ABSTRACT

In today's e-world, software applications are the driving force of modern business operations. But software development is viewed as one of the major problem areas of almost every corporation in the modern world as software development projects still fail to be delivered on time, within budget and with desired quality. This is mainly due to lack of planning in developing the software and the unforeseen factors that come into picture at a later stage of planning. One area of concentration in Software Project Management that has focused on these problems is Risk Management. Risk Management attempts to assess and then control the risks that make the software management a tedious and challenging job.

In the era of downsizing, consolidation, shrinking budgets, increasing technological sophistication and shorter development times, Risk Management can provide valuable insights to key project professionals in ensuring project success. Proper planning, analysis and monitoring of potential risk helps in addressing the issues long before the issues surface as problems and adversely affect project cost, performance and schedule.

With so many project deliverables driving the global organisation, insight into key project information is essential. Being able to identify the factors that can sabotage a seemingly smooth-running project is critical. Every project has risks and the biggest mistake would be to ignore those risks.

The management of risks is a central issue in the planning and management of any software venture. This thesis tries to analyze the various ways of measuring risks in software projects and design new metrics for measuring the same in different and more efficient ways so as to minimize the adverse effect of risks on development of software projects. Measuring and controlling the risks before they become a reality i.e. proactive risk management for software projects is the main focus area of the present work.

DECLARATION

I hereby certify that the work which is being presented in the thesis entitled, ***"Analyzing and Designing Risk Metrics for Software Projects"***, in partial fulfillment of the requirement for the award of degree of Master of Engineering in Software Engineering submitted in Computer Science and Engineering Department of Thapar Institute of Engineering and Technology (Deemed University), Patiala, is an authentic record of my own work carried out under the supervision of Ms. Inderveer Chana Lyall.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other university.

(Shivani Goel)

This is to certify that the above statement made by the candidate is correct and true to best of my knowledge.

(Inderveer Chana Lyall)
Senior Lecturer,
Computer Science and Engineering Department
Thapar Institute of Engineering and Technology, Patiala

Countersigend by

(Mr. R.S. Salaria)
Head,
Computer Sc. & Engg. Deptt.
Thapar Institute of Engg. & Tech.,
Patiala.

(Dr. D.S. Bawa)
Dean of Academic Affairs,
Thapar Institute of Engg. & Tech.,
Patiala.

The M.E. (Thesis)Viva-Voice examination of Shivani Goel, Roll No. 8023151 , M.E. (Software Engineering) , Thapar Institute of Engineering and Technology, Patiala has been held on_____.

Supervisor **External Examiner**

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my thesis supervisor, Ms. Inderveer Chana Lyall for her help, encouragement and support for suggestion of this work. Inspite of her hectic schedule she was always approachable and took her time off to attend to my problems and to give appropriate advice. I am also grateful to all my colleagues at CSED for advising me whenever required and I hope to have the cooperation and support from them in future too.

(Shivani Goel)

Roll No. 8023151

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ORGANIZATION OF THESIS

Chapter 1 defines the problem taken up in the thesis.

Chapter 2 deals with Risk Management Concepts.

Chapter 3 discusses Software metrics' concepts and their importance.

Chapter 4 highlights the research approach taken up in the thesis

Chapter 5 discusses the outcome of the survey.

Chapter 6 proposes techniques and metrics for handling risks in software projects

Chapter 7 concludes the study and throws light on future scope for the work done.

CHAPTER 1
INTRODUCTION

The focus of the present work is on software project risks. This work identifies the reasons why the software development projects fail to be delivered on time, within budget and with desired quality. One of the reasons of these failures are software engineering risks and lack in risk management. Software risk metrics help in measuring the software engineering risks and risk mitigation steps provide solutions or remedial actions that can be used to handle these risks and to lessen their impact on software projects. This chapter gives the insight into the current work.

1.1 Background

The software community continually attempts to develop technologies that will make it easier, faster and less expensive to build high-quality software. However, recent surveys show that most projects fail to meet their targets highlighting the inadequacies of traditional project management techniques to cope with the unique characteristics of this field. Despite the major breakthroughs in the discipline of software engineering, studies show that almost one third of projects are cancelled before completion. At the time of cancellation, they average about 100 percent over schedule and budget. Another one third are completed, but at the cost of overrunning schedule and budget by up to one third. Most of the remaining one third exceed the original planned schedule and cost, though by lesser amounts. Only a few software projects are completed in less than the originally planned schedule and cost. One of the main reasons is the inability of the companies to feel risk and the need for implementing risk management techniques rigorously.

