A STUDY ON
SOFTWARE DEVELOPMENT PROJECT RISK, RISK MANAGEMENT, PROJECT OUTCOMES AND THEIR INTER-RELATIONSHIP

Thesis Submitted to the

COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

for the award of the Degree of

Doctor of Philosophy

Under

THE FACULTY OF SOCIAL SCIENCE

by

Sam Thomas

Under the supervision of

Dr. M.Bhasi

SCHOOL OF MANAGEMENT STUDIES

COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

KOCHI - 682 022

JULY 2008

CERTIFICATE

Certified that this thesis entitled “**A STUDY ON SOFTWARE DEVELOPMENT PROJECT RISK, RISK MANAGEMENT, PROJECT OUTCOMES AND THEIR INTER-RELATIONSHIP”** submitted to the Cochin University of Science and Technology, Kochi for the award of the Degree of Doctor of Philosophy under the Faculty of Social Science, is the record of bonafide research carried out by Mr. Sam Thomas under my supervision and guidance at School of Management Studies, CUSAT. This work did not form part of any dissertation submitted for the award of any degree, diploma, associateship, fellowship or other similar title or recognition from this or any other institution.

Kochi – 22, **Dr. M. Bhasi**

01/07/2008. (Supervising Guide)

DECLARATION

I, Sam Thomas, hereby declare that the work presented in the thesis "**A STUDY ON SOFTWARE DEVELOPMENT PROJECT RISK, RISK MANAGEMENT, PROJECT OUTCOMES AND THEIR INTER-RELATIONSHIP"** being submitted to Cochin University of Science and Technology for award of Ph.D. degree under the Faculty of Social Science is the outcome of original work done by me under the supervision of **Dr.M. Bhasi**, Professor, School of Management Studies, Cochin University of Science and Technology, Kochi. This work did not form part of any dissertation submitted for the award of any degree, diploma, associateship, fellowship or other similar title or recognition from this or any other institution.

Kochi – 22, **Sam Thomas.**

01/07/2008

ACKNOWLEDGEMENT

I will always remain grateful to God for his abundant blessings during the course of this study. As I move towards the final destination of this remarkable journey of learning, I acknowledge the invaluable contributions of my teachers, family and friends.

This endeavour would not have been possible without the guidance and support of my Research Guide Dr. M. Bhasi, Professor, School of Management Studies. Words are inadequate to express my profound sense of gratitude to him for his patience and support during the course of this study. His suggestions and comments have helped me immensely to be an independent researcher.

My heart-felt thanks are due to Dr. Mary Joseph, Director, School of management studies and Dr. K Krishnan Nampoothiri, member of my Doctoral Committee for their support and encouragement.

I am extremely grateful to Dr. Joseph I. Injodey, Principal, Rajagiri College of Social Sciences, for giving me support and for providing the necessary facilities to complete the study. He was a true mentor who kept motivating me whenever I was slowing down in the race.

I am deeply indebted to all the software professionals, from whom the data for this study was collected, for their co-operation and patience in responding to the research questionnaire.

I remember my family with gratitude, for the love and encouragement I received from them. For my parents Mr. P J Thomas and Mrs. Thresiamma Thomas, it is a dream coming true. My wife Sheeja and our little kids Sandra and Rahul supported me with their hearts and minds.

All my colleagues in School of Management Studies, CUSAT and at Rajagiri School of Management have given me constant support during the period of my study. Among them, thanks are especially due to Dr. Anandakuttan B Unnithan, Dr. Sebastian Rupert Mampilly, Dr. Joshy Jacob, Prof. Veeva Mathew, Prof. Rosemary Varghese, Prof. K.A Zakkaria, Prof. Binoy Joseph and Mrs. Rose Paul.

Lastly, I would like to acknowledge the contribution of those whose names have not been mentioned but who have, nevertheless, played their part in making this research work successful.

Kochi – 22, **Sam Thomas**
01/07/2008

BIO DATA

SAM THOMAS

Pulickathazhe, Plassanal P.O., Palai, Kottayam- 686579

Education

1. MBA(PGDM), Finance & Systems , Indian Institute of Management , Bangalore
   CGPA 3.34 / 4 Year 1997
2. B.Tech,, Electronics Engineering, REC Calicut, 85% (Uni. 3rd Rank) Year 1991
3. Pre Degree from M G University, Kerala, 87% Year 1987
4. SSLC from Kerala, 95% (13th Rank in the State) Year 1985

Professional Experience

From Sept.2002	Associate Professor, Rajagiri School of Management, Cochin
April 1999 – September 2002	Lecturer, School of Management Studies, CUSAT
April 1998 to April 1999	Business Manager, NeST, Technopark
June 1997 to April 1998	Senior Product Executive, RAMCO Systems , Chennai
July 1991 to June 1995	Project Engineer, Electronic Research & Development Centre (ER&DC) Trivandrum

Personal Information

Date of birth : 8/5/1970 Gender : Male Marital Status : Married

Sam Thomas

ABSTRACT

Key Words: Risk, Risk Management, Project outcome

Failure of software development projects is a common phenomenon in many organizations around the world. The Standish Group research shows that a staggering 31.1% of projects will be cancelled before they ever get completed. 52.7% of projects will cost 189% of their original estimates. (Standish survey,1999). Software development project risk points to an aspect of a development task, process or environment which, if ignored, tends to adversely affect the project performance. The project performance can be measured in terms of performance on time and cost dimensions as well as on the product performance (quality of the product developed). Software project risk management is a mechanism for minimizing project risk

Observations from literature show that while many studies on these constructs are done in developed countries, there is scarcity of literature from India. Also, linkages among project risk, risk management and various dimensions of project outcome are generally overlooked in the IS literature. Hence this research was undertaken with an objective of studying software development project risk, risk management, project outcomes and their inter-relationship in the Indian context.

Validated instruments were used to measure risk, risk management and project outcome in software development projects undertaken in India. A second order factor model was developed for risk with five first order factors. Risk management was also identified as a second order construct with four first order factors. These structures were validated using confirmatory factor analysis. Variation in risk across categories of select organization / project characteristics was studied through a series of one way ANOVA tests. Regression model was developed for each of the risk factors by linking it to risk management factors and project /organization characteristics. Similarly regression models were developed for the project outcome measures linking them to risk factors.

Integrated models linking risk factors, risk management factors and project outcome measures were tested through structural equation modeling. Quality of the software developed was seen to have a positive relationship with risk management and negative relationship with risk. The other outcome variables, namely time overrun and cost over run, had strong positive relationship with risk. Risk management did not have direct effect on overrun variables. Risk was seen to be acting as an intervening variable between risk management and overrun variables.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AGFI	Adjusted Goodness of Fit Index
ANOVA	Analysis of Variance
CAGR	Cumulative Average Growth Rate
CFA	Confirmatory Factor Analysis
CFI	Comparative Fit Index
CMM	Capability Maturity Model
COM	Safety Compliance
GDP	Gross Domestic Product
GFI	Goodness of Fit index
HR	Human Resources
IRR	Interrater Reliability
IS	Information System
ISO	International Organization for Standardization
IT	Information Technology
MIS	Management Information System
MNC	Multi National Corporation
MTMM	Multitrait-Multimethod
NASSCOM	National Association of Software and Services Companies
NCR	National Capital Region
PCA	Principal Component Analysis
PCMM	People Capability Maturity Model
PM	Project Management
PMI	Project Management Institute
RMSR	Root Mean Squared Residual
SEI	Software Engineering Institute
SEM	Structural Equation Modeling
SPSS	Statistical Package for the Social Sciences
SME	Small to Medium Enterprise
VIF	Variance Inflation Factor

CHAPTER 1

INTRODUCTION

Section 1.1 presents a short review of the Indian Software industry. Section 1.2 describes the growing concern over software development project failures. The concept of software development project risk and the importance of risk management techniques are introduced in Section 1.3 and 1.4 respectively. Section 1.5 presents the content and organization of the thesis.

Growth and development of Indian software industry has gained worldwide attention and India has established her position among the market leaders in global software development. The Indian IT industry has achieved an iconic status in the Indian economy and is considered a highly significant economic growth engine in India's success. Today software development in the country is a nearly $50 billion strong industry, with a domestic market of $16 billion, contributing around 5.4 % of the GDP and employing over 1.5 million people. The industry has registered a CAGR of around 40 % over the last 6 years. (Source: NASSCOM).

TCS, Wipro, Infosys, Satyam and HCL are the top 5 Indian software companies both in terms of revenue and number of employees. Accenture, HP, Microsoft, IBM and EDC are some of the foreign MNCs who have a big presence in India. Even though the top players continue to lead growth, several high-performing SMEs have also come into the limelight. There are over 5000 registered software companies operating in India.

Service and software exports continue to be the mainstay of the sector contributing as much as USD 31.3 billion. The average size of contracts awarded to Indian firms is going up. Indian Service Providers have grown their share of contracts with values in excess of USD 50 million dollars from 1% in 2002 to 7% in 2006. Major components of IT Services exports include Custom Application Development and Maintenance, System Integration and IT Consulting, Application Management and IS Outsourcing/Infrastructure Management Services. Onsite assignments, where the software professionals with particular technical skills work at the premises of the foreign client, continue to be major revenue earners for Indian companies. However, India is projecting itself as an ideal offshore destination from the cost and execution point of view. The rapid growth in physical infrastructure and communication facilities has aided this process.

Majority of the software companies in India operate in a project mode. A major challenge for companies engaged in software development projects has been successful completion of the projects.

1.1 SOFTWARE PROJECT FAILURES

Project Management Institute (PMI) defines a project as “One shot, time limited goal directed major undertaking requiring the commitment of various skills and resources”. A project, by definition, is a temporary activity with a starting date, an end date, defined goals and responsibilities, a budget, a plan and involvement of multiple parties. Software development projects can be looked at as a category of projects executed with the objective of developing and delivering software products (Pressman, 1997). Software development projects may include new development, modification, re-use, re-engineering, maintenance, or any other activities that result in software products.

A project is usually deemed as successful if it meets the desired requirements, is completed on time and is delivered within budget (Powell and Klein, 1996).

There have been frequent reports of high profile cases of mismanaged software development projects (Charette, 1996). Reports indicate that large numbers of IS development efforts result in systems that do not function as intended, are not used, or are never delivered (Gibbs, 1994; Jones, 1995; Lyytinen and Hirschheim, 1987). The Standish Group research (1999) shows a staggering 31.1% of projects are cancelled before they get completed. Further results indicate that 52.7% of projects cost over 189% of their original estimates. Only 16.2% of software projects are completed on-time and on-budget. Even when these projects do get completed, many are no more than a mere shadow of their original specification requirements (Standish survey, 1999). A study conducted by Ewusi-Mensah and Przasnyski (1991) showed that 35% of abandoned projects are not abandoned until the implementation stage of the project’s life cycle. Gordon (1999) found that, on the average, a company will complete only 37% of its major IS projects on time and only 42% will be completed within budget.

The effects of software project failure are not limited to monetary aspects alone. If the project is intended to provide a company with a strategic advantage over its competitors, its failure could have devastating results on the company's market position, as well as its ability to survive. As software companies continue to invest time and resources into the development of software, a primary area of concern revolves around how software development problems and failures can be minimized.

1.2 SOFTWARE DEVELOPMENT PROJECT RISK

Barki et al. (1993) define software development project risk as the product of the uncertainty surrounding a software development project and the magnitude of potential loss associated with project failure. The uncertainty surrounding a software development project arises from factors that threaten its success (Barki et al., 1993). These factors have been labeled “risk factors” which threaten the successful completion of a software development project.

Most of the researchers on software project risk and risk management broadly agree on a two- step approach to software development project risk management: risk assessment and risk control (Boehm, 1991; Charette, 1996; Lyytinen, 1988; McFarlan, 1981). Risk assessment involves identifying, analyzing and prioritizing the risk factors that are likely to compromise a project's success, and risk control involves acting on each risk factor in order to eliminate or control it (Boehm, 1991). It is apparent that the second step cannot proceed without the first being completed successfully. Managers may be pursuing IS development projects which ultimately result in failure because they are not sufficiently aware of the risk involved. If managers have faulty perceptions of the risk associated, their management efforts are likely to be misdirected and they may unknowingly make risky decisions (Slovic, Fischhoff, and Lichtenstein, 1981).

One of the most common methods for risk identification has been the use of risk factor checklists (Boehm, 1991; Barki et. al. 1993; Schmidt et. al. 1996; Keil et al, 1998). These checklists present a list of all potential risks to the project manager and force him to check and decide which risk factors are applicable in that particular project.

A comprehensive list of software project risk may be obtained by combining the risk factors identified previously in the literature (e.g. Barki et al ,1993) with those factors identified by practicing project managers (e.g. Schmidt et al.,1996).

1.3 IMPORTANCE OF RISK MANAGEMENT TECHNIQUES

Once the risk factors are successfully identified and assessed, the next logical step is to manage the risk (Boehm, 1991). Software project risk management is one mechanism for minimizing project failure (Barki et. al., 1993; Boehm, 1991; Boehm and Ross, 1989; Keider, 1974; McFarlan, 1981).

Research on software risk management has primarily focused on crafting guidelines for specific tasks (Alter et al. 1978; McFarlan 1982; Boehm 1989; Charette 1996). Some of these researchers advocate a continuous view of risk management throughout the development cycle (Boehm, 1991; Alter et. al., 1978). Boehm argues that risk management strategies must be integrated into the software life cycle and has proposed the spiral model as an explicit means of doing so. Some studies recommend a discrete view where specific measures are adopted at selected stages of the development cycle (Davis, 1982; McFarlan, 1982). Because no single risk management framework is all-encompassing, scholars encourage a broad view that incorporates multiple perspectives of risk (Willcocks and Margetts, 1994; Keil et al., 1998; Lyytinen, Mathiassen, Ropponen, 1998).

1.4 CONTENT AND ORGANIZATION OF THE THESIS

Software development project risk and risk management are constructs which cannot be measured as single dimensions. The current research attempts to identify the different components of software development project risk and risk management and check for their impact on project outcome. Following the accepted procedures, validated instruments are developed for measuring risk and risk management. Comprehensive models linking risk, risk management and project outcome are proposed and statistically tested.

The work is presented in the thesis in nine chapters. The remaining eight chapters are organized as follows:

In Chapter 2, a review of literature on risk, risk management and project outcome is presented. Existing research models linking these constructs are analyzed. Observations from the literature review and motivation for the present study are also discussed here.

Chapter 3 presents the various aspects of the research methodology. The initial part of the chapter presents the rationale for the study, objectives of the research, concept models, hypothesis to be tested, variables in the study, scope of the study and sampling design. The second part explains the steps leading to the instrument development including the exploratory factor analysis on the pilot study data.

Chapter 4 builds on the discussion in chapter 3 on instrument development and its empirical validation. It gives the profile of the final sample collected. The results of the dimensionality analysis with Confirmatory Factor Analysis are also presented in this chapter.

Chapter 5 explores risk and risk management constructs further. The hypothesis stated in chapter 3 are tested and the results are presented here.

Chapter 6 explores the link between risk dimensions and risk management dimensions. Regression models connecting each risk component to risk management components and project / organizational characteristics are presented here.

Chapter 7 presents a basic model showing relationship among project outcome and risk. The model has each of the project outcome measures taken as the dependent variable and the risk dimensions as the independent variables.

In Chapter 8, the researcher discusses the use of Structural Equation Modeling (SEM) technique to validate the hypothesized models. It compares various models linking risk, risk management and project outcome and identifies the best fitting model based on SEM analysis. Second order models developed for risk and risk management are tested against their first order models and results are presented here.

Chapter 9 presents a summary of the results and findings of the research. The relevance of the research for practice is discussed. The limitations of this research work and scope for future research are also presented here.

****

CHAPTER 2

REVIEW OF LITERATURE

Section 2.1 presents a detailed literature review covering different facets of software development project risk. Section 2.2 describes previous research on risk management in software development projects. Section 2.3 presents some of the instruments for identification of risk items and risk management items. Section 2.4 reviews previous studies linking risk, risk management and project outcome. Gaps in literature are discussed and motivation for the present study is explained in Section 2.5. The summary and conclusion is provided in section 2.6.

Quality and success of a research is often a reflection of the time and effort invested in developing research ideas and concepts. The immediate goal of a literature survey is to determine whether the idea is worth pursuing or not. The first step of the procedure entails specifying the domain of the constructs. (Pinder, Wilkinson and Demack, 2003). This includes outlining what is included and excluded from the concept under study (Churchill, 1979). Hence this study of software project risk and risk management began with an examination of the literature.

In order to obtain a better understanding of software project risk and risk management constructs, an extensive literature review was performed. It was conducted mainly to identify those features of software development projects which researchers and practitioners have pointed out as factors that increase the riskiness of a development effort and the strategies they adopt to counter these factors. There have been a number of research studies on the issue of “risk in software development” and attempts have been made to classify them into various categories based on their similarities (Sumner, 2000). An extensive amount of literature was surveyed in order to ensure that no important factor was overlooked. In order to identify as many factors as possible, two general resources served as the basis.

First source of literature was articles within IS research which addressed the problems associated with software development projects. Majority of IS articles dealt with the types of problems that occurred in specific phases of the software development process. These articles either used empirical data to draw conclusions as to the effects of a particular risk factor or they proposed models that hypothesized how a few of the risk factors might impact a development effort. These articles taken individually do not provide a clear picture of the spectrum of the constructs. However, they provide a clear picture of the topics which have been studied by researchers. Second source of literature was articles written by practitioners detailing their experiences with software development projects (e.g., Boehm, 1983; Burchett, 1982; Casher, 1984; Keider, 1984; Kindel, 1992). Majority of articles in this group described the author's experiences with a particular development project, or consisted of a summary of their generalized observations from previous studies. These articles tell us about problems that appear to be encountered frequently in software development projects and how these problems can be mitigated.

2.1 SOFTWARE DEVELOPMENT PROJECT RISK

Cambridge Learner's Dictionary defines "risk" as the possibility of something bad happening. Researchers and practitioners in various domains have conducted studies on this topic. Though there are differences in perceptions and approaches to the same, an examination of literature reveals a great deal of similarity in conclusions. Typically, risk is described as some kind of an event that may or may not occur, coupled with a consequence that follows if the event occurs (Dedolph, 2003).

A simple definition of project risk states that it is a problem that has not yet occurred but which could cause loss to one's project if it did (Wiegers, 1998). The concept of risk is associated with a number of human endeavors ranging from space exploration and company acquisition to information systems development (Barki et. al., 1993).

The classical decision theory states that risk is perceived as reflecting variations in the distribution of likely outcomes and their subjective values. Hence a risky alternative is one where the variance is large and risk forms an important factor in evaluating alternative options. Decisions are said to be taken under risk when there is the possibility of more than one outcome resulting from the selection of an option. Furthermore, it is assumed that the probability of occurrence of each is known to the decision maker in advance. The variation in outcomes is said to be a consequence of factors which are beyond his control (Radford, 1978).

Empirical studies on how managers deal with risks show that the managers are not necessarily rational in reacting to risks. They look at a risky choice as one that contains a threat of a very poor performance (March and Shapira, 1987). Also, risk is not a probability concept; it deals with the magnitude of the bad outcome. Accordingly, managers act in a loss-aversive manner rather than a rational manner as predicted by the traditional theory. They seek to avoid risks rather than just accept them. They make fast decisions to avoid risks, negotiate uncertainty absorbing contracts, or just delay decisions if possible (MacCrimmon et al., 1984).

A software project risk points to an aspect of a development task, process or environment, which if ignored tends to increase the likelihood of software project failure (Lyyttinen et al., 1993). Software project risks are a major dilemma to information systems project managers (Jiang et. al., 2000). The reasons for variations in success can be attributed to risk factors which are technical, economic and behavioural in nature (Barki et. al., 1993; Lyyttinen, 1988). Such incidents pose danger to the development of a successful project leading to inadequate software operations, software re-work, implementation difficulty, delay or uncertainty (Boehm, 1991). McFarlan (1981) viewed project risks as failure to obtain all of the anticipated benefits because of implementation difficulties, much-higher-than-expected implementation time, and thus resulting in the development systems whose technical performance is considerably below estimates.

To summarize, risk has two components

1. The chance / probability that an undesirable event will occur.

2. The negative consequences / magnitude of loss because of the occurrence of this event.

Boehm (1989) defined Risk Exposure (RE) combining these two components as:

RE = probability of an unsatisfactory outcome * Magnitude of loss arising from this outcome.

Sherer (1994) viewed that software project risk could be estimated from the possibility of failure multiplied by the magnitude of its loss. Similarly, Rainer, Snyder and Carr (1991) described risk as a function of the vulnerability of an asset to a threat multiplied by the probability of the threat becoming a reality.

A precise calculation of the probabilities of negative outcomes and their magnitude is necessary in order to calculate risk exposure. However, there are numerous complexities in software development that make a proper estimation of outcome probabilities hard. Hence, assessing risk via a quantitative evaluation of probabilities could be very difficult and unreliable (Kaplan and Garrick, 1981).

In lieu of estimating the probabilities of a negative outcome, an alternative method has been devised. Kangari and Boyer (1989) adopted a method of risk assessment based on the use of natural language. Accordingly people were asked to express in a natural language, the relative weight and severity of loss arising due to the identified risk factors.

Barki et. al. (1993) put forth a modified definition of software project development risk by referring to the uncertainty surrounding a project.

As risk is a potential problem, an effective step to risk management would be through proper identification of risk factors (Fairley, 1994). The process of risk analysis can be broken down into three; risk identification, risk estimation and risk evaluation. This information enables managers to take steps to avoid potential problems before they become crisis situations. The initial step in the research process is the identification of potential software risks.

The extensive literature review resulted in the identification of over 100 risk factors. The next step was to try to group similar factors together in order to get a clearer picture of the general types of software project risk factors. This resulted in the creation of 12 general types of software project risk categories.

1. Team related factors
2. Effectiveness of Project Communication
3. Project Manager Characteristics
4. Organizational Climate and Support
5. External Factors
6. Role of the user
7. Formalization of project charter

8. Project estimation and planning
9. Tools and technology
10. Requirement stability and accuracy
11. Effectiveness of Project Monitoring
12. Cross cultural and gender issues

This is a list of potential risk categories associated with software development projects which tend to increase a project's likelihood of experiencing problems. Each of them is discussed elaborately in the following sections.

2.1.1 Team Related Factors

Team related risk items are very common in software development projects. These issues include frequent shuffling of team members, a highly diversified team, employee attrition, lack of skills among members, conflicts in the team, level of staff motivation and improper definition of responsibilities.

Employee turnover is a much studied phenomenon (Shaw, 1998). Team member turnover results in huge costs for the company, in terms of both direct and indirect costs. Direct costs are incurred on account of replacement and recruitment. Indirect costs arise out of pressure on remaining staff, costs of learning, product or service quality, organizational memory and the loss of social capital (Dess & Shaw, 2001). Considering the high costs associated with replacing IT staff and the value of their experience, companies need to devise mechanisms designed to keep IT staff longer (Mak and Sockel, 1999; Moore, 2000; Campion, 1991). Studies in an engineering services company revealed that employee turnover was noticed more among professionals who faced stress due to changing technologies and company requirements (Rao, 2006).

Studies have proved that higher levels of diversity within a team results in high levels of conflict, which is counter productive. Many projects also suffer from overstaffing where project teams are staffed with large numbers of unnecessary people who waste resources and add coordination problems (Corso, 1993). However studies conducted by Thomas (2000), Katzenbach and Smith (1993) have suggested that diversity provides an environment that encourages every individual to contribute her/his own ideas. This in turn leads to higher motivation and consequently higher quality team output.

"Team member skills" refers to the level of experience, knowledge, and skills that software development team members have. The literature has identified a range of skills that team members may not possess, but that can impact the development project. They include lack of experience and lack of software development knowledge which are critical for project success (Carmel and Sawyer, 1998; Curtis, 1988).

Team member experience is a crucial factor for the success of a project. It has been observed that those with a higher level of design skills and experience produce better design (Beaver et. al., 2006). The standard of work may turn out to be below the expected levels owing to lack of ability and experience of staff (Cooper, 1994). This lack of know – how can extend to hardware, operating systems and other software (Fuerst and Cheney, 1982). Charette (1996) observes that the presence of inexperienced personnel can lead to troubled software development projects.

The level of application experience possessed by the development team has also been identified as a critical risk factor in the IS literature. There are instances where software professionals do not possess the right skills or background to understand the business requirements or apply the right tools to model and produce the corresponding systems (Morgan, 2005). Jones (1994) identifies lack of application knowledge among project team members as a major cause for project failure. Casher (1984) argues that when project team members do not understand the application they are developing, the likelihood of software failures increases.

The quality of people in software teams is one of the most important factors in improving productivity and quality in software projects (Blackburn et al. 1996). Lack of adequate and frequent training adversely affects capabilities (Dominiak, 2006; Cappelli, 2000).

Conflicts among team members hamper the development of a software project. The reasons for conflicts may be task based, process based and relationship based (West, 1994). The most frequent source of conflict is usually disagreement over goals and work processes, which is often a result of communication gaps between team members. Such conflicts cause severe damage to team performance (Heckhausen, 1989). Collaborative problem solving frequently leads to disagreements among individuals or groups which can lead to conflicts within an organization. Robey et al. (1989) came across instances of open confrontations among software development team members in a troubled project.

The motivation level of the project team has a direct impact on the project. There are many studies linking motivation to productivity increase. A number of factors including reward structures and performance measures affect the motivational levels (Jerome and Kleiner, 1995). Gilb (1988) placed importance on staff motivation in determining whether project goals tend to be achieved satisfactorily. A team where members have a less satisfactory salary package, where promotions are conspicuous by their absence, staff motivation tends to be low. Setting deadlines and consequences for not meeting the same, is another way of motivating employees. Mc Comb and Smith (1991) also propose that the level of motivation of staff members determines the pace of their work. Reward and recognition are considered by organizations and managers as an important element in motivating individual employees (Cacioppe, 1999). High employee turnover, which is considered to be detrimental to organizational performance, is often the result of lack of reward for good performance and lack of opportunities for career advancement (Newcomb, 1999).

Another factor that has an impact on software development risk lies in the manner in which responsibility is assigned to each team member. Several sources have mentioned that if the responsibility of each team member is not clear, the project may have problems. Projects are likely to fail if each member is unclear about her/his responsibility (Evans, Piazza and Dolkas, 1983). One of the biggest problems in software project management has been to establish a proper accountability structure (Boehm, 1979). In other words, team members are often disorganized and managed with a very vague delineation of responsibilities. Thayer, Pyster and Wood (1980) found that an uncertainty surrounding the responsibility for various project functions led to a negative impact on the success of the project.

2.1.2 Effectiveness of Project Communication

Communication is a vital issue in software development projects. The three major stakeholders – the customer who financially sponsors the project, the contractor who performs the system development and the eventual user of the system – have to communicate regularly during the project (Deutche, 1991). Keider (1984) in his survey of MIS professionals found that ineffective communication was the single major factor which caused most number of software failures. Communication may breakdown because of different reasons including disagreement over goals and objectives, preconceived notions and opinions often leading to prejudice, semantic differences and even misunderstandings in non-verbal communication (Staehle, 1999).

Curtis (1988); Corbato and Clingen (1979) have indicated that communication problems are more likely to occur if there is a geographic separation among members. They view that projects involving multiple cities or countries are more likely to have problems. There also exist a number of management problems between the off – shore and on – site teams.

Lack of English communication skills is another major factor which affects the performance of a project team. Research conducted by India's National Index of Communication Skills shows that only 10 percent of IT applicants have adequate language skills (www.itbusinessedge.com). Good communication skills, especially spoken English skills, are mandatory and important in today's global business environment.

Organizations need to make provisions for effective intra organizational communications. There are a number of interfaces which enable the same (Malone, 1985).

Documentation is an integral part of a software system. It contains the information that is necessary to effectively and successfully develop, use and maintain a system. However in practice, the creation of appropriate documentation is largely neglected (Bayer and Muthig, 2006). Upon completion of the project, members are spread all over the company without adequate documentation of the essential components. This leads to loss of knowledge and experiences. Weiser and Morrison (1998) are of the view that project information is rarely retained in the appropriate manner by which it can be used for future reference. Project documentation should support communication during the project and address the information needs of various people— project members, project management, project supervision (Disterer, 2002). From the user’s perspective, the completeness of documentation with respect to the user’s tasks is an important indicator of the quality of software developed (Bayer and Muthig, 2006).

Teleconferencing and video conferencing systems and services are the main set of technologies developed to support group work (Sabri and Prasada, 1985). Unreliable telecommunication facilities such as slow internet connections or telephone lines, slow computer networks etc may seriously hamper the progress of work. Software companies in India face a number of telecommunication problems including shortages of telecommunication links, time lag in accessing such links, poor transmission and high cost of installation and use. Such limitation on telecommunication facilities has discouraged many foreign clients from considering offshore software development in India as a feasible option (Chakraborty and Dutta, 2001).

2.1.3 Project Manager Characteristics

The project manager plays an important role in the success of a software project (Mc Donough, 2000; Brown and Eisenhardt, 1995). The project manager should have a clear understanding of the issues, have an adequate grasp of relevant technologies and be capable in the political sphere (DeMarco, 1982).

An important component of providing a work atmosphere in which employees can perform well is the communication style of the project leader (Thacker and Yost, 1997). An effective manager understands the effects of mutual interactions between different parts of the project and steers them in the direction of continuous learning and adaptation (Augustine, Payne, Sencindiver and Cock, 2005).

Discharging the leadership tasks effectively in the challenging business environment of today requires experienced hands. It is not merely a question of following a process (Goleman, 2000). In the absence of adequate experience, the leader is unable to guide his team-mates properly. Brown and Eisenhardt (1995) and McDonough (2000) highlight how the leader’s people management skills affect project development. A project manager should benchmark himself against other leaders to measure his performance from time to time (Townsend, 2006).

Literature also mentions other risk factors such as personal bias in the selection of team members and multiple projects in hand to be completed, which adversely affect software development (Wallace, 1999).

2.1.4 Organizational Climate and Support

Organizational politics has been discussed with utmost importance in the literature over the last two decades. It is considered to be a primary component in contemporary business practices (Aronow, 2004). Organizational politics refers to behaviors that occur on an informal basis within an organization and involve intentional acts of influence which are designed to protect or enhance individuals' careers when conflicting sources of action are possible (Drory, 1993; Porter, Allen and Angle, 1983). Several sources have indicated that political conflicts and power plays can increase software project risk and negatively affect its outcome. Jones (1994) on the basis of his experience with cancelled projects estimates that corporate politics has been associated with more than one- third of them.

There are various risk factors which are related to organizational climate. The lack of top management support has been cited as a possible risk factor in software development projects (Boehm, 1989). Ewusi – Mensah and Przasnyski (1991) have said that it is an accepted principle or maxim in IS literature and practice that senior management support is essential for the success of software project. It has also been suggested that top management is necessary for successful project development (Jarvenpaa and Ives, 1991; O'Toole and O' Toole, 1966; Walton, 1989). Sauer (1993) suggests that support should come from senior management, user managers and from the organization itself.

Organizational politics may showcase themselves in a number of ways. Fox (1982) indicates that development projects will encounter a number of problems if organization pressurizes team members to keep development costs low. Further, Burchett (1982) notes that project managers faced problems of arbitrary and unrealistic deadlines. Schedule and budget should be based on a clear assessment of achievability. Organizational issues such as team building, group dynamics, project organization, and inter organizational relations are equally important (Nash and Redwine, 1999).

An unstable and unprofessional corporate environment could add to software project risk. It manifests itself through competitive pressures which radically change user requirements, making an entire project obsolete. Research has recognized and confirmed the existence of a work culture of long working hours coupled with work intensification and heightened levels of occupational stress. Work pressure and stress can lead to deterioration in the efficiency of staff. This can have an impact on the atmosphere in the workplace and the quality of the work produced.

Organizations with little or no experience implementing large projects face a number of risks such as not being able to access a similar industry benchmark, committing resources to an unsubstantiated plan and timeline etc. Hence they need to conduct a detailed scoping exercise which would define business and functional requirements (Kesner, 2006).

Risk is said to increase if tools and resources such as facilities, personnel, funding, finance, technology etc are not available for the project (Command, 1988). Humphrey et. al. (1991) suggests that organizations need to make development tools and expertise available to software project teams. Tait and Vessy (1988) found that if the time and funds available to complete a project are insufficient, system designers would resort to short cuts rather than undertaking the normal development procedures. This raises the risk of project failure. The limited availability of financial and managerial resources is one contributing factor to problems in software development (Anderson and Narsimhan, 1979). Keider (1984) notes that changes take place when the scope of the project is redefined, the availability of resources changes, or the cost of equipment and/or personnel changes.

2.1.5 External Factors

External factors refer to the interdependence on third parties such as vendors, which may affect the ability of the internal development team to complete a software development project on time without problems.

Boehm (1989) indicates that shortfalls on the part of sub – contractors may be a major source of risk. These include supplied components which are a poor match to a new application, mandated support tools and environments that are incompatible or poor in performance or functionality and commercial components that turn out to be immature and poorly supported.

Dependence on external agencies affects the progress of software projects (Rothwell and Dodgson, 1991). Risk associated with a software development project increases with the number of hardware and software suppliers. Products from different vendors are often incompatible and a system composed of equipment from assorted vendors makes the diagnosis of problems extremely complex (McFarlan, 1981; Ruthberg and Fisher, 1986; Kemerer and Sosa, 1988). Schmidt et. al. (1996) concluded that lack of control over consultants, vendors and sub contractors leads to problems regarding the schedule and quality of the outcome. This problem is further aggravated when there is no legal recourse due to poor contract specification.

External risk also includes risk factors arising out of interactions with regulatory and government agencies. For example, Indian workers are facing more visa restrictions, especially in the USA which largely affects the work assigned to them (www.bizasia.com). This causes the risk of work not being completed on time.

2.1.6 Role of the User

Boehm and Ross (1989) state that many software problems surface because users are unaware of the possible results that can be achieved with information systems and this in turn leads to a set of unrealistic expectations. Research on this topic from a risk perspective found user-related risks are a key threat to successful software development projects. These risks include unwilling IS users, lack of user support, lack of user experience, user shirking responsibility, user resistance to change and sheer number of users. Failure to assess the user-related risks and to adapt management methods accordingly can be a major source of system failure (Jiang et. al, 2000).

Information systems become obsolete as the mapping between user requirements and software deteriorates. Monitoring reliability of business software can provide an indication of how well user requirements are currently mapped to the corresponding information system (Heales, 1995). The process of obtaining user requirements varies according to the situation prevailing in the organization (Newman and Sabherwal, 1991). Keider (1984) says that if no effort is made to know what client requirements are, it may adversely affect the final outcome of project. Robey and Farrow (1982) indicate that an assessment of user needs and resulting client support aids in better understanding of the system.

Other sources have also suggested that project failure may be the result of inflated or unreasonable user expectations. Ginzberg (1981) identifies five areas of expectation which are likely to be important in determining user acceptance: the expectations pertaining to the goals and objectives of the systems, the importance of the problem being addressed, the way the system will be used, the impact the system will have on the organization, and the criteria that should be used to evaluate the system.

Lack of software experience on the part of the client is another major risk factor that affects its outcome (Barki et. al. 1993; Mc Farlan, 1981). Without adequate knowledge of the application, user requirements tend to be misstated leading to irrelevant results.

The general attitude of the client is a factor which influences project risk. Client attitude refers to unwilling users, resistance towards a new system and their feelings of responsibility (Alter, 1979; Alter and Ginzberg; 1978; Lyytinen et. al. 1993; Tait and Vessy, 1988; Andersen and Narsimhan, 1979; Charette, 1996). The users must feel a need for a system without which it cannot be successfully implemented (Guthrie 1974).

Communication obstacles between users and system developers are caused by a lack of understanding, as users and system developers have different jargons and terminologies (Junhe, 2004).

2.1.7 Formalization of Project Charter

The formalization of project charter forms an integral part of the software development process. Brooks (1995) says that defining the goals of the project is a difficult task. If goals are ambiguous and ill-defined, development problems may be encountered (Lyytinen and Hirchheem, 1987; Schmidt and Kozar, 1978). A clear cut goal is to be initiated failing which programmers would end up making their own assumptions about the project. Gaining frequent feedback from the customer helps promote synthesis of different points of view, but if the project goals are not well defined from the outset, individuals will tend to focus on their own goals (Jiang et. al., 2000).

Incidence of project failure is high in situations where the project is not adequately defined (Keider, 1984). Kindle (1992) suggests that, in order to prevent failures, a company should have a clear understanding of the project objectives and there should be a clear consensus on how the end results should be. Doll (1985) indicates that successful firms are more likely to have reached an agreement between top managers and project manager on a set of criteria for the development priorities. If multiple organizations are working on a project then differences in organizational goals may cause problems (Corbato and Clingen, 1979).

2.1.8 Project Estimation and Planning

A common problem encountered in software development projects is the incidence of poor estimation techniques for determining a project's budget and schedule.

There are several reasons why estimation poses a problem in software development projects. Non-existence of standards for estimating the time period of a project is the most common reason. Estimation also becomes a problem if it is not done by the project leader, but by anyone who is available to do so (Keider, 1974). It may also be difficult to estimate schedules if personnel availability for the project is unknown. An estimator needs to know who will be working on a project in order to estimate its duration of completion (Keider, 1974). DeRoze and Nyman (1978) indicate that inadequate estimates may occur because there are no simple rules for predicting software costs accurately and estimates generally do not allow for anticipated problems and changes.

According to Farquhar (1970), in an absence of accurate estimates managers do not know what resources to commit to a development effort. This often results in cost and schedule overruns. Poor estimates can lead to excessive schedule pressure or unrealistic schedules that can increase project risk (Boehm, 1987; Jones, 1994; Jones, 1995; Lyytinen et al., 1993). Tait and Vessey (1988) indicate that the likelihood of project failure increases in situations where severe financial or time constraints are present. Whittaker (1999) found that incorrect estimation leads to unrealistic project deadlines especially when the new development tools are difficult to understand.

Poor planning is another problem which has been identified in many software development projects. McCarthy (1979) notes that even if goals are clearly specified development projects often suffer from inadequate planning. Metzger (1981) based on his survey on unsuccessful projects, states that approximately half of them failed because of poor planning. Doll (1985) suggests that firms with more successful IS development efforts are almost three times as likely to have a written overall plan for systems development. Plans should be made for defect prevention (Jones, 1995), task assignments (Keider, 1984), establishment of milestones (Keider, 1984) and for backup and disaster recovery (Kindle, 1992).

2.1.9 Tools and Technology

Inappropriate methodology and tools used in the project is cited as a reason for the project not delivering what it is designed to deliver (Cash, McFarlan and McKenny, 1988). The approach that fits one kind of project very well may not be well suited to others.

The appropriateness of selected tools and technology refers to the risk of choosing hardware, software, language, methods, tools, etc. that are a bad fit with the task or team members. Thayer et al. (1980) have indicated that decision rules for selecting the correct software design techniques and strategies to be used in designing and testing software are generally not available. There are several reasons why selected technology and tools may be inappropriate for a project. Technology may be inappropriate and increase project risk because of inadequate physical capabilities. For example, if the selected equipment is unable to handle the transaction volume then the likelihood of project failure increases (Casher, 1984). The system used may turn out to be incompatible with the software selected for the project (McFarlan, 1981).

The mandatory use of inappropriate or unsuitable technology causes problems in a number of software projects (Command, 1988). Such instances, which are the result of political pressure, pose a risk to the project being developed (Jones, 1994). A certain technology is likely to be selected due to organizational politics and company rules (Boehm, 1989a; Command, 1988).

The selection of appropriate development tools is an area of concern in literature. Scacchi (1984) says that most companies do not have adequate software development tools and production methodologies. Anderson and Narsimhan (1979) observed that inappropriateness of development methodology was the risk factor having the greatest impact on project success. The respondents to Schmidt et al’s (1996) Delphi survey observe that the lack of an effective development methodology tends to cause quality problems resulting from poor estimation and relatively less flexibility for change.

Several sources have suggested that organizations should change their tools and techniques from time to time. If the technology is unreliable, inefficient, or exhibits problems, project outcome may be doubtful (Boehm, 1981; Command, 1988; Fairley, 1994; Lyytinen et. al. 1993). Technology may also be inappropriate if it is not easily modifiable (Casher, 1984).

