Managing risk in software development projects: A case study

Article *in* Industrial Management & Data Systems · March 2007

DOI: 10.1108/02635570710723859 · Source: DBLP

Prasanta Kumar Dey
Aston University

Stephen O. Ogunlana
Heriot-Watt University

The current issue and full text archive of this journal is available at
www.emeraldinsight.com/0263-5577.htm

Managing risk in software development projects: a case study

Prasanta Kumar Dey
Operations and Information Management, Aston Business School, Aston University, Birmingham, UK
Jason Kinch
First Caribbean Bank, Barbados, and
Stephen O. Ogunlana
Asian Institute of Technology, Bangkok, Thailand

Abstract

Purpose – The main objective of the paper is to develop a risk management framework for software development projects from developers' perspective.

Design/methodology/approach – This study uses a combined qualitative and quantitative technique with the active involvement of stakeholders in order to identify, analyze and respond to risks. The entire methodology has been explained using a case study on software development project in a public sector organization in Barbados.

Findings – Analytical approach to managing risk in software development ensures effective delivery of projects to clients.

Research limitations/implications – The proposed risk management framework has been applied to a single case.

Practical implications – Software development projects are characterized by technical complexity, market and financial uncertainties and competent manpower availability. Therefore, successful project accomplishment depends on addressing those issues throughout the project phases. Effective risk management ensures the success of projects.

Originality/value – There are several studies on managing risks in software development and information technology (IT) projects. Most of the studies identify and prioritize risks through empirical research in order to suggest mitigating measures. Although they are important to clients for future projects, these studies fail to provide any framework for risk management from software developers' perspective. Although a few studies introduced framework of risk management in software development, most of them are presented from clients' perspectives and very little effort has been made to integrate this with the software development cycle. As software developers absorb considerable amount of risks, an integrated framework for managing risks in software development from developers' perspective is needed.

Keywords Software engineering, Risk management

Paper type Research paper

1. Introduction

Software development projects on implementation provide information to support operations, management analysis and decision-making within an organization (Wang, 2001; Yang, 2001). However, they are vulnerable from time and cost overrun along with quality under-achievement. Additionally, presence of high level of bugs

Industrial Management & Data Systems
Vol. 107 No. 2, 2007
pp. 284-303

0263-5577
DOI 10.1108/02635570710723859

during initial period of trial and commercial use is not uncommon. Although managers claim that they manage the above issues quite efficiently, but there are evidences of lack of management of software development even by leading software developers. Software development projects suffer from market risk, financial risk, and technical risk (Schwalbe, 2002). The software developers must have favourable answers to the following questions in order to achieve success. Whether the developed software fulfils the customers' demand/requirement? How much competition it is likely to face? Whether benefits from the software surpass the cost of development? Is the project technically feasible? Will hardware, software, and networks function properly? Will the technology be available in time to meet project objectives? Is there any chance of the technology becoming obsolete before use? Will security system work throughout its life? There are examples of high profile IT project failure in the literature (Baccarini *et al.*, 2004). Evidences indicated that risks have not been managed effectively (Hedelin and Allwood, 2002).

Although some managers claim that they manage risk in their projects, there are evidences of not managing risks systematically. The managers quite often address technical risks. However, they seldom deal with market and financial risks, which are vital for a successful software development. Hence, there is a need for integrated risk management.

The success of software development depends on the criteria: functionality, quality and timeliness. Software is developed to perform a specific function. Unless it is successful in performing this function effectively, the purpose of the software development will be defeated. Customers should be delighted on the performance of the product. The software should be delivered to the customer on time (as scheduled). Quite often, a penalty clause is associated with the delivery of the software, as the client is expected to face substantial business setback if they cannot bring change in their system on time. However, project delay not only incurs cost due to penalty, but also there is always the chance of increased prices of materials and services with time (price escalation), loss of image, and incurring opportunity cost.

Risks are part and parcel of projects (Dey and Ogunlana, 2004). Software development projects are not different, as project planning is done with minimum information. However, the degree of risk varies with complexity, size (both in terms of schedule and budget), and location. Scope creep, lack of understanding of problems, ambiguous requirements, lack of resources, hardware, networking, and security issues are some of the common risk elements in software development projects. Therefore, there is a need to manage risk in software development. Although researchers and professional have written on risk management in software development (Keil *et al.*, 1998), very little work have been done in order to involve all the concerned stakeholders in managing risk and integrating the risk management process with a holistic project management approach (software development cycle). Various frameworks of software risk management have been suggested, some of which deal with only the application part and some models deal with risk holistically covering application, organization and inter-organization levels (Bandyopadhyay *et al.*, 1999). However, there is a dearth of literature on software development risk management from developers' perspective. This study presents a risk management framework from developers' perspective, which integrates the software development cycle and involves the concerned stakeholders. Accordingly, the objective of the study is to develop an integrated risk management framework for software development projects and apply the framework to a software development project in the Barbados government sector in order demonstrate its effectiveness.

2. Literature review

Risk refers to future conditions or circumstances that exist outside of the control of the project team that will have an adverse impact on the project if they occur. In other words, whereas an issue is a current problem that must be dealt with, a risk is a potential future problem that has not yet occurred.

Successful projects try to resolve potential problems before they occur. This is the art of risk management. A reactive project manager tries to resolve issues when they occur. A proactive project manager tries to resolve potential problems before they occur. Not all issues can be seen ahead of time, and some potential problems that seem unlikely to occur, may in fact occur. However, many problems can be seen ahead of time. They should be resolved through a proactive risk management process.

