The current issue and full text archive of this journal is available on Emerald Insight at:
www.emeraldinsight.com/1463-5771.htm

Received 8 July 2013
Revised 9 January 2014
Accepted 9 January 2014

Interpretive structural modelling of critical risk factors in software engineering project

Chitrasen Samantra, Saurav Datta and Siba Sankar Mahapatra
Department of Mechanical Engineering, National Institute of Technology, Rourkela, India, and
Bikash Ranjan Debata
Marketing Department, Rourkela Institute of Management Studies, Rourkela, India

Abstract

Purpose – Success of software projects depends on identification of project risks and managing the risks in a proactive manner. Risk management requires thorough insights into interrelationship of various risk factors for proposing strategies to minimize failure rate. The purpose of this paper is to develop a comprehensive structural model to interrelate important risk factors affecting the success of software projects.

Design/methodology/approach – Specifically, this study reveals how interpretive structural modelling helps the risk managers in identifying and understanding the interrelationship among various risk factors. A total of 23 risk factors (or risk sources) have been identified through an extensive literature review.

Findings – Necessary modelling information has been gathered from expert through a structured questionnaire survey. *Matrice d'Impacts croises-multipication appliqué an classment* analysis has been employed to classify the risk factors into four clusters such as autonomous, dependent, linkage and independent based on their driving and dependence power. Risk factors with strong dependence and weak driving power need urgent attention from managerial perspective.

Originality/value – The proposed model is useful for software managers/practitioners to address risk factors associated with complicated projects.

Keywords Company performance, Decision support systems, Corporate strategy

Paper type Research paper

1. Introduction

Software engineering concerns with design, creation and maintenance of software using latest tools, techniques and practices from computer science, project management, information technology and other application domains (Grimstad, 2006). Since execution of software projects are not always successful, their development is a challenging and important issue in the current scenario. Today, most of the software industries are concerned with failure and escalation of original budget due to delay in project implementation. In 1995, Chaos report of Standish Group reveals that only 16.2 per cent of software projects are completed on time and budget. Over 31.1 per cent of software projects are cancelled before they get completed and 52.7 per cent of the projects are escalated by 198 per cent of their original estimates.

The authors are really grateful for receiving continuous encouragement and wholehearted support provided by Professor Gunasekaran, Editor-in-Chief, *Benchmarking: An International Journal.* The authors also sincerely express the heartiest thanks to the anonymous reviewers for their valuable comments and suggestions to make the paper a good contributor.

Benchmarking: An International Journal
Vol. 23 No. 1, 2016
pp. 2-24
© Emerald Group Publishing Limited
1463-5771
DOI 10.1108/BIJ-07-2013-0071

In order to reduce the failure rate of software projects, managers need to pay attention to schedule management, finance management, unmet user requirements and quality management. Each of these areas appears as risk if not managed in an adequate manner (Kester, 2013). Generally, risk is defined as a potential future loss or undesirable outcome that may arise from some present action. However, software project risk factors are defined as a source that can pose a serious threat to the successful completion of software development project (March and Shapira, 1987). Failure to understand, identify and manage risk is often regarded as a major cause of software engineering project failure (Wallace *et al.*, 2004a, b). Therefore, a proactive systematic decision-making process via risk management is indeed required to manage underlying risks within the software project. Thus, risk management is the process that starts with identifying, analysing and managing threats to success and plan for necessary course of actions to reduce the chance of project failure. Researchers have often emphasized on categorization and prioritization of different sources and types of risks in order to minimize undesirable losses. Extensive literature review on project management suggests that there is paucity of simple and systematic tools to identify and classify risk factors concerning with software project issues. It is to be noted that risk factors not only affect an individual project but also influence other projects because they are interrelated. Therefore, it is important to understand the nature of risk factors and their interrelationship so that those factors which support other factors ("driving sources") and those which are more influenced by others ("dependent sources") are to be examined (Raj *et al.*, 2008). To this end, current research explores various risk sources in the software project management and develops a structural decision model for establishing the interrelationship among different risk sources through interpretive structural modelling (ISM) methodology. Moreover, the risks are classified depending upon their driving and driven power with the help of indirect relationship by *Matrice d'Impacts croises-multipication appliqué an classment* (MICMAC) analysis.

The main objectives of this research are as follows:

(1) to identify and analyse the interdependencies of different risk factors and their effect in successful execution of software engineering projects;

(2) to establish relationships among the identified risk factors through subjective judgement of experts in a structured manner;

(3) to propose an effective systematic procedure to analyse and classify the risk factors based on their driving and dependence power which can help managers in project risk assessment, treatment and control; and

(4) to develop a structured model which can represents graphically the interdependencies among the risk factors through casual links to make it effective to communicate among the managers for the formulation of project risk management strategy.

The remainder of the paper has been organized into six sections. The next section (Section 2) presents the brief literature review in the present context. The methodology for developing the interrelationship of various risk factors in software engineering projects is presented in Section 3. In Section 4, MICMAC analysis is presented to classify various sources risk. The results and discussions along with managerial implications are given in Section 5. Finally, in Section 6, conclusions are summarized and direction for future research are outlined.

2. Literature review

Extensive research has been conducted on risk management in software projects through identification, analysis and control of risks which threaten the assets of a software enterprise (Boehm, 1991). Risk is defined as a chance of danger, damage, loss, failure or any undesired/negative consequences. Büyüközkan and Ruan (2010) proposed an integrated approach based on fuzzy logic in a multi-criteria decision-making framework to identify and assess the intensity of risk factors. Hoodat and Rashidi (2009) developed a probabilistic model to analyse and assess the risks in software engineering projects. They used a risk tree structure to relate several risk sources and categorize different risks. Cerpa *et al.* (2010) used a logistic regression model to predict the project outcome and analyse effect of various factors on outcome. Li *et al.* (2012) proposed a two metric model-software process module with risk management and cost control module to calculate risk management efficiency and trustworthiness values of software process management. López and Salmeron (2012) presented a risk checklist which affects the performance of software projects. All risk factors are placed in a four quadrant matrix on the basis of their impact and probability ratings. Huang and Han (2008) explored application of cluster analysis technique to classify various risk factors. Nakatsu and Iacovou (2009) studied the effect of important risk factors on project outcome when software development projects are outsourced inshore and offshore using Delphi method.

Keil *et al.* (2008) investigated the software practitioners' risk perception and decision making whereas Jun *et al.* (2011) considered perception of vendors. It is concluded that process performance can be improved by enhancing planning and control for low-risk projects. Product performance can be improved by increasing user participation for high-risk projects. Bakker *et al.* (2010) investigated how risk management contributes to success of projects through meta-analysis of the empirical evidence. Jani (2011) proposed a simulation-based experiment for assessment of risk factors in software project development. Sharma and Gupta (2012) studied the influence demographics and organizational climate on risks in Indian context through exploratory factor analysis. Aloini *et al.* (2012) proposed an ISM technique to analyse the enterprise resource planning project risks. Saxena and Seth (2012) identified several key factors and their relation in supply chain risks and security management. They used ISM approach to analyse the interactions among factors for establishing the relationship between supply chain vulnerability and supply chain risk. Kumar *et al.* (2013) explored the relationship among awareness level of customers, encouragement and support of customers, environment-friendly distribution, effective training programme schedule for customer and recycling and reuse efforts of organization to evaluate the role of customers in green supply chain management model. Hachicha and Elmsalmi (2014) examined the relationships between risk variables for supply network management systems using ISM tool. The identified risk variables were then prioritized based on the MICMAC analysis in the same work. Raeesi *et al.* (2013) took a formal measure to identify several distortions which create barriers to entrepreneurship. They analysed the systematic interactions among barriers using ISM approach. Hatei *et al.* (2013) analysed the relationships among the risk factors involved in public-private partnerships projects using interpretive structure model. They identified twenty risk factors and, pursuant to the model's characteristics, classifying them into three categories such as; dominating factors, transferring factors and indicating factors.