2.1.10 Requirement Stability and Accuracy

Fox (1982) suggests that even if management can specify what users want, requirements are not likely to remain stable. Jones (1994) suggests that 80 % of all projects face the risk of creeping user requirements. Creeping user requirements refer to changes in requirements which occur after the formal requirements phase.

Some of these changes are essential but most projects underestimate the ripple effects that a requirements change has on design, coding, testing, personnel, assignments, communication, budget schedule and performance (Boehm, 1991). Requirement changes can result in excessive schedule pressure. Numerous authors have indicated that frequent fluctuations in requirements can increase project risk and uncertainty

Boehm (1989) identifies gold plating as another type of change in requirements. Gold plating occurs when complex requirements are added on to the project. The job gets bigger and disproportionately more expensive as more and more features are added to the project. The final outcome is not of much use to the user.

Changing requirements result in unsuccessful projects. According to Bruce and Pederson (1982) a change in requirements makes it difficult to estimate software potential cost. Project schedules which overlook requirement changes may result in budget overruns (Schmidt et.al., 1996). Ruthberg and Fisher (1986) observe that the chances of project failure loom large in a situation requiring a large number of requirement changes.

2.1.11 Effectiveness of Project Monitoring

Software development problems are also attributed to the fact that adequate time was not spent on various phases of software development. Thayer (1980) notes that techniques and aids for the same are not generally available. In order to control a large project, there must be a list of milestones which are clear, concrete and measurable (Brooks, 1995). A project needs to have an unbiased feedback in order to accurately monitor its progress (McComb and Smith, 1991).

Boehm and Ross (1989) stress upon the fact that once a development plan has been made, an effort should be taken to see that work progresses accordingly. The need for reviews at frequent intervals has been emphasized by practitioners in software development (Keider, 1974; Humphrey et al, 1991; Kindle, 1992). However, managers tend to overlook this fact (Boehm, 1979).

Henderson and Lee (1992) have suggested that problems may occur if a project manager lacks an adequate standard against which the quality of work done is to be compared (Scacchi, 1984). Keider (1984) has noted that between the completion of the feasibility study and formulation of the work plan, there could be a change in estimates which necessarily has to be reflected in the estimated time and budget, or else the project may fail. Projects should be able to accommodate changes in requirements in the project, in order to succeed (Boehm, 1983; Ruthberg and Fisher, 1986)

2.1.12 Cross Cultural and Gender Issues

Culture may be defined as a shared set of beliefs, attitudes, norms, values, and behaviour organized around a central theme and found among speakers of one language, in one time period, and in one geographic region (Triandis, 1995).

This particular risk factor discusses the significance of developing an awareness of cross-cultural factors in software development. It is rare for organizations to bother with the details of how personnel at lower levels in the hierarchy will run meetings, make decisions, solve problems, manage staff and communicate proposals. Yet, people from different cultures carry out all these procedures in diverse ways. The trouble is that each culture assumes their way is the correct one. Unexplained deviations from these norms are perceived as deviant and even devious (Pooley, 2005).

Language is a major issue, and perceptual differences in understanding language lead to communication barriers. Though English is the universally accepted language, there are instances where the same word denotes a different meaning to people. For example, ‘sanctioned a project’ means ‘allowed a project’ to the British and ‘prohibited a project’ to the Americans. Likewise, Asians prefer to work in groups and strongly value collectivism. However Westerners are more individualistic by nature. Further there are differences in risk taking capacity, techniques of conflict resolution, general business etiquette, attitude to relationship etc. These together affect efficient functioning of the project team which comprises of members from various nationalities

Joan Mann et.al proposes a research model linking cultural traits to risk factors in IS projects. Based on the Hofstede's (1991) dimensions of values frame work, they argue, taking Thailand as an example, about how the intensity of IS project risk factors vary based on the difference in cultural dimensions.

Failures in IS projects are also caused by ethical issues. Cultural differences may give rise to divergent perceptions of ethical issues and contradictory approaches to ethical reasoning (Oram and Headon, 2002).

Women now enjoy equal opportunities to higher education with men and hence the gender ratio at workplaces is more or less equalized. Given the growing presence of women in the workforce, gender specific issues need to be given more attention. Women employees have to face pressures of unearthly work timings (hindu.com). Compared to men, women's time commitments to paid employment are more influenced by the need to reserve time for dependent care and other family responsibilities (Fagan, 2001). There is overwhelming evidence that the burden of dependent care falls on women (Buffardi et al., 1999). Women respondents claimed to have suffered physical ailments as a result of working long hours. Other negative aspects about this job include traveling at night and traveling abroad and the social stigma attached to it.

2.2 RISK MANAGEMENT PRACTICES

This section provides an overview of the risk management practices suggested to address the risk factors identified in software development projects. Risk management is concerned with a phased and systematic approach to analyse and control the risks occurring in a specific context (Charette, 1996). The predominant purpose of risk management is to take the appropriate course of action to strike an optimal balance between likely benefits of such techniques and the exposure to risks (Powell et. al. 1996). Software project risk management is risk management applied to the development and/or deployment of software-intensive systems. Considerable improvements can be made in the software development process through the systematic applications of risk management techniques and guidelines (Alter et al. 1978; McFarlan 1982; Boehm 1989, 1991). Research on software risk management has primarily focused on crafting guidelines for specific tasks (Alter et al. 1978; McFarlan 1982; Boehm 1989; Charette 1996).

Boehm (1991) defined risk management as an emerging discipline whose objectives are to identify, address, and eliminate software risk items before they become either threats to successful operation or major sources of rework. A formal risk management programme is a structured way of evaluating risks to the software development process. A typical risk management framework involves implementing and monitoring measures to reduce risk. Project risk management encompasses both hard skills such as estimating and scheduling tasks, and soft skills, which include motivating and managing team members (Kirsch 1996).

Risk management strategies use observations from the past; they learn from analogical situations, and they use deductive reasoning to detect risky incidents. Over time, observations are generalized by crafting specific theories of cause-effect chains into generic risk items. In addition, risk management approaches feature a repertoire of risk resolution techniques. These are derived from local causal theories on how risky incidents affect software development and how interventions affect development trajectories. The techniques help formulate schematic plans for interventions that decrease the likely impact of risky incidents, or avoid it altogether.

A thorough review of literature on risk management strategies for software projects, helped to identify a range of risk resolutions techniques which are discussed under nine categories, namely;

1. Leadership Strategies
2. HR Policies
3. Training
4. Project Coordination
5. User Coordination
6. Requirement Management
7. Estimation Techniques

8. Appropriate Methodology
9. Project Control

Each of them is discussed elaborately in the following sections.

2.2.1 Leadership strategies

Making informed decisions by consciously assessing what can go wrong as well as the likelihood and severity of the impact is at the heart of risk management. The need to manage risk increases with system complexity (Higuera et. al. 1994; Haimes, 1991).

Communication lines, the structure of authority, and lines of accountability are significant in organizing the risk-based management process. It is important to effectively communicate risks to everyone involved and to reward reporting of omissions and errors (Keil, 1995). The management structure is also important to create a sense and discipline of accountability (Rochlin, 1993). Obtaining the top management commitment is also a crucial factor (Alter and Ginzberg, 1978). This can be achieved through increasing payoffs associated with successfully completing the project, creating opportunities for senior managers to publicly display their support for the project, and aligning the project with other goals that are viewed as central to the organization (Ropponen and Lyytinnen, 1997). Management should pay attention to the organizational behavioral aspects of the project and build a supporting base for the project within organization so that the project would not be abandoned even if the main advocate was to leave (Ewusi-Mensah et.al, 1991).

McFarlan (1981) suggested selection of a technical expert as the project manager to lead the team. The technical back ground of the project manager plays a very important role in risk minimization (Ropponen and Lyytinnen, 1997). The project manager/leader is to be backed by an assistant leader, who works closely with the project manager and is able to take over in his absence (Jurison, 1999). Finding skilful, open-minded software managers is instrumental in improving the risk management capability (Boehm, 1989). In order to reduce the need for self-justification, project managers should be rotated in and out of projects so that the people who initiated the project are replaced with people who will naturally have greater objectivity (Keil, 1995). Keil 1995 argued that the failure of the project could be avoided if the project managers adopt a broader view of project management spanning both the rational approach and the psychological approach.

2.2.2 HR policies

There is extensive empirical literature investigating the relationship between HR policies and organizational performance. The relationship between HR and business results is built on a simple premise that better deployment and use of HR should correlate with higher business performance. Those organizations where the HR department is supportive and helpful, encounter fewer incidents of software failure (Evans, 1986; Huselid, 1995; Ulrich, 1997).

Job matching is an essential feature of risk management. Matching an individual to a particular job is based on the idea that personnel differ to a considerable extent in abilities relevant for the successful performance of a task (Zeidner, Johnson and Scholarios, 1997). Tools which are used to match computer professionals to jobs include assessment centres and simulations (Nash, Redwine, 1999). Personnel systems designed around ability profiling have been shown to enhance organizational productivity and translate into economic benefits for employers (Hunter and Schmidt, 1982). When individuals are considered for a position they should be consulted before hand, lest it results in a problem (Anastasi and Urbina, 1997; Cronbach and Gleser, 1965). It will be easier to trace and correct errors if accountability is established for a particular task (Zmud, 1980).

Attendance systems are designed to tackle employee absenteeism and track the number of hours employees work. Flexibility in working time includes a variety of arrangements for employees such as part-time work; job sharing, flexi-time, fixed-term contracts, subcontracting and career/employment break schemes (Papalexandris, Kramar, 1997). These arrangements have been introduced for a variety of reasons which include economic factors, improvement in productivity and competitiveness, timely completion of work etc. (Brewster, 1994). Flexible hours of work and employment schedules affect family and employee satisfaction (VandenHeuval, 1993). Dissatisfaction with the job, stress and poor health have been found to be outcomes of heavy work schedules (Karasek et al., 1981; Karasek, 1979).

Bonuses and commissions do have a motivational effect on employees. Targets appear all the more achievable if there are a series of rewarded steps in the form of promotions and incentives (Garbett and Morton, 2006).

2.2.3 Training

The growing demand for software development requires increasingly productive people. For effective project management, team members need global experience and adequate training (Collins and Kirsch, 1999).

To prevent large variations in employee performance, effective methods of training should be adopted. Training is gaining importance owing to the magnitude of the problem of producing a sufficient number of well-trained software engineers (Martin, 1981). Training of employees is regarded as one of the most important functions of efficient resources management (Prytherch, 1986). Adequate and frequent training increases individual capabilities (Dominiak, 2006).). Effective training in technical aspects, project management, communication and other relevant areas can help newcomers become more effective (Deephouse et. al, 2005).

Training on risk management is essential for project managers. Symptoms such as “escalating commitment” (Keil, 1995), the “no-problem syndrome”,” risk aversion” etc can be improved by increasing their awareness of risk management methods (Boehm, 1989; Keil, 1995)

Firms are just beginning to emphasize a ‘software development team culture.’ Sawyer and Guinan’s (1998) study of a major software manufacturer revealed that exposure to group dynamics, conflict management, and listening skills seminars helps to develop this culture among software developers.

2.2.4 Project coordination

The project requires a series of coordination measures. Internal integration focuses on coordinating the project team members while external integration focuses on the external agencies (MaFarlan, 1981).

Projects involving new technology should rely more on internal integration tools that are designed to enhance the team’s technical competence and operation as an integrated unit (Jurison, 1999). Software project management requires different types of coordination at different stages and a major portion of the organizational design problem is choosing the particular type of coordination that matches the given uncertainty. Given the temporary nature of software projects (project teams are usually dissolved once their objectives are achieved) minor slippages in the control process can have a greater adverse impact than the same slippage in a more permanent organization (Nidumolu, 1995). Zmud (1980) suggests an impersonal mode of project coordination for low risk/uncertainty projects and group mode of coordination for high risk projects. This is endorsed by Alter and Ginzberg (1978) also. Nidumolu (1995) recommends vertical mode of coordination in high risk projects. Barki et. al's study (2001) shows higher levels of formal planning in high risk projects when cost control is taken as a measure of performance. But when system quality is studied as the performance objective, it is seen to have positive correlation with user planning (Barki et. al., 2001).

Nidumolu (1995) recommends adoption of a risk-based perspective rather than a structural perspective for coordination of software projects. He argues for exerting higher level of leadership and authority (vertical control) to ensure that the objectives are met. The risk-based perspective of software project coordination also takes into account the temporary nature of software projects and teams.

Coordinating with suppliers and sub-contractors is also an important risk management task (Boehm and Ross, 1989). In order to have the whole-hearted cooperation of sub contractors and suppliers, companies need to maintain excellent relationships and operations with them. The open minded attitude in the interactions with suppliers will enable companies to understand their needs better and handle conflicts better (Wong, 2002).

2.2.5 User coordination

Proper communication systems should be designed to integrate users into the development environment. Some of the strategies recommended include selection of a user as the project manager, creation of a user steering committee, frequent and in-depth meetings of this committee, a user-managed change control process, training of the user, frequency and detail of distribution of the minutes of the project meeting to key users, selection of users as team members, formal user specification approval process, progress reports prepared for corporate steering committee, giving users the responsibility for the installation of the system and letting users manage decisions on key action dates (McFarlan,1981; Boehm, 1991). Another measure is to insist on the mandatory use of the system developed and rely on diffusion and exposure (Alter and Gintzberg, 1978).

The risk that arises out of a lack of proper user commitment and support can be minimized by establishing the right contacts and creating a “home base” in the user organization (Alter et. al, 1978). User involvement during the definition stage of the project will also help to make users more realistic in their expectations of the outcome (Ginzberg, 1981).

The key to minimizing user risk includes user involvement, user participation and a clear statement of user requirements (Jiang et.al., 2000; Lin et al., 2004). Teaming with customers helps to reduce the overall risk to both sides. Formation of a risk team comprising of the developer, customer, systems architecture group, and contract management create a forum for open discussion of risks that crossed organizational lines (Dedolph, 2003). Projects with relatively little structure can benefit from external integration tools that create effective links between the project team and the client’s organization (Jurison, 1999). To integrate the technical and end-user perspectives, Kokol et al. (1991) put forward three propositions. First, during the requirement specifications phase, analysts must understand the user requirements and the environment in which the software will operate. Second, during the implementation phase, all parts of the specifications should be implemented correctly. Third, the implemented system must be validated, i.e. software developers must ensure that the implemented system represents a correct mapping of the specifications

Another useful practice is to have the client’s representatives participate in technical reviews to assure a common understanding of client needs and avoid future surprises (Jurison, 1999).

Project managers should be able to create and maintain long-term relationships with users and promote user commitment to the project (Addison and Vallabh, 2002). Boehm and Ross (1989) argue that the primary job of the software project manager is to structure the project to meet the ‘win’ conditions of various stakeholders.

User manuals need to be prepared with a lot of care. Wright (1988) observes that documentation involves the integration of three basic decisions - decisions about the content of the manual, decisions about the presentation of the information in the manual, and decisions about how the effectiveness of the manual should be evaluated. Carroll et. al. (1986) has developed a manual format called the "Minimal Manual" which significantly addresses the first two decisions. Good documentation can minimize the effects of a flawed interface (Gong, 1990).

2.2.6 Requirement Management

Requirements management has been identified as one of the most critical aspects in controlling technology related risks. Organizational analysis, user surveys, information hiding, task analysis, user characterization, requirement scrubbing (Boehm, 1991) have been discussed in this regard. Experimentation, synthesis from characteristics of the utilizing system, paying early attention to poorly defined parts and system functionality, allowing the project to be driven by the user community and not by the developers (Schmid et al, 1996; McFarlan, 1981) etc are measures suggested to bring requirement changes under control. Poor execution of requirement elicitation will almost guarantee that the final project is a complete failure (Hickey and Davis, 2004). Requirement issues need to be resolved as the project progresses (Keil et. al., 1998). Unnecessary changes or requirements should not be entertained.

The risk of unrealistic estimation of budget and cost can be minimised through software reuse and requirement scrubbing (Boehm, 1991).The system should be kept as simple as possible (Alter and Ginzberg, 1978) and should be designed to cost (Boehm, 1991). The project success rate increases with standardized processes (Deephouse et. al., 2005) and design methodologies (McFarlan, 1981). Multi context analysis of the features required will also reduce such risk (Charette, 1996).

2.2.7 Estimation techniques

Empirical studies have demonstrated a positive relationship between effective estimation and satisfactory project outcomes (Deephouse et al., 2005; Guinan, Cooprider, and Faraj, 1998). Reliable estimates are critical for effective project planning and monitoring (Boehm, 1983; Abdel-Hamid, 1993; Lederer et al., 1990; Kemerer and Sosa, 1998). For a project to be termed as effective, the essential pre requisites are visibility of objectives, plans, status and other indicators.

Software Cost Estimation is an empirical process that is applied to estimate the effort and budget required for the software product which is going to be developed. The process starts with the planning phase activities and is refined throughout the development. This is very important for managing and scheduling the software project (Arifoglu, 1993).

Improving planning effectiveness will, in turn, help to meet budgets and schedules (Deephouse et. al., 2005). Given the uncertainties involved in software project estimations, there may be flaws in the initial estimation. Rather than driving project managers to pursue unrealistic goals, the project manager should be asked to set explicit goals and be judged on how effective his decisions are in achieving those goals (Abdel-Hamid, 1993).

2.2.8 Appropriate Methodology

Project managers need to make a series of decisions at the initial stage and during the course of a software project to ensure a high quality software product (Ferrin et al., 2002). Suitable design approaches like prototyping, evolutionary approach, modular approach, simulation, modelling etc reduce the risk of lack of familiarity of the designers (Zmud, 1980; Boehm, 1991; Alter and Ginzberg, 1978).

Simulation has been used to address a variety of issues in software development projects including strategic management, project planning and control and process improvement. Simulation support also reduces the development time. Using simulation models of the designs, we can increase the accuracy of early estimates of performance (McBeath and Keezer, 1993).

Practitioners view prototyping as an ideal approach for communication with users, as it provides more flexible designs, and is better for the early detection of problems (Verner and Cerpa, 1997).

Software re-use, the use of software artifacts across multiple projects, is an important strategy for improving software development efficiency and increasing the quality of software systems. When re-use is attempted, developers usually have access to the implementation code which can be modified to match a new project's requirements (Ravichandran, 2003; Rothenberger and Dooley, 1999). The major savings are realized in the detailed design, coding and testing phases. For these phases, software can be used or adapted rather than uniquely developed (Margono and Rhoads, 1992).

Benchmarking has been defined as "the search for and the implementation of best practices" (Jones, 1995). In the area of software development it is perceived as an assessment method, which is concerned with the collection of quantitative data on topics such as size, effort, defects, schedules and costs (Jones, 1995; Beitz and Wieczorek, 2000). It also helps managers to identify the quantum of improvement required to be the best (Beitz and Wieczorek, 2000).

2.2.9 Project Control

Project control includes comparing actual progress to planned progress and taking corrective action when performance deviates significantly from the plan (Boehm and Papaccio, 1988). It involves collecting information about costs, schedules, and technical output such as code, designs, documentation, test plans, training materials, and procedures (Weinberg, 1985). Information can be gathered via meetings, interviews, walk-through, and formal technical reviews.

Quality assurance (QA) makes sure that the product meets user requirements and that it provides the desired functionality and quality. While the whole project team should be committed to building quality into the product, it is a general practice to have a separate individual or a group whose primary responsibility is quality assurance (Jurison, 1999). Abdel-Hamid (1993) asserted that there should be a mechanism for comparing activities performed with a standard of what should be carried out, a procedure for changing behavior if there is a need and a feedback method or mechanism.

Review meetings play a major role in project control. Their purpose is to assess progress and identify areas of deviations from the plan so that corrective action can be taken. Project review meetings provide visibility to plans and create opportunities for obtaining and enforcing commitments from the participants (Jurison, 1999). Project review meetings are most effective when they are scheduled at regular time intervals and follow an established agenda. These meetings are used to resolve interpersonal conflicts as well as technical issues. Team members discover that, as they interact, they spend less time dealing with interpersonal issues at these meetings and more time on solving important technical problems (Pattit and Wilemon, 2005).

Any significant deviations or variances from the plan require prompt attention from the project manager so that timely corrective action can be taken. Keil et al. (1998) stated that to avoid the problem of scope creep, project managers should inform users of the impact of scope changes about details of project cost and schedule. Project managers should be able to distinguish between desirable and absolutely necessary functionality. The project manager must be able to identify the source of the problem. If there is a major deviation from the plan, the project manager must decide whether re-planning future activities is warranted (Jurison, 1999).

Technical performance control, the process of assuring that all technical requirements are met, is normally exercised through a variety of design reviews. These reviews are usually held at major milestones (e.g. completion of requirements definition phase, design phase, or coding) but it can be held at other times during the project also. The progress towards important technical goals should be tracked through appropriate metrics during the project. The metrics provide project managers visibility of what has been achieved, and their trends offer predictions of what can be expected in the future (Jurison, 1999). Software engineers use different types of software development technical review for the purpose of detecting defects in software products (Sauer et. al., 2000).

2.3 CHECKLISTS ON SOFTWARE PROJECT RISK AND RISK MANAGEMENT

One of the most common methods for identifying the presence of risk factors and risk management strategies in a particular project has been the use of checklists. These checklists present a list of all potential risks and risk management factors that might be applicable in a software development project.

McFarlan (1982) was one of the initial researchers in identifying the risk factors in software development. McFarlan Risk Framework was a general framework applicable to any IT projects. He highlighted the failure of project managers to consider the aggregate risk of the portfolio of projects. Also, they did not recognise the fact that those different projects require different managerial approaches. He categorised projects based on Project size (Size in cost, time, staffing level, or number of affected parties), Experience with technology (Familiarity of the project team and the IS organization with the target technologies) and Project Structure (how well structured is the project). He suggested appropriate risk mitigation strategies for these risk groups.

One of the pioneering studies in this regard is the top 10 risk list of Boehm (1991). His list has been compiled by probing several large software projects and their common risks and is thus empirically grounded. The top ten items are:

Personnel Shortfalls, Unrealistic schedules and budgets, Developing the wrong functions and properties, Developing the wrong user interface, Gold-plating , Continuing stream of requirements changes , Shortfalls in externally furnished components , Shortfalls in externally performed tasks , Real-time performance shortfalls and Straining computer-science capabilities. The list is also well-known and has been widely applied in practice to orchestrate risk management plans (Ropponen and Lyytinnen, 1997). But in spite of its popularity and simplicity, Boehm's list forms an inductively derived collection of risk items and thus lacks a theoretical foundation (Lyytinen et al, 1998; Barki et. al. 1993).

Barki et al (1993) tried to overcome some of these limitations to produce a more comprehensive list of risk factors. Based on a review of the IS risk literature, he prepared a list of 35 risk factors. From the data collected from the project leaders and user representatives from 120 ongoing projects in 75 organizations, the list was reduced to 32 which, after factor analysis, became 23. Factor analysis revealed 5 general categories or dimensions of risk. Those dimensions are: organizational environment, technological newness, expertise, application size, and application complexity.

Jiang and Klein (2002) supplemented this study through a survey among project managers asking them to rank these risk categories in order of importance. They also identified the necessary risk management focus areas (user focus, institutional focus, commitment focus and simple focus) to counter these risk factors.

One of the most quoted international studies on software project risk factors was conducted by Schmidt et al. in 1996. In an attempt to compensate for some of the previous shortcomings in checklists of risk factors, Schmidt et al. (1996) conducted a survey of project managers and developed an extensive list of risk factors in software development. Their research was accomplished through three simultaneous Delphi surveys in three different settings: Hong Kong, Finland and the United States. In each country, a panel of project managers was formed and a “ranking-type” Delphi survey was used to solicit risk items from the panel. These risk factors were then consolidated into a comprehensive list of software project risk items. The initial list produced by the expert panels was grouped into 53 unique risk items and then further reduced to a list of 11 risk factors common to all countries. These factors are Lack of top management commitment to the project, Failure to gain user commitment, Misunderstanding the requirements, Lack of adequate user involvement, Failure to manage end user expectations, Changing scope/objections, Lack of required knowledge/skills in the project personnel, Lack of frozen requirements, Introduction of new technology, Insufficient/inappropriate staffing and Conflict between user departments. The third and final phase of the study involved the actual ranking of these risk factors. It was seen that the ranking of the factors varied significantly across the countries. These factors were categorized into four segments and risk management strategies were suggested for all the factors.

Keil et al (1998) improved upon their international Delphi study exploring the issue of IT project risk from the user perspective and compared it with risk perceptions of project managers. By understanding the differences in how users and project managers perceive the risks, insights could be gained that may help to ensure the successful delivery of systems. The Delphi study revealed that these two stakeholder groups have different perceptions of risk factors. Through a comparison with Schmidt et al. (1996) study on project manager risk perceptions, zones of concordance and discordance were identified.

Anja Mursu (2000) repeated Schmidt et al's Delphi study design in Nigeria to identify the major software development risks. The study produced a rank-order list of software risk factors which are significant different with the rankings of the earlier study.

Addison and Vallabh (2002) also found the top ten risk factors to be different in their survey. They also found significant variation in risk perception when respondents were grouped based on the number of years of experience.

The Software Engineering Institute (SEI) has contributed considerably over time to the study of project risk management and has produced a significant amount of literature with a comprehensive inventory of variables related to the assessment of software development project risk. SEI's Taxonomy-Based Risk Identification Instrument, which contains 194 questions, is probably one of the largest checklist of software development risk factors. These 194 questions are classified under various categories falling under three classes namely Product Engineering, Development Environment and Program Constraints.

Moynihan (1997) through his survey on project managers who had managed custom-built, software-intensive application and development projects that originate from external clients produced a huge collection (113) of risk related constructs. He made a comparison of these risk factors with the Barki variables and SEI themes. Though there was considerable overlap, he found that many of the real world issues are not captured in the Barki list and the SEI list.

The work by Ropponen and Lyytinen (1997) contributes to the empirical studies about the commonality and type of software development risks. Using a survey instrument, they empirically delineated six components of software development risk namely scheduling and timing risks, system functionality risks, subcontracting risks, requirement management risks, resource usage and performance risks, and personnel management risks.

Alter (1979) identified eight risk factors: nonexistent or unwilling users; multiple users or implementers; turnover among all parties; inability to specify purpose or usage; inability to cushion impact on others; loss or lack of support; lack of experience; and technical or cost effectiveness problems after studying the implementation of 56 Decision Support Systems.

Zmud (1980) found that technological complexity, the degree of novelty or structure of the application, technological change, and project size influence the outcome of large software development efforts.

McBeath and Keezer (2003) suggested that several sources of uncertainty should be taken into account in the management of software development projects. These are: complexity, lack of structure, or instability of project objectives; newness of the technology; users; IS management; upper management; and project size.

Linda Wallace (1999) developed a valid and reliable measure of software project risk to study risk from a common perspective and to compare findings across studies in a more meaningful manner. Following established practices of instrument development, her study developed and empirically tested a model of software project risk. The final instrument had 44 items measuring 23 risk factors. Their factor analysis revealed 6 general categories or dimensions of risk. Those dimensions are: User, Development Team, Organizational Environment, Project Complexity, Project Management and Requirements. Scales were designed to measure each of the six first-order dimensions of risk.

Boehm's model (1991) suggests a comprehensive set of steps and guidelines to manage software development risks. For each of the top 10 risk items, Boehm developed a set of risk-management techniques that “have been most successful to date in avoiding and resolving the source of risk”. The idea is that after detecting the most important risk items risk-managers can compile the associated set of risk management measures and plans. These risk management practices include Award-fee contracts, Bench marking, Contracts, Design to cost, Early user's manuals, Incremental development, Information hiding, Instrumentation, Mission analysis, Morale building, Pre-award audits, Pre-scheduling, Prototyping, Requirements scrubbing, Software Re-use, Team building, Tuning, User Surveys, Compatibility analysis, Competitive design, Cost-benefit analysis, Cross-training, Detailed multisource cost and schedule estimation, High change threshold Incremental development, Inspections, Job-matching, Modeling, OPS-concept formulation, Organizational analysis, Reference checking, Scenarios, Simulation, Staffing with top talent, Task Analysis, Technical analysis and User characterization

Davis' model (1982) is concerned with selecting procedures that lead to complete and correct information requirements. His argument is that one of the reasons for the high risk in systems development is the inappropriate methodologies used for obtaining and documenting user requirements. Therefore he suggested “packaging” different methods into alternative strategies that are then carefully explained. These strategies are linked to a risk management model which suggests the most successful requirements determination strategy for a given situation. These strategies include Asking from users, Deriving from existing systems and Synthesis from characteristics of the utilizing system and Discovering from experimentation

Alter and Ginzberg’s (1978) focused on problems associated with the organizational acceptance and implementation of the information system. They argued that implementing any system would involve uncertainty from the managerial point of view. Therefore, these uncertainties should be detected and appropriate measures should be taken to minimize their impact. They recognized and classified several risk resolution strategies into inhibiting or compensating strategies. Inhibiting strategies are ex ante— this means to avoid a particular problem, while compensating strategies are ex post— this means to make up for a previous error or problem. These strategies are: Avoid change, Hide complexity, Insist on mandatory use, Keep the system simple, Obtain management support, Obtain user commitment, Obtain user participation, Permit voluntary use, Provide training programs, Rely on diffusion and experience, Sell the system, Tailor system to people's capabilities, Use evolutionary approach, Use modular approach and Use prototypes

McFarlan (1982) classified risk resolution techniques into four types, namely External integration, Internal integration, Formal planning and Formal control mechanisms. The strategies are Selection of user as project manager, Creation of user steering committee, Frequency and depth of meetings of this committee, User-managed change control process, Frequency and detail of distribution of project team minutes to key users, Selection of users as team members, Formal user specification approval process, Progress reports prepared for corporate steering committee, Users to be responsible for education and installation of the system, Make users manage decisions on key action dates, Select a manager to lead the team, Conduct frequent team meetings, Implement regular preparation and distribution of minutes on key design decisions, Conduct regular technical status reviews, Manage low turnover of team members, Include high percentage of team members with significant previous work relationships, Encourage participation of team members in goal setting and deadline establishment, Get outside technical assistance, Milestone phases selection, Systems specification standards, Feasibility study specifications, Project approval process, Project post audit procedures, Periodic formal status reports versus plan, Change control disciplines, Conduct regular milestone presentation meetings and Communicate deviations from plan

2.4 REVIEW OF STUDIES LINKING PROJECT RISK, RISK MANAGEMENT AND PROJECT OUTCOME

The studies referred above consider software risks along several dimensions and have provided some empirically founded insights of typical software risks and risk management strategies to mitigate them. Overall, these studies provide insights into risk management deliberations, but are weak in explaining the true impact of risk and risk management practices on the project outcome. A few studies have gone further to establish how risk management efforts reduce the exposure to software risk and can thereby increase software quality and improve software development (e.g., Fairley, 1994; Nidumolu, 1995; Wallace et. al., 2004).

A number of system performance criteria have been developed and empirically tested. They include IS usage, user satisfaction, quality of decision making, cost/benefit analysis, team effectiveness, and project effectiveness. The triple criteria of project success – meeting cost, schedule and performance targets - have been widely used by researchers to analyze project success. Saarinen (1990) proposed a system success measure with four dimensions: system development process, system use, system quality, and organizational impacts. The identification of these distinct dimensions of system performance illustrates that a project can be both successful and unsuccessful at the same time depending on the metric selected. One of the most popular approaches is to categorize these measures under process performance measures and product performance measures (Barki et. al 2001; Nidumolu 1995; Deephouse, 2005; Wallace, 2000; Al-Hindi, 1996; Ravichandran, 1996). Product outcome refers to measures of the "successfulness" of the system that was developed. It looks at how the software developed scores on important parameters of software quality: reliability, maintainability, easiness to use, response time, meeting the requirements, user satisfaction etc. Process outcome measures refer to the "successfulness" of the development process of the project. The focus is on completing the project within budget, within schedule and the on the overall quality of the development process. Both aspects are important as the software delivered by the project may be of high quality but the project itself may have exceeded the time and cost projections. On the other hand, well managed projects which come in below cost and time budgets may deliver poor products.

Due to the difficulty in quantifying costs and benefits, measures based on perceptions have become particularly prominent in IS literature.

Linda Wallace (1999) validated the second order factor model of risk through the establishment of co-alignment, a structural model of the relationship between risk and project outcome – both product and process outcome. The result of this research has established a tentative link between project risk and project outcome and shows that the level of risk associated with a project can have an impact on the ability of the project to be finished on time and within budget.

Jiang et. al. (2000) has independently done a study similar to the one described above and arrived at similar conclusions. He also found that software project risk can better be expressed as a second order factor model and that there is negative link between risk and project success.

Based on her previous research, Wallace et al (2004) developed a model linking risk and project performance. This was guided by project management literature and socio technical theory. Six components of risk were extracted through principal component analysis and these six dimensions were further grouped under three dimensions namely social subsystem risk, technical subsystem risk and project management risk. The relationship of these second order dimensions of risk with product and process outcome of the project was studied through structural equation modeling.

Drawing from contingency research in Organizational theory and IS literature, Barki et al (2001) developed an integrative contingency model of software project risk management. The central hypothesis in the model is that the outcome of the software project is influenced by the fit between the project risk and how the project is managed. The outcome measures used are cost over run and quality of the final software delivered. The risk management practices studied are formal planning, internal integration and user participation. High risk score and the low risk score projects are separately analyzed. In each of these groups, sub groups scoring high and low on performance factors are separated. Thus the following four categories emerged: low risk high quality, low risk low quality, low risk high cost performance, high risk low cost performance. The ideal profiles for each risk category are calculated. The fit is measured as the deviation from the ideal profile. The deviations are seen to be negatively correlated with the performance supporting the contingency model.

Nidumolu's (1995) study was a pioneering effort in linking software project risk to project performance. His study linked coordination mechanism and risk drivers to project performance. Two types of coordination mechanisms were studied. Vertical coordination is the interaction through formal systems and procedures, and horizontal coordination is how they coordinate through mutual adjustments and interaction. A new research model was developed along the structural contingency perspective in Organizational theory and risk based perspective in software engineering. This model introduced residual performance risk, i.e., the difficulty in estimating the performance related outcomes in the later stages of the project, as an intervening variable clarifying the relationship between risk, coordination mechanisms and performance.

Na et al (2006) study is an extension of the Nidumolu (1995) study in Korea. They utilized measure development and analysis similar to the process described in the Nidumolu study. Three distinct models were proposed in this study. Model 1 assessed the impact of two popular risk management strategies and residual risk on objective performance. Model 2 assessed the impact of functional and systems development risk on objective performance. Finally, Model 3 assessed the impact of functional and systems development risk on subjective performance. The study provides insights into managerial strategies to reduce the possibility of software project overruns. Their findings reveal that both functional and system development risks are important predictors of software project performance.

Jiang and Klein's study (2000) relates to various software development risks on project effectiveness. The 12 most common software development risks proposed by Barki et al. (1993) were examined. Project effectiveness, which is a specific dimension of the system performance, was studied through the following measures: meeting project budget and schedules, amount of rework, quality of work, and operation efficiency. The results of the study indicated that different project risks would impact different aspects of system development to differing extents. The following two risk factors were seen to have a more significant impact on effectiveness: lack of general expertise of the team and lack of clear role definitions for team members.

Claes Wohlin's (2002) research work links project success to project characteristics. Project characteristics are variables such as complexity, competence, requirements stability, personnel turnover, geographical distribution, methods and tools, time pressure, information flow, top management priority of project, and project management. The project success variable studied is the timely completion of the project. He ranks and classifies projects into three categories based on the success parameter. Then the projects are ranked and classified again based on the scores on the project characteristic variables. These two rank lists are compared and correlated. This is the basis for computing an index to predict project success/failure based on project variables.

Deephouse et al (2005), through an exploratory study, developed a conceptual model linking effectiveness of software processes to the project outcome. Seven software process variables namely planning, process training, stable environment, user contact, design reviews, prototyping and cross functional teams were considered. Product quality and time overrun were taken as indicators of project performance. The study also tested the planning effectiveness as a mediating variable. Project characteristics such as the staffing level, percentage of work outsourced, type of application developed are tested as control variables.

Ropponen and Lyytinen (2000) researched on how risk management practices and environmental contingencies help in addressing the risk components in software projects. The study assumed that software development risk could be split into several distinct dimensions and that various risk management methods and practices could influence different components of the software risk. In the same vein, they assumed that there existed a connection between environmental contingencies and the capability to handle software risk. The study provided encouraging evidence of how the use of risk management methods could address some of these risks. At the same time, the study recommended that software organizations must tailor their risk management efforts to their development environment.

Jiang et al (2000) studied the relationship between the major risk factors and the responses to these risk factors (mitigation strategies). Response strategies were classified under four implementation focus areas namely user focus, institutional focus, commitment focus and simple focus. To examine the specific implementation focus adopted by project managers in different risk situations, a series of regression analyses were conducted with risk factors as dependent variables and each implantation focus area as an independent variable. He found that different risk situations warranted different approaches.

A similar analysis was performed by Addision and Vallabh (2002) to determine whether there were significant relationships between some of the demographic data (such as number of years of experience of the project manager) and the risks and risk controls. These tests were performed using regression analysis, to compare the controls to each of the risk factors to determine if they were effective in mitigating the occurrence of each risk factor. Some of them were seen as significant though none of the pairings resulted in high values of R^2.

Keil (1995) looked at another aspect of software project failure called project escalation. It refers to the continued support given to projects which are in deep trouble and are all set to fail. Why further resources were wasted on these projects was the research problem. His model linked a set of project factors, psychological factors, individual factors and organizational factors to project escalation behaviour.

Kirsch (1996) proposed to build an integrated contingency model of software project management linking project management practices to the characteristics of the project and attributes of the individuals involved. He hypothesized that the project characteristics directly affected project management practices while individual attributes may have both direct and moderating effects on such practices. These project management practices are, in turn, believed to affect project performance.

Walsh and Schneider (2002) had looked at software development risk from the agency theory point of view. He studied the causal relationship between team decisions and project success. According to him, team decisions are influenced by the agency effects (alignment between the interest of the member and that of the organization) as well as development ability of the member.

2.5 OBSERVATIONS FROM LITERATURE REVIEW AND MOTIVATION FOR THE CURRENT RESEARCH WORK

The researchers have adopted different approaches to developing checklists on risk and risk management. One approach has been to develop these checklists based primarily on their personal experience with software development projects. Boehm‘s top 10 is largely based on his experiences at TRW. The same was true with many studies in the 1980s and early 90s (e.g., Casher, 1984; McFarlan, 1982). Many of these checklists were criticized as they were not very systematic and coherent and lacked any theoretical basis (Ropponen and Lyytinen 1997; Schmidt et al. 1996).

Another approach adopted was to develop the checklist based on extensive literature review. The major work in this regard includes the literature from Software Engineering Institute. But many subsequent researchers have questioned the lack of empirical validation of these instruments developed. No steps were taken to contact practicing software project managers for input into the relative importance and accuracy of the identified risk factors. (Wallace, 1999)

The third approach was to elicit the list of risk factors from practicing managers. Many researchers have conducted surveys among the members of software projects (Moynihan, 1997, Jiang et. al. 2002). Many of these studies have been criticized for their limited focus on IS literature. They made no attempt to reconcile their findings with the IS literature in this area. This has limited their usefulness as a comprehensive practical tool for gauging project risk.

Linda Wallace (1999) study stands out as an attempt to develop a comprehensive measure of software development risk based on literature which is later validated with software professionals working in software projects. But she has focused only on in-house software development with USA companies most of which were non IT companies. Hence many of the non USA risk factors as well as risk factors specific to software development companies may not have been captured in her list.

Also, most of the previous research takes an isolated view of software project risk and risk management strategies. Very few studies have taken an integrated and comprehensive view of risk and risk mitigation strategies linked to project outcome. Arguments are largely based on anecdotal evidence or armchair theorizing. Empirical evidence on the relationship between risk and project outcome is rare and often fails to take into account various risk factors that may hinder success.

Linkages among software development risks, risk management strategies and various dimensions of system success are generally overlooked in IS literature. Yet, this is an important step for advancing our knowledge on project risks because it is very likely that different project risks may affect the various dimensions of system success differently. A particular control procedure or method may reduce only certain aspects of software development risk and not others. Linkages between risks and various dimensions of system success can help project managers to select the needed implementation strategies to achieve their desired project outcomes.

