IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-6, Issue-1 E-ISSN: 2347-2693

Risk Ananlysis and Estimation of Scheduling of Software Project – Using Stochastic Approach

Yogita Bindra[1*], Rajesh Garg[2], Navneet Kaur [3]

[1*] Department of Computer Science and Engineering, Ganpati Institute of Technology and Management, Bilaspur, India
[2]Department of Electronics and Communication Engineering, Seth Jai Parakash Polytechnic for Engineering, Damla, India
[3]Department of Computer Science and Engineering,Ganpati Institute of Technology and Management, Bilaspur, India
[*]*Corresponding Author: yobindra24@gmail.com, Tel.: +91-8397849610*

Available online at: www.ijcseonline.org

Received: 03/Jan/2018, Revised: 11/Jan/2018, Accepted: 21/Jan/2018, Published: 31/Jan/2018

Abstract—A project is a combination of several interrelated activities which must be performed in a certain order of its completion. To meet tight deadlines in software projects, managers need to understand key reservations about the scheduling techniques and how to use a schedule risk analysis to provide information crucial to a project's success. This paper describes an application of simulation which simulates the duration of the activities for analyzing schedule risk and providing reliable estimates of time. Monte Carlo Simulation Methods are mostly used for analyzing schedule risk. In this the random numbers are generated to simulate the software project number of times. The primary objective of the simulation is to find out the effect of uncertainties on the schedule of project completion. The designed simulator SRAES for a live website Filmtribe uncovered the critical paths and risky activities in the project and also provided the risk indices of those risky activities. The simulator also calculated the Project Completion Time in less than 1 minute which can take months to calculate analytically. Therefore, it is concluded that Monte Carlo Simulation is an important technique for risk analysis and estimation of scheduling in any type of software projects.

Keywords—Schedule Risk Analysis, Monte Carlo Simulation, Estimation

I. INTRODUCTION

Management is must in every type of project whether it's Software project or Hardware project. In any software project there are two most important factor i.e. budget and schedule. It is necessary that the project should be completed before the deadline and within budget to achieve the success in software projects. If the schedule slip then budget of the project also increased. The achievement of software project depends upon the nature of planning of the project, scheduling and controlling of the different activities of software project. McManus (2004) identified that 65% of the project failures are accounted by management issues and 35% by technical issues [5]. Program Evaluation and Review Technique (PERT) is a guide to assist and control the assets to meet the calendar date of the activities which include the high level of instability. To control the instability in planning procedure we apply the probabilistic Monte Carlo Simulation. Monte Carlo Simulation is concept that gives a numerical estimation of the stochastic highlight of the framework reaction. A simulation based software project risk management tool helps manager to identify high risk areas of software process and software product and plan, organize and manage the software development project in a cautious way. The paper is organized into 5 sections. The first section is the introduction of risk analysis and Monte Carlo Simulation. The second section discusses the designing of simulator. The third section discusses the network model of the system. The fourth section represents the results of the simulator. The fifth and last section is conclusion.

II. DESIGNING OF SIMULATOR

Simulator may be a software tool resulting from application of simulation to real life systems. The objective of this work is to design the simulator for risk analysis and estimation of scheduling. The tactic is named SRAES (Simulator for risk analysis and estimation of scheduling) that is mentioned in subsequent sections [7].The algorithm for the simulator is given below:

SRAES ALGORITHM:

1. Read input data for activity network corresponding to given project consisting of n activities and m

nodes and number of simulation runs, mean time and variance for each activity.

2. Perform steps 2 to 8 for 1000 simulation runs.
3. Generate random input data using Box-Muller Transformation method to generate random samples of time for each activity.
4. Perform Forward pass by traversing the network from starting node to finishing node.
5. Calculate pcv i.e. project completion value.
6. Perform Backward pass by traversing the network from finishing node to starting node.
7. Identify risky activities and simultaneously update their counter variable by incrementing that with 1 and add that risky activity to list of critical paths.
8. Print risk indices of each activity and store the results for further analysis.