Much of the failure can be avoided by the software project managers planning pro-actively and dealing with risk factors rather than waiting for problems to occur and then trying to handle those problems. Software Risk Management has been proposed as a solution to provide insight into potential problem areas and to identify, address and eliminate various types of risks before they derail the project. Software risk management in itself is not sufficient unless it measures the risk impact effectively and identify the severity of the risks.

According to Lord Kelvin, an eminent physicist, “When you can measure what you are speaking about, and can express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a

meager and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the stage of science. ” Therefore, risk metrics should be used for measuring the risks in software projects for effective software risk management.

1.2 Research Objective

According to David Gluch’s , A Construct for Describing Software Development Risks,

“A *risk is a combination of an abnormal event or failure and the consequences of that event or failure to a system's operators, users, or environment* .”

Risk management has reached a new level of importance in information age. The growth of networked information systems and distributed computing has created a potentially challenging environment. Today, Risk Management is more important in software development than previously realized. Considerable attention has been given to ensure comprehensive identification and objective assessment of project, process and product risks, to provide a clear understanding of the extent of risk exposure faced by a project. However identification and assessment will be worthless unless responses can be developed and implemented which really make a difference in addressing identified risks. Yet risk response development is perhaps the weakest part of the risk process and it is here that many organizations fail to gain the full benefits of project risk management.

If management will not have insight into what could go wrong, and if it fail to quantify the risks, more resources will be spent in correcting the problems that could have been avoided. Sooner, catastrophic problems (surprises) may occur without warning (and with no recovery possible), decisions will be made without complete information or adequate knowledge of future consequences, the overall probability of successful completion of the program is reduced and program will always be in a crisis.

By applying good risk management practices, the software projects will be able to take on more risk and also exploit opportunities that now have to be passed by. Further, time to act, rather than react, will be gained, along with a greater number of alternatives to choose from when problems are eventually encountered. As a result, the software projects will develop a risk taking ethic, one with a bias toward informed action.

The objective of the present work is to study already existing risk management practices and risk metrics being used in various software companies and suggesting some more risk metrics that will reduce the probability of an unforeseen risk. Along with that, a number of risk mitigation strategies for handling the risks proactively would also be suggested.

1.3 Defining The Problem

Success of a software project is not always certain. Risks can cause the project to miss its key objectives of cost, time and scope. Risk management deals with identifying the risks by monitoring the software development and warning the project manager about the risks of high priority before they become a reality. Though many organizations are using risk management techniques, but the measures used for identifying,

assessing and mitigating the risks are not updated regularly and hence the benefits from risk management are not generating results.

If the company makes risk management its top priority, it can learn from past experiences and reduce the chance for failure by managing risks before they become a reality. Most of the organizations attempting to manage risks seem to identify and assess their impact upto a certain extent only. To improve upon this, risk metrics should be used to identify risks in different categories and proper risk mitigation strategies should be developed to reduce the impact of the risks proactively.

The study presented here proposes some new risk metrics for software projects and also suggests some techniques that can be used in mitigating the identified risks based on current practices being used by companies and past researchers.

Definition: The guidelines outlined in this thesis offer a framework for developing effective risk metrics and maximizing the benefits to be achieved through proactive risk management. Main focus is on developing a few more metrics for risks in software projects similar to the already existing ones by studying the risk management practices currently being used in various software companies. For every software company, time is critical, therefore the goal should be to act early before a source of risk evolves into a major crisis.

1.4 Methodology Used

There can be a number of ways to work for the defined problem. Various research approaches are discussed in detail in chapter 4. The method currently used is the survey technique. A questionnaire consisting of 45 questions was formulated containing four different sections and sent to 20 companies developing software products. The responses from all were studied and the outcomes have been summarized in chapter 5. Based on the survey, better risk management approaches and software risk metrics are suggested in chapter 6 .

1.5 Conclusion

This chapter focused on the brief introduction of Risk management and the importance of measuring risks using risk metrics. It also defined the problem and the methodology taken up in the thesis.