Most of the studies in these domains have been done in developed countries but have come out with generalized conclusions regarding the risk factors in software development projects. This has been acknowledged as a major limitation of the research on software development risk factors. Many researchers have argued with empirical evidence against this generalization across countries.

The most quoted international study on software risk factors by Schmidt et al. in 1996 in Hong Kong, Finland and the United States showed that there is no consensus on the top risk factors across countries. Mursu (2000) replicated the Delphi study of Schmidt in Nigeria. He found significant differences in the risk factors and their importance in Nigeria compared to what the original study showed. Similarly Na et al. (2006) found that models developed with data collected in USA do not apply to organizations in Korea where the IT capability is known to be lower than in the USA. Specifically, their study suggests that, unlike the Nidumolu (1995) study conducted in the USA, residual risk is not a significant predictor of subjective performance measures such as software project process and product performance

Many researchers acknowledge that cultural differences can impact work related values and play a significant role in the success or failure of projects (Hofstede, 1991). Joan Mann and James P. Johnson proposed a research model for risk associated with information systems projects in Thailand. His model is based on the premise that the Thai culture is likely to impact the propensity for risk to occur. A risk factor significant in one culture may not be significant in another.

Most of the previous studies focus on software professionals working in non IT companies. This has influenced many of the risk factors identified in the studies (e.g. factors such as lack of support from top management, lack of expertise available etc). Not many studies have been done focusing only on software development in software companies.

Also, most of the studies focus either on in-house projects or completely outsourced projects where the end user is well defined. The outsourced projects covered in these studies are projects outsourced to IT companies in the same country. There is a substantial increase in software development outsourcing to firms overseas, especially in India. Multi-level outsourcing is also very popular. This will lead to project management issues such as lack of visibility about the final user, contractual problems, information asymmetry etc.

All these point to the need for more studies to be able to generalize across varying socio-economic contexts and also to develop insights into the risk- risk management- project outcome models in different contexts.

India presents many unique characteristics which will have impact on the risk and risk management. The working environment in software companies in India is different from that of USA. Some of the notable differences include

- Flexible working hours to absorb the risk impact without causing cost and time overruns.
- Low cost per employee in India resulting in high profit margins. This helps Indian companies to absorb the financial impact of the occurrence of a risk event.
- Infrastructure support in Indian companies is inferior to their USA / UK counterparts.
- The regulatory issues cause considerable risk— visa rules, work permits, minimum wage salaries, laws on working environments etc can pose specific problems
- The employee turnover is very high in India. Absence of proper in-house systems and documentation practices put companies at an immense disadvantage when employees leave without any notice.
- Legal and political risk is involved given the fact that the major chunk of the software development in India is for foreign clients.

Hence the validity of the findings of the international researchers needs to be tested in different environments such as India in order to assess their universal applicability.

2.6 SUMMARY AND CONCLUSION

The chapter has reviewed previous research on software development project risk, risk management, measurement scales and models linking risk and risk management to project outcome. The research by Wallace (1999) is specially noted for its rigour and comprehensiveness.

The major limitations of the study are also noted. The literature still lacks a comprehensive and validated study linking risk, risk management and project outcome. Also no major studies on these constructs are reported from India. The motivation for the present research is derived from these limitations.

****

CHAPTER 3

METHODOLOGY AND INSTRUMENT DEVELOPMENT

This chapter details the various aspects of the research methodology used and the work done to develop the final instrument used for data collection. Section 3.1 presents the methodology aspects. The rationale for the study, statement of the problem, objectives of the research, background of the study, hypothesis to be tested, variables in the study, scope of the study, research design, analysis design, sampling design and tools for data collection are discussed in this section. Section 3.2 looks at the instrument development process. The item generation process, validity analysis, pilot test, exploratory factor analysis and structure of the final instrument are presented under this head. Section 3.3 gives the conclusion of the chapter.

3.1 RESEARCH METHODOLOGY

3.1.1 Rationale for the study

Organizations across the globe are concerned about the high rate of failure of software development projects. International research agencies such as Standish Group report that one third of the software development projects get cancelled before they are completed. Experts in the area recommend that risk associated with software development projects must be identified and managed so as to improve the project success.

Risk and risk management are two important constructs for both researchers and practitioners in the area of software development. Risk assessment involves identifying, analyzing and prioritizing the risk items that are likely to adversely affect a project and risk management involves managing these risk items so as to eliminate or control them.

Most of the past research has taken independent views of software project risk and risk management. The combined impact of risk and risk management on the project outcome is rarely studied. No major studies on this topic are reported from India. The need is felt for more studies in order to generalize the findings of risk research across varying socio-economic contexts, and also to develop insights into the risk – risk management – project outcome models.

3.1.2 Statement of the problem

Risk and risk management are multidimensional constructs whose sub dimensions need to be studied and analysed. The literature review in chapter 2 has identified major gaps in research with respect to these constructs. This research tries to plug some of these gaps in research both in international as well as in Indian context. The following major problems are addressed in this study.

1. What are the major risk factors present in the software development projects in India?
2. What are the risk management practices which are most effective for these risk factors?
3. What is the independent and combined impact of risk and risk management on project outcome?

3.1.3 Objectives of the research

Primary Objective of the study is stated below:

To study software development project risk, risk management, project outcomes and their inter-relationship in the Indian context.

In order to achieve this primary objective, the following specific sub-objectives were stated

1 To study the extent and nature of risk in software development projects.

2 To study the risk management practices adopted in software development projects

3 To study how risk management practices influence risk in the project.

4 Develop a model linking risk and risk management to project outcome.

3.1.4 Theoretical background of the study

Chapter 2 has presented a detailed review of the previous research on software project risk and risk management.

Many researchers on software development attribute reasons for variations in project success to risk factors which are technical, economic and behavioral in nature (Barki et. al. 1993; Lyyttinen 1988). These risk factors pose danger to the development of a successful project leading to inadequate software operations, software re-work, implementation difficulty, delay or uncertainty (Boehm, 1991).

There are many arguments on how considerable improvements can be made in the software development process through the systematic applications of risk management techniques and guidelines (Alter et al. 1978; McFarlan 1982; Boehm 1989, 1991).

As mentioned in chapter 2, many researchers have explored and established a tentative negative link between project risk and project outcome (Wallace, 1999; Jiang et. al., 2002). Deephouse et al (2005) and Wohlin (2002) looked at the link between project success and risk management. Nidumolu (1995) linked a select set of measures of risk and risk management to project performance. Barki et al (2001) developed a model looking at how the outcome of the software project is influenced by the fit between the project risk and risk management. But these studies used a very limited set of risk factors and risk management practices. Linkages among the software development risk, risk management and the various dimensions of system success are generally overlooked in the literature.

The following conceptual model, which has an integrated and comprehensive view of risk, risk management and project outcomes, was tested in this study.

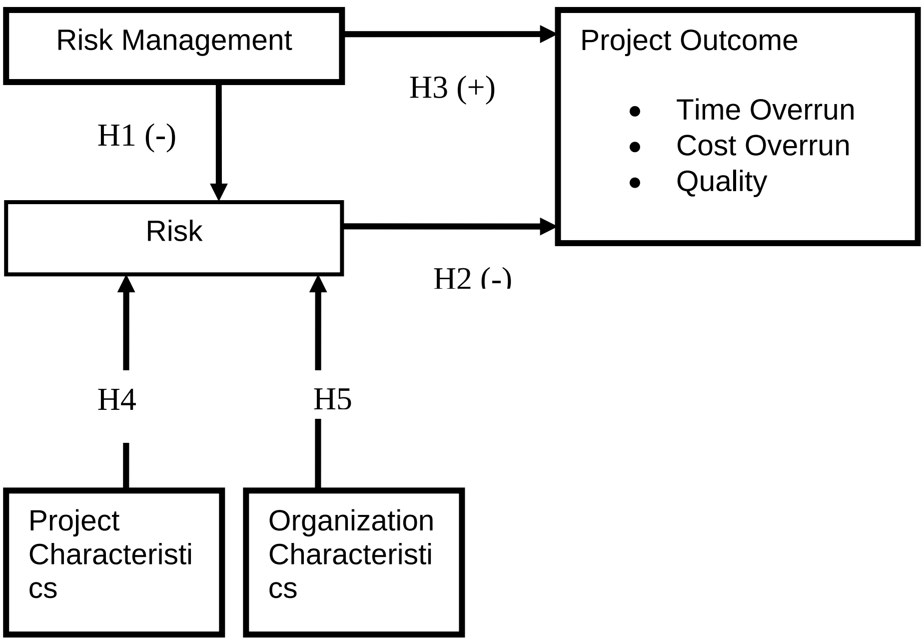


Fig 3.1: Conceptual model linking risk, risk management and project outcomes

3.1.5 Hypotheses

Based on literature, the researcher has formulated the following five (alternate) hypotheses on the anticipated relationship among the variables in the study.

H1: Risk management has significant negative relationship with risk

H2: Risk has significant negative relationship with project outcome

H3: Risk management has significant positive relationship with project outcome

H4: There is a significant variation in risk across categories of project characteristics

H5: There is a significant variation in risk across categories of organizational characteristics

3.1.6 Variables in the study

The theoretical and operational definitions of the variables in the study are given below. The scales used for measurement are either taken from published inventories or developed by the researcher. The scales are tested for their validity and reliability. The scale development process and the associated statistics are described in the subsequent sections.

1 <u>Risk</u>

Software project risk points to an aspect of a development task, process or environment, which if ignored tends to adversely affect the outcome of a software project (Lyyttinen et al. 1993). The risk factors are technical, economic and behavioral in nature (Barki et. al., 1993; Lyyttinen, 1988). They pose danger to the successful completion of a project leading to inadequate software operations and overruns (Boehm, 1991).

One of the most common methods for risk identification has been the use of risk factor checklists (Barki et. al., 1993; Wallace, 1999). These checklists present a list of all potential risks and check to what extent they are present in a project.

Operationally, risk is defined as the presence of the factors that will adversely affect the software development project. In this study, measurement of risk is made after the completion of the project. This researcher has developed a valid and reliable measure of software project risk modifying the instrument of Wallace (1999). The tool has 55 items measuring risk under the following sub dimensions: Team risk, user risk, planning and execution risk, external risk and project complexity risk. The respondent has to indicate the level of presence of each risk item in the project on a five point Likert type scale.

2 Risk Management

Risk management is concerned with a phased and systematic approach to analyse and control risks occurring in a specific context (Charette, 1996). Boehm (1991) defines risk management as an emerging discipline whose objectives are to identify, address, and eliminate software risk items before they become threats to successful completion of the project. Considerable improvements can be made in the software development process through the systematic applications of risk management techniques and guidelines (Alter et al. 1978; Davis 1982; McFarlan 1982; Boehm 1989, 1991). Research on software risk management has primarily focused on crafting guidelines for specific tasks (Alter et al. 1978; McFarlan 1982; Boehm 1989; Charette 1996).

Operationally, risk management has been defined as the presence of practices which are crafted to reduce the impact of risk in software projects. As in the case with risk, the measurement of risk management is also made after the completion of the project. Researcher has developed a valid and reliable measure of software project risk management drawing from the works of various researchers in this domain. The scale has 36 items measuring risk management under the sub dimensions HR management, execution management, project planning and user coordination. The respondent has to indicate the level of presence of each risk management item in the project on the five point Likert scale.

3 Project Outcome

Project Management Institute (PMI) defines project as “One shot, time limited, goal directed major undertaking requiring the commitment of various skills and resources”. Project involves unique or risky undertakings which have to be completed by a certain date, for a certain amount of money and within some expected level of performance. A project is usually deemed as successful if it meets the designed requirements, is completed on time and is delivered within budget (Powell and Klein, 1996). The triple criteria of project success – meeting cost, schedule and performance targets – have been widely used by researchers to analyze project success (Willaims, 1995; Barki et. al., 2001; Nidumolu 1995; Deephouse, 2005; Wallace et. al., 2004; Al-Hindi, 1996; Ravichandran, 1996).

Triple criteria approach of measuring the project outcome in terms of time overrun, cost overrun and quality was adopted in this study.

a) <u>Time overrun</u>

Time overrun is an indicator of poor project performance. It indicates how much the project has exceeded the planned time. It is reported as a percentage. A negative value for time overrun indicates a good performance and is known as time underrun.

Time overrun is calculated as

$$Time\ overrun = (Actual\ Time\ - Planned\ Time) * 100 \div Planned\ Time$$

In this study, the respondent was asked to report time overrun as a percentage. Data on planned time and actual time was also collected so that the researcher could compute and confirm the percentage value. This was done to improve the accuracy and consistency of the measurement.

b) <u>Cost overrun</u>

Cost overrun is an indicator of poor project performance. It indicates how much the project has exceeded the budgeted cost. It is reported as a percentage. A negative value for cost overrun indicates a good performance and is termed cost underrun.

Cost overrun is calculated as

$$Cost\ overrun = (Actual\ Cost\ - Planned\ Cost) * 100 \div Planned\ Cost$$

In this study, the respondent was asked to report the cost overrun as a percentage. Data on budgeted cost and actual cost was also collected so that the researcher could compute and confirm the percentage value. This was done to improve the accuracy and consistency of the measurement.

c) Quality

Quality is defined as conformance to the explicit and implicit characteristics that are expected of professionally developed software (Pressman, 2001).

Operationally it was defined as how the software developed is rated on the following characteristics of professionally developed software: reliability, portability, ease of use, maintainability and flexibility. The respondents were asked to rate the software developed on a five point Likert scale. The average score on these parameters was taken as a measure of the overall quality of the software developed.

4 Project characteristics

a) Type of software developed

The type of software was measured by checking whether the software developed belonged to any of the following categories- business application, engineering application, web application , system software, others.

b) Project Duration

Project duration was defined in terms of the actual number of calendar months taken to complete the project once it was initiated. Project duration was classified as one of the following: short (less than 6 months), medium (up to 2 years) and long (2 years and more).

c) Project team size

Project team size was defined as the number of members in the project team. It was classified as one of the following: (a) small (5 members or lesser) (b) medium (5 to 20 members) (c) large (more than 20 members)

d) Onsite–offshore split up

This is defined by checking whether the project was executed predominantly in the offshore development center in India or onsite. Three categories were defined. (a) Offshore (b) Mixed (c) Onsite

5 Organizational characteristics

a) Size of the organization

The size of the organization was measured in terms of the number of employees.It was classified into one of the following categories: Upto 100 employees, 100 to 1000 employees, 1000 to 10000 employees, above 10000 employees.

b) Nature of the organization

The nature of the organization was measured by determining which of the following categories it belonged to: Indian arm of a foreign MNC, Indian company with international operations, Indian company with domestic operations, others

c) Quality certifications

Information was collected on whether the organization had any of the following quality certifications: ISO, CMM and PCMM

6 Demographic variables of the respondent

Data was collected on the role of the respondent in the project.

3.1.7 Scope of the study

Scope of the study defines the boundaries of the research. The four elements characterizing the scope of the study are defined as below:

Population: Software development projects are taken as the basic unit of analysis namely sample unit. Risk, risk management and project outcome data of a software development project was collected from software professionals who were in the project team. Hence the population is defined as the completed software development projects undertaken by software development organizations based in India. The units of observation were software professionals in various roles who had been part of the project team from the beginning to the completion of the project. The population is finite but its exact number is not available.

Place of study: The study was conducted in Chennai, Bangalore, Cochin and Trivandrum

Period of the study: The research interest was to analyze the present scenario regarding the objectives. The period of data collection was from March 2006 to July 2006

Data Sources: Major source of data was primary data collected from the software professionals. Organization details of the respondents were collected through the secondary sources such as NASSCOM database and published documents in company websites.

3.1.8 Research Design

The study describes the characteristics of the risk and risk management in software projects. It tries to explain how these constructs are linked to the project outcome. So the research design is explanatory in nature.

3.1.9 Sampling Design

As mentioned earlier, there is no source giving the details of all software development projects with Indian software companies. National Association of Software Companies (NASSCOM) is the most respected and recognized body of Indian software industry. NASSCOM database has a region-wise listing of software companies in India. The NASSCOM database has details of approximately 1000 software companies operating in India. These companies account for over 90% of the revenue of the software industry. 92% of these companies are concentrated in the following Tier 1 cities: Bangalore, Chennai, Hyderabad, Bombay–Pune, Delhi (NCR) and Calcutta. Tier II cities such as Mysore, Cochin, and Coimbatore are also fast emerging as destinations for software companies. The researcher had conducted interviews with senior professionals who have worked in multiple locations to check whether the location plays a significant role in deciding the risk/risk management levels in a project. Their opinion was that there could be differences across different tiers of cities but within the same tier, location is immaterial. Analysis of pilot study data which had representation from locations such as Noida, Mumbai, Coimbatore, Mysore etc also supported the view that within the same tier, the location is irrelevant. Therefore it was decided to assume homogeneity across cities belonging to the same tier and to survey only representative cities. Chennai and Bangalore were selected as tier I representatives. Cochin and Trivandrum were selected as representatives of the tier II cities. Given the number of variables involved and the statistical analysis planned, the researcher had set a target of 600 as sample size.

NASSCOM list of software companies in the selected cities was accepted as the sample frame for software development companies. Letters were sent to the center heads/HR managers of all the companies requesting them to participate in the study. Reminder letters were sent after three weeks. 105 companies agreed to participate in the study. The regional distribution of the companies is shown below in table 3.1.

Table 3.1: The regional distribution of the companies agreeing to participate

Location	Number of companies	Number of companies
Bangalore	250	59
Chennai	105	34
Cochin	8	4
Trivandrum	15	8
Total	378	105

The researcher personally spoke/met center heads/HR managers and briefed them on the study. They were requested to distribute the questionnaire to one member from each of the recently completed projects. They were requested to collect data from maximum number of projects covering different types of projects and members in different roles but keeping the condition that only one response should be solicited from one project. The objective was to cover different types of companies, projects and respondents without collecting multiple responses from the same project.

3.1.10 Analysis design

The statistical package SPSS11.0 was used for data editing, coding and basic analysis. ANOVA, t-tests and correlation analysis were used for hypothesis testing. Exploratory Factor analysis was performed on the pilot data to understand the structure underlying the risk and risk constructs. The factor structures were verified with a confirmatory factor analysis on the final data. The basic models linking risk, risk management and project outcome were tested with regression analysis. The integrated and the comprehensive models were tested with structural equation modeling (SEM) using AMOS 4.0.

3.1.11 Tools of the data collection

Questionnaire method was used for data collection. The form was handed over to the respondent with a covering note. Necessary instructions were also mentioned. Figure 3.1 shows the instrument development process. The details are described in the next section and also in Chapter 4

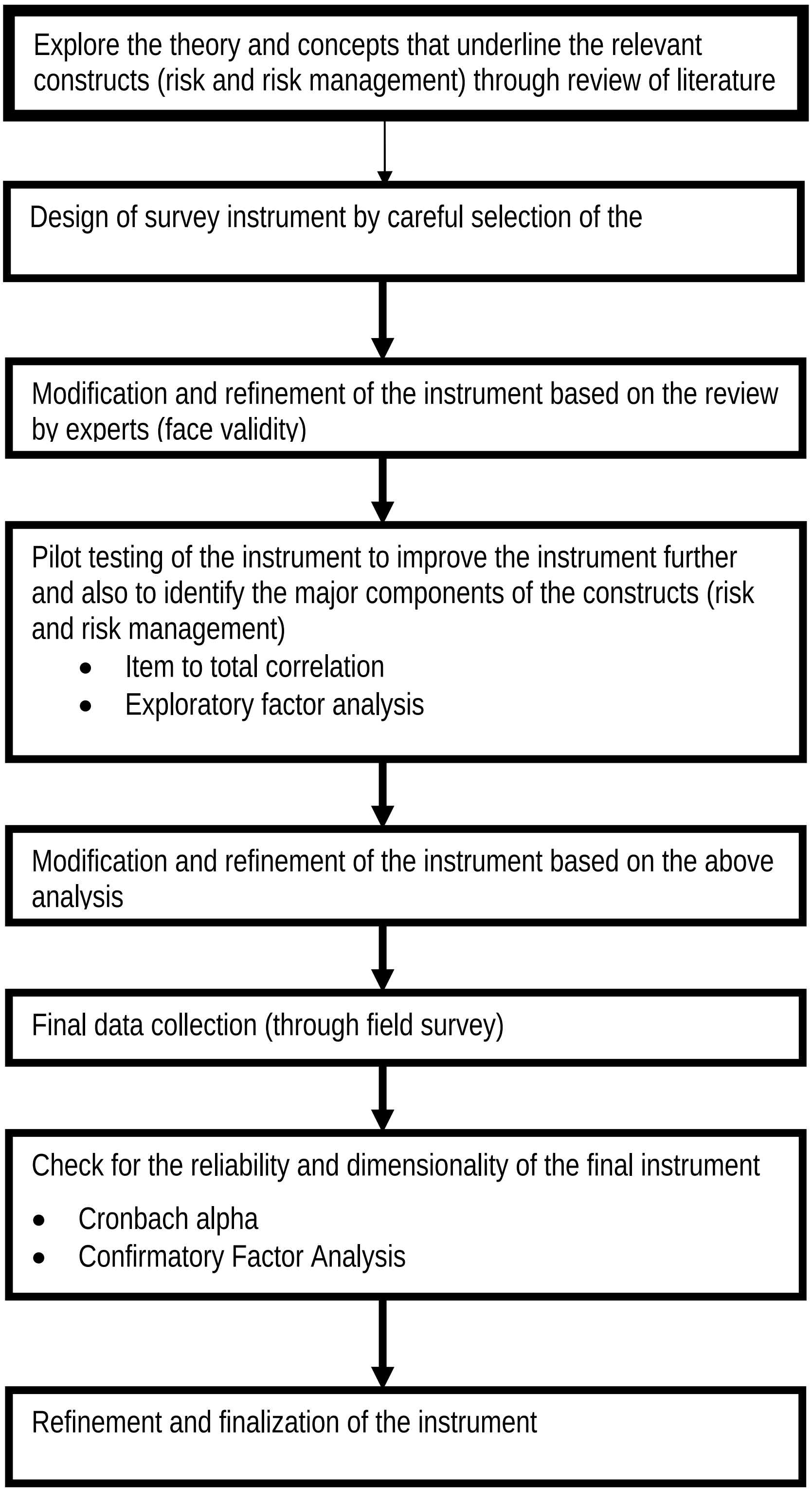


Fig. 3.2 Development and validation of the instrument

.

3.2 INSTRUMENT DEVELOPMENT

3.2.1 Item Generation

Focus in this phase was to develop a comprehensive instrument to measure the constructs of risk, risk management and project outcome. This was done based on the exhaustive survey of literature described in Chapter 2. As mentioned in the literature review, Wallace (1999) had developed a valid and reliable measure of software project risk to study risk from a common perspective. Following established practices of instrument development, her study developed and empirically tested a model of software project risk. Her final instrument had 44 items.

In the present study, Linda Wallace's instrument was taken as the base for measuring the software project risk. It was appended with items from other checklists such as Barki (1993), Boehm (1990) and SEI (1993) mentioned in section 2.4 of Chapter 2. India specific risk items were also added. The risk management instrument was developed by assembling various strategies mentioned in literature as identified in the literature review. Triple criteria measure of project success was used to measure project outcome.

This resulted in a draft questionnaire with 118 items for risk and 91 items for risk management. The project outcome was measured with 9 questions on the product quality and three questions each on time and cost overrun. The respondent had to indicate the presence of each risk/risk management item in his project on a five point Likert scale (strongly disagree; disagree; neither agree nor disagree; agree and strongly agree). Product quality was measured through a five point rating scale where the respondent rated the software developed on nine dimensions of software quality. The overruns (time and cost) were measured by asking the respondent to indicate the required data after consulting the project documents/appropriate authority. There were also questions covering the personal, project and organizational details.

3.2.2 Validity analysis

Validity is defined as the extent to which any measuring instrument measures what it is intended to measure (Carmines and Zeller, 1990). Different validity terms are used to illustrate the various aspects of validity. Any research instrument should be tested for validity, so that it could be used for meaningful analysis. The initial validity tests, namely content validity and face validity were performed for the draft questionnaire as explained below.

<u>Content validity</u>

Content validity of an instrument refers to the degree to which it provides an adequate depiction of the conceptual domain that it is designed to cover (Hair et al., 1998). In the case of content validity, the evidence is subjective and logical, rather than statistical. Content validity can be ensured if the items representing the various constructs of an instrument are substantiated by a comprehensive review of the relevant literature (Bohrnstedt, 1983).

The instrument had been developed on the basis of a detailed review and analysis of the prescriptive, conceptual, practitioner and empirical literature, so as to ensure the content validity.

<u>Face validity</u>

Generally, a measure is considered to have ‘face validity’ if the items are reasonably related to the perceived purpose of the measure (Kaplan and Scauzzo, 1993). Face validity is the subjective assessment of the correspondence between the individual items and the concept through rating by expert judges (Hair et al., 1998). In face validity, one looks at the measure and judges whether it seems a good translation of the construct under study. Face validity is also a subjective and logical measure, similar to content validity. The face validity can also be established through review of the instrument by experts in the field (Hair et al., 1998).

The draft questionnaire was given to five senior software professionals from the industry and five senior professors in software engineering. They were briefed about the purpose of the study and its scope. The experts were requested to scrutinize the questionnaire and to give their impressions regarding the relevance of contents of the questionnaire. They were requested to critically examine the questionnaire, and to give objective feedback and suggestions with regard to the comprehensiveness/coverage, redundancy level, consistency and number of items in each variable. They had to suggest necessary changes by simplifying, rewording, removing, replacing and supplementing the items. Based on the feedback from experts, the researcher modified the draft questionnaire. This resulted in a new questionnaire, referred to as ‘pilot questionnaire’, containing 73 items under risk construct and 43 items under risk management construct. The project outcome questions were retained without any change.

3.2.3 Pilot Test

The pilot questionnaire was administrated to a convenient sample of 200 software professionals with at least one year of software development experience. The goal of this exercise was to obtain a general assessment of the instruments’ appearance, to further eliminate items that did not contribute significantly to the value of the instrument, and to understand the underlying dimensions of the constructs under study.

The data collected from the pilot group was first scrutinized to identify the no response questions. If more than 80% of the respondents did not respond to a question, it was identified as a candidate to be removed or reworded.

<u>The item to total correlation</u>

The item–total correlation of every item in the scale was examined. An item–total correlation is the correlation between an item and the sum of the remaining items that constitute its scale. Items that have low item–total correlations may be candidates for elimination or rewording. But as Moore and Benbasat (1991) caution, before an item is deleted a check was made to ensure that the domain coverage (content validity) of the instrument did not suffer from the item’s omission.

Risk or risk management statements with low item–total correlations were identified. They were deleted if they did not affect the content validity in which case they were reworded

3.2.4 Exploratory Factor Analysis

Another way of gaining insight into the structure of the instrument is to perform an exploratory factor analysis. It is a procedure that can be used when data has been obtained on a large number of variables and the researcher wants to explore the nature of the underlying factor structure. Factor analysis refers to a class of procedures used for variable reduction and summarization. A large number of variables which are highly correlated can be reduced to a manageable level though this technique. The interrelationship among these variables are examined and used to define the underlying factors.

There are many methods of factor analysis such as Principal Component Analysis, Centroid Method and Maximum Likelihood Method. The most popular technique used is Principal Component Analysis (PCA). PCA is “simply a variable reduction procedure that (typically) results in a relatively small number of components that account for most of the variance in a set of observed variables” (Hatcher, 1994). This is recommended when there is no decision made regarding the number of underlying factors. It seeks to maximize the squared loadings of each factor extracted in turn.

Number of factors is another crucial decision to be taken in factor analysis. It is possible to have as many factors as there are variables, but in doing so no parsimony is gained. The objective is to decide on a set of interpretable number of factors which explains a substantial part of the variation in the data. Several procedures have been suggested for determining the number of factors under PCA. They include eigen values, scree plot, percentage of variance explained etc. An important output from factor analysis is factor loading matrix, which represents the correlation between factors and variables. A large absolute value indicates closer relationship of the variable with the factor. To get a simpler structure which can be better explained, the factor matrix is subjected to a rotation. Different rotation methods such as Varimax, and Quartimax are used to get a simple structure.

Factor Analysis on Risk Factors

The data from the pilot study was subjected to a PCA using a varimax rotation. The number of factors was decided looking into (a) literature support (b) percentage of variance explained (c) eigen values (d) interpretability of the factor structure.

There are many studies conducted on the factor structure underlying the risk construct. The Linda Wallace study (1999) had identified six factors which were named as team risk, requirements risk, organizational environment risk, user risk, process management risk, and project complexity risk. Barki et el. (1993) had demonstrated a five-factor structure, namely organizational environment, technological newness, expertise, application size, and application complexity. Ropponen and Lyytinen (1997) delineated six components of software development risk, namely scheduling and timing risks, system functionality risks, subcontracting risks, requirement management risks, resource usage and performance risks, and personnel management risks.

The percentage of variance explained approach ensures that a minimum percentage of the total variance is explained by the factors. No absolute cut offs are prescribed for all applications. If the factor structure explains 50% or even less in some cases, it is considered as satisfactory in social sciences (Hair et.al, 1998).An eigen value represents the amount of variance associated with a factor. In this approach only factors with eigen value more than 1.0 are to be retained.

The factor analysis on the pilot data resulted in a solution with six factors with eigen values above 1.0. The proportion of the variance explained by the six factors together was 62.8%. But these six factors obtained were hard to interpret. Such situations, where "... the number of factors retained tends to be larger than the number of factors that the researcher is willing to accept... (have) forced researchers to apply another criterion – that of substantive significance, which is applied after finding statistical significance" (Kim and Mueller, 1988). Two factors were found to have eigen value less than 1.1 with each explaining less than 2% of the total variance. Consequently, factor solutions with five and four factors were also analysed. An examination of the variables loading into the different solutions led to the selection of the five factor structure as the best interpretable solution. This explained 60.8% of the variance in the data. Though the sample size was not very high given the number of variables involved, Kaiser–Meyer–Oaklin measure of sampling adequacy (KMO value) obtained was above the adequate value of 0.7 (Hair et al., 1998). Bartlett Test of Sphericity was significant indicating that correlations exist among some of the response categories. The factor loading matrix was closely analysed after subjecting it to varimax rotation. While interpreting the rotated factor pattern, an item is said to load on a factor if the factor loading is 0.40 or greater (Moore, 1979). Using this criterion, the rotated pattern matrix was examined thoroughly. Items that did not load on a factor or items that crossload on multiple factors were identified and were deleted or reworded.

The final list modified based on the inputs from these three analysis (no response, item total correlation and factor analysis) had 60 items under risk construct. The factor loading matrix in shown in Table 3.3. Factor 1 contained items which were risk items related to the organization in general or the project team in particular. It covered items such as nature of project leadership, management support, team communication, quality of training and experience of the team members. Hence this factor was named as **Team Risk**. Factor 2 contained risk items arising in feasibility studies, resource estimation, planning, methodology and performance measurement. They were predominantly connected to project planning and execution phases. Hence it was named as **Project Planning and Execution Risk**. Factor 3 contained items related to the external environment namely regulation, subcontractors etc. It was named as **External Risk**. Factors 4 items were mainly user/client related issues such as meeting user expectation, conflict with the user and quality of documentation for the user. Hence it was called **User Risk**. The final factor contained items relating to the complexity of the work involved. It covered items such as complexity of requirements, changing requirements and number of links required to other systems. It was named **Project Complexity Risk.**

The percentage of variance explained by the factors are shown in table 3.2

Table 3.2 the percentage of variance explained by the factors for Risk Construct

Factor No	Factor Name	No of items under the factor	% of variance explained
1	Team Risk	26	24.9 %
2	Project Planning and Execution Risk	19	13.6 %
3	External Risk	5	9.5 %
4	User Risk	6	7.7 %
5	Project Complexity Risk	4	5.1 %
	TOTAL	60	60.8%

Table 3.3 Factor loading matrix for Risk Constructs

Risk Items	Factors (Dimensions)				
	I	II	III	IV	V
Staff motivation was very low	0.734				
Project goals and objectives were not agreed upon	0.729				

Project team communication was ineffective	0.716				
The corporate environment in the organization was not professional	0.715				
Project team members were inadequately trained	0.705				
Insufficient resources were provided for the project	0.696				
The project leadership did not have "people management skills"	0.687				
Politics in the organization had a negative impact on project	0.682				
The project leader was inexperienced.	0.677				
Team members were mostly inexperienced	0.675				
Reward structure for team member performance was poor	0.674				
The project manager was ineffective	0.662				
The offshore team did not fully understand the priorities of the on site team	0.648				
Team members were not familiar with the type of application being developed	0.645				
Frequent conflicts occurred among members of the project team	0.644				
Frequent shuffling of Project team members affected productivity	0.629				
Team members lacked communications skills in English	0.62				
Hardware infrastructure available was poor	0.617				

Top management support was lacking in the project	0.588				
The team faced cross cultural issues working for a foreign client	0.584				
Facilities such as video conferencing were unavailable	0.514				
The work pressure was so high that most of the employees had to work beyond the office hours	0.508				
There were restrictions on working hours for women members	0.486				
The project team was a highly diversified group	0.463				
Team member turnover (members resigning) was very high	0.461				
The project progress was not monitored closely enough*	0.461				
There were lot of communication gaps when out on an onsite assignment	0.454				
Responsibilities for project task assignments were not clearly defined		0.736			
Adequate time was not spent on various phases of software development (such as coding, testing)		0.731			
The project was started without proper feasibility studies		0.701			
The project required a change in currently used tools and techniques		0.677			
Inappropriate development methodology was used in this project		0.613			
The project was over dependent on a few key people		0.578			
The project planning was very poor		0.556			

The project team had the freedom to select the development platforms and tools		0.515			
Resource requirements were incorrectly estimated		0.511			
The project necessitated working on outdated technologies*		0.51			
Project schedules and budgets were continuously revised		0.496			
Adequate reference material was not available		0.459			
Project manager had multiple projects to manage at the same time.		0.459			
Performance measurements of individual members was incorrectly done*		0.42			
The procedures prescribed by quality standards were not strictly followed*		0.402			
The project involved modification of an existing software		-0.641			
The project manager had the freedom to select the project team		-0.72			
Too many external agencies were involved in the development project			0.672		
Women members had restrictions while traveling and staying outside			0.57		
Visa rejections to foreign countries was a major risk			0.567		
The telecommunication network was slow and unreliable			0.524		
Subcontractors were not meeting their commitments.			0.486		
The project had a clearly identified client / sponsor				0.806	

Clients did not have software experience				0.677	
Client expectations were unrealistic				0.674	
Documentation was very poor				0.641	
There were conflicts among client representatives				0.546	
There was a lack of cooperation from clients				0.487	
Large number of links were required to other systems					0.759
The project requirements were changed continually					0.60
The project had highly complex requirements					0.582
This was one of the largest projects attempted by the organization					0.509
Members who had developed the system specifications were doing the coding*					0.43

* deleted in the later stages

Factor Analysis on Risk Management Strategies

A similar exercise of factor analysis was performed on the risk management items. There was very limited literature support as no major factor structures were reported from international/national studies. But many authors had presented various schemes and strategies for risk mitigations which were reviewed in Chapter 2.

Looking at the criteria of literature support, eigen value, percentage of variance explained and the interpretability, a four factor structure was accepted as the best solution. This structure explained 59.5% of the variance. Kaiser–Meyer–Oaklin measure of sampling adequacy (KMO value) and Bartlett Test of Sphericity were found to be satisfactory. Items that did not load on a factor or items that crossload on multiple factors were identified and were deleted or reworded. The final list modified based on the inputs from these three analyses (non response, item total correlation and factor analysis) had 38 items under risk management.

A close examination of the items loading on to a factor resulted in naming the factors as shown below.

Factor 1 covered strategies adopted during the execution of the project. Examples are technical reviews, change request, modifying requirements etc. This factor was named as **Execution Management.** Factor 2 covered strategies for human resource management namely training, flexible working hours, rewards etc. This factor was called **Human Resource Management**. Factor 3 focused on strategies for client management. This factor was named **User Coordination**. Factor 4 had items related to project planning effectiveness. This was named **Project Planning**.

The percentage of variance explained by the factors is shown in table 3.4. The factor loading matrix is shown in Table 3.5.

Table 3.4:The percentage of variance explained by Factors of Risk Management construct

Factor No	Factor Name	No of items under the factor	% of variance explained
1	Execution Management	13	19.2 %
2	Human Resource Management	11	15.9 %
3	User Coordination	8	13.9 %
4	Project Planning	6	10.5 %
	Total	38	59.5%

Table 3.5: Factor loading matrix for Risk Management

	Factors (Dimensions)			
Risk Management Items	I	II	III	IV
There was an effective configuration management system	0.654			
Formal review of status reports versus plan was made periodically	0.604			

Formal user specification approval process was followed	0.601			
Post project audits were carried out to learn from previous projects	0.599			
Regular technical status reviews were conducted	0.589			
Simulation and scenario analysis was performed to anticipate future problems	0.568			
Risk assessment was performed regularly throughout the project.	0.556			
Job-matching was done to ensure that the right person gets the right job	0.548			
Minutes of the project team meetings was prepared and circulated among members	0.533			
Benchmarking was applied to ensure best quality software	0.511			
Project leaders were trained in project management techniques	0.502			
Once requirements were frozen, no change request was entertained	0.485			
Prototyping methodology was used in most of the cases	0.462			
The organization had very flexible working hours and stress was laid on completing the work in		0.698		
Promotions and salaries were tied to individual performance		0.662		
Outside technical assistance was sought whenever required		0.643		
Working beyond office hours was recognized and rewarded		0.625		
Attendance system was strictly enforced in the organization		0.539		

The organization structure was very flat		0.519		
Employees were asked to sign bonds to ensure they stay with the organization for a minimum		0.509		
An assistant project manager / leader was appointed for the project		0.477		
Adequate training was given to employees to make them competent		0.442		
Employees were consulted before they were assigned to a project		0.43		
HR department was very proactive and helpful		0.423		
Users evaluated the progress of the project regularly			0.643	
Effort was always made to hide the complexity from the user			0.613	
User's manuals were prepared with a lot of care			0.493	
Coordination with the user was performed through formal procedures			0.492	
There were informal contacts and communication channels between project			0.465	
Compatibility analysis was done with sub contractors and suppliers*			0.463	
Modifying from the existing system was preferred over developing from scratch*			0.431	
The user steering committee was very active			0.408	
Software re-use was done wherever possible				0.617
Planning tools were extensively used in this project				0.584

The software was always designed at minimum cost				0.57
Unnecessary requirements were removed before the development started				0.474
Individuals were held accountable for the tasks assigned to them				0.452
Detailed, multi source cost and schedule estimation was done as part of project planning				0.546

* deleted later

3.2.5 MTMM Tests for Inter rater reliability

A series of tests under the Multitrait-Multimethod (MTMM) category can be used to check for variation in responses across multiple methods or multiple informants used in data collection. As the present research used single informant, care had to be taken to minimize the effect of reporting bias.

The following measures suggested by Boyer and Verma (2000) were used to assess inter-rater agreement.

Inter-rater reliability (IRR)

IRR is a correlation measure used to assess inter-rater consistency. When there are two informants reporting on a given variable, IRR would be the correlation between the responses from two informants. A significant correlation coefficient indicates high inter rater agreement.

Simple paired t-test.

Paired t-test checks for a non-zero difference between the values tested. The paired t-test checks for a non-zero difference between the values tested. A non significant t- value indicates that the difference between the scores of the informants is not statistically different from zero and hence is an indicator of high inter-rater agreement.

For the purpose of MTMM test, the researcher collected data from 30 projects with a leading software company in Trivandrum. Three respondents were identified for each project: Two project members and one user representative. The questionnaire was *independently* administrated for data collection. The project members were asked to rate the project on the 60 items under risk and 38 items under risk management. One project member and the user representative had to answer the questions on project outcome (nine questions on quality and three each on time/cost overrun). The average score was calculated for each risk/risk management dimension (factor) by taking the average of all items loading onto that risk/risk management factor. Average quality score was calculated by taking average of the scores on the nine quality dimensions.

