Impact of Dynamicity and Causality on Cost Drivers in Effort Estimation

Suman Roychoudhury, Sagar Sunkle and Vinay Kulkarni
Tata Research Development and Design Centre, Tata Consultancy Services,
54B Hadapsar Industrial Estate, Pune - 411013, India

Keywords: Effort Estimation, Cost Drivers, COCOMO II, System Dynamics.

Abstract: Software cost estimation is an important step that decides upon the effective manpower, schedule, pricing, profit and success for executing any medium to large sized project. Depending upon the underlying development methodology (e.g., code-centric, model-driven, product-line etc.) and past experience, every enterprise follows some cost estimation strategy that may be derived and customized from a standard cost model (e.g., COCOMO II). However, most software cost estimation techniques that are done at the start of a project do not consider the dynamicity and causality among cost drivers that can alter the accuracy of estimation. In this paper, we investigate those cost drivers that are time and inter-dependent and use system dynamics to simulate their effect in effort estimation.

1 INTRODUCTION

Software cost estimation is a major challenge for any enterprise (Lum et al., 2003) as it helps to plan out the development schedule, the necessary manpower, the essential hardware and software infrastructure along with other support systems that together play a key role in the success of any software developmental project. There are several software estimation techniques that can be adopted (e.g., algorithmic cost modeling, expert judgment, estimation by analogy) by an organization based on the organization policy, the customer's budget, past experience etc.

These techniques can be broadly classified into two groups, namely, parametric cost estimation that follows a more formal methodology, and non-parametric cost estimation (i.e., expert, adhoc or analogy based), which is broadly informal and prone to human judgmental errors. Nevertheless, every organization must practice a formal or informal modeling technique, without which there could be unnecessary and uncontrollable spiraling cost in terms of manpower management, delay in delivery schedule etc. In case of selecting a formal modeling technique, the choice of a cost estimation model may be derived or customized from a standard cost estimation technique like COCOMO II (Boehm et al., 2000), or FPA (Albrecht et al., 1983) or organizations can build their own model from scratch.

However, there are several factors that influence the applicability of a cost estimation model including what development methodology one needs to adopt during a software construction process (e.g., code-centric software development, model-driven software development or product line based software development). Typically the cost model and cost benefits are different based on the methodology that is adopted in the development lifecycle. For example COCOMO II is ideally suitable for code-centric software development, which we have specialized for model-driven software development (Forrester, 1961). For SPLE, cost estimation technique such as COPLIMO can be applied (Boehm et al., 2004).

2 MOTIVATION

Most parametric cost estimation models consist of various cost factors or cost drivers that contribute significantly to the overall cost of the project. These cost drivers are calibrated from one organization to another and also depend on the nature of the project to be developed. Project managers typically use the calibrated values of these cost drivers and given the size of the project can estimate various project

Roychoudhury S., Sunkle S. and Kulkarni V..
Impact of Dynamicity and Causality on Cost Drivers in Effort Estimation.
DOI: 10.5220/0004969904090414
In *Proceedings of the 16th International Conference on Enterprise Information Systems* (ICEIS-2014), pages 409-414
ISBN: 978-989-758-028-4

parameters like manpower, schedule, infrastructure etc. at the start of a project. However, most of the project management decisions that are made do not take into consideration that the cost drivers themselves can change with time in addition to being inter-dependent. Both the time and inter-dependent nature of cost drivers can significantly alter the accuracy of cost estimation. Therefore, a more pragmatic approach towards better decision making would be to play out various cost estimation scenarios and reason about the causal relationship of the cost drivers besides considering their dynamic (i.e., time-dependent) nature. This is a significant deviation from standard cost estimation practice that tends to rely upon parametric models that are inherently static in nature and base their predictions by taking snapshots of a development situation at a given point of time. Specifically, this research investigates some of the time and inter-dependent nature of cost drivers for a particular cost estimation technique, i.e., COCOMO II and use system dynamics to play out various cost estimation scenarios for better decision making leading towards improved project planning and management. Due to lack of calibrated data, we used simulation techniques to capture the dynamic nature of cost drivers and better predict various cost estimation scenarios. While in this position paper we provide validation of our approach with a proof-of-concept (PoC) case study application, in future, we would like to extend this proof by validating our approach (simulated data) with actual data available from various projects.

