University of Sindh Journal of Information and Communication Technology (USJICT)
Volume 4, Issue 4, December 2020
ISSN-E: 2523-1235, ISSN-P: 2521-5582
Website: http://sujo.usindh.edu.pk/index.php/USJICT/

Why we need COCOMO III?

Sana Iqbal[1], Dr. Muhammad Shahab Siddiqui[2], Zazilah May[3]
[1, 2]Centre for Computing Research, Department of Computer Scienceand Software Engineering, Jinnah University for Women, Karachi, Pakistan.
[3]Electrical and Electronic Engineering Department, Universiti Teknologi PETRONAS,Malaysia
sanaiq2@gmail.com[1]; shahab.siddiqui@ieee.org[2]; zazilah@utp.edu.my[3]

Abstract: In software development phase, projects are often delayed and cross the budget. This problem occurs due to the inaccurate estimation of the project cost and effort for the project development. These estimations are dependent on the project size that may use metrics, LOC and function point analysis. Variety of techniques exist for project cost estimation, which includes algorithm-based models and expert opinion. Expert opinion is one of the ways to estimate project effort. But it doesn't support project re-estimation during its lifecycle, which helps in tracking project. Cost estimation using algorithmic model plays important role in delivering quality product. COCOMO is being used for cost estimation; but it needs to be updated based on new process models therefore another version of COCOMO should be developed. This paper draws attention towards what should be in proposed COCOMO III, to avoid project uncertainties, risks and deliver a quality product within budget and timeliness.

Keywords: COCOMO, Cost Estimation, Effort Estimation, Software Cost Drivers

I. Introduction

Cost Estimation problem often occurs in software projects development, which results in a worst project management. Software cost and effort estimation plays an important role in SDLC (Software development life cycle). It has been used in planning, monitoring and budgeting the development activities. It helps in timely and within budget delivery of a project. To solve cost and effort estimation problems variety of models were introduced. The one that is commonly used is COCOMO (Cost Constructive Model) [1,2] developed by Barry W. Boehm in 1981. Therefore, this version was referred as COCOMO I or COCOMO 81[3, 4] and later one as COCOMO II, which was introduced in 1997. COCOMO I play an important role in estimating cost and effort at an early phase while COCOMO II gives an effective calibration process. But in case of COCOMO I estimation at an early phase may leads to an estimation failure whereas COCOMO II [5, 6, 7] requires calibration to a specific context which can be difficult to adapt where there is incomplete or limited data. Therefore, there is a need to develop another improved version of COCOMO as it's been 22 years since the COCOMO II is updated and calibrated new data to software engineering. COCOMO II uses scaling factors, line of code and cost drivers to estimate the project cost but it is still lacking the accuracy. To overcome this now it's the time that COCOMO III should be developed.

This paper is organized as Section II covers literature review and related work. Section III briefly describes SWOT analysis for COCOMO III. In Section IV the need of COCOMO III is discussed. COCOMO III parameters are introduced in Section V. The paper is concluded in Section VI. Section VII is highlighting future directions.

II. Literature Review and Related Work

COCOMO is a model for estimation of cost, effort and schedule of software projects developed by Barry Boehm. It uses regression basic formula having parameters (cost drivers) derived from historical data and characteristics of current projects. This model was first developed in 1981 and it's another version named as COCOMO II was developed in 1997 along with cost drivers. These drivers are helpful in understanding various factors that can affect the cost of the project. But now these cost drivers [8, 9, 10] are not applicable as they were in the past there is a need to develop the improved version of the model according to the latest trends.

COCOMO I help in understanding the cost consequences of decisions in development and support of software project. Whereas COCOMO II offers great amount parameters along with the software cost estimation [11, 12]. COCOMO I is a chain of three sub model each of them is more detailed than other. The first sub model is known as Basic COCOMO which is responsible for measuring the lines of code (LOC) for calculating the development cost and effort of a software project. Basic COCOMO is further divided into three types of project modes that are Organic, semi-detached and embedded projects [13, 14]. Organic projects are simple and developed by good experienced small team, semi-

detached are medium sized projects developed by team with diversified experience and embedded projects are the complex projects developed by multiple teams having vast experience. The second sub model is known as Intermediate CCOMO that is similar to Basic COCOMO but additionally having the subjective cost drivers that are used to assess the software project attributes and personnel. Six level scale is used by an evaluator to check where attributes are falling. Each attribute is assessed and produces an Adjustment factor. The product of each attribute gives an Effort Adjustment Factor (EAF). This EAF is applied to all formulas of BASIC COCOMO. The third sub model is known as Detailed COCOMO which gives the estimation more accurately. It combines both the Basic and intermediate COCOMO.