The next chapter throws light on the concepts of software risk management.

CHAPTER 2 RISK MANAGEMENT CONCEPTS

Risk Management has reached a new level of importance in Information age. The complex process of software acquisition and development encompasses most, if not all, aspects associated with software risk management. The concepts outlines in this chapter are extracted through the literature survey. This chapter discusses theoretical background of Software risk management and the general practices used by different software companies in combating risk.

2.1 Software Project Management

Software Risk Management is one of the key areas of Software Project Management as suggested by the PMBOK [25].Over the past few years the project management software industry has gone through a major culture change as fundamental as the introduction of the first project management software systems back in the 1960's. Program costs have skyrocketed and reached into the tens of billions of dollars. Even minor schedule slips can cost hundreds of millions of dollars. In an era of tight resources, being even slightly wounded makes a program vulnerable.

Software Project Management was first used to manage the US space program. Its practice has now been expanded rapidly through the government, the military and the corporate world. Here is the main definition of what Project Management is:-

> "A methodological approach to achieve agreed upon results, within a specified time frame, with defined resources".

Software Project Management is an umbrella activity within Software Engineering[16]. It begins before any technical activity is initiated and continues throughout the definition, development and maintenance of computer software. Software Project Management focuses on 3 P's: People, Problem and Process

People: The pivotal element in all software projects is People**.** The People must be organized into effective teams. The cultivation of motivated, highly skilled software people is very important.

Problem: The Problem must be communicated from customer to developer, partitioned into its constitutional parts and positioned for work by a software team. Before a project can be planned, its objectives and scope should be established, alternative solutions should be considered and technical and management constraints should be identified.

Process: The Process must be adapted to the people and the problem[18]. A software process provides the framework from which a comprehensive plan for software development can be established. A small number of framework activities are applicable to all software projects- a number of different task sets- tasks, milestones, deliverables

and quality assurance points- enable the framework activities to be adapted to the characteristics of the software project and the requirements of the project team. The umbrella activities such as software quality assurance, configuration management and measurement - overlay the process model.

2.2 Software Risk Management

Software project managers take steps to ensure that their projects are done on time and within effort and cost constraints. However, project management involves far more than tracking effort and schedule. Risk management is a discipline for living with the possibility that future events may cause adverse effects. Risk Management adds to Project Management as Risk management looks ahead in the project and adds a structured approach for the identification and analysis of risks to begin planning.

The official definition of Risk management has been taken from the American Risk and Insurance Association[16].

"Risk management is the systematic process of managing an organization's risk exposures to achieve its objectives in a manner consistent with public interest, human safety, environmental factors and the law. It consists of the planning, organizing, leading, coordinating and controlling activities undertaken with the intent of providing an efficient pre-loss plan that minimizes the adverse impact of risk on the organization's resources, earnings and cash flows".

Risk Management definition by Dr. Barry W. Boehm [14]indicates that risk management is more important in software development than previously realized. SEI Definition of risk management sets forth a discipline and environment of proactive decisions and actions to

- ✓ assess continuously what can go wrong (risks)
- ✓ determine what risks are important to deal with
- ✓ implement strategies to deal with those risks.

Software Engineering Institute (SEI) Risk Management paradigm is depicted in Figure 2.1

Figure 2.1 SEI's risk Management Paradigm[26]

The paradigm illustrates a set of functions that are identified as continuous activities throughout the life cycle of a project. Each risk nominally goes through these functions sequentially, but the activity occurs continuously, concurrently and iteratively throughout the project life cycle. Table 2.1 specifies the activities performed during the risk management cycle:

Function	**Description**
Identify	Search for and locate risks before they become problems.
Analyze	Transform risk data into decision-making information. Evaluate impact, probability and timeframe, classify risks and prioritize risks.
Plan	Translate risk information into decisions and mitigating actions (both present and future) and implement those actions.
Track	Monitor risk indicators and mitigation actions.
Control	Correct for deviations from the risk mitigation plans.
Communicate	Provide information and feedback internal and external to the project on the risk activities, current risks, and emerging risks. *Note*: Communication happens throughout all the functions of risk management.

