

Risk Analysis in Software Cost Estimation: A Simulation-Based Approach

Vikram Singh¹, Varun Malik², Ruchi Mittal³

¹ Professor, Department of Computer Science and Engineering, Chaudhary Devi Lal University, Sirsa, India. Email: vikramsinghkuk@yahoo.com

² Assistant Professor, Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India

³ Professor, Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India.

Article History: Received: 10 November 2020; Revised 12 January 2021 Accepted: 27 January 2021; Published online: 5 April 2021

Abstract: Risk analysis and cost estimation are two important aspects of project planning that can either make the way or break the way to a project's success. At the same, both these tasks are difficult and painstaking, but whether someone likes it and not, the project's success depends heavily on them. As documented by Fredric Brooks Junior in his legendary book "The Mythical Man-Month," planning, scheduling, and estimation have been central to software engineering since its early days in the 1970s. Present communication presents a simulation-based approach to estimate the costing schedule of a software development project. The results show that simulation is expedient as well as efficient in terms of time, effort, and cost requirement and provides pragmatic results.

Keywords: Risk analysis, PERT, Monte Carlo computation, software effort estimation, simulation.

1. Introduction

For almost every project, the budget is of utmost importance. It requires correct estimation for the successful and fruitful completion of the project. But correct estimation is not easy to come by, before the actual beginning of the project because of inherent uncertainties. Estimation at an early stage of a project is as difficult as driving a car being blindfolded while taking directions from someone else (Wood 2002). In (Brooks 1975), Fred Brooks' observations were stemmed from almost a decade-long experience as the project lead of IBM's OS/360. He tried to manage the behind-schedule project by adding manpower. Later, he conceded that the decision had proved counter-productive and had further delayed the project. His experiences with OS/360 had led him to summarise his experiences in form of a book "The Mythical Man-Month" (Brooks 1975).

However, temporal and budgetary schedules not only get troubled by poor estimation but also by the inherent cost risks. And, according to Sharif and Basri (2011), risk can be defined as "*the likelihood of suffering loss which impacts the project in terms of poor quality, cost overrun, schedule slippage, etc.*" Thus, cost risks can disturb the budgets and can result in cost overrun. Therefore, it is intelligent to analyse the cost risks before beginning the actual project.

The purpose of the risk analysis is to forecast what may happen if a certain set of actions take place (Mansoorzadeh & Yusof 2011). So, accurate risk analysis and estimation of cost together can increase the chances of success in budget project completion. And for this, the use of simulation can be of great help. As described in Xing-xia and Jian-wen (2009), "*simulation is the practice of imitating the true behaviour of a system into a model based on some realistic conditions*". In this paper, Monte Carlo computation has been used which involves the use of random numbers to address the uncertainty inherent in an otherwise deterministic problem. It simulates a system by reproducing random numbers using some probability distribution and then analyses all the results collectively to forecast the estimates and unknown future risks (Hira 2001).

2. Review of Literature

Effort estimation in software projects comprises an early but important activity. The size of software, usually measured in KLOC, works as the most popular input for the estimation of software effort. Authors in (Moataz, Irfan & Jarallah 2013) have proposed a framework for software effort prediction by using UML models created early in SDLC. Uncertainty in software size is accounted for by a probability density function

Effort estimation accounts for an important activity in the software development life cycle and it comprises an important part of the planning and management of a software project. But, more often than not, in the software engineering practice, the software engineering principles like cost estimation are not given due importance in the lifecycle of a software artifact. In Trendowicz (2013), the author has presented the CoBRA model for effort estimating in a software project. This method takes into account the artifact parameters and human judgment. The scope of the method extends well beyond effort estimation and it helps in managing project-related risks as well. Trendowicz (2013) has recorded numerous real-world cases wherein the CoBRA has been put to use in varied real-world software project planning situations.