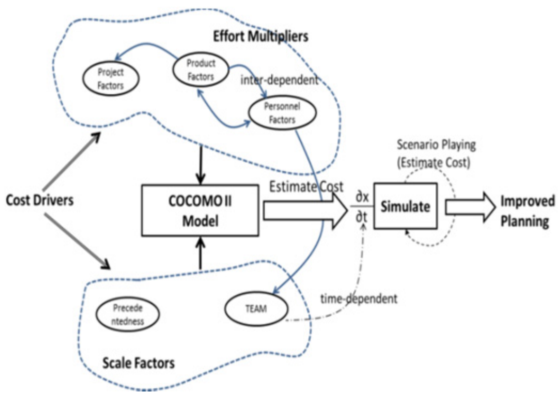

Figure 1: Simulating time and inter-dependent cost drivers for better planning.

Fig 1 demonstrates our line of attack with respect to COCOMO II cost drivers, but we believe the approach is general enough to be applied to any parametric cost estimation model. The paper is organized as follows. Section 3 introduces the dynamic factors that influence cost estimation. In Section 4, we present our PoC case study and simulate the effect of dynamic and the inter-dependent cost drivers. We compare the related approaches in Section 5 before concluding in Section 6.

3 DYNAMIC COST DRIVERS

In this section, we introduce some of the dynamic cost drivers (i.e., both scale factors and effort multipliers) in COCOMO II that we realized in course of our investigation. The dynamic nature of these cost drivers imparts a time-dependent behavior to them such that they vary throughout the software development life cycle. Below we identify and explain only those scale factors that change with time.

Precedentedness(PREC). This scale factor attempts to capture similarity in products developed by an organization. PREC is further defined in terms of:

a. Organizational understanding of product objectives
b. Experience in working with related software systems

From our past experience in developing large business-critical information systems as well as from inputs received via a questionnaire to a team of developers and project managers, we found that both a) and b) generally increase with time.

Architecture/Risk Resolution(RESL). This scale factor indicates the extent to which an organization implements a risk management plan. Out of the several characteristics that define RESL, we list down the ones that are dynamic in nature.

a. Level of uncertainty in key architecture drivers
b. Number and criticality of risk items

Team Cohesion(TEAM). The team cohesion scale factor as described in COCOMO II considers synchronization in the objectives and cultures of stakeholders and their experience in operating as a team. Response to our questionnaire reveals that almost all the characteristic ratings that define TEAM increase over time. The remaining two scale factors product flexibility (FLEX) and maturity (PMAT) do not show significant dynamic behaviour and therefore do not influence dynamic estimation.

The COCOMO II effort multipliers (EM) are typically to be used after the software life-cycle architecture has been developed. Out of the seventeen different effort multipliers as described in COCOMO II, only those which we realized to be dynamic are discussed here. Among the Product

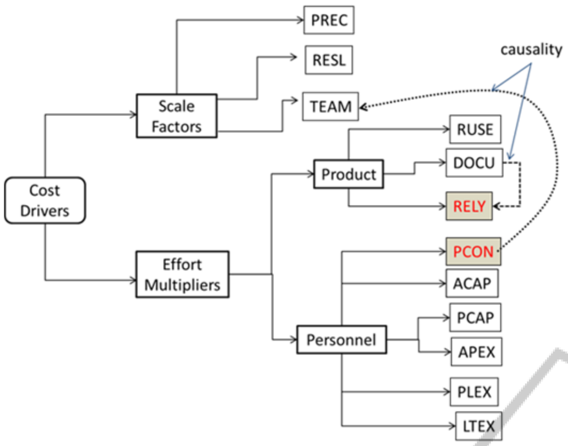

Figure 2: Dynamic and Causal Cost Drivers in COCOMO II.

factors in EM, our study suggests that only *Development for Reuse* (RUSE) and *Documentation for Life Cycle Needs* (DOCU) change with time, i.e., reusability of assets as well as clarity of documents generally improves over time. None of the Platform factors (e.g., storage constraint) in EM is time-dependent whereas almost all the Personnel factors (e.g., analyst capability (ACAP), programmer capability (PCAP)) in EM are dynamic in nature. Also effort multipliers that are related to experience like *Application Experience* (APEX), *Language* and *Tool Experience* (LTEX) and *Platform Experience* (PLEX) are time-dependant.

