

# Efforts Estimation by Combining the Use Case Point and COCOMO

Chetan Nagar

Ph.D Student, Mewar University (Gangrar)  
Chittodgarh Rajasthan India

Anurag Dixit

Phd, Dean-Professor (CS/IT) BRCM CET, Bahal  
Bhiwani

## ABSTRACT

Software efforts estimation is tedious task for every software industry. Many software efforts estimation models are invented to make efforts estimation accurate. Unfortunately no model is suitable for all kind of software industries. This paper used Use Case method for efforts estimation for a small software company. For many projects we have not got good results. These papers modify Use Case Point and apply on same project of same company and we have got some better result. This paper modify the method Jorgensen has described 12 expert-based best practices, one of the best practices said that combine estimates from different experts and estimation strategies [2]. It is always suggested that we must use more than one method for estimation, but there is no model is exists which support this concept. This paper combines the Use Case point and COCOMO. We are predicting the Line of Code with the help of Use Cases. Use Case used in the method must be more specific not more generalized. More recently; the use of Use Cases for software effort estimation has gained wide popularity. Researchers from academia as well as industry have shown interest in the Use Case based approaches because of the promising results obtained along with their early applicability

A strong monitoring policy is always required to make estimation as a success. We have to make a check list with the date of completion and must follow the checklist. If work is not done on the time some necessary action must be taken to compensate the deviation [14].

## Keywords

KLOC (Kilo Line of Code), UCP (Use Case Point), FP (Function Point), these are all unit of software size. Software Efforts estimation, Person-month, Person-Hours these are units of efforts.

## 1. INTRODUCTION

For a better efforts estimation it is always suggest that we must use more than one method. The efforts estimated by the different methods must be compare to check the difference. If difference is more then we need to recalculate it. COCOMO and Use Case Point are two most widely used methods for the efforts estimation. KLOC is key input for COCOMO, to predict the KLOC we have to divide the project into module, module into sub module and sub module into function until it become easy to predict the KLOC required to build this function. Use Case contains the functional requirement of the system, So we can directly predict KLOC from the Use Case.

In general following methods are more popular for efforts estimation.

- A. COCOMO Model [9][10].
- B. Function Point Based Estimation [9].

- C. Use Case Point Based Estimation [1][3][6][4][7][8][15].
- D. Expert Judgment.[11][12][13].
- E. Estimation by Analogy [11].
- F. Software Efforts Estimation using Soft Computing Techniques.
- G. Software efforts estimation using Neural Network Techniques.

The above specified methods are more common and this are used by industries. Excluding this methods method there are many other methods, but now a days those are not much popular. In this paper section II contain overview of Use Case Point approach and COCOMO model. Section III containing the new way of estimation suggested by this paper. Section IV containing how we can validate the efforts calculated by the suggested method, section V suggests a monitoring policy with Use Case, section VI show the result obtained by using this methods and Section VII concludes the work.

## 2. METHODS OF ESTIMATION

There are lot of methods which can be use for efforts estimation, but an industry wants a simple and accurate way of efforts estimation. We must use more calculative method as compare than more predictive approach. COCOMO and Use Case Point are more calculative approach which is covering many factors that may affect the cost. But Expert judgment is more predictive approach, in which much experience is required. A rich set of old data is required for better estimation.

COCOMO is one of the popular and old model of efforts estimation. In this model we have to estimate the line of code. Counting of line of code is one of the difficult tasks when project is complex and new to us. In such situation we have to divide that project in module and divide that module into sub module to make problem less complex. Most senior person of your team should take responsibility to count KLOC, because we need to estimate KLOC before writing it. We can use old data to predict KLOC, but no project will completely same with previous project, so some intelligence and experience will be required for a better estimation. Here we will use advance COCOMO for estimation.

### A. COCOMO

$$\text{Efforts} = a * (\text{KLOC})^b * \text{EAF}$$

Here a and b are complexity factor.

