Available online at http://acfa.apeejay.edu
APEEJAY JOURNAL OF COMPUTER SCIENCE AND APPLICATIONS
ISSN:0974-5742(P)
Vol. 3, January 2015

An Influence Diagram Based Approach -for Estimating Schedule overrun in Software Industry

Kawal Jeet[1], Yadvirender Rana[2], Ruichi Xin[3]
[1]*Department of Computer Science, D.A.V College, Jalandhar, India,*
[2]*AScT-Elect., RTMgr , ASTTBC , Surrey, BC, Canada*
[3]*P.Eng, APEGBC, Burnaby, BC, Canada*
kawaljeet80@yahoo.com[1], ysrana@hotmail.com[2], ray.rc.xin@gmail.com[3]

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. This paper aims to handle this with the help of Influence Diagram (ID) based solution. The five main risk factors having adverse impact on schedule 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 (consisting of probability of occurrence of risk factors) and using expert' opinion (collected through survey). Impact of these risk factors are modeled into an ID based system has been constructed that calculates schedule overrun.
Keywords: *Influence Diagram, Schedule Overrun, Schedule Slippage, Software Development, Schedule Management, Risk Management, Bayesian Network, K2 Algorithm.*

1. Introduction

Many software development projects (SDP) fails to meet the quality standards, cost and scheduled deadlines. According to Ramesh et al. (2001), 60% of the projects fail due to this delay or schedule overrun. So it is to deal with priority. Large number of subjective techniques were developed and used in software industry to deal with 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 timely delivery, we have developed a scientific technique named 'Management of Schedule Slippage' (MaSS) which is an enhancement of MaSO (Vijay et al. 2010). It is based on Risk Management (Karolak 1995) and utilizes Influence Diagram (ID) (Osmundsona 2003). In the subsequent subsection, we discuss Risk Management and ID briefly.

1.1 Risk Management

Software risk management is a key discipline for making effective decisions and communicating the results within software organizations. The purpose of risk management is to identify potential managerial and technical problems before they occur so that actions can be taken that reduce or eliminate the likelihood or 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 software development project. This process consists of the following activities (Verner):

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

The focus of this research is risk prioritization which is a sub activity of risk analysis phases. It helps in reducing the number of failed SDP by early identification of high risk elements and ID has been used for the same.

1.2 Influence Diagram

An influence diagram is a simple visual representation of a decision problem. Influence diagrams offer an intuitive way to identify and display the essential elements, including decisions, uncertainties, and objectives, and how they influence each other. It provide a clear, graphical picture of a problem and helps in showing important relationships and relevance It provide a means to compare alternatives (Karolak 1995).The modeling is based on probabilistic theory. According to (Park et al. 1998), Influence diagrams use shapes called nodes and arrows called arcs, which enable the diagram to function as a graphical representation of a system. Nodes represent system variables while arcs represent influences between variables.the network contains circles or ovals called chance nodes which represents a discrete random variable, diamond called utility node which

represents the desirability of different event combinations involved in the network. ID is actually an enhancement of Bayesian Networks BN which is a probabilistic network used for reasoning under uncertainty but can be constructed using chance nodes only.
Normally, an arc in an ID denotes an influence. For instance if there is an arc from M to O it means the node at the tail (M or N in Fig. 1) of the arc influences the value (or the probability distribution over the possible values) of the node at the head (O in Fig. 1) of the arc.

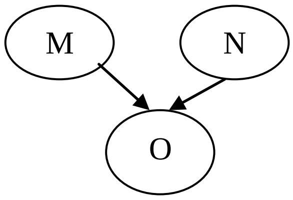

Fig. 1 Cause-effect relationship

According to Baye's Theorem (jaffery et al. 1990)

$$P(O|M) = \frac{P(M|O).P(O)}{P(M)} \quad (1)$$

In this formula O is the hypothesis required to be tested. M is the evidence that confirms or disconfirms the hypothesis.

2. Motivation

Since the early 80's, IDs have been used in a wide variety of applications like cyber crime detection (Jeffery 1990), management of training of the staff involved in development of a software project (Abouzakhar 2003), etc.
With respect to the present research area, many opaque and empirical studies have been done for the management of schedule of a SDP (Boehma et al. 2004 and Bhanu et al. 2008). The authors of (Houstan et al.2001 and Abdel-Hamid 1990) describe the way to compress the schedule of a SDP. These studies provided 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, we can calculate the exact slippage or delay in schedule.