Correlation Analysis and paired t-test were performed on the response scores from the two informants (project members for risk/risk management factors, project member and user representative for outcome measures). The results are shown in Table 3.6

Table 3.6: Results of the correlation analysis and paired T tests on two informants

	Significance of the correlation coefficient between the values given by informants	Significance of the difference in response scores of the two informants (paired t test)
Team Risk	Significant	Non significant
Project planning and	Significant	Non significant
External Risk	Significant	Non significant
User Risk	Significant	Non significant
Complexity Risk	Significant	Non significant
Project Management	Significant	Non significant
HRM	Significant	Non significant
User coordination	Significant	Non significant
Project Planning	Significant	Non significant
Quality	*Non Significant*	*Significant*
Time Overrun	Significant	Non significant
Cost Overrun	Significant	Non significant

As seen from the table, all measures except quality shows high inter-rater agreement. A detailed analysis of the items under quality measure showed that the following two items are contributing to the disagreement.

(1) The quality of the development process

(2) Satisfaction of the user with the developed software

The Project member is better equipped to answer item (1) and the user is more qualified to answer item (2). Hence it was felt that these items would have informant bias and hence were dropped from the final questionnaire.

3.2.6 Instrument for final survey

The final instrument was developed from the pilot study questionnaire, incorporating all the modifications and corrections mentioned above. The final instrument had 60 variables representing the risk construct and 38 variables under risk management. The respondent was asked to indicate to what extent these risk and risk management items were present in his/her last completed project for which he/she was selected as the representative. The respondent had to indicate the presence of each risk/risk management item in the project on a five point Likert scale (strongly disagree; disagree; neither disagree nor agree; agree and strongly agree).

Project outcome was measured in terms of time overrun, cost overrun and quality. Time and cost overruns were measured with three questions each. (1) Percentage of overrun (2) budgeted value (3) actual value. The respondent was requested to consult the project documents/appropriate authority to get the correct overrun data. Product quality was measured through a five point rating scale where the respondent rated the software developed on seven dimensions of software quality namely Reliability, Ease of use, Maintainability, Portability, Flexibility, Testability and Documentation Quality. The average score on these seven dimensions was taken as a measure of quality

A single informant, self reporting methodology was adopted for data collection. Though this method has potential reporting bias, it is very commonly used in academic research (Pinsonneault and Kraemer, 1993; Wallace et al, 2004; Barki, 1993; Jiang et. al., 2000, Ropponen and Lyytinen, 1997, Deephouse et al 2005). Inter-rater reliability tests and Harman's one factor tests also showed favorable results. Please refer to annexure 2 for a note on the self reporting methodology.

The important classification/demographic variables included in the instrument were:

- Organization details: the number of employees, nature of the organization and quality certifications.

- Project details: type of project, duration of the project, number of team members, onsite/off shore content of the project and country of the user.
- Personal details: Role of the respondent in the project

The instrument was organized in three parts. Part A had 28 questions related to the classification/demographic information and also the project outcome. Part B had 60 risk variables presented in alphabetical order. Part C had the 38 risk management variables again presented in alphabetical order. The instrument is shown in Appendix 1.

3.3 SUMMARY AND CONCLUSION

This chapter presented various aspects of research methodology used in the study. It also explained the questionnaire development process. The draft questionnaire prepared based on literature review was edited by experts to improve its content and face validity. Exploratory factor analysis was performed on the pilot data to understand the underlying factor structure. This exercise helped in variable reduction and also to identify items for further editing based on the level of loading. A five-factor structure emerged for risk construct and a four-factor structure was accepted for the risk management construct. These factor structures are comparable to those developed by previous researchers. The exact number and names of the factors may differ, but the themes are properly captured.

After incorporating the modifications and corrections from this exercise, the final instrument to be used for the final survey was developed.

****

CHAPTER 4

DATA COLLECTION AND VALIDATION OF THE FINAL INSTRUMENT

Chapter 4 builds on the discussion in chapter 3 on the concept of instrument development and its empirical validation. The instrument development process was explained in the previous chapter. This chapter looks at the final data collection and confirmatory analysis using the final data. Section 4.1 describes the final sample profile. Section 4.2 explains the reliability analysis using Cronbach Alpha. Section 4.3 describes the validity and dimensionality analysis of the instrument. Section 4.4 discusses the generation of risk factor scores and risk management factor scores for each project. Section 4.5 presents the conclusion of the chapter.

4.1 SAMPLE PROFILE

The sampling design was described in section 3.1.6 of chapter 3. The researcher distributed 1350 questionnaires to the 105 companies who agreed to participate. After two rounds of reminders, 574 filled questionnaires were collected back from 95 companies. Detailed examination of the data based on grossly missing or inappropriate values resulted in the deletion of 47 records. Thus the final data set had 527 usable records representing 527 projects from 95 companies. Table 4.1 gives sample collection details

Table 4.1: The sample collection details

Location	No of companies agreeing to participate	No. of companies participated	No. of responses (projects represented)	Final no. of records
Bangalore	59	44	231	221
Chennai	34	30	189	173
Cochin	6	6	40	29
Trivandrum	15	15	114	104
Total	105	95	574	527

On checking the sample data, it was found that the following groups are sufficiently represented: (a) Project representatives in different roles (b) different types of projects (c) organizations of different sizes. Sample details in terms of project and organizational characteristics were analyzed. The details are reported in the form of a series of tables. From these tables given below, it is clear that the sample possessed considerable diversity.

Table 4.2 shows the role of the respondent in the project. Most of the respondents were project members or project leaders

Table 4.2 Role of Respondents in the project

Role	Number	Percent (%)
Project Manager	50	9.5
Project Leader	84	15.9
Project Member	270	51.2
QA Member	32	6.1
System Analyst	43	8.2
Implementation Member	33	6.3
Others	15	2.8
Total	527	100.0

As seen in table 4.3, majority of the sample projects came from Business Application domain, followed by the Engineering Applications and Web Applications. This is in line with the industry trend.

Table 4.3 Type of the project in the sample

Domain	Number	Percent (%)
Business Application	254	48.3
Engineering Application	106	20.2
System Software	55	10.5
Web Application	92	17.5
Others	19	3.6
Missing Values	1	0.2
Total	527	100

The table 4.4 describes the distribution based on team size. Majority of the projects in the sample had team size between 5- 20 members. The responses revealed that team size varied from one person to 220 members.

Table 4.4 Classification of projects based on the number of members in the project team

Team size	Number	Percent (%)
Upto 5 members (small)	134	25.4
5 upto 20 members medium)	281	53.3
More than 20 members (large)	102	19.4
Missing Values	10	1.9
Total	527	100

Projects executed for clients from different countries as well domestic projects were accounted for in the sample as shown in Table 4.5. Most of the projects were for the USA clients followed by European and Asian countries. This is representative of the actual scenario.

Table 4.5 Classification of projects based on the country of origin of the client

Country	Number	Percent (%)
India (domestic)	28	5.3
USA	314	59.6
UK	42	8.0
Other European countries	45	8.5
Canada	5	0.9
Australia	30	5.7
Japan	6	1.1
Other Asian Countries	25	4.7
Others	32	6.1
Total	527	100.0

The sample showed a reasonably good mix of onsite and off shore projects. Majority of projects had both onsite and offshore content. The details are presented in table 4.6.

Table 4.6 Classification of projects based on onsite /offshore nature of the project

Onsite /Offshore projects	Number	Percent (%)
Offshore	177	33.6
Mixed	298	56.5
Onsite	45	8.5
Missing values	7	1.3
Total	527	100

Details regarding the duration of the projects in the sample are presented in table 4.7. Over 70% of the projects were of less than 2 years duration. Detailed scrutiny of the data showed that the mean duration is 11.7 months.

Table 4.7 Classification of projects based on project duration

Duration of the project	Number	Percent (%)
Less than 6 months (short)	198	37.6
More than 6 months upto 2 years (	185	35.1
2 years and above (long)	78	14.8
Missing values	66	12.5
Total	527	100.0

Table 4.8 shows the distribution of responses based on the organization size in terms of number of employees. The data showed that organizations of varied sizes were represented sufficiently in the final sample.

Table 4.8 Classification of projects based on number of employees in the Organization

Number of employees	Number	Percent (%)
Upto and including 100 members	65	12.3
Above 100 upto and including 1000	167	31.7
Above 1000 upto and including 10000	154	29.2
Above 10000	115	21.8
Missing Values	26	4.9
Total	527	100

Data was collected on the nature of the organization. Most of the responses came from Indian companies with international projects (such as Wipro and Infosys) or from the Indian subsidiaries of MNCs (such as Accenture, HP etc). The details are shown in table 4.9.

Table 4.9 Classification of projects based on the nature of the organization

Nature of the organization	Number	Percent (%)
Local company with domestic projects	21	4.0
Indian company with International Projects	303	57.5
Subsidiary of an MNC	182	34.5
Others	5	0.9
Missing Values	16	3.0
Total	527	100

Table 4.10 shows that most of the projects in the sample were from organizations which are either ISO certified or CMM certified or both. 26.7% of the organizations had PCMM certification. 4.5% of the respondents came from companies with no certification.

Table 4.10 Classification of projects based on the quality certification of the Organization (overlapping permitted)

Certification	Number	Percent (%)
ISO certified	456	79.4
CMM Certified	492	85.7
PCMM certified	153	26.7
Other certifications	213	37
No certifications	25	4.5
Missing values	7	1.2
Total	527	100

The sample was classified in terms of the project outcome variables namely cost overrun and time overrun. The data regarding the cost performance was available from 320 projects. 45% of them reported cost overrun. The details regarding the performance of the projects with reference to the time dimension could be collected from 390 projects. 35% of them reported time overrun. The details are shown in table 4.11 and table 4.12.

Table 4.11 Classification of projects based on cost performance

Cost performance	Number	Percent (%)
Cost underrun	37	7
No underrun or overrun	140	26.6
Cost overrun	143	27.1
Missing Values	207	39.3
Total	527	100

Table 4.12 Classification of projects based on time performance

Time performance	Number	Percent (%)
Time underrun	56	10.6
No underrun or overrun	190	37.4
Time overrun	137	26
Missing Values	137	26
Total	527	100

Table 4.13 presents the cross tabulated data covering both time overrun and cost overrun. 82 projects reported on time and on cost completion. 53 projects had both time and cost overrun.

Table 4.13 Cross tabulation of projects based on cost and time performance (in numbers)

		Time Performance		
Cost performance		Underrun	No run	Overrun
	Underrun	20	14	3
	No run	9	82	41
	Overrun	8	75	53

4.2 RELIABILITY ANALYSIS

Reliability of an instrument is defined as the extent to which any measuring instrument yields the same result on repeated trials (Carmines and Zeller, 1990). It is the degree to which the instrument yields a true score of the variable (factor) under consideration. The instrument is not considered as reliable to the extent to which it contains measurement error (Neale and Liebert, 1986).

There are several methods to establish the reliability of a measuring instrument. These include test-retest method, equivalent forms, split-halves method and internal consistency method. Of all these methods, the internal consistency method is the most popular method, especially in field studies. The advantage of this method is that it requires only one administration, and consequently this method is considered to be the most general form of reliability estimation (Sureshchandar et al., 2001). In this method, reliability is operationalized as 'internal consistency', which is the degree of inter-correlation among the items that constitute the scale (Nunnally, 1978). The internal consistency can be estimated using a reliability coefficient called Cronbach's alpha (α) (Cronbach, 1951). An alpha value of 0.70 or above is considered to be the criterion for demonstrating strong internal consistency of established scales (Nunnally, 1978).

The reliability of the instrument developed in the current study was tested by computing Cronbach alpha (α) value for each of the five risk factors as well as for the entire set. The item-total correlation was tested for each risk item under each factor. Based on a detailed scrutiny of the item correlation matrix of the items, the following items were removed from further analysis.

- Members who had developed the system specifications were doing the coding
- Performance measurements of individual members was incorrectly done
- The project necessitated working on outdated technologies

The final values of Cronbach alpha for the risk factors are presented in Table 4.14. As seen from the table, all the factors had Cronbach alpha value above 0.7, which testified the reliability of the instrument.

Table 4.14 Results of Reliability Analysis for Risk

Sl No	Factors of Risk	No. of items	Cronbach' Alpha (α)
1	Team Risk	26	0.9667
2	Project Planning And Execution Risk	16	0.9196
3	External Risk	5	0.8466
4	User Risk	6	0.8309
5	Project Complexity Risk	4	0.7663
	Overall fit	57	0.9765

Cronbach alpha (α) values were computed for the risk management factors also. One factor namely user coordination, showed an alpha value less than 0.7. A close examination of items under this dimension resulted in the removal of the following items.

- Compatibility analysis was done with sub contractors and suppliers
- Modifying from the existing system was preferred over developing from scratch

Table 4.15 shows the final reliability scores obtained for Risk Management which comprises of four factors. All factors as well as the overall scale were seen to have Cronbach alpha above the acceptable threshold of 0.7.

Table 4.15 Results of Reliability Analysis for Risk Management

Sl	Risk Management	No. of	Cronbach's
1	Execution Management	13	0.9204
2	Human Resource Management	11	0.8630
3	User Coordination	6	0.7001
4	Project Planning	6	0.8059
	Overall fit	36	0.9536

4.3 VALIDITY AND DIMENSIONALITY ANALYSIS

Evidence for 'convergent validity' is obtained when a measure correlates well with other measures that are believed to measure the same construct (Kaplan and Scauzzo, 1993). In other words, convergent validity is the degree to which various approaches to construct measurements are similar to (converge on) other approaches that they theoretically should be similar to (Sureshchander et al., 2001).

Unidimensionality refers to the existence of a single construct/trait underlying a set of measures (Hair et al., 1998). The most important and fundamental assumption in measurement theory is that a set of items forming an instrument measures just one thing in common. Items within a measure are useful only to the extent they share a common nucleus - the characteristics to be measured (Nunnally, 1978).

Because risk and its five dimensions as well as risk management and its four dimensions could not be directly observed, they were considered to be latent variables. A latent variable is a construct which “cannot be measured directly, but can be represented or measured by one or more variables (indicators)” (Hair et. al., 1998). For example, the team risk could not be measured precisely or directly observed, but questions could be asked to assess how team issues might increase the riskiness of a software development project. By combining the answers to these questions a measure of the underlying (latent) construct of team risk could be obtained.

Confirmatory Factor Analysis (CFA) which is part of the structural equation modeling techniques can be used to estimate a measurement model that specifies the relationship between observed indicators and their underlying latent constructs. The measurement model specifies how latent constructs are measured by the observed variables and it assesses the construct validity and reliability of the observed variables (Joreskog and Sorbom, 1989). Estimation of the measurement model is used to assess the fit of the data to a hypothesized model. CFA is often used when the number of factors is known beforehand and each variable is allowed to associate with only one factor.

Confirmatory Factor Analysis was performed on the final data to confirm the structure developed with the Exploratory Factor Analysis on the pilot study data. The general paradigm suggested by Anderson and Gerbing (1988) was followed in the current research in order to test a model of software project risk and software project risk management. The measurement models of each of the five dimensions of software project risk were first assessed and then a structural model linking these five dimensions was tested. The same exercise was repeated for software project risk management.

Software package AMOS 4.0 was used to do the Confirmatory Factor Analysis. The following are the commonly used fit indices which help to assess the fit between a model and a data set which in turn proves its validity.

The Goodness-of-Fit Index (GFI): This is one of the most commonly reported measures of model fit. The GFI is a non-statistical measure that ranges in value from 0 (poor fit) to 1 (perfect fit). The higher the GFI, the better the model fit is considered to be. There is no definitive value that indicates "good" model fit, although values above .90 are usually considered to be favorable to conclude that there is a good fit between the proposed model and the observed data. However, the GFI does not take sample size into account, so its value may be biased upwards in large samples.

Adjusted Goodness-of-Fit Index (AGFI) The AGFI is an extension of the GFI that attempts to address the issue of sample size. The AGFI is similar to the GFI, but it is adjusted by the ratio of the degrees of freedom for the proposed model to the degrees of freedom for the null (Hair et al., 1998). Again, there is no set standard for an acceptable AGFI. In some instances, values of .80 or greater are often considered an indication of good model fit (Taylor and Todd, 1995), although values as low as .70 have been considered acceptable (e.g., Smith, 1996; Straub, 1995).

The Comparative Fit Index (CFI) is another measure of overall goodness of fit that uses a Chi-square distribution. The CFI produces a value between 0 and 1, with 1 indicating a perfect fit. As a rule of thumb for this statistic, values of .90 or better are often considered to indicate good fit (e.g., Chin and Todd, 1995), although there has been some argument in favor of lower values suggesting adequate fit as well. This is one of the most used measures of unidimensionality of the scale.

Bentler-Bonett Fit Index (NFI or TLI). NFI or TLI is a good indicator of the convergent validity of the questionnaire. A scale with TLI values of 0.9 or above is an indication of strong convergent validity (Bentler and Bonett, 1980).

More discussion on structural equation modeling is given in chapter 8.

4.3.1 CFA on Risk dimensions

The CFA was performed on each of the five dimensions of software project risk as well as on the full model. Each risk dimension was tested independently to assess the validity. Figure 4.1 shows a typical measurement model drawn in AMOS 4.0 for Team risk dimension. The model pictorially represents how the risk items (indicators) are linked to a latent variable called Team risk. Similar models were drawn and tested for the other dimensions namely project planning and execution risk, user risk, external risk and project complexity risk.

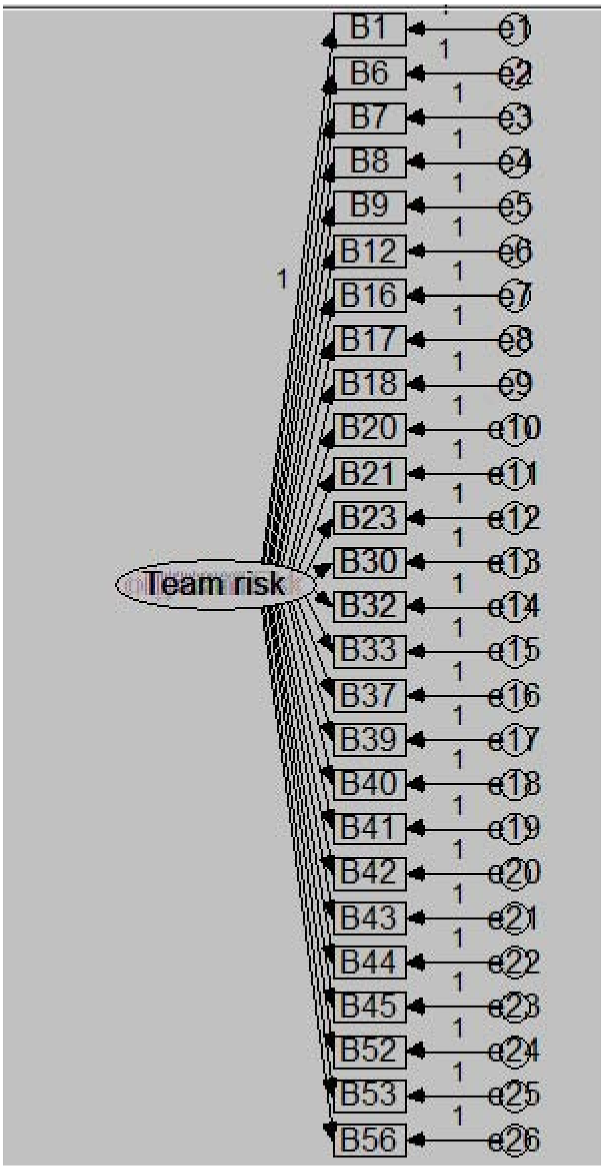


Fig 4.1 Measurement model for the Team risk dimension of software project risk

AMOS 4.0 represents the latent variables through ovals and indicators through rectangles. The arrows show the relationships. Every indicator term has to be associated with a measurement error term. In the figure, the B1, B6, B7 etc are the risk items loading on to the factor Team risk. e1 to e26 are the associated error terms.

All the indices namely GFI, AGFI, CFI and NFI were above the minimum required levels for all dimensions except for the project planning and execution risk.

If a scale is found to violate the above stipulations, its items are examined and those with the least item-total correlations are removed enhance the fit indices beyond the minimum requirements. While doing so, it is mandatory to call for the researcher's judgment, as otherwise, reliable scale lacking content validity will result (Ahire et al., 1996). Keeping this in mind, a detailed scrutiny of the items under the project planning and execution risk was performed and this resulted in deletion of the following items

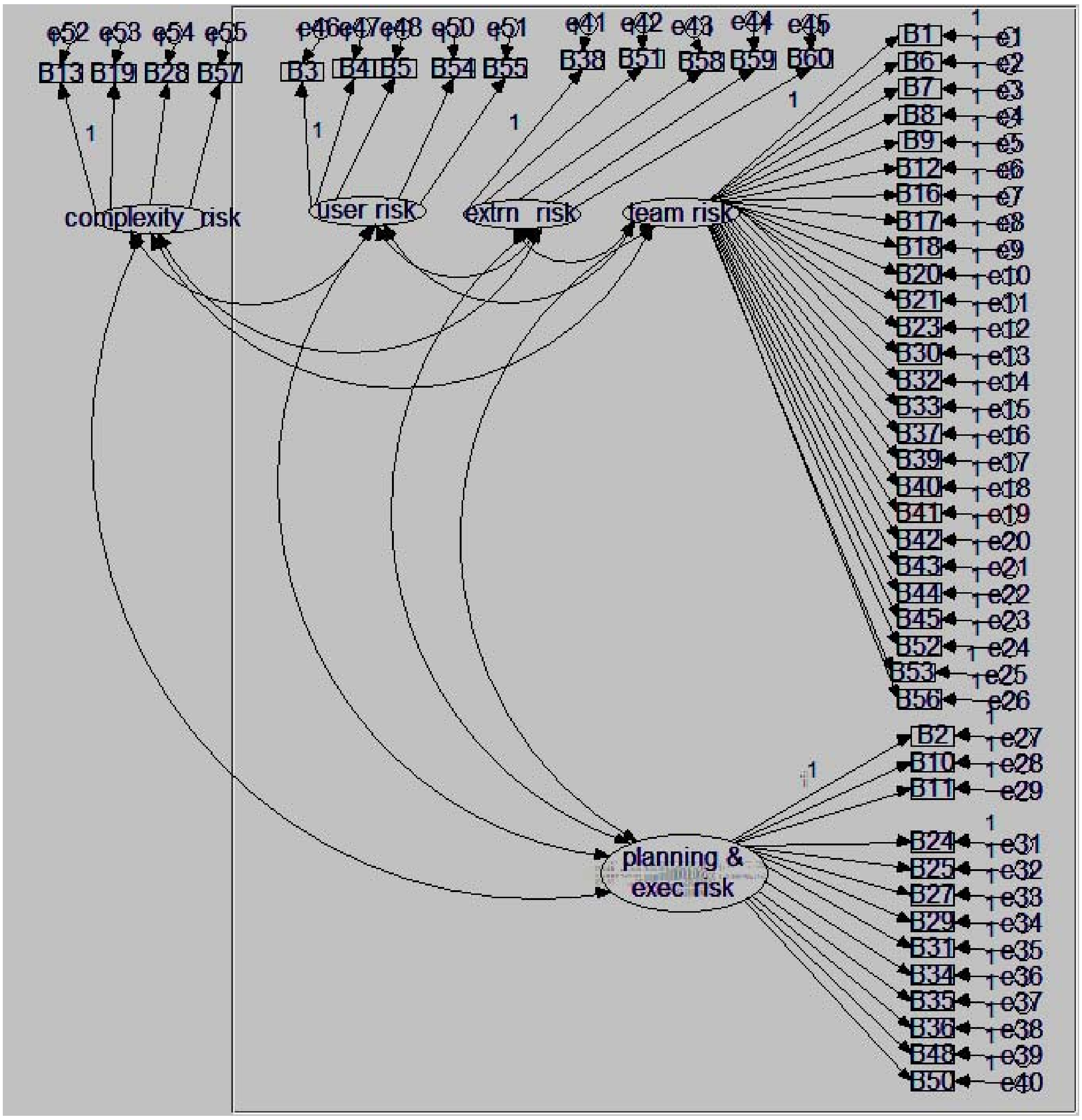
- The project progress was not monitored closely enough
- The procedures prescribed by quality standards were not strictly followed

The modified model was tested again and all indices were seen to be above the minimum threshold. The values are reported in table 4.16.

Table 4.16 Results of Confirmatory Factor Analysis for the Dimension of Risk

Sl No	Factors of Risk	No. of items	GFI	AGFI	CFI	NFI
1	Team Risk	26	0.905	0.889	0.955	0.927
2	Project Planning And Execution Risk	14	0.804	0.732	0.787	0.773
3	External Risk	5	0.914	0.743	0.894	0.894
4	User Risk	6	0.933	0.843	0.932	0.927
5	Project Complexity Risk	4	0.995	0.973	0.993	0.990

This was followed by the testing of the full model of software project risk linking these five dimensions. Confirmatory factor analysis helps to test that indicator variables load highly on predetermined factors, but do not load highly on unrelated factors (Hatcher, 1994). Each factor was connected to every other factor by a two-headed arrow, meaning that every factor was allowed to co-vary with every other factor. However, none of the indicators were allowed to co-vary, which is standard practice with structural equation modeling (Hatcher, 1994). The measurement model is shown in figure 4.2 and the CFA results are produced below the figure.



GFI = 0.908 **AGFI =0.848** **CFI = 0.895** **NFI=0.896**

Fig 4.2 Full Measurement Model for the software project risk

These findings supported the validity and reliability of the constructs and their indicators. These results indicated a reasonable fit between the proposed model and the collected data. Similar fit measures have been found acceptable in previous research (Hair et al., 1998; Henry and Stone, 1994). Thus it can be concluded that the final data supported the proposed five factor structure of risk.

4.3.2 CFA on Risk management dimensions

The CFA was performed on each of these four dimensions of risk management. Each risk management dimension was tested independently to assess the validity. Figure 4.3 shows a typical measurement model drawn in AMOS 4.0 for HR management dimension. Similar models were drawn and tested for the other dimensions namely execution management, user coordination and project planning.

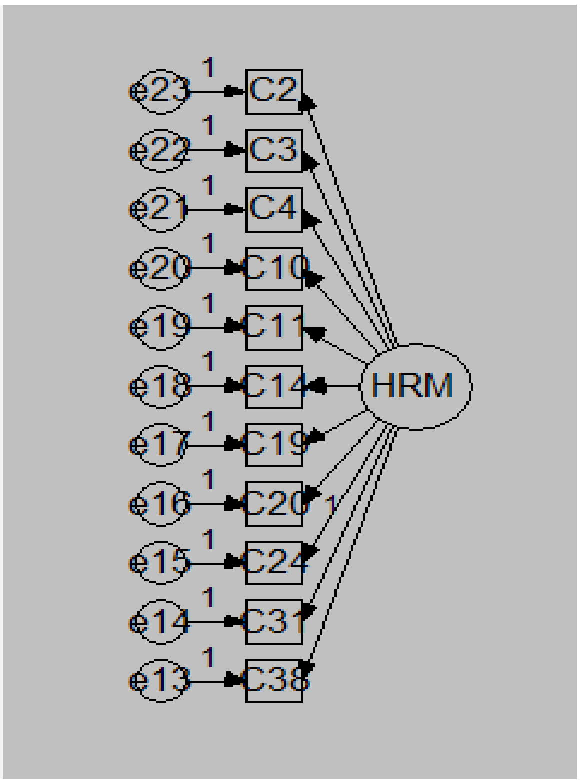


Fig 4.3 Measurement Model for the HR management dimension of risk management

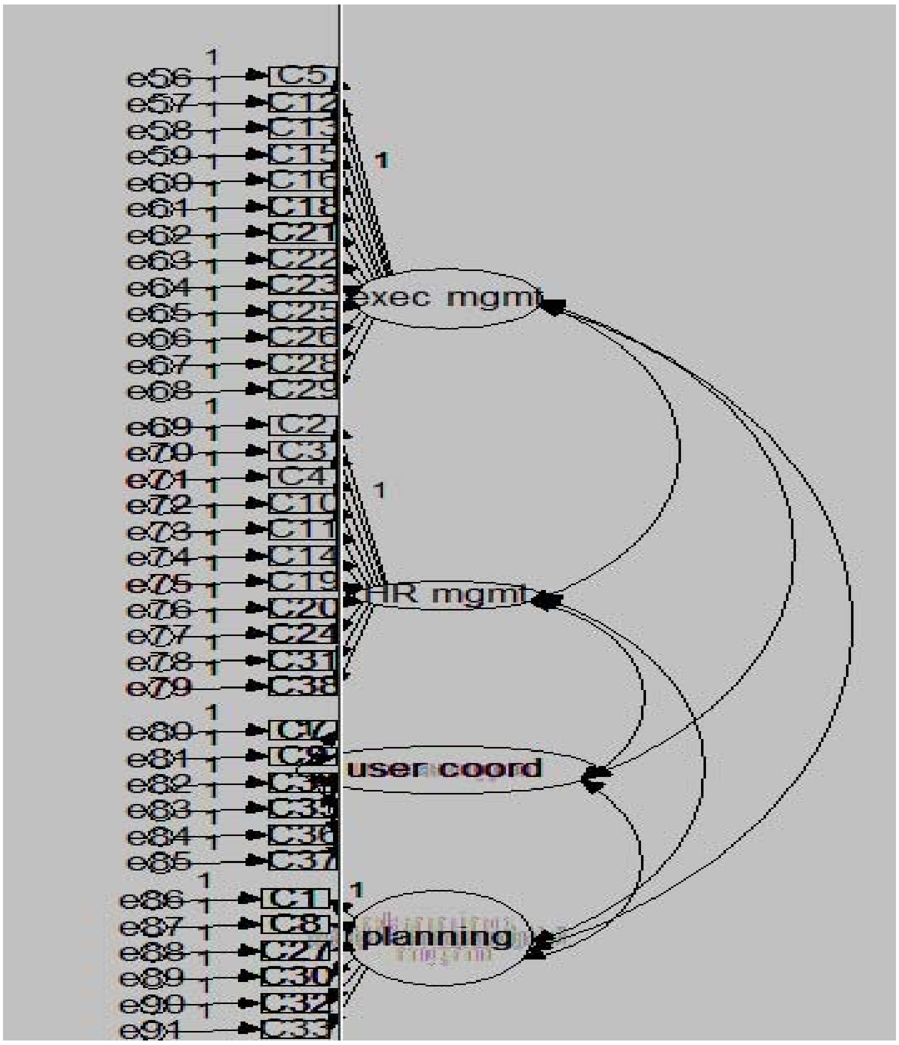
All the indices namely GFI, AGFI, CFI and NFI were above the minimum required levels for all dimensions of risk management. The values are reported in table 4.17.

Table 4.17 Results of Confirmatory Factor Analysis for the Dimension of Risk Management

Sl No	Factors of Risk Management	No. of items	GFI	AGFI	CFI	NFI
1	Execution Management Strategies	13	0.937	0.912	0.953	0.937
2	Human Resource Management Strategies	11	0.930	0.895	0.913	0.895
3	User Cordination Strategies	6	0.983	0.961	0.975	0.965
4	Project Planning Strategies	6	0.994	0.986	0.999	0.989

The above table clearly shows that all dimensions of risk management show sufficient goodness of fit since all the values are above 0.8 and most of them are above 0.9 which is considered to be very high statistically. This proved the high convergent validity and unidimensionality of the risk management construct.

As in the case of risk, a full model was tested linking the four dimensions of the risk management. The model is depicted in figure 4.4 and the CFA results are stated below.



GFI = 0.921 **AGFI =0.853** **CFI = 0.914** **NFI=0.907**

Fig 4.4 Full Measurement Model for the software project risk management

Combined, these findings support the validity and reliability of the constructs and their indicators. Thus the proposed four factor structure of risk management was accepted.

4.4 GENERATING THE RISK SCORES AND RISK MANAGEMENT SCORES

After the factor structure had been identified and validated, the next step was to generate appropriate scores to represent these factors for further statistical analysis. There were three popular methods available for the researcher. 1. Use one variable from the set of variables loading on to a factor as a surrogate to represent that factor. 2. Use summated scale formed by combining all variables loading heavily onto that factor. 3. Use factor score computed by the statistical package based on the rotated factor loading matrix.

For an instrument whose validity, reliability and dimensionality have been proven, summated scales are recommended (Hair et. al 1998) from the following considerations. (a) These scales are easily replicated across studies. (b) They are easy to interpret. (c) They represent multiple aspects of the concept under measure.

The researcher used summated scales for this study. The score was calculated for each risk factor (dimension) by taking the average of all items included under that risk factor. Similarly risk management scores were also calculated for each of the four factors of risk management by taking the average of all items included under that risk management factor.

4.5 CONCLUSION

The final sample had sufficient number of projects represented from all domains namely business application, engineering application, web application and system software. The data showed that the responses came from different organizations showing considerable diversity in sample. There was a good mix of responses from onsite and offshore projects.

The internal scale reliability (Cronbach α) values for the five factors of risk as well as for the four factors of risk management were found to be acceptable. Confirmatory Factor Analysis (CFA) performed on the final data confirmed the factor structure derived from the pilot data.

Hence it can be concluded that all the factors used in this study exhibit high unidimensionality, convergent validity and reliability. The instrument thus standardized can be used to measure the levels of risk and risk management in software development projects with software development organizations in India

****

CHAPTER 5

ANALYSIS OF RISK AND RISK MANAGEMENT CONSTRUCTS

This chapter provides insights into the nature of risk and risk management constructs through the testing of a series of hypotheses. Section 5.1 introduces the statistical techniques used for hypothesis testing. Section 5.2 presents the results of the hypotheses testing. Section 5.3 gives the summary and conclusion of the chapter.

5.1 INTRODUCTION

Section 3.1.5 of chapter 5 presented five hypotheses on the anticipated relationship among the variables in the study. This section presents the results of the hypothesis tests.

Risk, risk management and project outcome measures are metric in nature whereas project and organization characteristics are categories. Correlation (Pearson correlation and canonical correlation) techniques were used for testing hypothesis H1 to H3 which were about relationship among metric variables. ANOVA was used for testing hypothesis H4 and H5 which checked for variation of the metric dependent variable risk across categories of the independent variables such as project characteristics and organizational characteristics.

Correlation is a bivariate measure of association (strength) of the relationship between two metric variables. Pearson correlation is the most popular measure of correlation, sometimes called *product moment correlation*. Pearson's r is a measure of association which varies from -1 to +1, with 0 indicating no relationship (random pairing of values) and 1 indicating perfect relationship. A canonical correlation is the correlation of two canonical (latent) variables, one representing a set of independent variables, the other a set of dependent variables. The purpose of canonical correlation is to identify the relation of the two sets of variables, not to model the individual variables.

Pearson and canonical correlations share the following set of assumptions which were met by the variables under study.

1. Metric data: Risk, risk management and project outcome measures were metric in nature

2. Linear relationships: This was checked through the x-y scatter graph of risk – risk management component pairs, risk – project outcome measures and risk management- project outcome measures. It was seen that the two variables being correlated can be better described by a straight line than by any curvilinear function
3. Homoscedasticity was assumed: The error variance was assumed to be the same at any point along the linear relationship.
4. No outliers. Scatter plots were used to spot outliers visually. No outliers were spotted for the variables under study.
5. Common underlying normal distributions, for purposes of assessing significance of correlation. The central limit theorem demonstrates, however, that for large samples, indices used in significance testing will be normally distributed even when the variables themselves are not normally distributed, and therefore significance testing may be employed.

Analysis of variance (ANOVA) is used to uncover the effects of categorical independent variables (called "factors") on an interval dependent variable. The key statistic in ANOVA is the F-test of difference of group means, testing if the means of the groups formed by values of the independent are different enough not to have occurred by chance. If the group means do not differ significantly then it is inferred that the independent variable(s) do not have an effect on the dependent variable. If the F test shows that, overall, the independent variable(s) are related to the dependent variable, then *multiple comparison tests* of significance are used to explore which values of the independent variable(s) have the most to do with the relationship.

Some key ANOVA assumptions are that the groups formed by the independent variable(s) are relatively equal in size and have similar variances on the dependent variable ("homogeneity of variances"). Like any other parametric tests, ANOVA also assumes normality.

In the present study, the assumption of equal sizes were not met for most of the subgroups as evident from respondent attributes analysis in chapter 4. But ANOVA implemented in SPSS has built- in mechanism to support unequal group sizes. Hence this will not affect the interpretation very much. The assumption that the dependent variable should have the same variance in each category of the independent variable was tested through Levene's test of homogeneity of variance.

The test failed to indicate equal variance in certain cases. However failure to meet the assumption of homogeneity of variances is not fatal to ANOVA. Moore (1995) suggested a rule of thumb that the ratio of largest to smallest group variances should be 4:1 or less. This condition was met in all cases in this study.

5.2 HYPOTHESES TESTING

This section presents the output of the hypotheses testing. The result of the testing of each hypothesis is presented under separate subsection.

5.2.1 Testing hypothesis 1

H1: Risk management has significant negative relationship with risk

H1 is the alternate hypothesis stated to test the null hypothesis that risk management has no significant relationship with risk. Correlation values were computed for each pair of the five risk dimensions and four risk management dimensions. The significance is tested at 5% significance (1-tail) level. The results are presented in the table 5.1.

Table 5.1 Correlation between risk dimensions and risk management dimensions

		Execution management	HR Management	User Coordination	Project Planning
Team risk	Correlation	-.374*	-.312*	-.292*	-.426*
	Sig. (1-tailed)	.000	.000	.000	.000
planning and execution risk	Correlation	-.416*	-.413*	-.354*	-.464*
	Sig. (1-tailed)	.000	.000	.000	.000
External Risk	Correlation	-.086*	-.004	-.017	-.212*
	Sig. (1-tailed)	.023	.409	.349	.000
User Risk	Correlation	-.443*	-.370*	-.349*	-.496*
	Sig. (1-tailed)	.000	.000	.000	.000
Complexity Risk	Correlation	-.221*	-.209*	-.193*	-.313*
	Sig. (1-tailed)	.000	.000	.000	.000

(* indicates items significant at 5% significance level.)

Out of the twenty correlation values produced, eighteen of them were found to be statistically significant at 5% level. (17 of them were significant even at 1%

level). The canonical correlation was tested between the risk and risk management variables treating them as latent variables having sub-dimensions. The first canonical correlation value was a significant **0.6265**. Thus the null hypothesis of no relationship was rejected. These findings supported the alternate hypothesis that there was a significant negative relationship between risk and risk management. The higher the risk management score, the lower the risk score.

There is enough support for this finding in literature. Risk management is concerned with a phased systematic approach to analyse and control the risks occurring in a specific context (Charette, 1996). Risk management strategies are deployed in a project so as to reduce the adverse impact of risks (Powell et. al). Ropponen and Lyytinen (1997) found that the risk levels in the software development project varied based on the extent and frequency of risk management applications. Jiang et al (2000) and Addision and Vallabh (2002) also had shown the negative relationship between risk mitigation strategies and project risk.

5.2.2 Testing hypothesis 2

H2: Risk has significant negative relationship with project outcome

The null hypothesis that risk has no significant relationship with project outcome was tested here. Project outcome was measured in terms of time overrun, cost overrun and quality of the software. Overruns are unfavorable outcome while quality is a desirable outcome. Hence the above alternate hypothesis was restated in terms of each of the project outcome measures.

H2a: Risk has significant negative relationship with quality of the software developed

H2b: Risk has significant positive relationship with cost overrun

H2c: Risk has significant positive relationship with time overrun

Each project outcome measure was correlated with each risk dimension. The table 5.2 shows the bivariate correlation between every pair of risk dimensions and project outcome measures.

Table 5.2 Correlation between risk dimensions and project outcome

		Time over run	Cost over run	Overall quality score
Team Risk	Correlation	.623	.519*	-.462*
	Sig. (1-tailed)	.000	.000	.000
Planning and execution risk	Correlation	.548*	.498*	-.367*
	Sig. (1-tailed)	.000	.000	.000
External Risk	Correlation	.495*	.384*	-.284*
	Sig. (1-tailed)	.000	.000	.000
User Risk	Correlation	.567*	.479*	-.433*
	Sig. (1-tailed)	.000	.000	.000
Project Complexity Risk	Correlation	.463*	.378*	-.263*
	Sig. (1-tailed)	.000	.000	.000

(* indicates items significant at 5% significance level.)

The results showed a significant negative correlation between the risk dimensions and the quality measure indicating that higher risk scores were associated with lower quality scores. There was a significant and moderate positive correlation between the risk scores and time overrun values indicating that as the risk scores increased, the time overrun also increased. Similarly correlation between risk and cost overrun was also significant and positive. Canonical correlation between project outcome measures and risk was computed and reported in table 5.3. All values were seen to be statistically significant.