Chapman and Cooper (1983), defines risk as "exposure to the possibility of economic or financial loss or gains, physical damage or injury or delay as a consequence of the uncertainty associated with pursuing a course of action." The task of risk management can be approached systematically by breaking it down to the following three stages:

(1) risk identification;

(2) risk analysis; and

(3) risk responses.

Tummala and Leung (1999) developed a methodology for risk management governing risk identification, measurement, assessment, evaluation and risk control and monitoring. They applied the methodology for managing cost risk for an EHV transmission line project.

Williams (1995) reviewed the various researches in project risk management. He described various risk identification and analysis tools being used by researchers and practitioners. Finally, the management structures and procedures needed to manage risk are discussed in his work.

Turner (1999) suggested expert judgment, plan decomposition, assumption analysis, decision drives and brainstorming for identification of risk factors effectively in a project. Perry and Hayes (1985), suggested a checklist of risk that may occur throughout the life span of any project. The Delphi technique has been used by Dey (1999) to identify risk factors. Outside the field of engineering and construction, an approach for risk identification in product innovation has been reported by Halman and Keizer (1998).

Table I summarises the applications of various qualitative and quantitative tools and techniques in risk analysis.

Most of the analyses done so far are centred on analyzing the duration of the project. The management is interested in two aspects; the total duration and which activities are critical in determining that duration. Many authors have presented the distribution of time duration of activities as classical β distribution (Farnum and Stanton, 1987). Benry (1989) proposed his own distributions for practical simulations.

Recently, a number of systematic models have been proposed for use in the risk-evaluation phase. Kangari and Riggs (1989) classified these methods into two

Table I. Summary of risk analysis tools and techniques

Method	Keynotes	Application and previous study: Who and when	Topic
Influence diagram	> Risk identification > Brain storming and Delphi technique > Relationship of variables	Ashley and Bonner (1987)	Identification of political risks in international project
		Yingsutthipun (1998)	Identification of risks in transportation projects in Thailand
Monte Carlo simulation (MCS)	> Distribution form > Variables' correlation	Songer *et al.* (1997)	Debt cover ratio (project cashflow) in a tollway project
		Chau (1995)	Distribution form for cost estimate
		Wall (1997)	Distribution form and correlation between variables in building costs
		Dey and Ogunlana (2001)	Project time risk analysis through simulation
PERT	(Same as above) > Network scheduling	Hatush and Skitmore (1997)	Contractor's performance estimate for contractual purpose
Sensitivity analysis	> Deterministic > Variables' correlation	Yeo (1990, 1991)	Probabilistic element in sensitivity analysis for cost estimate
		Woodward (1995)	Survey on use of sensitivity analysis in BOT project in UK
MCDM	> Multi-objective > Subjectivity	Moselhi and Deb (1993)	Project alternative selection under risk
		Dozzi *et al.* (1996)	Bid mark-up decision making
AHP	> Systematic approach to incorporate subjectivity > Consistency of judgement	Dey *et al.* (1994)	Risk analysis for contingency allocation
		Mustafa and Al-Bahar (1991)	Risk analysis for international construction project
		Zhi (1995)	Risk analysis for oversea construction project
		Nadeem (1998)	Risk analysis for BOT project in Pakistan
Fuzzy set approach (FSA)	> Vagueness of subjective judgement	Kangari and Riggs (1989)	Risk assessment by linguistic analysis
		Diekmann (1992)	Combination of influence diagram with fuzzy set approach

(continued)

Method	Keynotes	Application and previous study Who and when	Topic
		Lorterapong and Moselhi (1996)	Network scheduling by fuzzy set approach
		Paek *et al.* (1993)	Risk pricing in construction project through fuzzy set approach
Neural network approach (NNA)	> Implicit relationship of variables	Chua *et al.* (1997)	Development of budget performance model
		Boussabaine and Kaka (1998)	Cost flow prediction in construction project
Decision tree	> Expected value	Haimes *et al.* (1990)	Multi-objective decision tree
Fault tree analysis	> Accident analysis > Safety management	Tulsiani *et al.* (1990)	Risk evaluator
Risk checklist	> From experiences	Perry and Hayes (1985)	Risk and its management in construction project
Risk mapping	> Two dimensionality of risk	Williams (1996)	Two dimensionality of project risk
Cause/effect diagram	> Risk identification	Dey (1997)	Symbiosis of organizational reengineering and project risk management for effective implementation of projects
Delphi technique	> Subjectivity	Dey, 1997	Same as above
Combined AHP and Decision tree	Probability, severity and expected monitory value	Dey (2001c)	Decision support system for risk management

Source: Dey and Ogunlana (2004)

Table I.

categories: classical models (i.e. probability analysis and Monte Carlo simulation), and conceptual models (i.e. fuzzy-set analysis). They noted that probability models suffer from two major limitations. Some models require detailed quantitative information, which is not normally available at the time of planning, and the applicability of such models to real project risk analysis is limited, because agencies participating in the project have a problem with making precise decisions. The problems are ill-defined and vague, and they thus require subjective evaluations, which classical models cannot handle. Mustafa and Al-Bahar (1991) and Dey *et al.* (1994) used analytic hierarchy process, a multiple attribute decision-making technique, for risk analysis of construction projects with the involvement of the concerned stakeholders.

