AN AUTOMATED SOFTWARE COST AND SCHEDULE MEASUREMENT TOOL FOR OO SYSTEM USING PREDICTIVE OBJECT POINT SIZING METRICS

[1]Vijay Yadav,[2]Dr. Vibhash Yadav,[3]Prof. Raghuraj Singh
[1]Research Scholar,[2]Associate Professor,[3]Professor
[1]Computer Science and Engineering,
[1]AKTU,Lucknow,India

Abstract— Software cost measurement is still a challenging task for software developers. It is the prediction of the resources required for a software development process. The accurate and timely estimation of cost at the early phase of its development or when the architecture of project is fully understood is required. While estimating software cost, system size is widely taken as main parameter of the system development effort. In this paper, using Predictive Object Point software sizing metrics set, first the prediction of effort is done and then the estimation of cost as well as schedule of an OO system for post architectural has been done. For the estimation of effort, schedule and cost a tool named APA has been made using COCOMO II model and POP calculator. It takes POP count and converts it into LOC_{POP} using the linear regression equation and thus calculates effort, schedule and cost of the project. The results are presented here to show that POP Count can be used to estimate the effort, schedule and cost of an object oriented system.

Index Terms— Cost estimation, schedule measurement, Effort prediction, object-oriented measurement, Software Metrics,Predictive Object Point, Tool and automation.

I. INTRODUCTION

Estimating cost is an important aspect in software engineering. It often results in the success or failure of the deal between the two parties i.e. the company and the customer. Effort, schedule and staffing are the important derivable for cost estimation as they are important inputs for activities such as staffing, planning, bidding etc. In order to estimate the effort, schedule and cost many models have been developed in the past such as SEER-SEM [9], SLIM [7], PRICE-S [8], Knowledge Plan [10] and COCOMO[2]. These models still have their roots as these are the core models developed in early 1980's and 90's. They are still investigated, widely used in practice, literature and their adaptive nature makes them robust and useful [1]. The model taken into consideration in this paper is COCOMO II [2] for several reasons:

- COCOMO II has been integrated by several companies for estimation purpose like PRICE System.
- COCOMO is an open model so all detail such as estimation equations, the values that are used by the cost drivers and scale factors are available [3].
- As the model is well defined and does not depend upon proprietary algorithms therefore can be calibrated to any software development environment according to their needs in order to provide accurate results.

II. AN INTRODUCTION TO COCOMO II MODEL

COCOMO II [2] model was developed by Barry Boehm. It is upgrade model from the early release of its ancestors i.e. Basic COCOMO and Intermediate COCOMO. The model uses Function Points (FP), Source Lines of Codes (SLOC) and Object Points as size estimates.

The Model has Five Scale Factors and 17 effort multipliers or cost drivers for post architectural and 7 effort multipliers for early design. These scale factors and effort multipliers play an important role in the estimation of effort, which in turn will be used to estimate cost. The scale factors and effort multipliers are given as follows:

A. Scale Factors

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

B. Cost Drivers/Effort Multipliers

a. Product Factors

Required Software Reliability (RELY)

- Data Base Size (DATA)
- Product Complexity (CPLX)
- Developed for Reusability (RUSE)
- Documentation Match to Life-Cycle Needs (DOCU)

b. Platform Factors

- Execution Time Constraint (TIME)
- Main Storage Constraint (STOR)
- Platform Volatility (PVOL)

c. Personal Factors

- Analyst Capability (ACAP)
- Programmer Capability (PCAP)
- Personnel Continuity (PCON)
- Applications Experience (APEX)
- Platform Experience (PLEX)
- Language and Tool Experience (LTEX)

d. Project Factors

- Use of Software Tools (TOOL)
- Multisite Development (SITE)
- Required Development Schedule (SCED)

COCOMO II uses the following formula to calculate effort which is in PM (Person Month), and amount of calendar time i.e. TDEV.