3.1 Causality among Cost Drivers

Certain cost drivers may not have a dynamic effect on effort estimation but their changing values may have direct influence on other cost drivers. One such cost driver (effort multiplier)is *Personnel Continuity* (PCON) that is not intrinsically dynamic in nature but it has a causal effect on *team cohesiveness* (TEAM) cost factor. This finding led to our investigation towards unearthing the causal relationship among various cost drivers. This causal relationship is independent of the dynamic nature of cost drivers. For example, in model-driven development, we often require better tooling capabilities (static) which in turn demand developers with modeling experience in addition to programming. However there could be certain cost drivers that are both dynamic and inter-dependent. For example, team cohesiveness (TEAM), which is dynamic can alter if there is a change in personnel continuity (i.e., PCON), which is a static cost driver. Similarly, improved or better documentation can lead to better product design or understanding which in turn can improve product reliability (i.e., RELY). This dynamic and causal nature of cost drivers is summarized in Fig 2. Since most of the cost drivers are project specific, we will further explore this causal relationship in the light of a case study that is introduced in the following section.

4 SIMULATING EFFORT ESTIMATION USING SYSTEM DYNAMICS

In this section, we will demonstrate several scenarios by simulation and show how time and inter-dependent cost drivers can affect cost estimation and thereby influence the decision making process in terms of manpower requirement, personnel skills, tooling infrastructure etc. For our case study purpose, we chose cost estimation of development projects that involve mobile-enabling of business applications (Roychoudhury et al., 2011). Typically these applications needs to be platform agnostic (i.e., browser-based hybrid) and robust on security features and performance.

4.1 Static Effort Estimation – Base Case

As a base case, we first do a static estimation without considering dynamicity or causality in cost drivers. The scale factors we considered as per our past experience are given below:

PREC	FLEX	RESL	TEAM	PMAT
3.72	4.05	5.65	4.38	1.56

Assuming the projects are of 30-50 KSLOC and early design effort multipliers to be nominal, the combined sum of scale factors will be $\Sigma SFj = 19.36$. According to (Boehm et al., 2000) , the value of E is given by Equation 1 as follows:

$$E = B + 0.01 \times \sum_{j=1}^{5} SF_j \qquad (1)$$

where B = 0.91 (for COCOMO II.2000)

4.2 Dynamic Estimation Using System Dynamics

As stated earlier, we have chosen System Dynamics (SD) to play out dynamic cost estimation scenarios because of its inherent formalism that helps to simulate the behaviour of complex systems and understand how they respond over a period of time (Meadows, 2008). There are three core elements in a SD model, namely, stocks, flows, and variables

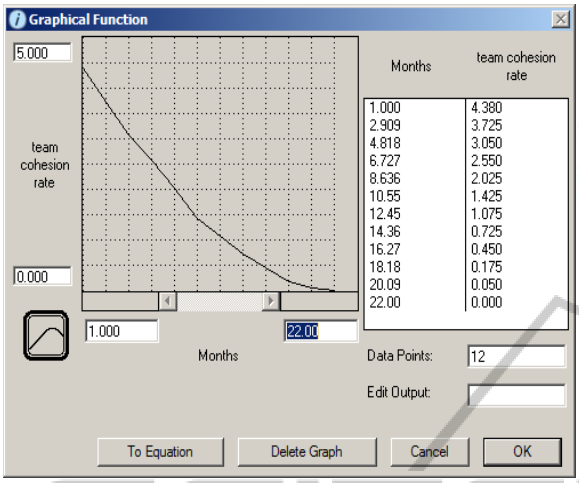

Figure 3: Dynamicity in Team Cohesion.

