Risk management in information technology projects

Prasanta Kumar Dey*

Operations and Information Management, Aston Business School,
Aston University, Birmingham B4 7ET, UK
E-mail: p.k.dey@aston.ac.uk.
* Corresponding author

Jason Kinch

First Caribbean Bank, Barbados
E-mail: jasonkinch@gmail.com

Abstract: There are several studies on managing risks in information technology (IT) projects. Most of the studies identify and prioritise 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 IT developers' perspective. Although a few studies introduced a framework of risk management in IT projects, most of them are presented from clients' perspectives and very little effort has been made to integrate this with the project management cycle. As IT developers absorb a considerable amount of risk, an integrated framework for managing risks in IT projects from developers' perspective is needed in order to ensure success in IT projects. The main objective of the paper is to develop a risk management framework for IT projects from the developers' perspective. This study uses a combined qualitative and quantitative technique with the active involvement of stakeholders in order to identify, analyse and respond to risks. The entire methodology has been explained using a case study on an information technology project in a public sector organisation in Barbados.

Reference to this paper should be made as follows: Dey, P.K. and Kinch, J., (2008) 'Risk management in information technology projects', *Int. J. Risk Assessment and Management*, Vol. 9, No. 3, pp.311–329.

Keywords: analytical framework; information technology projects; risk management.

Biographical notes: Dr Prasanta Kumar Dey is a senior lecturer in Aston Business School, Birmingham, UK. He specialises in operations and information management. He is the founding co-editor of the *International Journal of Energy Sector Management*, an Emerald Journal. Prior to joining Aston, he worked in Indian Oil Corporation limited (a five hundred fortune company), India, as a project executive for 13 years and worked in various oil refining and pipelines transportation projects. Subsequently, he directed and lectured in a graduate project management program in the University of the West Indies, Barbados, for 5 years. Dr Dey holds Bachelor in Mechanical Engineering, Master's in Industrial Engineering and Management, and PhD in Production Engineering. He has published extensively in the international refereed journals. His current research interest includes total quality management, supply chain management, project management and risk management.

Jason Kinch holds a BSc in Computer Science and a Masters in Project Management from the University of the West Indies, Cave Hill campus. He is also certified as a Microsoft Professional (MCP) as well as a Macromedia ColdFusion Developer. He is currently pursuing his Certified Information Systems Security Professional (CISSP) designation. He has worked on a number of software development projects in roles from developer to team leader. He has worked as senior programmer/analyst in a number of financial institutions, and he is currently Information Technology Manager at a local insurance company.

1 Introduction

Information technology (IT) projects on implementation provide information to support operations, management analysis and decision-making within an organisation (Wang, 2001; Yang, 2001). However, they are vulnerable to time and cost overrun, along with quality under-achievement. Additionally, the presence of a high level of bugs during the initial period of trial and commercial use is not uncommon. Although managers claim that they manage the above issues quite efficiently, there is evidence of a lack of management of IT projects even by leading IT providers. IT projects suffer from market risk, financial risk and technical risk (Schwalbe, 2002). The IT providers must have favourable answers to the following questions in order to achieve success. Does the IT project fulfil the customer's demand/requirement? How much competition is it likely to face? Will the benefits arising from the IT surpass the cost of development? Is the project technically feasible? Will the 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 the security system work throughout its life? There are examples of high profile IT project failures in the literature (Baccarini et al., 2004). Evidence indicates that risks have not been managed effectively (Hedelin and Allwood, 2002).

Although some managers claim that they manage risk in their projects, there is evidence 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 successful software development. Hence, there is a need for integrated risk management.

The success of IT projects depends on the criteria: functionality, quality and timeliness. IT projects are implemented to facilitate a specific organisational function. Unless it is successful in performing this function, the purpose of the IT project is defeated. Customers should be delighted with the performance of the product. The IT projects must be delivered to the customer on time (as scheduled). Quite often, a penalty clause is associated with the delivery of the IT projects, as the client is expected to face a substantial business setback if they cannot bring about a change in their system on time. However, project delay not only incurs cost due to penalty, there is always a 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, 2004). IT projects are not exceptions, as IT project planning is done with the minimum of 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. There is, therefore, a need to manage risk in IT projects .

