ECE 3311 Software engineering 1, Review Of Paper On Cost Estimation in Software, Northeastern Univ. Boston, Massachusetts

Nasser M. Abbasi

11/14/1993 Compiled on November 16, 2018 at 11:28am [public]

1 Introduction

Before I start in the review of the paper, I give an overview on the subject itself, this way the paper review will have the necessary background laid out.

The paper in about the subject of Software Cost Estimating, in software cost estimating 7 different methods or techniques are used to carry the cost estimations, these are: ○ *Algorithmic cost modeling* ○ *Expert judgment* ○ *Estimation by analogy* ○ *Parkinson's law* ○ *Pricing to win* ○ *Top down estimation* ○ *Bottom up estimation*

Within Algorithmic cost modeling, a popular method exist which s called the COCOMO model. This is a costing model whose parameters can be adjusted to the particular modes of working. This model will generate cost estimate via the use of a formula, this formula comes in 3 forms, the basic (simple) form, the intermediate form, and the advanced or detailed form.

The paper I have chosen to study and evaluate deals with the COCOMO model. I'll first explain and overview the paper itself , then I'll critique the paper. The way I will review this paper is by first giving an overall view of the structure of the paper, then a more detailed view is taken. First the paper information is given.

2 Paper Information

Paper title: Cost Estimation for the reuse and prototype software development life-cycles.

Authors: David Balda and David Gusafson.

Journal: ACM Software Engineering notes, ACM-SIGSOFT, Volume 15, Number 3.

Date published : July 1990 .

3 First level review

The paper introduces two new cost models (formulas) that are derived from the basic COCOMO formula, the justification for the need for these new formulas is based on the fact that the original COCOMO formula was targeted towards the waterfall model of software development. The new cost models derived in this papers are for the reuse and prototype software development life-cycle models.

The main reason the authors give for why the original COCOMO cost model is not suitable for use in the reuse and prototype life-cycle, is because the COCOMO model did not account for the cost involved when the requirements are known to be unstable or incomplete or when extensive code and design information are reused in the future, since these 2 issues will not arise in the waterfall life-cycle as much as they would arise in the prototype and the reuse life-cycles, respectively. So, this was the basic reason given for the need to extend and modify the original COCOMO cost model. The paper will only derived new cost models from the original basic COCOMO cost model (formula), and not from the intermitted or detailed formulas, even through the authors do discuss the intermitted and detailed original COCOMO cost models, but not in great details as the focus of the paper was the basic (simple) COCOMO cost model.

The paper starts by outlining why the original COCOMO cost model can not be used as is for the the reuse and prototype software life-cycle. In doing so the authors give a short outline of the waterfall, reuse, and the prototype life-cycles.

After this, the authors discuss the basic (simple) COCOMO formula, they explain its nature and parameters, then a new formula for the reuse life-cycle is hypothesized, this formula contains 3 additional terms than the original COCOMO basic formula, the new terms account for specific activities that exist in reuse life-cycle model but not in the waterfall model. After this, the authors produce a final cost model formula based on the hypothesized one, the final reuse cost estimation formula has a new parameter added and a table to look up this new parameter from.

The same strategy were used by the authors in deriving a new formula for the prototype life-cycle model, a formula is Hypothesized which contains 3 terms and 3 new parameters that are unique to this life-cycle model from both the waterfall or the reuse model. assumptions were given on the validity of this formula, but finally the authors gave 4 different cost models for the prototype life-cycle and they indicate why 4 models were given, this is becuase the data avaiable for the authors to devleop one formula were not sufficient to decide on which one to consider as the most accurate cost model.

The conclusion of the paper gives a cautious view on the use and validity of these formulas derived, and that additional data from actual software projects where these formulas are used in their cost estimations are required before the validity of these new models can be shown.

4 Second level review

The authors give these justifications for the need to modify the basic COCOMO cost model for use with the reuse life-cycle development model:

1. In reuse life-cycle there is additional effort to develop code that is targeted for future reuse. This additional effort is note accounted for the the waterfall model cost estimation basic formula.

2. Domain analysis is not accounted for in the COCOMO model, but domain analysis is used in the reuse life-cycle. Domain analysis requires significant effort, this is the analysis to be done to design the component for the largest possible future use.

3. The COCOMO formula reliability cost modifier does not account for the frequency of reuse of the component, the COCOMO detailed formula does have a reliability modifier, but that is only based on the required reliability, the authors felt that this modifier should be adjust based on the frequency of reuse of the software component being used in order to offset the cost estimation itself.

The hypothesized formula to use is

$$PM = \alpha_1 N_1^{\beta} + \alpha_2 N_2^{\beta} + \alpha_3 N_3^{\beta} + \alpha_4 N_4^{\beta}$$

where N_1 = KSDI for unique code developed, N_2=KSDI for developed code for reuse,N_3=KSDI from unchanged reused components, and N_4=KSDI from modified reused components. compare this to the original basic COCOMO formula shows the difference:

$$PM = \alpha KSDI^{\beta}$$

In both expression above, KSDI stands for estimate of thousands of delivered source instructions. while α is a complexity coefficient and β is the complexity exponent, the β parameters are the same in both formulas, and PM is the programmer months of efforts. The authors also gave 6 assumptions to help to simplify the use of the first formula above.The authors then express α_2 in terms of α_3 by assuming that the effort to develop a component for reuse is 20 times that effort to reuse a reusable component, so they write $\alpha_2 = 20\alpha_3$, next they introduce new parameter γ to represent relation between the effort to reuse code and the effort to develop unique code, this leads to $\alpha_2 = 20\gamma\alpha_1$, given all these relations between $\alpha_1, \alpha_2, \alpha_3$, and the number of times a component must be reused to realize an economic benefit, the values of γ were determined, they give 2 cases for determining γ. The final cost estimation formula for the reuse life-cycle then becomes:

$$PM = \alpha N_1^{\beta} + 20\gamma\alpha N_2^{\beta}$$

where N_1=unique KSDI, N_2=KSDI developed for reuse. a table is given on page 10 of the paper for γ, they also mention that the overall values of γ is

$$.0909 \leq \gamma \leq .1739$$

and α, β are the same for the basic COCOMO cost model.

