatility (PVOL)
- Analyst Capability (ACAP)
- Programmer Capability (PCAP)
- Personnel Continuity (PCON)
- Applications Experience (APEX)
- Platform Experience (PLEX)
- Language and Tool Experience (LTEX)
- Use of Software Tools (TOOL)
- Multisite Development (SITE)
- Required Development Schedule (SCED)

3. Case Studies

Project 1 Advocate's Desktop

As first case study we have taken a project in consideration which was developed for an very famous advocate firm. The project takes care billing and case information of the firm.

After completion of project we calculated the efforts (Person-Month) using COCOMO II and got the actual time taken to develop the project. Total line of code of C# is 7187 i.e. 7.1 KLOC.

Scale Factors (SF_j)	Very Low	Low	Nominal	High	Very High	Extra High	Our Value
PREC	thoroughly unprecedented	Largely unprecedented	somewhat unprecedented	generally familiar	largely familiar	thoroughly familiar	3.72
	6.2	4.96	3.72	2.48	1.24	0	
FLEX	rigorous	occasional relaxation	some relaxation	general conformity	some conformity	general goals	3.04
	5.07	4.05	3.04	2.03	1.01	0	
RESL	little (20%)	some (40%)	often (60%)	generally (75%)	mostly (90%)	full (100%)	4.24
	7.07	5.62	4.24	2.83	1.41	0	
TEAM	very difficult interactions	some difficult interac-tions	basically cooperative interactions	largely cooperative	highly cooperative	seamless interactions	3.29
	5.48	4.38	3.29	2.19	1.1	0	
PMAT	7.8	6.24	4.68	3.12	1.56	0	4.68

Here

$$\sum_{j=1}^{5} SFj = 18.97$$

$$E = B + 0.01 \times \sum_{j=1}^{5} SFj = 1.097$$

Baseline Effort Constants: **A** = 2.94; **B** = 0.91
Baseline Schedule Constants: **C** = 3.67; **D** = 0.28

Driver	Symbol	VL	L	N	H	VH	XH	Our Value
RELY	EM_1	0.82	0.92	1.00	1.10	1.26		.82
DATA	EM_2		0.90	1.00	1.14	1.28		0.9
CPLX	EM_3	0.73	0.87	1.00	1.17	1.34	1.74	.87
RUSE	EM_4		0.95	1.00	1.07	1.15	1.24	1.0
DOCU	EM_5	0.81	0.91	1.00	1.11	1.23		1.0
TIME	EM_6			1.00	1.11	1.29	1.63	1.0
STOR	EM_7			1.00	1.05	1.17	1.46	1.0
PVOL	EM_8		0.87	1.00	1.15	1.30		0.87
ACAP	EM_9	1.42	1.19	1.00	0.85	0.71		0.85
PCAP	EM_{10}	1.34	1.15	1.00	0.88	0.76		0.88
PCON	EM11	1.29	1.12	1.00	0.90	0.81		0.90
APEX	EM12	1.22	1.10	1.00	0.88	0.81		0.88
PLEX	EM13	1.19	1.09	1.00	0.91	0.85		0.85
LTEX	EM14	1.20	1.09	1.00	0.91	0.84		0.91
TOOL	EM15	1.17	1.09	1.00	0.90	0.78		0.90
SITE	EM16	1.22	1.09	1.00	0.93	0.86	0.80	0.80
SCED	EM17	1.43	1.14	1.00	1.00	1.00		1.0

$$\prod_{i=1}^{n} EM = 0.184295$$

$$PM = A \times SizeE \times \prod_{i=1}^{n} EM$$

$$\text{where } E = B + 0.01 \times \sum_{j=1}^{5} SFj$$

Applying the values on formula :
Here we have

A = 2.94
Size = 7.1

$$E = 1.097$$
$$B = .91$$
$$\prod EM = 0.184295$$

$$PM = 4.67724$$

Actual time taken for this project is 4 Months.
We have applied above COCOMO II formula on 4 software projects which produces us the results as per following table. Also we have taken the actual person month in the table.

S.N.	Name of Project	Technology	KLOC	$\prod_{i=1}^{n} EM$	$\sum_{j=1}^{5} SF_j$	Estimated PM using COCOMO II	Actual PM
1	Advocate's Desktop	C#.Net	7.1	0.184	18.97	4.6	4
2	Online TrueLogic	DotNet	15.01	0.204	18.97	11.83	9.5
3	Online Project Management	Java	7.8	.204	18.97	5.76	4
4	Unit Converter	Android	1.6	.1546	18.97	.76 (22 Days)	.53 (16 Days)

4. CONCLUSION

This brief article shows how to make cost estimates using COCOMO II for a sample project, and outlines basic steps, terms, and tools used. Obviously, ad hoc estimates are prone to error. COCOMO II make it easy for you to clarify not only an expected project cost and duration, but also prompt you to verify all basic sides of a software project by providing clear, compact, and concise terms, methodology, which are tested on a wide range of real-life projects and thus reduce essentially project risks and provide reasonable grounds for communication with a project stockholder. Paper presents difference in between estimation by COCOMO II and actual time taken by the project.

5. REFERENCE:

- Arlene F. Minkiewicz, The Evolution of Software Size: A Search for Value, CrossTalk, The Journal of Defense Software Engineering, April 2009 p.p. 23-26
- Basavaraj, M.J. and Shet, K.C. (2008). Empirical Validation of Software Development effort multiplier of Intermediate COCOMO Model. Journal of Software, vol.3, No. 5 pp. 65-71.
- Boehm Barry (2000). Software Cost Estimation using COCOMO. Prentice Hall, US.
- Barry Boehm, Ricardo Valerdi, COCOMO Suite Methodology and Evolution CROSSTALK The Journal of Defense Software April 2005, p.p. 20-25
- Coe, David j. Improving Consistency of Use Case Points Estimates. CrossTalk, The Journal of Defense Software Engineering, March 2008 pp 8-12
- Demirors, O.; Gencel, C, Conceptual Association of Functional Size Measurement Methods , IEEE Society, Volume 26, Issue 3, May-June 2009 p.p. 71 – 78
- Evans Michael, Alex Abela, and Tom Beltz(2002). Seven Characteristics of Dysfunctional Software Projects. The Journal of Defense Software Engineering.

ISSN 2229-5518

- Ye Yang, Zhihao Chen, Ricardo Valerdi, Barry Boehm, "Effect of Schedule Compression on Project Effort", *CSSE Tech Report, 2005*