Although researchers and professionals have written on risk management in IT projects (Keil et al., 1998), very little work has been done in order to involve all the concerned stakeholders in identifying and analysing risk and integrating the risk management process with an overall project management cycle. Various frameworks of risk management have been suggested, some of which deal only with the application part and some models deal with risk holistically, covering application, organisation and inter-organisation levels (Bandyopadhyay et al., 1999). However, there is a dearth of literature on IT project risk management from the developers' perspective. This study presents a risk management framework from the developers' perspective, which is integrated with the IT project management cycle and involves the stakeholders concerned with managing risk. Accordingly, the objective of this study is to develop an integrated risk management framework for IT projects and apply the framework to an IT project in order demonstrate its effectiveness.

2 Literature review

Risk refers to future conditions or circumstances that exist outside 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) define 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 into the following three stages:

- risk identification
- risk analysis
- risk responses.

Tummala et al. (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 research 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 judgement, plan decomposition, assumption analysis, decision drives and brainstorming for the effective identification of risk factors 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 to risk identification in product innovation has been reported by Halman and Keizer (1998).

Most of the analyses done so far are centred on analysing 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 Beta distribution (e.g. 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 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 an 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 the risk of software development (Alter and Ginzberg 1978; Barki et al., 1993; Boehm, 1991; Boehm and Ross, 1989; 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, they did not evolve any strategy for risk mitigation. Boehm's work (1991) was probably the most significant as he identified the top ten 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 currently available methods for managing risk in IT projects are not comprehensive (Bandyopadhyay et al., 1999) as they deal with specific types of risk. 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 Persson (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. (1983) addressed the issue of risk monitoring to ensure the effective implementation of risk control measures. Huang et al. (2004) introduced a risk prioritising method using an analytical hierarchy process for enterprise resource planning implementation, which is a combination of software development and a 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. As IT providers also carry a considerable amount of risk, there is a need for managing from the developers' perspective, with the involvement of the concerned stakeholders, with the consideration of both qualitative and quantitative risk factors and integrating the risk management process with the project management cycle.

3 Methodology

This study adopts a case study method for developing a risk management framework for IT projects and involves the concerned stakeholders using focus group discussions. Figure 1 shows the proposed risk management framework for IT projects

Figure 1 consists of the following steps:

3.1 Step 1: Analysing functional requirements

Software performs organisational 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 a change in business processes, process reengineering is quite common while establishing functional requirements and selecting information technology solutions.

3.2 Step 2: Establishing the scope of software development project and developing a 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 the formation of a work breakdown structure.

3.3 Step 3: Identifying risky work packages

The work packages which are vulnerable to time, cost, and quality targets failures are then identified with the active involvement of both functional and IT executives.

3.4 Step 4: Identifying risk events

Risk events are then identified for each risky work package, using various tools and group consensus.

3.5 Step 5: Analysing risk

The probability and severity of the risk events are analysed using qualitative and quantitative tools, with the active involvement of the stakeholders.