Various researchers have reported on risk of software development (Alter and Ginzberg, 1978; Barki *et al.*, 1993; Boehm and Ross, 1989; Boehm, 1991; Charette, 1989; Johnson, 1995; Jones, 1994; McFarlan, 1981). Unfortunately, much of what has been written on risk is based either on anecdotal evidence or on studies limited to a narrow portion of the development process (Keil *et al.*, 1998). Additionally, risks were not identified by involving the process operators. Although Boehm and Ross (1989) and McFarlan (1981) prioritised the identified risks and provided some meaningful classification, but did not evolve any strategy for risk mitigation. Boehm's (1991) work was probably the most significant as he identified the top 10 list of software risk items using his experience in the defence industry. Keil *et al.* (1998) developed a framework for systematic risk classification and strategy development, but did not link it with the software development cycle. Baccarini *et al.* (2004) identified and prioritised IT project risks through empirical research and suggested possible responses, but did not provide any framework for software risk management.

The methods for managing risk in software development currently available are not comprehensive (Bandyopadhyay *et al.*, 1999) as they deal with specific types of risk. As informed by Bandyopadhyay *et al.* (1999), Vitale (1986) has proposed a framework for identifying the strategic risks of IT, Rainer *et al.* (1991) have proposed a risk analysis process for IT by combining qualitative and quantitative methodologies, and Epich and Person (1994) have proposed a disaster recovery plan to reduce IT risks by methodically resituating business functions in the event of a disaster. Eloff *et al.* (1993) addressed the issue of risk monitoring to ensure effective implementation of risk control measures. Huang *et al.* (2004) introduced a risk prioritising method using analytic hierarchy process for enterprise resource planning implementation, which is a combination of software development and process reengineering project. Although the suggested framework considers both qualitative and quantitative factors and involves the concerned stakeholders, it presents the entire problem from the clients' perspectives. Therefore, there is need for analysing the risk management issues in software development from the developers' perspective, with the involvement of the stakeholders, with the consideration of both qualitative and quantitative risk factors and integrating the risk management process with the software development cycle (project management).

This study develops an integrated framework for managing risk in software development with the involvement of the concerned stakeholders and applies the framework to an organization within the Government of Barbados.

3. The risk management framework

Figure 1 shows the proposed risk management framework. It consists of the following steps:

(1) *Analyzing functional requirements*. Software performs organizational functions in an integrated way. Hence, a strong requirement analysis with the involvement of functional people is required for its success. As software applications often call for change in business processes, process reengineering is quite common while establishing functional requirements and selecting information technology solutions.

(2) *Establishing scope of software development project and developing work breakdown structure*. Functional requirement analysis along with information system design establishes software development project scope. Classification of the entire scope of the project in various modules and sub-modules in a hierarchical structure leads to formation of work breakdown structure.

(3) *Identifying risky work packages*. The work packages which are vulnerable to time, cost, and quality targets failures are then identified with active involvement of both functional and IT executives.

(4) *Identifying risk events*. Risk events are then identified for each risky work package using various tools and group consensus.

(5) *Analyzing risk*. Probability and severity of the risk events are analyzed using qualitative and quantitative tools with the active involvement of the stakeholders.

(6) *Developing risk management plan*. A risk management plan is then develop to in response to adverse risk events before they occur. The plan is evaluated with respect to contribution in reducing the effect of risk. Risk responses are implemented if they have potential to reduce project risk substantially. Actual project is then implemented.

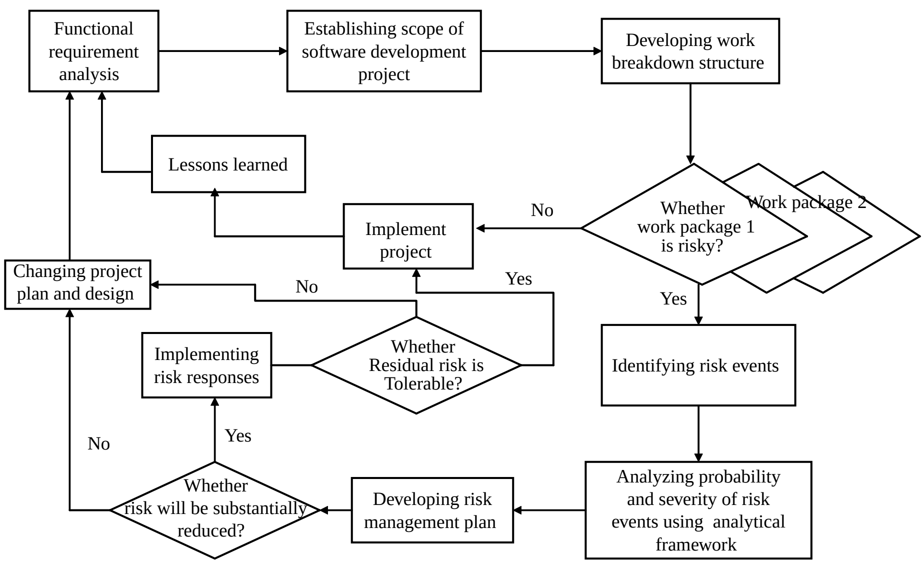

Figure 1.
Risk management model for software development

(7) *Controlling risk*. Risk management plan suggests various strategies for all likely risk events depending on the probability and severity along with the perceptions of the stakeholders. Therefore, a dynamic control mechanism needs to be established so as to make faster decisions when any risk event occurs.

The above steps are explained with a case example of software development project in the public sector in Barbados.