2.1 Design of MaSS

Schedule slippage 'always' occurred in 1%, 'usually' in 31%, 'sometimes' in 50%, 'rarely' in 15%, and, 'never' in 2% of the SDP (Browning 2002). According to (Genuchten 1991), 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 (Houstan 2001).
- Unnecessary features: Adding more functionality/ features than actually required (Abdul-Hamid 1990).
- Requirement Instability/volatility: User keeps on changing the statement of requirements leading to confusion among developers (Browing 2002).
- Inaccurate Estimate of Resources: Requirement of resources like amount of hardware, software or persons etc. are inaccurately estimated leading to have an impact on timely delivery of resources (Verner 2008).
- Immature Technology: Technology used to build software is very new and not very much used (Verner 2008).

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 are presented as chance nodes as shown in Fig. 2. 'Schedule Overrun' also represented as chance node as it is dependent on the probability of occurrence of these risk factors. Slippage in schedule is driven by the parent node 'Schedule Overrun' and is represented by diamond shaped node.

2.2 Use of SMILE

SMILE and java has been used to develop MaSS. SMILE is developed by Decision Systems Laboratory (DSL) [10]. 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) (www.sun.java.com).

2.3 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 in the beginning of section 3.

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 rarely occurs in the present SDP.

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. In order to suit the given organization, people, environment and risk factor these impact values are to be 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 McManus (2004), 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.

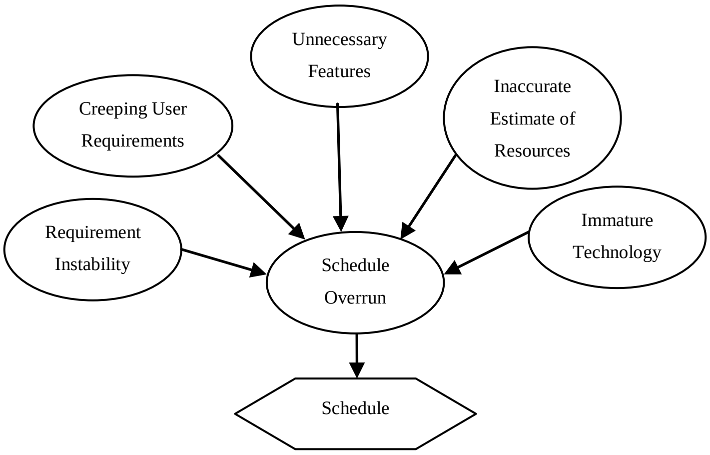

Fig. 2 Basic ID with probability tables to calculate Schedule Overrun

2.4 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.

Project managers (users) can enter the impacts of all five risk factors responsible for delay in the delivery of project, as shown in Fig. 3.

As shown in Fig. 4, after accepting inputs from the user, MaSS asks for another set of inputs. 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. 6.