Table 2.1: Risk Management Activities[26]

A formal risk management process provides a number of benefits to the project team. It gives us a structured mechanism to provide visibility into threats to project success. By considering the potential impact of each risk item, we can make sure we focus on controlling the most severe risks first. A team approach allows the various project stakeholders to collaboratively address their shared risks and to assign responsibility for

risk mitigation to the most appropriate individuals. We can combine risk assessment with project estimation to quantify possible schedule slippage if certain risks materialize into problems. Without a formal approach, we cannot ensure that our risk management actions will be initiated in a timely fashion, completed as planned and effective. The net result of these activities is to help avoid preventable surprises late in the project, and therefore improve the chance of meeting our commitments.

2.2.1 Risk Management Principles

The SEI identifies seven principles that provide a framework to accomplish effective risk management[26]. They are:

1.Maintain a global perspective **–** view software risks within the context of system and the business problem that it is intended to solve.

2.Take a forward –looking view- think about the risks that may arise in the future(due to changes in the software), establish contingency plans so that future events are manageable.

3. Encourage open communication- if someone states a potential risk, don't discount it. If a risk is proposed in an informal manner, consider it. Encourage all stakeholders and users to suggest risks at any time.

4. Integrate- a consideration of risk must be integrated into the software process.

5. Emphasis on a continuous process- the team must be vigilant throughout the software process, modifying identified risks as more information is known and adding new ones as better insight is achieved.

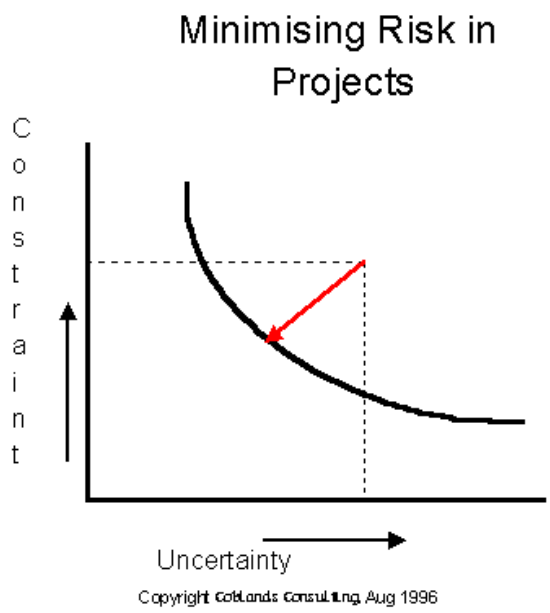
6. Develop a shared product vision- if all stakeholders share the same vision of the software, it is likely that better risk identification and assessment will occur.

7. Encourage teamwork- the talents, skills and knowledge of all stakeholders should be pooled when risk management activities are conducted.

The principles of software project risk management can be stated very simply. Failure to manage risks is characterized by inability to decide what to do, when to do it, and whether enough has been done.

Risk Management is a facet of Quality, using basic techniques of analysis and measurement to ensure that risks are properly identified, classified and managed. In order to manage risks we have to understand what a risk is. We all face constraints in our

projects and also uncertainty. So we can minimize the risk in the project either by eliminating constraints (a nice conceit) or by finding and reducing uncertainty.



The curved line indicates the 'acceptable level of risk', whatever that may be in the individual case. The risk may be reduced to an acceptable level by reducing either or both of uncertainty and constraint. In practice, few people have the opportunity to reduce constraint, so most focus on the reduction of uncertainty. It is also worth noting from the diagram that total elimination of risk is rarely achieved. So we have to consider how to manage that remaining risk most effectively.

Fig 2.2:Uncertainty and constraints for dealing with risk

2.2.2 Risk Management Activities

Risk management is an umbrella activity and it needs to be carried out using a number of sub-activities. Risk management involves several important steps, each of which is illustrated in Fig.2.3.