Fu *et al.* (2012) developed a probabilistic model based on design structure matrix to evaluate risk of change propagation from requirements to software development projects.

The model is also capable to estimate the schedule and cost of a software project. Critical examination of literature reveals that most of the studies focus on identification and analysis of risk sources. However, limited studies have been devoted to understand the effect of risks and their importance. Thus, this research attempts to address this vital issue in the context of software project risk management by proposing an innovative decision support model using ISM approach. ISM is a well-established methodology especially used for identifying the relationships among the particular items which address a problem or an issue (Sage, 1977).

3. Research methodology

In order to develop interrelation among various risk factors in software project development, an ISM approach has been employed. The relevant data for ISM model are collected through a cross-sectional questionnaire survey.

3.1 ISM approach

ISM is an interpretive method which is often used in the case of complex situations arising in the system. This method facilitates researchers to understand the complex relationship between many elements associated in the system by developing a comprehensive structured systematic model. The advantage of ISM method lies in converting the unclear, poorly defined mental models into a well-defined hierarchical model for better understanding of complex issues (Warfield, 1994). Moreover, ISM is a well-established methodology for constructing and analysing the fundamentals of interrelationships between the elements in complex systems. This method helps to impose order and direction on the complexity of relationship among the elements of a system so that influence can be analysed between the elements (Mandal and Deshmukh, 1994; Sharma *et al.*, 1995; Singh *et al.*, 2003). ISM methodology has three important characteristics. First, it is interpretive as judgement of the experts decides whether and how the elements are related. Second, it is structural as a complete structure is extracted from the set of elements on the basis of their relationship. Third, it is a modelling technique as a complete structure is represented by diagraph model depicting specific relationships (Raj *et al.*, 2008). When these aforementioned characters inherently exist, it is entitled as "ISM". More precisely, ISM is an interpretive learning process that supports the people to structure their collective knowledge and enhance the ability to understand the complexity of interrelationships between elements through a hierarchical systematic structured model. Many studies in the past have applied ISM approach in various fields and successfully analysed how interrelationship among the element affects to the performance of the overall system (Qureshi *et al.*, 2008; Yang *et al.*, 2008; Khurana *et al.*, 2010; Pfohl *et al.*, 2011; Aloini *et al.*, 2012; Debata *et al.*, 2012). Various steps involved in ISM methodology are as follows:

(1) Identification of the elements relevant to the issue or problem.

(2) Establishing contextual relations among the identified elements. This represents the possible statement of relationship whether the relations are comparative, influence and natural or temporary type. In the present study, an influence type contextual relationship has been chosen. This means one risk influence to another risk element.

(3) Developing a structural self-interaction matrix (SSIM) on the basis of pairwise comparison of the elements.

(4) Construction of reachability matrix from the SSIM and checking for transitivity property. Transitivity of a reachability matrix is the basic assumption of relations that if an element A is related to B, and B is related to C, then it should be considered as A is related to C (Ravi and Shankar, 2005). Transitivity of elements in a matrix leads to construct the final reachability matrix. Reachability matrix is a binary matrix in which the entries V, A, O and X of the SSIM are converted into 1 and 0.

(5) In this step, the obtained reachability matrix is partitioned into different levels.

(6) Drawing a directed graph by removing the transitivity links and also on the basis of reachability matrix.

(7) Conversion of diagraph into ISM model by replacing element nodes with statements.

(8) In last, check the conceptual inconsistency of developed ISM model.

The application of aforesaid steps in applying ISM methodology for analysing software engineering project risks have been explained in more detail in the following subsections.

3.2 Identification of risk factors

In software engineering projects, identification of different risks factors which influence to undesirable project outcome is a critical task. The field experience and insightful perception is indeed required to mitigate the areas of concern. Past studies have been devoted to identify the sources from where risk arises in software engineering projects. On the basis of comprehensive literature survey and opinion of expert', a total of 23 important risk factors have been identified in the present work for the analysis of their interrelationship that affecting to the success of software engineering projects directly or indirectly. A questionnaire-based survey has been conducted to test the validity of each of the identified risk factor affecting to the performance of software engineering projects. The identified risk factors and their sources have been presented in Table I.

3.3 Survey administration

The aim of the questionnaire survey was to collect the relevant data from the experts or industry personnel for establishing a relationship matrix as a first step towards developing an ISM-based model. A questionnaire containing 23 risk factors of software engineering projects has been administered to the respondents with an instruction to compare each and every pair of criteria depicted in SSIM. A respondent is requested to compare the column statement to the row statement for each cell and to select an appropriate value from the symbol set (V, A, X or O) according to their perception towards direct relationship between two risk factors at a time. The relational descriptions of symbols have been provided in the questionnaire in which V represents relation when the factor i influences or reaches the factor j, but not in the opposite direction, A for the relation when factor j influences or reaches the factor i but reverse is not possible, X for the relation both i and j factors are interrelated and O represents the case of relation when both i and j factors are unrelated to each other. The survey used convenient sampling to select the respondents through Tata Consultancy Services lab mailing list which contains 175 members primarily experts in software project management discipline. In total, 175 members primarily experts were participated who had more than ten years' experience in software project practice. The detailed

Table I. Software project risk factors and their references/sources

	Risk factors	Description	Descriptive references
1	Lack of good estimation in projects	Lack of good estimation in projects acts as risk in software engineering projects, which refers to the lack of experience of a personnel's towards forecasting the project duration, budget, software cost and expenditure of man and machinery. Good estimation of projects can reduce the unexpected cost of software and also helps to make a project success	Hoodat and Rashidi (2009) and Wallace *et al.* (2004a)
2	Unrealistic schedule	The risk and uncertainty due to unrealistic schedule can impact the software project performance. Poor planning and control often leads to influence the unrealistic schedules. As a result, excessive schedule pressure or unrealistic schedules that can increase the project risk	Hoodat and Rashidi (2009), Keider (1984) and Wallace *et al.* (2004a)
3	Human errors	Risk that a propensity for certain common mistakes by people; the making of an error as a natural result of being human. Human errors are another major risk factor which may occur due to large size of architecture, complexity of architecture and lack of knowledge. Moreover, failure of tools and hardware's may also invite the human error. As a result, human error can impact the project performance as well as increase the uncertainty of a projects outcome	Hoodat and Rashidi (2009) and Keider (1984)
4	Lack of testing	Lack of testing during the system development project is one of the risk factor that cited in the literature. In each and every steps of development process, testing is needed in order to achieve better quality of a software product. Lack of testing can impact the quality, reliability and cost of a product in software engineering projects	Hoodat and Rashidi (2009) and Wallace *et al.* (2004a)
5	Lack of monitoring	The risk or uncertainty arises due to lack of monitoring during the management of software projects, may causes the failure of the projects. Lack of monitoring can increase the possibility of project delay, poor quality and cost of the software product. Lack of experience and schedule pressure is the main drivers to invite lack of monitoring of software projects in the view of managers concern	Hoodat and Rashidi (2009) and Wallace *et al.* (2004a)
6	Complexity of architecture	Architecture complexity is intend to increase as components and infrastructure built using new technology with large number of links to existing systems and external entities. The number of adaptation of units can also be an inherent cause to increase the project complexity. Complexity becomes in terms of cost, time estimates and specification of requirements, hardware needs, business process and engineering activities, and also the involvement of multiple organizational units. For example, if a new system works in a multiple sites, it may be difficult to define all requirements precisely, because different sites serve different customers and also have different policy or procedures	Hoodat and Rashidi (2009) and Schmidt *et al.* (2001)

(*continued*)

Table I.

