ISSN: 2277-9655
Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

To Enhance the Effort Estimation Accuracy of Cocomo Model Using Function Point

Inderpreet Kaur Walia[*1], Manpreet Kaur[2]
[*1,2] Lovely Professional University, Jalandhar, Punjab, India
Inder2walia@gmail.com

Abstract

Effort estimation means to estimate the efforts according to the expectations of stakeholders before implementation of the project. There are many top-down and bottom-up approaches that are recommended by stakeholders for estimating the efforts .There are many other models that accurately estimates the efforts. However, it is suggested that the models should be used in pair to estimate the efforts accurately. This paper combines function point and cocomo model. Cocomo is the constructive cost model that is based on the lines of code and is considered as the best model among all as it produces least efforts. Function point is the another most accepted technique for effort estimation and is based on the functional size. This paper propose, the combination of function point and cocomo will produce less efforts than cocomo or any other model

Keywords: Effort estimation, Esimation models,Cocomo model,Function point

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 analyse the complex problems in large scale industries or other diverse fields like education, defence, law, medical and many others[1]. It is an activity performed by the system analyst by applying various mathematical techniques to achieve the objectives. The activities performed by system analyst or other decision makers are like searching the problem,analyse the problem, gathering information, formulate the problem, apply techniques to solve the problem and finding solution to solve the problem or to enhance the system. System analysis decomposes the system into subsystem in order to diagnose the problem more efficiently.System analysis is important in large scale industries to make the effective decisions regarding budget where cost and time are the major constraints. System analysis is required to identify what kind of and how much data is to be gathered. It is also a process of analyzing what should be the input and what will be the output. System analysis is also required to restrict the changes made by one system to other. Effort estimation is done to estimate the cost of development project. Most effort estimation models rely on empirical derivation, using regression analysisof a collection of historical project data.[2] The most important principle behind cost and time estimation is to estimate the project size.From project size,we mean Kloc or it can be functional based.

Estimation Models

There are many such models that helps to calculate the effort estimation mathematically. The models are Walston-Felix ,Doty model,Baily-Basily model,Halstead metrics,Cocomo.These models are based on the size of project(Kloc).Some other existing techniques are Function point, use case point, Neural network techniques,Soft computing technique and many more.

Literature Survey

Iman Attarzadeh, Amin Mehranzadeh, Ali Barati (2012) describes[2],In software development, the project manager has to face the problems regarding cost, time and staff estimation. This is one of the critical task in software development process. This paper provides better view oh hybrid model ANN-COCOMO i.e Cocomo model using artificial neural network for effective effort estimation. Software estimation is classified as algorithmic and non algorithmic technique.Cocomo is considered as the best model that follows algorithmic techniques such as regression technique that is based on historical data.ANN is a mathematical technique to calculate the working condition of human brain.ANN basically makes fine adjustments of attributes using historical data. Due to change in the business environment, the relationship among attributes become vague. To overcome this problem, this paper proposed ANN-COCOMO model.

Ömer Faruk SARAÇ, Nevcihan DURU (2103) narrate[3], software effort estimation is a challenging task in software project management. There are number of techniques or models that are already implemented for software effort estimation. Some models are used in combination to achieve better accuracy in software effort estimation. One such model is cocomo which is highly recommended model for effort estimation. It provides better results or more accuracy in effort estimation when combined with neural networks. In this paper, a novel method, combining COCOMO used ANN with K-Means is used to estimate effort and possible boundaries[3]. K means is an algorithm that decompose the sets into subsets and perform functions to form a complete set. This set is formed from the ANN output generated and within this set upper and lower boundaries are evaluated for effort estimation. In this paper it is concluded that the proposed model gives better accuracy than cocomo and ANN alone

Iman Attarzadeh, Siew Hock (2010) represent [5]Software effort estimation is an important part of software development activities. As software development process become complex day by day as it is affected by number of factor like size,complexity,cost and time. To control these factors, an effective cost estimation model is required. Cocomo is considered as the best suited model for effort estimation. This paper focus on improving the reliability of software cost estimation by proposing a model cocom2 with some sensible features of neural network such as interpretability and learning ability[4].In this paper back propagation technique of neural network is applied on two different data sets i.e cocomo dataset and artificial dataset to estimate software efforts. The estimated efforts are evaluated by taking MRE of different-2 projects and MMRE is achieved from all the calculated MRE's. The proposed neural network provides low MRE than cocomo dataset i.e it provides better effort estimation.