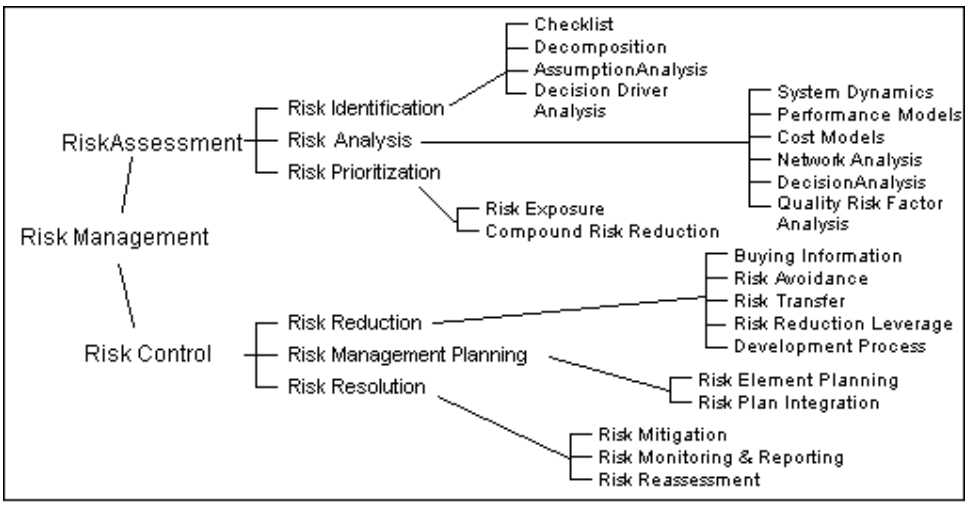


Figure 1. Steps in Risk Management

Figure 2.3: Risk Management Steps[39]

Risk Assessment : Risk Assessment comprises of Risk Identification, Risk Analysis and Risk Prioritization.

Risk Identification: It is the process of specifying the threats to the project plan[16]. There are two types of risks.
Generic risks - Risks those are common to all software projects, such as misunderstanding the requirements, losing key personnel, or allowing insufficient time for testing.

Project-specific risks – The threats that result from the particular vulnerabilities of the given project. For example, a vendor may be promising network software by a particular date, but there is some risk that the network software will not be ready on time.

Any one of the following technique can be used for identifying risks[17]:

1. Preparing checklists- Checklists evolve as time progresses and encapsulate organizational learning from past experiences. Organization should put continuous efforts in keeping the checklists current, updating them with the potential risks and learning from the projects.

2. Examining organizational history- By examining the track record of an organization, we can identify the pattern of risks. For example, when one gets a new project from an existing customer, a review of the previous projects done with him may reveal patterns like the customer changing requirements very often.

3. Framework based risk categorization – It is the process of identifying all possible generic and project specific risks for broad categories of risks. Table 2.3 shows an example of a framework based risk categorization.

Risk category	**Generic Risks**	**Project-specific risks**
Technology Risk	Complete change in technology (from client server to internet computing)	Technology used in a particular project may become obsolete
People related risks	Attrition	Inability to find people with skill set specific to the project

Table 2.2: framework based risk categorization[17]

4. Using Literature and buying Information- While identifying the risks for a project, it is surely worth while to look at the industry trends and using the available literature and information judiciously. Information buying is the process of sharing and aggregating information from all the competitive organizations to identify the market and business risks. For example, an organization which does not keep pace with the salaries in the market, cannot attract or retain top talent.

5. Simulations to identify performance risks- Statistical based simulation techniques are used for identifying performance risks. For example, in the Internet projects, a site when may become popular

should be able to scale and handle the increased traffic. Simulation models provide a good way to predict how the site would handle the traffic and plan for the risks.

6. Decision trees – Decision trees present a pictorial way to represent risks. These are used in conjunction with probabilities assigned to various anticipated events to quantify risks.

Risk Analysis: Analyze the risks you have identified, so that you can understand as much as possible about when, why and where they might occur. There are many techniques you can use to enhance your understanding, including system dynamics models, cost models, performance models, network analysis , decision analysis and more. Specify how those areas of uncertainty can impact the performance of the project, either in duration, cost or meeting the users' requirements.

Risk Prioritization: Now that you have itemized all risks, you must use your understanding to assign priorities to the risks. A priority scheme enables you to devote your limited resources only to the most threatening risks. Usually, priorities are based on the risk exposure, which takes into account not only likely impact but also the probability of occurrence.