Authors in Santos and Bela (2013) have investigated the role of factors like cost, quality, schedule, and scope in software development projects and have established that irrespective of the nature of an engineering project – traditional engineering or software engineering – the aforementioned process management variables are critical to the decision-making process in any engineering project. And, particularly in software development projects, the planning and execution phases are subjected to the creativity of individuals with key portfolios. In Santos and Bela (2013), the authors have found that the cost, quality, schedule, and scope of critical tasks in the software development projects be monitored and controlled effectively. Authors have also reported a toolkit based on data mining techniques to find the correlations between the most relevant parameters of human resource, the project underhand, and the organisation e.g. experience of individuals, the complexity of the project, and maturity of organizational practices. This toolkit has been developed for helping the management in planning, monitoring, and controlling a software development project.

Authors in Uzzafer (2017) have found a research gap in that estimated costs can be represented as the percentiles of probability distributions, but, these estimates may not be sub-additives. Uzzafer (2017) has reported a model which takes probability distributions as its input and produces as output, the sub-additive cost estimates. The model proposed herein is based on expectations in contrast with the traditional approach where percentiles of probability distributions determine the estimated costs. The proposed model works on the premise that estimated costs of components add up to give a pretty good estimate of the overall project cost. Other way round, the estimated cost of a project can be decomposed into sub-estimates to represent the estimated costs of project components.

In Kumar and Yadav (2015) authors have underlined the importance and vitality of risk management for the success of software development projects. Whereas every SDLC phase poses a potential risk involving people, technology, money, time, hardware, and software resources. Sources of software risks since it involves hardware, software, technology, people, cost, and schedule. A whole lot of risk factors affect the software development process, but, Kumar and Yadav (2015) have focused on working out how the outcome of a software development project is possibly correlated with risk factors. Bayesian Belief Network-based probabilistic model has been proposed to estimate/assess the risk in the software process.

Whereas many researchers have considered the task of software cost estimation with plausible precision as unachievable, there are others who believe that this task can be achieved using established principles of mathematics and management. Briciu, Filip and Indries (2016) have addressed the problem faced in the estimation of software cost owing to rapid changes in software process, principles, and practices. Herein, a genetic algorithm-based model has been reported for estimating the cost of software products.

The distinct nature of the software process, its produce and failure of hardware cost and risk estimation tools in the software industry have led researchers to work on risk management in the software industry for a long. Software engineering as an engineering discipline is aimed at identification, assessment, alleviation, and monitoring of risk, among other objectives. Further, the risk profile of large software systems is quite different from small-scale systems. Maruf, Ghazia & Urooj (2018) have underlined this difference and presents a detailed comparison of risk management models for software projects.

3. Network Representation of Software Project

The network of a website development project is considered here on which simulation is applied for its risk analysis and cost estimation. Here, as shown in Figure 1, PERT (Project Evaluation and Review Technique) network representation is used, according to which the circles represent the nodes and the numbers inside the circles represent the node numbers (Sharma 2009). The arcs/edges between two nodes represent an activity and the number along with an activity represents the activity number (written in bold). The dark circles represent the starting and finishing activities of the project.

