Available online at http://acfa.apeejay.edu
APEEJAY JOURNAL OF COMPUTER SCIENCE AND APPLICATIONS
ISSN:0974-5742(P)
Vol. 2, January 2014

An Influence Diagram Based Approach for Estimating Schedule Overrun In Software Industry

Kawal Jeet[1], Yadvirender Rana[2] and Rajinder Singh Minhas[3]
[1]Department of Computer Science, DAV College, Jalandhar, India
AScT-Elect., RTMgr, ASTTBC, Surrey, BC, Canada
[1]Department of Computer Science, MLUDAV College, Phagwara, India
[1]kawaljeet80@yahoo.com, [2]vsrana@hotmail.com, [3]minhas_rajinder@yahoo.com

ABSTRACT
Delay in delivery of project has always been a major cause of failure of a software development project. The main reason is ignorance about prioritization of risk factors having adverse impact on schedule of a software development project. The five main risk factors are creeping user requirements, requirement instability, use of unnecessary features, inaccurate estimate of resources and use of immature technology in the project. With the help of existing databases and using expert' opinion (collected through survey), impact of these risk factors are modeled. This model has been developed by using K2 algorithmic technique for building Influence Diagram and calculates the schedule overrun. It helps in the management of schedule throughout the life cycle of the software development project.
Keywords: *Influence Disgram, Schedule Overrun, Software Development Project, Schedule Management, Risk Management, K2 Algorithm.*

1. Introduction

Many software development projects (SDP) fails to meet the quality standards, cost and scheduled deadlines. 60% of the projects fail due to the delay in delivery (schedule overrun) of SDP [10]. So, it has to be dealt with priority. Large number of subjective techniques were developed and used in software industry for timely delivery of projects. But due to lack of strong scientific theory and clarity these techniques are no more in use. In order to manage schedule, we have developed a scientific technique named 'Management of Schedule Slippage' (MaSS) which is an enhancement of MaSO [6]. It is based on Risk Management [9] and utilizes Influence Diagram (ID) [14]. In the subsequent subsection, we discuss Risk Management and ID briefly.

1.1 Risk Management

The purpose of risk management is to identify potential managerial and technical problems (called risk factor) before they occur so that actions can be taken to reduce or eliminate the impact of these problems should they occur The risk management process is a continuous process for systematically addressing risk throughout the life cycle of a SDP. This process consists of the following activities [8]:

- Potential problems are identified
- The likelihood and consequences of these risks are understood
- The priority order in which risks should be addressed are established
- Treatment alternatives appropriate for each potential problem above its risk threshold are recommended.
- Appropriate treatments are selected for risks above their thresholds.
- The effectiveness of each treatment is monitored
- Information is captured to improve risk management policies
- The risk management process and procedures are regularly evaluated and improved

The focus of this research is risk prioritization. It helps in reducing the number of failed SDP by early identification of high risk elements by using ID.

1.2 Influence Diagram

An influence diagram (ID) is an enhancement of Bayesian Network (BN). They are simple visual representation of a decision problem. They offer an intuitive way to identify and display the essential elements, including decisions, uncertainties, and objectives, and how they influence each other. They provide a clear, graphical picture of a problem and helps in showing important relationships. The modeling is based on probabilistic theory. They use circle/oval called nodes and arrows called arcs [14]. Nodes represent system variables while arcs represent influences between variables. In case of ID, the desirability of different event combinations involved in the network is called utility node and is represented by a diamond shape.

2. MOTIVATION

Since the early 80's, IDs have been used in a wide variety of applications like cyber crime detection [1], management of training of the staff involved in development of a software project [7] etc.

With respect to the present research area, many opaque and empirical studies have been done for the management of schedule of a SDP [5]. The authors of [11] describe the way to compress the schedule of a SDP. These studies provide illuminating insights into management of schedule but are weak in explaining its true impact. Moreover, the uncertainty in occurrence of these risk factors is also not taken into consideration.

In this paper we identify the risk factors having an influence on schedule of a SDP. By using ID constructed by using these risk factors, we calculate the exact slippage or delay in schedule.