III. NETWORK MODEL

SRAES is based on one of the two popular concepts i.e. CPM and PERT. It involves modelling of system under the study and represents the system in the form of network model. In this paper, a website Filmtribe is considered for the simulation experiment. Here, PERT (Project Evaluation and Review Technique) network representation is used for representing the website Filmtribe as a network which consists of 24 activities and 19 nodes/events as shown in Figure1.

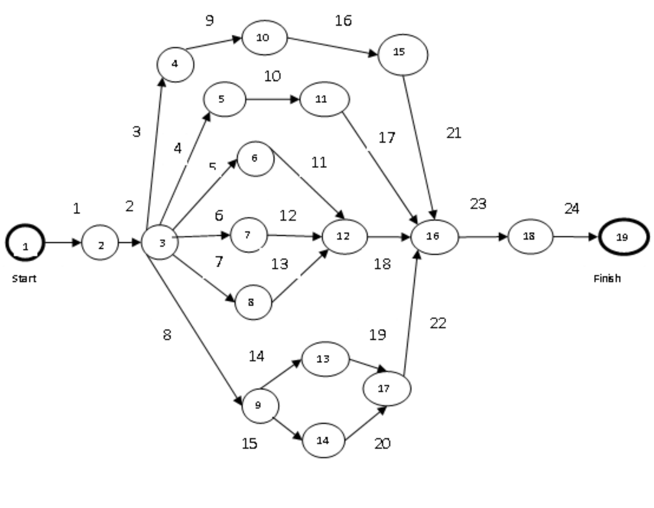


Figure 1: Network representation of website Filmtribe

Each activity is determined by its starting node, finishing node, three time estimates – Optimistic, Most Likely, Pessimistic time. On the basis of these three parameters, the mue (mean) and sigma (standard deviation) values are calculated for each activity using the following formulas:

$$\text{Mean} = \mu_n = (OE\,[I] + 4ME\,[I] + PE\,[I])/6 \qquad \text{----------} (1)$$

$$\text{Standard deviation} = \sigma_n = (PE\,[I] - OE\,[I])/\,6 \qquad \text{----------} (2)$$

Where, OE [I] = Optimistic Estimate of I^{th} activity
ME [I] = Most likely Estimate of I^{th} activity.
PE [I] = Pessimistic Estimate of time of I^{th} activity

The calculated values of mue and sigma along with the starting and finishing nodes of project activities are listed in Table 1.

Table 1 Time estimates of all activities.

Activity no.	snode	fnode	Mue	Sigma
1	1	2	2	0.33
2	2	3	2.5	0.5
3	3	4	6.0	0.67
4	3	5	6.0	0.67
5	3	6	8.0	1.0
6	3	7	8.33	1.3
7	3	8	8.0	1.0
8	3	9	6.0	0.67
9	4	10	5.16	0.83
10	5	11	6.0	0.67
11	6	12	4.1	0.5
12	7	12	4.6	0.67
13	8	12	4.6	0.67
14	9	13	5.0	0.67
15	9	14	8.0	1.0
16	10	15	3.0	0.33
17	11	16	8.0	1.0
18	12	16	5.16	0.83
19	13	17	4	0.67
20	14	17	3.16	0.5
21	15	16	5.16	0.83
22	17	16	5.16	0.83
23	16	18	7.83	0.83
24	18	19	9.33	1.67

The mue and sigma values represent the measure of uncertainty associated with each activity [6]. Using the values of mue and sigma random samples of duration can be generated using box-muller transformation method [7]:

$$x = (-2 \log_e r1)^{1/2} \cos(2\pi\, r2)$$
$$rdv[i] = x * sigma[i] + mue[i]$$

Here, r1 and r2 are two random numbers in the range of (0,1), x is a sample from standardized normal distribution, rdv[i] is the generated random duration value of the i^{th} activity.

