Schedule Risk Analysis Simulator using Beta Distribution

Isha Sharma
Department of Computer Science and Applications,
Kurukshetra University, Kurukshetra, Haryana (INDIA)
ishasharma211@yahoo.com

Dr. P.K. Suri
(Dean, Faculty of Engg. & Technology, Professor & Chairman)
Department of Computer Science and Applications,
Kurukshetra University, Kurukshetra, Haryana (INDIA)
pksuritf25@yahoo.com

***Abstract*—This paper describes an application of simulation and Modelling in Software risk management. This paper describes a simulation based software risk management tool which helps manager to identify high risk areas of software process. In this paper an endeavour has been made to build up a Stochastic Simulator which helps in decision making to identify the critical activities which are given due priorities during the development of Software Project. In response to new information or revised estimates, it may be necessary to reassign resources, cancel optional tasks, etc. Project management tools that make projections while treating decisions about tasks and resource assignments as static will not yield realistic results. The usual PERT procedure may lead to overly optimistic results as many pass which are not critical but slightly shorter than critical on the basis of estimated activity duration or average durations. Due to randomness of durations, these pass under some combination of activity durations, could become longer than the average longest path. Such paths would be ignored while using the PERT technique on the basis of the average durations. In order to overcome this problem and be more reasonable, the said Stochastic Simulator has been designed by generating random samples from a specific probability distribution associated with that particular activity of SPM. The said simulator is also not bugged with overly estimated results.**

Keywords- SPM; Activity Network; Critical Activities; Critical Path; CPM; PERT; Simulation

I. INTRODUCTION

Successful project management is intricate and multifaceted. Tasks must be assigned to resources with different characteristics, taking complex dependencies, constraints and uncertainties into consideration, attempting to meet goals related to costs and time [1]. To design a SPM system is to simulate the application of a project plan to see how uncertainties about task duration, etc., affect the outcome

Is software project management really different from management of any engineering project, for example, house construction project? "Yes" and "No" as well.

Project management tool described in this paper can be used by establishing an activity network for the software project. These networks are prepared and used as follows:

•A set of project activities is identified;

•Interdependencies between project activities are established;

•Concurrencies between project activities are established;

•Stochastic duration required to carry out each activity is generated;

•High risk areas i.e. critical activities are identified.

Aim of schedule risk analysis is to decrease the time delivery [2]. A missed schedule can reduce market impact, create customer dissatisfaction and can raise internal costs also. Following questions are important while analysing software project:

•How do we distribute human resources over calendar time?

•What software development activities are expected?

•What task parallelism is expected?

•What milestones might be used to show progress?

•Are risk management tools/methods available?

Most of the major issues of concern to a planner and project manager are addressed by a variety of computer based tools. Depending on the budget available, it is possible to choose a tool with the sophistication and functionality to suit your needs. However, in one area both the standard techniques and the tools still provide little if any support. Schedule risk modelling need not be unduly complex or time consuming. Probabilistic representations allow for much more realistic predictions than are possible by conventional methods, so they make it possible for plans to be realistic without a large amount of detail. Risk modelling enables planners to give a complete view of a project, from the top down, to whatever level of detail is appropriate.

II. Simulation of Software Project Network

Projects usually begin with some predetermined objectives and some ideas about how to meet these objectives. The project consists of completion of a set of tasks which will conclude in meeting the desired objectives [3]. One of the most important responsibilities of the Project Team is to choose that Project Process which maximizes the likelihood that the time objectives, cost and functionality will be met. Next, the Project Team creates estimates of task completion into a form capable of being turned into a mathematical risk profile. The easiest way to create the required task completion data is to estimate the following three values for each task: (1) the most likely time to complete the task. (2) The minimum or most optimistic time to complete the task. (3) The maximum or most pessimistic time to complete the task. This methodology creates mathematical risk profiles from these three parameters using beta or log normal probability distributions in a manner transparent to the user. These three estimates for every project activity are supplied by domain expert from his/her knowledge of the past projects of similar type. For Project modelling efforts, the only data required consists of tasks, the predecessor/successor relationships between tasks, and data about the risks associated with each activity(3 time estimates) [4].

To simulate the variability inherent in risk, the Monte Carlo approach requires multiple passes over the project network. Somewhere between 1000 and 2000 passes or trials is sufficient. The Monte Carlo approach calls upon random number generators to generate task completion times. For a given task, the series of estimates that stem from the random number generator will, in the aggregate, be adjusted to closely approximate the risk profile provided for that task. Each pass or trial creates estimates of task completion, milestone completion, and project completion. The multiple estimates of project completion are compiled into a histogram and a cumulative probability distribution graph. This output graphs are alternate representations of the underlying risk profile of project completion time. The simulation system can create risk profile representations of the time required to reach any part of the project. In this way the probability of meeting particular date for specific project or milestone or task can be estimated [5].