	Risk factors	Description	Descriptive references
7	Lack of reassessment of management cycle	Risk will increase tremendously if management cycle not functioning effectively. Reassessment of management cycle from top to bottom level is essential for the success of software projects. For example, if wrong decision is played by the top management regarding a particular operation that will affect to the bottom-level management and ultimately affect to the entire project in terms of serious loss. Thus, the role of management cycle is critical that responsible for all activities at every level of organization	Hoodat and Rashidi (2009), Wallace *et al.* (2004a) and Kanter (1997)
8	Lack of employment of manager experience	To overcome the organizational difficulties and resistance to change, experienced managerial persons are highly required. They can utilize relevant experience and knowledge as well as power to manage the resources. As a result an experience manager can make and implement better decision to control the uncertainty situations with in the software projects. So lack of employment of manager experience cited as a risk in the literature	Hoodat and Rashidi (2009) and Wallace *et al.* (2004a)
9	Lack of enough skill	The nature of software engineering projects are basically complex because of the combinations of many hardware and software, as well as a wide range of organizational, human and political issues. Therefore, there is a need of significant project management skills and also required operators with better skill to adopt the knowledge of advance techniques for software projects. Noticeably, high-skilled operators and management personnel's have major contributions towards making the project success. Thus lack of enough skill is considered as a risk that can impact on the project performance	Hoodat and Rashidi (2009) and Wallace *et al.* (2004a)
10	Inadequate design and documentation	The elements of design and documentation include central planning or decentralization, specific control and specialization, and workforce management. Dynamic and inadequate documentation invites risks as it is impossible to coordinate similar activities. Because, responsibilities are not adequately shared out with in the dynamic documentation system. For example, the information like who is in overall charge, how far any control extent and many more. Thus, inadequate design and documentation is another major risk of software projects often cited in the literature	Hoodat and Rashidi (2009), Wallace *et al.* (2004a) and Zhang and Dilts (2004)
11	Inadequate knowledge about tools, techniques and programming language	Inadequate knowledge about tools, techniques and programming language is another risk that can make failure of the software project. To overcome this problem, well train operators or personnel's will be required in the deployment stage. Non-train employees do not know adequately how to use the tools, techniques and programming language during the system development	Hoodat and Rashidi (2009) and Wallace *et al.* (2004a)
12	Lack of project standard	Every project should come under some standard and a guideline which was regulated and established by National Standard Commission; for example, CARE standard act in UK. The project beyond or lack of that standard guideline can impact on cost, quality and performance of a software product. Thus lack of project standard is considered as one of the risk factor in software engineering projects	Hoodat and Rashidi (2009) and Wallace *et al.* (2004a)

(*continued*)

	Risk factors	Description	Descriptive references
13	Inadequate budget	Poorly formulated budget can affect to the quality of a software development project. So this is also taken as risk in software projects	Hoodat and Rashidi (2009) and Wallace *et al.* (2004a)
14	Inadequate of requirements	There are several kinds of requirements needed in software projects such as; functional, interface, data, security and quality, etc. But many projects do not manage their requirements effectively. Managers store their requirement in paper documents rather store the requirements in a database or the repository of a requirements tool (www.jot.fm/issues/issue_2007_01/column2/). Thus, scattered requirements are difficult to find, sort, query and maintain itself which can impact to the performance of software projects	Hoodat and Rashidi (2009) and Wallace *et al.* (2004a)
15	Lack of report for requirements	Reports with lack of necessary detail of requirements regarding the usable methods and techniques will cause the requirement engineer to waste time and delay the project. A poor documented report can produce poor software products that must be in poor quality outcome. So lack of report for requirements is also considered as a risk factor in software project often cited in the literature	Hoodat and Rashidi (2009) and Wallace *et al.* (2004a)
16	Lack of analysis for change of requirements	Lack of analysis of frequently changed requirements in software engineering projects is considered as a risk that can impact on the cost, quality and performance of the software project outcomes	Hoodat and Rashidi (2009) and Wallace *et al.* (2004a)
17	Lack of trust between partners	Lack of trust is one of the main risk factor in software engineering projects. The degree of trust represents the honesty, generosity between the partners and also the overall competence to others. If there is no trust between partners then the problems, such as unwilling to pass on sensitive information, unable to agree towards the decision of finance management may come to arise. Due to this, lack of trust invites the risk, and affects to the scenario of cooperation or collaboration which may lead to damage the stability of an organization	Alawamleh and Popplewell (2011), Zaheer *et al.* (1998) and Sinha *et al.* (2004)
18	Heterogeneity of partners	Heterogeneity of partners of software engineering projects or information technology systems adds another riskiness factor that often cited in the literature. Heterogeneity refers the differences that subsist between the partners in terms of unsuited hardware, operating systems, difference in languages, and sharing the information. Such type of risks usually exists in IT infrastructure, working methods and business practices when the nature between possible partners is heterogeneous	Alawamleh and Popplewell (2011), Sari *et al.* (2007) and Sing and Kant (2008)
19	Wrong partner/s selection	Insufficient information about the partners and conflict between partners are the real cause of wrong partner/s selection. Organizational conflict problems can reveal through the conflict relationships, task conflict and conflict over process. The consequences of these factors can badly impact on the software development projects. Thus, wrong partner/s selection is considered as a risk in software projects	Alawamleh and Popplewell (2011), Sari *et al.* (2007) and Wilmot and Hocker (2001)

(*continued*)

Table I.

Table I.

	Risk factors	Description	Descriptive references
20	Software cost risks	Software cost risks is one of the most important risk factors in software projects which mostly depends on the cost of a projects. Several sub factors like, lack of good estimation of projects, unrealistic schedule, human errors and the changes in terms of management, technology, personnel and environment may also responsible for increasing software cost risks. The consequences of aforementioned risks will damage the software projects and increase the cost of software	Expert opinion
21	Software quality risks	Risks that can affect to the quality of software are called as software quality risks. Loss of technical equipment's, lack of stability between personnel, lack of skill towards programming knowledge and training, undesired event in costs and requirements, weakness of management and lack of project standard are the main causes which may arise in the view of risks, and affects to the quality of a software engineering projects	Expert opinion
22	Software scheduling risks	Scheduling risk is the main cause to delay the software projects and can effect in financial damage during project life cycle. Human errors, improper planning, lack of monitoring, inadequate business pressure, shortages and changes in software projects are the main influencing factors responsible for increase the scheduling risks with in software engineering projects	Hoodat and Rashidi (2009) and Thayer *et al.* (1980)
23	Software requirement risks	Risk or uncertainty surrounding in the system requirement is one of the major concerns that can be affecting to project performance. Usually, changing requirements are not the only expected requirement-related problem in software development projects. Moreover, incorrect, unclear, ambiguous and unusable requirements may also enhance the requirement risks associated in software engineering projects	Hoodat and Rashidi (2009) and Wallace *et al.* (2004a)

questionnaire has been mailed to the identified experts with a request to explore best of their experience and expertise in assessing various risk quantifying factors of the software engineering exercise. Experts have been personally requested to avoid biasness in responding various issues related to software project risk scenario. No face to face interviews have been conducted. Respondents have been provided a couple month of time duration to understand, to analyse and to recapitulate their experience in addressing interactions among the risk factors as depicted in the detailed questionnaire. Thereafter, response data have been received and those have been critically analysed. The decision judgement of the aforesaid expert group has been considered fully reliable and ultimate (organization specific) which could be utilized on investigating interrelationship among various risk influencing factors in relation to software project practice. Out of 175, only 55 respondents participated in the survey with a response rate of 32 percentages approximately. Finally, 48 correct and complete responses are used for further analysis. The remaining responses are rejected due to incompleteness and irrationality.