3.6 Step 6: Developing a risk management plan

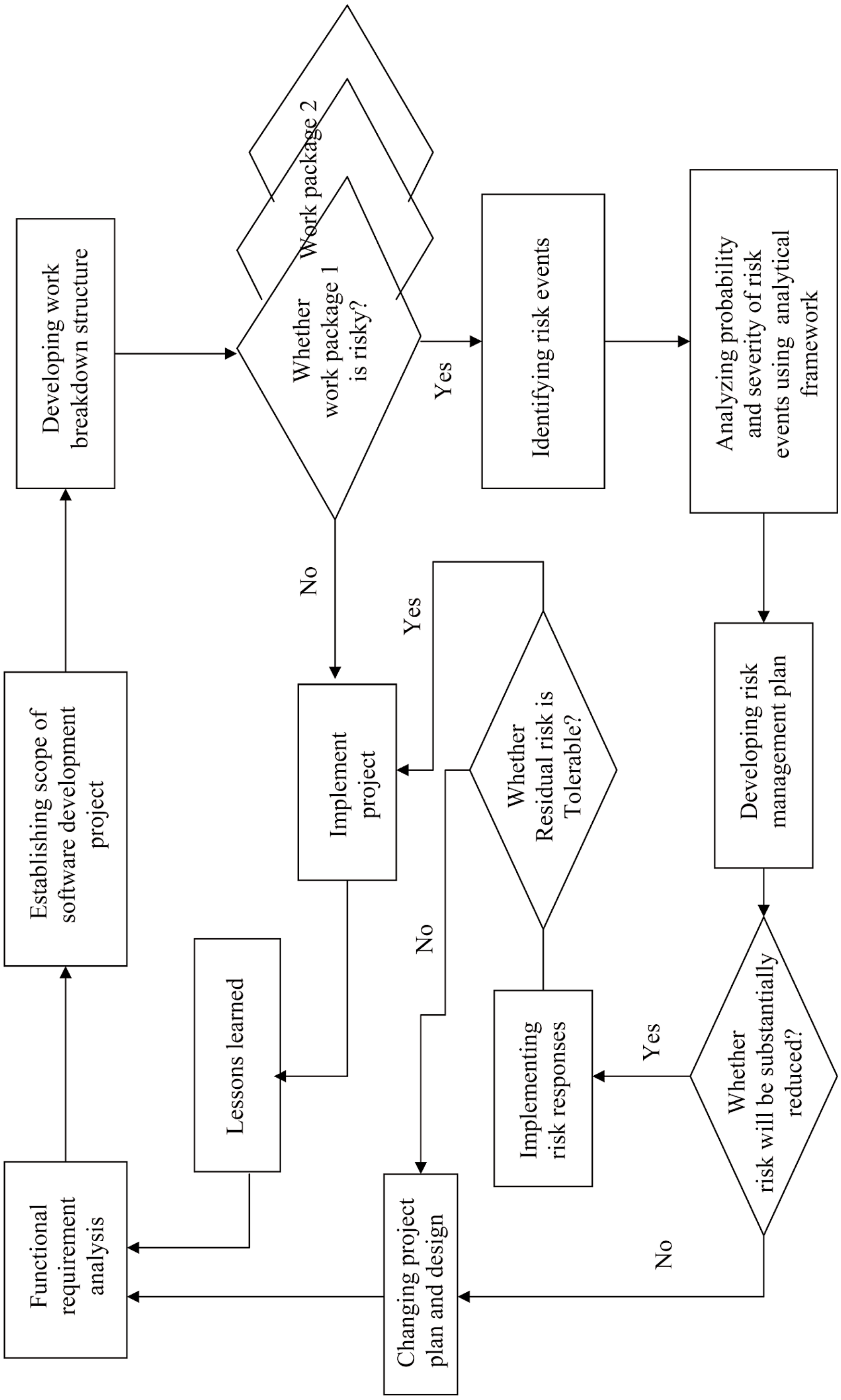
A risk management plan is then developed in response to adverse risk events before they occur. The plan is evaluated with respect to contribution in reducing the effect of risk. Risk responses are implemented if they have the potential to reduce project risk substantially. The actual project is then implemented.

3.7 Step 7: Controlling risk

A risk management plan suggests various strategies for all likely risk events depending on the probability and severity, along with the perceptions of the stakeholders. A dynamic control mechanism, therefore, 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.

Figure 1 Proposed risk management framework for IT projects



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 3000 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 an IT project for an 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 1/1000 of the value of the undelivered software per month.

4.1 Step 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 Step 2: Scope and work breakdown structure

The detailed requirement analysis helped in deriving the project scope with the active involvement of the 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 for 'Data conversion of existing data':

- reception and application receipt module
- registry module
- drawing office module
- planning module
- integration
- training and documentation modules.