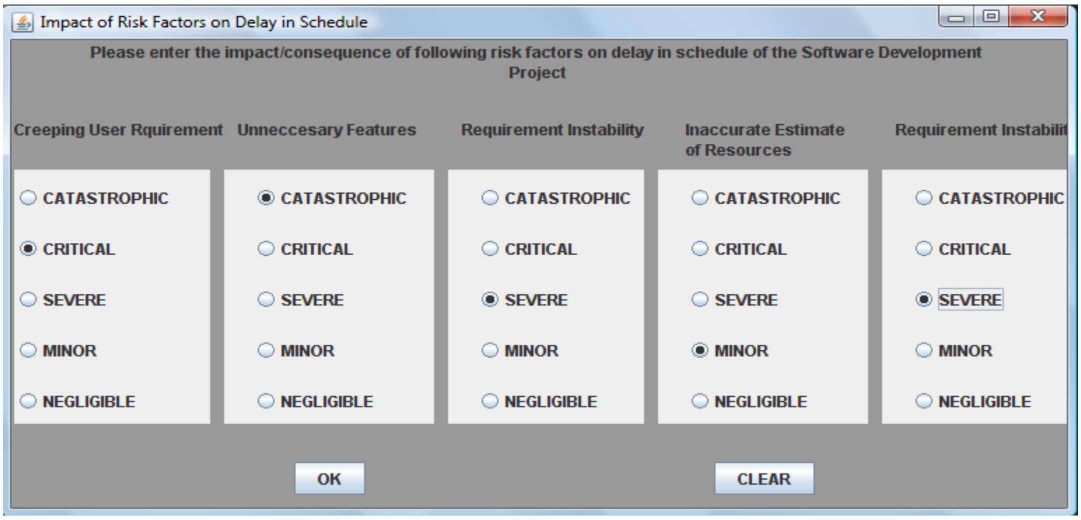

Fig. 3. Interface to input impact of risk factors on Delay in Schedule

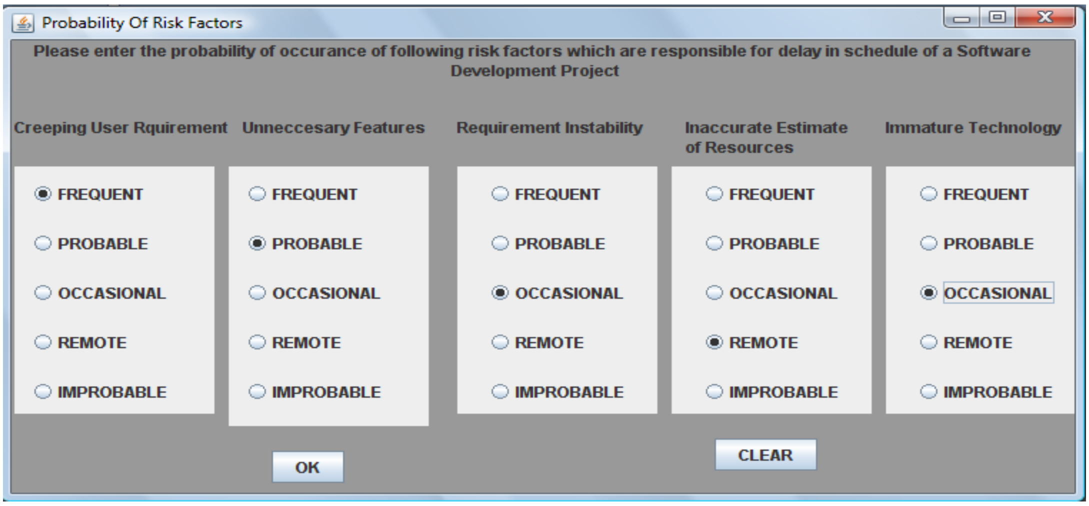

Fig. 4. Interface to set the evidences/probabilities for risk factors

2.5 Generating Conditional Property Tables

In order to generate CPT for node 'Schedule Overrun' of ID, impacts of risk factors entered as shown in Fig. 3 are normalized. This means their relative strength of influence on child node is calculated (Balram Das 2004). This can be done using function normalized as discussed below.

Function normalize is

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.

ti=5 if impact is catastrophic

ti =4 if impact is critical

ti =3 if impact is severe

ti =2 if impact is minor

ti =1 if impact is negligible

where i=1,2,3

Relative weight of risk factor (i) $w_i = t_i / \sum t_i$

end function

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

$$0 \le w_i \le 1$$

$$w_1 + w_2 + \dots\dots\dots + w_n = 1 \quad (2)$$

For instance if the impact of various risk factors on delay in schedule is as entered in Fig. 3 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 t4=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 $\sum_{i=1}^{3} w_i$ which comes out to be w1+w2+w3 (calculated above)= 4/17+5/17+3/17+2/17+3/17=17/17=1 as discussed in (2).
K2 algorithmic technique (Cooper 1999) could be used to CPT for 'Schedule Overrun'. 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 1 is one such sample database. Each row of this table indicated the probability of occurrence of all the risk factors and 'Seclude Overrun' in a particular case. Fig. 5 indicates these individually.

Table 1 Database maintaining probability of occurrence of risk factors

Schedule Overrun	**Requirement Instability**	**Unnecessary Features**	**Creeping User Req.**	**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
Frequent	Occasional	Improbable	Improbable	Improbable	Frequent

