International Journal of Scientific & Engineering Research, Volume 5, Issue 6, June-2014
ISSN 2229-5518

Impact of human factors in effort estimation using COCOMO- II

Meenakshi Gaur, T.N. Sharma, Vipin Jain

Abstract- After 1970, software development is continuously progressing with one of the major industry in the world and presently it's dominating the whole world. To gain an understanding of software, it is essential to inspect the characteristics of software that make it different from other things that human beings build. The differences can be defined in three areas.
(1) Software is a logical rather than physical system element
(2) Software is strategized not manufactured
(3) Most software's are custom build.
All these three tasks make the prediction of software developments very tedious job. Software cost estimation has been growing in importance till today. In the 1940's, when the computer era began back there were few computers in use and applications were mostly small, one person projects. As phase moved on, computers became extensive. Applications produced in number, size and importance. Costs to develop software raised as well. As a result of that progress, the consequences of errors in software cost estimation became more severe too. Still today, a lot of cost estimates of software projects are not very precise, mostly too low. This is not an unexpected fact that various difficulties are observed when estimating software costs. The main amount of the total costs of a project arises from the salaries of the employees. Other costs, as license fees or new apparatus for example, occur only once and are not too hard to evaluation. The costs for the human personnel on the other hand are highly correlated to the effort needed to perform the project. Therefore getting an accurate enough estimate of the total effort in order to make a reasonable estimate of the costs is must. This paper presents a study using COCOMO-II cost estimation on a small project of a private company. This paper also shows the impact on complete effort of the project, if the human factors are wrongly estimated.

Index terms- COCOMO II cost estimation, Scale Factor, Effort Multiplier

1. Introduction

The determination of software cost estimation is to:
Define the resources required to create, verify, and validate the software product, and manage these activities.
Enumerate, insofar as is applied, the insecurity and risk inherent in this estimate.

All estimates are made based upon some form of analogy: Expert Judgment, Historical Analogy, Models and Rules-of-Thumb. The part these approaches play in producing an estimate depends upon where one is in the overall life-cycle.
Typically, estimates are made using a combination of these four methods. Model-based estimates along with high-level analogies are the principal source of estimates in early theoretical stages. As a project develops and the requirements and design are better understood, analogy estimates based upon more detailed functional decompositions become the primary technique of estimation, through model-based estimates used as a means of estimate validation or as a sanity-check.

Historical analogy estimation methods are based upon using the software size, effort, or cost of a comparable project from the past. When the term .analogy. is used in this document, it will mean that the comparison is made using measures or data that has been recorded from completed software projects.
Expert judgment estimates are made by the estimator based upon what he or she remembers it took previous similar projects to complete or how big they were. This is typically a subjective estimate based upon what the estimator remembers from previous projects and gets modified mentally as deemed appropriate.
Model-based estimates are estimates made using mathematical relationships or parametric cost models. Parametric cost models are realistic relationships derived by using statistical techniques applied to data from previous projects. . Software cost models provide approximations of cost, effort and schedule.

The cost estimation process includes a number of iterative steps. The reason for the iteration over the different steps is that cost estimation is part of the larger planning and design procedure, in which the system is planned to fit performance, cost, and schedule constraints along with reconciliation and review of the different estimates. Although, in practice, the steps are often performed in a different order and are highly iterative, these steps will be discussed in the sequence that they are numbered for ease of exposition and because this is one of the ideal sequences. Software project plans include product size, estimates of cost, resources, schedules, key milestones and staffing levels. The software estimation process deliberated in the following subsections describes the steps for developing software estimates. Establishing this process early in the life-cycle will result in greater accuracy and credibility of estimates and a clearer understanding of the factors that influence software development costs. This process also delivers methods for project personnel to identify and monitor cost and schedule risk factors.

