ISSN No. 0976-5697

Volume 8, No. 5, May – June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Performance Analysis of Development Time and Effort in COCOMO using MTTR

Anshita Singh
Department of IT,
BabasahebBhimraoAmbedkar University,
Lucknow-UP, India

Dr. Raj Shree
Department of IT,
BabasahebBhimraoAmbedkar University,
Lucknow-UP, India

***Abstract*:** In this paper, we improve the software reliability by using COCOMO. As we know COCOMO (constructive cost model) is one of the widely used software cost estimation model. It is investigated as popular model for software cost estimation which depends on several variables like loc (line of code), effort estimation, development time or duration estimation, persons required. In this paper researcher investigates the rate of development time to improve the accuracy of cost estimation and reliability. By this reliability can also improve because at the development time we are going to include MTTR (mean time to repair). By adding MTTR value failure removes from the software and software become reliable. As we know software cost estimation is the process of predicting effort required to develop a software system. Some other models used to estimate cost are SLIM, COCOMO, FP (Function point) [3]. Software time estimation is the process of estimating the time required to develop the software. It supports the planning and tracking of software projects. The formula proposed in this paper calculates the accuracy of the software reliability. As the reliability improves it reduces the chance of failure.

Keywords: COCOMO, software cost estimation, KLOC, MTTR, Duration estimation.

I. INTRODUCTION

Software cost estimation is the process for improving the software reliability. Reliability have many attributes[9], which the software cost estimation we focus on effort,development time and people required to complete the software[14].Two major types of cost estimation method are algorithmic models and non-algorithmic models.Some popular estimation models includes boehm's, cocomo, putnam – svelte[2]. In this paper we are using only cocomo, cocomo is developed in 1981 by "barry boehm", it gives the estimation of man-month that will take to develop a software product.

In COCOMO there are three different models:

1. Basic Model
2. Intermediate Model
3. Detailed Model

There are three classes of software project:

- Organic Mode
- Semi- Detached Project
- Embedded Project

ORGANIC MODE: In Organic Mode, small, easy software project with small member of team are handle.

SEMI-DETACHED: In Semi- Detached Project an intermediate projects in which teams with mixed experience level are handle.

EMBEDDED PROJECT: In Embedded Project tight hardware, software and operational constraints are handled.

Software Reliability metrics have by and large, evolved from hardware reliability metrics. However, hardware metrics cannot be used without modification because of the differing nature of software and hardware failure.

There are some metrics which have been used to improve software reliability are [5,6]:

1. Probability of failure on demand
2. Rate of failure occurrence (ROCOF)
3. Mean time to failure (MTTF)
4. Availability
5. Mean time to repair (MTTR)

In this paper we are going to use MTTR.
MTTR – mean time to repair the failure

MTTR = Total Outage time / Number of failure

II. SOFTWARE RELIABILITY USING COCOMO

A. *Software Cost Estimation*

Software cost estimation is a technique to estimate no. of time, cost, effort, number of person required in software development phase [7]. By the cost estimation at an earlier stage we can predict the amount of effort, time and staff [10]. There are several methods to improve software reliability and software cost estimation [11].

B. *COCOMO (Constructive Cost Model)*

COCOMO models are technique that is designed to estimate development time by collecting data from vast application objects. This detailed we considered to observed equation that is closed to actual development time. All of the estimated effort is expressed as person months (PM). Barry Boehm introduced a software cost estimation model in 1981 which is called as the constructive cost model (COCOMO) [12]. It forms by the study of 63 project at TRW aerospace, where Barry Boehm was Director of software research and Technology in 1981[4].

PARTS OF COCOMO

- BASIC MODEL
- INTERMEDIATE MODEL

- DETAILED MODEL

BASIC MODEL: Basic COCOMO is best for small, early, simpler.
Limitation: Its result is not so smooth, it is not so accurate.
INTERMEDIATE MODEL [1]: It is advance version of basic COCOMO. This "Model makes use of set of cost drivers attributes" to compute the cost of software
Limitation: This model is a product with many components is difficult to estimate.
Detailed COCOMO: It is same as the intermediate Model, but detailed model can estimate the effort (E), duration estimation (D) ,person required (P) of each development phases, subsystem modules.
There are three classes in COCOMO:

- ORGANIC MODE
- SEMI-DETACHED PROJECT
- EMBEDDED PROJECT

ORGANIC MODE: In this small and simple software project handle with a less member of team and they have good experience.
SEMI-DETACHED PROJEC: In this intermediate project are handled with the mixed experience team member are present. EMBEDDED PROJECT: In this class, projects with tight hardware, software and operation constraints are handling.