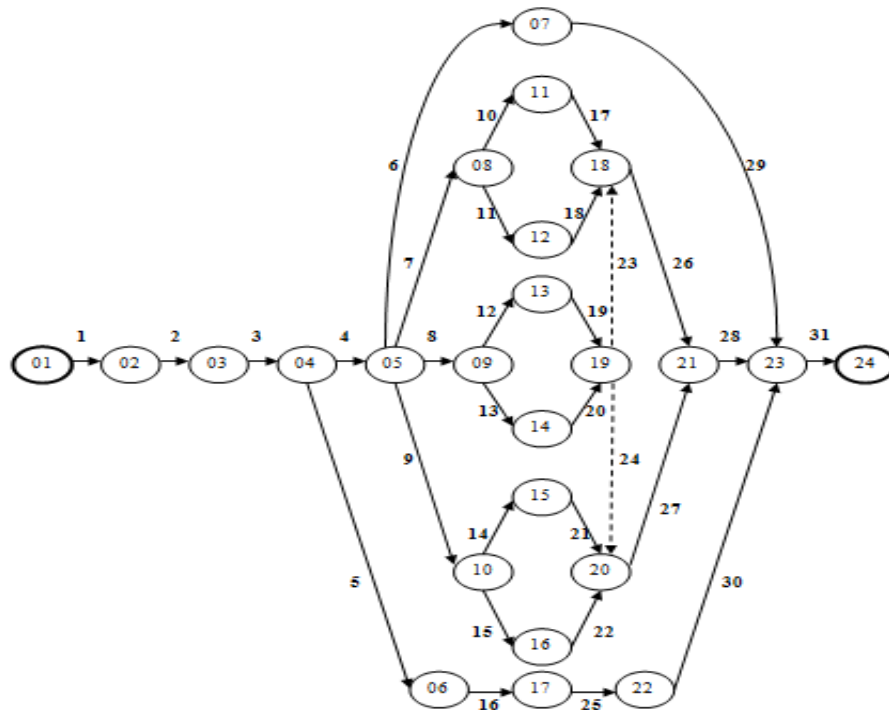


Figure 1. PERT network representation of a website development project.

For each activity, mean (M) and standard deviation (SD) values are calculated initially, since they are the measure of uncertainty in a project (Sharma & Suri 2011), using the following formulae:

$$M = (OCE[N] + 4 * MCE[N] + PCE[N]) / 6 \quad \text{----- (1)}$$

$$SD = (PCE[N] - OCE[N]) / 6 \quad \text{----- (2)}$$

Where OCE[N] is the Optimistic Cost Estimate of Nth activity duration, MCE[N] is the Most likely Cost Estimate of Nth activity duration, and PCE[N] is the Pessimistic Cost Estimate of time of Nth activity. The M and SD values and the starting nodes (Sn[i]) and finishing nodes (Fn[i]) of different coordinating activities in the network diagram are presented in Table 1.

Table 1. Cost estimates of all activities.

Activity	Sn[i]	Fn[i]	M	SD	Activity	Sn[i]	Fn[i]	M	SD
1	1	2	1450	216.67	17	11	18	2050	116.67

2	2	3	1200	133.33	18	12	18	1500	166.67
3	3	4	1500	166.67	19	13	19	2000	166.67
4	4	5	1500	166.67	20	14	19	1583.33	250
5	4	6	600	66.67	21	15	20	2000	166.67
6	5	7	2000	333.33	22	16	20	1500	166.67
7	5	8	1666.67	133.33	23	19	18	0	0
8	5	9	1500	166.67	24	19	20	0	0
9	5	10	1500	166.67	25	17	22	400	66.67
10	8	11	2466.67	200	26	18	21	2083.33	250
11	8	12	2000	166.67	27	20	21	2083.33	250
12	9	13	2000	166.67	28	21	23	3000	333.33
13	9	14	2000	166.67	29	7	23	0	0
14	10	15	2000	166.67	30	22	23	1916.67	250
15	10	16	2000	166.67	31	23	24	4000	1000
16	6	17	700	66.67					

4. Simulator Design

A simulator viz. SIM_RACE (Simulator for Risk Analysis and Cost Estimation) has been designed here, which simulates the cost of various coordinating activities in the website development project to address the uncertainty that can be present in it. The simulator has been designed in C language on Intel core i5 processor, under the windows operating system. The algorithm of the simulator is given as below:

Algorithm: SIM_RACE

- i. Read input parameters for the project activity network -
 Number of activities (M),
 Number of nodes (N),
 Start node number,
 Sink node number,
 Mean(m) and standard deviation (SD) values of each of M activity durations, and
 Total number of simulation runs (TRUN).
- ii. Define and initialize local variable 'CRUN=0' to keep track of current simulation run
- iii. Repeat steps 4 and 5 while CRUN<=TRUN.
- iv. Generate the 'random cost value' (RCV) of each activity using mean and standard deviation values using the following Box-Muller transformation:
 - a. $C = (-2\log_e x_1)^{1/2} \cos(2\pi x_2)$
 - b. $RCV = SD * C + M$
- v. For determination of critical activities and cost estimation, go through the following steps
 - a. Traverse the network in the forward direction.
 - b. Calculate project completion cost (PCC).
 - c. Traverse the network in the backward direction.
 - d. Mark critical activities and determine the critical path.
- vi. Calculate criticality indices.