Table 5.3 Canonical correlation between risk and project outcome

	Time overrun	Cost Overrun	Quality
Risk (5 dimensions)	0.6850	0.621	0.5506

Thus the null hypothesis that there is no relationship between risk and project outcome is rejected. Risk was seen to have positive correlation with time and cost overrun and negative correlation with quality. The nature of these relationships is explored in the subsequent chapters.

These findings on the relationship between risk and project outcome are in agreement with previous research. Wallace (1999) related risk with overrun variables and quality and reported correlation values of 0.70 and -0.52 respectively. Nidumolu's (1995) found a significant correlation value of -0.53 between risk and project performance. Barki et al's (2001) integrative contingency model also showed a significant negative correlation between risk and dimensions and project performance. Jiang et. al., (2000) related various software development risks to project outcome measures such as meeting project budget and schedules, amount of rework and quality of work. He also commented on the negative correlation between risk and project outcome.

5.2.3 Testing hypothesis 3

H3: Risk management has significant positive relationship with project outcome

As in the case of risk, this hypothesis was restated in terms of the project outcome measures namely time overrun, cost overrun and quality.

H3a: Risk management has significant positive relationship with the quality

H3b: Risk management has significant negative relationship with cost overrun

H3c: Risk management has significant negative relationship with time overrun

Table 5.4 shows the correlation between each pair of risk management dimensions and project outcome measures.

Table 5.4 Correlation between risk management dimensions and project outcome

		Time over run	Cost over run	Quality
Execution management	Correlation	-.262*	-.289*	.486*
	Sig. (1-tailed)	.000	.000	.000
HR Management	Correlation	-.233*	-.213*	.419*
	Sig. (1-tailed)	.000	.000	.000
User Coordination	Correlation	-.236*	-.269*	.406*
	Sig. (1-tailed)	.000	.000	.000
Project Planning	Correlation	-.260*	-.303*	.489*
	Sig. (1-tailed)	.000	.000	.000

(* indicates items significant at 5% significance level.)

The canonical correlation between risk management and each project outcome measure was seen to be statistically significant. The values are reported in table 5.5.

Table 5.5 Canonical correlation between risk management and project outcome

	Time overrun	Cost Overrun	Quality
Risk (5 dimensions)	0.2527	0.265	0.5140

The results showed a significant and moderate correlation between the risk management and quality. Similarly there was a low but significant negative correlation between the risk management and the time and cost overrun. Thus the null hypothesis of no relationship is rejected supporting the alternate hypothesis that there is a significant relationship between risk management and project outcomes. The nature of these relationships is explored in the next chapter.

These finding are in line with the literature though comprehensive studies linking risk management practices to project outcome are rare. Nidumolu (1995) found a low but significant positive correlation coordination mechanisms and project outcome (in the range of 0.2 to 0.36). Barki et. al., (2001) also looked into how the three selected project management strategies favorably affected the select outcome measures. Deephouse et al (2005) demonstrated a positive link between measures of software processes to the project outcome measures.

5.2.4 Testing hypothesis 4

H4: There is a significant variation in risk across categories of select project characteristics

Based on the literature, variation in risk across categories of the following project characteristics was tested. (a) Type of software developed (Ropponen and Lyttinen, 1997) (b) onsite/offshore split up of the project (Earl,1996; NASSCOM) (c) duration of the project (Brooks, 1995; McFarlan, 1982; Chaos report 1998) (d) number of members in the project team (Brooks, 1995; MaFarlan, 1982; Chaos report 1998). The alternate hypothesis H4 was restated in terms of these characteristics.

H4a: There is a significant difference in risk based on the type of software developed

H4b: There is a significant difference in risk based on the offshore/onsite content of the project

H4c: There is a significant difference in risk based on the duration of the project

H4d: There is a significant difference in risk based on the number of members in the project team

These hypotheses were tested through ANOVA to see whether there was any variation in the mean risk scores across categories of each of these project characteristics.

H4a: There is a significant difference in risk based on the type of software developed

Table 5.6 shows the ANOVA results for the variation in risk dimensions across categories of software developed.

Table 5.6 Risk Dimensions and type of software developed

Risk dimension	Business		Engineering		System		Web		Others		F-value	P-value
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD		
Team Risk	2.257	.97	2.002	.60	2.573	.94	2.227	.87	2.313	.7653	4.26	.002*
Plan &exec	2.634	.90	2.330	.79	2.918	.88	2.592	.91	2.714	8404	4.37	.002*
External	2.419	.98	2.045	.69	2.607	1.0	2.378	.90	2.200	.8137	4.80	.001*
User	2.417	.87	2.132	.64	2.709	1.0	2.483	.94	2.394	.8556	4.54	.001*
Complexity	2.642	.98	2.422	.89	2.918	.89	2.654	.84	2.539	.9834	2.67	.031*

(* indicates items significant at 5% significance level.)

All dimensions of risk showed significant differences in the mean score across various categories of software developed. Hence the alternate hypothesis that there is a significant difference in the risk based on the type of software developed is supported.

To analyze the statistical significance of this variation pair-wise, error graphs were plotted for each risk dimension. A pair of variables is statistically different if their graphs show no overlap. The graph for team risk is shown in figure 5.1. Graphs for the other risk dimensions also showed similar shapes and hence are not reproduced.

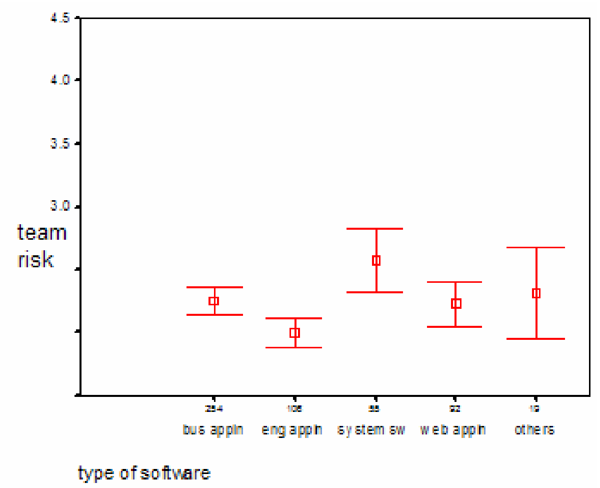


Figure 5.1 Error Bars with type of software developed on X axis and team risk on Y axis

From the graph it can be seen that engineering software shows minimum risk score. The engineering and business domains showed statistically significant difference in risk scores. Ropponen and Lyttinen (1997) found significant variation in risk across type of software developed. They found that when the requirement for user interactivity was high, the risk of the project increased. Business applications usually demanded higher user interactivity compared to engineering software. Engineering software development is more skilled and defined.

H4b: There is a significant difference in risk based on the offshore/onsite content of the project

The software development projects executed by the software companies in India were categorized based on the quantum of work done at the client side (onsite) and in the Indian development center (offshore). The variation in risk scores across the categories were analyzed and presented in the table below.

Table 5.7 ANOVA test Results for Risk based on Offshore/Onsite Content

Risk dimensions	Offshore		Mixed		Onsite		F-value	P-value
	Mean	SD	Mean	SD	Mean	SD		
Team Risk	2.1425	.6503	2.2081	.9291	2.8256	.9552	12.137	.000*
Plan & exec	2.5303	.7991	2.5654	.9475	3.1222	.8368	8.507	.000*
External Risk	2.1062	.7437	2.43490	.9263	2.81333	1.0697	14.327	.000*
User Risk	2.3098	.7770	2.3557	.8892	3.0741	.9187	15.356	.000*
Complexity	2.4703	.8241	2.6326	.9875	3.233	.8893	12.166	.000*

(* indicates items significant at 5% significance level.)

The ANOVA showed highly significant differences between the onsite–offshore split up of the project. Error bars were analyzed for all the risk dimensions with reference to the onsite/offshore categories. Figure 5.2 shows the error bars for the team risk and other graphs also had similar shapes.

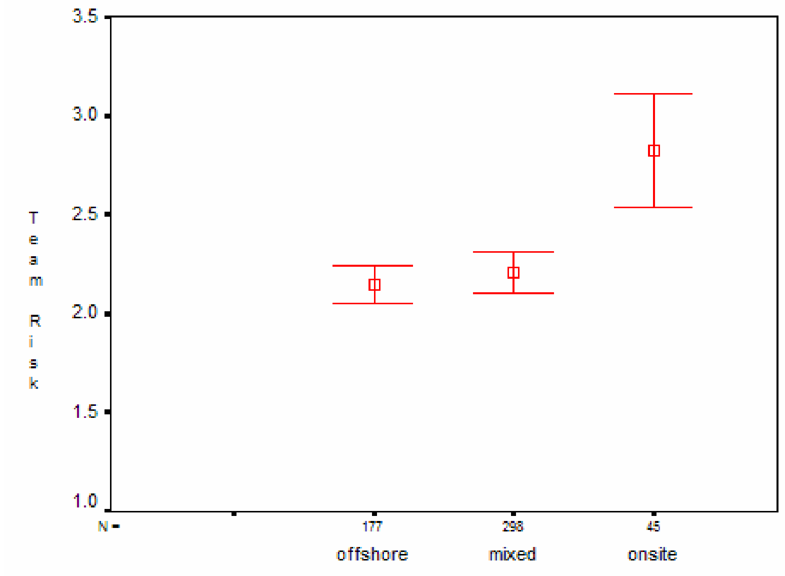


Figure 5.2 Error bar for Team risk versus onsite/offshore categories

It was seen that risk scores of onsite projects were significantly higher than their offshore counterparts. As the project became more offshore based, the risk involved came down. Hence the hypothesis H4b is supported.

Earl (1996) argued that proper outsourcing could help in reduction of much production related risk factors though there could be an increase in legal and contractual risk. India is projecting itself as an ideal destination for offshore software development. NASSCOM highlights Indian advantages in terms of highly skilled labour and quality focused working environment. The findings are in line with these arguments.

H4c: There is a significant difference in risk based on the duration of the project

Project duration was hypothesized to be affecting the risk. Table 5.8 shows the ANOVA results for the variation in risk dimension scores across categories defined based on the duration of the project.

Table 5.8 Risk dimensions and the duration of the project

Risk dimensions	Short		Medium		Long		F-value	P-value
	Mean	SD	Mean	SD	Mean	SD		
Team	2.4252	.9291	2.0792	.8793	2.2421	.8124	7.230	.001*
Plan& exec	2.8128	.9076	2.4653	.9435	2.5082	.8389	7.684	.001
External	2.5616	.9757	2.2378	.9063	2.2897	.8209	6.385	.002*
User	2.5791	.9361	2.2766	.8738	2.3632	.7629	5.802	.003*
Complexity	2.8422	.9029	2.5135	1.005	2.5032	.9496	6.843	.001*

(* indicates items significant at 5% significance level.)

The results indicated statistically significant variation in risk across categories defined based on project duration. Hence hypothesis H4c is supported

This was analyzed further with pair-wise t-tests and the results are summarized in Table 5.10. The analysis revealed that the short term projects had the highest risk score.

Table 5.9: t-tests results for risk and project duration categories

Risk Dimension	Short vs Medium		Medium vs Long		Short vs Long	
	t Statistic	P Value	t Statistic	P Value	t Statistic	P Value
Team Risk	3.674	0.000*	-0.348	0.728	2.563	0.011*
planning & execution	3.358	0.001*	-0.436	0.663	2.176	0.030*
External Risk	3.369	0.001*	0.077	0.939	2.767	0.006*
User Risk	3.737	0.000*	-1.403	0.162	1.525	0.128
Complexity Risk	3.264	0.001*	-0.762	0.447	1.813	0.071

(* indicates items significant at 5% significance level.)

There had been many studies looking at the impact of project duration on the successful running of the software development projects (Brooks, 1995; McFarlan, 1982, Chaos report 1998). Many of these researchers observed that the longer projects performed significantly worse than the shorter ones (Chaos report 1998). They recommended keeping the project size as small as possible or to decompose the project into smaller units.

But the findings of this study are not in line with the literature. The findings show that the risk scores in the short term projects are significantly higher than those in the medium term and long term projects. Detailed discussion with industry experts revealed that many of the short duration projects executed with Indian companies were not the normal small projects which can be properly completed in the short period. They were projects requiring more duration but forced to be completed within a shorter period because of competitive pressures or budget/resource constraints. Hence these projects demanded overtime use of resources and compromises on organization policies and project procedures. This in turn would lead to high risk scores.

H4d: There is a significant difference in risk based on the number of members in the project team

ANOVA results presented in Table 5.10 shows that there is no significant association seen between the risk scores and the project team size. The null hypothesis of no association is supported and the stated alternate hypothesis H4d can not be accepted.

Table 5.10 Risk dimensions and the team size

Risk dimension	Small		Medium		Large		F-value	P-value
	Mean	SD	Mean	SD	Mean	SD		
Team	2.246	.899	2.280	.8822	2.087	.7545	1.889	.152
Plan & exec	2.648	.951	2.613	.8940	2.505	.8732	0.776	0.461
External	2.292	1.00	2.427	.9104	2.229	.7378	2.186	.113
User	2.390	.925	2.449	.8909	2.285	.7744	1.314	.270
Complexity	2.563	.949	2.611	.9584	2.752	.9129	1.235	.292

Pressman (2001) highlighted the increased communication and coordination problems associated with increased team size. Chaos report (1998) found that as team size increases, the project failure rate also increases. Deephouse (2005) also checked for the impact of team size on project effectiveness but failed to make any generalization.

The present study failed to detect any statistically significant differences in risk scores based on team size. One of the reasons could be that 80% of the projects in the sample had teams of size less than 20. Hence the risk arising out of coordination and communication issues in big teams might not reflected here.

5.2.5 Testing hypothesis 5

H5: There is a significant variation in risk across categories of select organizational characteristics

Based on literature review, the following organizational characteristics were selected for checking the association with risk dimensions. (a) Size of the organization in terms of number of employees (Chaos, 1998) (b) Nature of the organization (Ropponen and Lyytinen, 1997) (c) Quality certifications possessed by the organization (Asundi, 2001).

Hence the hypothesis H5 was restated as below

H5a: There is a significant difference in risk based on the size of the organization

H5b: There is a significant difference in risk based on the nature of the organization

H5c: There is a significant difference in risk based on the quality certifications possessed by the organizations.

Each of these hypotheses was tested and analyzed separately.

H5a: There is a significant difference in risk based on the size of the organization

Table 5.11 gives the results of the ANOVA linking risk with size of the organization

Table 5.11:ANOVA test Results for Risk and Number of Employees

Risk dimensions	Small		Medium		Large		Very Large		F-Value	P-Value
	Mean	SD	Mean	SD	Mean	SD	Mean	SD		
Team	3.104	.8833	2.022	.6889	2.140	.8597	2.209	.7826	30.89	.000*
Plan& exec	3.34	.6573	2.451	.8482	2.508	.9349	2.518	.7963	19.47	.000*
External	3.138	.9459	2.034	.8362	2.332	.8635	2.361	.7822	26.46	.000*
User	3.243	.7419	2.217	.7801	2.319	.8612	2.356	.8469	26.45	.000*
Complexity	3.096	.8193	2.465	.8942	2.600	.9499	2.582	.9821	7.401	.000*

(* indicates items significant at 5% significance level.)

From the results it was clear that there were significant differences in risk across categories of organizations based on size supporting hypothesis H5a. To study this further, error bars were analyzed for all the risk dimensions. The error graph for team risk is shown under figure 5.3 and others had shown similar shapes.

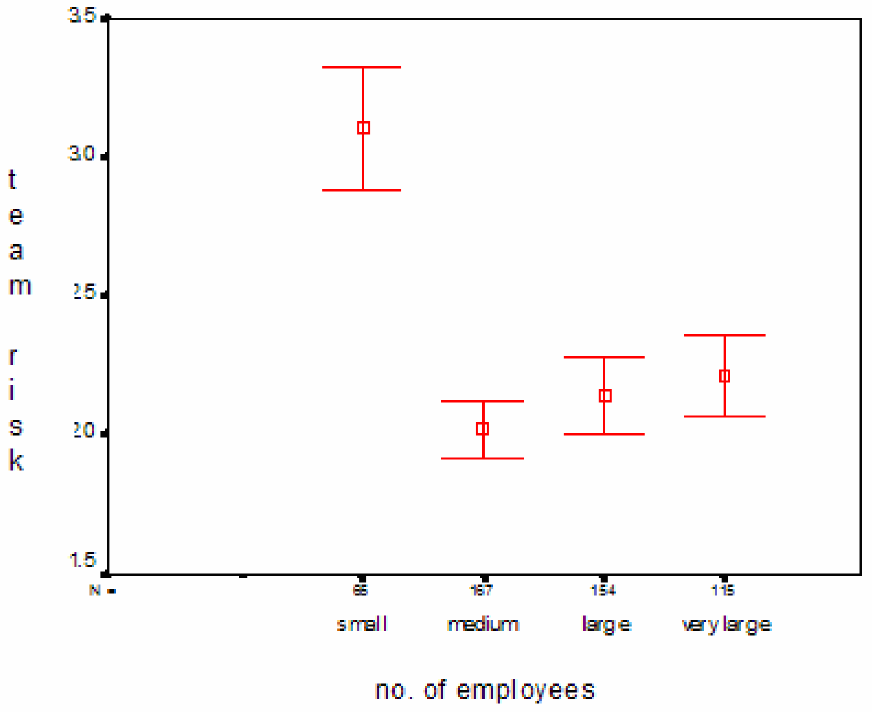


Figure 5.3 Team risk with number of employees.

This study showed that the projects from small sized companies reported statistically significant higher risk scores. But there was no significant difference seen among medium, large and very large companies. Small companies, which are small ventures without established systems and procedures account for the statistical difference. Otherwise the organization size is a non issue. This is generally in line with the Chaos report (1998) finding that there is no significant correlation between the company size and project success.

H5b: There is a significant difference in risk based on the nature of the organization

The nature of the organization was defined as one of the following: Indian company with domestic operations, Indian company with international operations, Indian arm of a foreign MNC. The table 5.12 below shows the ANOVA results for hypothesis linking risk dimensions to nature of the project organization.

Table 5.12 Risk dimensions and the nature of the Company

Risk dimensions	Indian domestic		Indian International		Unit of MNC		F-Value	P-Valu
	Mean	SD	Mean	SD	Mean	SD		
Team Risk	3.402	.55	1.926	.7216	2.470	.818	33.39	.000*
Plan & exec	3.496	.43	2.295	.9094	2.844	.757	23.72	.000*
External Risk	3.447	.68	2.153	.7527	2.430	.968	18.56	.000*
User Risk	3.412	.52	2.115	.8045	2.637	.830	25.41	.000*
Complexity	3.333	.58	2.472	.9425	2.730	.955	7.033	.000*

(* indicates items significant at 5% significance level.)

As seen all the F values were highly significant indicating a significant association between risk scores and the nature of the company. Thus the hypothesis H5b is supported.

The projects from Indian companies with international projects showed the lowest risk scores. The statistical significance of these findings was supported with pair-wise t-tests (Table 5.13). Indian companies with domestic projects reported the highest risk scores.

Table 5.13 t-tests results for Risk and nature of the company

Risk Dimensions	Indian domestic vs Indian international		Indian international vs MNC		Indian domestic vs MNC	
	T Statistic	P Value	t Statistic	P Value	t Statistic	P Value
Team Risk	10.88	0.000*	-6.74	0.000*	5.87	0.000*
Planning & execution	8.33	0.000*	-3.43	0.001*	6.16	0.000*
External Risk	6.13	0.000*	-2.85	0.005*	4.11	0.000*
User Risk	11.53	0.000*	-7.47	0.000*	6.91	0.000*
Complexity Risk	10.46	0.000*	-6.70	0.000*	5.97	0.000*

(* indicates items significant at 5% significance level.)

These results go well with the findings on the association of risk with the size of the organization. It was seen that small sized companies had the highest risk scores. Most of the Indian companies with only domestic projects are small sized companies. These findings are in line with McFarlan (1981) arguments that companies with high structure and well defined procedures are likely to experience lower risk.

H5c: There is a significant difference in risk based on the quality certifications possessed by the organizations.

The following three major quality certifications were considered – ISO9001, CMM and PCMM.

ISO 9001 is the creation of the International Organization for Standardization (ISO), a Swiss-based federation of national standards bodies. ISO 9001 is part of the ISO 9000 family of standards. ISO 9001 targets the manufacturing process, although it also includes manufacturing services and software development. The Capability Maturity Model for Software (CMM) is a model used by many organizations to identify best practices useful in helping them increase the maturity of their processes. It is a process improvement approach that is based on a process model. A process model is a structured collection of practices that describe the characteristics of effective processes; the practices included are those proven by experience to be effective. People Capability Maturity Model (PCMM) is the roadmap for implementing best work practices in an organization through different stages. Each progressive stage produces a transformation in the organization by equipping it with better practices for attracting, developing, organizing, motivating and retaining its workforce. The primary aim of PCMM is to improve the capability of the workforce.

To check the hypothesis H5c, the following pairs are compared.

1. Quality certified (at least one of the above certification) companies and noncertified companies
2. CMM certified companies and CMM noncertified companies
3. ISO certified companies and ISO noncertified companies
4. PCMM certified companies and PCMM noncertified companies

The t test results are summarized in table 5.14

Table 5.14 t test results for risk and quality certification

Risk Dimensions	Certified vs non certified		CMM vs non CMM		ISO vs non ISO		PCMM vs non PCMM	
	t-stat	P Value	t- Stat	P Value	t- Stat	P Value	t- Stat	P Value
Team risk	.248	.804	2.964	0.86	1.379	0.169	.781	.377
Plan & exec	.250	0.803	1.846	.305	-0.524	0.600	.306	.581
External risk	-1.461	0.145	1.056	0.175	0.911	0.363	1.054	.305
User risk	.485	0.628	1.208	.272	-0.151	0.880	1.920	.166
Complexity	-.767	0.443	.613	.434	-0.178	0.859	2.239	.135

The results show that there is no significant association between the certification status of the organization and the risk scores. The hypothesis H5c is not supported.

Competitive pressures have induced many Indian software firms to apply for and receive quality certifications such as the ISO-9001 and CMM. Although such quality certifications have become increasingly popular in the USA and Europe, there is very little evidence on the impact of quality certification on risk scores. This finding is supported the study by Asundi (2001) who found no relationship between the quality certification status and the organizational performance of the Indian software companies.

5.3 CONCLUSION

Correlation analysis showed a strong negative correlation between risk dimensions and risk management dimensions. The results showed a negative correlation between risk and quality of software developed and a positive correlation between risk and overrun variables. Similarly risk management were positively correlated with quality and negatively correlated with time overrun as well as cost overrun.

The study showed that the engineering software had the minimum risk score compared to other domains. Project duration was seen to be influencing the risk scores. Short term projects had the highest risk scores. Projects from small-sized organizations reported higher risk scores. The quality certification status of the organization was not seen to be influencing the risk scores.

Thus it can be concluded that there is a relationship among risk, risk management and project outcome. There are some organizational and project characteristics which will have an impact on these relationships. These findings lay the foundation for the work in the subsequent chapters which focus on defining the nature of these relationships.

****

CHAPTER 6

RELATIONSHIP OF RISK WITH RISK MANAGEMENT, PROJECT CHARACTERISTICS AND ORGANIZATIONAL CHARACTERISTICS

Section 6.1 sets the background for the work in this chapter. Section 6.2 describes the proposed regression analysis. Section 6.3 describes the regression model for each of the five risk dimensions. Finally section 6.4 gives the summary and conclusion of the chapter.

6.1 INTRODUCTION

The view that the level of risk in a software development project is influenced by risk management is endorsed by many researchers (Charrette 1996). But most of these studies are theoretical in nature. Much empirical work has been done on identification of risk factors in software projects. However less attention has been paid to the process of managing risks and understanding the impact of the risk management practices on the risk factors that are to be controlled. The nature of the impact of project and organizational characteristics on the risk also needs more analysis.

Some of the major studies linking risk to risk management and project and organizational characteristics were reviewed in section 2.4 of chapter 2. They include Ropponen and Lyytinen's (1997) empirical research on how risk management practices and environmental contingencies help to address the risk components in software projects, Jiang et al's (2000) study on the relationship between the major risk factors and the risk mitigation strategies and Addision and Vallabh's (2002) analysis linking risk controls and risk. But these empirical studies were not based on validated measures of risk and risk management.

The hypotheses tests in chapter 5 showed that that there was a significant negative correlation between the risk and risk management. It was also seen that many organizational and project attributes had significant association with risk. This section tries to answer, through a series of multiple regression analysis, how these relationships are defined.

6.2 PROPOSED REGRESSION ANALYSIS

Multiple regression analysis is done to study the relationship of a single dependent variable with several independent variables and develop the model which evolves maximum contribution to the variable tested. The multiple regression equation takes the form y = b1x1 + b2x2 + ... + bnxn + c. The b's are the regression coefficients, representing the amount the dependent variable y changes when the corresponding independent variable x changes by 1 unit. The c is the constant, where the regression line intercepts the y axis, representing the amount the dependent y will be when all the independent variables are 0. The standardized versions of the b coefficients are the beta weights, and the ratio of the beta coefficients is the ratio of the relative predictive power of the independent variables.

Dummy variables are a way of introducing nominal or ordinal variables to a regression equation. The standard approach to modeling categorical variables is to include the categorical variables in the regression equation by converting each level of the categorical variable into a variable of its own, usually coded 0 or 1. One of the levels is kept out of the regression model. The omitted category is the reference category because b coefficients must be interpreted with reference to it. The interpretation of b coefficients is different when dummy variables are present. Normally, without dummy variables, the b coefficient is the amount the dependent variable increases when the independent variable associated with the b coefficient increases by one unit. When using a dummy variable, the b coefficient shows how much more the dependent variable increases (or decreases if b is negative) when the dummy variable increases one unit (thus shifting from 0=not present to 1=present) compared to the reference category.

R^2, the coefficient of multiple determination, is the percentage of the variance in the dependent explained uniquely or jointly by the independents. The F test is used to test the significance of R, which is the same as testing the significance of R^2, and this is equivalent to testing the significance of the regression model as a whole. T-tests are used to assess the significance of individual b coefficients.

Multiple regressions have few important assumptions: linearity of relationships, the same level of relationship throughout the range of the independent variable ("homoscedasticity"), interval or near-interval data, absence of outliers, normality of the dependent variable, absence of autocorrelation, absence of high multicollinearity among independent variables etc. The output of the regression analysis is reliable only to the extent of these assumptions being met.

This research used a series of multiple regression analysis to identify the predictors of the risk factors. Stepwise multiple regression method was used to identify the best fitting models. Stepwise regression involves (1) identifying an initial model, (2) iteratively "stepping," i.e. repeatedly altering the model at the previous step by adding or removing a predictor variable in accordance with the "stepping criteria," and (3) terminating the search when stepping is no longer possible given the stepping criteria, or when a specified maximum number of steps has been reached.

Regression analysis was performed with each risk dimension as the dependent variable and risk management dimensions and the select project and organizations characteristics as the independent variables. Since the characteristic variables were category variables, they were incorporated into the regression analysis as dummy variables. The dummy variables used and the reference categories are shown below.

Variable	**Reference category**
Type of software developed	Business Application
Project duration	Short
Onsite / offshore split up	Onsite
Size of the organization	Small
Status of the organization	Local Domestic

6.2.1 Validating the basic assumptions in regression analysis

Some of the basic assumptions of regression analysis were checked and verified as part of the regression analysis. They are presented before discussing the regression models.

<u>Multivariate Normality and Normally distributed residual error:</u> All the dependent variables in the regression analysis are expected to be normally distributed. Small variations are not considered to be serious. This was done through plotting

standardized residuals and checking for normality visually. Sample plot for team risk residual error plot obtained from the regression analysis is shown in figure 6.1. As seen, it is roughly normal.

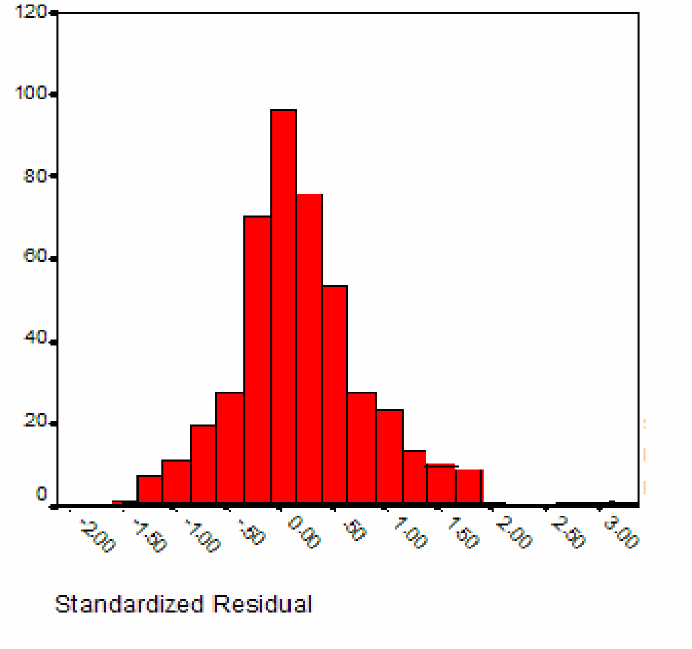


Figure 6.1 Standardized residual plot for team risk

Lack of multicollinearity: Multicollinearity is the inter-correlation of independent variables. To assess multivariate multicollinearity, one uses tolerance or VIF (Variance Inflation Factor), which builds in the regressing of each independent variable on all the others. VIF is the reciprocal of tolerance. VIF > 5 (or > 10 according to some researchers) is an arbitrary but common cut-off criterion for multicollinearity. The tolerance and VIF values are reported with all regression output tables in the next section. All the tolerance and VIF values were seen to be in the acceptable range.

Independent observations (absence of autocorrelation) leading to uncorrelated error terms. Current values should not be correlated with previous values in a data series. This was checked with The Durbin-Watson coefficient (d). The value of d ranges from 0 to 4. As a rule of thumb, d should be between 1.5 and 2.5 to indicate independence of observations. The Durbin-Watson coefficient values are reported with each analysis. As seen, all values fell in the acceptable range of 1.5 to 2.5 indicating independence of observations.

Linearity: Regression analysis is a linear procedure. To the extent nonlinear relationships are present, conventional regression analysis will underestimate the relationship. Linearity was tested through the x-y scatter graph of risk – risk management dimensions.

<u>Homoscedasticity:</u> This means that the variance of residual error should be constant for all values of the independent variable(s). This was checked with a residual plot with residuals on the Y axis against predicted values on the X axis. A sample plot for team risk is shown in figure 6.2

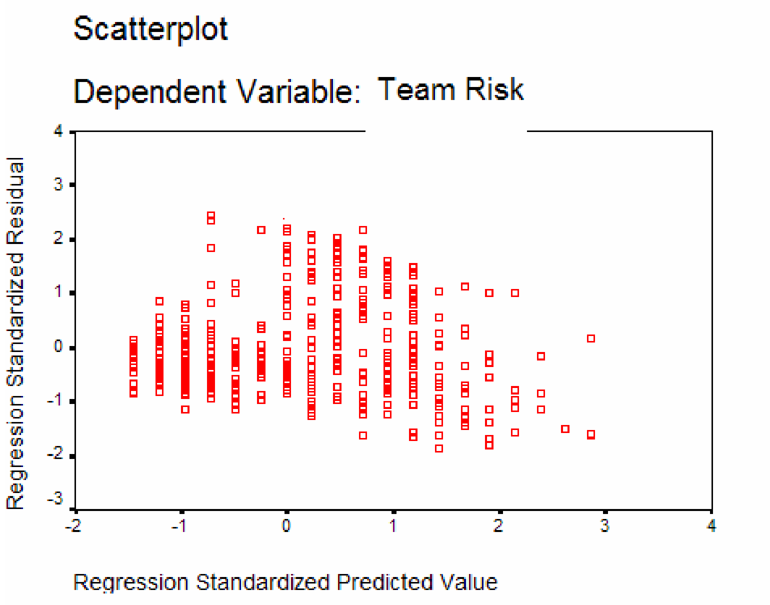


Figure 6.2: Plot with standardized residuals along Y axis and predicted values along X axis

6.3 REGRESSION MODELS FOR THE RISK FACTORS

Regression analysis results were generated for each risk dimension. Only the final model of the stepwise regression is reported. The significant variables in the final model, their coefficients (both standardized and unstandardized), significance values and collinearity statistics are shown in tables. R^2 value, adjusted R^2 value, the significance of the overall model and the Durbin Watson statistic are also reported.

6.3.1 Predictors of Team risk

Table 6.1 shows the output of the regression analysis with Team risk as dependent factor and the risk management dimensions and select project and organization characteristics as independent variables.

Table 6.1 Regression output for team risk

MODEL	Unstandardized Coefficients		Std Coeff.	T stat	Sig.	Collinearity Statistics	
	B	S.E.	Beta			Tolerance	VIF
(Constant)	4.900	.233		20.995	.000		
Planning	-.383	.058	-.293	-6.630	.000	.781	1.281
Indian MNC	-.677	.133	-.376	-5.088	.000	.279	3.579
Foreign MNC	-.318	.136	-.169	-2.347	.019	.293	3.409
Medium size	-.802	.123	-.421	-6.538	.000	.367	2.724
Large Size	-.682	.124	-.353	-5.481	.000	.367	2.724
Very large size	-.625	.133	-.289	-4.698	.000	.403	2.484

'F' ratio 40.832 Sig. = .000 $R^2 = 0.373$ Adjusted $R^2 = 0.364$
Durbin Watson = 1.83

The regression model was able to explain 36.4% of variation in Team risk. Among the risk management strategies, project planning was seen to have maximum negative association with team risk. One unit increase in project planning would decrease team risk by 0.383 units. Size and status of the organization were the other two variables in the final model. They are category variables which are put into the model as dummy variables and are to be understood differently. Being an Indian MNC would decrease the risk by 0.677 compared to being a local domestic company (reference category). Similarly, being a foreign MNC would decrease the risk by 0.318 compared to being a local domestic company. Also, being medium sized would result in risk reduction by 0.802 units, large sized by 0.682 units and very large sized by 0.625 units all compared to small sized.

The regression equation can be written as

$$\begin{aligned} Team\ risk = {} & 4.90 - 0.383 * project\ planning - 0.677 * Indian\, MNC - \\ & 0.318 * foreign\, MNC - 0.802 * medium\ sized - 0.682 * \\ & large\ sized - 0.625 * very\ large\ sized \end{aligned}$$

6.3.2 Predictors of Project planning and execution risk

Table 6.2 shows the output of the regression analysis with Project planning and execution risk as the dependent factor

Table 6.2 Regression output for planning and execution risk

MODEL	Unstandardized Coefficients		Std Coeff.	T stat	Sig.	Collinearity Statistics	
	B	S.E.	Beta			Tolerance	VIF
(Constant)	5.376	.252		21.356	.000		
Project	-.544	.060	-.408	-9.007	.000	.831	1.204
Indian MNC	-.656	.136	-.357	-4.818	.000	.310	3.225
Foreign MNC	-.410	.136	-.214	-3.025	.003	.342	2.927
Medium size	-.192	.081	-.099	-2.357	.019	.970	1.031

‘ F’ ratio 43.406 Sig. = .000 R^2 = 0.296 Adjusted R^2 =0.289 Durbin Watson = 1.725

The model captured 28.9% of the variation in project planning and execution risk. Project planning and execution risk was best mitigated by project planning strategies. Status of the company and size were also seen to have significant impact. Being an Indian / foreign MNC would help to reduce project planning and execution risk compared to local domestic companies. Effect of being a medium sized company was also captured in the final model.

The regression equation can be written as

$$\begin{aligned} \text{Project planning}\,\&\,\text{execution risk} = 5.376 - 0.544 * \textit{project planning} & - \\ 0.656 * \textit{Indian MNC} & \\ - 0.410 * \textit{foreign MNC} & \\ - 0.192 * \textit{medium sized} & \end{aligned}$$

6.3.3 Predictors of External risk

Here the independent variables were regressed with the dependent variable –External risk. The results are shown in table 6.3

Table 6.3 Regression output for External risk

MODEL	Unstandardized Coefficients		Std Coeff.	T stat	Sig.	Collinearity Statistics	
	B	S.E.	Beta			Tolerance	VIF
(Constant)	3.541	.284		12.480	.000		
Indian MNC	-.779	.143	-.418	-5.439	.000	.295	3.389
Foreign MNC	-.576	.143	-.296	-4.017	.000	.321	3.112
Medium size	-.324	.087	-.164	-3.706	.000	.888	1.126
Planning	-.559	.095	-.414	-5.868	.000	.351	2.849
HR strategies	.263	.093	.204	2.838	.005	.338	2.956
User coordn	.198	.067	.172	2.939	.003	.509	1.966
Mixed mode	.303	.089	.161	3.398	.001	.774	1.292
Onsite mode	.329	.153	.100	2.150	.032	.814	1.229
Medium Dur	-.279	.089	-.148	-3.131	.002	.783	1.277
Long duration	-.259	.111	-.108	-2.335	.020	.818	1.223
Engg software	-.235	.102	-.100	-2.293	.022	.914	1.094

F' ratio 15.166 Sig. = .000 $R^2 = 0.291$ Adjusted $R^2 = 0.272$
Durbin Watson = 1.666

The model explained 27.2% of the variation in the dependent variable.

External risk is the risk induced by external factors such as telecommunication network, visa regulators, subcontractors etc. External risk could be reduced by proper project planning. HR and user strategies were seen to be counterproductive in mitigating external risk. This means that too much of internal and user focus may come at the expense of controlling external risk. Also MNCs reported less external risk compared to local domestic companies. Engineering projects and long duration projects had less external risk compared to business projects and short duration projects respectively. External risk increased as the project moved from offshore to onsite.

The regression equation can be written as

$$\begin{aligned}External\,risk\; &= 3.541 - 0.559 * project\ planning + 0.263 * HR\,Management + \\ &0.198 * user\,coordination - 0.779 * Indian\,MNC - \\ &0.576 * foreign\,MNC - 0.324 * medium\,sized - \\ &0.279 * medium\,duration - 0.259 * long\ duration - \\ &0.235 * engineering\ project + 0.303 * mixed + 0.329 * onsite\end{aligned}$$

6.3.4 Predictors of User risk

Table 6.4 presents the regression results with user risk as the dependent variable.

Table 6.4 Regression results for User risk

MODEL	Unstandardized Coefficients		Std Coeff.	T stat	Sig.	Collinearity Statistics	
	B	S.E.	Beta			Tolerance	VIF
(Constant)	5.278	.230		22.925	.000		
Project planning	-.366	.085	-.281	-4.314	.000	.350	2.855
User Coordination	-.225	.086	-.170	-2.616	.009	.352	2.841
Indian MNC	-.252	.077	-.140	-3.274	.001	.806	1.240
Onsite mode	-.484	.136	.152	3.551	.000	.810	1.235
Mixed mode	-.192	.078	.106	2.460	.014	.799	1.251
Medium size	-.651	.117	-.343	-5.570	.000	.391	2.555
Large size	-.515	.120	-.268	-4.313	.000	.385	2.599
Very large size	-.463	.125	-.215	-3.695	.000	.438	2.282

F' ratio 33.117 Sig. = .000 R^2 = 0.393 Adjusted R^2 =0.381
Durbin Watson = 2.310

Regression results for user related risk showed significant F value and the model was able to explain 38.1% of the variation in the dependent variable. Project planning and the user coordination strategies were seen to be the most effective risk mitigation strategies for user risk. Larger sized companies had significantly lower user risk. User risk decreased as the work moved onsite which is understandable as the work happens closer to the user.