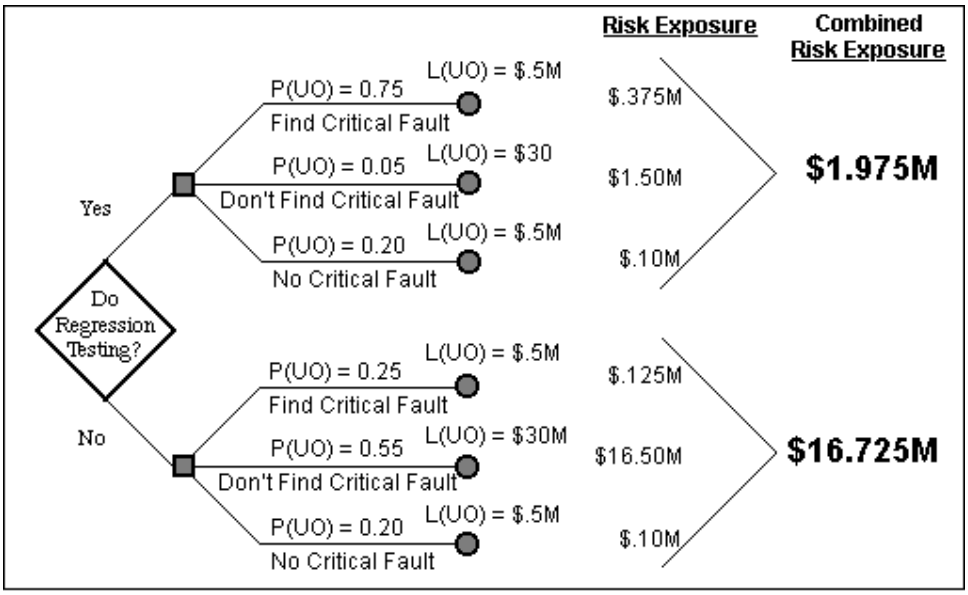


Figure 2. Examples of Risk Exposure Calculation

Figure 2.4 : Risk exposure calculation example

The risk exposure is computed from the risk impact and the risk probability, so you must estimate each of these risk aspects. To see how the quantification is done, consider the analysis depicted in Figure 2.4. You may decide that the risks with highest priority are severe and must be handled before any other ones with lesser impacts .

Risk Control: The notion of control acknowledges that we may not be able to eliminate all risks. Instead, we may be able to minimize the risk, or mitigate it by taking action to handle the unwanted outcome in an acceptable way. Therefore, risk control involves Risk Reduction, Risk Management Planning and Risk Resolution.

Risk Reduction: There are three strategies for risk reduction[19]:

1. Avoiding the risk, by changing requirements for performance or functionality
2. Transferring the risk, by allocating risks to other systems or by buying insurance to cover any financial loss should the risk become a reality
3. Assuming the risk, by accepting it and controlling it with the project's resources

To aid decision-making about risk reduction, we must take into account the cost of reducing the risk. We call risk leverage the difference in risk exposure divided by the cost of reducing the risk. In other words, risk reduction leverage is (risk exposure before reduction-risk exposure after reduction)/(cost of risk reduction).

If the leverage value is not high enough to justify the action, then we can look for other, less costly or more effective reduction techniques. In some cases, we can choose a development process to help reduce the risk. For example, prototyping can improve understanding of the requirements and design, so selecting a prototyping process can reduce many project risks.

Risk Management Planning: For all those risks which are deemed to be significant, we should have an emergency plan in place before it happens . This approach is called Proactive risk planning. We prepare a RMMM (Risk Mitigation, Monitoring and Management)Plan, which documents all work performed as part of risk analysis and is used by the project manager as part of the overall project plan.

Risk Resolution: Risk resolution is a means of minimizing the impact of a risk. Take whatever actions are possible in advance to reduce the effect of risk. It is better to spend money on mitigation than to include contingency in the plan. Risk mitigation can also be called the back up plan to combat a risk. Risk Monitoring is a project tracking activity that is used to access whether predicted risks do, in fact, occur ; to ensure that risk aversion steps defined for the risk are being properly applied and to collect information that can be used for future risk analysis. Risk Re-assessment helps in identifying the new risks and updating the old ones.