5. Results and Discussion

The output obtained after running the simulator for 1000 runs is presented here using tables and graphs. Table 2 shows the results obtained after the first simulation run which spans over the forward

and backward pass through the project’s activity network. The results comprise of the values of different parameters such as RCV (Random Cost Value), ESC (Earliest Starting Cost), EFC (Earliest Finishing Cost), LSC (Latest Starting Cost), LFC (Latest Finishing Cost), and CRT (signifies whether a project activity is critical¹ or not).

Table 2. Calculated cost values in 1st run

Run = 1	PCV= 20848.320 (in rupees)					
Activity	RCV	ESC	EFC	LSC	LFC	CRT
1	1502.18	0	1502.18	0	1502.18	1
2	1189.1	1502.18	2691.28	1502.18	2691.28	1
3	1541.82	2691.28	4233.1	2691.28	4233.1	1
4	1207.97	4233.1	5441.07	4233.1	5441.07	1
.
.
.
30	2202.03	5681.48	7883.51	15143.2	17345.2	0
31	3503.1	17345.2	20848.3	17345.2	20848.3	1

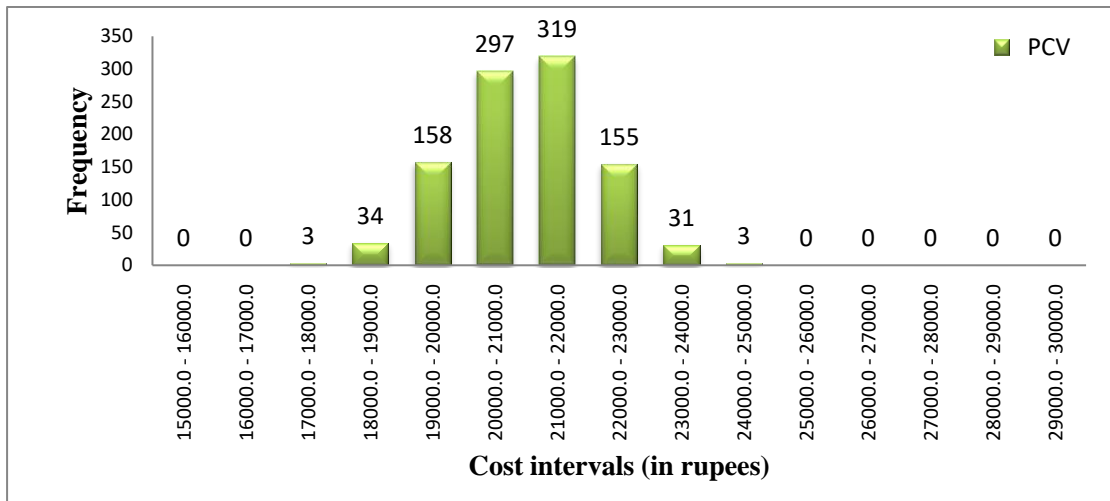
In each simulation run, PCC and critical path are determined which gives the estimate about the total cost to complete the project and the set of critical activities due to which the cost may overrun. Table 3 presents the values of these parameters in different simulation runs.

Table 3. PCC and Critical paths in different runs.

Run	PCC (in rupees)	Critical Path
1	20848.3	1 2 3 4 7 10 17 26 28 31
2	19490.7	1 2 3 4 7 10 17 26 28 31
3	21013.2	1 2 3 4 7 10 17 26 28 31
4	22901.4	1 2 3 4 7 10 17 26 28 31
.	.	.
.	.	.
.	.	.
999	19311.1	1 2 3 4 7 10 17 26 28 31
1000	21404.9	1 2 3 4 8 12 19 23 26 28 31

Using the values of PCC in 1000 simulation runs, the frequency of PCC values is calculated which is shown below with the help of Graph 1.