The regression equation can be written as

$$User\ risk = 5.278 - 0.366 * project\ planning - 0.225 * user\ coordination$$
$$- 0.252 * Indian\ MNC - 0.651 * medium\ sized - 0.515 * \text{large}\ sized$$
$$- 0.463 * very\ \text{large}\ sized - 0.192 * mixed - 0.484 * onsite$$

6.3.5 Predictors of Project complexity risk

The results are shown in table 6.5

Table 6.5 Regression results for Project Complexity risk

MODEL	Unstandardized Coefficients		Std Coeff.	T stat	Sig.	Collinearity Statistics	
	B	S.E.	Beta			Tolerance	VIF
(Constant)	4.309	.266		16.205	.000		
Project planning	-.448	.065	-.317	-6.845	.000	.939	1.065
Onsite mode	.767	.167	.223	4.603	.000	.864	1.158
Mixed mode	.282	.098	.144	2.893	.004	.817	1.224
Medium size	-.270	.094	-.131	-2.883	.004	.977	1.024

F' ratio 20.572 Sig. = .000 R^2 = 0.166 Adjusted R^2 =0.158
Durbin Watson = 2.136

Regression results for project complexity risk showed a significant model with a low R^2 value. The model was capable of explaining only 15.8% of the variation in the dependent variable. Planning strategies were found to be most effective. Onsite / offshore methodology and size were the significant dummy variables. It was seen that complexity risk increased as the work moved onsite.

The equation is

$$project\ complexity\ risk = 4.309 - 0.448 * project\ planning -$$
$$0.270 * medium\ sized\ + 0.282 * mixed -$$
$$0.767 * onsite$$

6.4 CONCLUSION

Regression models were derived for each of the risk dimensions linking it to risk management dimensions and select project and organizational characteristics. Though there were many similarities among these models, the exact nature of the relationships differed. All the Durbin Watson statistics and VIF statistics were seen to be in the acceptable zone.

Project planning was seen to be the most effective strategy in mitigating various risk dimensions. It appeared in all the regression models. User coordination was found to have significant correlation with user risk. The importance of project planning as a risk mitigation strategy has been highlighted in the literature. Barki et. al.,(2001) showed that formal planning helps to improve project performance in high risk projects. Deephouse et.al., (2005) also found that software project planning is the most important determinant of improving project performance. The influence of user coordination strategies on user risk is self explanatory as these are the measures designed to reduce the user related risk items.

All the regression models were statistically significant even at 1% significance level. But the percentage of variance in the risk component explained varied from 38.1% for user risk to 15.8% for project complexity risk. Thus the explanatory powers of these models can be termed only as moderate.

****

CHAPTER 7

RELATIONSHIP OF PROJECT OUTCOME WITH RISK

The previous chapters showed that there is a relationship among risk, risk management and project outcome. This chapter and chapter 8 try to model these relationships. Section 7.1 introduces the various models to be checked. Section 7.2 presents the results of the regression analysis linking risk dimensions to project outcome. Section 7.3 presents the conclusion of the chapter.

7.1 INTRODUCTION TO THE RISK - RISK MANAGEMENT - OUTCOME MODELS

A model can be described as a theoretical construct representing underlying processes through a set of variables and a set of logical and quantitative relations between them. They are simplified frameworks designed to illuminate complex processes. A major objective of this research is to explore the interrelationship among project outcome, risk and risk management. This section, based on the literature support, puts forward two possible predictor models for project outcome measures.

7.1.1 Model 1: Risk – Project outcome model

The model 1, shown in figure 7.1, focuses on predicting the project outcome variables in terms of the risk dimensions. It does not include risk management dimensions as predictor variables.

This model is derived from the study of Wallace (1999). She had developed an empirical model explaining project outcome measures in terms of risk dimensions. This model found support from many researchers (e.g., Fairley, 1994; Wallace et. al, 2004, Jiang et. al., 2000).

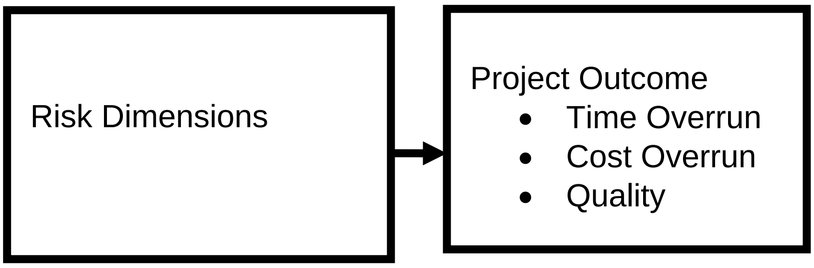


Fig 7.1: Model 1

7.1.2 Model 2: Risk - Risk Management – Project Outcome model

Many researchers do not subscribe to the view that risk is the only predictor of project outcome. There are arguments in favor of including risk management into the model (Wallace, 1999). Many researchers look at risk management as a continuous process where additional information and risk status are utilized to refine the risk list and the risk management plans (Charrette, 1996). The Barki et. al., (2001) model demonstrated how the project outcome is influenced by the fit between risk and risk management. Wohlin's (2002) research also linked a set of risk variables and risk management variables to project success.

Nidumolu's(1995) pioneering study proposed a different approach. His model was an integrative model with select risk management dimensions affecting the project outcome directly as well as indirectly through an intervening variable called residual performance risk. This mixed model has found acceptance among many researchers like Kwan-Sik Na who replicated this study in Korea and developed variants of the mixed model (Na et al, 2006).

These arguments are incorporated into model 2 shown in figure 7.2. This integrative model has the risk management dimensions hypothesized to affect the project outcome directly and also indirectly through risk dimensions.

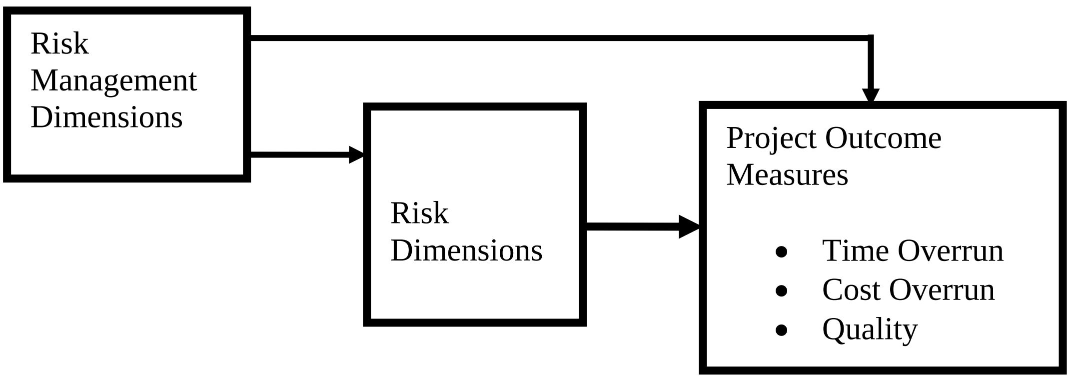


Fig 7.2: Model 2

Model 1 was tested through multiple regression analysis. Each of the project outcome variables was selected as the dependent variable in the regression models. The risk components were taken as independent variables. Structural equation modeling was used to test Model 2.

Important assumptions on the dependent variable in the regression analysis were mentioned in the previous chapter. They include linearity of relationships, the same level of relationship throughout the range of the independent variable ("homoscedasticity"), interval or near-interval data, absence of outliers, normality assumptions, absence of autocorrelation, absence of high multi-collinearity among independent variables etc. Some of the basic assumptions of regression analysis were checked and reported for the dependent variables namely time overrun, cost overrun and quality.

7.2 MODEL 1: LINKING PROJECT OUTCOME TO RISK

A series of regression analyses were performed with each project outcome variable as dependent variable and the five risk dimensions as the independent variables. Step-wise regression method was used to get the best fitting models. The final models with their associated statistics are reported.

7.2.1 Linking quality to risk dimensions

Table 7.1 shows the output of the regression model linking quality of the software (dependent variable) to risk dimensions (independent variables).

Table 7.1 Output of the regression model linking quality to risk dimensions

MODEL	Unstandardized Coefficients		Std Coeff.	T stat	Sig.	Collinearity Statistics	
	B	S.E.	Beta			Tolerance	VIF
(Constant)	4.881	.064		76.026	.000		
Team risk	-.275	.046	-.445	-5.926	.000	.258	3.874
User risk	-.115	.037	-.188	-3.124	.002	.400	2.500
External risk	0.0991	.035	.169	2.850	.005	.416	2.403

'F' ratio 54.82 Sig. = .000 R^2 = 0.239 Adjusted R^2 = 0.235
Durbin Watson = 1.962

The model explained 23.5% variation in the dependent variable (quality) mainly through the independent variables team risk and user risk. They carried negative coefficients in the model. The VIF value and Durbin Watson were acceptable. The residual error plot (figure 7.4) had a normal shape. The scatter diagram (figure 7.5) between the residual and predicted values showed no particular trend indicating no major heteroscedasticity.

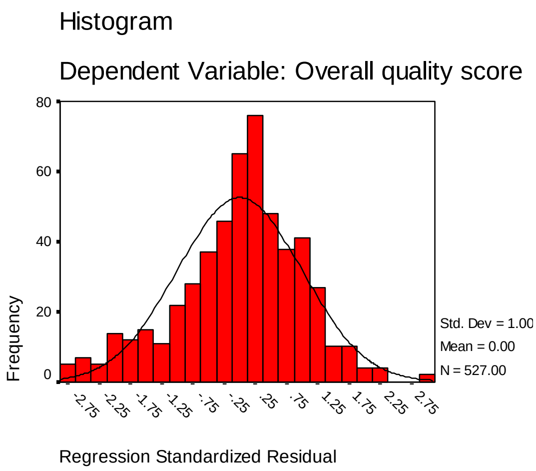


Figure 7.3 Residual error plots for Model 1-Quality

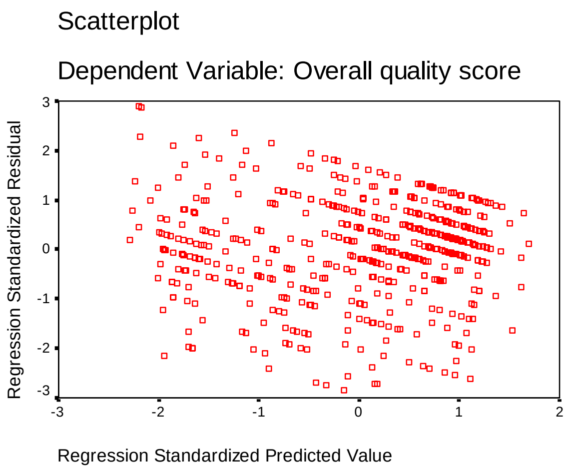


Figure 7.4 Scatter diagram for Model 1-Quality

7.2.2 Linking Time overrun to risk dimensions

Table 7.2 shows the output of the regression model linking time overrun (dependent variable) to risk dimensions (independent variables).

Table 7.2 Output of the regression model linking time overrun to risk dimensions

MODEL	Unstd Coefficients		Std Coeff.	T stat	Sig.	Collinearity Statistics	
	B	S.E.	Beta			Tolerance	VIF
(Constant)	-22.7	1.886		12.05	.00		
Team risk	6.75	1.215	.401	5.564	.00	.294	3.396
Planning & execution	2.34	1.028	.143	2.281	.02	.391	2.557
User risk	2.37	1.153	.142	2.061	.04	.324	3.091

‘ F’ ratio 88.88 Sig. = .000 R^2 = 0.409 Adjusted R^2 = 0.404
Durbin Watson = 2.153

The final model had three risk dimensions namely team risk, project planning and execution risk and user risk. The model fitted well with 40.4% of the variation in time overrun explained by the model. The VIF and Durbin Watson statistics were in the normal range. The standardized residual plot in figure 7.6 and the scatter diagram in figure 7.7 showed acceptable shapes.

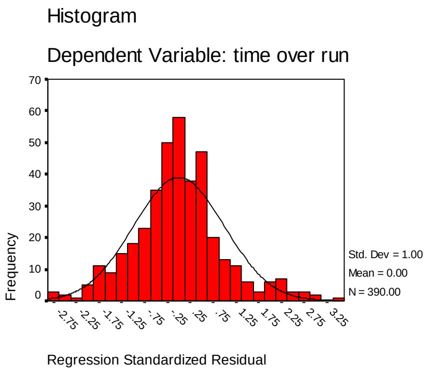


Figure 7.5 Residual error plots for Model 1-Time overrun

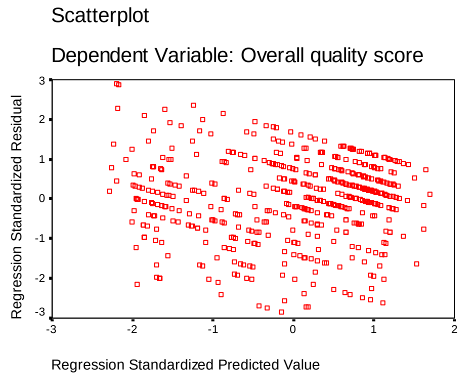


Figure 7.6 Scatter diagram for Model 1-Time overrun

7.2.3 Linking Cost overrun to risk dimensions

The results of the regression analysis linking cost overrun (dependent variable) to risk dimensions (independent variables) are shown in table 7.3.

Table 7.3 Results of the regression analysis linking cost overrun to risk dimensions

MODEL	Unstd Coefficients		Std Coeff.	T stat	Sig.	Collinearity Statistics	
	B	S.E.	Beta			Tolerance	VIF
(Constant)	-16.1	2.165		-7.4	.00		
Team risk	5.12	1.082	.332	4.736	.00	.450	2.220
Project planning and	3.95	1.104	.251	3.583	.00	.450	2.220

'F' ratio 67.06 Sig. = .000 $R^2 = 0.297$ Adjusted R^2 = 0.293 Durbin Watson = 1.976

The model was statistically significant, and explained 29.3% of the variation in cost overrun. The cost overrun had a direct relationship with team risk and project planning and execution risk. The VIF and DW statistics are acceptable and the residual and the scatter plots are produced in figure 7.8 and 7.9 respectively.

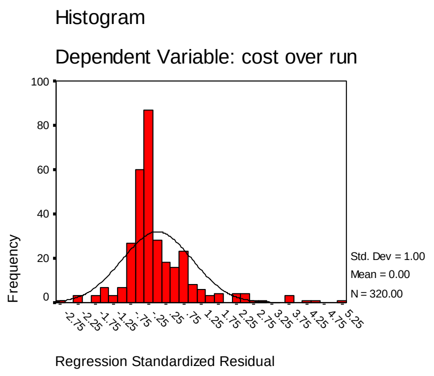


Figure 7.7 Residual error plots for Model 1-Cost overrun

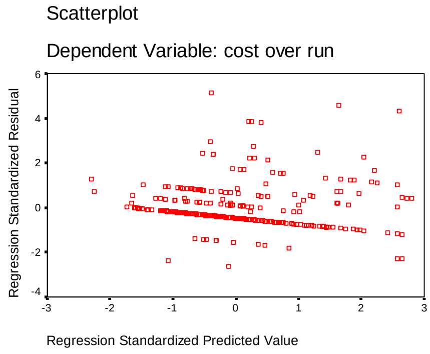


Figure 7.8 Scatter diagram for Model 1-Cost overrun

7.3 CONCLUSION

The focus of this chapter was to develop a basic model linking risk to project outcome. Risk had five dimensions and project outcome had three measures namely quality, time overrun and cost overrun. Each of these outcome variables was separately linked to the risk dimensions. All the models were statistically significant. The basic assumptions of regression analysis such as normality of residual plot, homoscadacity and absence of high multi-collinearity were seen to be satisfied for all models.

Team risk and planning and execution risks were seen to be the major risk factors influencing the outcome measures. Team risk was seen to be adversely affecting all the three dimensions of project outcome. Planning and execution risk figured in the final model for both time and cost overruns. User risk had influence on time overrun only. External risk figured in the quality model but with an inconsistent loading sign. The models were limited in their power to explain variation in project outcome.

This indicates that there is scope for improving the basic risk-project outcome model. Hence model 2 linking risk and risk management to project outcome was taken up for testing.

****

CHAPTER 8

INTEGRATED MODELS LINKING RISK, RISK MANAGEMENT AND PROJECT OUTCOMES

This chapter tests the first order and second order variants of the proposed model linking risk and risk management to project outcomes. Structural Equation Modeling (SEM) technique is used to compare and identify the best fitting model. Section 8.1 introduces the concept of first order and second order factor models. Structural Equation Modeling concepts are explained in section 8.2. Risk and risk management as second order constructs are tested in section 8.3 and 8.4 respectively. Section 8.5 introduces the first order and second order models of project outcome measures. The integrated models linking risk, risk management to each of the project outcomes namely quality, time overrun and cost overrun are presented in section 8.6, 8.7 and 8.8 respectively. Section 8.9 gives the conclusion of the chapter.

8.1 INTRODUCTION TO SECOND ORDER FACTOR MODELS

The focus of chapter 7 was to develop basic models where each project outcome variable was regressed with the five dimensions of software project risk. These models could be called the first order models where the factors of a construct (risk) were permitted to correlate with the dependent variable directly. Though all these models were statistically valid and hence acceptable, they suffered from the problems of relatively low R^2 values, many of the risk dimensions not figuring in the final model and having loading signs inconsistent with the hypothesized signs.

When the first order model fails to provide an acceptable solution, a second-order model can be used to put a structure on the correlations between the first order factors (Rindskopf and Rose, 1988). The second-order factor model implies that there is another latent construct which governs the correlations among the first order factors. Thus the first order factors are not allowed to correlate, but rather their co-variation is explained by the 2^{nd} order construct. The second order model is more parsimonious or simple than the first order model.

Risk as a first order model was tested in section 4.3.1 of chapter 4. The model is reproduced in figure 8.1. The five dimensions of software project risk were proposed to be associated and were allowed to co-vary. Each factor was connected to every other factor by a two-headed arrow, which meant that every factor was allowed to co-vary with every other factor. However, none of the indicators were allowed to co-vary, which is the standard practice with structural equation modeling (Hatcher, 1994).

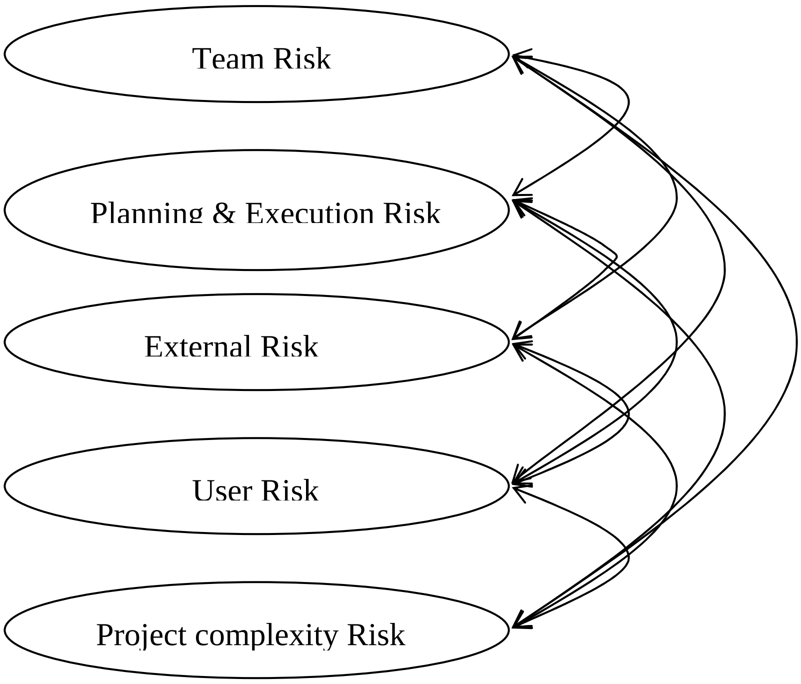


Figure 8.1: First order model of Project Risk

In the second order model, the five dimensions form into another latent construct, called project risk. It hypothesizes project risk as a second-order factor that explains the relationship among the five dimensions of risk. This is represented in figure 8.2.

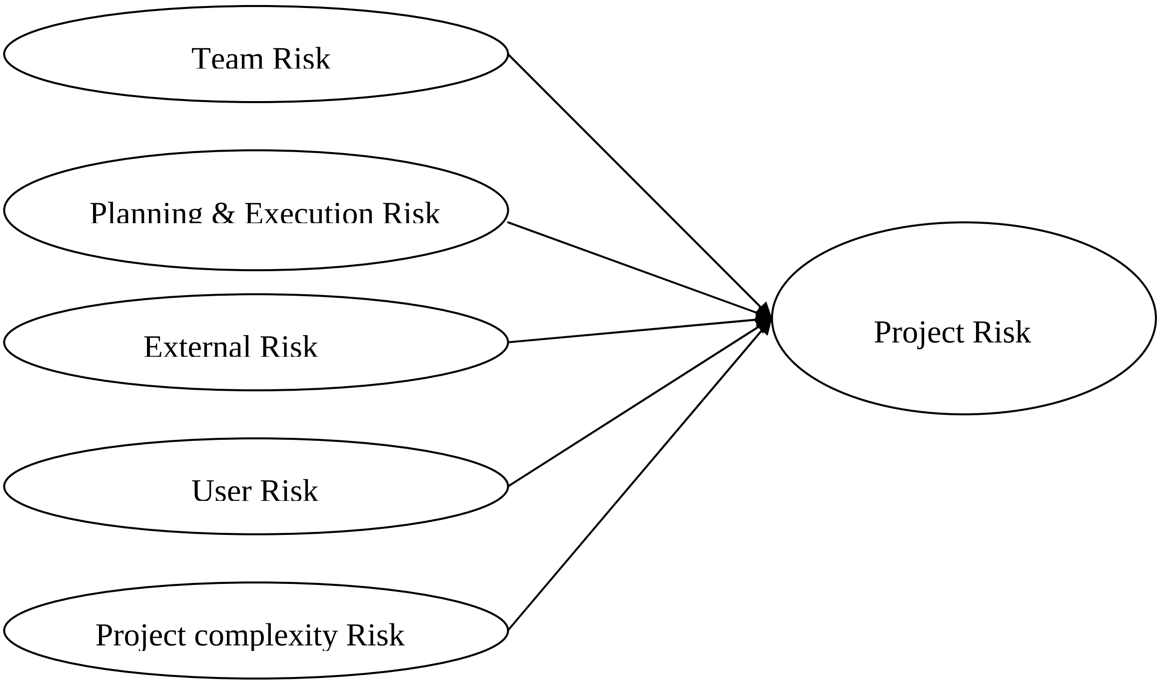


Figure 8.2: Second order model of Project Risk

Similarly the first order and the second order models for project risk management are shown in figure 8.3 and figure 8.4 respectively.

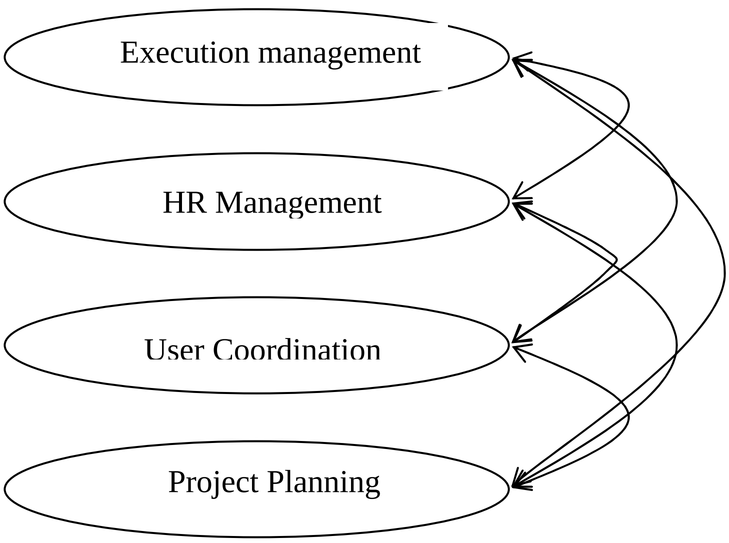


Figure 8.3: First order model for Project Risk Management

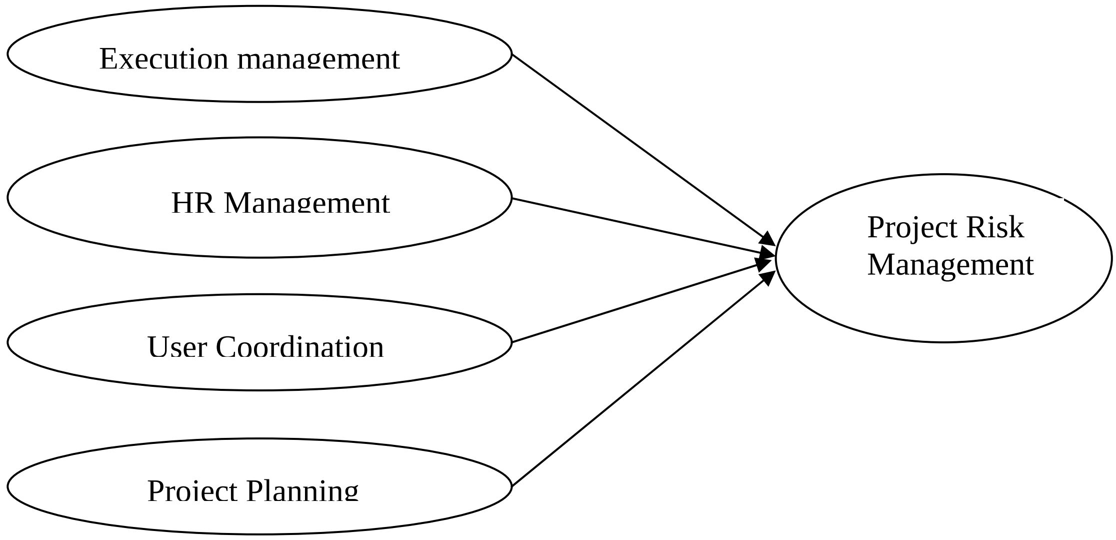


Figure 8.4: Second order model for Project Risk Management

Structural equation modeling is often used for testing theory associated with latent variable models because it enables the inference of complex relationships among variables which cannot be directly observed. Structural Equation Modeling can be used to confirm whether risk and risk management are better viewed as second-order factor models rather than first-order factor models.

8.2 INTRODUCTION TO STRUCTURAL EQUATION MODELING

Structural Equation Modeling (SEM) is a multivariate statistical methodology, which takes a confirmatory approach to the analysis of a structural theory. SEM provides researchers with the ability to accommodate multiple interrelated dependence relationships in a single model. Its closest analogy is multiple regression analysis, which can estimate a single relationship. But SEM can estimate many equations at once, and they can be interrelated, meaning that the dependent variable in one equation can be an independent variable in other equations. This allows the researcher to model complex relationships that are not possible with other multivariate techniques (Hair et al., 1998).

Advantages of SEM compared to multiple regression include more flexible assumptions (particularly allowing interpretation even in the face of multi-collinearity), use of confirmatory factor analysis to reduce measurement error by having multiple indicators per latent variable, graphical modeling interface, the desirability of testing models overall rather than coefficients individually, the ability to test models with multiple dependents, the ability to model mediating variables, the ability to model error terms and the ability to handle difficult data (time series with autocorrelated error, non-normal data, incomplete data).

The overall fit of a model can be assessed using a number of fit indices. There is broad consensus that no single measure of overall fit should be relied on exclusively and a variety of different indices should be consulted (Tanaka, 1993). The indices used include Chi-square (χ^2), Goodness of Fit Index (GFI) (Joreskog and Sorbom, 1989), Non-normed Fit Index (NNFI) (Bentler and Bonett, 1980), Comparative Fit Index (CFI) (Bentler, 1990) and Root Mean Squared Residual (RMSR). GFI, AGFI, CFI and NFI are mentioned in chapter 4. Other measures are discussed here and the fit measures are summarized in Table 8.1.

The chi-square fit statistic. The fit statistic provides a statistical test of the null hypothesis that a predicted model fits the observed data (Hatcher, 1994). It compares the correlation/covariance matrix that is predicted by a model with the values in the observed correlation/covariance matrix. If a proposed model is a good fit with the observed data then the value will be small relative to the degrees of freedom in the model. A major drawback of the chi square statistic is its sensitivity to sample size. In cases where the sample size is greater than 200 subjects, the chi square is often significant (thus signifying poor fit) even if all other indicators show that the model provides a good fit (Bentler and Bonett, 1980; Hatcher, 1994; Joreskog and Sorbom, 1989). Since the sample size in the current research is large, another fit statistic, **the normed chi-square fit** measure can also be used to help assess model fit. The normed chi square adjusts the chi square statistic by the degrees of freedom of the model (Joreskog, 1993). The desired value for the normed chi square has varied in the literature, with some recommending that the value be less than 3.0 (Chin and Todd, 1995), while other suggest that values falling between 1.0 and 5.0 are indicators of adequate model fit (Segars and Grover, 1993; Wheaton, Muthen, Alwin, and Summers, 1977).

RMSR The root mean square residual (RMSR) is the square root of the mean of the squared residuals (the average of the residuals between observed and predicted input matrices) (Hair et al., 1998). A RMSR value of .05 or less is usually used as an indication of very good model fit while values upto 0.1 can be taken as an indicator of moderate fit

RMSEA The root mean square error of approximation (RMSEA) attempts to correct the tendency of the chi square statistic to reject any model with a large sample size (Hair et al., 1998). The RMSEA is computed based on sample size and the noncentrality parameter and degrees of freedom for the proposed model (Browne and Cudeck, 1993; Steiger, 1990). The value produced by the RMSEA represents the goodness-of-fit that could be expected if the model were estimated in the population and not just the sample used for the estimation (Hair et al., 1998). RMSEA value of .05 or less would indicate a good model fit, although values of .08 or less would still indicate a reasonable model fit (Browne and Cudeck, 1993)

Table 8.1: Summary of fit measures and guidelines for their acceptable values

Indicators of fit	**Target Values**	**Target Values for**
Normed Chi-square (χ2	< 3	< 5
GFI	>0.90	>0.80
AGFI	>0.80	>0.70
RMSR	<0.05	<0.10
RMSEA	<0.05	<0.08
CFI	>0.90	>0.80

8.3 EVALUATING RISK AS A SECOND ORDER CONSTRUCT

The first order and second order models of risk were shown in figure 8.1 and 8.2 respectively. Both these models were evaluated with SEM using AMOS 4.0. Table 8.1 shows the fit measures for both the models.

Table 8.2: Fit measures for the risk models.

Fit measures	Values for First order model	Values for Second order model
Chi square	4388	4455
Normed chi square	3.337	3.375
GFI	0.908	0.908
AGFI	0.848	0.845
CFI	0.895	0.893
RMSR	0.083	0.084
RMSEA	0.051	0.054
Parsimony Ratio	0.952	0.958

The first order model (measurement) of risk had many more parameters to be estimated than the second order (theoretical) model shown. The first order factor model had ten correlations, where the second order factor model had only five paths. The second-order factor model is more parsimonious (simple) than the first-order factor model because it is a constrained version of the measurement model. As a result, the goodness of fit measures can never be better for the second order factor model than they were in the first order factor model (Wallace, 1999). Therefore, the question is whether the second-order model, which is more parsimonious than the first-order model, can do as good a job of accounting for the co-variances between the first order factors. The adequacy of the second order model can be determined by examining the Target (T) coefficient (Marsh and Hocevar, 1985) where

$$T = chi\ square(\,first\ order\ \mathrm{model}) \div chi\ square(\mathrm{second}\ order\ \mathrm{model})$$

The T coefficient has an upper bound of 1, with higher values (>0.7) implying that the relationship among the first order factors is sufficiently captured by the second order factor (Marsh and Hocevar, 1985). In that case, it can be concluded that the second-order factor model fits no worse and is preferred on the basis of parsimony (Rindskopf and Rose, 1988).

As seen in table 8.1, for the second order risk model, with the exception of the chi square statistic, the measures of fit were either identical or very close to the values of the first order model. The T coefficient for comparing the first and second order models was a very high 0.985. Therefore, since the second-order factor model represented a more parsimonious representation of the model, it should be accepted over the first order model as a better representation of model structure.

Further support for the second-order factor models was found in the magnitude and the significance of the estimated parameters. Figure 8.5 shows the loadings of each of the first order factors onto the second order factor of risk. The t-values of all the loadings were significant at 1% level. These loadings, or parameter estimates, are similar to the reliability measures between a set of indicators and the construct that they measure. The high magnitude and significance of the loadings further validated the second order factor models.

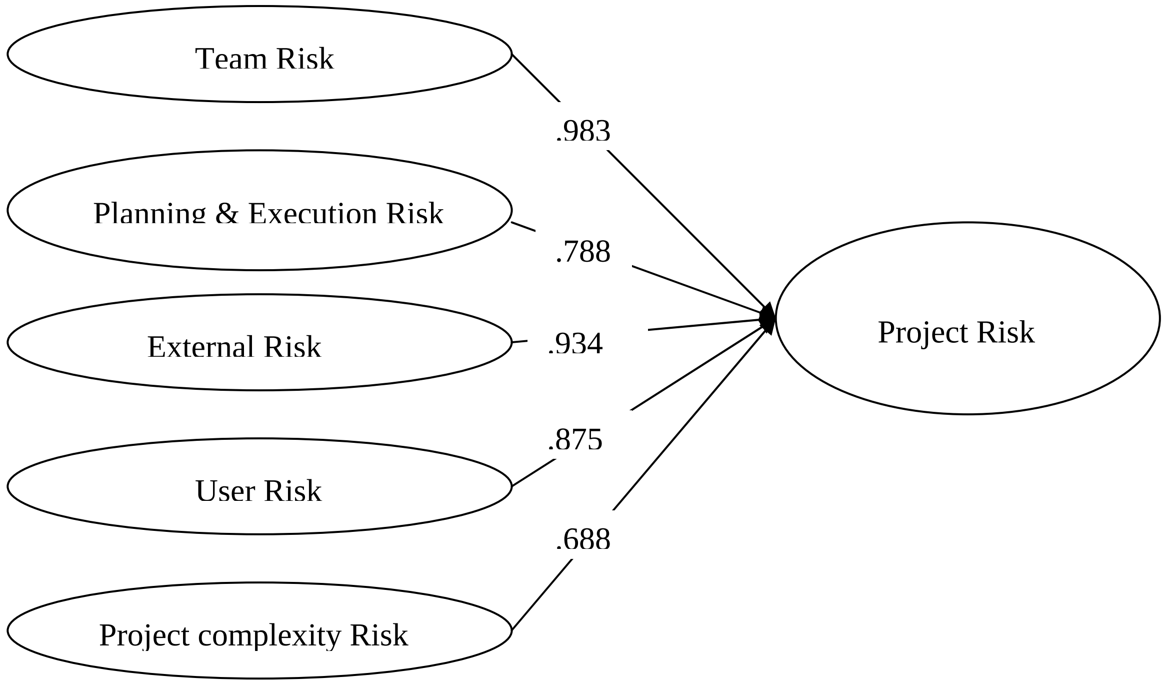


Figure 8.5 Parameter estimates between First and Second order factors of risk

8.4 EVALUATING RISK MANAGEMENT AS A SECOND ORDER CONSTRUCT

The first order model (figure 8.3) and the second order model (figure 8.4) for risk management were also compared. Table 8.2 shows the fit measures for both the models.

Table 8.3: Fit measures for the risk management models.

Fit measures	Values for First	Values for Second
Chi square	1346	1350
Normed chi	2.321	2.321
GFI	0.871	0.871
AGFI	0.853	0.853
CFI	0.914	0.914
RMSR	0.049	0.049
RMSEA	0.047	0.047
Parsimony	0.921	0.924

As in the case of risk, the second-order factor model for risk management was more parsimonious than its first-order factor model counterpart. With the exception of the chi square statistic, the remaining measures of fit were identical to the values of the first order model. The T coefficient for comparing the measurement and theoretical model is a very high 0.997. Since the second-order factor model represented a more parsimonious representation of the model, it should be accepted over the baseline as a better representation of model structure.

Figure 8.6 shows the loadings of each of the first order factors onto the second order factor of risk management.

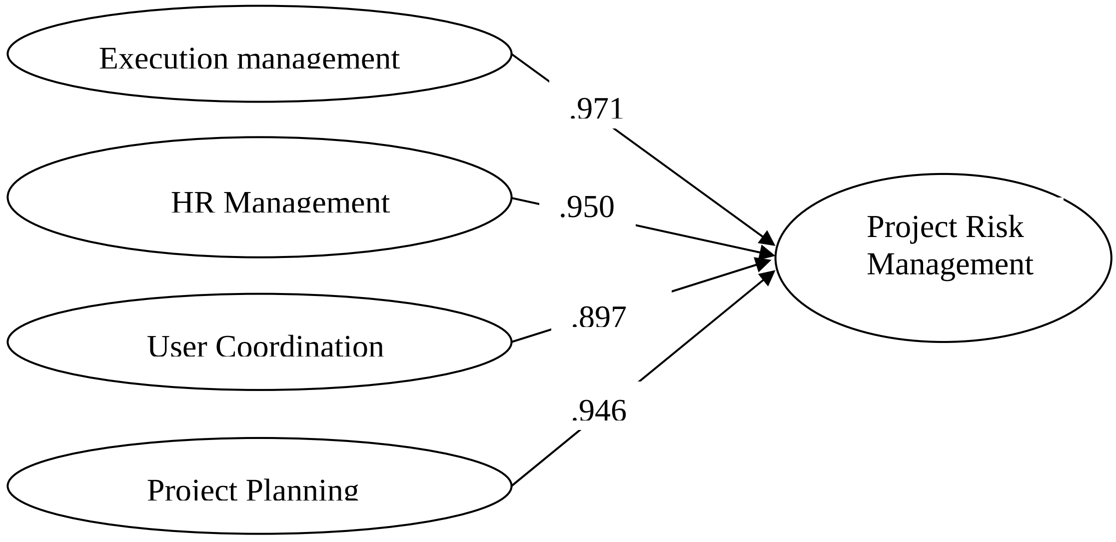


Figure 8.6 Parameter estimates between first and second order factors of risk management

The high magnitude and significance of the loadings further validated the second order factor model for risk management.

Therefore, the conceptualization of project risk and risk management as second order constructs consisting of sub-dimensional constructs appear to be the best representation.

8.5 LINKING PROJECT OUTCOME TO RISK AND RISK MANAGEMENT

The integrated model (figure 8.7) with risk management linked to project outcome directly as well as indirectly through the intervening variable risk was tested. Separate models were tested for each of the project outcome measures.

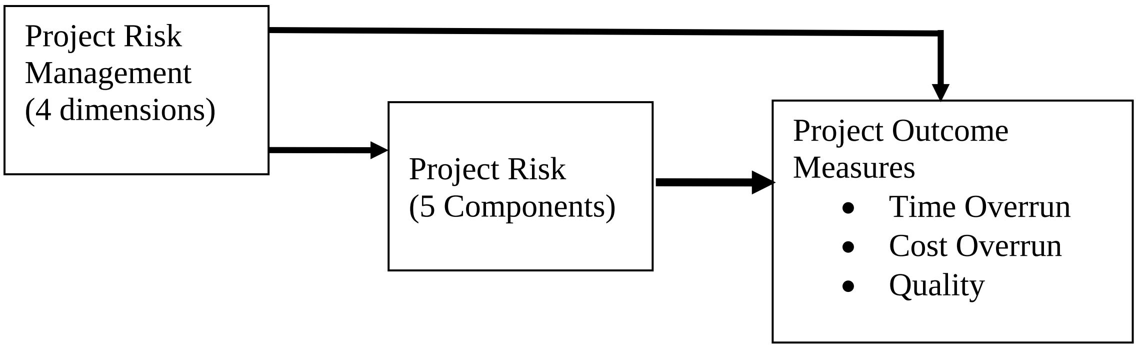


Figure 8.7 Integrated model

Two variants of this model were proposed and tested for each roject outcome measure. The first variant was a first order model where risk and risk management were treated as first order factors. This model implied that the five dimensions of risk as well as the four dimensions of risk management were independent of the others in their ability to predict project outcome. The other model variant was a second order model where risk and risk management figured as second order constructs where the first order factors were treated as acting collectively as members of a common system. In practical terms, this meant that any one dimension of risk or risk management would not be sufficient in explaining the outcome of a project. Rather, all dimensions were necessary adequately represent the effects of project risk and risk management on project outcome.

This research has taken the common approach of testing both the variants of the integrated model for each of the project outcome measures. The first order model for the project outcome measure “quality” is shown in Figure 8.8. Each risk factor was linked directly to the project outcome quality and each risk management factor was linked to the outcome variable quality as well as to each of the risk factors.

The alternative model was the second order model where each of the first order factors were shown to impact project outcome measure quality only through the underlying constructs namely software project risk and risk management. Figure 8.9 shows the second order model for the project outcome measure “quality”.