To summarize the risk management activities we can say,

Risk Assessment in a project is the most difficult phase of all to carry out. Constraints are usually difficult to remove, though they are important to understand. For instance, a constraint that the project must be finished in time to reflect a new piece of legislation is easy to understand. Manpower constraints are often more uncertain, such as the availability of skilled staff at the critical phase of the project. Risk Assessment is done to explore each task in the project plan and seek uncertainty. It is essential to be absolutely clear in thinking.

2.2.3 Risk Categories

When risks are analyzed it's important to quantify the level of uncertainty and the degree of loss associated with each risk. To accomplish this, different categories of risks are considered [1][16]: -

1. ***Requirements risk:*** Unclear or uncertain requirements introduce large risks. This is the most common type of risk and is probably responsible for most failed or delayed projects. Competitive forces, business agreements with new partners force the organization and its software systems to change. Besides this, users find it hard to visualize software until they see it. This makes requirements highly unclear and subject to change. The best way to handle this risk is to adopt a flexible development process.
2. ***Technology risk:*** You may find (usually fairly late in the process!) that the technology being used is unable to satisfy the system requirements. For example we could assume that the database we use is transactional but when actually building the system we find that it is not transactional under certain circumstances that could happen frequently. This will have a big impact on the system we deliver. A flexible development process that allows us to modify system designs or platforms late in the process can be helpful in mitigating this risk.
3. ***Business risk:*** These are risks introduced by business decisions [21]. For example, a deal with a vendor does not get signed in time to use the platform that we will deploy the application on. A conflict with a partner who supplies part of the solution prevents us from proceeding with the project or the partner goes out of business. Besides "due diligence" when negotiating with such partners, flexibility in our development process will help work around these risks if they do happen.

4. ***Political risk:*** These are the most difficult to overcome. Large organizations tend to behave like large families. Members are busy jockeying for power and influence over each other. Unfortunately, your project may be threatening to some groups and they will protect themselves. This may mean that people do not co-operate, budgets get cut or projects get cancelled. A contingency plan for this sort of risk is hard to define beforehand since they can become an embarrassment. Plus if people know your contingency plans they can still sabotage you.
5. ***Resource risk:*** You do not get the people, money, facilities or equipment when needed [15]. This is something that probably all of us have experienced. It usually has bad affects on the project's schedule and the project team's health! Planning for

this may involve identifying alternate resource sources that may be able to help. For example another team may be willing to share some of their servers till yours come in.

6. ***Skills risk:*** Your team does not have all the requisite skills to do the job. For example they may be unfamiliar with the technology used. Or they may not be familiar with the business process and so on. Bringing in external consultants who can help get you quickly up to speed can mitigate this risk. For example, on a project to migrate from one application server to another, we brought in a person who was an expert on the second application server to mentor us.
7. ***Deployment & support risk:*** The project may not be deployable when ready since the required infrastructure is not in place. The Support team may not be ready for training or are over stretched. Working closely with the Deployment & Support teams can help reduce this risk. Often this happens because these folks have no idea what you need or are doing.
8. ***Integration risk:*** Most applications need to integrate with other applications. Miscommunication and misunderstandings cause systems not to share accepted interfaces and they do not work together as expected. Communication is key to reducing this risk. A flexible development process is a great help since one can try different designs in case of intractable issues.
9. ***Schedule risk:*** Schedules may contain conflicts, components are not available when necessary, the delivery date is happening at an extremely busy time. Communication between all the interested parties can help reduce this risk. For example we may find that timing a product upgrade at the end of a quarter is not a good idea. The sales folks are trying to reach their quotas and they do not want to re-price and re-educate customers just when they are about to close the deals. Good project planning is also important to prevent such conflicts from happening.
10. ***Maintenance & enhancement risk:*** The project cannot be maintained and enhanced properly because of inadequate documentation, the support team was not properly trained or the technical platform is obsolete. Good planning and time spent for training and documentation can help reduce these risks. A flexible development process if not properly implemented can make this problem worse. It is important that time be budgeted for training, documentation and support.
11. ***Design risk:*** Bad design decisions have an impact on the system performance or it ability to satisfy the requirements [6]. A bad design could make the software

unusable or useless for the user. Having a flexible development process that can accommodate user input and changes late in the process can reduce this risk.