A. Effort Calculation

$$PM = A*(size)^{E} * \prod_{i=1}^{n} EM_i \qquad (1)$$

here n is 17 for post architecture and 7 for early design

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

B. Calendar Time (Schedule)

$$TDEV = C*(PM)^{F} \qquad (2)$$ Where,

$F = D + 0.2 * 0.01 \sum_{j=1}^{5} SF_j$ And

$D = 0.2 * (E-B)$

Where A and B are the Baseline Effort Constants and C and D are Baseline Schedule Constants having the following values:
A = 2.94, B = 0.91, C = 3.67 and D = 0.28
Given below is the table of cost drivers and effort multipliers as in COCOMO II 2000 [3].

Table - 1

The List of Scale Factors and Cost Drivers along with their values.

Drivers	Symbol	VL	L	N	H	VH	XH
PREC	SF_1	6.20	4.96	3.72	2.48	1.24	0.00
FLEX	SF_2	5.07	4.05	3.04	2.03	1.01	0.00
RESL	SF_3	7.07	5.65	4.24	2.83	1.41	0.00
TEAM	SF_4	5.48	4.38	3.29	2.19	1.10	0.00
PMAT	SF_5	7.80	6.24	4.68	3.12	1.56	0.00
RELY	EM_1	0.82	0.92	1.00	1.10	1.26	
DATA	EM_2		0.90	1.00	1.14	1.28	
CPLX	EM_3	0.73	0.87	1.00	1.17	1.34	1.74
RUSE	EM_4		0.95	1.00	1.07	1.15	1.24
DOCU	EM_5	0.81	0.91	1.00	1.11	1.23	
TIME	EM_6			1.00	1.11	1.29	1.63
STOR	EM_7			1.00	1.05	1.17	1.46
PVOL	EM_8		0.87	1.00	1.15	1.30	
ACAP	EM_9	1.42	1.19	1.00	0.85	0.71	
PCAP	EM_{10}	1.34	1.15	1.00	0.88	0.76	
PCON	EM_{11}	1.29	1.12	1.00	0.90	0.81	
APEX	EM_{12}	1.22	1.10	1.00	0.88	0.81	
PLEX	EM_{13}	1.19	1.09	1.00	0.91	0.85	
LTEX	EM_{14}	1.20	1.09	1.00	0.91	0.84	
TOOL	EM_{15}	1.17	1.09	1.00	0.90	0.78	
SITE	EM_{16}	1.22	1.09	1.00	0.93	0.86	0.80
SCED	EM_{17}	1.43	1.14	1.00	1.00	1.00	

Where **VL:** Very Low, **L:** Low, **N:** Nominal, **H:** High, **VH:** Very High, **XH:** Extra High

III. POP METRIC AS EFFORT ESTIMATOR

The problem of cost estimation in software engineering has been addressed by many articles and books [11]. Research shows that many factors can affect the development cost of software projects, including human, organization, and process factors. However, one important factor is the size of the product to be developed [11][17]. Many measures of size have been proposed, ranging from source lines of code to functional size measures such as function points, Predictive Object Point (POPs) or use case conversion points (UCCPs) etc. Function points measurement is inadequate for object oriented software for prediction of efforts.

While focusing on the impact of object-oriented design properties on development effort, we chose Predictive Object Point Sizing Metric introduced by PRICE Systems [12][18] to measure the size of OO projects designed specifically for Object Oriented software and result from the measurement of the object-oriented properties of the system. The reason being POP metric is the better indicator of the size of Object Oriented software projects than did the FP metric [5] [6].

First the size may be estimated in terms of Estimated KLOC (LOC_{POP}) through regression analysis method [15]. The below mentioned formulation is used [15] for estimation in terms of size.

$$LOC_{POP} = b_0 + b_1 * POP$$

Where b_0 and b_1 are calculated as follows:

$$b_0 = (\text{avg KLOC}) - b_1 * (\text{avg POP}) \qquad (3)$$

$$b_1 = \frac{(\sum POP * KLOC) - \left[\frac{(\sum POP) * (\sum KLOC)}{n}\right]}{\left(\sum POP^2\right) - \left[\frac{(\sum POP)^2}{n}\right]} \qquad (4)$$

After size, the cost of object oriented software may be evaluated through POP.