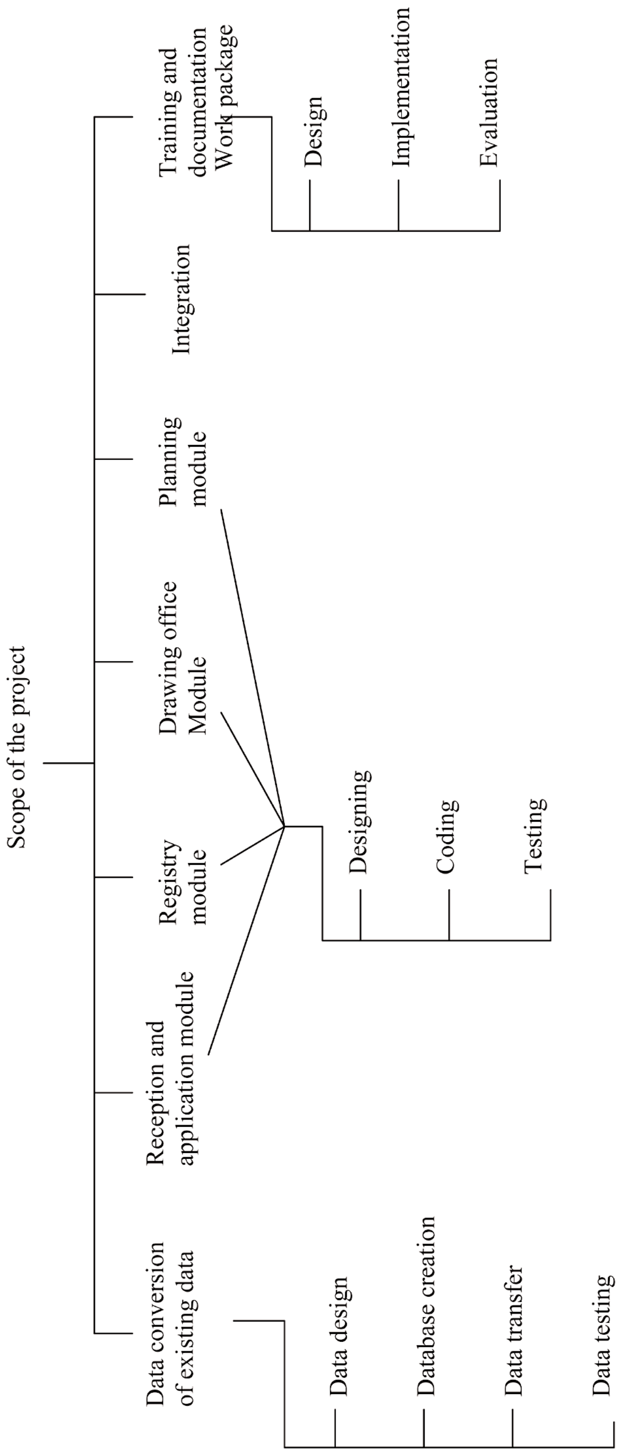
Other than 'data conversion of existing data', 'integration', and 'training and documentation' work packages, each work package had the following common activities:

- design
- coding
- 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 IT project under study.

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



4.3 Step 3: Identifying risky work packages

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

4.4 Step 4: Identifying risk events

Risk events do not allow a project to achieve its goals. Various qualitative and quantitative tools can be applied to identify the potential risks of any project (Dey et al., 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 do 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.
- Problems in coding and unit test – The quality of coding is judged by testing. Various types of testing are planned to ensure quality in IT projects depending on their characteristics. However, if the programmers, who are developing the codes, are held responsible for testing, it is difficult to check the quality properly. Hence, it is always suggested that a different group be established for unit testing. This does not usually occur in IT projects, because the team leader on a project reviews all codes and establishes coding guidelines and standards.

4.5 Step 5: Analysing 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 analyse risk. This study adopts a risk-mapping method (Dey, 2001b) to determine the probability and severity of identified risk events. Table 1 shows the risk map for the project under study. The risk map was developed through a brainstorming session and group consensus among functional and IT executives of TCPO and the executives of the IT provider. The analysis reveals the following:

- Loss or unavailability of key personnel during the course of the project was found to have the highest likelihood of occurrence and severity. Employee turnover during the project will have a tremendous negative impact, as it is extremely difficult to get competent, experienced technical persons within a short period and moreover, it takes time for them to adjust to a new environment. This has a negative impact on the productivity of IT 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 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 an experienced team who identified all the technical requirements of the project.
- The coding and unit testing were also not very risky activities, as an 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 they were to occur at all, the experienced developer could easily handle them.

Table 1 Risk severity to probability factor matrix

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

The risk events were given a severity/probability rating based on Table 2.

Table 2 The ratings

Severity probability factor rating (SPR)	*Strategy*
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

The severity probability factor rating (SPR) in Table 3 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).

Table 3 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 *Probability*	High	Very high

The Drawing office work package was found to be the most risky, owing to its extremely technical nature. Loss of 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 a tremendous setback if this risk event occurred.