3. DESIGN OF MASS

According to [12], and based on extensive interviews we have conducted with 45 software professionals, it has been identified that the following risk factors have more adverse impact on schedule of a SDP than others. Brief descriptions of these factors are as follows:

- Creeping User Requirement: User keeps on posing the requirements about the project throughout the system development [5].
- Unnecessary features: Adding more functionality/ features than actually required [5].
- Requirement Instability/volatility: User keeps on changing the statement of requirements leading to confusion among developers [16].
- Inaccurate Estimate of Resources: Requirement of resources like amount of hardware, software or persons etc. are inaccurately estimated [13]. As an example if number of persons required is estimated inaccurately to be 20 but actual requirements is 30 then ultimately due to shortage of staff there will be schedule slippage.
- Immature Technology: Technology used to build software is very new [13]. As compared to mature technology, working with immature technology takes more time.

The aim of designing MaSS using ID is to calculate the delay (in months) in the delivery of the project due to the risk factors mentioned above. These risk factors and schedule overrun are system variables and are presented as chance nodes as shown in Fig. 1. 'Schedule Overrun' is dependent on the probability of occurrence of these risk factors. So arcs are shown to indicate this influence. Information in 'Schedule Overrun' is maintained in the form of conditional probability table (CPT) which is a table maintaining probability of occurrence of the node conditioned on probability of parent nodes. Output of 'Schedule Overrun' node is in the form of probability. So in order to get more usable and understandable information utility node called 'Schedule Slippage' is added as shown in Fig.1.

3.1 USE OF SMILE

SMILE and java has been used to develop MaSS. SMILE is developed by Decision Systems Laboratory (DSL) [15]. It provides platform independent library built in C++ classes for reasoning in BNs and IDs. It helps in graphically building probabilistic models. SMILE libraries can be accessed from within Java applications by using jSMILE (a Java Native Interface (JNI) library). The interface for MaSS is developed by using NETBEANS (an IDE for java) [15].

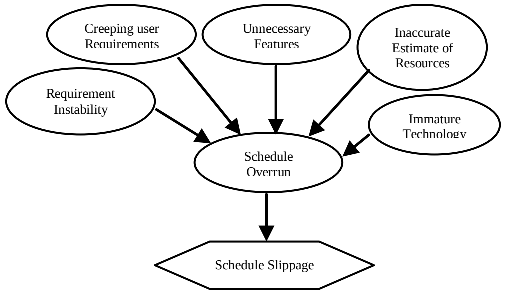

Figure 1. Basic ID with probability tables to calculate Schedule Overrun

3.2 MEASUREMENT SCALE FOR INFLUENCE

For each node (of ID) of MaSS, a measurement scale i.e. a categorization (such as frequent, probable etc.) of possible outcomes is required. This scale is required to be as close as possible to the way the management conducts assessment in that organization, so that MaSS fits well into the organization. With the help of experts, following five categories (called possible outcomes) have been identified for the nodes discussed above.

- Frequent: If the risk factor occurs very often.
- Probable: If it occurs less frequently.
- Occasional: If it occurs at a normal frequency.
- Remote: If it occurs less.
- Improbable: If the risk occurs rarely.

Since all the risk factors do not have the same effect on schedule overrun, so severity level of each risk factor has to be identified. These impact values are to developed based on experience and past historic data.
The measurement scale for severity of risk factor is developed by extensive interviews form 45 software engineers. According to these interviews and with reference to [9], the default scale for the above is found to be:

- Catastrophic: If the risk factor has very severe impact/loss.
- Critical: If loss is lesser severe.
- Serious: If loss due to the risk factor is normal.
- Minor: If consequence or loss is less.
- Negligible: If consequence or loss is least.

4. MASS AUTOMATION

By using NETBEANS (for IDE), jSMILE (for ID) and SMILE wrappers (for dealing with databases), MaSS model has been developed with the following functionalities.