A. Methodology Used:

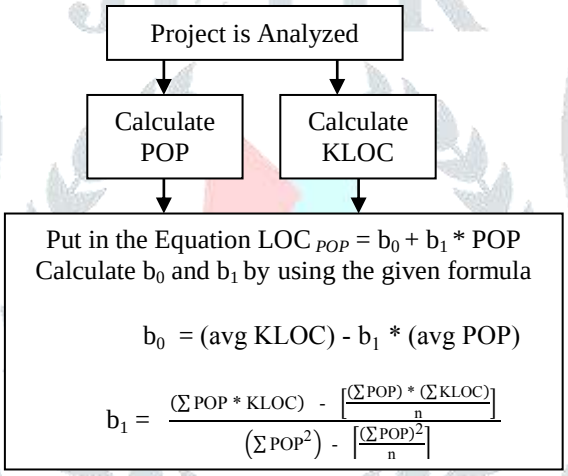

Fig.1: The Linear Regression Model

The Fig.1 above shows the linear regression model which is used as the base of the proposed model shown in Fig.2. This base model uses linear regression equation which is of the form $LOC_{POP} = b_0 + b_1 * POP$, where b_0 and b_1 are the regression constant for which the previous values of KLOC and POP of analyzed projects is required. The values of these metrics will be stored in the database of the tool which is being developed for future analysis and calculations.

B. The Proposed Model:

The model being proposed is given in Fig.2. This model will use the regression system in order to calculate the LOC_{POP} through POP and then will estimate the effort and cost. Here whenever a new project is imported in the model it will calculate its POP and KLOC and then automatically update the values of the regression constant b_0 and b_1. It then uses the POP of the imported project to calculate new value of LOC_{POP} and then by using this value of LOC_{POP} it will calculate the effort and the cost of that imported project. Thus firstly KLOC and POP values are used in order to calculate the updated values of b_0 and b_1 and then only the POP value is used in order to calculate LOC_{POP} each time whenever a new project being analyzed. This is so because here the analysis is done considering the post architectural rather than that of early design in which there is no idea about how much KLOC will be needed which can lead to inaccurate measurement as the estimations are rough in early design. Whereas the post architecture covers the actual development and maintenance of software products, it is used when top-level design is complete and hence detailed information is known about the project [4][16]. Thus in post architecture KLOC is known. This is the reason why each time continuous updating of the b_0 and b_1 is required whenever any new project is analyzed. This maintains the accuracy in the relationship between LOC_{POP} and POP.

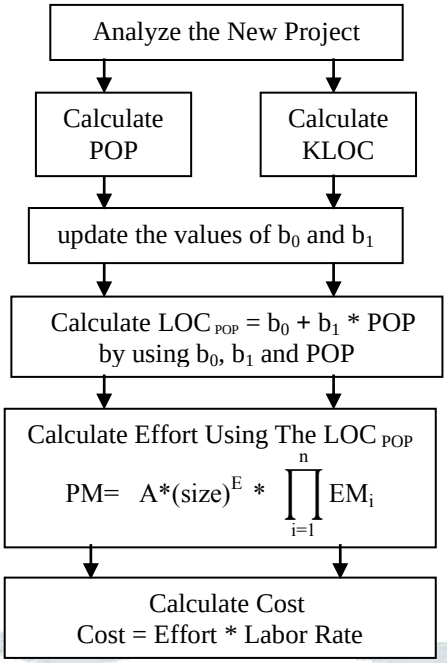

Fig.2: The Proposed Model for estimation of cost using POP Metrics Count

IV. Analysis And Results

For the estimation of effort, schedule and cost a tool named as APA has been made. The tool uses COCOMO II model. It takes POP count and converts it into LOC_{POP} using the linear regression equation and thus calculates effort, schedule and cost of the project.

The proposed formula for Effort estimation in person month is shown below:

$$PM = A*(size)^E * \prod_{i=1}^{n} EM_i \qquad (5)$$

Where $E = B + 0.01* \sum_{j=1}^{5} SF_j$ and size is in the form

$$LOC_{POP} = b_0 + b_1 * POP.$$

Calendar Time (Schedule)

$$TDEV= C * (PM)^F \qquad (6)$$ Where,

$$F = D + 0.2 * 0.01 \sum_{j=1}^{5} SF_j \quad \text{and}$$

$$D = 0.2* (E-B)$$

Where A and B are the Baseline Effort Constants and C and D are Baseline Schedule Constants having following values A = 2.94, B = 0.91, C = 3.67 and D = 0.28 [3]

The cost then can be evaluated as,

$$Cost = Effort * Labor Rate \qquad (7)$$

Fig.3 shows the snapshots of the tool APA for estimation of cost of software projects. It calculates the value of the constant b_0 and b_1 and the Labor Rate has to be entered by the user. The tool also shows the POP and calculated LOC_{POP}. It also estimates the effort for software project.