**TABLE I  
COMPLEXITY FACTOR**

Model	A	B
Organic (simple in terms of size and complexity)	3.2	1.05
Semi-ditched ( average in terms of size and complexity)	3.0	1.12
Embedded ( Complex)	2.8	1.20

In Intermediate COCOMO only 17 EAF are used, but in advance COCOMO we are using 22 EAF. Typical values for EAF range from 0.9 to 1.4.

#### B. Use Case Point based Estimation [[1][3][4][6][7][8][15]

It is another popular and efficient method of efforts estimation. Here we will calculate use case and actors

$$UUCP = \text{Use case} + \text{Actors}$$

Using the following table1 we can calculate Use Case used in a project

**TABLE II  
USE CASE CALCULATION**

Use case type	Description	Quantity	Weight Factor	Sub total
Simple	3 or fewer transaction		1	
Average	5 to 7 transaction		5	
Complex	Greater than 7 transaction		10	
<b>TOTAL</b>				

By using the following table2, we can estimate actors used in a project

**TABLE III  
ACTOR CALCULATION**

Use case type	Description	Quantity	Weight Factor	Sub total
Simple	3 or fewer transaction		1	
Average	5 to 7 transaction		2	
Complex	Greater than 7 transaction		3	
<b>TOTAL</b>				

$$UCP = UUCP * TCF * EF$$

TCF is Technical Complexity Factor, which is sum of 13 complexity parameters.

EF is Experience Factor, which is sum of 08 complexity parameters.

$$\text{Effort} = UCP * ER \text{ (Efforts will be in man hours)}$$

ER is efforts rate. ER is the number of man hours will require to write a UCP.

### 3. COMBINATION OF USE CASE POINT AND KLOC AS METHOD OF EFFORTS ESTIMATION.

This paper is suggesting a method in which we are combining COCOMO and Use Case Point; here we will estimate KLOC on basis of Use Case Point. We will predict no of KLOC will required to build the Use Case. Use Case shows the functional requirement of the system so it is the best way to estimate the efforts by using the use case point.

#### A Use Case Point

**TABLE IV  
USE CASE CALCULATION**

Use Case Type	Description	Quantity	Weight Factor	Sub total	No of KLOC per Use Case	Sub Total
Simple	3 or fewer transaction		1			
Average	5 to 7 transaction		5			
Complex	Greater than 7 transaction		10			
<b>TOTAL</b>						

We also required some amount of KLOC to write actor (it is a person or system interacting with our system)

**TABLE V  
ACTOR CALCULATION**

Use Case Type	Description	Quantity	Weight Factor	Sub total	No of KLOC per Actor	Sub Total
Simple	Simple actors are those which communicate to system through API.		5			
Average	Actors who are interacting with the system through		10			

	some protoco l					
Com plex	Comple x actor is interacti ng normall y through GUI.		15			
<b>TOTAL</b>						

Now we will use two methods Use Case Point and COCOMO  
First we are using Use Case Point

$$UUCP = UAW + UUCW$$

$$UCP = UUCP * TCF * EF$$

Effort = UCP \* No of Man Hours for signal UCP.

Schneider and winters proposed number of staff hours per Use Case point depends on the environmental factors. The number of factors in E1 through E6 that are below 3 are counted and added to the number of factors in E7 through E8 that are above 3. If the total is 2 or less, the general idea is to use twenty staff hours per UCP; if the total is 3 or 4, use twenty-eight staff hours per UCP. If the number exceeds 5, it is usually recommended that changes should be made to the project so the number can be adjusted, because in this case, the risk is unacceptably high. Another possibility is to increase the number of staff hours to thirty-six per Use Case points.

B. Second we are using COCOMO

$KLOC = KLOC$  estimated by Use Case Point approach +  $KLOC$  required for GUI +  $KLOC$  required to achieve non functional requirements which is not covered in Use Case Point Approach .

$$Efforts = a * (KLOC)^b * EAF$$

EAF is efforts adjustment Factors. In Intermediate COCOMO only 17 EAF are used, but in advance COCOMO we are using 22 EAF. Typical values for EAF range from 0.9 to 1.4.