2. COCOMO II- Cost Estimation Model

The Constructive Cost Model (COCOMO II), developed by Boehm, provides a formula to estimate the number of man-months it will take to develop a piece of software based on the amount of lines of code and a number of project characteristics (scale factors and effort multipliers). The effort, expressed as person-months (PM), can be calculated with the following

ISSN 2229-5518

2.1 COCOMO II-formula:

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

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

The size of a project is expressed in k SLOC. The parameters A and B are constant factors and the values for these two parameters were obtained by calibration of the projects in the COCOMO II database and are initially equal to 2.94 and 0.91 respectively. In the exponent of the formula, one finds the scale factors (SF) that account for the economies or diseconomies of scale encountered for software projects of different sizes. The effort multipliers (EM) on the other hand are factors that have a linear influence on the effort. Effort multipliers are divided into platform factors, product factors, project factors and personnel factors. Each effort is multiplier and scale factor has a range of rating levels from very high to very low. The weight allocated to the rating level of an effort multiplier indicates the amount of extra effort you need compared with a nominal rating level.

2.2 COCOMO II- Cost drivers

Cost Driver Description
RELY: - Required Software Reliability
DATA: - Data base size
RUSE: - Developed for Reusability
DOCU: - Documentation needs
CPLX: - Product Complexity
TIME: - Execution Time Constraints
STOR: - Main storage Constraints
PVOL: - Platform Volatility
ACAP: - Analyst Capability
PCAP: - Programmer Capability
APEX: - Application Experience
PLEX: - Platform Experience
LTEX: - Language and Tool Experience
PCON: - Personnel Continuity
TOOL: - Use of Software Tools
SITE: - Multisite Development
SCED: - Required Development Schedule

2.3 COCOMO II- Scale factors

Scale Factor Description
Precedentedness (PREC):- Reflects the previous experience of the organization. Development Flexibility (FLEX):- Reflects the degree of flexibility in the development process.
Risk Resolution (RESL):- Reflects the extent of risk analysis carried out.
Team Cohesion (TEAM):- Reflects how well the development team knows each other and work together.
Process Maturity (PMAT):- Reflects the process maturity of the organization

3. Case Study

Project: Project Management
We are taking here a Company's project named as "Project Management". This is a small scale project. In this the management of any project is done. After completion of project the effort calculated in Person Month using calibrating the COCOMO-II cost estimation. Total line of code in language is 13362 i.e. 13.36 KLOC

Scale Factors (SFj)	Very Low	Low	Nominal	High	Very High	Extra High	Our Value
PREC	thoroughly unprecedented 6.2	largely unprecedented 4.96	somewhat unprecedented 3.72	generally familiar 2.48	largely familiar 1.24	thoroughly familiar 0	3.72
FLEX	Rigorous 5.07	occasional relaxation 4.05	some relaxation 3.04	general conformity 2.03	some conformity 1.01	general goals 0	3.04
RESL	Little (20%) 7.07	Some (40%) 5.62	Often (60%) 4.24	Generally (75%) 2.83	Mostly (90%) 1.41	Full (100%) 0	5.62
TEAM	very difficult interactions 5.48	some difficult interactions 4.38	basically cooperative interactions 3.29	largely cooperative 2.19	highly cooperative 1.1	seamless interactions 0	Not Applicable
PMAT	7.8	6.24	4.68	3.12	1.56	0	4.68

5

ISSN 2229-5518

$$\sum_{j=1} SFj = 17.06$$

$$E = B + 0.01 \times \sum_{j=1}^{5} SFj = 1.0806$$

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	EM1	0.82	0.92	1.00	1.10	1.26		.82
DATA	EM2		0.90	1.00	1.14	1.28		1.00
CPLX	EM3	0.73	0.87	1.00	1.17	1.34	1.74	.87
RUSE	EM4		0.95	1.00	1.07	1.15	1.24	1.0
DOCU	EM5	0.81	0.91	1.00	1.11	1.23		0.91
TIME	EM6			1.00	1.11	1.29	1.63	1.0
STOR	EM7			1.00	1.05	1.17	1.46	1.0
PVOL	EM8		0.87	1.00	1.15	1.30		0.87
ACAP	EM9	1.42	1.19	1.00	0.85	0.71		0.85
PCAP	EM10	1.34	1.15	1.00	0.88	0.76		1.0
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	0.86
SCED	EM17	1.43	1.14	1.00	1.00	1.00		1.0

$$\prod_{i=1}^{n} EM = 0.22764$$

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

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

Applying the values on formula:
Here we have
A = 2.94
Size = 13.36
E = 1.0806
B = .91
$\prod EM = 0.22764$