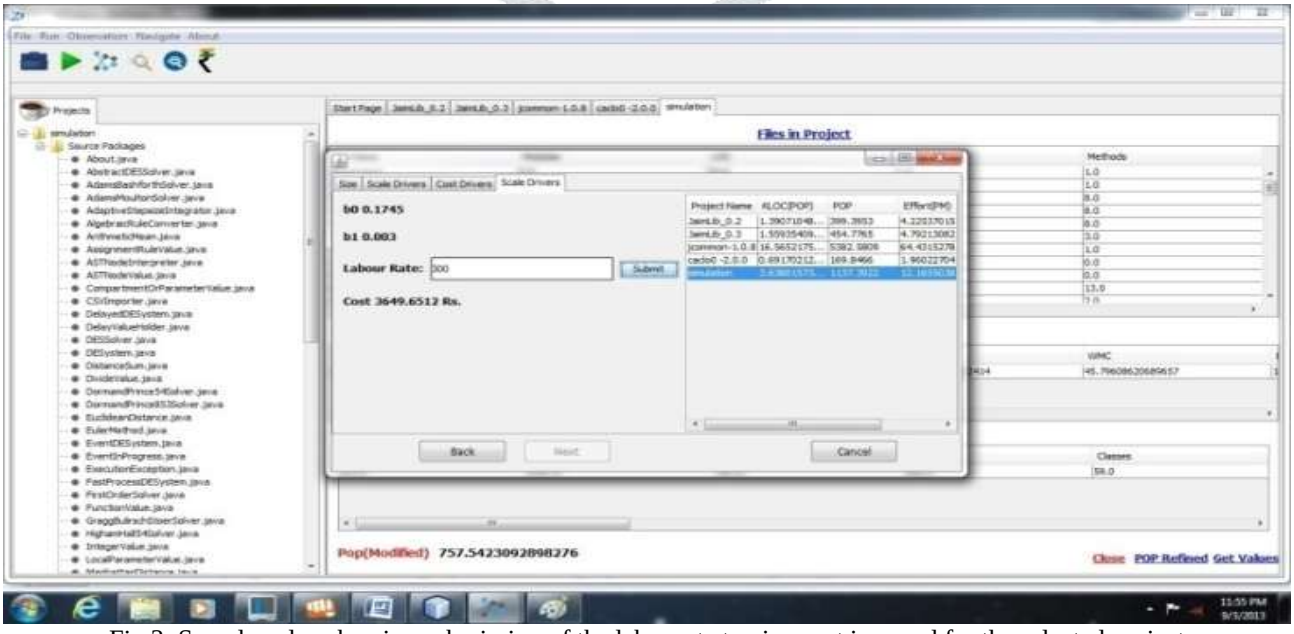

Fig.3: Sample value showing submission of the labor rate to give cost incurred for the selected project

Fig.4 shows the snapshots of the tool showing estimation of estimated cost, POP, LOC_{POP}, measured effort and finally estimated schedule of software projects.