[15]COCOMO (BASIC MODEL)
$E = a_b (KLOC)^{b_b}$
$D = c_b(E)^{d_b}$
$P = E/D$
Where E is effort applied in person-months.
KLOC is kilo line of code for the project.
D is Development time of the software.
P is total number of person required to accomplish the project.
a_b , b_b , c_b , d_b are the coefficients.

C. Reliability Testing

Software reliability testing is one of the major phases to form software reliability [16]. It tests the software ability to function in given environment conditional consistently that helps to resolve the problem of the reliability.

Some parameter involved in reliability testing is as follows [13]:

- Probability of failure free operation.
- Length of time of failure free operation.
- The environment in which it is executed.

Key parameter [6, 8]:

- MTTF : MEAN TIME TO FAILURE
- MTTR: MEAN TIME TO REPAIR
- MTBF: MEAN TIME BETWEEN FAILURE

MTTF = total operating time / no. of failure
MTTR = total outage time / no. of failure
MTBF = MTTF + MTTR

III. PROPOSED METHODOLOGY

COCOMO is most widely used software estimation models in the world. This model is developed by Barry Boehm in 1981. It shows the estimate of the no. of man-months that will take to develop a software product.
$E = a_b(KLOC)^{b_b}$
$D = c_b(E)^{d_b}$
$P = E/D$

TABLE 1: The coefficients a_b, b_b, c_b, d_b for three modes

Software project	a_b	b_b	c_b	d_b
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Example 1
Consider a software project using semi- dethatched mode with 30,000 lines of code. We will obtain estimation for this project as follows [15] –

i) Effort estimation
$E = a_b(KLOC)^{b_b}$
$E = 3.0\ (30)^{1.12}$
Where line of code = 30000 = 30 KLOC
E = 135 Person-month

ii) Duration estimation
$D = c_b(E)^{d_b}$
$D = 2.5\ (135)^{0.35}$
D = 14 months

iii) Person estimation

P = E/D
P = 135/ 14 = 9.64
P = 10 Person approx.

Now, in this paper researcher makes a relation between development time and mean time to repair. In COCOMO development time of software is failure, but in this mean time to repair a failure is not mention (when failure occurs). But in purposed method we are including mean time to repair software.
$D = c_b (E)^{d_b}$ from the COCOMO
NOW, from the purposed method
When the failure occur then the development time increase.
$D_{MTTR} = D + MTTR = c_b (E)^{d_b} + MTTR$
D from the COCOMO
MTTR = total outage time / number of failure
(From the reliability testing)
In COCOMO effort is $E = a_b(KLCO)^{b_b}$
When we repair the failure there may be line of code increases then the Effort will also change.
So, E (when the failure repair) = $a_b(KLOC)^{b_b} + a_b(KLOC)^{b_b}$
Example 2
Consider a software project using semi- dethatched mode with 30,000 lines of code. We will obtain estimation with

the repairing time of failure for this project as follows (using coefficient value from table 1) –

E = 135 without any failure from the example 1

Now, we include repairing of failure then KLOC may be increase. While repairing a failure 2000 line of code increase let,

So, putting in purposed formula

E (when the failure repair) = $a_b(KLOC)^{b}_{b} + a_b(KLOC)^{b}_{b}$

$= 135 + 3.0\ (2)^{1.12}$

E (when the failure repair) = 135 + 7 = 142

Development time estimation including mean time to repair the failure.

D (when the failure repair) = D + MTTR

(D = 14 months) from example 1

MTTR = total outage time / number of failure

Let failure repairing time be;

TABLE2

S. No.	Failure Occurrence	Repairing time (In Days)
1	1st failure	6
2	2nd failure	4
3	3rd failure	8

MTTR = 6+4+8 /3 = 6 Days = 1/5 months

D (when the failure repair) = D + MTTR

=14 + 1/5 = 70 +1 /5

D (when the failure repair) = 14.2

Now,

Persons estimation = E (when the failure repair) / D (when the failure repair)

= 142 /14.2

Persons estimation =10 persons (same no persons as in example 1)

By including failure Repair time, we don't need extra person as we have proved in above example.

So, by this we can improve the reliability of the system by using the relation of COCOMO and MTTR.

IV. EXPERIMENTAL RESULT

Basic COCOMO (without failure)

Case 1: LOC = 30,000 = 30KLOC

Case 2: LOC = 50,000 = 50KLOC

We have already applied in purposed method:

TABLE 3

Case	Effort	Duration Time <D> (months)	Persons
1	135	14	10
2	240	17	14

For Case 1: When MTTR is added then

TABLE 4.