The overall impact on the project is then determined using the schedule impact scale and the contract clause on schedule overrun. Table 4 shows the schedule impact scale.

Table 4 Impact scale

	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 one month grace period allowed by the Town and Country Planning Department.

The contract clause for a 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 1/1000 of the value of the undelivered software per month. Start here

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

The cost of the software per month = cost of development/development period
= 400,000/12
= $33,333.33.

The penalty was 1/1000 of the cost per month for each day delay.

Therefore, the daily penalty = cost per month /1000
= 33,333.33/1000
= $33.33

The impact of risk events on each work package is then determined using the information in Tables 1, 2 and 4. Table 5 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 the overall project schedule in the event of delay in any activity
- each module will be executed simultaneously
- data conversion, module implementation and training will be executed sequentially
- risk events are independent
- all the figures in Table 5 are expected time overrun.

Accordingly, the expected time overrun was 161 days (44% of the original schedule) and the 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 a significant intangible cost like loss of image, opportunity cost, etc.

Table 5 Impact on schedule of risk occurrence

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

4.6 Step 6: Developing a risk management plan

Risk analysis leads to the derivation of a few effective risk responses in line with principles such as avoidance, transfer, reduction and absorption. 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 website
- implement a scope management plan
- research all limitations of development environment and compare with software requirements
- 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 were recommended (Table 6).

Table 6 Strategies against risk events

Risk events	*Risk strategy*	*Strategy type*
Loss/ lack of resources	Losing resources at critical points of software development has high negative impact. This can be avoided by ensuring that project team members are 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. Techniques like process reengineering and benchmarking can be employed for deriving the requirements with the active involvement of functional people of the client's organisation.	Transference/ Reduction
Inadequate design	This type of problem is present in any software development roject. Use of 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, website	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

The risk plan (Table 7) for the project under study is then derived based on the SPR and the above risk strategies.

Table 7 Risk management plan

Risk event	*Impact/s on project*	*L*	*C*	*R**	*Risk Treatment/s*	*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

Table 7 Risk management plan (continued)

Risk event	*Impact/s on project*	*L*	*C*	*R**	*Risk Treatment/s*	*Responsibility*	*Project phase*
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 development	Impact quality	2	3	2	Research all limitations of development environment and compare with software requirements	Project Technical Lead	Planning, , design
Unavailable customer contact	Impact quality	3	2	2	Utilise internet technologies to stay in contact e.g. E-mail, project website		All
Problems in coding and unit	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

4.7 Step 7: Controlling risk

Risk management planning develops a detailed strategy for risk responses depending on the nature of likely risks. Another round of brainstorming was conducted to determine the 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 was tolerable before implementation. To control risk in the project under study, a small group was formed with representatives from both the 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 IT 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 the authorisation of the project team for software development. Although there was scope creep, the scope management plans with the involvement of the 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 was little time and cost overrun. An appropriate software development environment was 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 efforts of stakeholders. It helped tremendously to appraise project progress as well, accelerating decision making. The coding and unit testing were trouble free, as third party inspection was organised for testing.

5 Validation of the proposed framework

The proposed IT project risk management framework was then validated through a questionnaire survey among the participating key personnel. This was carried out in order to reveal the following:

- the overall significance and importance of managing risk in IT projects
- the acceptability of the method
- the usability of the proposed framework
- the usability of the qualitative and quantitative method of managing risk for IT projects
- the comprehensibility of the proposed framework
- the implementability of the framework
- the acceptability of the research findings
- the adequate stakeholders' involvement in the process managing risk
- the applicability of the methodology and the framework in other organisations
- the future improvement of the framework and method of application.

In overall response, the participants have been fairly positive about the framework for IT project risk management. They have indicated that the basic principle and application of the proposed framework is quite user friendly and the steps of the framework are easy to implement and helpful, as they consider the decisions of individual stakeholders before reaching a consensus. Their qualitative answers to the last question of the questionnaire have revealed that the holistic and integrated nature of the model is appreciable. However, they have agreed that the success of its use would largely depend on the number of stakeholders involved and their experiences. They pointed out that identifying the relevant risk factors that influence decision making have been very critical. They suggested that a more detailed understanding of the theory and application of IT project risk management prior to the focus group would have greatly reduced the duration of the exercise.