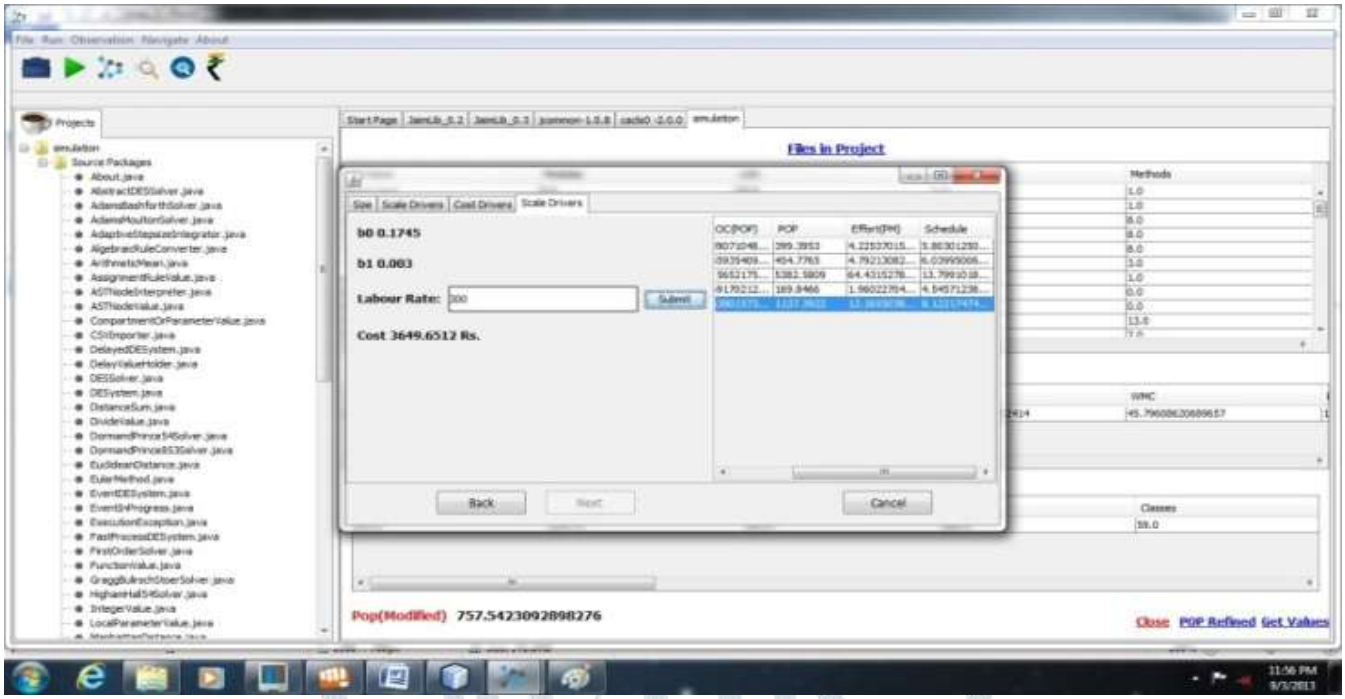

Fig.4: Sample value showing calculated schedule required to build up the selected project.

V. Conclusion And Future Work

From the results presented in this study, we may conclude that Predictive Object Point Metrics can be used to build Useful cost estimation model for object-oriented systems. Effort prediction may be done using POP count and consequently schedule and cost may also be estimated for software projects.

The nature of the relationships between design product properties and cost in object-oriented (OO) systems has been understood. Size, based on design information through POP Count shows a strong relationship to cost. A tool named APA has been built for estimation of cost and schedule by using POP Count for software project based on object oriented properties.

This study hence not only helped to get some understanding of the systems but also proved that POP a better effort predictor can be used to estimate cost and schedule of software projects. Another future study prospect would be to get an idea of how accurate can cost predictions be based on such information at different stages of the development. Although that answer depends on the specific context of application.

VI. Acknowledgment

I am feeling great sense of self-satisfaction and good experience having accomplished my Ph.D research paper entitled “An Automated Software Cost and Schedule Measurement Tool for OO System using Predictive Object Point Sizing Metrics”. I would like to thanks Dr. Vibhash Yadav (Associate Professor), Department of Information Technology, Rajkiya Engineering College, Banda, my Supervisor for his guidance, support, motivation and encouragement throughout the period this work was carried out. His readiness for consultation at all times, his educative comments, his concern and assistance even with practical things have been invaluable.

I would like to express special thanks to Prof. Raghuraj Singh (HOD Computer Science and Engineering, HBTU, Kanpur) my research guide who have contributed their precious time and effort to help me. Without whom it would not have been possible for me to understand and complete my paper.”.

References

[1] Vu Nguyen,(2010)," Improved Size and Effort Estimation Models For Software Maintenance", Phd Dissertation ,University Of Southern California.