4. The application

The Town and Country Planning Office (TCPO) is the government agency that regulates building construction in Barbados. This department receives up to 3,000 applications a month for permission to construct buildings. Currently an application to build takes between three months to three years to reach the approval stage. This is partly due to inadequate application tracking procedures. The department planned a software project for application tracking management system. The estimated cost and planned duration of the project are $400,000 and 12 months, respectively. The contract had the following stipulation (Town and Country Planning Application Processing System Tender Document, 2001):

> In the case of a delay of more than one month the supplier shall be liable to pay damages calculated from the expiry of the contractual period for each day the delay lasts, such damages to be fixed at 1/1,000 of the value of the undelivered software per month.

4.1 Requirement analysis

A detailed requirement analysis was done using business process reengineering framework (Dey, 2001a) with the involvement of both functional as well as IT people of the TCPO.

4.2 Scope and work breakdown structure

The detailed requirement analysis helped in deriving the project scope with the active involvement of owner's project group and the software developer's representatives. The entire scope of the project was classified to form a hierarchical structure (work breakdown structure). The project had the following work packages:

- data conversion of existing data;
- reception and application receipt module;
- registry module;
- drawing office module;
- planning module;
- integration; and
- training and documentation modules.

Other than "data conversion of existing data," "integration," and "training and documentation" work packages, each work package had the following common activities:

- design;
- coding; and
- testing.

The “data conversion of existing data work package” had “data design,” “database creation,” “data transfer,” and “testing” activities. The work package, “training and documentation” had “design,” “implementation,” and “evaluation” activities.

Figure 2 shows the work breakdown structure of the project under study.

4.3 Identifying risky work packages

The project people decided to analyze risk at the project level instead of work package level due to the nature of the project (concise and small).

4.4 Identifying risk events

Risk events do not allow project to achieve its goals. Various qualitative and quantitative tools can be applied to identify potential risks of any project (Dey, 2002).

The risk events for the project under study were identified through brainstorming among experienced executives of the owner (TCPO) and the developer. The identified risks were as follows:

- *Incorrect requirements/specification*. The requirements phase of a software project is one of the most crucial phases. If the requirements are not exact then the application will not meet the needs of the users.
- *Incompatible development environment*. The language used to develop the software does not easily lend itself to the development of the particular application.
- *Inadequate design*. The design of the database and or the data structures does not adequately cover all the data to be processed by the system.
- *Loss/lack of resources*. The Loss or unavailability of key personnel during the course of the project.
- *Unavailable customer contact*. Effective communication between client/customers and developer throughout the software development helps in achieving success.
- *Scope Creep*. The requirements of the project are continually added to thus causing scope creep.

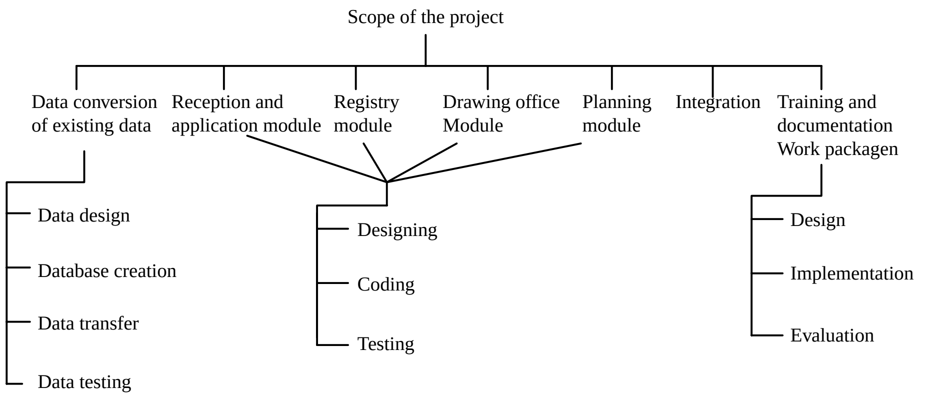

Figure 2.
Work break down structure of the software development project under study

- *Problems in coding and unit test*. Quality of coding is judged by testing. Various types of testing are planned to ensure quality in software development depending on heir characteristics. However, if the programmers, who are developing the codes, are held responsible for testing, it is difficult to properly check the quality. Hence, it is always suggested to establish a different group for unit testing. This usually does not occur in software development projects, because the team leader on a project reviews all codes and establish coding guidelines and standards.

4.5 Analyzing risks

Risk analysis is the process of evaluating risks to assess the range of possible project outcomes. This helps the project manager to develop an effective risk management plan. Various qualitative and quantitative tools and techniques are currently employed to analyze risk. This study adopts risk-mapping method (Dey, 2001b) to determine probability and severity of identified risk events. Table II shows the risk map for the project understudy. The risk map was developed through brainstorming session and group consensus among functional and IT executives of TCPO and the executives of software developer.

The analysis reveals the following:

Loss or unavailability of key personal during the course of the project was found to have the highest likelihood of occurrence and severity. The employee turn over during the project will have tremendous negative impacts, as it is extremely difficult to get competent experienced technical persons within a short period and moreover, it takes time for them to adjust in a new environment. These have negative impact on productivity of software development projects.

Incorrect/incomplete requirements/specification was found to be the next most crippling event on the software development project under study because the requirements/specifications are like the foundation of a building. Incomplete or incorrect specifications will mean either recoding a complete section or making serious modifications to already written software.

Inadequate design, unavailable customer contact and scope creep were found to have a medium probability with scope creep having the greatest degree of severity, followed

	Very high					Loss/lack of resources
	High			Scope creep	Incorrect requirements or specifications	
Severity	*Medium*		Incompatible development environment	Inadequate design		
	Low					
	Very low	Code and unit test		Unavailable customer contact		
		Very low	*Low*	*Medium* *Probability*	*High*	*Very high*

Table II. Risk severity to probability factor matrix by inadequate design, which had a medium severity followed by customer contact with very low severity. The electronic communications modes can be effectively used to re-establish customer contacts. However, if there is scope creep, it would be difficult to complete the project within time and budget. Chances of occurrence of inadequacy in design and its severity is medium, as the project executives feel that they would be able to address the design issues as and when required with their existing project team.