CPM and PERT pick one path for emphasis and ignore the possible effects of all other paths, including near critical paths and risky paths. Monte Carlo Simulation methods, on the other hand, provide a mechanism for the Project Team to examine all paths, to take the inherent risk into account and to clearly prioritize the tasks in order of the likelihood that the task will delay the project[6][7].

III. Design of Simulator

Simulator is a software tool resulting from application of modelling and simulation to real life systems. It involves modelling of system under the study i.e. software project, representing the model in network form and execution of computer form (network form) and evaluation of simulation experiment [8]. The aim of this work is to design the simulator for schedule risk analysis. The method is called SRAS (Schedule Risk Analysis Simulator) which is discussed in subsequent sections.

IV. NETWORK REPRESENTATION OF EXAMPLE SOFTWARE PROJECT

A software project is considered for the simulation experiment. With respect to each of the software project activity three parameters namely optimistic duration, pessimistic duration and most likely duration is established. These three time values provide a measure of uncertainty associated with each activity.

Optimistic Time: The optimistic time is the shortest possible time in which the activity can be finished.

Most Likely Time: The most likely time is the estimate of the normal time the activity would take.

Pessimistic Time: The pessimistic time represents the longest time the activity could take if everything goes wrong

Schedule planning of an example project is shown as the Figure. 1 and the data of each activity's estimated duration such as optimistic duration Ak, pessimistic duration Bk, and most likely duration Mk are given in Table I.

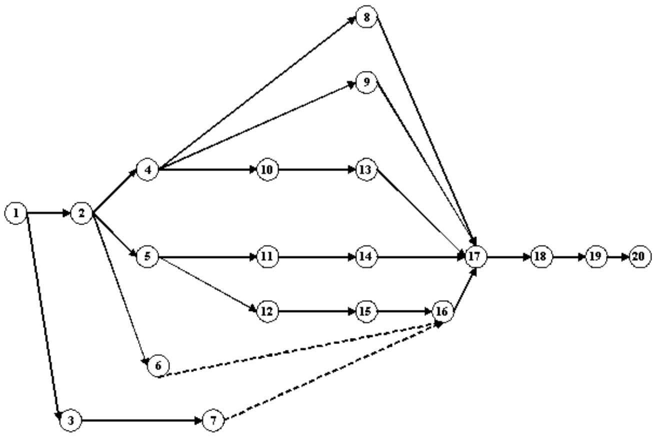

Figure. 1 Network Diagram of software project

In Table I, each activity is specified by its starting node, finishing node and its duration μ(mean) and σ (standard deviation) are computed by assuming that the duration of each activity was given by a Beta distribution i.e.,

$x \in \beta(\alpha 1, \alpha 2)$ where

$\alpha 1 = \mu\ ((\mu\ (1 - \mu) / \sigma) - 1)$

$\alpha 2 = (1 - \mu)\ ((\mu\ (1 - \mu) / \sigma) - 1)$

Algorithm: SRAS_Simulator

Step-1: Read input data for activity network corresponding to given software project consisting of n activities and m nodes.

Step-2: Compute the values of parameters μ (i) (mean) and σ (i) (sigma) belonging to beta distribution

Step-3: Initialize simulation run counter

Step-4:Generate activity duration samples using Beta Distribution

[Generate pseudorandom numbers r1, r2 € [0, 1]

Calculate $g\ (A_k + (B_k - A_k)*r1) = (r1)^{k1-1}*(1-r1)^{k2-1} / (B_k - A_k)$ where k1=4, k2=4.

Calculate $g\ (M_k) = (M_k - A_k)^{k1-1}*(B_k - M_k)^{k2-1} / (B_k - A_k)^{(k1+k2-1)}$

If $g\ (A_k + (B_k - A_k)*r1) / g\ (M_k) >= r2$ then

TIME (K) = $A_k + (B_k - A_k)*r1\ \forall$ K =1, 2...N

Else go to step 3]

Step-5: Traverse the network for forward pass

Step-6: Traverse the network for backward pass

Step-7: Compute the risk index counter for each activity.

Step-8: Print risk indices of each activity.

This simulator is designed using C++ language under Windows operating system on an Intel compatible machine The system discussed here is stochastic and dynamic in nature. The next – event discrete simulation model, has been used for conducting simulation experiment

Activity(K)	**START [K]**	**FINISH[K]**	**A[K]**	**M[K]**	**B[K]**
1	1	2	7	15	35
2	1	3	2	5	8
3	2	4	4	7	15
4	2	5	3	7	15
5	2	6	7	15	30
6	3	7	5	10	25
7	4	8	1	3	10
8	4	9	1	2	3
9	4	10	2	3	3
10	5	11	1	2	5
11	5	12	10	14	40
12	6	16	0	0	0
13	7	16	0	0	0
14	10	13	1	3	10
15	11	14	10	15	30
16	12	15	1	4	10
17	14	17	1	2	5
18	15	16	7	20	40
19	16	17	1	3	5
20	8	17	15	30	100
21	9	17	10	20	50
22	13	17	7	10	15
23	17	18	6	10	20
24	18	19	2	7	15
25	19	20	2	7	20

TABLE I Estimated duration of each activity

V. RESULTS

Some of the crucial questions that proposed schedule risk simulator set out to answer are:

•What are the risks associated with achieving time targets?