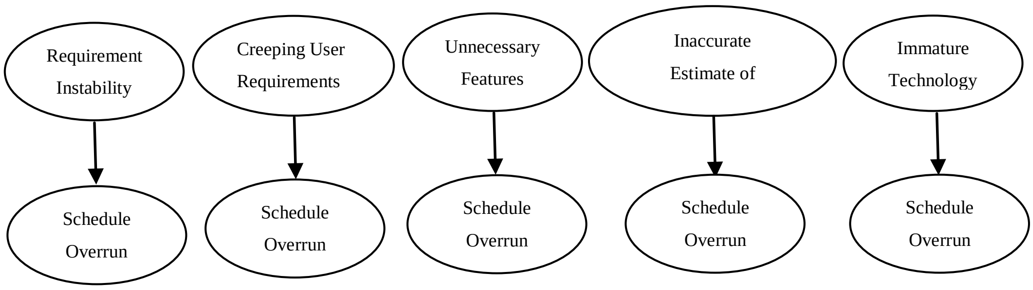

Fig. 5 Temporary Bayesian networks for each risk factor and Schedule Overrun

In order to calculate the joint CPT for 'Schedule Overrun' with respect to all the risk factors occurring simultaneously (Dimitri 2002), temporarily ID's and hence CPT's are to be constructed for each individual relationship (indicated in Fig. 5) by using K2 technique.

CPT is developed for: 'Creeping User Requirements'---'Schedule Overrun', 'Unnecessary Features----'Schedule Overrun','Requirement Instability'---'Schedule Overrun', 'Inaccurate Estimate of Resources'---'Schedule Overrun' and 'ImmatureTechnology'---'Schedule Overrun' as shown in Table 2-4. Let c1, c2, c3,c4,c5 denotes matrices holding the CPT (as shown in Table 2, 3,4,5,6) of child nodes from each of the temporary ID's built as mentioned above. For instance from temporary BN containing 'Immature Technology' and 'Schedule Overrun' we can get CPT c5 as p(Schedule_Overrun | Immature_ Technology) as discussed in section 1.2. Similarly c2, c3,c4,c5 can be calculated.
CPT for 'Schedule Overrun' can be calculated by using following formula. According to (Balram Das 2004),

$$p(x^q \mid y1^{q1}, y2^{q2}, y3^{q3}, y4^{q4}, y5^{q5}) = \sum_{j=1}^{3} w^j p(x^q \mid y_j{}^{qj}) \quad (4)$$

where q,qj = {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 in section 3.3.
This formula indicates joint probability of possible outcome of factor x to be q provided y1,y2,y3,y4,y5 has possible occurrence to be q1,q2,q3,q4,q5 respectively is equal to summation of multiplication of each risk factor's normalized weight and probability of occurance of x with outcome q with respect to risk factor and its outcome.
For instance,
Input: Impacts of risk factors as entered in Fig. 3.
Probability of occurrence of these risk factors as entered in Fig. 4.
Output: Probability of occur ace of 'Schedule Overrun' to be Occasional.
Impacts entered in Fig. 3 are normalized as 4/17,5/17,3/17,2/17,3/17 as mentioned above.

Table 2 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 3 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 4 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 5 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 6 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

From table 2, 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 table 3, 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 4, 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 5, 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 6, 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 section 3.4 the normalized weights w1, w2 and w3 will be 4/17,5/17, 3/17, 2/17,3/17.So by using (4)

$$p(x^{Occasional} \mid y1^{Frequent}, y2^{\Pr obable}, y3^{Occasional}, y4^{\mathrm{Re}mote}, y5^{Occasional}) =$$

4/17*0.166667+5/17*0.285714+3/17*0.142857+2/17*0.25+3/17*0.285714=0.0392157647058823529411764705882 35+0.084033529411764705882352941176471+0.025210058823529411764705882352941+0.029411764705882352941176470588235+0.050420117647058823529411764705882 = 0.228291 as shown in Table 8.

2.6 Evaluation of Result

MaSS uses the above built ID and calculates the actual slippage in schedule (in months) due to already mentioned risk factors. ‘Schedule Slippage’ is a utility node which represents the desirability of different event combinations involved in the network. It is driven by ‘Schedule Overrun’, as shown in Fig. 2 and as discussed in section 3.