Incompatible development environment risk was not prominent in this project, as the project has experienced team who identified all technical requirements of the project.

The coding and unit testing were also not very risky activities, as experienced owner and developers were involved. The project executives felt that there were very low chances of occurrence of any problem for these activities and if at all they occurred, the experienced developer could easily handle them.

The risk events were given a severity/probability rating based on Table III.

Severity probability factor rating (SPR) in Table IV is used to determine the strategy that will be used to approach a particular risk. This intersecting matrix of risk severity and probability provides a simple straightforward way to numerically quantify risk. More sophisticated numerical analysis techniques are not required to establish where resources should be applied to build appropriate risk mitigation strategies and contingency plans (Royer, 2000).

The drawing office work package was found to be the most risky owing to the extremely technical nature. Loss of the human resources during the project would have catastrophic impact on the project. Not only would getting new experienced personnel for this work package be difficult and time consuming, but also training them to perform at the desired level would be an uphill task. The project was likely to face tremendous set back if this risk event occurs.

The overall impact on project is then determined using the schedule impact scale and the contract clause on schedule overrun. Table V shows the schedule impact scale.

Table III.
The ratings

Severity probability factor rating (SPR)	
4	Avoid these risks all together. Recognise them from the start and plan to avoid them from occurring
3	Mitigation strategy and detailed contingency plan
2	Mitigation strategy and outline contingency plan
1	Mitigation strategy
0	Treat as a project assumption

Table IV.
Severity probability factor rating (SPR)

Severity	*Very high*	3	3	3	3	4
	High	2	2	2	3	3
	Medium	1	2	2	2	3
	Low	1	1	2	2	3
	Very low	0	1	1	2	3
		Very low	*Low*	*Medium*	*High*	*Very high*
				Probability		

The contract clause for schedule overrun states that in the case of a delay of more than one month the supplier shall be liable to pay damages calculated from the expiry of the contractual period for each day the delay lasts, such damages to be fixed at 1/1,000 of the value of the undelivered software per month.

The project schedule completion was 12 month and budget was $400,000.

Cost of the software per month = cost of development/development period

$$= \frac{400,000}{12} = \$33,333.33$$

The penalty was 1/1,000 of the cost per month for each day delay. Therefore,

$$\text{daily penalty} = \text{cost per month}/1,000 = 33,333.33/1,000 = \$33.33$$

The impact of risk events on each work package is then determined using the information in Tables II, III and V. Table VI shows the impact of risk events on each work package. The overall likely delay was then determined with the following assumptions:

- there will be no cascading effect on overall project schedule in the event of delay in any activity;
- each module would be executed simultaneously;

	Very low	Low	Medium	High	Very high
Schedule	Negligible (<2 days)	<7 days over completion date	<25 days over completion date	>25 days <35 days over completion date	>50 days over completion date

Note: Completion date does not include the acceptable 1-month grace period allowed by the Town and Country Planning Department

Table V. Impact scale

Risk	Data conversion of existing data (day)	The reception and application receipt module (day)	Registry module (day)	Drawing office module (day)	Planning module (day)	Training and documentation (day)
Incorrect requirements or specifications	3	5	8	10	16	5
Scope creep	3	4	6	20	11	4
Loss/lack of resources	2	4	3	55	25	0
Inadequate design	6	8	6	18	5	6
Incompatible development environment	1	6	6	4	5	5
Unavailable customer contact	0	3	4	6	5	5
Code and unit test	0	1	2	2	5	6
Total	15	31	35	115	72	31

Table VI. Impact on schedule of risk occurrence

- data conversion, module implementation, and training are executed sequentially;
- risk events are independent; and
- all the figures in Table VI are expected time overrun.

Accordingly, the expected time overrun was 161 days (44 percent of original schedule) and expected cost overrun was $4366.23 (the figure does not include any amount other than penalty/loss).

The result revealed that a significant time overrun was anticipated for the project under study. Although there was comparatively less cost implication, the delay would incur significant intangible cost like losing of image, opportunity cost, etc.

4.6 Developing risk management plan

Risk analysis lead to deriving a few effective risk responses in line with the principles like avoidance, transferring, reduction, and absorbing. Hoffman (2001) states the following mitigation strategies to handle the most prominent software risks:

- model functional requirements;
- have each project team member *au fait* with all aspects of the project;
- use software modelling tools to assist in the design phase;
- utilise internet technologies to stay in contact, e.g. e-mail, project web site;
- implement a scope management plan;
- research all limitations of development environment and compare with software requirements; and
- have a software inspection process and ensure independent testing is done using strong test cases.

Based on the SPR for the potential risk and the impact on the schedule and budget of their occurrence the following strategies (Table VII) were recommended.

The risk plan (Table VIII) for the project under study is then derived based on the SPR and the above risk strategies.

4.7 Controlling risk

Risk management planning develops a detailed strategy for risk responses depending on the nature of likely risks. Another round of brainstorming session was conducted to determine cost – benefits of actions against each risk event. Accordingly, risks responses were implemented. Another round of brainstorming was done to determine whether the residual risk is tolerable before implementation. To control risk in the project under study, a small group was formed with representatives from both owner (TCPO) and developer. The group worked very closely with the project monitoring and control group. They maintained a risk register to monitor each risk event along with the implementation progress of each work package. This register helped in making various decisions across the project phases.