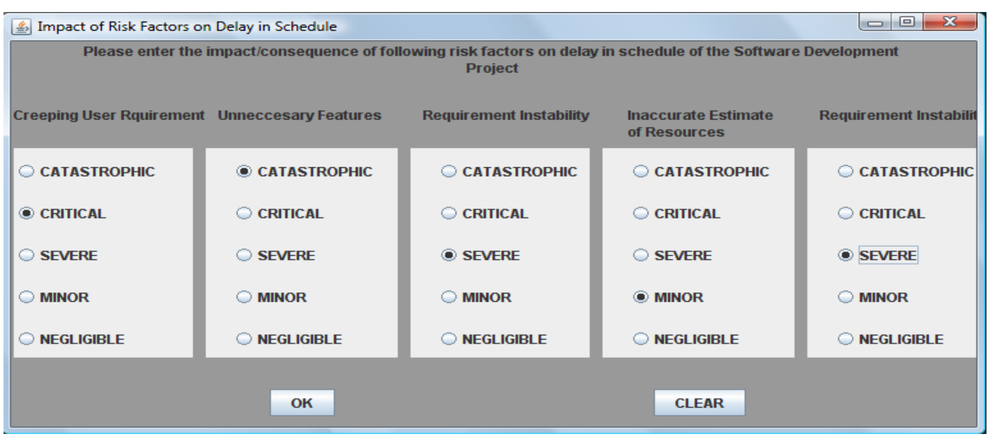

Figure 2. Interface to input impact of risk factors on Delay in Schedule

Project managers (users) can enter the impacts of all five risk factors responsible for delay in the delivery of project, as shown in Fig. 2. After accepting inputs from the user, MaSS asks for another set of inputs as shown in Fig. 3, The user can set the evidence (i.e., probability of occurrence of the risk factors in the project) using this interface. MaSS generates the ID (.xdsl file) in the background, and schedule slippage is displayed as shown in Fig. 4.

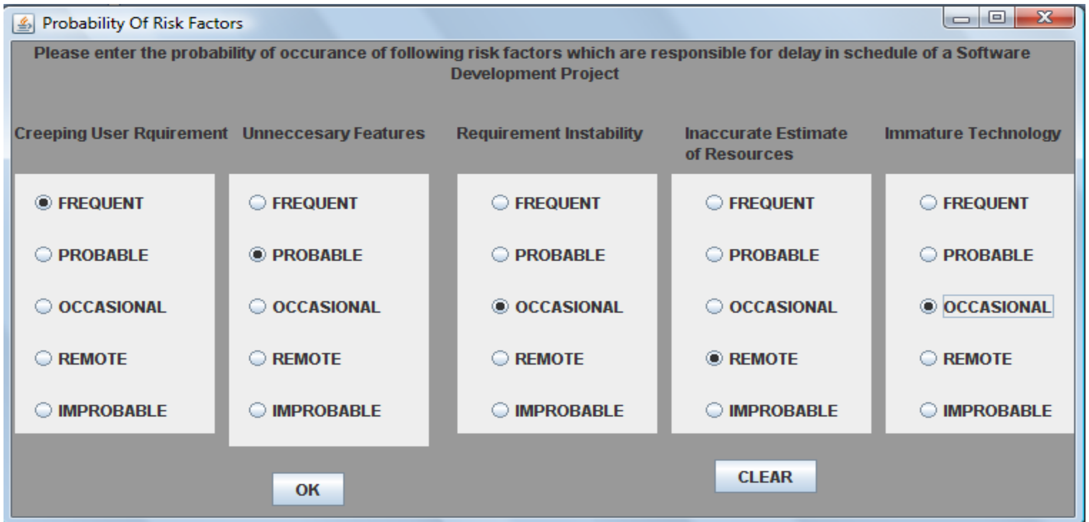

Figure 3. Interface to set the evidences/probabilities for risk factors

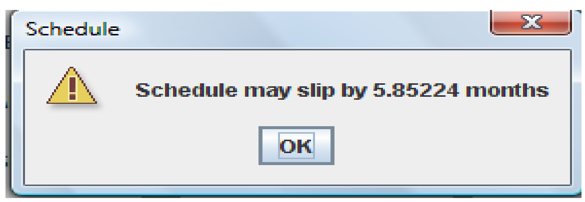

Figure 4. Schedule Slippage in months

4.1 Generating Conditional Property Tables

In order to develop ID, probability tables are required to be populated for all of its chance nodes. Probability of occurrence of all possible outcomes of nodes without any parent nodes (Creeping User Requirements, Requirement Instability, Immature Technology, Reliance on few key persons, Unnecessary Features) is equally likely. But for nodes having parent nodes conditional probability tables (CPT) are required to be generated. To generate CPT for node 'Schedule Overrun' of ID, first of all impacts of risk factors entered as shown in Fig. 2 are normalized. This means relative strength of influence of each parent node on child node is calculated [2]. This can be done using function normalized as discussed below.