3.4 Formation of SSIM

This is the most important and demanding phase of ISM methodology where the contextual relationship among the risk factors based on experts opinion is incorporated. Keeping this in mind, the questionnaire has been designed in such a way that the existence of a relation between any two risk factors (*i* and *j*) and the associated direction of the relation execution is questioned. Thereafter the participants decide upon the pairwise relationship between the risk factors. Based on expert's feedback on 23 identified software project risks, the SSIM was developed and presented in Table II. The entries in the SSIM matrix was based on the maximum responses obtained for the pair of risk factors.

3.5 Construction of reachability matrix

The SSIM has been transformed into reachability matrix by two sub-steps. First, SSIM is converted into initial reachability matrix by substituting the entry of each cell (V, A, X and O) into binary digits (1 or 0) as per the following rules:

- if the (*i*, j) entry in the SSIM is V, then the (*i*, j) entry in the initial reachability matrix becomes 1 and the (*i*, j) entry becomes 0;
- if the (*i*, j) entry in the SSIM is A, then the (*i*, j) entry in the initial reachability matrix becomes 0 and the (*i*, j) entry becomes 1;
- if the (*i*, j) entry in the SSIM is X, then both the (*i*, j) and (*i*, j) entries of the initial reachability matrix becomes 1; and
- if the (*i*, j) entry in the SSIM is O, then both the (*i*, j) and (*i*, j) entries of the initial reachability matrix becomes 0.

Based on the following rules, the initial reachability matrix has been prepared as shown in Table III. In second sub-step, initial reachability matrix has been transformed into final reachability matrix by checking the transitivity property. After integrating the transitivity concept as mentioned in fourth step of ISM methodology, the final reachability matrix has been constructed and furnished in Table IV. Final reachability matrix also represents the driving power and dependence of each risk factor. Driving power of each risk is the summation of total number of risk interactions in the row (including itself) which it affects.

	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2
1	O	V	V	V	O	O	O	O	O	O	O	O	O	O	A	A	O	O	O	O	O	O
2	O	V	V	V	A	A	O	O	O	O	O	O	O	O	O	A	O	O	A	O	A	
3	O	V	V	V	A	A	A	O	O	O	O	O	O	O	A	A	O	O	A	O		
4	O	O	V	V	O	O	O	O	O	O	O	O	X	O	A	A	O	O	O			
5	O	O	O	V	O	A	A	O	O	O	O	O	O	O	A	A	A	A				
6	O	O	O	V	O	A	O	O	O	O	O	O	O	O	O	O	O					
7	O	O	O	V	A	A	A	O	O	O	O	O	O	O	O	O						
8	O	V	O	O	O	O	O	A	A	O	O	O	O	O	O							
9	O	V	V	O	A	O	O	O	O	O	O	O	O	A								
10	O	O	V	O	A	A	A	O	O	O	O	O	O									
11	O	V	V	O	A	A	A	O	O	O	O	O										
12	O	O	V	A	O	O	O	O	O	O	O											
13	O	O	V	O	O	O	O	O	O	O												
14	V	O	O	O	O	O	O	A	A													
15	V	O	O	O	O	A	A	A														
16	V	O	V	O	O	A	A															
17	V	O	V	O	V	A																
18	V	V	V	V	V																	
19	V	V	V	V																		
20	A	A	X																			
21	X	X																				
22	O																					
23																						

Table II.
Structural self-interaction matrix (SSIM)

Elements	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
3	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
4	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0
5	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
6	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
7	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
8	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
9	1	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0
10	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0
11	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0
12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
15	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1
16	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	0	0	0	0	1	0	1
17	0	0	1	0	1	0	1	0	0	1	1	0	0	0	1	1	1	0	1	0	1	0	1
18	0	1	1	0	1	1	1	0	0	1	1	0	0	0	1	1	1	1	1	1	1	1	1
19	0	1	1	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1
20	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

Table III.
Initial reachability matrix

Table IV. Final reachability matrix with driving and dependence power

Elements	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	Driving power
1	1	0	0	0	0	0	0	0	0	0	0	1*	0	0	0	0	0	0	0	1	1	1	1*	6
2	0	1	0	0	0	0	0	0	0	0	0	1*	0	0	0	0	0	0	0	1	1	1	1*	6
3	0	1	1	0	0	0	0	0	0	0	0	1*	0	0	0	0	0	0	0	1	1	1	1*	7
4	0	0	0	1	0	0	0	0	0	0	1	1*	0	0	0	0	0	0	0	1	1	1*	1*	7
5	0	1	1	0	1	0	0	0	0	0	0	1*	0	0	0	0	0	0	0	1	1*	1*	1*	8
6	0	1*	1*	0	1	1	0	0	0	0	0	1*	0	0	0	0	0	0	0	1	1*	1*	1*	9
7	0	1*	1*	0	1	0	1	0	0	0	0	1*	0	0	0	0	0	0	0	1	1*	1*	1*	9
8	1	1	1	1	1	0	0	1	0	0	1*	1*	0	0	0	0	0	0	0	1*	1*	1	1*	12
9	1	1*	1	1	1	0	0	0	1	0	1*	1*	0	0	0	0	0	0	0	1*	1	1	1*	12
10	1*	1*	1*	1*	0	0	0	0	1	1	1*	1*	0	0	0	0	0	0	0	1*	1	1*	1*	13
11	0	0	0	1	0	0	0	0	0	0	1	1*	0	0	0	0	0	0	0	1*	1	1	1*	7
12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1*	1	1*	1*	5
13	0	0	0	0	0	0	0	0	0	0	0	1*	1	0	0	0	0	0	0	1*	1	1*	1*	6
14	0	0	0	0	0	0	0	0	0	0	0	1*	0	1	0	0	0	0	0	1*	1*	1*	1	6
15	1*	1*	1*	1*	1*	0	0	1	0	0	1*	1*	0	1	1	0	0	0	0	1*	1*	1*	1	14
16	1*	1*	1*	1*	1*	0	0	1	0	0	1*	1*	0	1	1	1	0	0	0	1*	1	1*	1	15
17	1*	1*	1	1*	1	0	1	1*	1*	1	1	1*	0	1*	1	1	1	0	1	1*	1	1*	1	20
18	1*	1	1	1*	1	1	1	1*	1*	1	1	1*	0	1*	1	1	1	1	1	1	1	1	1	22
19	1*	1	1	1*	1*	0	1	0	1	1	1	1*	0	0	0	0	0	0	1	1	1	1	1	15
20	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1*	1*	5
21	0	0	0	0	0	0	0	0	0	0	0	1*	0	0	0	0	0	0	0	1	1	1	1	5
22	0	0	0	0	0	0	0	0	0	0	0	1*	0	0	0	0	0	0	0	1	1	1	1*	5
23	0	0	0	0	0	0	0	0	0	0	0	1*	0	0	0	0	0	0	0	1	1	1*	1	5
Dependence	9	13	12	10	11	2	4	5	5	4	10	23	1	5	4	3	2	1	3	23	23	23	23	219/219

Note: 1* entries are indicated as transitivity

However, dependence of each risk is the summation of total number of risk interactions in the column (including itself) by which it is affected. Based on these driving power and dependence, software project risks have later been classified in Section 4.

3.6 Level partitioning

Level partitioning helps for constructing the diagraph model based on the final reachability matrix (Warfield, 1977). Final reachability matrix provides the information about the reachability and antecedent set for each risk factor. The reachability set of the element is the set of elements that contains the element itself and other elements to which it may reach, whereas the antecedent set contains the element itself and the other elements which may reach to it (Mandal and Deshmukh, 1994). More precisely, reachability set of the risk is the set of elements of a final reachability matrix which contain 1 in row of that particular risk. Conversely, antecedent set of the risk is the set of elements which contain 1 in column of that particular risk (Pfohl *et al.*, 2011). Based on the reachability set and antecedent set, the intersection sets have been derived for all elements. Intersection sets are the common elements of both reachability set and the antecedent set. The case where the elements of reachability and intersection sets are same, that is the indicator of top-level element. For example, in the present case, five risk factors such as lack of project standard; software quality risks; software cost risks; software requirement risks; and software scheduling risks have been identified as top-level elements as shown in Table V (Iteration 1). The significance of top-level elements is that they will not influence any other element above their own level in the hierarchy. Once the top-level element is recognized, then it is discarded from further hierarchical consideration (i.e. separated out that elements from all the different sets). Similarly, the next level of elements has been partitioned by the same process. The stepwise level partitions of all 23 risk factors have been completed in nine iterations as shown in Tables V-VII. The summary of all partition levels has been represented in Table VIII.