The software development project under study was completed within scheduled time and budget. There was no resource related issue throughout the project. This was mainly due to proactive actions during authorizing the project team for software development. Although there was scope creep, the scope management plans with the involvement of

Risk events	Risk strategy	Strategy type
Loss/lack of resources	Loosing resources at critical points of software development has high negative impact. This can be avoided by ensuring that project team members au fait with all aspects of the project	Avoidance
Scope creep	Scope creep plagues most projects (especially software development) and causes time and cost overrun. However, scope creep is sometimes needed to address quality issues in software development. A dynamic scope management plan with involvement of client will improve project performance	Transference/reduction
Incorrect requirements or specifications	Incorrect specifications of software projects are major issues in managing software development projects, as clearly establishing client's requirements is not always an easy task. Technique like process reengineering and benchmarking can be employed for deriving the requirements with the active involvement of functional people of the client's organizations	Transference/reduction
Inadequate Design	This type of problem is present in any software development project. Use software modelling tools helps reduce the effect of inadequate design.	Reduction
Incompatible development environment	Researching all limitations of development environment before implementation helps reducing risk drastically	Reduction
Unavailable customer contact	Client's involvement in software development is one of the most important factors of success. Effective communication between developer and client can be maintained using information technology, e.g. e-mail, web site	Transference/reduction
Problems in coding and unit test	Developing a software inspection process and ensuring independent testing can reduce the problems related to coding and unit test	Reduction

Table VII. Strategies against risk events

client resolved the matter very fast, as the time and cost implications were studied and decisions were made accordingly. Changes in specifications and designs were made in various modules with the consideration of time and cost implications with the involvement of the client. Hence, although specifications and designs were changed from time to time to improve quality, there were little time and cost overrun. Software development environment was appropriately selected before implementation. The experience of the client and contractor/developer helped to achieve this before project work started. A communication infrastructure using web-based technology was established to integrate the effors of stakeholders. It helped tremendously to appraise the project progress as well accelerating decision making. The coding and unit testing were trouble free, as third party inspection was organized for testing.

5. Conclusion

The following are the general benefits that can be achieved from the application of risk management in projects (Perry and Hayes, 1985):

Table VIII.
Risk management plan

Risk event	Impact/s on project	L	C	R	Risk treatment(s)	Key Responsibility	Project phase
Loss/lack of resources	Project stalls while new resources are acquired and trained. Impacts schedule and budget.	4	5	4	Have each project team member au fait with all aspects of the project	Project technical lead	All phases
Scope creep	Schedule and budget over-runs	5	3	3	Develop a scope management plan with the involvement of Client	Project manager	Planning
Incorrect requirements or specifications	Impacts quality, budget and schedule	4	4	3	Model functional requirements analysis using process reengineering and benchmarking	Project manager, project technical lead	Planning
Inadequate design	Impact on quality	3	3	2	Use software modelling tools	Project technical lead	Design
Incompatible development environment	Impact quality	2	3	2	Research all limitations of development environment and compare with software requirements	Project technical lead	Planning, design, development
Unavailable customer contact	Impact quality	3	2	2	Utilise internet technologies to stay in contact, e.g. e-mail, project web site		All
Problems in coding and unit testing	Impact quality				Have a software inspection process and ensure independent testing is done using strong test cases		Development, testing

Notes: L, likelihood: 5, almost certain; 4, likely; 3, possible; 2, unlikely; 1, rare; C, consequence: 5, severe; 4, major; 3, moderate; 2, minor; 1, negligible; R, risk level: 4, extreme; 3, high; 2, medium; 1, low

- the issue/problems of the project are clarified understood and allowed for right from the start;
- decisions are supported by thorough analysis of available data;
- the structure and definition of the project are continually and objectively monitored;
- contingency planning allows prompt, controlled and pre-evaluated responses to risk that materialize;
- clearer definitions of the specific risk associated with a project;
- it builds-up of a statistical profile of historical risk to allow better modelling for future projects;
- it encourages problem solving and providing innovative solutions to the risk problems within a project; and
- it provides a basis for project organization structure and appropriate responsibility matrix.

Effective risk management in software development ensures successful accomplishment of projects with customers' satisfaction, functional achievement, and overall better financial performance of the organizations. Managing risk dynamically throughout the project phase will ensure user/customer/client involvement, management commitment, clear specification and design, appropriate planning, realistic expectations, competent and committed staff, and clear vision and objectives.

Like any other project, software development has inherent risks of not achieving its objectives. Therefore, a risk management plan along with other work plan is absolutely necessary in order to achieve time, cost and quality of the project. Although there are numerous tools and techniques for managing risks (identifications, analysis, developing responses, and controlling) in projects, effectiveness of management depends on developing a framework of risk management, integrating it with the project management cycle and an institutional framework for its practice. Risk management requires involvement of stakeholders in interactive ways, as experience is the best means for managing risk along with a quantitative framework. Risk management should also be integrated with the decision-making processes in managing projects, as risk management reveals the rationales for making appropriate decisions.

The proposed risk management framework has the following advantages:

- It provides an analytical framework for managing risk from software developer perspectives.
- It involves all the concerned stakeholders for risk analysis and deriving responses.
- It takes both subjective and objective approach to derive specific responses for managing risk.
- It is totally integrated with the software development cycle. Hence, its practice can be easily institutionalized.
- The approach to risk management is user friendly and not complex.

Similar approach may be adopted not only to manage any type of information technology project, but also for projects across various types of industries.