The authors then turn to the *evolutionary prototype cost estimation model*, they start similarly by showing why the basic COCOMO model is inappropriate for prototype life-cycle, the main basic reason they give, is that in the original COCOMO model, the requirement stability cost modifier factor is based on the assumption of minor changes in the requirements, but in the prototype model, major changes in the requirements are possible until the final one is reached, major or extensive requirements changes which require significant re-analysis or redesign are simply not accounted for and requires recalculations using the basic COCOMO model, this is the reasons the authors gave for the justification for deriving a new cost model.

The authors hypothesis the formula to be:

$$PM = \alpha \ \operatorname{Pr} e^{\beta} + \alpha \ lts^{\beta} + \alpha \ Tun^{\beta}$$

where $\operatorname{Pr} e$ = LOC developed for initial prototype. lts=LOC developed during iterations and Tun=LOC developed to convert prototype to developed product. LOC is line of code. Then the authors give assumptions to simplify the use of the above formula, they gave 3 assumptions for this.

Now, the authors say that due to small data available to them to verify this formula, they give 4 different variations of the above formula , each one they postulated based on the number of observations. they caution very strongly against the use of these formulas without confirming them with other methods of cost estimation. They mentioned as a guideline that a study have shown that the prototype life-cycle requires 40% less effort and contains 45% less code than products developed using the waterfall life-cycle. they attribute this change to the validation of the requirements in stronger terms in the prototype model as opposed to that in the waterfall .

5 Conclusion of the paper

Two cost models were given both based on the basic COCOMO model, the authors outline that the reuse cost model depicts a factor for the increased effort needed to develop a component for reuse, while it also depicts a factor for the decreased effort in reusing a component. For the prototype formula, several models were given, the authors indicate that these models might not be very accurate and that additional data is required to validate them.

6 Critique of paper

First I give *specific critique* relating to sections of the paper, Then below that I give general critique on the overall paper.

1. The authors in section 3.2.1 of the paper, say that the COCOMO formula reliability cost modifier adjust the cost based only on the required reliability while the reuse cost formula should account for the required reliability modified by frequency of reused, and

that is one of the reasons for the need of coming up with new formula for reuse, yet, in the last line in the abstract they say that their new formula (model) are derived from the basic COCOMO model, not the detailed one. The authors should have explained this seemingly unrelated reference more clearly.

2. In the introduction, the authors mention that the COCOMO formula does have an equivalent formula for determining cost of using existing software, but that it oversimplifies the process of designing components to be reused, this is at last line in first page of the paper. Yet, the authors do not explain how the COCOMO model oversimplifies the process of designing components for reuse. do they mean they COCOMO model has an inadequate cost modifier? this was not clear to me.

3. In 3.2.1 section in the paper, the concept of domain analysis in reuse life-cycle was stressed to be critical in terms of cost effect , yet the authors did not give the reader a concrete definition of this term and what they mean by it. this left the reader with having to look up other material to learn about the domain analysis role in reuse life-cycle. I think the authors should have spend one or 2 lines to explain this term.

4. Section 3.2.3 in the paper were useful to give the reader an idea about assumptions to help use these cost models, the authors gave an assumption, and a reality counterpart to each point. But it was not clear to me how these assumptions will simplify the use of the formulae being developed, this is the same impression I got from 3.3.3 of the paper, where assumptions on simplifying the use of the cost model for the prototype life-cycle model were presented.

5. In section 3.2.5 of the paper where the authors give the final form of the cost formula, the N_3 term is not used in the formula, yet it is defined below it. this give some confusion as to the reason for this term inclusion below the formula and what is its rule.

6. In section 3.3.2 of the paper, the term Tun which is the lines of code developed to convert prototype to a deliverable product, seems to indicate that some part of the prototype code is used in the final product, the word they used "convert" gave me this expression, I think the authors should have made it clear in their cost model how much of the prototype code is reused in the final product, may be a cost modifier to reflect this part was needed, at any extent, the practice of using part of the prototype code in the final product is not recommended.

6.1 General critique on the overall paper

The prototype cost model was developed based on limited data sets for analysis by the authors, the authors have overcame this by providing several formulas, I think the authors should have instead spend more time on trying to improve the prototype formula by looking at more projects and then come up with one formula as they did with the reuse cost model. This would help make their cost model for prototyping more accessible for use.

The paper is very good in giving the reader an idea on how cost models can be developed for different development models other than the waterfall, this paper can be used as a guide to develop cost models based on the COCOMO basic model for the formal transformation and exploratory programming. By studying this paper I got a better idea on how cost models work, and the method used to derive them when they are all based from the original COCOMO model. It would interesting research project to develop such a cost model for the formal transformation life-cycle and compare that with the cost model of the ones already derived.

The paper showed have used more diagrams , some of the tables give in the paper could be better comprehended if in addition to them a curve s shown showing how the different parameters changed. In particular in reference to table 4.

The paper lack any supporting data to confirm the validity of the cost models developed, no projects have been cost estimated using these formula.

The paper give a good references list, about 24 were listed.

This concludes my review of this paper.