Function normalize
Input: *Impact of each risk factor involved i1,i2,…,in*
Output: *Normalized weights of given risk factors w1, w2…wn*
For each of the risks involved, ti is assigned value on the basis of the severity of its impact.
t_i*=5 if impact is catastrophic*
t_i *=4 if impact is critical*
t_i *=3 if impact is severe*
t_i *=2 if impact is minor*
t_i *=1 if impact is negligible*
where i=1,2,3
Relative weight of risk factor (i) $w_i = t_i / \sum t_i$ ………. (1)
end function

For all the risk factors i=1, 2,…n, the relative weights w_i comes out to be in the range [0,1]. Sum of relative weights of all the risk factors comes out to be 1.

$$0 \le w_i \le 1$$
$$w_1 + w_2 + \dots\dots\dots + w_n = 1 \qquad \dots\dots\dots\dots (2)$$

For instance if the impact of various risk factors on delay in schedule is as entered in Fig. 2 then relative or normalize weights are calculated as discussed below.

For 'Creeping User Requirements' the impact is entered to be 'Critical.

So as discussed in *function normalize* t1 = 4.

Similarly, for 'Unnecessary features' the impact is entered to be 'catastrophic'. So t2=5.

For 'Requirement Instability' the impact is entered to be 'Severe'. So t3=3.

For 'Reliance on Few Fey Persons' the impact is entered to be 'Minor'. So t4=2.

For 'Immature Technology' the impact is entered to be 'Severe'. So t5=3.

Corresponding relative weights calculated by using *function normalize* is

w1=t1/ (t1+t2+t3+t4+t5) =4/ (4+5+3+2+3) =4/17
w2=t2/ (t1+t2+t3+t4+t5) =5/ (4+5+3+2+3) =5/17
w3=t3/(t1+t2+t3+t4+t5)= 3/(4+5+3+2+3)=3/17 (3)
w4=t3/(t1+t2+t3+t4+t5)= 2/(4+5+3+2+3)=2/17
w5=t3/(t1+t2+t3+t4+t5)= 3/(4+5+3+2+3)=3/17

After getting normalized weights we can countercheck the weights by calculating $\sum_{i=1}^{n} wi$ which comes out to be w1+w2+w3+w4+w5 (calculated above) = 4/17+5/17+3/17+2/17+3/17=17/17=1.

In order to calculate the joint CPT for 'Schedule Overrun' with respect to all the risk factors occurring simultaneously [4], temporary ID's and hence CPT's are to be constructed for each individual relationship (indicated in Fig. 5) by using K2 technique. K2 is an algorithm for constructing a BN from a database of records. For the current ID, this database is built by past historic data and expert interviews. Table I is one such sample database. Each row of this table indicated the probability of occurrence of all the risk factors and 'Seclude Overrun'. Fig. 5 indicates these individually.

So, K2 algorithmic technique [3] could be used to calculate conditional probability table (CPT) for 'Schedule Overrun'. CPT developed for: 'Creeping User Requirements'---'Schedule Overrun', 'Unnecessary Features----'Schedule Overrun', 'Requirement Instability'---'Schedule Overrun', 'Inaccurate Estimate of Resources'---'Schedule Overrun' and 'Immature Technology'---'Schedule Overrun' is shown in Table II-VI.

TABLE I. DATABASE MAINTAINING PROBABILITY OF OCCURRENCE OF RISK FACTORS

Schedule Overrun	**Requirement Instability**	**Unnecessary Features**	**Creeping User Requirements**	**Inaccurate Estimate of Resources**	**Immature Technology**
Frequent	Remote	Occasional	Remote	Improbable	Frequent
Probable	Remote	Remote	Probable	Remote	Improbable
Occasional	Improbable	Improbable	Occasional	Improbable	Occasional
Remote	Improbable	Improbable	Improbable	Occasional	Improbable
Improbable	Remote	Remote	Remote	Remote	Remote
Remote	Occasional	Probable	Probable	Frequent	Occasional
Improbable	Probable	Frequent	Remote	Occasional	Frequent
Occasional	Probable	Probable	Probable	Probable	Probable
Remote	Frequent	Occasional	Frequent	Occasional	Occasional