For instance, if probability of occurrence of risk factors is as entered in Fig. 4 then ‘Schedule Overrun’ for each possible outcome as obtained from its CPT populated in section 3.4 and are shown in Table 7 is

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

A utility table is associated with this node which holds the schedule slippage 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 a s shown in Table 9 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= 5.85224months as shown in Fig. 6

Table 7 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

Table 8 Calculation of single conditional probability for Schedule Overrun with outcome to be occasional and impacts and probabilities as entered in Fig. 3 and Fig. 4

Risk Factor	**Probability**	**Outcome (Value)**	**Normalized Weights**	**Contribution of Risk Factor (Value Impact)**
Creeping Requirement	Frequent	166667 (Table. 2)	4/17	0.039215764705882352941176
Unnecessary Features	Probable	0.285714 (Table. 3)	5/17	0.084033529411764705882352
Requirement Instability	Occasional	0.142857 (Table. 4)	3/17	0.025210058823529411764705
Inaccurate Estimate of Resources	Remote	0.25 (Table. 5)	2/17	0.029411764705882352941176
Immature Technology	Occasional	0.285714 (Table. 6)	3/17	0.050420117647058823529411

Table 9 Utility table for schedule slippage

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

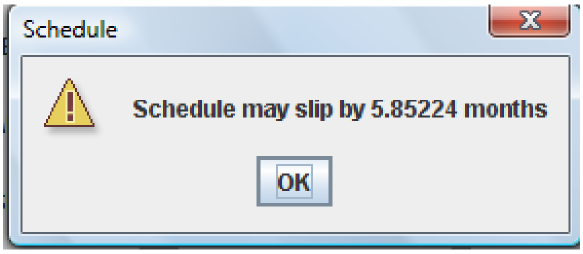

Fig. 6. Schedule Slippage in months

Conclusion:

This paper outlines an enhancement of MaSO. The model discussed in this paper integrated two more risk factors to MaSO which are also found to have an adverse impact on schedule of a project. On the basis of impacts and probability of occurrence of all these five risk factors, more accurate estimates of delay in delivery of project could be done at any stage of SDP.

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 months (in months).
- Although the response for a live schedule overrun management is encouraging but depending on the organization, some more factors that may have an impact on the schedule of a software development project. As a result, this work is scalable.

References:

[1] B.Ramesh, M,Jarke:Toward Reference Models for Requirements Traceability. IEEE Transactions on Software Engineering. 58-93 (Jan 2010).

[2] Kawal Jeet, R Dhir ,VK Mago and R S Minhas: MaSO:A Tool for Aiding the Management of Schedule Overrun. IEEE 2nd International Advance Computing Conference 2010. 416-421 (2010).

[3] Dale Walter Karolak, Software Engineering Risk Management, Wiley-IEEE Computer Society, 1995.or IEEE Std 1540-2001 IEEE Standard for Software Life Cycle Processes—Risk Management.

[4] J.S. Osmundsona, J.B. Michaelb, Martin ,J. Machniakc, M.A. Grossmand:Quality management metrics for software development. Information and Management .799 – 812 (Sept. 2003).

[5] June M. Verner: 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 (2008).

[6] K S Park and S H Kim:Artificial intelligence approaches to determination of CNC machining parameters in manufacturing: a review. Artificial Intelligence in Engineering.127-134 (1998).

[7] D.R. Jeffery and G. Low: Calibrating estimation tools for software development. Software Engineering Journal IEEE. 215-221 (1990).

[8] Abouzakhar, N.S., Gani, A., Manson, G., Abuitbel, M., King, D. 2003: Bayesian Learning Networks Approach to Cybercrime Detection. Postgraduate Net-working Conference.

[9] B.Boehma, C. Abts, S. Chulani: Software development cost estimation approaches – A survey. Annals of Software Engineering, 177–205(2004).

[10] Kawal Jeet, Vijay Kumar Mago, Bhanu Prasad, Rajinder Singh Minhas: MaST: A Tool for Aiding the Staff Training Management by Using Influence Diagrams. SETP. 156-164(2008).

[11] www.genie.sis.pitt.edu Accessed on 27th Feb., 2008.

[12] www.sun.java.com Accessed on 27th Feb, 2008.

[13] Dan X. Houstan, Gerald T. Mackulak, James S. Collofello:Stoichastic simulation of risk factor potential effects for software development risk management. Journal of Systems and Software. 247-257 (2001).

[14] T.K. Abdel-Hamid:Investigating the cost/schedule trade-off in software development Software. IEEE.97-105 (1990).

[15] T.R. Browning, S.D. Eppinger: Modeling impacts of process architecture on cost and schedule risk in product development. IEEE Transactions on Engineering Management. 428- 442 (2002).

[16] M. Van Genuchten: Why is software late? An empirical study of reasons for delay in software development. IEEE Transactions on Software Engineering. 582-590 (1991).