Element	Reachability set	Antecedent set	Interaction set	Level
1	1,12,20,21,22,23	1,8,9,10,15,16,17,18,19	1	
2	2,12,20,21,22,23	2,3,5,6,7,8,9,10,15,16,17,18,19	2	
3	2,3,12,20,21,22,23	3,5,6,7,8,9,10,15,16,17,18,19	3	
4	4,11,12,20,21,22,23	4,8,9,10,11,15,16,17,18,19	4,11	
5	2,3,5,12,20,21,22,23	5,6,7,8,9,10,15,16,17,18,19	5	
6	2,3,5,6,12,20,21,22,23	6,18	6	
7	2,3,5,7,12,20,21,22,23	7,17,18,19	7	
8	1,2,3,4,5,8,11,12,20,21,22,23	8,15,16,17,18	8	
9	1,2,3,4,5,9,11,12,20,21,22,23	9,10,17,18,19	9	
10	1,2,3,4, 9,10,11,12,20,21,22,23	10,17,18,19	10	
11	4,11,12,20,21,22,23	4,8,9,10,11,15,16,17,18,19	4,11	
12	12,20,21,22,23	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23	12,20,21,22,23	I
13	12,13,20,21,22,23	13	13	
14	12,14,20,21,22,23	14,15,16,17,18	14	
15	1,2,3,4,5,8,11,12,14,15,20,21,22,23	15,16,17,18	15	
16	1,2,3,4,5,8,11,12,14,15,16,20,21,22,23	16,17,18	16	
17	1,2,3,4,5,7,8,9,10,11,12,14,15,16,17,19,20,21,22,23	17,18	17	
18	1,2,3,4,5,6,7,8,9,10,11,12,14,15,16,17,18,19,20,21,22,23	18	18	
19	1,2,3,4,5,7,9,10,11,12,19,20,21,22,23	17,18,19	19	
20	12,20,21,22,23	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23	12,20,21,22,23	I
21	12,20,21,22,23	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23	12,20,21,22,23	I
22	12,20,21,22,23	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23	12,20,21,22,23	I
23	12,20,21,22,23	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23	12,20,21,22,23	I

Table V. Iteration 1

Table VI. Iteration 2

Element	Reachability set	Antecedent set	Interaction set	Level
1	1	1,8,9,10,15,16,17,18,19	1	II
2	2	2,3,5,6,7,8,9,10,15,16,17,18,19	2	II
3	2,3	3,5,6,7,8,9,10,15,16,17,18,19	3	
4	4,11	4,8,9,10,11,15,16,17,18,19	4,11	II
5	2,3,5	5,6,7,8,9,10,15,16,17,18,19	5	
6	2,3,5,6	6,18	6	
7	2,3,5,7	7,17,18,19	7	
8	1,2,3,4,5,8,11	8,15,16,17,18	8	
9	1,2,3,4,5,9,11	9,10,17,18,19	9	
10	1,2,3,4, 9,10,11	10,17,18,19	10	
11	4,11	4,8,9,10,11,15,16,17,18,19	4,11	II
13	13	13	13	II
14	14	14,15,16,17,18	14	II
15	1,2,3,4,5,8,11,14,15	15,16,17,18	15	
16	1,2,3,4,5,8,11,14,15,16	16,17,18	16	
17	1,2,3,4,5,7,8,9,10,11,14,15,16,17,19	17,18	17	
18	1,2,3,4,5,6,7,8,9,10,11,14,15,16,17,18,19	18	18	
19	1,2,3,4,5,7,9,10,11,19	17,18,19	19	

Table VII. Iteration 9

Element	Reachability set	Antecedent set	Interaction set	Level
18	18	18	18	IX

Table VIII. Summary of level partitioning

Element	Reachability set	Antecedent set	Interaction set	Level
1	1,12,20,21,22,23	1,8,9,10,15,16,17,18,19	1	II
2	2,12,20,21,22,23	2,3,5,6,7,8,9,10,15,16,17,18,19	2	II
3	2,3,12,20,21,22,23	3,5,6,7,8,9,10,15,16,17,18,19	3	III
4	4,11,12,20,21,22,23	4,8,9,10,11,15,16,17,18,19	4,11	II
5	2,3,5,12,20,21,22,23	5,6,7,8,9,10,15,16,17,18,19	5	IV
6	2,3,5,6,12,20,21,22,23	6,18	6	V
7	2,3,5,7,12,20,21,22,23	7,17,18,19	7	V
8	1,2,3,4,5,8,11,12,20,21,22,23	8,15,16,17,18	8	V
9	1,2,3,4,5,9,11,12,20,21,22,23	9,10,17,18,19	9	V
10	1,2,3,4, 9,10,11,12,20,21,22,23	10,17,18,19	10	VI
11	4,11,12,20,21,22,23	4,8,9,10,11,15,16,17,18,19	4,11	II
12	12,20,21,22,23	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23	12,20,21,22,23	I
13	12,13,20,21,22,23	13	13	II
14	12,14,20,21,22,23	14,15,16,17,18	14	II
15	1,2,3,4,5,8,11,12,14,15,20,21,22,23	15,16,17,18	15	VI
16	1,2,3,4,5,8,11,12,14,15,16,20,21,22,23	16,17,18	16	VII
17	1,2,3,4,5,7,8,9,10,11,12,14,15,16,17,19,20,21,22,23	17,18	17	VIII
18	1,2,3,4,5,6,7,8,9,10,11,12,14,15,16,17,18,19,20,21,22,23	18	18	IX
19	1,2,3,4,5,7,9,10,11,12,19,20,21,22,23	17,18,19	19	VII
20	12,20,21,22,23	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23	12,20,21,22,23	I
21	12,20,21,22,23	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23	12,20,21,22,23	I
22	12,20,21,22,23	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23	12,20,21,22,23	I
23	12,20,21,22,23	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23	12,20,21,22,23	I

3.7 Development of ISM model
After level partitioning, lower triangular form of reachability matrix has been prepared by arranging the elements according to their levels. After removing the indirect links, a diagraph has been drawn by means of nodes or vertices and lines of edges. The relationship between elements *i* and *j* has been shown by an arrow which connects from *i* to *j*. This constructed diagraph has been converted into an ISM-based model by mentioning the descriptions of elements within it (Figure 1). The elements of ISM model has been connected in a complete hierarchical form with no feedbacks or no cycles.