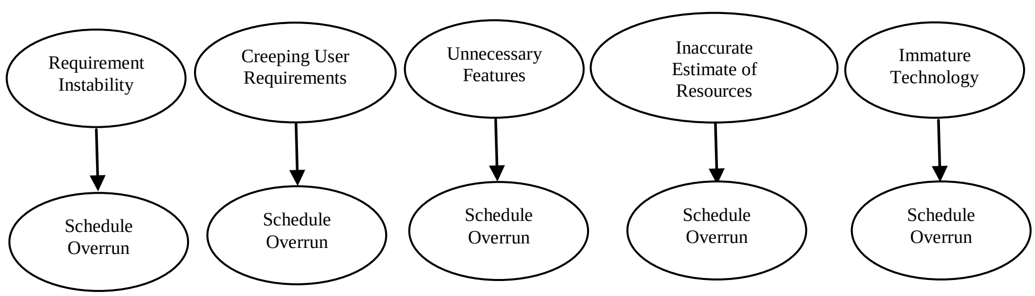

Figure 5. Temporary Bayesian networks for each risk factor and *Schedule Overrun*

TABLE II. CPT FOR SCHEDULE OVERRUN DEPENDING ON *CREEPING USER REQUIREMENT*

Creeping Requirement	**Frequent**	**Probable**	**Occasional**	**Remote**	**Improbable**
Frequent	0.166667	0.125	0.166667	0.25	0.285714
Probable	0.166667	0.25	0.166667	0.125	0.142857
Occasional	**0.166667**	0.25	0.333333	0.125	0.142857
Remote	0.333333	0.25	0.166667	0.125	0.285714
Improbable	0.166667	0.125	0.166667	0.375	0.142857

TABLE III. CPT FOR SCHEDULE OVERRUN DEPENDING ON *UNNECESSARY FEATURES*

Unnecessary Features	Frequent	Probable	Occasional	Remote	Improbable
Frequent	0.166667	0.142857	0.285714	0.142857	0.25
Probable	0.166667	0.142857	0.142857	0.285714	0.125
Occasional	0.166667	**0.285714**	0.142857	0.142857	0.25
Remote	0.166667	0.285714	0.285714	0.142857	0.25
Improbable	0.333333	0.142857	0.142857	0.285714	0.125

TABLE IV. CPT SCHEDULE OVERRUN DEPENDING ON *REQUIREMENT INSTABILITY*

Requirement Instability	Frequent	Probable	Occasional	Remote	Improbable
Frequent	0.166667	0.142857	0.285714	0.25	0.142857
Probable	0.166667	0.142857	0.142857	0.25	0.142857
Occasional	0.166667	0.285714	**0.142857**	0.125	0.285714
Remote	0.333333	0.142857	0.285714	0.125	0.285714
Improbable	0.166667	0.285714	0.142857	0.25	0.142857

TABLE V. CPT SCHEDULE OVERRUN DEPENDING ON *INACCURATE ESTIMATE OF RESOURCES*

Inaccurate Estimate of Resources	Frequent	Probable	Occasional	Remote	Improbable
Frequent	0.125	0.166667	0.142857	0.125	0.333333
Probable	0.25	0.333333	0.142857	0.25	0.166667
Occasional	0.125	0.166667	0.285714	**0.25**	0.166667
Remote	0.25	0.166667	0.142857	0.25	0.166667
Improbable	0.25	0.166667	0.285714	0.125	0.166667

TABLE VI. CPT SCHEDULE OVERRUN DEPENDING ON *IMMATURE TECHNOLOGY*

Immature Technology	Frequent	Probable	Occasional	Remote	Improbable
Frequent	0.166667	0.125	0.142857	0.142857	0.285714
Probable	0.333333	0.25	0.142857	0.285714	0.142857
Occasional	0.166667	0.125	**0.285714**	0.285714	0.142857
Remote	0.166667	0.25	0.142857	0.142857	0.285714
Improbable	0.166667	0.25	0.285714	0.142857	0.142857

TABLE VII. CALCULATION OF SINGLE CONDITIONAL PROBABILITY FOR *SCHEDULE OVERRUN* WITH OUTCOME TO BE OCCASIONAL AND IMPACTS AND PROBABILITIES AS ENTERED IN FIG. 2 AND FIG. 3

Risk Factor	Probability	Outcome (Value)	Normalized Weights	Contribution of Risk Factor (Value X Impact)
Creeping Requirement	Frequent	0.166667 (Table. II)	4/17	0.0392157647058823529411 76
Unnecessary Features	Probable	0.285714 (Table. III)	5/17	0.0840335294117647058823 52
Requirement Instability	Occasional	0.142857 (Table. IV)	3/17	0.0252100588235294117647 05
Inaccurate Estimate of Resources	Remote	0.25 (Table. V)	2/17	0.0294117647058823529411 76
Immature Technology	Occasional	0.285714 (Table. VI)	3/17	0.0504201176470588235294 11