Chetan Nagar, Anurag Dixit (2012) in this paper[5]they discuss, software effort estimation is an important task in software industry as the failure and success of the software industry relies on reliability of software effort estimation. There are many models for estimating software efforts. But there is not one perfect model that is globally acceptable. This paper proposes a hybrid model that combines cocomo and use case point. Use case point is another most accepted technique for effort estimation. In this paper generalized form of use case point is taken rather than specialized form. Also,use case point possesses technical requirements of the system. Cocomo is based on the Kloc of the projects to estimate the efforts. Cocomo consists of cost drivers that is actually needed for effort estimation. It This paper concluded that to estimate kloc we have to divide the project into module and module into the sub module until we are able to estimate the KLOC[5]. In the end checklist is prepared that includes name of use case, date of completion, actual date of completion and action taken if not completed on time.

COCOMO Model

Cocomo is constructive cost model developed by Boehm for the estimation of total cost needed for the development of the project.Cocomo is used for effort estimation for different sized projects. COCOMO is amathematical equation that can be fit to measurements of effort for different-sized completed projects, providing estimates for future projects[6]

Basically,it depends on kloc of the project.Cocomo81 is originated model which is further decomposed into 3 sub-parts:Basic,Intermediate and Detailed.The basic cocomo formula for effort estimation is:

Effort=a*(kloc)^b

Values of a and b depends upon the type of projects and tge project type is depend on the size of project.

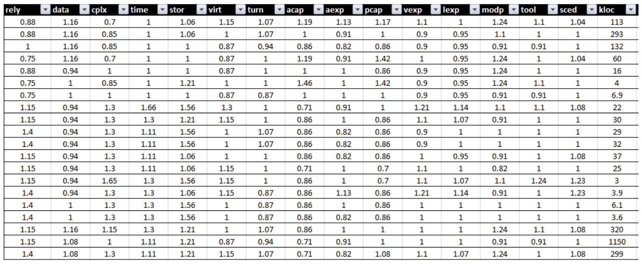
If $0<size<50$, then project type is organic
$50<size<300$, then project type is semidetatched
$Size>300$, then project type is embeded

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

In our paper,we are using Intermediate cocomo model in which extra feature is added i.e 15 cost drivers. These 15 cost drivers have fixed values which are multiplied to get effort adjustment factor(EAF). These 15 cost drivers are classified into four types of attributes: product attributes ,project attribute, personnel attributes and hardware attributs.The effort estimation for intermediate cocomo model

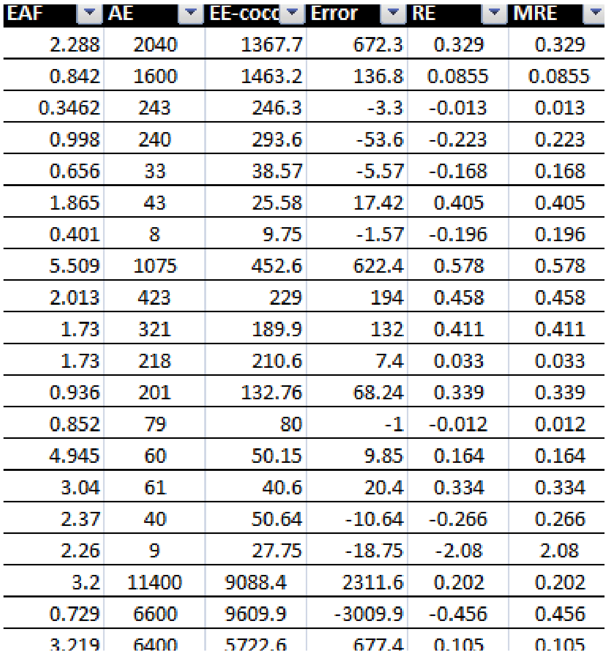
Effort=a*(kloc)^b*EAF

Cocomo2 is the current version in which there are 17 cost drivers and 5 scaling factors.The dataset used is NASA93[7] in which there are 63 projects which contains value for 15 cost drivers,kloc and actual effort.