4. MICMAC analysis

The abbreviation of MICMAC is the "*Matrice d'Impacts croises-multipication appliqué an classment*" means cross-impact matrix multiplication applied to classification

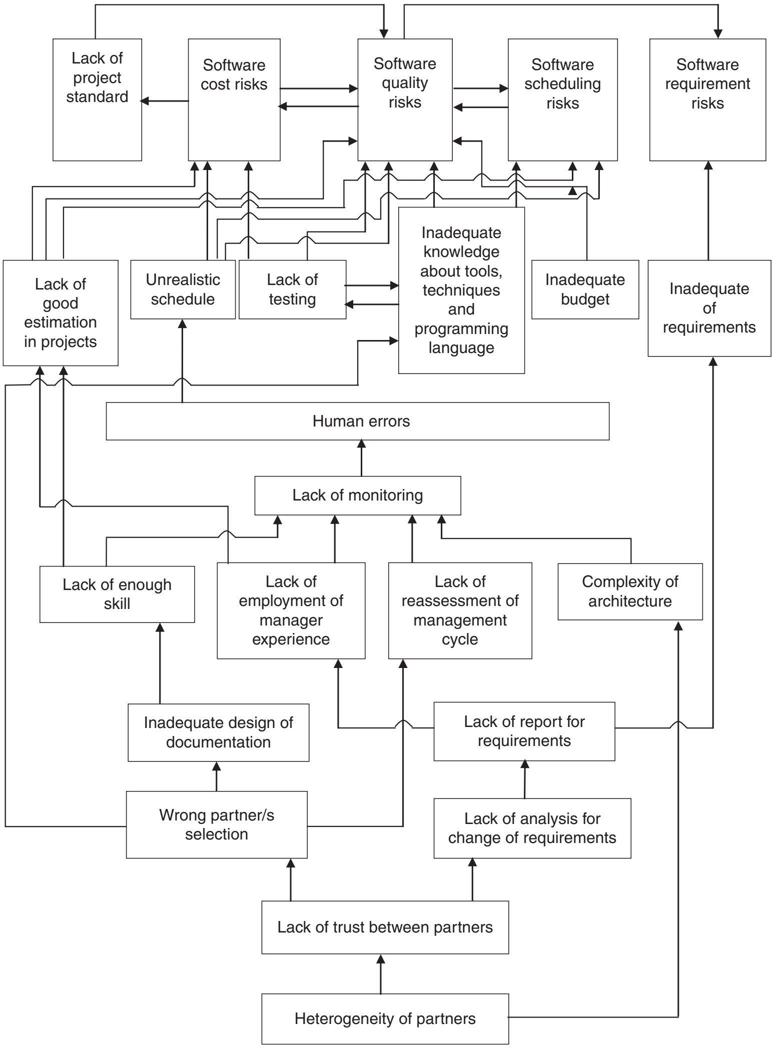

Figure 1.
ISM-based model

(Sharma *et al.*, 1995). MICMAC analysis is a part of structural analysis aims to identify the most important variables of a system from matrix that establishes the relations among them (Villacorta *et al.*, 2012). In this study, the identification and classification of risk is essentially required for the implementation of risk management strategy in software engineering project. MICMAC is an indirect classification method which helps to critically analyse the scope of the risks (Saxena and Sushil, 1990). The objective of MICMAC analysis is to analyse and classify the risk elements based on their driving power and dependence. Based on the concept of MICMAC, all risk factors have been classified into four clusters of risks according to their driving power and dependence value (Figure 2).

Cluster I consists of autonomous risk factors which have weak driving power and weak dependence. There are eight risk factors which come under autonomous cluster namely, inadequate budget, lack of reassessment of management cycle, inadequate knowledge about tools and techniques, complexity of architecture, lack of testing, lack of good estimation in projects and lack of monitoring. These risk factors are comparatively separated from the system although a few existence of links which may not be strong and do not have much influence on the system.

Cluster II includes the dependent risk factors which have weak driving power and strong dependence. A total number of seven risk factors have been identified in this cluster. Mostly top-level risk factors of ISM model are come under this category. In the present study, top-level factors namely, software cost risks, software quality risks, software scheduling risks, software requirement risks and lack of project standard are shown in dependent cluster. Top-level factors are most resulting action of risks in software projects. The factors having strong dependence property indicates that it is being strongly influenced by other risk factors and thereby increases in software project risks. Thus, managers should pay special attention to these risks.

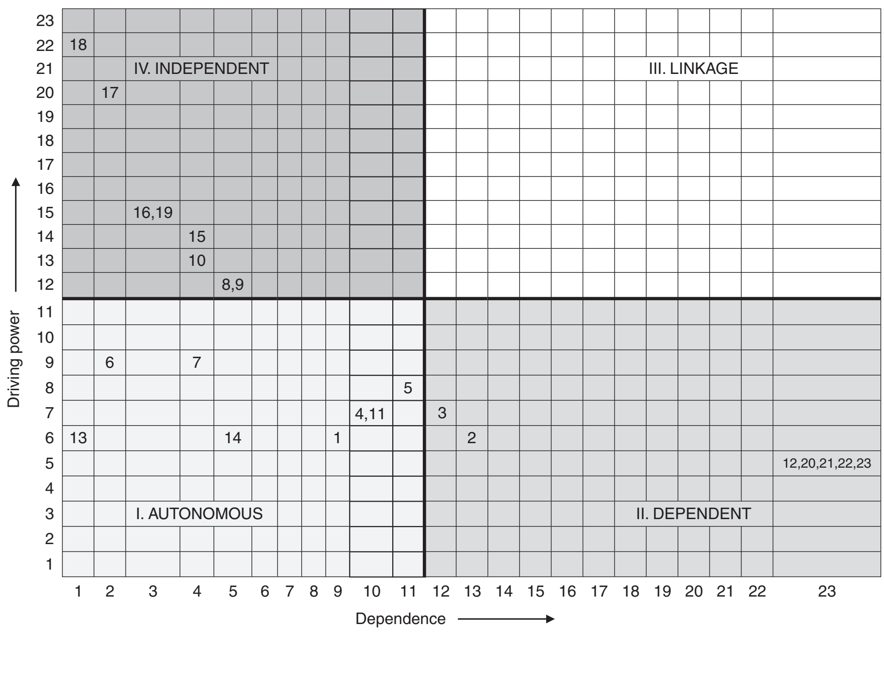

Figure 2.
Driving power-dependence diagram

Cluster III comprises linkage risk factors which have both strong driving and dependence power. The risk factors associates with linkage clusters are unstable, because if any change occurs on these risks that will have an effect on other risks. In this research, there is no risk factors exist in the linkage cluster.

In Cluster IV, all independent risk factors are clustered that have strong driving power but weak dependence. Eight risk factors are identified in this cluster namely, lack of enough skill, lack of employment of manager experience, inadequate design and documentation, lack of report for requirements, lack of analysis for change of requirements, lack of trust between partners, wrong partner/s selection and heterogeneity of partners. The factor which has very strong driving power in the independent cluster is called as "key factor". Heterogeneity of partners has been observed as a key risk factor which has maximum driving power (22). It seems strongly influencing to other risk factors (Figure 1). The driving and dependence power of each risk factor are previously described in Section 3.5 and shown in Table IV.

5. Results and discussions

The results of this study provide an understanding of identified software project risks in different levels of ISM model. The developed hierarchical ISM model comprises 23 software project risk factors in different levels from top to bottom. Understanding the impact of risk at each level is indeed important as it would help managers to construct and implement successful risk management strategy towards achieving success of software project. In this research software cost risks, software quality risks, software scheduling risks, software requirement risks and lack of project standard have been found placed in top level as shown in ISM model (Figure 1). These are the risks which can produce major impact on software engineering projects because all other risks which are being placed just below the top level, strongly influences to them. Thus, managers should pay special attention to control these aforementioned risks for reducing the chance of project failure. Moreover, lower level risks such as; heterogeneity of partners, lack of trust between partners, wrong partner/s selection, lack of analysis for change of requirements, inadequate design and documentation, lack of report for requirements, lack of enough skill, lack of monitoring, human errors and others are strongly influence to the middle-level factors like lack of good estimation in projects, unrealistic schedule, lack of testing and others (Figure 1). Also, aforesaid middle-level risks are again seemed to influence to top level in the ISM diagraph. Top-level factors are more risky than the others that can pose serious impact to the projects. However, lower level factors are mainly responsible for increasing the degree of risk extent as they are influencing strongly to the top-level factors. In this regard, it is observed that interdependency among various risk factors plays an important role for the assessment of risk impact on the software development projects. Moreover, this can also be an important insight into the extent body knowledge to the managers towards implementing appropriate risk management strategy for the reduction of overall risk extent.