TABLE VIII. PARTIAL CPT FOR SCHEDULE OVERRUN

Creeping User Requirements	Frequent									
Unnecessary Features	Probable									
Requirement Instability	Occasional									
Inaccurate Estimate of Resources	Occasional					Remote				
Immature Technology	Frequent	Probable	Occasional	Remote	Improbable	Frequent	Probable	Occasional	Remote	Improbable
Frequent	0.177871	0.170518	0.173669	0.173669	0.19888	0.17577	0.168417	0.171569	0.171569	0.196779
Probable	0.182073	0.167367	0.148459	0.173669	0.148459	0.194678	0.179972	0.161064	0.186275	0.161064
Occasional	0.211485	0.204132	0.232493	0.232493	0.207283	0.207283	0.19993	**0.228291**	0.228291	0.203081
Remote	0.259104	0.27381	0.254902	0.254902	0.280112	0.271709	0.286415	0.267507	0.267507	0.292717
Improbable	0.169468	0.184174	0.190476	0.165266	0.165266	0.15056	0.165266	0.171569	0.146359	0.146359

CPT for ‘Schedule Overrun’ can be calculated by using following formula. According to [2],

$$p(x^q \mid y1^{q1}, y2^{q2}, y3^{q3}, y4^{q4}, y5^{q5}) = \sum_{j=1}^{3} w^j p(x^q \mid y_j^{qj})$$

Where q, q_j = {Frequent, Probable, Occasional, Remote and Improbable}.
j=1,2,3,4,5
x=Schedule Overrun
y1=Creeping User Requirement
y2=Requirement Instability
y3=Unnecessary Features
y4= Inaccurate Estimate of Resources
y5= Immature Technology

wj is normalized impact of risk factor j as obtained by function normalize discussed above.

For instance,

Input: Impacts of risk factors as entered in Fig. 2.
Probability of occurrence of risk factors as entered in Fig. 3.
Output: Probability of occurrence of ‘Schedule Overrun’ to be Occasional.

From table II, it is obtained that if probability of occurrence of ‘Creeping user requirement’ is ‘Frequent’ then probability of occurrence of ‘Schedule Overrun’ to be occasional is 0.166667.

From III, it is obtained that if probability of occurrence of ‘Unnecessary Features’ is ‘Probable then probability of occurrence of ‘Schedule Overrun’ to be occasional is 0.285714.

From table IV, it is obtained that if probability of occurrence of 'Requirement Instability' is 'Occasional' then probability of occurrence of 'Schedule Overrun' to be occasional is 0.142857.
From table V, it is obtained that if probability of occurrence of 'Inaccurate Estimate of Resources' is 'Remote' then probability of occurrence of 'Schedule Overrun' to be occasional is 0.25.
From table VI, it is obtained that if probability of occurrence of 'Immature Technology' is 'Occasional' then probability of occurrence of 'Schedule Overrun' to be occasional is 0.285714.

As mentioned in above the normalized weights w1, w2, w3, w4 and w5 will be 4/17,5/17, 3/17, 2/17,3/17.So by using (4)

$$p(x^{Occasional} \mid y1^{Frequent}, y2^{Probable}, y3^{Occasional}, y4^{Remote}, y5^{Occasional}) =$$

4/17*0.166667+5/17*0.285714+3/17*0.142857+2/17*0.25+3/17*0.285714=0.039215764705882352941176470588235+0.084033529411764705882352941176471+0.025210058823529411764705882352941+0.029411764705882352941176470588235+0.050420117647058823529411764705882 = 0.228291 as shown in Table VII. The partial CPT thus developed (using the above method) is shown in table VIII.

5. Evaluation of Result

MaSS uses the above built ID and calculates the actual slippage in schedule (in months) due to risk factors mentioned above. For instance, if probability of occurrence of risk factors is as entered in Fig. 3 then 'Schedule Overrun' for each possible outcome as obtained from its CPT populated above and is shown in table IX.