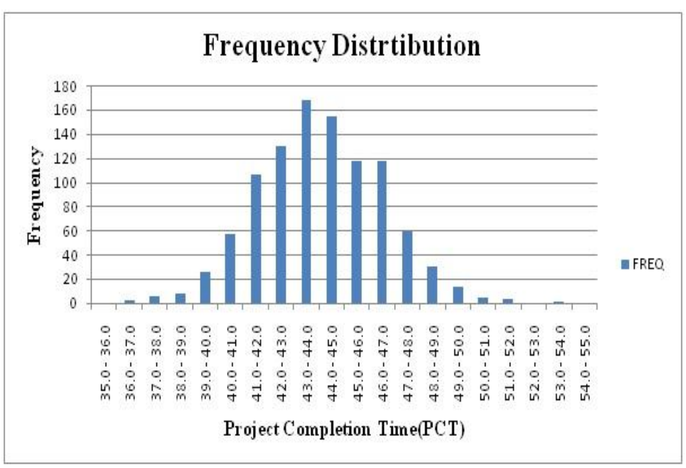
IV. RESULTS AND DISCUSSION

After 1000 simulation runs, the results generated by the simulator are presented here with the help of tables and graphs. For achieving successful on-time completion, correct estimates are very crucial. For this, Project completion value (PCV) is calculated at the end of each run to estimate the value of total time to complete a project. Table 2 presents the value of PCVs in different runs.

Table 2 PCV values (in days)

RUN	PCV	--------------	RUN	PCV
1	43.974	--------------	991	45.758
2	42.894	--------------	992	42.149
3	45.955	--------------	993	40.622
4	46.717	--------------	994	46.008
5	43.147	--------------	995	45.729
6	43.837	--------------	996	43.316
7	46.778	--------------	997	42.115
8	41.171	--------------	998	46.214
9	41.379	--------------	999	42.650
10	46.051	--------------	1000	44.347

Now, by using these PCV values, the frequency of PCV values can be calculated which represents the number of times a PCV value repeats itself in different runs. Graph 1, shows the frequency distribution of PCV's in 1000 runs in different intervals.



Graph 1: Frequency distribution of PCVs.

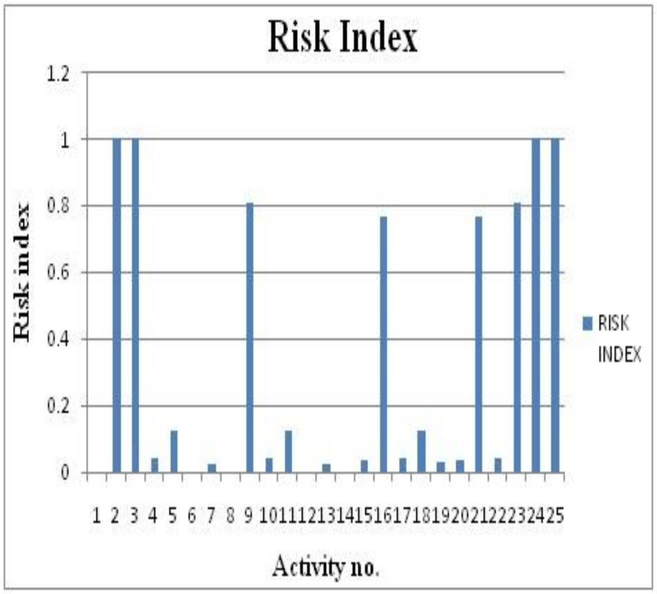
Here the duration of 43.0-44.0 days occurred for the maximum times and it means chances are higher for the completion of the website FilmTribe project in this time interval.

After this, risk index of every activity is calculated.The ratio of number of times an activity becomes critical to the Total number of simulation runs gives the risk index of an activity. Table 3 shows the risk indices of all activities. The risky activities collectively form a critical path. The critical path is the longest path in the network and it may cause delay to a project due to the risk of schedule slippage [8]. An activity with risk index lying in the range of 0.7 to 1.0 is considered to be riskier than others. Hence, as shown in table 4, activities 1, 2, 8, 15, 20, 22, 23 and 24 are riskier than other activities and the critical path made by these activities that is 1-2-8-15-20-22-23-24 is the most critical path. Therefore these activities are required to be handled more carefully than others.