MICMAC analysis has been carried out for the 23 risk factors and classified into four clusters (autonomous, dependent, linkage, independent) based on their driving power and dependence. The risk factors are namely, software cost risks, software quality risks, software scheduling risks, software requirement risks and lack of project standard are dependent factors. The impact of these risks depends on other remaining risks of software projects and affects seriously to the system. Similarly, the risk factors like lack of enough skill, lack of employment of managerial experience, inadequate design and documentation, lack of report for requirements, lack of analysis for change of requirements, lack of trust between partners, wrong partner/s selection and heterogeneity of partners have been found independent having strong driving power. These are the risks which play important role to influence others and finally intensifies the strength of impact on software engineering projects. As a result, this cluster analysis may help project managers to understand and assess the intensity of risks as well as to manage these risks by implementing a proactive risk management strategy in future. The results of the present study support a socio-technical perspective providing an ISM approach to conceptualize the category of software project risks and to understand interrelationships between 23 risk factors that have been identified.

Apart from discussing outcome and implications of the present research, it is important to address the limitations of this study. The source of risk is evidently enormous but not limited. In this paper, a total of 23 software risk factors that had been selected in relation to their possible effect towards software project area but there may be some other factors that may also affect the success of software projects need to be taken under consideration.

6. Conclusions

The major contribution of this work is to provide empirical evidence highflying how interrelationships among various risk factors affect to the software engineering projects. Based on extensive literature review and expert opinion, a total of 23 risk factors seemed to impose negative impact on schedule time, quality, cost, requirement or total failure for the software projects have been identified. In this research, an ISM approach has been applied to understand the significant relationships and interdependencies among these 23 identified risk factors associated in software projects. ISM provides a systematic hierarchical structured model helpful in managerial context to identify and understand the interrelationships among different risk factors. Moreover, the direct and indirect relationships between risks can also be identified from the ISM-based model. The process has been found systematic, efficient and capable of producing a structured model which graphically represents the original problem that can be communicated more effectively to the others. Another contribution of this study is the MICMAC analysis that provides a concept of identification and classification of software project risk factors in four different clusters based on their driving and dependence power. The result of this analysis provides an understanding of risk factors as a function of driving power and dependence. The above research findings provide important guidelines to the software project managers that they would plan strategically to implement a proactive risk management strategy for the success of software projects. The future scope of this work can further be extended to fuzzy ISM, a step ahead of binary ISM to deal with uncertainty involved in human judgement process. Furthermore, this model can be tested and validated using structural equation modelling.

References

Alawamleh, M. and Popplewell, K. (2011), "Interpretive structural modelling of risk sources in a virtual organisation", *International Journal of Production Research*, Vol. 49 No. 20, pp. 6041-6063.

Aloini, D., Dulmin, R. and Mininno, V. (2012), "Risk assessment in ERP projects", *Information Systems*, Vol. 37 No. 3, pp. 183-199.

Bakker, K.D., Boonstra, A. and Wortmann, H. (2010), "Does risk management contribute to IT project success? A meta-analysis of empirical evidence", *International Journal of Project Management*, Vol. 28 No. 5, pp. 493-503.

Boehm, B.W. (1991), "Software risk management: principles and practices", *IEEE Software*, Vol. 8 No. 1, pp. 32-41.

Büyüközkan, G. and Ruan, D. (2010), "Choquet integral based aggregation approach to software development risk assessment", *Information Sciences*, Vol. 180 No. 3, pp. 441-451.

Cerpa, N., Bardeen, M., Kitchenham, B. and Verner, J. (2010), "Evaluating logistic regression models to estimate software project outcomes", *Information and Software Technology*, Vol. 52 No. 9, pp. 934-944.

Debata, B.R., Patnaik, B., Mahapatra, S.S. and Sreekumar (2012), "An integrated approach for service quality improvement in medical tourism: an Indian perspective", *International Journal of Services and Operations Management*, Vol. 13 No. 1, pp. 119-145.

Fu, Y., Li, M. and Chen, F. (2012), "Impact propagation and risk assessment of requirement changes for software development projects based on design structure matrix", *International Journal of Project Management*, Vol. 30 No. 3, pp. 363-373.

Grimstad, S. (2006), "Software effort estimation error", PhD dissertation submitted in the Department of Informatics, Faculty of Mathematics and Natural Sciences, University of Oslo, Oslo, November.

Hachicha, W. and Elmsalmi, M. (2014), "An integrated approach based-structural modeling for risk prioritization in supply network management", *Journal of Risk Research*, Vol. 17 No. 10, pp. 1301-1324.

Hatei, S., Wei, L., ShouQing, W. and Yongjian, K. (2013), "PPP project risk relationships based on the interpretive structure model. Advances in information", *Sciences and Service Sciences*, Vol. 5 No. 9, pp. 618-627.

Hoodat, H. and Rashidi, H. (2009), "Classification and analysis of risks in software engineering", *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, Vol. 3 No. 8, pp. 2044-2050.

Huang, S.J. and Han, W.M. (2008), "Exploring the relationship between software project duration and risk exposure: a cluster analysis", *Information and Management*, Vol. 45 No. 3, pp. 175-182.

Jani, A. (2011), "Escalation of commitment in troubled IT projects: influence of project risk factors and self-efficacy on the perception of risk and the commitment to a failing project", *International Journal of Project Management*, Vol. 29 No. 7, pp. 934-945.

Jun, L., Qiuzhen, W. and Qingguo, M. (2011), "The effects of project uncertainty and risk management on IS development project performance: a vendor perspective", *International Journal of Project Management*, Vol. 29 No. 7, pp. 923-933.

Kanter, R.M. (1997), "Restoring people to the heart of the organisation of the future", in Hesselbein, F. *et al.* (Eds), *The Organisation of the Future*, Jossey-Bass, San Francisco, CA, pp. 139-150.

Keider, S.P. (1984), "Why systems development projects fail", *Journal of Information Systems Management*, Vol. 1 No. 3, pp. 33-38.

Keil, M., Li, L., Mathiassen, L. and Zheng, G. (2008), "The influence of checklists and roles on software practitioner risk perception and decision-making", *The Journal of Systems and Software*, Vol. 81 No. 6, pp. 908-919.

Kester, Q.A. (2013), "Application of formal concept analysis to visualization of the evaluation of risks matrix in software engineering projects", *International Journal of Science, Engineering and Technology Research*, Vol. 2 No. 1, pp. 220-225.

Khurana, M.K., Mishra, P.K., Jain, R. and Singh, A.R. (2010), "Modeling of information sharing enablers for building trust in Indian manufacturing industry: an integrated ISM and fuzzy MICMAC approach", *International Journal of Engineering Science and Technology*, Vol. 2 No. 6, pp. 1651-1669.

Kumar, S., Luthra, S. and Haleem, A. (2013), "Customer involvement in greening the supply chain: an interpretive structural modeling methodology", *Journal of Industrial Engineering International*, Vol. 9 No. 6, pp. 1-13.

Li, J., Li, M., Wu, D. and Song, H. (2012), "An integrated risk measurement and optimization model for trustworthy software process management", *Information Sciences*, Vol. 191, pp. 47-60.

López, C. and Salmeron, J.L. (2012), "Risks response strategies for supporting practitioners decision-making in software projects", *Procedia Technology*, Vol. 5, pp. 437-444.

Mandal, A. and Deshmukh, S.G. (1994), "Vendor selection using interpretive structural modelling (ISM)", *International Journal of Operations and Production Management*, Vol. 14 No. 6, pp. 52-59.

March, J.G. and Shapira, Z. (1987), "Managerial perspectives on risk and risk taking", *Management Science*, Vol. 33 No. 11, pp. 1404-1418.

