International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161

Available at http://inpressco.com/category/ijcet

General Article

Combining COCOMO and use case for better effort estimation

Inderpreet Kaur Walia[Å*] and Manpreet Kaur[Å]

[Å]Department of Computer Science and Information Technology, Lovely Professional University, Jalandhar, Punjab, India

Accepted 01 March 2014, Available online 01 April 2014, **Vol.4, No.2 (April 2014)**

Abstract

Effort estimation the most important term used in software engineering and has a great value in software industry. As in software industries, Cost of the project and the time to complete the project is predicted by estimating efforts. There are many traditional methods for estimating efforts along with some defined factors, like COCOMO, Use-Case point, Function Point and many more. Cocomo is considered as the most suitable and reliable cost estimation model. However, at early phases; project effort estimation is difficult and more challenging to the project manager. If estimates are made ate the early stages, it will help to keep track rest of the project in terms of time and cost. In this paper, we discuss COCOMO model along with cost drivers and Use case point With Technical and Environmental factors. In this paper, there will be enhancement in the cost drivers by adding some technical and environmental factors.

Keywords: *Cocomo, efforts, use case points.*

1. Introduction

Effort estimation is a part of system analysis. System analysis is a process or procedure that performs operations to accomplish the business goals. System analysis means to analyze the complex problems in large scale industries or other diverse fields like education, defense, law, medical and many others (Inderpreet kaur, Manpreet kaur,2013). Several estimation models are there to predict efforts with some factors that affect the efforts. Most effort estimation models rely on empirical derivation, using regression analysis of a collection of historical project data (Bingchiang Jeng, Dowming Yeh, Deron Wang, Shu-Lan Chu And Chia-Mei Chen et al,2011)Cocomo is constructive cost model that affects with 15 cost drivers and in cocomo 2, there are 17 cost drivers. These cost drivers provides a ripple effect in the cost. The underlying assumption for COCOMO (like FPA) is that a statistically significant historical database exists to drive the statistical factoring**(** Chetan Nagar).Similarly, factors involved in use case points are Environmental factors and technical factors that causes a very small change in efforts. As Estimating efforts means predicting efforts and predictions are not always correct. A single estimation model not always gives correct estimation. By merging models somehow gives better result than one estimation model. COCOMO model takes kloc as the input and generate single output. The use case model is used to predict the of future software system at an early development stage that helps to provide full control over the project progress. To develop an estimation model we have to follow these steps:

1. Determine a list of potential / most important effort cost drivers.
2. Determine a scaling model for each effort and cost driver.
3. Select initial estimation model.
4. Measure and estimate projects and compare.
5. Evaluate quality of estimation as part of project post-mortem.
6. Update and validate model at appropriate intervals.(FarshadFaghih)

2. Cocomo Model

COCOMO is the traditional cost estimation model in which size is the major input parameter. . COCOMO is mathematical equation that can be fit to measurements of effort for different-sized completed projects, providing estimates for future projects (A. Chamundeswar, Chitra Babu,2010).It was first developed by Bary Boehm for effort estimation. . It is independent model which is well documented and cannot be depended upon any software vendor. Its later version is cocomo 2. COCOMO 2 model preserve the originality of COCOMO model i.e. openness of the COCOMO. The main difference between cocomo 1 and cocomo 2 is that in cocomo 1 there are 15 cost drivers. In cocomo 2, there are 17 cost drivers along with 5 scaling factors.

2.1 Assumptions

1) The cost driver is assumed to be the total number of lines delivered during project development'

*Corresponding author: **Inderpreet Kaur Walia**

2)The cocomo model is applied from designed phase to test phase. For rest of the phases, we have to use other model
3) A COCOMO Staff-Month (SM) consists of 152 hours of working time(Farshad Faghih)
4)COCOMO assumes that the cost drivers are phase dependent
5) Project can be managed by both developer and customer.

Table 1: difference between cocomo1 and cocomo 2

COCOMO1	COCOMO2
1. It is basic model.	1. It is extension model of basic model.
2. It follows waterfall model.	2. It follows three phases concept.
3.There are 15 cost drivers are present in COCOMO1.	3. There are 17 drivers present in this model.
4. There are data 63 data points in it.	4. There are 61data points in it.
5. In this model reengineering concept is Followed.	5. Software reusability is used in it
6. It is measured in KDSI	6. It is measured in KLOC

Table 2: Modes of cocomo model

Project type	a	b
organic	3.2	1.05
semidetached	3.0	1.12
embeded	2.8	1.20

These value depend upon the kloc.
If kloc<50, the project type is organic and we consider corresponding values of a and b.
If 50<kloc<300, the project type is semi detached
If kloc>300,then project type is embedded.(A. Chamundeswar , Chitra Babu,2010)

Efforts= EAF*a(kloc)^b
Where EAF= multiplication of 15 cost drivers.