•When can the project as a whole be completed?

•Can it be delivered on time?

A project simulation uses a model that translates the uncertainties specified at a detailed level of the project into their potential impact on project objectives. Simulations are typically performed using the Monte Carlo technique. In a simulation, the project model is computed many times (iterated), with the input values randomized from a probability distribution function (e.g., duration of schedule activities) chosen for each iteration from the probability distributions of each variable. A probability distribution (e.g., completion date) is calculated.

Our model simulated, tabulated, and plotted the frequency and distribution functions for, 1000 simulated project schedule outcomes. Table II shows the simulated result of each activity of the software project

Table III shows the risk (criticality) index values for various activities.

During each simulation run of the network an activity becomes risky (critical) or non risky (non- critical). The ratio of Number of times an activity becomes critical to the Total Number of simulation runs, gives the risk index of an activity. Total Number of simulation runs, gives the criticality index of an activity.

Results of simulation experiment depict that:

• Activity 1, 23, 24 and 25 became risky during all the simulation runs.

• Activity 4 became risky during 87.6 percent of simulation runs.

• Activity 11, 16, 18 and 19 became risky during 85.9 percent of simulation runs

Activity	Expected Duration (days) - Input			Output		
K	OPT[K]	MOST[K]	PES[K]	Mu[K]	Variance[K]	Sigma[K]
1	7	15	35	21	21.7	4.66
2	2	5	8	5	1.00	1.00
3	4	7	15	9.5	3.36	1.83
4	3	7	15	9	4.00	2.00
5	7	15	30	18.5	14.69	3.83
6	5	10	25	15	11.11	3.33
7	1	3	10	5.5	2.25	1.50
8	1	2	3	2	0.11	0.33
9	2	3	3	2.5	0.02	0.16
10	1	2	5	3	0.44	0.66
11	10	14	40	25	25.00	5.00
12	0	0	0	0.00	0.00	0.00
13	0	0	0	0.00	0.00	0.00
14	1	3	10	5.5	2.25	1.50
15	10	15	30	20	11.11	3.33
16	1	4	10	5.5	2.25	1.50
17	1	2	5	3	0.44	0.66
18	7	20	40	23.5	30.25	5.50
19	1	3	5	3	0.44	0.66
20	15	30	100	57.5	200.69	14.16
21	10	20	50	30	44.44	6.66
22	7	10	15	11	1.77	1.33
23	6	10	20	13	5.44	2.33
24	2	7	15	8.5	4.69	2.16
25	2	7	20	11	9.00	3.00

TABLE II: Simulated result of each activity of the software project

Activities	Risk Index	Activities	Risk Index
1	**1**	14	0.003
2	0	15	0.019
3	0.148	**16**	**0.859**
4	**0.876**	17	0.019
5	0	**18**	**0.859**
6	0	**19**	**0.859**
7	0.133	20	0.133
8	0.016	21	0.016
9	0.003	22	0.003
10	0.019	**23**	**1**
11	**0.859**	**24**	**1**
12	0	**25**	**1**
13	0		

TABLE III: Risk Indices of Activities

Figure 2 presents the result of Table III in graphical form. The criticality index measures how often one specific task was on the critical path during the simulation

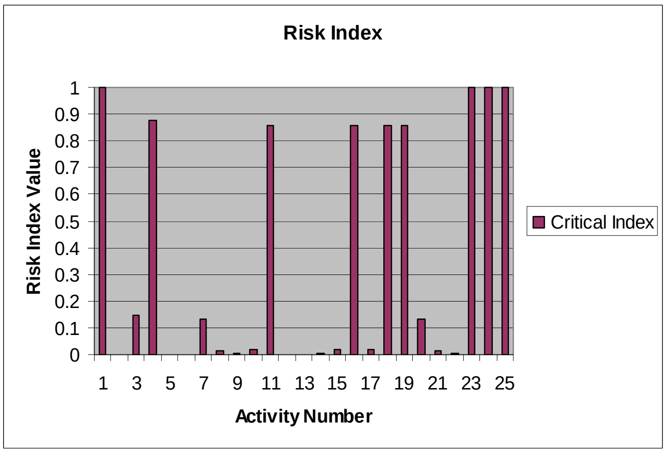

Figure 2. Critical Index Chart

Figure 3 shows the simulated frequency distribution chart for the software project being simulated.