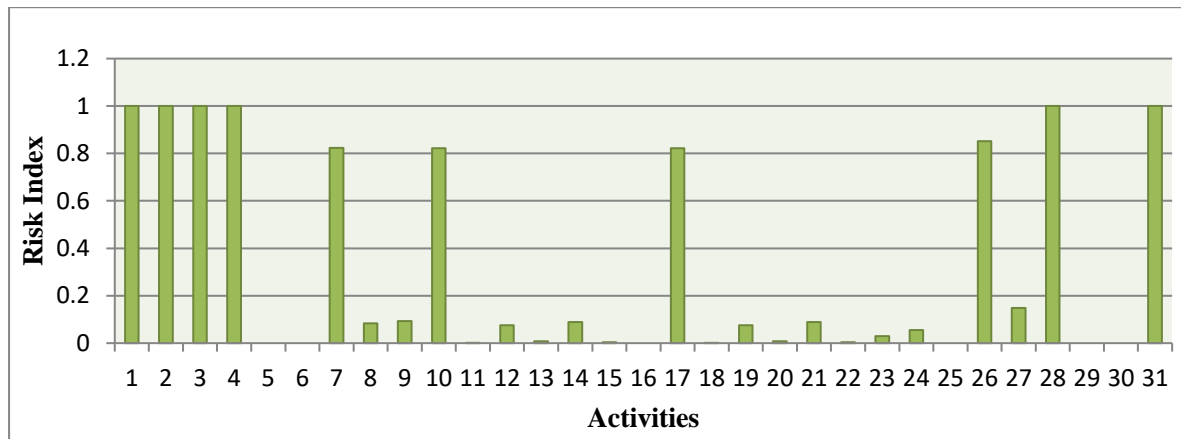
¹ An activity is defined as critical, if any delay in its accomplishment causes a corresponding delay in the project completion time.



Graph 1. Frequency distribution of PCC's values in different runs.

Here, the graph shows that the interval of 21000.0-22000.0 rupees occurred for the maximum number of times that is 319 times, which means chances are higher for the budget to fall under this interval. Using this information, decisions regarding the project budget can be made aptly.

Next, Graph 2 shows the risk indices of the coordinating activities of the project. The risk index is calculated by dividing the total number of times an activity becomes critical to the total number of simulation runs (Deo 2003).



Graph 2. Risk indices of activities.

Activities having risk indices falling in the range of 0.8-1.0 are more critical as compared to others. Therefore, according to the graph, activities 1, 2, 3, 4, 7, 10, 17, 26, 28, and 31 are the most critical activities and needed to be handled with more care.

As already stated that correct estimation is essential for a project, with the help of the designed simulator it can be attained quite accurately. For this, the results of the simulator are needed to be analysed carefully. So, after analysing the results of the simulator it is observed that values of run number 791 were repeated maximum times and therefore its results can be used as the estimates of cost. Results of run number 791 are shown with the help of Table 4.

Table 4. Results of run 791

RUN =791	PCC= 21321.171 (in rupees)					
Activity	RCV	ESC	EFC	LSC	LFC	CRT
1	1587.76	0	1587.76	0	1587.76	1
2	1046.503	1587.76	2634.263	1587.76	2634.263	1
3	1626.045	2634.263	4260.308	2634.263	4260.308	1
4	1397.467	4260.308	5657.776	4260.308	5657.776	1
5	582.612	4260.308	4842.92	12870.245	13452.857	0
.
.
.
30	2160.126	5923.034	8083.159	14532.97	16693.096	0
31	4628.074	16693.096	21321.171	16693.096	21321.171	1

6. Conclusion

Risk analysis and cost estimation ensure the success of a project if done correctly. By using the described approach, a project manager can find the answers to “how much will it cost”, “what should be the estimates”, “what are the critical activities” and “what-if” like questions before the very beginning of the project. The designed simulator has provided quality results through Monte Carlo Simulation using less time, effort, and cost. The results obtained are neither pessimistic nor optimistic but realistic, and that increases the chances of in-budget project completion. And that is, after all, the ultimate goal of any project!