Effort	Duration time (months)	Person
142	14.2	10

After repairing the failure.

MTTR = Total outage time / no. of failure

TABLE 5

S. No.	Failure	Repairing time (In Days)
1	1st failure	8
2	2nd failure	10

Now, MTTR = 8+10/ 2 = 18/2

= 9 days

MTTR = 0.3 months

Putting value in purposed formula –

$D_{(MTTR)}$ = D + MTTR

(D = 17 from table 3)

$D_{(MTTR)}$ = 17 + 0.3

= 17.3

Now, we include repairing of failure then KLOC may be increase. While repairing a failure 2000 line of code increase let,

So, putting in purposed formula

E (when the failure repair) = $a_b(KLOC)^{b}_{b} + a_b(KLOC)^{b}_{b}$

$= 240 + 3.0\ (2)^{1.12}$

E (when the failure repair) = 240 + 7 = 247

Persons estimation = E (when the failure repair) / D (when the failure repairs)

= 247/ 17.3 = 14.2

Persons estimation =14 persons approx.

For Case 2: When MTTR is added then

TABLE 6.

Effort	Duration time (months)	Person
247	17.3	14

From the experiment result by adding MTTR in development time and effort required in repairing time in effort estimation, we get that cost is somehow increases by increases of effort but extra person is needed but reliability is improving overall.

V. CONCLUSION

In this paper, we have purposed the use of mean time to repair when failure in basic COCOMO in development time of software. In basic COCOMO it show the development time of software with any failure occur and in purposed work are including occurrence of failure and there repair time. COCOMO is software cost estimation model. Over all we can say reliability is improving by repairing the failure. We purposed a new formula for development time or duration estimation time with including mean time to repair the failure. By applying we see that number of person is same, but effort is increasing when we repair the failure .overall reliability is improving, cost is increasing because duration time, effort estimation of the software is increasing. At initial we taking cost of effort, duration time, person required in the software without any failure occur.

But now, in purposed formula we are repairing the failure if it occurs that why cost is increasing and reliability of software is also increasing, as we know cost is directly proportional to reliability.

VI. REFERENCES

1. Neetu Kushwahal, Suryakane," Software Cost Estimation using the Improved Fuzzy Logic Framework," 978-1-4799-3064-7/14/$31. 00©20 14 IEEE.
2. Ma Junha, MU Lingling, "Comparison Study on methods of software cost estimation ,"Hebei Provincial Construction of Science and Technology Research Program No.2009-150
3. CHRIS F. KEMERER, "An Empirical Validation of Software Cost Estimation Models," 01987 ACM OOOl-0782/87/0500-0416 750
4. Riyanarto Sarno, Johannes Sidabutar, and Sarworsi, "Improving the accuracy of COCOMO's Effort Estimation Based on Neural Networks and Fuzzy Logic Model," 2015 International Conference on Information, Communication Technology and System (ICTS)
5. Susan Stanley, "MTBF,MTTR,MTTF & FIT" Explanation of Terms Senior Technical Support Engineer, IMC Networks
6. Bruce Benton, "Model-Based Time and Cost Estimation in a Software**"** 2009 Sixth International Conference on Information Technology: New Generations
7. Iman Attarzadeh, Siew Hock Ow, "A Novel Soft Computing Model to Increase the Accuracy Of Software Development Cost Estimation," 978- 1-4244-5586-7/10/$26.00 C 2010 IEEE
8. Harry G. Kwatny," ENGINEERING RELIABILITY FAILURE MODELS," retrieved from http://www.pages.drexel.edu/~hgk22/courses/MEM361/MEM361-Lecture3.pdf
9. Software Reliability definition from Wikipedia, retrieved from https://en.wikipedia.org/wiki/COCOMO
10. Taqi Shah, "Title-Software Cost and Effort Estimation in Software Engineering Process".
11. Jiantao pan, "Software Reliability" 18-849b Dependable Embedded Systems Spring 1999.
12. Barry W.Boehm, "COCOMO Definition" from Wikipedia, retrieved from https://en.wikipedia.org/wiki/COCOMO
13. "Introduction to Software Reliability Estimation", © Lionel Briand 2010, retrieved from http://www.uio.no/studier/emner/matnat/ifi/INF4290/v10/undervisningsmateriale/INF4290-SRE.pdf.
14. Ilkka Karanta, "Methods and problems of software reliability estimation" ESPOO 2006.
15. Mrs Anuradha A. Puntambekar, "Software Engineering" Technical publication, ISBN 9789350380727, January 2011.
16. Reliability testing definition from tutorial point, retrieved from http://www.tutorialspoint.com/software_testing_dictionary/reliability_testing.htm