rely	data	cplx	time	stor	virt	turn	acap	aexp	pcap	vexp	lexp	modp	tool	sced	kloc
0.88	1.16	0.7	1	1.06	1.15	1.07	1.19	1.13	1.17	1.1	1	1.24	1.1	1.04	113
0.88	1.16	0.85	1	1.06	1	1.07	1	0.91	1	0.9	0.95	1.1	1	1	293
1	1.16	0.85	1	1	0.87	0.94	0.86	0.82	0.86	0.9	0.95	0.91	0.91	1	132
0.75	1.16	0.7	1	1	0.87	1	1.19	0.91	1.42	1	0.95	1.24	1	1.04	60
0.88	0.94	1	1	1	0.87	1	1	1	0.86	0.9	0.95	1.24	1	1	16
0.75	1	0.85	1	1.21	1	1	1.46	1	1.42	0.9	0.95	1.24	1.1	1	4
0.75	1	1	1	1	0.87	0.87	1	1	1	0.9	0.95	0.91	0.91	1	6.9
1.15	0.94	1.3	1.66	1.56	1.3	1	0.71	0.91	1	1.21	1.14	1.1	1.1	1.08	22
1.15	0.94	1.3	1.3	1.21	1.15	1	0.86	1	0.86	1.1	1.07	0.91	1	1	30
1.4	0.94	1.3	1.11	1.56	1	1.07	0.86	0.82	0.86	0.9	1	1	1	1	29
1.4	0.94	1.3	1.11	1.56	1	1.07	0.86	0.82	0.86	0.9	1	1	1	1	32
1.15	0.94	1.3	1.11	1.06	1	1	0.86	0.82	0.86	1	0.95	0.91	1	1.08	37
1.15	0.94	1.3	1.11	1.06	1.15	1	0.71	1	0.7	1.1	1	0.82	1	1	25
1.15	0.94	1.65	1.3	1.56	1.15	1	0.86	1	0.7	1.1	1.07	1.1	1.24	1.23	3
1.4	0.94	1.3	1.3	1.06	1.15	0.87	0.86	1.13	0.86	1.21	1.14	0.91	1	1.23	3.9
1.4	1	1.3	1.3	1.56	1	0.87	0.86	1	0.86	1	1	1	1	1	6.1
1.4	1	1.3	1.3	1.56	1	0.87	0.86	0.82	0.86	1	1	1	1	1	3.6
1.15	1.16	1.15	1.3	1.21	1	1.07	0.86	1	1	1	1	1.24	1.1	1.08	320
1.15	1.08	1	1.11	1.21	0.87	0.94	0.71	0.91	1	1	1	0.91	0.91	1	1150
1.4	1.08	1.3	1.11	1.21	1.15	1.07	0.71	0.82	1.08	1.1	1.07	1.24	1	1.08	299

Fig:cost drivers and their values in NASA93



EAF	AE	EE-cocc	Error	RE	MRE
2.288	2040	1367.7	672.3	0.329	0.329
0.842	1600	1463.2	136.8	0.0855	0.0855
0.3462	243	246.3	-3.3	-0.013	0.013
0.998	240	293.6	-53.6	-0.223	0.223
0.656	33	38.57	-5.57	-0.168	0.168
1.865	43	25.58	17.42	0.405	0.405
0.401	8	9.75	-1.57	-0.196	0.196
5.509	1075	452.6	622.4	0.578	0.578
2.013	423	229	194	0.458	0.458
1.73	321	189.9	132	0.411	0.411
1.73	218	210.6	7.4	0.033	0.033
0.936	201	132.76	68.24	0.339	0.339
0.852	79	80	-1	-0.012	0.012
4.945	60	50.15	9.85	0.164	0.164
3.04	61	40.6	20.4	0.334	0.334
2.37	40	50.64	-10.64	-0.266	0.266
2.26	9	27.75	-18.75	-2.08	2.08
3.2	11400	9088.4	2311.6	0.202	0.202
0.729	6600	9609.9	-3009.9	-0.456	0.456
3.219	6400	5722.6	677.4	0.105	0.105

Fig: calculated MRE

Evaluation process

EAF= Multiplication of 15 cost drivers

Estimated Efforts(EE)= a*(kloc)^b*EAF

Error=Actual efforts-estimated efforts

Relative errors= (Actual efforts-estimated efforts)/ Actual efforts

MRE= abs(RE)