COCOMO II [13, 14] breaks into four sub models. Each of them estimates the different inputs and effort of activities in a software project. The first sub model is an Application Composition which uses application points as an input and estimates the effort of developing prototype systems. Early Design is the second sub model which initially calculates an effort based on project requirements, design options and used FP (function point) as an input. The third sub model is known as Reuse which estimates the effort for combining automatically generated reusable components and uses generated LOC (line of code) as an input. Post Architectural is the fourth sub model of COCOMO II which estimates the effort for the development of design specifications of a system and it uses source code lines as an input. Extensive work for an accurate estimation has been done in the past but these traditional models are lagging and needs to be upgraded according to the latest trends.

Nasir et. al. [15] discussed the various techniques of software cost estimation and identified their strengths and weaknesses. It has also presented a project estimation process and elaborate different models derived from COCOMO I and COCOMO II. Boehm et. al. Presented that COCOMO II [16] was a great model till 2005 but now it doesn't fulfil the current development requirements for cost estimation and reusability.

Arnuphaptrairong et. al. [17] surveyed in finding the most accurate model for effort estimation but concluded that available models need to be improved for estimating performance. Saeed et. al. [18] presented that attributes such as organization culture and size are not used to characterize or classify the estimation techniques. This ensures that the less importance is given to the context attributes by the community of software engineering in determining the state of practice in that domain study. Rajeswari and Beena [19] has presented that cost estimation is a most challenging matter for many organizations. The software cost estimation must meet the quality, efficiency and timelines.

III. SWOT Analysis for COCOMO III

SWOT (strength, weaknesses, opportunities and threats) analysis is used to measure these four aspects for identifying feasibility of proposed COCOMO III based on COCOMO I and COCOMO II. Table 1 shows a comparative analysis of the 3 COCOMOs.

3.1 Strengths

- COCOMO I cost drivers are useful in understanding the influence of various factors that affect the software project cost.
- Expert involvement is not much required in COCOCMO II.
- Historical data is not required for estimation in COCOMO II.

3.2 Weaknesses

- According to COCOMO I success largely depends on the model tuning, using past data that might not available always.
- Significant effort of an expert is required in calibrating COCOMO II model.
- COCOMO II doesn't deal with incomplete data and relies on model input in the provision of the required information.
- Across different contexts COCOMO II is not simple to reuse due to its fixed character.
- COCOMO II can't assess the uncertainty of estimation caused by the effort drivers' uncertainty.
- Its difficulty to adapt COCOMO II where there is limited expertise and incomplete data.
- Effort estimation for project activities are fixed within the COCOMO II model.

3.3 Opportunities

COCOMO provides opportunities for development, calibration and evolution of new models that contributes to value-based software engineering.

3.4 Threats

COCOMO leads peoples to believe that they can produce an accurate estimation. But practically, it does not adequately account the project uncertainty level that could poses threat for project objectives and estimation matrices.

IV. Need of COCOMO III

To overcome the deficiencies of COCOMO I and COCOMO II, there is a need of improved version of the model that can deal with the following:

- High quality product
- Developed timely and within budget

- Adaption of COCOMO (even) under limited expertise and incomplete data
- Estimation of effort for project activities should not be fixed with some specific model
- Model should be simpler to reuse across different context
- Adding Risk management and capability of handling uncertainty of estimation

Table 1: Comparative Analysis of COCOMO

COCOMO Models	Drivers	Process Models	Measurement Units
COCOMO I	Cost drivers	Waterfall	Line of Code (LOC)
COCOMO II	Cost drivers, Scale drivers	Rational Unified Process (RUP)	Function Points (FP)
Proposed COCOMO III	Productivity, Reliability, Quality, Availability, Verification and Validation, Usability, Software EngineeringStandards, Risk Assessment	Agile Development Methodology	User Story, Story Points (SP)

- Addressing Personnel turnover, software safety and hardware issues also
- Including Problems of Agile development teams, which are most commonly faced

With the help of predicted assets delivering a fully functional, quality product within time and budget is the greatest challenge in software project development. To meet this challenge, it's the right time to introduce the COCOMO III.