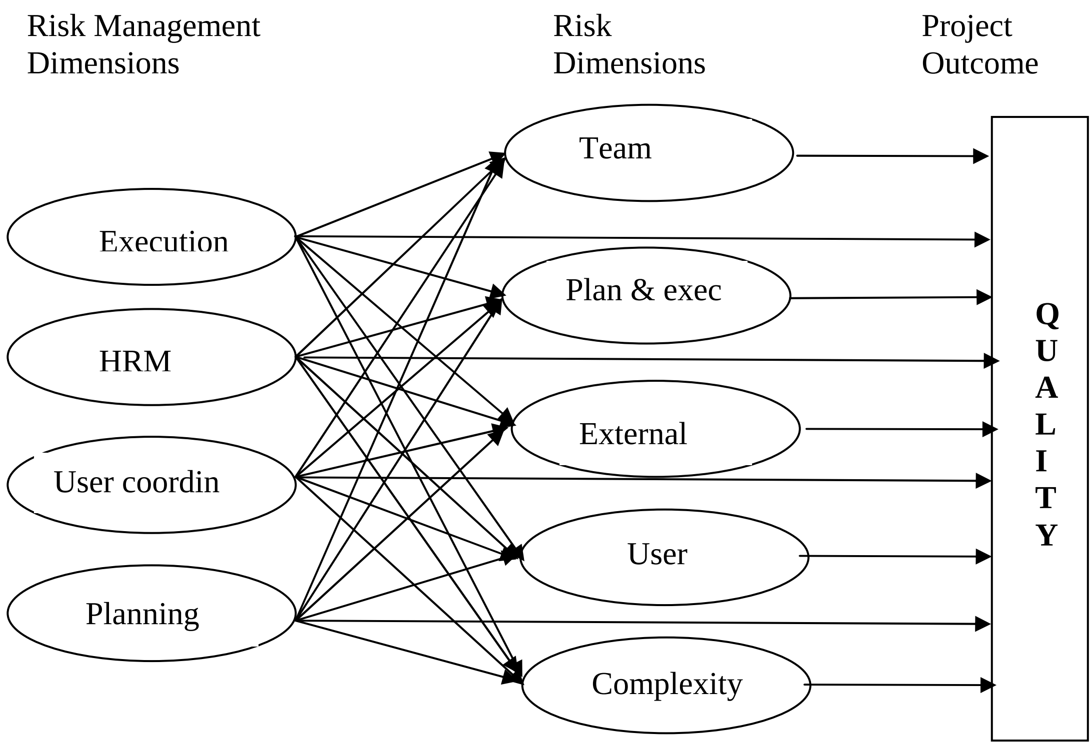


Figure 8.8 First order model of quality of software developed

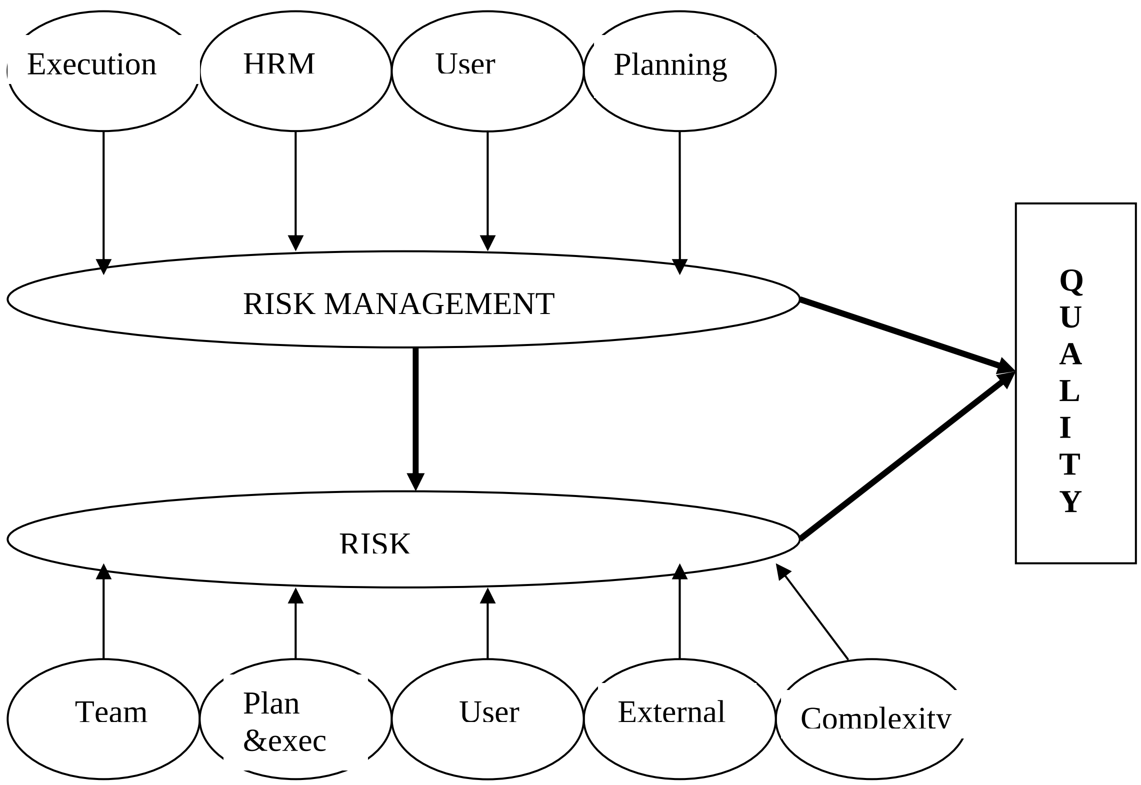


Figure 8.9 Second order model of quality of software developed

8.6 MODELING THE OUTCOME VARIABLE - QUALITY

Both the first order model (figure 8.8) and the second order model (figure 8.9) linking risk and risk management to quality of the software developed were tested with SEM. The values of the fit measures are reported in Table 8.3.

Table 8.4: Fit measures for the quality models.

Fit measures	Values for the first order model for quality	Values for Second order model for quality
Chi square	7167	7303
Normed chi square	1.803	1.823
GFI	0.868	0.863
AGFI	0.743	0.740
CFI	0.875	0.870
RMSR	0.091	0.098
RMSEA	0.049	0.050
Parsimony Ratio	0.949	0.957

The normed chi square and RMSEA indicated very good fit and other indicators indicated moderate fit. The moderate nature of fit values may be attributed to several causes. The first is the complexity of the model. The models has 55 indicator items linked to 5 risk dimensions and 36 indicator items linked to 4 risk management dimensions. The risk and risk management dimensions are linked among themselves and to the outcome variable. Complexity of the models may adversely affect some of these measures (Bearden, Sharma, and Teel, 1982). Second, the large sample size could influence the results, especially the chi square value (Joreskog and Sorbom, 1989). Finally, there could be many other project / organizational characteristics that may have a direct impact on the outcome but are not included in the model. Under these circumstances the moderate fit of the proposed model may be tolerated and accepted (Wallace, 1999). Figure 8.10 shows the AMOS representation of the first order model and figure 8.11 shows the AMOS representation of the second order model.

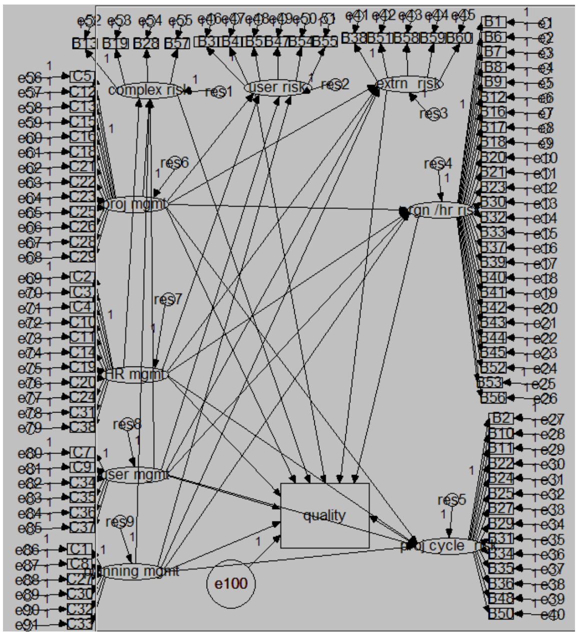


Figure 8.10 AMOS representation of the first order model of quality

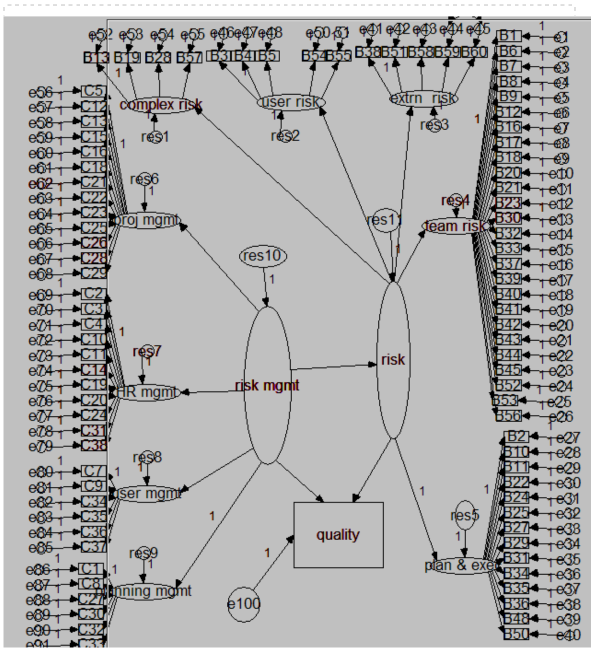


Figure 8.11 AMOS representation of the second order model of quality

The second order model was more parsimonious than the first order model. The first order model had 29 major paths linking the risk dimensions, risk management dimensions and quality where as the second order model had only 12 paths. The fit measures of the second order model were very close to that of the first model. The value for the T-coefficient comparing the models was 0.981. Thus it can be seen that the second factor model is more parsimonious than the first order model and does almost as good a job of accounting for the co-variance in the data as the first order model (Marsh and Hocevar, 1985).

The models were also compared based on the magnitude and the significance path coefficients. As mentioned earlier, these loadings are similar to the reliability measures and high magnitude and significance of the loadings further validate the model.

Table 8.4a and 8.4b shows the values and significance of all the loadings that are significant at $p < 0.05$.

Table 8.5a: Standardized path coefficients for the first order quality model

(only significant paths are shown)		
Paths	Standardized	Significance (P)
execution mgmt -> complexity risk	0.62	0.040
HR mgmt -> complexity risk	1.762	0.004
HR mgmt -> user risk	1.887	0.000
HR mgmt -> external risk	2.516	0.000
HR mgmt -> team risk	2.205	0.000
HR mgmt -> planning and execution risk	1.989	0.000
user mgmt -> planning and execution	0.974	0.039
user mgmt -> team risk	1.209	0.026
user mgmt -> external risk	1.75	0.005
user mgmt -> user risk	1.06	0.026
user mgmt -> complexity risk	1.117	0.036
planning mgmt -> complexity risk	-3.667	0.000
planning mgmt -> user risk	-3.63	0.000
planning mgmt -> external risk	-4.591	0.000
planning mgmt -> team risk	-4.17	0.000
planning mgmt ->planning and execution	-3.613	0.000
planning and execution risk -> Quality	0.234	0.030

Table 8.5b: Standardized path coefficients for the Second Order quality model

(only significant paths are shown)		
Paths	Standardized	Significance
risk management -> risk	*-0.43*	*0.000*
risk -> complexity	0.675	0.000
risk -> user risk	0.943	0.000
risk -> external	1.063	0.000
risk -> team risk	0.967	0.000
risk -> planning and execution	0.829	0.000
risk management -> execution	0.937	0.000
risk management -> HR mgmt	0.937	0.000
risk management -> user mgmt	0.94	0.000
risk management -> planning	0.991	0.000
risk management -> Quality	*0.365*	*0.000*
risk -> Quality		*0.000*

-0.36

The results indicated that out of the 29 paths in the first order model, only 17 paths were significant at p= 0.05 level and only 9 paths were significant at p = 0.01 level. Also, there was no consistency in the sign of the loadings even on the significant paths. In other words, the first order factors do not do a good job of explaining quality.

The second-order factor model had all the 12 paths highly significant at p = 0.01 level. Quality had a significant negative link with risk and positive link with risk management. The risk had a significant negative link with risk management. This shows risk management has an impact on quality directly as well as indirectly through the intervening variable risk. The squared multiple correlation for quality was 0.376 which is more than sufficient (Wallace 1999). Also, the loadings of the first order factors onto the second order construct were strong and positive, thereby providing further support for the model (Segars and Grover., 1993).

8.7 MODELING THE OUTCOME VARIABLE -TIME OVERRUN

A similar analysis was performed on the second project outcome variable "time overrun". Both the first order model and the second order model were tested. The models were similar to those tested for quality except that the variable quality was replaced with time overrun. The values of the fit measures are reported in Table 8.4.

Table 8.6: Fit measures for the time overrun models.

Fit measures	Values for the first order	Values for Second order
Chi square	7159	7283
Normed chi square	1.801	1.818
GFI	0.769	0.764
AGFI	0.743	0.742
CFI	0.895	0.891
RMSR	0.107	0.111
RMSEA	0.049	0.050
Parsimony Ratio	0.949	0.957

The normed chi square and RMSEA indicated very good fit and other indicators indicated moderate fit. Given the high level of complexity, the model could be accepted as moderately fitting. The fit measures of the second order model were very close to that of the first model. The value for the T-coefficient was a high 0.983.

Models were also compared based on the magnitude and the significance path coefficients, as in the case of quality. Tables 8.5a and 8.5b show the values and significance of the loadings that are significant at $p < 0.05$.

Table 8.7a: Standardized path coefficients for the First order time overrun model

(only significant paths are shown)		
Paths	Standardized	Significance (P)
execution mgmt -> complexity	0.638	0.041
HR mgmt -> complexity	1.701	0.007
HR mgmt -> user risk	1.844	0.000
HR mgmt -> external risk	2.421	0.000
HR mgmt -> team risk	2.135	0.001
HR mgmt -> planning and execution	1.927	0.000
user mgmt -> planning and execution	1.119	0.026
user mgmt -> team risk	1.376	0.017
user mgmt -> external risk	1.897	0.004
user mgmt -> user risk	1.171	0.019
user mgmt -> complexity	1.305	0.026
planning mgmt -> complexity	-3.808	0.000
planning mgmt -> user risk	-3.675	0.000
planning mgmt -> external risk	-4.639	0.000
planning mgmt -> team risk	-4.258	0.000
planning mgmt -> planning and execution	-3.693	0.000
team risk -> Time overrun	0.481	0.017
planning and execution risk -> Time	0.194	0.045

Table 8.7b: Standardized path coefficients for the Second Order time overrun model

(only significant paths are shown)				
Paths			Standardized	Significance
risk management	*->*	*risk*	*-0.430*	*0.000*
risk	->		0.688	0.000
risk	->	user risk	0.943	0.000
risk	->	external	1.066	0.000
risk	->	team risk	0.963	0.000
risk	->	planning and execution	0.837	0.000
risk management	->		0.935	0.000
risk management	->	HR	0.936	0.000
risk management	->	user	0.943	0.000
risk management	->	planning	0.990	0.000
risk	*->*	*Time*	*0.687*	*0.000*
risk management	->	Time	0.032	0.521

The results indicated that only 19 paths out of the 29 paths were significant in the first order model at p= 0.05 level and only 8 paths were significant at p = 0.01 level. Also, execution management was seen to be the only risk management dimension negatively linked to risk. All other risk management dimensions shared a positive relationship with risk which was hard to explain. Overall, the first order factors do not do a good job of explaining time overrun.

The second-order factor model had 11 paths highly significant at p = 0.01 level. The risk management - time overrun link was seen to be statistically insignificant. Time overrun had a significant and positive link with risk. The risk management was not seen to be significant in affecting time overrun. But it had a significant negative link with risk construct. This showed risk management did not have a direct impact on time overrun but had a significant indirect impact through the intervening variable risk. The squared multiple correlation for the time overrun was a strong 0.531. Thus the second order model finds support in the analysis.

8.8 MODELING THE OUTCOME VARIABLE -COST OVERRUN

The final outcome variable cost overrun was also modeled and analyzed similar to the other two variables. The values of the fit measures for the two models are reported in Table 8.5.

Table 8.8: fit measures for the cost overrun models.

Fit measures	Values for the direct	Values for Second order
Chi square	7140	7263
Normed chi square	1.797	1.813
GFI	0.769	0.765
AGFI	0.744	0.742
CFI	0.896	0.891
RMSR	0.082	0.106
RMSEA	0.049	0.050
Parsimony Ratio	0.949	0.957

Like the previous models, the fit measures threw up a mixed picture. The normed chi square and RMSEA indicated very good fit and other indicators indicated moderate fit. Overall the model could be accepted as moderately fitting. The fit measures of the second order model were very close to that of the first model. The value for the T-coefficient was a high 0.986. As expected, the parsimony ratio was better for the second order model.

The magnitude and the significance of the path coefficients were computed and reported for both the first order model and the second order model. Table 8.6a and 8.6b shows the values and significance of all the loadings that are significant at $p < 0.05$.

Table 8.9a: Standardized path coefficients for the First order cost overrun model

(only significant paths are shown)		
Standardized Path coefficients	Standardized Path	Significance (P)
execution mgmt ->	0.64	0.041
HR mgmt -> complexity risk	1.709	0.006
HR mgmt -> user risk	1.837	0.000
HR mgmt -> external risk	2.396	0.000
HR mgmt -> team risk	2.123	0.000
HR mgmt -> planning and execution risk	1.914	0.000
user mgmt -> planning and execution risk	1.076	0.029
user mgmt -> team risk	1.319	0.020
user mgmt -> external risk	1.834	0.004
user mgmt -> user risk	1.12	0.022

user mgmt -> complexity risk		1.244	0.031
planning mgmt -> complexity risk		-3.764	0.000
planning mgmt -> user risk		-3.618	0.000
planning mgmt -> external risk		-4.556	0.000
planning mgmt -> team risk		-4.193	0.000
planning mgmt -> planning and execution		-3.643	0.000
planning and execution risk -> cost overrun		0.337	0.003

Table 8.9b: Standardized path coefficients for the Second Order cost overrun model

(only significant paths are shown)		
Paths	Standardized Path	Significance (P)
risk management -> risk	-0.432	0.000
risk -> complexity risk	0.684	0.000
risk -> user risk	0.941	0.000
risk -> external risk	1.067	0.000
risk -> team risk	0.960	0.000
Risk -> planning and execution risk	0.839	0.000
risk management -> execution mgmt	0.935	0.000
risk management -> HR mgmt	0.935	0.000
risk management -> user mgmt	0.944	0.000
risk management -> planning mgmt	0.989	0.000
risk -> Cost overrun	0.461	0.000
Risk management -> Cost overrun	-0.130	0.024

The results again indicated that the first order model was not capable of explaining the variation in the cost overrun variable. Only 17 paths had significance at p= 0.05 level and only 9 paths were significant at p = 0.01 level. There was also the sign inconsistency problem making the model hard to explain. The second-order factor model had all 12 paths significant at 5% level and 11 paths significant at p = 0.01 level. The cost overrun had a significant positive link with risk. The risk management had a weak and less significant negative link with cost overrun. This means risk management has a weak direct impact on cost overrun but has a very significant and strong indirect impact through the intervening variable risk. The squared multiple correlation for the time overrun was an adequate 0.281. Thus the second order model is supported.

8.9 CONCLUSION

The second order factor models for risk as well as risk management were tested against their first order model counterparts. The second order models were seen to be equally acceptable from the fit values point of view but were much superior from the parsimony point of view.

The first order model and the second order model for risk and risk management were independently tested with each of the project outcome variables. The second order models were seen to be doing a better job of explaining the project outcome. It means that risk and risk management affect the project outcome collectively rather than as individual components. This finds support from the previous research of Wallace (1999).

The quality of the software developed had a positive relationship with risk management and negative relationship with risk. The other outcome variables namely time overrun and cost overrun had strong negative relationship with risk. Risk management did not have much effect on overrun variables. Risk was seen to be acting as an intervening variable between risk management and overrun variables.

The direct link between risk and project outcome is consistent with literature. Wallace (1999) proved that the second order model of risk directly influenced the outcome measures. Jiang and Klein (2000) also demonstrated the negative impact of risk items on a range of project effectiveness measures.

The findings on the direct as well as indirect impact of risk management on project outcome also find support from literature. Deephouse et.al (2005) found that effectiveness of the software processes had a stronger and direct linkage with project quality than with overrun measures. Nidumolu (1995) showed that the risk control measures such as coordination strategies may have a direct and/or indirect impact on project outcome measures. The study did not differentiate among various measures of project performance.

Many items in the risk management measure used in this study have one to one correspondence with dimensions of quality measure. For example, documentation, which is a dimension of the quality measure, can be hypothesized to be directly influenced by the risk management item “Manuals were prepared with lot of care”. On the other hand, the impact of risk management on time and cost reduction is more implicit than explicit. Hence the link between risk management and overrun variables can be assumed to be more indirect compared to the quality measure.

Thus it can be concluded that only an integrated model accounting for the direct and indirect impact of risk and risk management can explain the variation in project outcome measures satisfactorily.

****

CHAPTER 9

SUMMARY AND CONCLUSION

This Chapter gives a summary of the thesis. The research findings are summarized in Section 9.1. Section 9.2 discusses the implications for practice and section 9.3 explains the scope for future research. The limitations of this study are discussed in section 9.4. Section 9.5 contains the conclusion drawn from the study.

Failure of software development projects is a common problem reported from organizations across the globe. These failures can be attributed to various risk factors present in the software development projects. Experts in the area recommend that risk associated with software development projects must be identified and managed throughout the course of a development project. Various research studies have looked at the presence of risk and risk management in software development. But these studies had many limitations.

Most of the studies were reported from developed countries and hence conclusions cannot be generalized. Comprehensive and validated measures of risk and risk management were rarely used in these studies. The results were not validated across different types of projects or different types of organizations. Also, linkages among risk, risk management and project outcome were generally overlooked. No major research work on this theme was reported from India though India is one of the top destinations of software development.

Motivation for this research was derived from these limitations. The major objective of this research was to obtain a better understanding of software development project risk and risk management by identifying the themes that characterize them and link these constructs to the important project outcome measures. After the analysis of data the researcher could come out with some valuable information with regard to these constructs and their inter-linkages.

9.1 RESEARCH FINDINGS

The research was initiated with specific objectives such as to study the nature of risk and risk management, to explore the link between risk and risk management and to develop a model linking risk and risk management to project outcome. The major findings with respect to these objectives are discussed below.

9.1.1 Developing insights as well as reliable measures for risk and risk management

Software project risk was seen to be a multidimensional construct. Risk could not be directly observed but was indirectly measured through a series of indicators. The indicators were picked up from previous studies. These indicators were divided into a five factor structure namely team risk, planning and execution risk, user risk, external risk and project complexity risk. This finding went well with the previous research. There could be a difference in the number of factors but all the themes were properly captured. The Wallace study (1999) had identified six factors which were named as team risk, requirements risk, organizational environment risk, user risk, process management risk, and project complexity risk. Barki et.al (1993) had demonstrated a 5 factor structure namely organizational environment, technological newness, expertise, application size, and application complexity. Ropponen and Lyytinen (1997) study delineated six components of software development risk namely scheduling and timing risks, system functionality risks, subcontracting risks, requirement management risks, resource usage, performance risks and personnel management risks. The present research captured all these dimensions but under different factor names.

The study also developed a validated measure of software project risk in the Indian context. The researcher followed the accepted procedures in instrument development. Scales were developed for each of these risk dimensions. Various validity and reliability tests were conducted to finalize the instrument. Confirmatory Factor Analysis was used on the final data to confirm the risk factor structure that emerged from the exploratory factor analysis on the pilot data. A second order factor model was hypothesized for risk construct which was positively tested using structural equation modeling. This was in line with the findings of Wallace (1999).

The second construct used in the study namely risk management also was subjected to rigorous analysis. There were no validated measures readily available for risk management constructs. A series of indicators suggested by researchers such as McFarlan (1982), Davis (1982), Boehm 1989 and Alter and Ginzberg (1978) were assembled together to build a measure of software project risk management.

After a series of validity and reliability tests, a four factor structure emerged for risk management. The factors were project planning, execution management, human resource management and user coordination. The four factor structure was confirmed through Confirmatory Factor Analysis. A second order model for risk management was successfully tested through Structural Equation Modelling.

9.1.2 Exploring the link between risk and risk management construct.

Risk management is defined as a discipline designed to reduce or eliminate risks in a project (Boehm, 1989). Literature suggested a negative link between risk and risk management constructs. Most of the empirical studies in this regard looked at few selected risk items linked to selected risk management items. This research used elaborate and validated scales of risk and risk management to explore the hypothesized negative relationship between these constructs.

The research demonstrated the strong negative link between risk and risk management. The first canonical correlation coefficient was 0.6265. A series of ANOVA tests showed that risk scores varied significantly across categories of the following project and organizational characteristics: type of software developed, project duration, onsite / offshore split up, size of the organization and nature of the organization. These findings were generally in agreement with literature.

Regression models were developed for each risk dimension linking it to risk management dimensions and the relevant organizational and project characteristics. These models defined how each risk dimension was influenced by risk management strategies and the project and organizational characteristics. This analysis was in line with the work of many researchers such as Ropponen and Lyytinen (1997), Jiang et al (2000), Addision and Vallabh (2002). The exact nature of these models was different from previous studies. But this was expected as the models were developed many years back and that too in different countries.

Project planning was seen to be the most effective strategy in mitigating the various risk components. It appeared in all the regression models. User coordination strategies were found to have significant correlation with user risk. The most important project / organization characteristics present in risk models were: on site – offshore content of the project, size of the project organization, nature of the organization and project duration. This supplemented the work of previous researchers in crafting guidelines for specific risk items (Alter et al. 1978; McFarlan 1982; Boehm 1989; Charette 1996).

9.1.3 Model linking risk, risk management and project outcome

The research explored a model linking risk and risk management constructs to each of the project outcome variables namely quality, time overrun and cost overrun. The linkages among risk, risk management and project outcome were explored through testing of various models. The first model was a basic regression model where the project outcome measures were linked only to the risk dimensions. This was in line with some of the previous researchers (Wallace and Keil, 2004; Jiang et. al., 2000). Each project outcome variable was taken as the dependent variable with the five risk dimensions as independent variables. All models were statistically significant. But the explanatory powers of the models were limited with poor R^2 values. Also the loading signs of some of the risk dimensions were inconsistent with theory.

The second set of models had risk management linked to project outcome directly as well as indirectly with risk as the intervening variable (Nidumolu 1995, Na et al, 2006). Two variants of these models were tested with structural equation modelling. The first variant model called first order model treated the five factors of risk as well as the four dimensions of risk management as independent of the others in their ability to predict project outcome. The second variant models called second order factor models assumed that risk and risk management dimensions were acting collectively as members of a common system where one dimension would not be sufficient in explaining the outcome of a project. The second order factor models were seen to be more parsimonious and consistent with the loading signs. The analysis showed that risk had a direct link with all the project outcome measures. Risk management had a direct link only with quality. The impact of risk management on other outcome measures namely time and cost overruns was through the intervening variable risk. These findings supplemented many of the previous studies (Wallace, 1999; Jiang and Klein, 1999; Nidumolu, 1995; Deephouse, 2005)

9.2 IMPLICATIONS FOR PRACTICE

Software project managers across the globe are faced with the failure of software development projects on a regular basis. Proper identification of risk associated with software development projects and application of appropriate risk management strategies can reduce project failures.

This study has provided dimensions of software project risk and risk management that project managers might use for managing the project. Software project managers can evaluate a project based on team, planning and execution, user, external and complexity risks. They can counter these risks through risk management strategies namely project planning, execution management, user coordination and HR management.

This study has proposed specific strategies for reducing each risk dimension. But it also demonstrated that individual dimensions of risk are not independent in their impact on the outcome of a project. All the dimensions interact with each other to comprise software project risk and influence the outcome of a project. All the dimensions of risk must be managed in order to reduce the overall risk of a project and improve the chances of project success. Therefore, it is important that the project managers focus on the entire spectrum of potential risks though some level of prioritization can be made based on the levels of risk dimensions.

Organizations may benefit from using the results of this study to develop risk profiles for each of their software development projects. The instrument developed in this study can be used to create a risk profile that represents the assessment of risk associated with a particular project. Potentially high risk projects could be recognized earlier on and more appropriate decisions could be made about the desire to continue with a high risk project. By being able to better assess risk in a project, organizations can do a better job of balancing the number of high risk projects that they are undertaking with a complementary set of lower risk projects.

Organizations can also administer the instrument at multiple points during a project and track the changes in risk in a project as it progresses from beginning to end. Then they could monitor the risk mitigation strategies selected to counteract risk and by re-assessing risk at a later point they could determine the effectiveness of their strategy. Over time, this information would be a very valuable resource for identifying the appropriate risk management tools to be applied at different stages of a given project.

9.3 SCOPE FOR FUTURE RESEARCH

This study provided a good review of the existing research work on software project risk and risk management strategies. Various models linking risk related constructs were also reviewed and compared. This gives a strong theoretical foundation for future academic research.

This study developed and empirically tested two measurement models: one for software development risk and the other for risk management. These measures are grounded in both practice and theory. These measurement instruments identified the most prevalent risk items in software development and risk management techniques which can be used to counter these risks. The existence of validated and reliable measures will enable numerous future researchers to approach these constructs from the same perspective.

The study made extensive use of statistical techniques for developing and testing theories. These statistical techniques were explained in details so as to help new researchers to apply these tools in their research.

The study demonstrated that risk varied across different categories of projects or organizations. But more focused research work needs to be done in modeling risk and identifying appropriate risk management strategies for each of the categories. This knowledge could further assist project managers to tailor their risk management strategies more appropriately.

This research has developed general models linking project risk, risk management and project outcome. But the explanatory powers of the models are limited as indicated by the relatively low R^2 values. The field would benefit from a rigorous study of other factors which could have an influence on the relationship connecting project risk, risk management and project outcome. The field still lacks a framework that explains interaction of these constructs within the larger structure of other constructs.

Another potential area of research could involve administering the instrument in this study to different stakeholders involved with the same project and comparing their perceptions of risk and risk management. It is reasonable to believe that different project participants would view risk and risk management differently and the differences in their assessment would provide greater insights into the significance and direction of their differences.

Future studies could help to identify risk and risk management factors throughout the project rather than after a project has been completed as in this research. If prescriptions could be given on managing risks in projects which are at different points of the project life cycle, it will help the project manager to manage the projects better.

9.5 LIMITATIONS OF THE PRESENT WORK

Software project risk and risk management are two emerging domains. Both of them are complex constructs which many researchers constantly work on. Hence it is quite possible that this research may not have captured every aspect of these constructs even though an extensive literature review was conducted and experts in the area were consulted for inputs.

Though the scales developed and used in the study were validated, there is always scope for further refinement in order to increase their level of reliability and their ability to explain the variance associated with the constructs they measure.

NASSCOM directory was used as the sample frame for this study. The results should not be completely discounted for the possible extension to software development projects that take place in organizations outside of NASSCOM. The replication of the study across a broader sampling frame would provide additional validity for the findings and further empirical support for related theoretical studies.

A single-respondent or informant was used in this study. Although it is common to use a single respondent in academic research (Pinsonneault and Kraemer, 1993), it would be more desirable to have multiple respondents from each project independently assessing risk and outcome in order to validate the results.

9.5 CONCLUSION

India has become the major destination of software development and there is no dearth of software projects. However, the study on risk, risk management and project outcome with respect to software development projects was not easy. Lack of published material in the Indian context was the first challenge. This was overcome by the use of international literature and local expert opinion. Data collection posed the next major challenge. The delicacy of revealing project related information and opinion posed a major hurdle. The senior management of the organization had to be taken into confidence with regard to the confidentiality and the strictly academic nature of the study. The participating companies were promised a consolidated report of the research.

The study has identified major risk and risk management factors in the Indian context. Models grounded in theory are developed and empirically validated. The findings of the study could be used by practicing managers for better risk management. The models developed in this research can be refined and improved further by future researchers. The objectives laid down in the beginning of the research could be finally achieved to a high degree of satisfaction. As in all research, this work too has its limitations mentioned earlier.

This research was a very important learning experience for the researcher. Though the researcher had practical exposure to project management with software development organizations for six years, this research has brought in new dimensions to his understanding of software development risk and risk management. Also, this work has helped him to appreciate the role and application of research methodology in management research.

****

REFERENCES

1. Abdel Hamid, T.K., (1993). Software Project Control: An Experimental Investigation of Judgement with Fallible Information. *IEEE Transactions on Software Engineering, 19(6),* 603-612.

2. Addison, T, Vallabh, S., (2002). Controlling Software Project Risks – An Empirical Study of Methods used by Experienced Project Managers. *Proceedings of SAICSIT 2002*, 128 – 140.

3. Ahire, S.L., Golhar, D.Y., Waller, M.A., (1996). Development and Validation of TQM Implementation Constructs. *Decision Science 27*, 23 – 56.

4. Alter, S., (1979). Implementation risk analysis. *TIMS Studies in Management Sciences 13 (2),* 103-19.

5. Alter, S., Ginzberg, M., (1978). Managing Uncertainty in MIS Implementation. *Sloan Management Review (Fall 1978),* 23 - 31

6. Anastasi, A., and Urbina S., (1997). *Psychological Testing. 7th ed.* Prentice-Hall, Upper Saddle River, NJ.

7. Anderson, J., and Narasimhan, R., (1979). Assessing Implementation Risk: A Methodological Approach. *Management Science* 25th June, 512 – 521.

8. Anderson, J.C., Gerbing, D.W., (1979). Structural Equation Modelling in Practice: A Review and Recommended Two –Step Approach. *Psychological Bulletin 103(3),* 411 – 423.

9. Arifoglu, A., (1993). A Methodology for Software Cost Estimation. *Software Engineering Notes 18(2),* 96-105.

10. Aronow, J. A. P., (2004). *The Impact of Organizational Politics on the work of the Internal Human Resource Professional*. University of Wisconsin – Stout, May 2004

11. Asundi, J.M., Arora, A., Arunachalam, V.S. and Fernandes, R. (2001). The Indian Software Services Industry. *Research Policy (30) 8,* , 1267-1287

12. Augustine, S. Payne, B. Sencindiver, F., Cock, S., (2005). Agile Project Management - Steering from the Edges. *Communications of the ACM 48 (12),* 85 - 89

13. Barki, H., Rivard, S., Talbot, J., (1993). Toward an Assessment of Software Development Risk. *Journal of Management Information Systems 10(2),* 203-225.

14. Barki, H., Rivard, S., Talbot, J., (2001). An integrative Contingency Model of Software Project Risk Management. *Journal of Management Information Systems 17 (2),* 37-69.

15. Bayer, J. and Muthig, D., (2006). *A View-Based Approach for Improving Software Documentation Practices. ecbs, 269-278*. 13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems (ECBS'06)

16. Bearden, W.O., Sharma, S., Teel, J.E., (1982). Sample Size Effects on Chi Square and Other Statistics Used in Evaluating Causal Models. *Journal of Marketing Research 19,* 425 – 430.

17. Beaver, M. Justin, S., Guy, A., (2006). The Effects of Development Team Skill on Software Product Quality. *ACM SIGSOFT Software Engineering Notes 31(3),* 1 – 5.

18. Beitz, A., Wieczorek, I., (2000). *Applying benchmarking to learn from best practices*. Proceedings of 2nd International Conference on Product-Focused Software Process Improvement, Oulu.

19. Bentler, P.M., (1990). Comparative fit indexes in structural models. *Psychological Bulletin 107*, 238 – 246.

20. Bentler, P.M., Bonett, D.G., (1980). Significance Tests and Goodness of Fit in the Analysis of Covariance Structures. *Psychological Bulletin 88*, 588 -606.

21. Blackburn, J.D., Scudder, G.D., Van Wassenhove, L.N., (1996). Improving speed and productivity of software development: A Global Survey of Software Developers. *IEEE Transactions on Software Engineering 22(12),* 875 – 885.

22. Boehm, B. W., (1979). R&D Trends and Defense Needs. In P. Wegner (Ed.), *Research Directions in Software Engineering (44-86)*. Cambridge, Massachusetts MIT Press.

23. Boehm, B. W., (1981). *Software Engineering Economics*. Englewood Cliffs. NJ-Prentice- Hall.

24. Boehm, B. W., (1983). Seven Basic Principles of Software Engineering. *Journal of systems and software 3(1),* 3 – 24.

25. Boehm, B. W., (1987). Improving Software Productivity. *IEEE Computing 20(9),* 43-57.

26. Boehm, B. W., (1989). *Software Risk Management. Tutorial*. IEEE Computer Society Press, Los Alamitos, CA.

27. Boehm, B. W., (1989). *Risk Management Practices: The Six Basic Steps. Software Risk Management* 115-147. IEEE Computer Society Press, Piscataway, NJ, USA

28. Boehm, B. W., (1991). Software Risk Management: Principles and Practices. *IEEE Software 8(1),* 32-41.

29. Boehm, B., Papaccio, P., (1988). Understanding and controlling software costs. *IEEE Transactions on Software Engineering 14 (10),* 1462-1477.

30. Boehm, B., Ross, R., (1989). Theory-W Software Project Management Principles and Examples. *IEEE Transactions on Software Engineering 15 (7),* 902-916.

31. Boyer, K.K., Verma, R., (2000). Multiple raters in survey-based operations management research. *Production and Operations Management 9 (2),* 128-140.

32. Brewster, C., Hegewisch, A Tregaskis, O., and Mayne, L. (1994). Flexible working practices: the controversy and the evidence. In Brewster, C. and. Hegewisch, A (Eds), *Policy and Practice in European Human Resource Management.* Routledge, London.

33. Brooks, F. P., (1995). *The Mythical Man-Month: Essays on Software Engineering*. Addison – Wesley Publishing Co. Massachussets

34. Brown, S.L., Eisenhardt, K.M., (1995). Product development: Past research, present findings and future findings. *Academy of Management Review Vol. 20,* No. 2, pp. 343-378.

35. Browne, M.W., Cudeck, R., (1993). Alternative Ways of Assessing Model Fit. In K.A. B.J.S. Long (Ed.), *Testing Structural Equation Models 136 – 162*, Newbury Park, CA: Sage.

36. Bruce, P., & Pederson, S. M., (1982). *The Software Development Project: Planning and Management*, New York, NY: John Wiley & Sons, Inc.

37. Buffardi, L.C., Smith, J.L., O' Brien, A.S. and Erdwins, C.J., (1999). The impact of dependent-care responsibility and gender on work attitudes. *Journal of Occupational Health Psychology 4 (4)*, 356-67.

38. Burchett, R., (1982). Avoiding Disaster in Project Control. *Data Processing Digest 28 (6),* 1 – 3, 1994.

39. Cacioppe. R., (1999). Using Team – Individual Reward and Recognition Strategies to Drive Organizational Success. *Leadership and Organizational Development Journal 20/6,* 322 – 331.

40. Campion M A., (1991). Managing and Measurement in turnover – Comparison of Alternative measures and recommendations for research. *Journal of Applied Psychology 76 (2),* 199 – 212.

41. Cappelli, P., (2000). Managing Without Commitment. *Organizational Dynamics 28(4),* 11-24.

42. Carmel E., Sawyer, S., (1998). Packaged software development teams: what makes them different? *Information Technology and People 11(1),* 7-19.

43. Carmines, E.G., Zeller, R.A., (1990). *Reliability and Validity Assessment*, Sage Publications, USA.

44. Carroll, J. M., Smith-Kerker, P. L., Ford, J. R., Mazur, S. A., (1986). *The Minimal Manual. Technical Report RC 11637 (#52295),* Yorktown Heights, NY: IBM, Thomas J. Watson Research Center.

45. Cash, Jr. J.I., McFarlan, F.W., McKenney, J.L., (1988). *Corporate information systems management: The issues facing senior executives (2nd Ed.),* Homewood: Irwin, IL.

46. Casher, J. D., (1984). How to Control Risk and Effectively Reduce the Chance of Failure. *Management Review 73(6),* 50-54.