6 Conclusion

Effective risk management in IT projects ensures the successful accomplishment of projects with customers' satisfaction, functional achievement, and overall better financial performance of the organisations. 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 in not achieving its objectives. Therefore, a risk management plan, along with other work plans, 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 the 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 the software developer perspective and involves all the concerned stakeholders for risk analysis and deriving responses.
- it takes both a subjective and an objective approach to derive specific responses for managing risk
- it is totally integrated with the software development cycle, hence, its practice can be easily institutionalised
- the approach to risk management is user friendly and not complex.

A 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.

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.286–295.

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.437–444.

Barki, H., Rivard, S. and Talbot, J. (1993) 'Toward an assessment of software development risk', *Journal of Management Information Systems*, Vol. 10, No. 2, pp.203–225.

Benry, J. (1989) 'A new distribution function for risk analysis', *Journal of the Operational Research Society*, Vol. 40, pp.1121–1127.

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

Chapman, C.B. and Cooper, D.F. (1983) 'Risk analysis: testing some prejudices', *European Journal of Operational Research*, Vol. 14, No. 3, pp.238–247.

Charette, R.N. (1989) 'Software engineering risk analysis and management', York: Intertext.

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.

Dey, P.K. (1999) 'Process Re-engineering for effective implementation of projects', *International Journal of Project Management*, Vol. 17, No. 3, pp.147–159.

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

Dey, P.K. (2001a) 'Reengineering materials management: a case study on Indian refinery, *Business Process Management Journal*, Emerald, MCB University Press, 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.235–245.

Dey, P.K. (2002) 'An integrated assessment model for cross-country pipelines', *Environmental Impact Review*, Vol. 22, No. 6, pp.703–721.

Dey, P.K. (2004) 'Analytic hierarchy process helps evaluate projects in Indian oil pipeline industry', *International Journal of Operations and Production Management*, Vol. 24, No. 6, pp.588–604.

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

Dey, P.K. and Ogunlana, S.O. (2004) 'Selection and application of risk management tools and techniques for Build-operate-transfer Projects', *Industrial Management and Data Systems*, Vol. 104, No. 4, pp.334–346.

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 Persson, J. (1994) 'A fire drill for business', *Information Strategy: The Executive's Journal*, pp.44–47.

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.287–290.

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.

Hedelin, L. and Allwood, C.M. (2002) 'IT and strategic decision-making' , *Industrial Management and Data System*, Vol. 102, No. 3, pp.125–135.

Hoffman, D.C. (2001) 'Mitigating Software Development Risk' 2001, 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 Systems*, Vol. 104, No. 8, pp.681–688.

Johnson, J. (1995) 'Chaos: The dollar drain of IT project failures', *Application Development Trends,* Vol. 2, No. 1, pp.41–47.

Jones, C. (1994) *Assessment and control of software risks*, Englewood Cliffs, NJ: Prentice-Hall.

Kangari, R and Riggs, L.S. (1989) 'Construction risk assessment by linguistics', *IEEE Trans. Eng. Manag.,* Vol. 36, No. 2, pp.126–131.

Keil, M. and 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.

McFarlan, F.W. (1981) 'Portfolio approach to information systems', *Harvard Business Review*, Vol. 59, No. 5, pp.142–150.

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.

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.129–147.

Royer, P S. ' Risk management', *PM Network*, September 2000, pp.31–39.

Schwalbe, K. (2002) 'Information technology project management, Canada: Thomson Learning,

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

Tummala, V., Rao, M. and Leung, Y.H. (1999) 'Applying a risk management process (RMP) to manage cost risk for an EHV transmission line projects', *International Journal of Project Management,* Vol.17, No. 4, pp.223–235.

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

Vitale, M.R. (1986) 'The growing risks of information system success', *MIS Quarterly*, Vol. 10, No. 4, pp.327–334.

Wang, S. (2001) 'Designing information systems for e-commerce', *Industrial Management and Data Systems*, Vol. 101, No. 6, pp.304–315.

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

Yang, Y.H. (2001) 'Software quality management and ISO 9000 implementation', *Industrial Management and Data Systems*, Vol. 101, No. 7, pp.329–338.