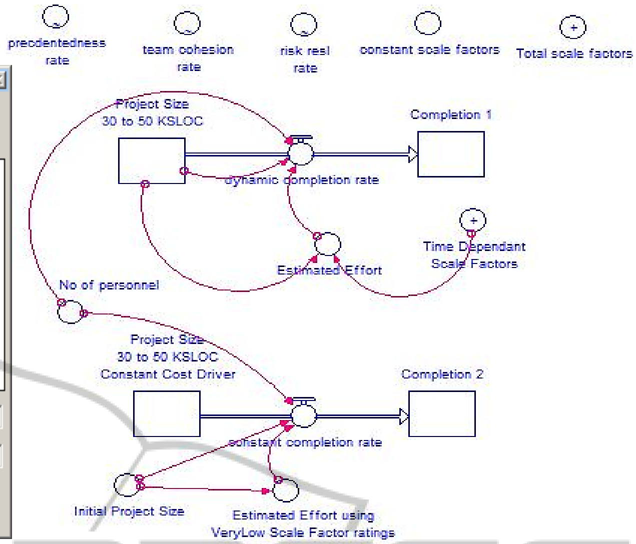

Figure 4: Simulating the effect of dynamic cost drivers.

(Forrester, 1961). Stocks are accumulations that characterize state of a system. Stocks generate information upon which decisions and actions are based.

4.2.1 Time-Dependency among Cost Drivers

Let us now consider the result of dynamic cost drivers on effort estimation. For our first scenario, let us assume for simplicity, that the values of effort multipliers are nominal (i.e., 1) throughout the project development phase whereas the scale factors change with time (e.g., see Fig 3).

Fig 4 shows the SD model for scenario 1. In this figure, the stocks represent states like "project size" and "project completion" (rectangles) that dynamically change as the project executes. The scale factors are typically modeled as variables (circles) whereas the project completion rate is modeled as flow. The upper part of the figure (in dotted line) depicts the dynamic cost estimation model (i.e., Scenario 1) while the bottom part of the figure simulates the base case (static).

Fig 5a and 5b shows the plots for dynamic cost estimation w.r.t the base case. The y-axis shows the size of the projects that ranges between 30-50K SLOC (Fig 5a – 50K SLOC, Fig 5b – 30K SLOC), whereas the x-axis shows the duration of the projects in months. The *red line* shows how the project would have executed statically whereas the *blue line* shows the effect of dynamic cost drivers on effort estimation. It can be clearly seen from the figure, that there is a distinct deviation in effort estimation between the static and the dynamic technique

4.2.2 Causality among Cost Drivers

In this scenario we will take into consideration how inter-dependent cost drivers can impact decision making process during effort estimation. The system dynamic model for this scenario is shown in Fig 6.

The plus (+) sign in the relationship illustrates a positive causal relationship whereas the negative (-) sign captures a negative relationship. For example, use of model-driven tools may necessitate higher developer skills or experience, while superior user experience (via use of graphical widgets) may have negative impact on performance or execution time. Similarly cost drivers like platform volatility (PVOL) or support for multiple mobile OS can be achieved either via browser based (i.e., html5) hybrid approach or individually constructing each application on a native platform. While the latter come with rich user experience, the former come with a reduced cost and minimal developer experience. Fig 7 shows such a scenario where option A makes use of standard tools with less developer experience (green line) and option B makes use of model-driven tools with higher developer experience. Although option A helps in completing the project early but it comes with higher initial tooling plus developer cost. Therefore, there needs to be a balance among various cost factors or a causality study has to be carried out, such that correct decisions can be made towards reaching a desired solution.

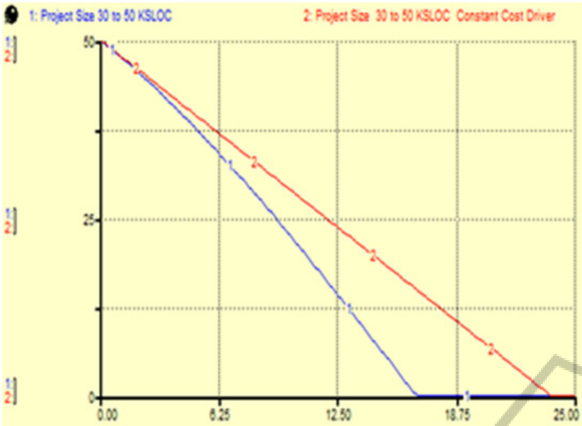

Figure 5a: Simulating 50K SLOC Effort Estimation.