47. Chakraborty, C., Dutta, D., (2001). *Indian Software Industry: Growth Patterns, Constraints and Government Initiatives*. Retrieved from http://web.mit.edu/profit/htdocs/thesis/Madhav-SDM-Thesis.doc.

48. Charette, R. N., (1996). *Software Engineering Risk Analysis and Management*. McGraw-Hill Software Engineering Series.

49. Chin, W.W., Todd, P.A., (1995). On the use, Usefulness, and Ease of Use of Structural Equation Modelling in MIS Research: A Note of Caution. *MIS Quarterly 19(2),* 237 – 246

50. Churchill. P. C., (1979). A Paradigm for Developing better measures of Marketing Constructs. *Journal of Marketing Research 16,* 64 – 73.

51. Collins, R.W., Kirsch, L.J., (1999). *Crossing Boundaries: The Deployment of Global IT Solutions.* Pennaflex, Cincinnati, Ohio.

52. Command, U. A. F. S., (1988). *Software Risk Abatement*. (AFSC/AFLC Pamphlet 800-45). Andrews AFB, Md.

53. Cooper, K. G., (1994). The rework cycle- benchmarking for the project manager. *Project Management Journal 24(1),* 17 – 22.

54. Corbato F.J., and Clingen, C.T. (1979). A managerial view of the Multics system development. In P. Wegner (Ed.), *Research Directions in Software Technology (pp. 139 – 158)*. The MIT Press, Cambridge.

55. Corso, A. (1993). *Cost-Benefit Analysis on a Large-Scale ADA Project.* Metodi Die Conduzione Aziendale, University of Genova, Genova., 1992 ACM 0-89791-504-81 921 0500- 0338.

56. Cronbach, L. J. H., Glesner, G. C., (1965). *Psychological Testing and Personnel Decisions,* Urbana: University of Illinois Press, 1965.

57. Cronbach, L.J., (1951). Coefficient Alpha and the Internal Structure of Tests. *Psychometrica 16(3),* 297 – 334.

58. Curtis, B., (1988). A field study of the software design process for large systems. *Communications of the ACM 31(11),* 1268-1287.

59. Davis, G.B., (1982). Strategies for Information Requirements Determination. *IBM Systems Journal* 21(1), 4-30.

60. Dedolph, F., Michael, (2003). The Neglected Management Activity: Software Risk Management. *Bell Labs Technical Journal 8(3),* 91–95.

61. Deephouse, C., Mukhopadhyay, T., Goldenson, D. R., Kellner, Marc I. (2005). Software Processes and Project Performance. *Journal of Management Information Systems (Winter 1995-96) 12(3),* 185-203.

62. DeMarco, T., (1982). *Controlling Software Projects*, New York, Yourdon Press.

63. DeRoze, B. C., Nyman, T. H., (1978). The Software Life Cycle - A Management and Technical Challenge in the Department of Defense. *IEEE Transactions on Software Engineering 4(4),* 309-318.

64. Dess, G. D., Shaw, J. D., (2001). Voluntary Turnover, Social capital and organizational performance, *Academy of Management Review 26 (3),* 446 – 456.

65. Disterer, G., (2002). Management of Project Knowledge and Experiences. *Journal of Knowledge Management 6(5),* 512 – 520.

66. Doll, W. J., (1985). Avenues for Top Management, Involvement in successful MIS Development. *MIS quarterly 9 (1),* 17-35.

67. Dominiak, M., (2006). To increase productivity in lean times, invest in training. *Television wreck 25(15),* 8 – 18.

68. Drory, A., (1993). Perceived Political Climate and Job Attitudes. *Organization Studies 14 (1),* 59 – 71.

69. Earl, M. J. (1996). The Risks of Outsourcing IT. *Sloan Management Review 37(3),* 26-32.

70. Evans, M. W., Piazza, P., Dolkas, B., (1983). *Principles of Productive Software,* John Wiley and Sons, Inc. New York, NY, USA.

71. Evans, P., (1986). The strategic outcomes of human resource management. *Human Resource Management 25(1),* 149-167.

72. Ewusi-Mensah, and K., Przasnyski, Z. H., (1991). On Information Systems Project Abandonment: An Exploratory Study of Organizational Practices. *MIS Quarterly 15(1),* 67-85.

73. Fagan, C., (2001). The temporal reorganization of employment and the household rhythm of work schedules. *American Behavioral Scientist 44(7),* 1199-1212.

74. Fairley, R., (1994). Risk Management for Software Projects. *IEEE Software 11 (3),* 57 – 67.

75. Farquhar, J. A., (1970). *A Preliminary Inquiry into the Software Estimation Process (Technical Report, AD F12 052),* Alexandria, VA: Defense Documentation Center.

76. Ferrin, D.M., Muthler, D., Muller, M.J., (2002). *Six sigma and simulation, so what's the correlation?* Proceedings of the 2002 Winter Simulation Conference, Manchester, 1439-43.

77. Fox, J. M., (1982). *Software and Its Development*, Englewood Cliffs, N.J.: Prentice- Hall, Galbraith

78. Fuerst, W L., Cheney P H., (1982). Factors affecting the perceived support systems utilization of computer based decision support systems in the oil industry. *Decision Sciences 13(3),* 443 – 69.

79. Garbett, J., Morton, J. (2006). Retrieved from *www.marketingweek.co.uk* on 27.04.06.

80. Gibbs, W.W., (1994). Software's Chronic Crisis. *Scientific American 271*, 86 - 95

81. Gilb, T., (1988). Estimating the Risk. In T. Gilb (Ed.), *Principles of Software Engineering Management (pp. 71-82)*: Addison-Wesley.

82. Ginzberg, M., (1981). Early Diagnosis of MIS Implementation Failure: Promising Results and Unanswered Questions. *Management Science, 27(4),* 459-478.

83. Goleman, D., (2000). Leadership that gets results. *Harvard Business Review, March/April 1978, 2,* 78-90.

84. Gong, R., (1990). The Development of a Task-Oriented, Minimal Content User's Manual, *ACM Sigchi Bulletin 21(3)* 29-33.

85. Gordon, P., (1999). To err is human, to estimate divine. *Information Week,* 65 – 72.

86. Guinan, P., Cooprider, J. G., Faraj, S., (1998). Enabling software development team performance during requirements definition: A behavioural versus technical approach. *Information Systems Research 9(2),* 101–125.

87. Guthrie, A., (1974). Attitudes of User-Managers towards Management Information Systems, *Management Informatics 3(5),* 221-232.

88. Haimes, Yacov Y., (1991). Total Risk Management. *Risk Analysis 11(2),* 169-171.

89. Hair, J.F., Anderson, R.E., Tatham, R.L., Black, W.C., (1998). *Multivariate Data Analysis,* Prentice-Hall International, New Jersey, USA.

90. Hatcher, L., (1994). *A Step by Step Approach to Using the SAS System for Factor Analysis and Structural Equation Modelling.* Cary NC: SAS Institute Inc.

91. Heales, J., (1995). An Empirical Analysis of Information Systems Change. In DeGross, J.I., Ariav, G., Beath, C., Hoyer, R. & Kemerer, C. (eds) *Proceedings of 16th International Conference on Information Systems. pp. 352–353*. ICIS, Pittsburgh, PA.

92. Heckhausen, H., (1989). *Motivation and Handein, 2ed.,* Springer Verlag, Hamburg

93. Henderson, J. and Lee, S., (1992). Managing Information System Design Teams: A Control Theories and Perspective. *Management Science 38 (6),* 757-777.

94. Henry, J.W., and Stone, R.W., (1994). A Structural Equation Modeling of End-User Satisfaction with a Computer-Based Medical Information System. *Information Resources Management Journal 7(3),* 21 – 33.

95. Hickey, A. M., and Davis, A, M., (2004). Unified Model of Requirements Elicitation. *Journal of Management Information Systems (Spring 2004) 20(4),* 65–84.

96. Hofstede, G., (1991). *Cultures and organizations: Software of the Mind: Intercultural Cooperation and its importance for Survival.* McGraw-Hill International, London

97. Humphrey, W. S., Snyder, T. R., & Willis, R. R., (1991). Software Process Improvement at Hughes Aircraft. *IEEE Software 8(7),* 11-23.

98. Hunter, J.E., Schmidt, F.L., (1982). Fitting people to jobs: The impact of personnel selection on national productivity. In E.A. Fleishman & M.D. Dunnette (Eds.), *Human performance and productivity: Vol. 1,* Human capability assessment 233-284. Hillsdale, N.J.- Erlbaum.

99. Huselid, M., (1995). The impact of HRM practices on turnover, productivity, and corporate financial performance. *Academy of Management Journal 38*, 673-703.

100. Jarvenpaa, S. L., Ives, B., (1991). Executive involvement and Participation in the Management of Information Technology. *MIS Quarterly 15 (2),* 205-277.

101. Jerome, L., Kleiner, B.H., (1995). Employee morale and its impact on service: What companies do to create a positive service experience, *Managing Service Quality, 5(6),* 21 – 25.

102. Jiang, James, J. and Klein, G., (2000). Software Development Risks to Project Effectiveness, *Journal of Systems and Software 52(1),* 3.

103. Jiang, James, J. and Klein, G., (2002). The Importance of building a foundation for user- involvement in Information System Projects, *Project Management Journal 33(1),* 20 – 26.

104. Jones, C., (1994). *Assessment and Control of Software Risks.* Englewood Cliffs, NJ: Yourdon Press.

105. Jones, C., (1995). Risks of Software System Failure or Disaster. *American Programmer 8(3),* 2-9.

106. Joreskog, K.G., (1993). Testing Structural Equation Models. In K.A. Bollen & J.S. Long (Eds.). *Testing Structural Equation Models*, Newbury Park, CA, Sage Publications.

107. J¨oreskog, K.G., & S¨orbom, D. (1996). *LISREL 8: Structural equation modeling with the SIMPLIS command language*. Chicago, IL: Scientific Software International.

108. Junhe., (2004). *Knowledge Impacts of User Participation: A Cognitive Perspective.* SIGMIS'04, April 22–24, 2004, Tucson, Arizona, USA.

109. Jurison, J., (1999). Software Project Management: The Manager's View. *Communications of AIS 2(3),* Article 17.

110. Kangari, R., Boyer, L. T., (1989). Risk Management by Expert Systems. *Project Management Journal 20(1),* 40 – 48.

111. Kaplan, R.M., Scauzzo, D.P., (1993). *Psychological Testing: Principles, applications and issues,* Pacific Grove, CA.

112. Kaplan, S., Garrick, J. B., (1981). On the Quantitative definition of risk. *Risk Analysis 1,* 11 – 27.

113. Karasek, R.A., (1979). Job demands, Job Decision Latitude, and Mental Strain: Implications for Job Design. *Administration Science Quarterly 24*, 385-408.

114. Karasek, R.A., Baker, D., Marxer, F., Ahlbon, A., and Theorell, T., (1981). Job decision latitude, job demands, and cardiovascular disease: a prospective study of Swedish men. *American Journal of Public Health 71,* 694-705.

115. Katzenbach J R., Smith, D. K., (1993). *The Wisdom of Times: Creating the high Performance Organization,* HBS Press, Boston M A.

116. Keider, S. P., (1974). Why Projects Fail. *Datamation 20 (12),* 53 – 55.

117. Keider, S.P., (1984). Why Systems Development Projects Fail. *Journal of Information Systems Management 1(3),* 33 – 38.

118. Keil, M., (1995). Pulling the Plug: Software Project Management and the Problem of Project Escalation. *MIS Quarterly 19(4),* 421-441.

119. Keil, M., Cule, E. Paul., Lyytinen. Kyle., Schmidt, C., Roy, A., (1998) Framework for identifying software project risks. *Communications of the ACM 41(11),* 765-83.

120. Kemerer, C.F., Sosa, G.L. (1988). Barriers to successful strategic information systems. *Planning Review 16(5),* 20–23.

121. Kesner, R.K., (2006). Managing ERP System Deployment, A consideration of best Practices, Projects and Profits. *ICFAI University Press, June,* 27 – 47.

122. Kim, E.S., Muller, R.S., (1988). IC-Processed Piezoelectric Microphone. *IEEE Electr. Dev. Letters, 8,* 467-468, October, 1988.

123. Kindle, S., (1992). The Computer that ate the company. *Financial World 161 (7),* 96-98.

124. Kirsch, J. L., (1996). The management of complex tasks in organizations: controlling the systems development process, *Organization Science, Vol.7(1),* 1 - 21

125. Kokol, P., Zumer, V., Stiglic, B., (1991). *New Evaluation Framework for Assessing the Reliability of Engineering, Software Systems Design Paradigm.* In (ed.) Proceedings of RRES'1991, Milano, Elsevier.

126. Lin, Z.F., Chen, D.Q., Wen, F., (2004). Discussing risk management strategy in IT outsourcing. *Journal of Science, (2),* 133-135.

127. Lyytinen K., Mathiassen L., Ropponen J., (1998). Attention Shaping and Software Risk – A categorical Analysis of Four Classical Risk Management Approaches. *Information Systems Approach 9(3),* 233 – 255.

128. Lyytinen, K, (1988). Expectation Failure Concept and Systems Analysts View of Information System Failures: Results of an Exploratory Study. *Information and Management 14,* 45 – 46.

129. Lyytinen, K., Hirschheim, R. A., (1987). Information Systems Failure: A Survey and Classification of the Empirical Literature. *In Oxford Surveys in Information Technology, 4 (Ed. Zorkoczy, P. I.)* Oxford University Press, Oxford, 257-309.

130. Lyytinen, K., Mathiassen, L., Ropponen, J., (1993). *An Organizational Analysis of Software Risk Management Approaches.* Working Paper, University of Jyvaskyla, Jyvaskyla, Finland.

131. MacCrimmon, Kenneth R., Wehrung, Donald A., (1984). The Risk in –Basket, *Journal of Business 57(3),* 367 – 387.

132. Mak, B., Sockel, H., (1999). A Confirmatory factor analysis of IS employee motivation and retention. *Information and Management 38,* 265 – 276.

133. Malone, Thomas W., (1985). Interfaces in Organizations: Supporting Group Work. In: Borman, Lorraine and Curtis, Bill (eds.) *Proceedings of the ACM*

CHI 85 Human Factors in Computing Systems Conference April 14-18, 1985, San Francisco, California. 65.

134. Mann, Joan. (2000). Undergraduate global IT education: An experiential approach using the concept of fit. *IRMA Conference 2000,* 917-918

135. March, J. G., Shapira, Z., (1987). Managerial Perspectives on Risk and Risk Taking. *Management Science 33(11),* 1404-1418.

136. Margono, J., Rhoads, T. E., (1992). Software Reuse Economics: Cost-Benefit Analysis on a Large-Scale ADA Project. *Proceedings of 14th International Conference on Software Engineering,* 338-348, May11-15, Melbourne, Australia.

137. Marsh, H.W., Hocevar, D., (1985). Application of Confirmatory Factor Analysis to the study of Self- Concept: First and Higher Order Factor Models and Their Invariance across Groups. *Psychological Bulletin 97,* 562 – 582.

138. Martin, N., (1981). Industrial Software Training Opportunities for Computer Professionals. *ACM '81,* November 9-11.

139. McBeath, D.F., Keezer, W.S., (1993). Simulation Support of Software Development. *Proceedings of the 1993 Winter Simulation Conference*, Evans, G.W., Mollaghasemi, M., Russell, E.C., and Biles, W.E., eds., *IEEE*, Piscataway, NJ, 1143.

140. McCarthy, R., (1979). Applying the Technique of Configuration Management to Software. In D. J. Reifer (Ed.), Tutorial: *Software Management, 263-268* Washington, D.C.: IEEE Computer Society Press.

141. McComb, D., & Smith, J. Y., (1991). System Project Failure: The Heuristics of Risk. *Journal of Information Systems Management 8(1),* 25-34.

142. McDonough E.F. III., (2000). Investigations of factors contributing to the success of cross-functional teams. *Journal of Product Innovation Management 17,* 221 – 235.

143. McFarlan, F. W., (1981). Portfolio Approach to Information Systems. *Harvard Business Review 59(5),* 142-150.

144. McFarlan, F. W., (1982). Portfolio Approach to Information Systems. *Journal of Systems Management (January),* 12-19.

145. Metzger, P W., (1981). *Managing a Programming Project (2nd ed.).* Englewood Cliffs, NJ: Prentice-Hall.

146. Moore, D. S., (1995). *The basic practice of statistics*. NY: Freeman and Co

147. Moore, G. C., & Benbasat, I., (1991). Development of an Instrument to Measure the Perceptions of Adopting an Information Technology Innovation. *Information Systems Research 2(3),* 192-222.

148. Moore, J. H., (1979). A Framework for MIS Software Development Projects. *MIS Quarterly 3(1),* 29 – 38.

149. Moore, J.E., (2000). One road to turnover: an examination of work exhaustion in technology professionals. *MIS Quarterly 24(1),* 141-68.

150. Morgan, Jeannette, N., (2005). Why the software industry needs a ghost buster. *Communications of the ACM 48(8),* 129 – 133.

151. Moynihan, T., (1997). How experienced Project Managers access risk. *IEEE Software 14 (3),* 35-41.

152. Mursu, A., (2000). *University of Jyväskylä, Dept. of Computer Science and Information Systems,* PL 35, FIN-40351 Jyväskylä, Finland.

153. Na, Kwan-sik, Simpson, J., T., Li, Xiaotong, Singh, T., Kim, Ki-Yoon., (2006). Software development risk and project performance measurement: Evidence in Korea, *Journal of Systems and Software 80(4),* 596-605.

154. Nash S., Redwine, S., (1999). People and Organizations in Software Production: A Review of the Literature. *Computer Personnel* 11(3), 1021.

155. Neale, J.M., Liebert, R.M., (1986). *Science and Behaviour: An Introduction to Methods of Research*. Prentice – Hall International Inc., New Jersey.

156. Newman, M., Sabherwal, R., (1991). Information system development: four process scenarios with case studies. *Journal of Information Systems 5,* 84–101.

157. Nidumolu, S., (1995). The Effect of Coordination and Uncertainty on Software Project Performance: Residual Performance Risk as an Intervening Variable. *Information Systems Research,* 191-219.

158. Nunnally, J.C., (1978). *Psychometric Theory*. New York: McGraw – Hill.

159. Oram, D., Headon, M., (2002). Ethics, cultural differences, and the new economy. H. Vomácková (ed.), *Economic chances for the 3rd millennium (conference),* Liberec, Czech Republic, 11-12 September 2001, 70-73.

160. O'Toole, R.J.W. and O'Toole, E. F., (1966). Top Executive Involvement in the EDP Function. *PMM and Co Management Controls (June),* 125-127.

161. Papalexandris, N., Kramar, R., (1997). Flexible working patterns: Towards reconciliation of family and work. *Employee Relations, 19(6),* 581 – 595.

162. Pinder, J., Wilkinson, S. J., Demack, S., (2003). A Method for Evaluating Workplace Utility. *Property Management 21(4),* 218 – 229.

163. Pinsonneault, A., Kraemer, K.L., (1993). The Impact of Information Technology on Middle Managers. *MIS Quarterly 17(3),* 271-292.

164. Pooley, R., (2005). When Cultures Collide. *Management Services 49(1),* 28-31.

165. Porter, L. M., Allen, R. W., Angel, H. L., (1983). The Politics of Upward influence in organizations. *Organizational Influence Process*, 408- 422.

166. Powell, P. L., Klein, J. H., (1996). Risk Management for Information Systems Development. *Journal of Information Technology 11,* 309-311.

167. Pressman, R., (1997), *Software Engineering: A practitioners' approach, 4th edition,* McGraw-Hill Higher Education.

168. Prytherch, R. (1986). Introduction. In Prytherch, R. (eds.), *Staff Training in Libraries, - The British Experience,* Gower, Aldershot.

169. Radford, K. J., (1978). *The Journal of the Operational Research Society*, Vol. 29(7), pp. 677-682

170. Rainer, R.K. Jr., Synder, C.A., Carr, H.H., (1991), Risk Analysis for Information for Information Technology. *Journal of Management Information systems 8(1),* 129-147.

171. Rao, T V., (2006). Old Pillars of People Management. Indian Management, *Journal of AIMA 45(6),* 74 – 81.

172. Ravichandran, T., (2003). Special issue on component-based software development. *ACM SIGMIS Database 34 (4),* 45-46.

173. Rindskopf, D., Rose, T., (1988). Some Theory and Applications of Confirmatory Second Order Factor Analysis. *Multivariate Behavioural Research 23,* 51 – 67.

174. Robey, D., Farrow, D. L. (1982). User Involvement in Information Systems Development: A Conflict Model and Empirical Test. *Management Science 28(1),* 73-85.

175. Robey, D., Farrow, D. L., Franz, C. R., (1989). Group Process and Conflict in Systems Development. *Management Science 35(10),* 1172-1191.

176. Rochlin, G. I., (1993). Defining “high reliability” organizations in practice: A taxonomic prologue. K. H. Roberts (Ed), *New challenges to understanding organizations*. New York: Macmillan.

177. Ropponen, J., Lyytinen, K., (1997). Can software risk management improve system development: an exploratory study? *European Journal of Information Systems 6(1),* 41-50.

178. Rothenberger, M.A., Dooley, K.J. (1999). A performance measure for software reuses projects. *Decision Sciences 30(4),* 1131--1153.

179. Rothwell, R., M. Dodgson., (1991). External Linkages and Innovation in Small and Medium-sized Enterprises. *R&D Management 21(2),* 125-137.

180. Ruthberg, Z. G., Fisher, B. T., (1986). *Work Priority Scheme for EDP Audit and Computer Security Review.* National Bureau of Standards Technical Report.

181. Saarinen, T., (1990). System Development Methodology and Project Success. *Information and Management 19(3),* 183-193.

182. Sabri S, Prasada B. (1985). Video conferencing systems. *Proceedings of the IEEE 73(4),* 671-88.

183. Sauer, C., (1993). *Support and Support Management in Information Systems Projects: Building a Conceptual Framework.* Paper presented at the 4th Australian Conference on Information Systems, Brisbane, Australia.

184. Sauer, C., Jeffrey D. R., Land. L., Yetton, P. (2000). Transactions on Software Engineering. *IEEE 26(1),* 1 – 15.

185. Sawyer, S., (1998). Software Development: Processes and Performance. *IBM Systems Journal 37(4),* 552-569.

186. Scacchi, W., (1984). Managing Software Engineering Projects: A Social Analysis. *IEEE Transactions on Software Engineering 10(1),* 49-59.

187. Schmidt, R., Lyytinen, K., Keil, M., & Cule, P. (1996). *Identifying Software Project Risks: An International Delphi Study*. Paper presented at the Proceedings of the 17th International Conference on Information Systems, Cleveland, OH.

188. Schmitt, J., Kozar. K., (1978). Management's role in Information System Development Failures: A Case Study. *MIS Quarterly 2,* 7-16.

189. Segars, A.H., Grover, V., (1993). Re-Examining Perceived Ease of Use and Usefulness: A Confirmatory Factor Analysis. *MIS Quarterly 17(4),* 517 – 525.

190. Shaw, J. D., (1998). An Organization level analysis of voluntary and involuntary turn over. *Academy of Management Review 22(1),* 226 – 56.

191. Sherer, S. A., (1994). Measuring Software Failure Risk: Methodology and an Example. *Journal of Systems Software 25(3),* 257-269.

192. Slovic, P., Fischhoff, B., Lichtenstein, S., (1981). Perceived Risk: Psychological Factors and Social Implications. *Proc. R. Soc. Lond 376*, 17 – 34.

193. Smith, H.J., (1996). Information Privacy: Measuring Individuals' Concerns about Organizational Practices. *MIS Quarterly 20(2),* 167 – 196.

194. Standish Group International, (1999). *Chaos: A Recipe for Success.* The Standish Group International.

195. Steiger, J.H., (1990).Structural Model Evaluation and Modification: An Interval Estimation Approach. *Multivariate Behavioural Research 25,*173– 180.

196. Straub, D.W., (1995). Validating Instruments in MIS Research. *MIS Quarterly 13(2),* 146 – 169.

197. Sumner, M., (2000). *Risk Factor in Enterprise Wide Information Management Systems Projects.* Proceedings of the 2000 ACM SIGCPR Conference on Computer personnel research, 180 – 187, April 2000, Chicago, Illinois, United states.

198. Sureshchander, G.S., Rajendran, C., Anantharaman, R.N., (2001). A Holistic Model for Total Quality Service. *International Journal of Service Industry Management 12,* 378 – 412.

199. Tait, P., Vessey, I., (1988). The Effect of User Involvement on System Success: A Contingency Approach. *MIS Quarterly 12 (1),* 91-110.

200. Tanaka, J.S., (1993). Multifaceted conceptions of fit in structural equation models. In: K.A. Bollen and J.S. Long (eds.), *Testing Structural Equation Models.* Sage, Newbury Park, CA.

201. Taylor, S., Todd, P.A., (1995). Understanding Information Technology Usage: A Test of Competing Models. *Information Systems Research 6(2),* 144 – 176.

202. Thacker, R.A. & Yost, (1997). Training students to become effective workplace team leaders, *Team Performance Management – An International Journal 8(¾),* 89-94.

203. Thayer, R. H., Pyster, A., Wood, R. C., (1980). The Challenge of Software Engineering Project Management. *IEEE Computer (August) 13(8),* 51-59.

204. Thomas, R., (2000). *Management of Diversity*, Gabler, Weisbaden.

205. Townsend. P., (2006). Leadership is Personal & Active. *The Strategist-Business Standard, June 27,* 3.

206. Triandis, Harry, C., (1995). Cross Cultural Perspectives on Personality, In R. Hogan, J. Johnson and S. Briggs (eds.). *Handbook of Personality Psychology*. San Diego/London: Academic Press 439 – 464.

207. Ulrich, D., (1997). *Human Resource Champions: The Next Agenda for Adding Value and Delivering Results,* Harvard Business School Press, Boston, MA.

208. VandenHeuval, A., (1993). *When Roles Overlap: Workers with Family Responsibilities.* Australian Institute of Family Studies, Work and Family Unit, Department of Industrial Relations, Monograph No. 14, Melbourne. 15(7), 902-916.

209. Verner, J. M., Cerpa, N., (1997). *The Effect of Department Size on Developer Attitudes to Prototyping*. ICSE 97, Boston MA IJSA.

210. Wallace, L., (1999). *The development of an instrument to measure software project risk .* Thesis (January 1999), Georgia University.

211. Wallace, L., Keil, M., (2000). Software project risks and their effect on outcomes. *Communications of the ACM 47(4),* 68 – 73.

212. Wallace, L., Keil, M., Rai, A., (2004). Understanding Software Project Risk: A Cluster Analysis. *Information and Management 42(1)*, 115 – 125.

213. Walsh, K. R., Schneider, H., (2002). The Role of Motivation and Risk Behaviour in Software Development Success. *Information Research 7(3),* 2002.

214. Walton, R. E., (1989). *Up and Running: Integrating Information Technology and the Organization*. Boston, Massachussets, HBS Press.

215. Weinberg, G. M., (1985). *The Psychology of Computer Programming*. John Wiley & Sons, Inc., New York, NY, 1985.

216. Weiser M., Morrison.J., (1998). Project Memory– Information Management for project teams. *Journal of Management Information Systems 14(4),* 149 – 166.

217. West M A., (1994). *Effective Team Work*. Leicester, UK, British Psychological Society Books

218. Wheaton, B., Muthen, D., Alwin, D., Summers, G., (1977). Assessing Reliability and Stability in Panel Models. In D. Heise (Ed.), *Sociological Methodology*. San Fransisco: Jossey – Bass.

219. Whittaker, B., (1999). What went wrong? Unsuccessful information technology projects. *Information Management and Computer Security 7(1),* 23 – 30.

220. Wiegers, K., (1998). Know your enemy: Software Risk Management. *Software Development 6(10),* 38-42.

221. Willcocks L., Margetts H., (1994). Risk assessment and Information Systems. *European Journal of Information Systems 5(2),* 127 – 138.

222. Williams, T.M., (1995). A classified bibliography of research relating to project risk. *European Journal of Operational Research 85,* 18-38.

223. Wohlin, C., (2002). *Is Prior Knowledge of a Programming Language Important for Software Quality?* ISESE, 27, 2002, International Symposium on Empirical Software Engineering (ISESE'02).

224. Wong, A., (2002). Sustaining Company Performance through Partnering with Suppliers. *International Journal of Quality & Reliability Management 19 (5),* 567.

225. Wright, P., (1988). Issues of Content and Presentation in Document Design. In M.Helander (ed.), *Handbook of Human-Computer Interaction 629-652,* North- Holland, Elsevier Science Publishers, B. V.

226. Zeidner, J., Johnson, C.D., Scholarios, D., (1997). Evaluating Military Selection and Classification Systems in the Multiple Job Contexts. *Military Psychology Vol. 9,* 169-86.

227. Zmud, R.W., (1980). Management of large software development efforts. *MIS Quarterly 4(2),* 45-55.

APPENDIX 1

The instrument used for data collection and the request letter to the respondent.

<u>Request Letter</u>

Dear Sir,

The failure of software development projects is a common occurrence in many organizations around the world. Software development projects are risky endeavors that many companies are undertaking with disastrous results.

To reduce the high rate of failure in software projects, managers need better tools to assess and manage the risks associated with a software development effort. However, before such tools can be developed, a better understanding of the dimensions of software development risk and the risk management practices is required.

At School of Management Studies, Cochin University, I am conducting a research to address these issues and to develop a usable tool for software project risk and risk management assessment. I have reviewed international studies on similar topics and developed a questionnaire.

As an individual who has participated in software development projects you are in a unique position to comment intelligently on risk factors that affect software projects. Your response is very important, as I am only able to administer this survey to a limited number of individuals involved in software development projects. You may be assured of complete confidentiality. The results of this research will ultimately help the profession by creating a usable tool for software project risk assessment and risk management.

The results of the final study will be summarized and be made available to you.

Thank you for your assistance.

Sincerely,

Mr. Sam Thomas
Research Scholar,
School Of Management Studies,
Cochin University of Science & Technology,
Cochin 22
sam@rajagiri.edu
Ph: 98461 52127

This survey is carried out to study the risk and risk management practices associated with software development projects in India. Based on your experience with software development projects, you are identified by your organization as a qualified respondent to participate in this survey.

Please answer these questions on the basis of your last completed project for which you are selected as a representative:

PART A

1. Which of the following best describes your role in the project?
   - Project manager
   - Project leader / assistant project leader
   - Member of the development team
   - Member of the quality assurance team
   - System analyst
   - Member of the implementation team
   - Others [please specify]

2. Which of the following best describes your most recently completed project?
   - Software developed by your organization for internal use in your organization
   - Software developed by your organization for the internal use of the client
   - Software developed for external sale as packaged software by your organization
   - Software developed for external sale as packaged software by the client
   - Other [please specify]

3. The software developed was
   - Business application software
   - Engineering application software
   - System software
   - web application software
   - Other[please specify]

4. If the client was a foreign organization, please specify the country of its location.

5. What was the estimated duration of the project (in calendar months)?

6. In how many calendar months was the project actually completed?

7. How many members were there in the project team?

8. Which country are you stationed at currently
   - a) India b) USA c) UK d) Other ------------- [please specify]

9. What percentage of this project was done on site and offshore

On site % Offshore %

10. By approximately what percentage, if any, did actual costs for the project exceeded originally budgeted costs? ___________%

11. By approximately what percentage, if any, did actual completion time for the project exceeded originally budgeted completion time? __________%

12. Please indicate the extent to which each of the statements accurately characterizes your project. Mark your response as 1 if you Strongly disagree, 2 if Moderately disagree, 3 if Neutral, 4 if Moderately agree, 5 if Strongly agree

	Strongly Disagree 1	2	3	4	Strongly Agree 5
a) The software developed is reliable					
b) The software developed is easy to use					
c) The software developed is easy to maintain					
d) The software developed is portable					
e) The software developed is flexible (can be modified and upgraded in future)					
f) It is easy to test whether the system working correctly					
g) The software developed is well documented					

13. What is the approximate number of employees in your organization?:

14. How do you describe your company?
    a) A locally registered company with domestic business
    b) A locally registered company with international business
    c) Branch / unit of a company with operations across India
    d) Branch / unit of a multinational company
    e) Others

17. The major focus of your company is

    Domestic market International market

18. How old is your company (in years)

19. The major software development activities of your company are related to
    a) Engineering applications
    b) Business applications
    c) Web-based applications
    d) Device drivers
    e) Embedded applications
    f) Others

20. The annual turnover of your company

    g) < 10 crores
    h) 10 – 100 crores
    i) 100 – 500 crores
    j) 500 – 1000 crores
    k) > 1000 crores

21. What are the certifications (such as CMM , ISO) your company have?

22. How many years of experience do you have in the software field ?

23. How many years of experience do you have in the present organization?

24. How many projects have you completed in your career before this project?

25. Your age? ____________

26. Gender? Male Female

27. Educational qualifications:

28. Software certifications you possess:

PART B – RISK FACTORS

Please indicate the extent to which you agree that each of the following statements applied to your project. Your response can be indicated by "√" mark in the appropriate column against each item.

Strongly Disagree	Moderately Disagree	Neutral	Moderately Agree	Strongly Agree
1	2	3	4	5

	Strongly Disagree 1	2	3	4	Strongly Agree 5
1. Top management support was lacking for the project					
2. Adequate time was not spent on various phases of software development (such as coding, testing, documentation)					
3. Client expectations were unrealistic					
4. Clients did not have software experience					
5. Documentation was very poor					
6. Facilities such as video conferencing were unavailable					
7. Frequent conflicts occurred among members of the project team					
8. Frequent shuffling of project team affected productivity					
9. Hardware infrastructure available was poor					
10. Inappropriate development methodology was used in the project					
11. Adequate reference material were not available					
12. Insufficient resources were provided for the project					
13. Large number of links were required to other systems					
14. Members who had developed the system specifications were doing the coding					
15. Performance measurements of individual members was correctly done					
16. Politics in the organization had a negative impact on project					
17. Reward structure for performance was poor					
18. Project goals and objectives were not agreed upon					
19. The project had highly complex requirements					
20. The project leader was inexperienced					
21. The project leadership did not have "people management skills"					
22. The project manager had the freedom to select the project team					
23. The project manager was ineffective					
24. Project manager had multiple projects to manage at the same time					
25. The project planning was very poor					
26. The project progress was not monitored closely					
27. The project required a change in currently used tools and techniques					
28. The project requirements were changed continuously					

29. Project schedules and budgets were continuously revised
30. Project team communication was ineffective
31. The project team had the freedom to select the development platforms and tools
32. Project team members were inadequately trained
33. The project team was a highly diversified group
34. The project was started without proper feasibility studies
35. Resource requirements were incorrectly estimated
36. Responsibilities for project assignments were not clearly defined
37. Staff motivation was very low
38. Subcontractors were not meeting their commitments
39. The team faced cross cultural issues in working for a foreign client
40. Team member turnover (members resigning) was very high
41. Team members lacked communications skills in English
42. Team members were mostly inexperienced
43. Team members were not familiar with the type of application being developed
44. The corporate environment in the organization was not professional
45. The offshore team did not fully understand the priorities of the on site team
46. The procedures prescribed by quality standards were not strictly followed
47. The project had a clearly identified client / sponsor
48. The project involved modification of an existing software
49. The project necessitated working on outdated technologies
50. The project was over dependent on a few key people
51. The telecommunication network was slow and unreliable
52. The work pressure was so high that most of the employees had to work beyond the office hours
53. There were lot of communication gaps when out on an onsite assignment
54. There was a lack of cooperation from clients
55. There were conflicts among client representatives
56. There were restrictions on working hours for women members
57. This was one of the largest projects attempted by the organization
58. Too many external agencies were involved in the development project
59. Visa rejections to foreign countries was a major risk
60. Women members had restrictions while traveling and staying outside

PART C – RISK MANAGEMENT PRACTICES

Using the following scale, please read through the list of statements that follow and indicate with a checkmark ("√") the extent to which each of the following statements accurately applied to your project

Strongly Disagree	Moderately Disagree	Neutral	Moderately Agree	Strongly Agree
1	2	3	4	5

	Strongly Disagree 1	2	3	4	Strongly Agree 5
1. Individuals were held accountable for the tasks assigned to them					
2. Adequate training is given to employees to make them competent					
3. An assistant project manager / leader was appointed for the project					
4. Attendance was strictly enforced in the organization					
5. Bench marking was applied to ensure best quality software					
6. Compatibility analysis was done for sub-contractors and suppliers					
7. Coordination with the user was ensured through formal procedures					
8. Detailed, multi source cost and schedule estimation was done as part of project planning					
9. Always attempted hide the complexity from the user					
10. Employees were asked to sign bonds to ensure their stay with the organization for a minimum period					
11. Employees were consulted before they were assigned to a project					
12. Formal review of status reports versus plan was made periodically					
13. Formal user specification approval process was followed					
14. HR department was very proactive and helpful					
15. Job-matching was done to ensure that right person gets the right job					
16. Minutes of the project team meetings was prepared and circulated among members					
17. Modifying the existing system was preferred to development from scratch					
18. Once requirements were frozen, no request for change was entertained					
19. The organization had very flexible working hours and focus was on completing the work in time.					
20. Outside technical assistance was sought whenever required					
21. Planning tools were extensively used in the project					
22. Post-project audits are carried out to learn from previous projects					
23. Project leaders were trained in project management techniques					
24. Promotions and salaries were tied to individual performance					
25. Prototyping methodology was used in most of the cases					
26. Regular technical status reviews were conducted					

27. Unnecessary requirements were removed before the development started
28. Risk assessment was performed regularly throughout the project
29. Simulation and scenario analysis was performed to anticipate future problems
30. Software was re-used wherever possible
31. The organization structure was very flat
32. The software was always designed at minimum cost
33. There was an effective configuration management system
34. There were informal contacts and communication channels between project members and the users
35. The user steering committee was very active
36. Users evaluated the progress of the project regularly
37. User manuals were carefully prepared
38. Working beyond office hours was recognized and rewarded

APPENDIX 2

A Note on Self Reporting Methodology Used In the Study

In spite of its limitations with respect to reporting bias, perceptual measures are very commonly used in management research. Perceptual measures are a viable alternative as long as rigorous examinations of validity are performed and multiple items are used (Ketokivi & Schroeder). Presence of risk items and use of risk management strategies in a software project are largely internal issues of the project. Hence measuring the perception of the project stakeholders is the best way of characterizing these constructs. The tool used in this study has 60 items measuring risk and 38 items measuring risk management. Quality is measured through 7 items. These measures were subjected to a range of reliability and validity tests.

The best way of measuring the project outcome would be to personally verify the project records by the researcher. But these records are strictly confidential and none of the organizations were willing to disclose these records directly to the researcher. The user could be thought of as another source for collecting data on project outcome. But again, most of the companies refused to part with contact details of the user. Also, most of the software development in India happens for foreign clients and hence a survey on the user was practically very difficult. Hence the next best way was to request the respondent to check with the records/concerned authorities and then commit an answer.

As seen in the MTMM tests, the high level of agreement between the user representative and the project representative is another reason why the data was collected from the project representative only.

A single respondent or informant was used in this study. Although it has been suggested that the administration of a single instrument to a single informant to simultaneously measure both independent and dependent variables may result in bias (Venkatraman & Ramanujarn, 1987), it is common to use a single respondent in academic research (Pinsonneault & Kraemer, 1993).

For administrative reasons it was not possible to obtain a large number of multiple respondents from the same project. The MTMM tests also showed high consensus among project representatives. Measures based on perceptions of single respondent have become particularly prominent in literature. As mentioned in literature review, the measurement model for risk and project outcome is taken from Wallace (1999). Barki (1993) had developed and validated an instrument to measure risk using self-reporting methodology. The famous Nidumolu study (1996) and its replications in other countries (Na et. al., 2006) linking project performance to risk used self-reporting from project participants. There are many refereed studies such as Jiang and Klein (2000), Ropponen and Lyytinen (2000) and Deephouse et al (1996) who have developed their arguments based responses from a single informant a survey among project members collecting their perception on presence of risk, risk management and the project outcome.

Furthermore, the researcher has taken measures such as scale reordering which can minimize the common method bias (Podsakoff & Organ, 1986). Harman’s one factor test is an indicator of common method variance. In an exploratory factor analysis, if all the variables under a construct load on to one factor or if the factors are too general to be named, high level of common method variance can be assumed. This was not applicable in this research as seen in the factor structure.

****