Table 3 Risk index

Activity	Risk Index	Activity	Risk Index
1	**1**	13	0.004
2	**1**	14	0.038
3	0.046	**15**	**0.766**
4	0.123	16	0.046
5	0.002	17	0.123
6	0.023	18	0.029
7	0.004	19	0.038
8	**0.804**	**20**	**0.766**
9	0.046	21	0.046
10	0.123	**22**	**0.804**
11	0.002	**23**	**1**
12	0.023	**24**	**1**

Graph 2 shows the plotted form of table 3 that is it shows the critical activities with their respective risk index. The criticality index measures how often one specific task was on the critical path during the simulation. For delivering a project on time we need correct time estimates of all activities and if we can have them then the chances of on-time project completion increase.



Graph 2 Activities v/s Risk Index

V. CONCLUSION

The research concludes that software projects are used in every aspect of human life, but the development of these software projects is not an easy task. Attaining the goal of successful on-time project completion is extremely important. The designed simulator SRAES for a live website Filmtribe uncovered the critical paths and risky activities in the project and also provided the risk indices of those risky activities. It is observed that the higher is the risk index of the activity, the higher should be the attention paid to that activity. The simulator also calculated the Project Completion Time in less than 1 minute which can take months to calculate analytically. Future work can be done on the cost estimation of the software project by considering the project completion value and risk indices of the critical activities. Therefore, it is concluded that Monte Carlo Simulation is an important technique for risk analysis and estimation of scheduling in any type of software projects.

REFERENCES

[1] Miler, J. & Górski, J. 2001. "*Implementing risk management in software projects*", 3rd national conference on software engineering, Otwack, Poland.
[2] Xing-xia, W. and Jian-wen, H. 2009. "*Risk Analysis of Construction Schedule Based on Monte Carlo Simulation*". Computer Network and Multimedia Technology, IEEE, Wuhan, pp. 1 – 4
[3] Sharma, S.D. 2009. *Operations Research*, 15th Ed. Reprint, Kedar Nath and Ram Nath, Meerut.
[4] Hira, D. S. 2001. *System Simulation*, S. Chand & Company Pvt. Ltd, New Delhi.
[5] Subhas C. Misra, Vinod Kumar and Uma kumar (2006), "*Different Techniques for Risk Management in Software Engineering: A Review*", ASAC Banff, Alberta.
[6] Sharma, I. et al. 2011. "*Schedule Risk Analysis Simulator using Beta Distribution*". International Journal of Computer Science and Engineering (IJCSE), Vol. 3, No. 6, pp.2408-2414.
[7] Deo, N. 2003. *System Simulation with Digital Computer*, Prentice-Hall of India, New Delhi.
[8] Pressman, R. S. 2006. *Software Engineering - A Practitioner's Approach*, fifth edition, McGraw-Hill.

Authors Profile

Er. Yogita Bindra received B.Tech in Computer Science and Engineering in 2015 and currently pursuing her M.Tech in Computer Science and Engineering from Ganpati Institute of Technology and Management, Bilaspur, Haryana.

Dr. Rajesh Garg received M.Tech from National Institute of Technology, Kurukshetra, and Ph.D. from Chaudhary Devi Lal University in 2005 and 2012, respectively. He has been sharing his knowledge for more than 18 years as a Lecturer in Seth Jai Parkash Polytechnic for Engineering, Damla, Haryana. His research interests cover simulation, risk management, embedded systems and also software testing, with over 15 technical publications. Apart from this, Dr. Garg is a regular blood donor and an active social worker.

Er. Navneet Kaur received the B.Tech. and M.Tech. degrees in Computer Science and Engineering in 2011 and 2013, respectively. Presently, she is working as an Assistant Professor in Computer Science and Engineering Department in Ganpati Institute of Technology and Management ,Bilaspur ,Haryana.