V. Parameters of COCOMO III

The University of Southern California, Center for System and Software Engineering (USC-CSSE) has presented an overview of the model COCOMO III in 2016. The center has considered the function points and SNAP points as a model size driver. The model can be applied at different phases of project asearly, post-architecture and re-estimation of the project throughout its lifecycle. COCOMO III will be capable of transformation of legacy systems, accurate cost estimation of a project life cycle, alternatives analysis, estimation of single and multiple components. The intention of the center is to develop a model that directly take different size inputs. Following cost drivers [20] are initially setup by the center:

Product Attributes

FAIL - Impact of Software Failure
CPLX - Product Complexity
RUSE - Developed for Reusability
SECU - Required Software Security

Dropped: Database Size and Documentation Match to lifecycle needs.

Platform Attributes

PLAT - Platform Constraints Combined Execution and Storage Constraints
PVOL - Platform Volatility

Personnel Attributes

ACAP - Analyst Capability
PCAP - Programmer Capability
PCON - Personnel Continuity
APEX - Applications Experience
LTEX - Language and Tool Experience
PLEX - Platform Experience

Project Attributes

PREC - Precedentedness
FLEX - Development Flexibility
RESL - Opportunity & Risk Resolution
TEAM - Stakeholder Team Cohesion
PCUS - Process Capability & Usage
TOOL - Use of Software Tools
SITE - Multisite Development

Additional Attributes

As we are still working on proposed COCOMO III at our research center (JUW-CCR, Centre for Computing Research, Department of Computer Science and Software Engineering, Jinnah University for Women, Karachi, Pakistan). Here, we are proposing some of the additional attributes to be considered as cost drivers as follows:

QUAL – Quality in Process
USAB – Software Usability
RELI – Software Reliability
STND – Software Engineering Standards

VI. Conclusion

Software projects got delayed due to its complex development process because variety of factors affect it such as product complexity that is being developed and human behavior. Project cost estimation plays important role in the whole process of development. COCOMO models are commonly being used for cost estimation which requires improvement in diligence and adoption because it is lacking accuracy in estimations as industries are much competitive. To cope with the advancements in development and deliver a quality product with accurate estimation it's a time to make improvement in COCOMO model and proposea new version as COCOMO III.

VII. Future Direction

As COCOMO is a post-IR 3.0 (Industrial Revolution 3.0 occurred in 1969) outcome and as now, we are focusing on COCOMO III in the year 2020. So, when IR 4.0 be occurring in around 2070 based of Artificial Intelligence or Cyber Physical Systems, which may also change the structure of Software Development Life Cycle (SDLC) to produce intelligent systems for IR 4.0 and will lead to futuristic version of COCOMO.

References

[1] R. K. Sachan, A. Nigam, A. Singh, S. Singh, M. Choudhary, A. Tiwari, and D. S. Kushwaha, "Optimizing Basic COCOMO Model Using Simplified Genetic Algorithm", *Procedia Computer Science*, 2016, doi: 10.1016/j.procs. 2016.06.107.

[2] R. E. (Dick) Fairley, "The Influence of COCOMO on Software Engineering Education and Training", *J. System Software*, 2007, doi: 10.1016/j.jss.2006.09.044.

[3] B. S. Barry Boehm, Chris Abts, A. Winsor Brown, Sunita Chulani, Bradford K. Clark, Ellis Horowitz, Ray Madachy, and Donald J. Reifer, "Software Cost Estimation with Cocomo II", *Englewood Cliffs, NJ:Prentice-Hall*, 2000.