**TABLE VI**  
**COST DRIVERS**

S No	Cost Driver	Value	Description
1	DATA		Database size.
2	CPLX		Product complexity.
3	TIME		Execution time constraint.
4	STOR		Main storage constraint.
5	RUSE		Required reusability.
6	DOCU		Documentation match to life-cycle needs.
7	PVOL		Platform volatility.
8	SCED		Scheduling factor.

9	RELY		Required reliability.
10	TOOL		Use of software tools.
11	APEX		Application experience.
12	ACAP		Analyst capability.
13	PCAP		Programmer capability.
14	PLEX		Platform experience.
15	LTEX		Language and tools experience.
16	PCON		Personnel continuity.
17	SITE		Multisite development.

Scale factors are new in COCOMO II. The effect of scale factor is in 1.01 to 1.26 ranges

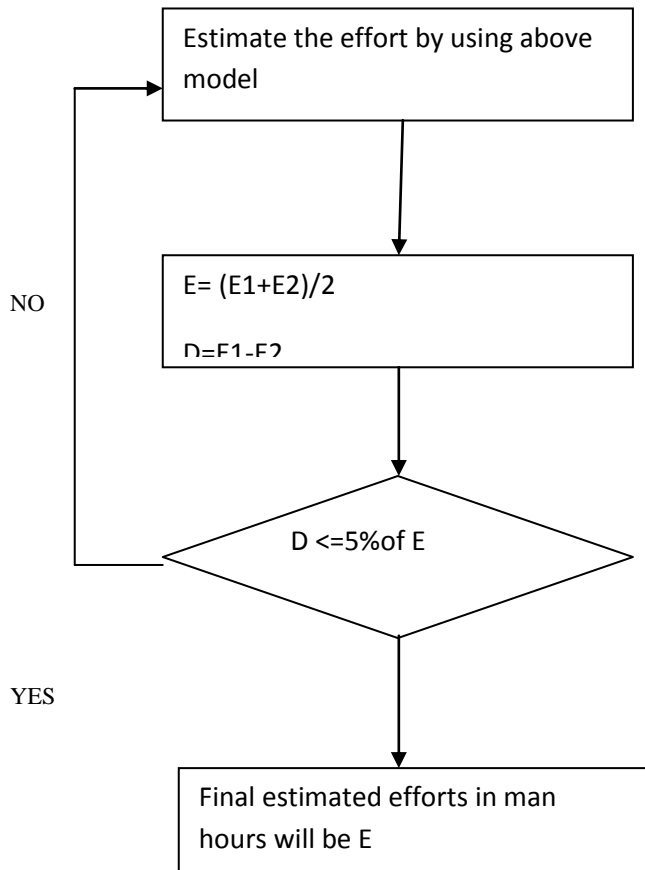
**TABLE VII**  
**NEW COST DRIVERS**

S No	Cost Driver	Value	Description
18	PREC		Precedence.
19	PMAT		Process maturity.
20	TEAM		Team cohesion.
21	FLEX		Development flexibility.
22	RESL		Architecture and risk resolution.

COCOMO model estimate efforts in Person-Month, We will multiply 152 into efforts estimated by COCOMO to convert Person-Month into Man-Hours [16][17].

#### 4. VALIDATION OF EFFORTS ESTIMATION

We have used two methods for the efforts estimation one was Use Case Point and another one was COCOMO. Now we have to compare efforts estimated by these two methods. If we have found the difference is more than 5 % of the average of efforts estimated by both these methods than recalculation for both the methods are required.



**FIGURE 1: VALIDATION OF EFFORTS ESTIMATED USING USE CASE POINT AND KLOC**

**For small project  $D \leq 8\%$  of  $E$**

$E1$  will be effort estimated by Use Case Point approach and  $E2$  will be the efforts estimated by COCOMO and  $D$  will be difference between  $E1$  and  $E2$ .

## 5. MONITORING POLICEIS

Strong monitoring policy is always required to make the project success. Use the following table to make a check list along with the date of completion.

**TABLE VIII  
CHECK LIST USING USE CASE**

S No	Name of Use Case	Date of Completion	Actual date of Completion	Action Taken to Compensate

All the use case of the project must be listed in this table, if any use case is not completing on the time, some action must be taken to compensate it. Monitoring must be at a regular interval and this interval must be very short. a better project management can save an immense amount of revenue along with the better quality control and optimized development

schedules. Success of schedule is depending on the monitoring policies used [14].