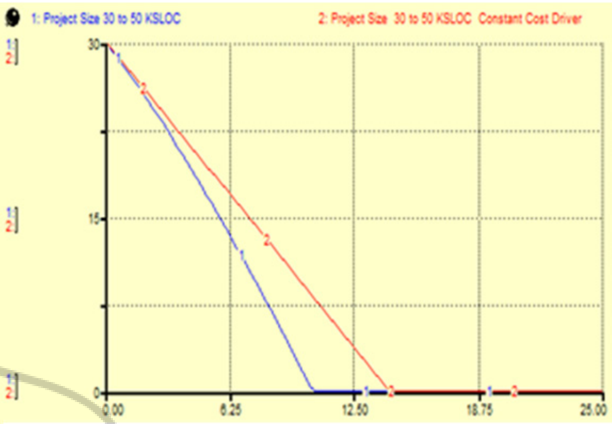

Figure 5b: Simulating 30K SLOC Effort Estimation.

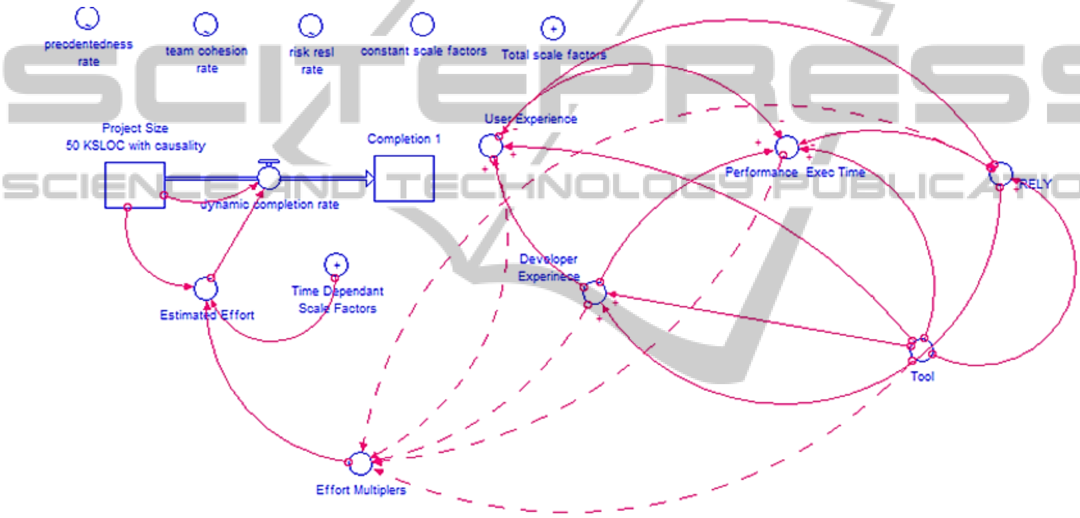

Figure 6: SD model for Scenario 2 capturing causality.

5 RELATED WORK

For Code-Centric software development there are a variety of techniques such as Walston-Felix Model (Walston et al., 1977), Bailey-Basili Model (Bailey et al., 1981), COCOMO Basic and COCOMO II (Boehm et al., 2000), FPA (Albrecht et al., 1983) and Doty Model for KLOC > 9. For Product Line-Centric software development a popular cost estimation technique that has been adapted from COCOMO is called COPLIMO (Boehm et al., 2004). For Model-Centric software development paradigm we have identified our own customized version of COCOMO taking into account the relevant cost drivers that are model specific (Sunkle et al., 2012). Most of the above techniques take a static view of cost estimation where cost is estimated upfront taking into account the relevant factors that go into cost. A way to improve the simulated data by using Monte Carlo simulation is presented in (Kläset al., 2008). This is something we intend to take up in our future work. The seminal paper that tried combining COCOMO with system dynamics was (Smith, 1991); however that work was mainly focused on sensitivity analysis to enhance COCOMO's cost driver set. (Madachy, 1996) used system dynamics model of an inspection-based software lifecycle process that served to examine the effects of inspection practices on cost, scheduling and quality throughout the lifecycle. Fuzzy and neural network based adaptive techniques have been applied in the past to improve the accuracy of cost estimation (Azzeh et al., 2010). Nevertheless, none of them consider the dynamic nature of certain cost drivers and combines it with simulation to improve the accuracy of cost estimation.