References

- [1]. Wood, D. (2002). Risk Simulation Techniques to Aid Project Cost - Time Planning and Management, *Jour. Risk Management*, 4, 41–60. DOI: <https://doi.org/10.1057/palgrave.rm.8240108>
- [2]. Brooks, F. P. Jr. (1975). *The Mythical Man-Month: Essays on Software Engineering*, Reading, Mass.: Addison - Wesley Publishing Company. URL: https://umich.instructure.com/files/884126/download?download_frd=1
- [3]. Sharif, A. M. & Basri, S. (2011). A Study on Risk Assessment for Small and Medium Software Development Projects, *International Journal on New Computer Architectures and their Applications*, 1(2), 325-335. URL: <http://oaji.net/pdf.html?n=2015/544-1430918787.pdf>
- [4]. Mansoorzadeh, S. & Yusof, S. M. (2011). Reliable Project Scheduling with Combination of Risk Management and Critical Chain Schedule, *Research and Development, IEEE Student Conference on, IEEE, Cyberjaya*, 442-447. DOI: 10.1109/SCOReD.2011.6148780.
- [5]. Xing-xia, W. & Jian-wen, H. (2009). Risk Analysis of Construction Schedule Based on Monte Carlo Simulation, *Computer Network and Multimedia Technology*, IEEE, Wuhan, 1-4. DOI: 10.1109/CNMT.2009.5374816.
- [6]. Hira, D.S. (2001). *System Simulation*, Sultan Chand and Company Private Limited, New Delhi.
- [7]. Moataz, A., Irfan, A. & Jarallah, A. (2013). Probabilistic size proxy for software effort prediction: A Framework, *Information and Software Technology*, vol. 55, pp. 241-251. URL: <https://dl.acm.org/doi/abs/10.1016/j.infsof.2012.08.001>.
- [8]. Trendowicz, A. (2013). *Software Cost Estimation, Benchmarking, and Risk Assessment*, Springer - Verlag, Berlin Heidelberg. URL: <https://link.springer.com/content/pdf/bfm%3A978-3-642-30764-5%2F2%2F1.pdf>.
- [9]. Santos, J. & Belo, O. (2013). *Estimating Risk Management in Software Engineering Projects*, In Perner P. (eds) *Advances in Data Mining. Applications and Theoretical Aspects. ICDM 2013. Lecture Notes in Computer Science*, vol 7987. Springer, Berlin, Heidelberg. URL: https://link.springer.com/chapter/10.1007/978-3-642-39736-3_7.
- [10]. Uzzafer, M. (2017). Cost Estimation of Software Projects: A Sub-additive Approach, *The Journal of Modern Project Management, North America*, 4. URL: <https://www.journalmodernpm.com/index.php/jmpm/article/view/JMPM01211/0>.

- [11]. Kumar, C. & Yadav, D. K. (2015). A Probabilistic Software Risk Assessment and Estimation Model for Software Projects, *Procedia Computer Science*, 54, 353-361. URL:<https://reader.elsevier.com/reader/sd/pii/S1877050915013654?>
- [12]. Briciu, C.V., Filip, I. & Indries, I. I. (2016). Methods for Cost Estimation in Software Project Management, *IOP Conference Series: Materials Science and Engineering*, Wuhan, China. URL: <https://iopscience.iop.org/article/10.1088/1757-899X/106/1/012008/pdf>.
- [13]. Maruf, P., Ghazia, Q., & Urooj, P. (2018). A Critical Analysis of Software Risk Management Techniques in Large Scale Systems, *IEEE Access*, 6. URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8291743&tag=1>.
- [14]. Sharma, S. D. (2009). *Operations Research*, 15e, Kedar Nath and Ram Nath, Meerut.
- [15]. Sharma, I. & Suri, P. K. (2011). Schedule Risk Analysis Simulator using Beta Distribution, *International Journal on Computer Science and Engineering*, 3(6), 2408-2414. URL: <http://www.enggjournals.com/ijcse/doc/IJCSE11-03-06-122.pdf>.
- [16]. Deo, N. (2003). *System Simulation with Digital Computer*, Prentice-Hall of India, New Delhi.