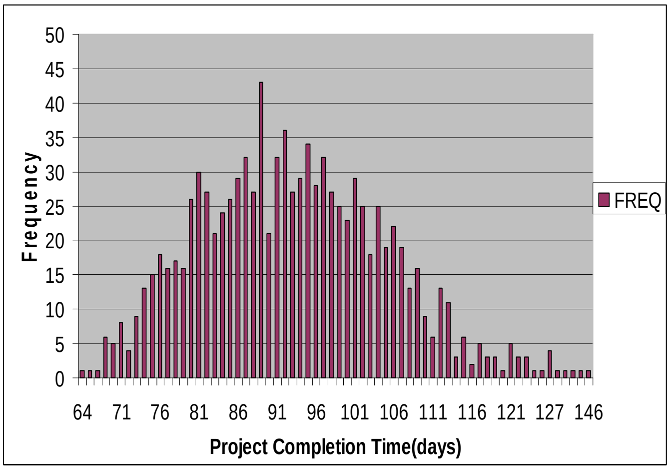

Figure 3. Simulated Frequency Distribution Chart

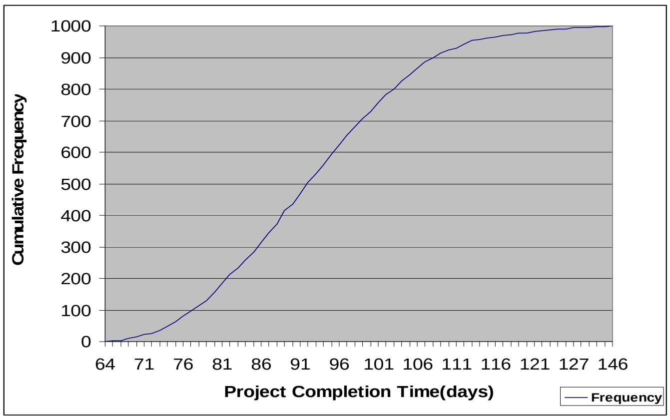

Figure 4. Simulated Cumulative Frequency Distribution Chart

A diagram based on the frequency data is generated showing the frequency of total duration values among the 1000 simulations within the specified intervals (bars in the figure).The duration 89 days occurs for the maximum times during simulation experiment

The graphical output for the cumulative frequency for the software project is shown in figure 4. This S-curve shows the cumulative frequency of project completion by a particular date

For example, there is a 50 percent probability that the project will be finished within 92 days of its start. Project completion dates toward the left have higher risks while those toward the right have lower risk.

VI. CONCLUSION

In this worldwide competition, project management is paid increasing attention. Managing risks in software projects has been recognized as a very important process in order to achieve project objectives in terms of time. The motivation for wanting to incorporate simulation into Schedule risk Analysis is clear as finding critical activities with simulated data yields fruitful information such as estimation of software project/activity completion time. Simulation offers the possibility for representing the complexity that is necessary for realistic reasoning about a software project, including the inherent uncertainty. The Schedule Risk Analysis simulator presented is the result of the combination of Risk Management and Project Planning. The framework should replace traditional project planning

The concept of Simulation used in the present work helps in identifying the areas of higher operational risks where more concentration is required to achieve specific targets. It will be an asset to project managers to focus on critical scenarios of the software development activities.

Hence it is concluded that greater the criticality index value associated with an activity, greater is the risk involved, and will provide effective decision making during the development phase and planning schedule estimation. This in turn implies that such risk prone areas of software project need to be treated above par vis-à-vis rest of the process

REFERENCES

[1] David, Joslin. , William, Poole. ," Agent-based simulation for software project planning", Winter Simulation Conference - WSC, pp. 1059-1066, 2005
[2] Karolak, D. W., "Software Engineering Risk Management", IEEE, Washington, DC.1996.
[3] Project Management Institute," A guide to the project management body of knowledge: PMBOK guide", Project Management Institute Inc., Newtown Square, PA, 2004
[4] Oleh Kostetsky," A Simulation Approach for managing Engineering Projects", IEEE, 1986, pp 318-324
[5] Bowers, J. A., Data for project risk analyses, International Journal of Project Management, 1994, vol. 12, no. 1, pp. 9-16.
[6] Charette, R.N., Applications Strategies for Risk Analysis, McGraw-Hill, New York, 1990
[7] Cooper, D. F., and Broadleaf Capital International , Project risk management guidelines: managing risk in large projects and complex procurements, J. Wiley, West Sussex, England; Hoboken, NJ,2005.
[8] Deo, Narsingh, "System Simulation with Digital Computer:, Prentice Hall of India, New Delhi, 2003.