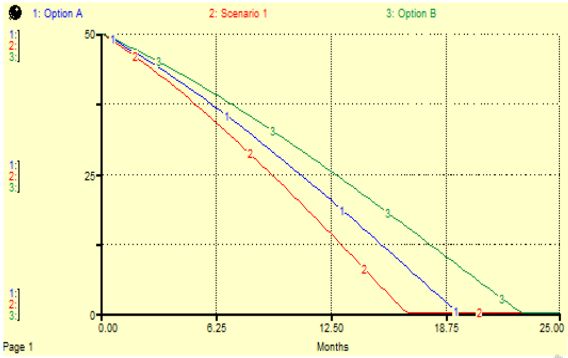

Figure 7: Causality and its effect on Effort Estimation Decisions.

6 CONCLUSION

Software cost estimation is an important step towards managing various aspects of a project like manpower, schedule, risk etc., which can indirectly influence the outcome of a project in terms of varying degree of success or failure. Until now, most of the parametric cost estimation techniques have estimated cost from a static point of view. However, in this position paper, we introduced the dynamic nature of cost drivers and using simulation techniques we demonstrated how they impact the cost estimation of software development projects. Moreover, we realized there are inherent causal relationships among cost drivers that results in trade-off among several decision choices. In addition, using the notion of scenario playing we demonstrated how risks like attrition can be played out in advance, thereby allowing teams to have early contingency plans in place for certain foreseeable situations. Although our proof-of-concept was based on analyzing the dynamic and causal nature of COCOMO II cost drivers, we believe the concept is general enough to be applied to any other parametric cost estimation model as well.

REFERENCES

Albrecht, A. J., Gaffney Jr., J.E., 1983. Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation. *IEEE Transactions on Software Engineering*, *Vol. 9, Issue 6*, pp. 639-648.

Azzeh, M., Neagu, D., Cowling, P.I., 2010: 'Fuzzy grey relational analysis for software effort estimation', *Empirical Software Engineering, Vol. 15, Issue 1*, pp. 60-90, Springer.

Bailey, J.W., Basili, V.R., 1981. A meta-model for software development resource expenditures. In *Internatoinal Conference in Software Engineering, ICSE'81* pp. 107–116.

Boehm, B., Abts, C., Brown, W., Chulani, S., Clark, B., Madachy, R., Reifer, D., Steece, Bert., 2000. *Software Cost Estimation with Cocomo II*. Prentice Hall.

Boehm, B., Brown, A.W., Madachy, R., Yang, Y., 2004. A Software Product Line Life Cycle Cost Estimation Model. *Empirical Software Engineering, ISESE*, pp. 156-164, IEEE.

Forrester, J., 1961. Industrial Dynamics, *MIT Press*.

Kläs, M., Trendowicz, A., Wickenkamp, A., Münch, J., Kikuchi, N., Ishigai, Y., 2008. The Use of Simulation Techniques for Hybrid Software Cost Estimation and Risk Analysis. *Advances in Computers*, pp. 115 - 174.

Lum, K.., Bramble, M., Hihn, J., Hackney, J., Khorrami, M., Monson, E., 2003. Handbook of Software Cost Estimation,*Report, Jet Propulsion Laboratory*.

Madachy, R.J., 1996. System dynamics modeling of an inspection-based process. In *Internatoinal Conference in Software Engineering, ICSE'96*, pp. 376 - 386

Meadows, D., 2008. *Thinking in systems : a primer*. Chelsea Green Publishing, Vermont.

Walston, C.E., Felix, C.P., 1977. A method of Programming Measurement and Estimation.*IBM Systems Journal, 16, (1)*, pp. 54–73.

Roychoudhury, S., and Kulkarni, V., 2011. Mobile-Enabling Enterprise Business Applications using Model-Driven Engineering Techniques. In Proc. *2nd Workshop on Software Engineering for Mobile Application Development, MobiCase'11*.

Smith, R.W., 1991. Investigating the utility of coupling COCOMO with a system dynamics simulation of software development. *Master Thesis, Naval Postgraduate School.*

Sunkle, S., Kulkarni, V., 2012. Cost Estimation For Model-driven Engineering. In *MoDELS'12*, pp. 659-675