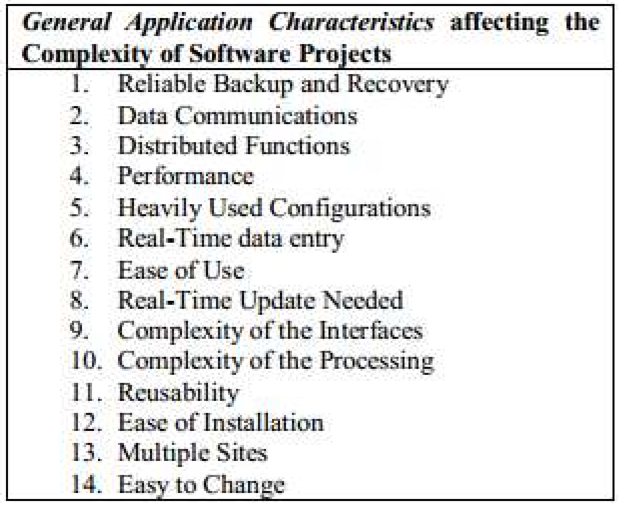
Calculated MMRE of 20 projects for cocomo

MMRE=0.685

Function Point

The main objective of FPA is to determine the size based on functional requirements of the software application.[8]. Function point analysis is a size estimation technique purely based upon the functions and are not depend upon the technology used to develop the project.Function Point counting involves classifying software items into transactions and dataEntities[9]. pureThese entities are classified into 5 classes, external inputs(EI),external outputs(EO),External queries(EQ), Internal logical files(ILF),external interface files(EIF).There are toatal 15 software items that have some weighted counts ranging as low,medium and high. These weights are summed up to get unadjusted function points.According to IFPUG Unadjusted function points are further classified into unadjusted data function points that includes EI and EO and unadjusted transaction function points that contains ILF, EIF and EQ. Function point depends upon the number of test cases involved.Value adjustment factor is another important factor for function point analysis which is calculated by taking the sum of 14 general system characterstic whose values ranges from 0 to 5

Table:General System Chracterstics[10]



General Application Characteristics affecting the Complexity of Software Projects
1. Reliable Backup and Recovery
2. Data Communications
3. Distributed Functions
4. Performance
5. Heavily Used Configurations
6. Real-Time data entry
7. Ease of Use
8. Real-Time Update Needed
9. Complexity of the Interfaces
10. Complexity of the Processing
11. Reusability
12. Ease of Installation
13. Multiple Sites
14. Easy to Change

FPA=VAF*UFP

FPA=function point analysis

VAF =value adjustment factor

UFP=unadjusted fuction point

$$VAF=0.65+0.01\sum_{i=1}^{14} Ci$$

UFP=EI+EO+EQ+ILF+EIF

Conclusion

Estimating efforts accurately decides the software failure or success. Various models and techniques are applied to estimate the efforts accurately. Among all the models cocomo provides the estimated efforts close to actual efforts.in this paper,we are proposing a hybrid model of cocomo and function point that produces estimated efforts less than cocomo and function point alone.

ISSN: 2277-9655
Impact Factor: 1.852

References

[1] Edward A. Beimborn(2003) "*Systems Analysis – What, Why And How*" College Of Engineering And Applied Science University Of Wisconsin—Milwaukee

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

[3] Ömer Faruk SARAÇ, Nevcihan DURU et al,2013 "A Novel Method For Software Effort Estimation: Estimating With Boundaries"

[4] Iman Attarzadeh, Siew Hock Ow et al,2010 "Proposing a New Software Cost Estimation Model Based on Artificial Neural Networks" Proc. Of *Institute of electrical and electronics engineers*

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

[6] A. Chamundeswar , Chitra Babu et al,2010 "*Estimating Size and Effort Dr. James A. Bednar*

[7] https://www.promise.site.uottawa.ca/SERepository/datasets/cocomo81.arff

[8] Dr. James A.Bednar et al,2011"*An Extended Function Point Approach for Size Estimation of Object-Oriented Software*"

[9] Erik Stensrud et al,2010 "*Estimating with Enhanced Object Points vs. Functio Points*"

*[10]*A Review S.K. Mohanty1& A.K. Biso "*Software Effort Estimation Approaches*"