References

Alter, S. and Ginzberg, M. (1978), "Managing uncertainty in MIS implementation", *Sloan Management Review*, Vol. 20 No. 1, pp. 23-31.

Ashley, D.B. and Bonner, J.J. (1987), "Political risks in international construction", *Journal of Construction Engineering and Management*, Vol. 113 No. 3, pp. 447-67.

Baccarini, D., Salm, G. and Love, P.E.D. (2004), "Management of risks in information technology projects", *Industrial Management and Data System*, Vol. 104 No. 4, pp. 286-95.

Bandyopadhyay, K., Mykytyn, P.P. and Mykytyn, K. (1999), "A framework for integrated risk management in information technology", *Management Decision*, Vol. 37 No. 5, pp. 437-44.

Barki, H., Rivard, S. and Talbot, J. (1993), "Toward an assessment of software development risk", *Journal of Management Information System*, Vol. 10 No. 2, pp. 203-25.

Benry, J. (1989), "A new distribution function for risk analysis", *Journal of the Operational Research Society*, Vol. 40, pp. 1121-7.

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

Boehm, B. and Ross, R. (1989), "Theory-W software project management: principles and examples", *IEEE Transactions on Software Engineering*, Vol. 15 No. 7, pp. 902-16.

Boussabaine, A.H. and Kaka, A.P. (1998), "A neural networks approach for cost flow forecasting", *Construction Management and Economics*, Vol. 16, pp. 471-9.

Chapman, C.B. and Cooper, D.F. (1983), "Risk analysis: testing some prejudices", *European Journal of Operational Research*, Vol. 14, pp. 238-47.

Charette, R.N. (1989), *Software Engineering Risk Analysis and Management*, Intertext, New York, NY.

Chau, K.W. (1995), "The validity of the triangular distribution assumption in Monte Carlo simulation of construction costs: empirical evidence from Hong Kong", *Construction Management and Economics*, Vol. 13, pp. 15-21.

Chua, D.K.H., Kog, Y.C., Loh, P.K. and Jaselskis, E.J. (1997), "Model for construction budget performance-neutral network approach", *Journal of Construction Engineering Management, September*, pp. 214-22.

Dey, P.K. (1997), "Symbiosis of organizational reengineering and risk management for effective implementation of projects", Doctoral thesis, Jadavpur University, Calcutta.

Dey, P.K. (1999), "Process re-engineering for effective implementation of projects", *International Journal of Project Management*, Vol. 17 No. 3, pp. 147-59.

Dey, P.K. (2001a), "Reengineering materials management: a case study on Indian refinery", *Business Process Management Journal*, Vol. 7 No. 5, pp. 394-408.

Dey, P.K. (2001b), "Integrated approach to project feasibility analysis: a case study", *Impact Assessment and Project Appraisal*, Vol. 19 No. 3, pp. 235-45.

Dey, P.K. (2001c), "Decision support system for risk management: a case study", *Management Decision*, Vol. 39 No. 8, pp. 634-48.

Dey, P.K. (2002), "An integrated assessment model for cross-country pipelines", *Environmental Impact Review*, Vol. 22, pp. 703-21.

Dey, P.K. and Ogunlana, S.O. (2001), "Project time risk analysis through simulation", *Cost Engineering*, Vol. 43 No. 7, pp. 24-32.

Dey, P.K. and Ogunlana, S.O. (2004), "Selection and application of risk management tools and techniques for Build-operate-transfer Projects", *Industrial Management & Data Systems*, Vol. 104 No. 4, pp. 334-46.

Dey, P.K., Tabucanon, M.T. and Ogunlana, S.O. (1994), "Planning for project control through risk analysis; a case of petroleum pipeline laying project", *International Journal of Project Management*, Vol. 12 No. 1, pp. 23-33.

Diekmann, J.E. (1992), "Risk analysis: lessons from artificial intelligence", *International Journal of Project Management*, Vol. 10 No. 2, pp. 75-80.

Dozzi, S.P., AbouRizk, S.M. and Schroeder, S.L. (1996), "Utility-theory model for bid markup decisions", *Journal of Construction Engineering and Management*, pp. 119-24, June.

Eloff, J.H.P., Labuschangne, L. and Badenhorst, K.P. (1993), "A comparative framework for risk analysis methods", *Computers and Security*, Vol. 12 No. 6, pp. 597-603.

Epich, R. and Person, J. (1994), "A fire drill for business", *Information strategy: The Executive's Journal*, pp. 44-7.

Farnum, N.R. and Stanton, L.W. (1987), "Some results concerning the estimation of beta distribution parameters in PERT", *Journal of the Operational Research Society*, Vol. 38 No. 3, pp. 287-90.

Haimes, Y.Y., Li, D. and Tulsiani, V. (1990), "Multi objective decision-tree analysis", *Risk Analysis*, Vol. 10 No. 1, pp. 111-29.

Halman, J.I.M. and Keizer, J.A. (1998), "Risk management in product innovation projects", *International Journal of Project and Business Risk Management*, Vol. 2 No. 2.

Hatush, Z. and Skitmore, M. (1997), "Assessment and evaluation of contractor data against client goals using PERT approach", *Construction Management and Economics*, Vol. 15, pp. 327-40.

Hedelin, L. and Allwood, C.M. (2002), "IT and strategic decision-making", *Industrial Management and Data System*, Vol. 102 No. 3, pp. 125-35.