[4] T. N. Sharma, "Analysis of Software Cost Estimation using COCOMO II", *Int. J. Sci. Eng. Res.*, ISSN 2229-5518, Vol. 2, No. 6, June 2011.

[5] L. V. Patil, S. K. Badjate, and S. D. Joshi, "Develop Efficient Technique of Cost Estimation Model for Software Applications", *Int. J. Comput. Appl.*, (0975-8887) Vol. 87, No. 16, PP. 18-22, February 2014, doi: 10.5120/15292-3951.

[6] S. Kumari and S. Pushkar, "Comparison and Analysis of Different Software Cost Estimation Methods", *Int. J. Adv. Comput. Sci. Appl.*, Vol. 4, No. 1, pp. 153-157, 2013, doi: 10.14569/ ijacsa.2013.040124.

[7] P. Musilek, W. Pedrycz, Nan Sun and G. Succi, "On the sensitivity of COCOMO II software cost estimation model," *Proceedings Eighth IEEE Symposium on Software Metrics*, Ottawa, Ontario, Canada, 2002, pp. 13-20, doi: 10.1109/ METRIC.2002.1011321.

[8] M. M. Albakri and M. R. J. Qureshi, "Measuring Effectiveness of COCOMO I and COCOMO II Using a Case Study", *academia.edu*, 2012.

[9] M. M. Albakri and M. R. J. Qureshi, "Empirical Estimation of COCOMO I and COCOMO II Using a Case Study", *The 2012 Int. Conf. on Software Engineering Research and Practice, semanticscholar*, Corpus ID: 55743858, 2012.

[10] R. T. Futrell, D. F. Shafer, and L. I. Shafer, "Quality Software Project Management", ISBN-13: 978-0130912978, *Software Quality Inst. Series*, 2002.

[11] A. Trendowicz and R. Jeffery, "Software Project Effort Estimation: Foundations and best Practice Guidelines for Success", *Springer*, 2014.

[12] S. Hajar Arbain, N. Azizah Ali, and N. Haszlinna Mustaffa, "Adoption of Machine Learning Techniques in Software Effort Estimation: An Overview", *IOP Conference Series: Materials Science and Engineering*, 2019, doi: 10.1088/ 1757-899X/551/1/012074.

[13] A. Idri, A. Abran, and L. Kjiri, "COCOMO Cost Model Using Fuzzy Logic", *Proceedings of the Joint Conference on Information Sciences, 7th International Conference on Fuzzy Theory and Technology*, Atlantic City, New Jersey, February 27 – March 3, 2000.

[14] A. Mittal, K. Parkash, and H. Mittal, "Software Cost Estimation using Fuzzy Logic", *ACM SIGSOFT Software Engineering Notes*, Vol. 35 No. 1, doi: 10.1145/1668862.1668866, November 2010.

[15] M. Nasir, "A Survey of Software Estimation Techniques and Project Planning Practices", *Proc. Seventh ACIS Int. Conf. on Software Engineering, Artificial Intelligence, Networking, & Parallel/Distributed Computing, SNPD 2006, including Second ACIS Int. Worshop on SAWN 2006*, doi: 10.1109/SNPD-SAWN.2006.11.

[16] B. W. Boehm, B. K. Clark, E. Horowitz, R. Maduchy, R. Selby, and C. Westland, "An overview of the COCOMO 2.0 Software Cost Model", *Software Technology Conference*, 1995.

[17] T. Arnuphaptrairong, "A Literature Survey on the Accuracy of Software Effort Estimation Models", *Lecture Notes in Engineering and Computer Science*, 2016.

[18] A. Saeed, W. H. Butt, F. Kazmi, and M. Arif, "Survey of Software Development Effort Estimation Techniques," *ACM International Conference Proceeding Series*, 2018, doi: 10.1145/3185089.3185140.

[19] K. Rajeswari and Dr. R. Beena, "A Critique on Software Cost Estimation", *Int. J. Pure Applied Mathematics*, Vol. 118, No. 20, PP. 3851–3862, 2018.

[20] Brad Clark (Mod.), COCOMO III Workshop, USC CSSE Annual Research Review 2014, *University of Southern California, Center for System and Software Engineering (USC-CSSE)*.