## 6. RESULT

We have applied this approach on some successful project s of a small company and we have got following result.

**TABLE IX  
RESULTS**

S No	Project	Efforts Estimated (Man hours )		Actual Efforts	Remark
1	A	1590		1670	
2	B	1320	1600	1585	
3	C	2720		2770	
4	D	1272		1245	

In the project B we got two estimations first is too below from the actual efforts because we are using more generalised Use Case and second one is closer than actual efforts because we divide this generalised Use Cases into more specific cases.

## 7. CONCLUSION

To estimate the KLOC we have to divide the project into module and module into the sub module until we are able to estimate the KLOC. Use Case Point show the functional requirement of the system .So it is one of the good way of estimation .We know that in this paper most of the portion is known.. We tried to illustrate that how we can combine Use Case and KLOC.

## 8. REFERENCES

- [1] Anda, B., Benestad, H.C., Hove, S.E.: “A multiple case study of Effort Estimation based on Use Case Points” Empirical Software Engineering, 2005
- [2] M. Jorgensen. A review of studies on expert estimation of software development effort.Journal of Systems and Software, 70(1-2):37–60, 2004
- [3] Diev, S.: Use Cases modeling and software estimation: Applying Use Case Points. ACM Software Engineering Notes, November 2006
- [4] Ochodek, M., Nawrocki, J., Kwarciak, K.: Simplifying effort estimation based on Use Case Points. Journal of Information and Software Technology, 0950-5849 © 2010 Elsevier B.V
- [5] Keung, J.; NICTA Ltd., Sydney, NSW “Software Development Cost Estimation Using Analogy: A Review” Software Engineering Conference, 2009. ASWEC '09. **On page(s):** 327 – 336.
- [6] <http://www.codeproject.com/KB/architecture/usecasepoi nts.aspx>.
- [7] Edward R Carroll “Estimating Software Based on Use Case Point” Proceeding OOPSLA '05 Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, Systems, languages, and applications

- [8] Mohammed Wajahat Kamal and Moataz A.Ahmed “ A Proposed Framework for Use Case based Effort Estimation using Fuzzy Logic: Building upon the outcomes of a Systematic Literature Review ” International Journal on New Computer Architectures and Their Applications (IJNCAA) 1(4): 976- 999 The Society of Digital Information and Wireless Communications, 2011 (ISSN: 2220-9085)
- [9] Roger E Masse “An Analysis of the Evolution of COCOMO and Function Points ” “University of Maryland, July, 1997.
- [10] Basavaraj M.J, Dr. K.C Shet “Empirical validation of Software development Effort multipliers of Intermediate COCOMO Model” JOURNAL OF SOFTWARE, VOL. 3, NO. 5, MAY 2008.
- [11] <http://www.scribd.com/doc/51198168/45/Expert-Judgment-Method>
- [12] Gray, A.R.; MacDonell, S.G.; Shepperd, M.J “Factors systematically associated with errors in subjective estimates of software development effort: the stability of expert judgment”, Software Metrics Symposium, 1999. Proceedings. Sixth IEEE International IssueDate: 1999 On page(s): 216 – 227.
- [13] Orgensen, M:” Practical guidelines for expert-judgment-based software effort estimation” Software, IEEE **Issue Date:** May-June 2005 **Volume:** 22 **Issue:**3 **On page(s):** 57 – 63.
- [14] Chetan Nagar and Dr Anurag Dixit, “Software Project Management with Control Based Monitoring” International Journal of Advances in Science & Technology, October-2011, volume 3 No 4.
- [15] M. M. Ochodek, J. Nawrocki , K. Kwarciak “Simplifying effort estimation based on Use Case Points” , Journal Information and Software Technology Volume 53 Issue 3, March, 2011 , Butterworth-Heinemann Newton, MA, USA.
- [16] <http://www.softstarsystems.com/faq.htm>
- [17] B. Boehm. *Software Engineering Economics*. Prentice Hall, 1981