Hoffman, D.C. (2001), *Mitigating Software Development Risk 200*, available at: www.svspin.org/Events/Presentations/Virt_mit_risk.ppt

Huang, S., Chang, I., Li, S. and Lin, M. (2004), "Assessing risk in ERP projects: identify and prioritise the factors", *Industrial Management and Data System*, Vol. 104 No. 8, pp. 681-8.

Johnson, J. (1995), "Chaos: the dollar drain of IT project failures", *Application development trends*, Vol. 2 No. 1, pp. 41-7.

Jones, C. (1994), *Assessment and Control of Software Risks*, Prentice-Hall, Englewood Cliffs, NJ.

Kangari, R. and Riggs, L.S. (1989), "Construction risk assessment by linguistics", *IEEE Trans. Manag.*, Vol. 36 No. 2, pp. 126-31.

Keil, M., Cule, P.E., Lyytinen, K. and Schmidt, R.C. (1998), "A framework for identifying software project risks", *Communications of the ACM*, Vol. 41 No. 11, pp. 76-83.

Lorterapong, P. and Moselhi, O. (1996), "Project-network analysis using fuzzy sets theory", *Journal of Construction Engineering and Management, December*, pp. 308-18.

McFarlan, F.W. (1981), "Portfolio approach to information systems", *Harvard Business Review*, Vol. 59 No. 5, pp. 142-50.

Moselhi, O. and Deb, B. (1993), "Project selection considering risk", *Construction Management and Economics*, Vol. 11, pp. 45-52.

Mustafa, M.A. and Al-Bahar, J.F. (1991), "Project risk assessment using the analytic hierarchy process", *IEEE Trans. Eng. Manag.*, Vol. 38 No. 1, pp. 46-52.

Nadeem, A. (1998), "Prospects and risks of BOT projects for infrastructure development in Pakistan", AIT Master's Thesis ST98-35, Bangkok.

Paek, J.H., Lee, Y.W. and Ock, J.H. (1993), "Pricing construction risk: fuzzy set application", *Journal of Construction Engineering and Management*, Vol. 119 No. 4, pp. 743-56.

Perry, J.G. and Hayes, R.W. (1985), "Risk and its management in construction projects", *Proceedings of the Institute of Civil Engineering*, Vol. 78 No. 1, pp. 499-521.

Rainer, R.K., Snyder, C.A. and Carr, H.H. (1991), "Risk analysis for information technology", *Journal of Management information systems*, Vol. 8 No. 1, pp. 129-47.

Royer, P.S. (2000), "Risk management", *PM Network*, pp. 31-9, September.

Schwalbe, K. (2002), *Information Technology Project Management*, Thomson Learning, Scarborough.

Songer, A.D. *et al.*, (1997), "Risk analysis for revenue dependent infrastructure project", *Construction Management and Economics*, Vol. 31, pp. 377-82.

Town and Country Planning Application Processing System Tender Document (2001), General Conditions of Tender section 6:1 Liquidated damages for delay.

Tulsiani, V., Haimes, Y.Y. and Li, D. (1990), "Distribution analyzer and risk evaluator (DARE) using fault trees", *Risk Analysis*, Vol. 10 No. 4, pp. 521-38.

Tummala, V.M.R. and Leung, Y.H. (1999), "Applying a risk management process (RMP) to manage cost risk for an EHV transmission line project", *International Journal of Project Management*, Vol. 17 No. 4, pp. 223-35.

Turner, J.R. (1999), *Handbook of Project-based Management*, McGraw Hill, New York, NY.

Vitale, M.R. (1986), "The growing risks of information system success", *MIS Quarterly*, Vol. 10 No. 4, pp. 327-34.

Wall, D.M. (1997), "Distribtions and correlation in Monte Carlo simulation", *Construction Management and Economics*, Vol. 15, pp. 241-58.

Wang, S. (2001), "Designing information systems for e-commerce", *Industrial Management & Data Systems*, Vol. 101 No. 6, pp. 304-15.

Williams, T.M. (1995), "A classified bibliography of recent research relating to project risk management", *European Journal of Operational Research*, Vol. 85 No. 1, pp. 18-38.

Williams, T.M. (1996), "The two-dimensionality of project risk", *International Journal of Project Management*, Vol. 14 No. 3, pp. 185-6.

Woodward, D.G. (1995), "Use of sensitivity analysis in build-operate-transfer project evaluation", *International Journal of Project Management*, Vol. 13 No. 4, pp. 239-46.

Yang, Y.H. (2001), "Software quality management and ISO 9000 implementation", *Industrial Management & Data Systems*, Vol. 101 No. 7, pp. 329-38.

Yeo, A. (1991), "Project cost sensitivity and variability analysis", *International Journal of Project Management*, Vol. 9 No. 2, pp. 111-6.

Yeo, K.T. (1990), "Risks, classification of estimates and contingency", *J. Manage. Eng.*, Vol. 6 No. 4, pp. 458-70.

Yingsutthipun, J. (1998), "The key-risk identification of Thailand's transport infrastructure projects based on the build-operate-transfer scheme", AIT Master's thesis, ST98-53, Bangkok.

Zhi, H. (1995), "Risk management for overseas construction projects", *International Journal of Project Management*, Vol. 13 No. 4, pp. 231-7.

Further reading

Dey, P.K. and Gupta, S.S. (1999), "Decision support system for pipeline route selection", *Cost Engineering*, Vol. 41 No. 10, pp. 29-35.

Meredith, J.R. and Mantel, S.J. (2000), *Project Management: A Managerial Approach*, 4th ed., Wiley, New York, NY.

Corresponding author

Prasanta Kumar Dey can be contacted at: p.k.dey@aston.ac.uk