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

[3] Center for Software Engineering (2000), “COCOMO II Model Definition Manual,” Computer Science Department, University of Southern California, Los Angeles, Ca. 90089, URL :http://csse.usc.edu/csse/research/COCOMOII/cocomo_main.html.

[4] Clark B., Chulani S., and Boehm B.W., (1998), “Calibrating the COCOMO II Post Architecture Model,” International Conference on Software Engineering.

[5] Judge T.R and Williams A. ," OO Estimation – an Investigation Of The Predictive Object Points (POP) Sizing Metric in an Industrial Setting", Parallax Solutions Ltd, Coventry, UK.

[6] Shubha Jain., Vijay Yadav and Raghuraj Singh., (2013). “OO Estimation through Automation of thePredictive Object Points Sizing Metric”, International Journal of Computer Engineering and Technology (IJCET), Volume 4, Issue 3, pp 410 - 412.

[7] Putman, L.H,"A General Empirical Solution To The Macro Software Sizing And Estimation Problem", IEEE Trans. Soft. Eng. SE-4, 4(1978).

[8] Freiman, F.R and Park, R.E, (1979) ," The Price Software Cost Model : RCA Government System Division", IEEE .

[9] Galorath 2002, SEER-SEM Software Estimation, Planning and Project Control - User's Manual, Galorath Incorporated.

[10] KnowledgePlan (2005), Software Productivity Research LLC, “KnowledgePlan User’s Guide, Version 4.1”, SPR.

[11] L. Briand, K. El Emam, K. Maxwell, D. Surmann, I. Wieczorek, "An Assessment and Comparison of Common Software Cost Estimation Models". In: Proceedings of the 21st International Conference on Software Engineering, ICSE 99, (Los Angeles, USA 1999) 313-322.

[12] F. Minkiewicz,(1999),"Measuring Object Oriented Software with Predictive Object Points", Project Control for Software Quality, A. C. Rob Kusters, Fred Heemstra and Erik van Veenendaal, Ed.: Shaker Publishing, 1999.

[13] Ghazy Assassa, Hatim Aboalsamh, Amel Al Hussan, "Software cost estimation: A tool for object oriented console applications", Proceedings of the 1st National Information Technology Symposium, NITS, Riyadh, Saudi Arabia, 2006, pp. 276-284. http://www.ksu.edu.sa/sites/Colleges/ComputerSciences/Documents/NITS/ID164.pdf

[14] Al-Hussan,A.A,(2005),"Cost Estimation Model for Object Oriented Software Console Applications ", M.Sc Project, Department of Computer Science , College of Computer and Information Sciences, King Saud University .

[15] Shubha Jain., Vijay Yadav and Raghuraj Singh., (2013)." An Approach for OO Software Size Estimation using Predictive Object Point Metrics" International Conference on Computing for Sustainable Global Development (INDIACom)., IEEE sponsors Delhi Section.,Bharati Vidyapeeth, New Delhi.

[16] Vijay Yadav, Shubha Jain, Prof. Raghuraj Singh.,(2014). "Mapping Predictive Object Points(POP) with Software Size in OO Environment"., International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463 Vol. 3 Issue 1, January-2014, pp: (395-399), Impact Factor: 1.252, Available online at: www.erpublications.com. (Published).

[17] Vijay Yadav, Dr. Vibhash Yadav, Prof. Raghuraj Singh.,(2016). "Introducing New OO Metric For Simplification In Predictive Object Points (POP) Estimation Process In OO Environment"., International Journal Of Engineering Sciences & Research Technology (IJESRT), ISSN: 2277-9655 Vol. 5 Issue 1, January-2016, pp: (716-723), Impact Factor: 3.785, Available online at: www.ijesrt.com. (Published).

[18] Vijay Yadav, Dr. Vibhash Yadav, Prof. Raghuraj Singh.,(2016). "Identifying The Relationship Between Original And Refined POP Metric For OO Software"., International Journal Of Research Fellow For Engineering (IJRFE),ISSN: 2320-7396 Vol. 4 Issue 11, November-2016, pp: (39-46),Available online at: www.ijrfe.com. (Published).