TABLE IX. SCHEDULE OVERRUN FOR EACH POSSIBLE OUTCOME ON THE BASIS OF ENTRIES IN FIG. 3

Probability of occurrence	Values
Frequent	0.171569
Probable	0.161064
Occasional	0.228291
Improbable	0.267507
Remote	0.171569

Table X. UTILITY TABLE FOR SCHEDULE SLIPPAGE

Schedule Overrun	**Frequent**	**Probable**	**Occasional**	**Remote**	**Improbable**
Value	12	9	6	3	1

A utility table associated with node 'Schedule Slippage' holds the slippage in schedule in months for each possible outcome of 'Schedule Overrun'. This table could be populated with values obtained by expert interviews.
For instance if the utility table is as shown in Table X then it means if schedule overrun happens to be 'Frequent' then 'Schedule Slippage' will be by 12 months. For the case discussed above, the slippage in schedule or delay in project schedule will be 0.171569*12+0.161064*9+0.228291*6+0.267507*3+0.171569*1 = 2.058828+1.449536+1.369746 +0.802521+0.171569= 5.85224 months as shown in Fig. 4

6. Conclusion

This paper outlines an enhancement of MaSO (Jeet 2010). The model discussed in this paper integrated two more risk factors to MaSO. Like MaSO, MaSS provides the following contributions for modeling the management process:
It provides a graphical view of the problem and makes it possible for experts and decision makers to discuss interdependencies of events, without using any formal mathematical, probabilistic or statistical notations. It reduces large volumes of data required in the management process. It helps in detecting delay in delivery of project (in months). On the basis of impacts and probability of occurrence of five risk factors, more accurate estimates of delay in delivery of project could be found.

Acknowledgement
We wish to acknowledge the efforts of Mr. Harjit Singh, Senior Consultant, Tata Consultancy Services, Delhi, India; Kawalpreet Kaur, System Administrator, Oracle India Pvt. Ltd., Noida, India.

References:

[1] Abouzakhar, N.S., Gani, A., Manson, G., Abuitbel, M., King, D. (2003): Bayesian Learning Networks Approach To Cybercrime Detection, Postgraduate Net-Working Conference.

[2] Balram Das. (2004): Generating Conditional Probabilities For Bayesian Networks: Easing The Knowledge Acquisition Problem.

[3] Cooper G.F. (1999): A Bayesian Method For The Induction Of Probabilistic Networks From Data, Machine Learning, Vol. 9:309-347.

[4] Dimitri P. Bertsekas, John N. Tsitsiklis. (2002): Introduction To Probability, Athena Scientific.

[5] Houstan, D. X., Mackulak, G.T., Collofello, J.S. (2001): Stochastic Simulation Of Risk Factor Potential Effects For Software Development Risk Management, Journal Of Systems And Software: 247-257.

[6] Jeet, K., Dhir, R., Mago, V.K., Minhas, R.S. (2010): Maso: A Tool For Aiding The Management Of Schedule Overrun, Ieee 2nd International Advance Computing Conference 2010:416-421.

[7] Jeet, K., Mago, V.K., Prasad, B., Minhas, R.S. (2008): Mast: A Tool For Aiding The Staff Training Management By Using Influence Diagrams. Setp: 156-164.

[8] Karolak, D.W. (1995): Software Engineering Risk Management, Wiley-Ieee Computer Society.

[9] Mcmanus. J. (2004): Risk Management In Software Development Projects, 3-20, Butterworth-Heinemann.

[10] Molokken, K., Jorgensen, M. (2003). A Review Of Software Surveys On Software Effort Estimation. Ieee, Empirical Software Engineering. Isese 2003: 223 – 230.

[11] Park, K.S., Kim. H. (1998): Artificial Intelligence Approaches To Determination Of Cnc Machining Parameters In Manufacturing: A Review. Artificial Intelligence In Engineering: 127-134.

[12] Ramesh, B.M., Jarke. (2010): Toward Reference Models For Requirements Traceability, Ieee Transactions On Software Engineering: 58-93.

[13] Verner, J.M. (2008): Quality Software Development: What Do We Need To Improve In The Software Development Process? , International Conference On Software Engineering International Workshop On Software Quality: 1-2.

[14] Www.Genie.Sis.Pitt.Edu Accessed On 27th Feb., 2010.

[15] Www.Sun.Java.Com Accessed On 27th Feb, 2010.

[16] Zowghi, D., Nurmuliani, N. (2002): A Study Of The Impact Of Requirements Volatility On Software Project Performance, Ieee, Software Engineering Conference. Ninth Asia-Pacific: 3-11.

[17] Singh, G., & Singh, S. Medical Images Edge Detection Using Mathematical Based Morphology. .*",Apeejay Journal Of Computer Science And Applications"*, Vol. 1, 2013, Pp. 1-4.