3. Use-Case Point

It is another popular and efficient method of efforts estimation. Here we will calculate use case and actors **(**Chetan Nagar and Anuragh Dixit,2012**).** In use case point, estimation can be made in four parts.
a) Calculate estimated number of actors
b) Calculate estimated number of use cases
c) Calculation of technical factors
d) Calculation of environmental factors

3.1 Estimations are made using following steps

Step1: Calculate the weight of actors (UAW): Actors weight is assigned according to the three categories of actors. Simple, average and complex. A simple actor has a defined API,average actor is a person or a system that interacts with other system through TCP/IP and complex actor interacts through GUI interfaces**(** Shinji Kusumotoy)
Step 2: calculate the weight of use cases(UUCW). The weights are assigned to the use cases according to the transactions they process. A simple use case made 3 or fewer transaction; average use case process 4 to7 transactions and the complex use case process more than 7 transactions.
Step 3. Calculate technical factors (TCF)

TCF=(0.1*TC factors)+0.6

T1	Distributed system	2
T2	Response or throughput performance objectives	1
T3	End-user efficiency (online)	1
T4	Complex internal processing	1
T5	Code must be reusable	1
T6	Easy to install	0.5
T7	Easy to use	0.5
T8	Portable	2
T9	Easy to change	1
T10	Concurrent	1
T11	Includes special security features	1
T12	Provides direct access for third parties	1
T13	Special user training facilities are required	1

Fig 1: Technical Factors

Step 4: calculate environmental factors(EF)-EF is calculated by multilpying each factor by its corresponding weight and add all the values that we get after multiplication
EF=1.4+(-0.03*Efactor)

Factor	Description	Weight
F1	Familiar with RUP	1.5
F2	Application experience	0.5
F3	Object-oriented experience	1
F4	Lead analyst capability	0.5
F5	Motivation	1
F6	Stable requirements	2
F7	Part-time workers	-1
F8	Difficult programming language	2

Fig 2: Enviornmental factor

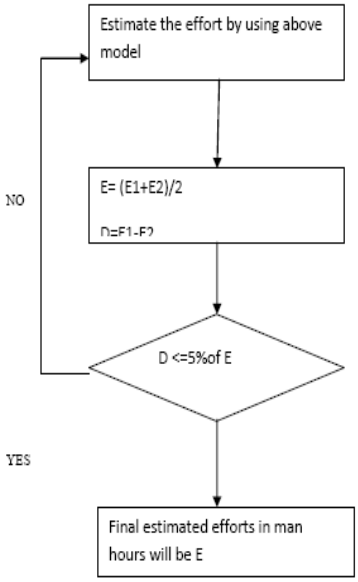
Step 5= calculate unadjusted use case point(UUCP)= This is achieved by adding unadjusted actor weight and unadjusted use case weight
UUCP=UAW+UUCW
Step 6- The final step is to calculate Use case point(UCP)
UCP=UUCP*TCF*EF

4. Combining use Case and Cocomo

As we have already discuss that by combining two or more estimation model will give better result than single model. So, we merge cocomo and use case point. We are taking some technical factors and environmental factors and combine them into the cost drivers to get better estimation result.
Evaluation process:

- First calculate Efforts using cocomo,E1
- Calculate efforts using use case point,E2
- Calculate E, (E1+E2)/2
- Then calculate difference,D= E1-E2

If the difference calculated is less than 5 % of Efforts calculated, and then process will again start otherwise final efforts are calculated in person month



Conclusion

This paper concludes that by combining two estimation models will produce better result than cocomo and use case alone. In this paper, basic concepts of cocomo are discussed that is already known. It just gives an how we can combine cocomo and use case point to get the better results.

References

Inderpreet kaur walia,Manpreet kaur(2013),To Enhance The Effort Estimation Accuracy Of Cocomo Model Using Function Point, *International journal of engineering and sciences & research technology*

Bingchiang Jeng, Dowming Yeh, Deron Wang, Shu-Lan Chu And Chia-Mei Chen et al(2011), A Specific Effort Estimation Method Using Function Point, *Journal Of Information Science And Engineering.*

Chetan Nagar, Software efforts estimation using Use Case Point approach by increasing Technical Complexity and Experience Factors, *International Journal on Computer Science and Engineering*, Vol. 3. Pp-3377-3345

Farshad Faghih,Software Effort and Schedule Estimation

A. Chamundeswar , Chitra Babu(2010),An Extended Function Point Approach for Size Estimation of Object-Oriented Software.

Chetan Nagar and Anuragh Dixit(2012), Efforts Estimation by Combining the Use Case Point and COCOMO, *International Journal of Computer Applications* (0975 – 8887) Volume 52 – No. 7, Chittodgarh Rajasthan India, pp. 1-5

Shinji Kusumotoy, Fumikazu Matukaway, Katsuro Inouey,Shigeo Hanabusaz, Yuusuke Maegawaz,Effort Estimation Tool Based on Use Case Points Method, *Graduate School Of Information Science And Technology*, Osaka University,1-3, Machikaneyama, Toyonaka, Osaka 560-8531, Japan