PM = 11.02

In above example we have evaluated programmer's capability as 1.0 and analyst's capability as 0.85. Suppose the estimator has wrongly evaluated the value of programmer's capability and analyst's capability (Human Capabilities). In this example if we put various values for PCAP and ACAP we will have the huge difference from the result, we got. See the following table:

ISSN 2229-5518

TABLE 1

PM FOR DIFFERENCE VALUES OF PROGRAMMER'S CAPABILITIES AND ANALYST'S CAPABILITIES

ACAP	PCAP	Multiplication of scale factors	PM
1.42	1.34	0.509588868	24.67
1.42	1.15	0.43733373	21.17
1.42	1.00	0.3802902	18.41
1.42	0.88	0.334655376	16.19
1.42	0.76	0.289020552	13.99
1.19	1.34	0.427049826	20.67
1.19	1.15	0.366497985	17.74
1.19	1.00	0.3186939	15.42
1.19	0.88	0.280450632	13.58
1.19	0.76	0.242207364	11.72
1.00	1.34	0.3588654	17.37
1.00	1.15	0.3079815	14.91
1.00	1.00	0.26781	12.96
1.00	0.88	0.2356728	11.41
1.00	0.76	0.2035356	9.85
0.85	1.34	0.30503559	14.77
0.85	1.15	0.261784275	12.67
0.85	1.00	0.2276385	11.02
0.85	0.88	0.20032188	9.69
0.85	0.76	0.17300526	8.37
0.71	1.34	0.254794434	12.33
0.71	1.15	0.218666865	10.58
0.71	1.00	0.190151	9.20
0.71	0.88	0.167327688	8.09
0.71	0.76	0.144510276	6.99

4. CONCLUSION

This brief article shows that proper evaluation of (human factors) programmer's and analyst capability is must for better and correct cost estimation of any of the software projects. There should be proper mechanism for evaluation of programmer's and analyst's capability including the total experience, language experience, system experience etc.

Value of each effort multiplier plays an important role in the final estimated efforts. In the given table only two of the factors PCAP & ACAP are changed. This can be seen that even in minor changes, in any of the multiplier, causes huge changes in the final estimation.

By this paper I can also conclude that for small scale projects as the example I have taken here, we can also calculate their cost by COCOMO-II Cost Estimation technique by calibration of its factors according to requirement of the project.

5. REFERENCES

[1] Anil Bhardwaj, T.N.Sharma, Anita Bhardwaj(2010), Software Cost Estimation using COCOMO II, Journal of Mathematics and System Sciences, V.6(1) pp.14-22.

[2] T.N.Sharma, Anil Bhardwaj, Jagdish Prasad(2010), On Software Cost Estimation, Advances in Information Theory and Operations Research, pp.329-337, Om Parkash (Ed.)

[3] Mauricio Aguiar, TI Metricas, on COCOMO II Local Calibration Using Function Points

[4] Sagar Sunkle, Vinay Kulkarni on Interpretation of COCOMO II Cost Drivers for Cost Estimation in Model-driven Engineering

[5] T.N.Sharma, Anita Sharma, Anil Bhardwaj (2011), Programmer's Capability on Software Cost Estimation, p.p. 349-355 , Vol 10 No. 4

[6] Boehm Barry (2000). Software Boehm Cost Estimation using COCOMO. Prentice Hall, US.

- *Ms Meenakshi Gaur is currently pursuing masters degree program in Computer Science engineering from Rajasthan Technical University(SKIT College), Jaipur, Rajasthan, India, PH-09887660092, E-mail: meenakshi.career@gmail.com*
- *Dr. T.N. Sharama, P.hd. in Computer Science engineering from Rajasthan University, Jaipur, Rajasthan, India, E-mail: tnsharma@rediffmail.com*
- *Mr Vipin Jain, M.Tech. in Computer Science engineering from Gyan Vihar University, Jaipur, Rajasthan, India, E-mail: vipin@skit.ac.in*