Nakatsu, R.T. and Iacovou, C.L. (2009), "A comparative study of important risk factors involved in offshore and domestic outsourcing of software development projects: a two-panel Delphi study", *Information and Management*, Vol. 46 No. 1, pp. 57-68.

Pfohl, H.-C., Gallus, P. and Thomas, D. (2011), "Interpretive structural modeling of supply chain risks", *International Journal of Physical Distribution and Logistics Management*, Vol. 4 No. 9, pp. 839-859.

Qureshi, M.N., Kumar, D. and Kumar, P. (2008), "An integrated model to identify and classify the key criteria and their role in the assessment of 3PL services providers", *Asia Pacific Journal of Marketing and Logistics*, Vol. 20 No. 2, pp. 227-249.

Raeesi, R., Dastranj, M., Mohammadi, S. and Rasouli, E. (2013), "Understanding the interactions among the barriers to entrepreneurship using interpretive structural modeling", *International Journal of Business and Management*, Vol. 8 No. 13, pp. 56-72.

Raj, T., Shankar, R. and Suhaib, M. (2008), "An ISM approach for modelling the enablers of flexible manufacturing system: the case for India", *International Journal of Production Research*, Vol. 46 No. 24, pp. 6883-6912.

Ravi, V. and Shankar, R. (2005), "Analysis of interactions among the barriers of reverse logistics", *Technology Forecasting and Social Change*, Vol. 72 No. 8, pp. 1011-1029.

Sage, A.P. (1977), *Interpretive Structural Modeling: Methodology for Large-Scale Systems*, McGraw-Hill, New York, NY.

Sari, B., Amaitik, S. and Kilic, S.E. (2007), "A neural network model for the assessment of partners' performance in virtual enterprises", *The International Journal of Advanced Manufacturing Technology*, Vol. 34 Nos 7/8, pp. 816-825.

Saxena, A. and Seth, N. (2012), "Supply chain risk and security management: an interpretive structural modelling approach", *International Journal of Logistics Economics and Globalisation*, Vol. 4 Nos 1/2, pp. 117-132.

Saxena, J.P. and Sushil, V.P. (1990), "Impact of indirect relationships in classification of variables – a MICMAC analysis for energy conservation", *Systems Research*, Vol. 7 No. 4, pp. 245-253.

Schmidt, R., Lyytinen, K., Keil, M. and Cule, P. (2001), "Identifying software project risks: an international Delphi study", *Journal of Management Information Systems*, Vol. 17 No. 4, pp. 5-36.

Sharma, A. and Gupta, A. (2012), "Impact of organisational climate and demographics on project specific risks in context to Indian software industry", *International Journal of Project Management*, Vol. 30 No. 2, pp. 176-187.

Sharma, H.D., Gupta, A.D. and Sushil (1995), "The objectives of waste management in India: a future inquiry", *Technological Forecasting and Social Change*, Vol. 48 No. 3, pp. 285-309.

Singh, M.D. and Kant, R. (2008), "Knowledge management barriers: an interpretive structural modeling approach", *International Journal of Management Science and Engineering Management*, Vol. 3 No. 2, pp. 141-150.

Singh, M.D., Shankar, R., Narain, R. and Agarwal, A. (2003), "An interpretive structural modeling of knowledge management in engineering industries", *Journal of Advances in Management Research*, Vol. 1 No. 1, pp. 28-40.

Sinha, P.R., Whitman, L.E. and Malzahn, D. (2004), "Methodology to mitigate supplier risk in an aerospace supply chain", *Supply Chain Management: An International Journal*, Vol. 9 No. 2, pp. 154-168.

Thayer, R.H., Pyster, A. and Wood, R.C. (1980), "The challenge of software engineering project management", *Computer Archive*, Vol. 13 No. 8, pp. 51-59.

Villacorta, P.J., Masegosa, A.D., Castellanos, D. and Lamata, M.T. (2012), "A linguistic approach to structural analysis in prospective studies", in Greco, S., Bouchon-Meunier, B., Coletti, G., Fedrizzi, M., Matarazzo, B. and Yager, R.R. (Eds), *Advances on Computational Intelligence*, Part I, Springer, Berlin, Heidelberg, pp. 150-159, 297.

Wallace, L., Keil, M. and Rai, A. (2004a), "How software project risk affects project performance: an investigation of dimensions risk and an exploratory model", *Decision Sciences*, Vol. 35 No. 2, pp. 289-321.

Wallace, L., Keil, M. and Rai, A. (2004b), "Understanding software project risk: a cluster analysis", *Information and Management*, Vol. 42 No. 1, pp. 115-125.

Warfield, J.N. (1977), *Societal Systems: Planning, Policy, and Complexity*, Wiley Interscience, New York, NY.

Warfield, J.N. (1994), *A Science of Generic Design: Managing Complexity Through Systems Design*, 2nd ed., Iowa State University Press, Ames, IA.

Wilmot, W.W. and Hocker, J.L. (2001), *Interpersonal Conflict*, McGraw-Hill, New York, NY.

Yang, J.L., Chiu, H.N. and Tzeng, G.-H. (2008), "Vendor selection by integrated fuzzy MCDM techniques with independent and interdependent relationships", *Information Sciences*, Vol. 178 No. 21, pp. 4166-4183.

Zaheer, A., McEvily, B. and Perrone, V. (1998), "Does trust matter? Exploring the effects of interorganizational and interpersonal trust on performance", *Organization Science*, Vol. 9 No. 2, pp. 141-159.

Zhang, Y. and Dilts, D. (2004), "System dynamic of supply chain network organization structure", *Information Systems and E-Business Management*, Vol. 2 Nos 2/3, pp. 187-206.

Further reading

Standish Group (1995), "The chaos report", *The Standish Group International Inc.*, Vol. 1 No. 1, pp. 1-9.

About the authors

Chitrasen Samantra obtained his Bachelor of Mechanical Engineering Degree from the Biju Patnaik University of Technology, Rourkela, India. He completed his Master of Technology in Production Engineering in 2012 at the National Institute of Technology, Rourkela. Currently, he is a Research Scholar in the Department of Mechanical Engineering, NIT Rourkela, India. His research interests include risk and uncertainty management in supply chain, decision making and information science, multi-criteria group decision making and application of fuzzy set theory in real life decision problems. He is currently pursuing PhD in the area of Risk and Uncertainty Management in Supply Chain. He has published few research papers in reputed international journals/conference proceedings.

Dr Saurav Datta is an Assistant Professor in the Department of Mechanical Engineering, National Institute of Technology, Rourkela, India. His current area of research includes weld quality optimization, modelling and simulation of production processes, multi-criteria decision making, risk and uncertainty management and agile project management. He has published a number of papers in journal of national/international repute and presented a number of papers in various conferences/symposia in India and abroad. He is presently guiding a number of master/doctoral research scholars. He is currently dealing with few sponsored projects. Dr Saurav Datta is the corresponding author and can be contacted at: sdatta@nitrkl.ac.in

Siba Sankar Mahapatra is a Professor in the Department of Mechanical Engineering, National Institute of Technology, Rourkela, India. He has more than 20 years of experience in teaching and research. His current area of research includes multi-criteria decision making, quality engineering, assembly line balancing, group technology, neural networks and non-traditional optimization and simulation. He has published more than 40 research articles in journals of national and international repute. He has written few books related to his research work. He is also dealing with few sponsored projects.

Dr Bikash Ranjan Debata is a Senior Lecturer in the Department of Marketing at the Rourkela Institute of Management Studies, Rourkela, India. He has four years of corporate experience and four years of experience in teaching. His area of interest includes service quality management, cross-culture marketing communication, strategic planning and knowledge management. He has published more than ten research papers in peer reviewed various international and national journals and conferences.

For instructions on how to order reprints of this article, please visit our website:
www.emeraldgrouppublishing.com/licensing/reprints.htm
Or contact us for further details: **permissions@emeraldinsight.com**