12. ***Other risks:*** This is a catch all for risks that are hard to foresee and predict. Examples could be a hurricane shutting down your offices for a week, a fire in the building, a development server crashing, virus attacks and so on. Since they are hard to foresee they can be hard to plan for. However, many of these involve a loss of resources or their unavailability. We can arrange for back up locations to handle such situations. For example people can be set up to work out of home if needed, project data is stored in more than one location, we can get a spare server from another team and so on.

2.2.4 Risk Elements and Risk Factors

Risk Elements *:* The three major elements of software risk are technical, cost and scheduling problems. Technical risks, such as maintainability or reusability, are associated with the overall performance of the software system. Cost risks, such as budget or fixed costs, are associated with the cost of the software system. Scheduling risks, such as overlapping projects with shared resources, are associated with completion of the software system during development.

Risk Factors: Software risk factors can be thought of as more specific subcategories of the three general risk elements. Risk factors are more closely related to software issues. A software risk factor can be a subcategory of more than one risk element Each software risk factor can have an influence on each risk element. That influence can be categorized as low, medium, or high. Ten important risk factors can be categorized as:-

1. ***Organization.*** The maturity of the organization's structure, communications, functions, and leadership.

2. ***Estimation.*** The accuracy of the estimations of resources and schedules needed during software development and their costs.

3. ***Monitoring.*** The ability to identify problems.

4. ***Development Methodology***. The methods by which the software is developed.

5. ***Tools.*** The software tools used when software is developed.

6. ***Risk Culture***. The management decision-making process in which risks are considered.

7. ***Usability***. The functionality of the software product once it is delivered to the end-user or customer.

8. ***Correctness***. Whether the product suits the customer's needs.

9. ***Reliability***. How long the software performs for the customer without bugs.

10. ***Personnel***. The number of people used in development and their abilities.

Each factor can have a different influence on each element. For instance, using a poor estimation technique may have little technical impact on a software project but lead to faulty prediction of the size and cost of a project, and therefore, greatly extend the development schedule. Software risk factors can also be classified by their importance, either major or minor, to product or process risk. This classification helps in identifying the critical risks so that proactive strategy can be taken for mitigating those risks before other less critical risks. Product risks can be handled for that product/ project only while process risks indicate serious care to be taken to improve the process.

2.2.5 When to use Risk Management

Setting up a project is like planting a tree - people often neglect to make certain that the roots are well laid, the tree is supported, it has plenty of suitable soil in which to grow, and is cared for and watered copiously in the early stages. Money invested in reducing risk in the early stages of a project is money well invested. Any risks incurred during the project have to be diagnosed and fixed. Also things have already gone wrong by the time the problem is detected and this will add to the costs. Figure 2.5 illustrates the way costs of correcting risks at the initial start up phase of a project balance against costs of correcting and managing a failure. The rate of increase in cost of risk is exponential and any risks that can be reduced or eliminated during the start up phase will pay a generous dividend in limiting the total project cost. It is much better to reduce the risks at the start

up phase of the project than to allow a contingency on a basis that things are bound to go wrong, but we don't know what!

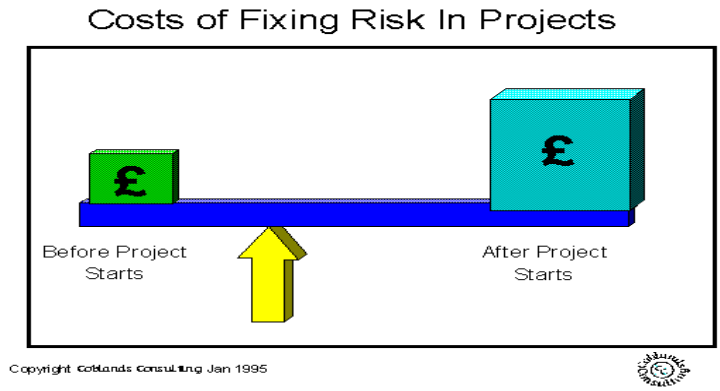


Figure 2.5: Costs of fixing Risk in projects

Thus, we can conclude that risk management should be an umbrella activity which should be started in the very early stages of the project development. We should include risk metrics in all the life cycle phases of software project development, so that the risks should be identified as early as possible.

2.3 Conclusion

This chapter focused on theoretical background of Software Project Risk Management , the risk management steps , risk categories , risk elements and factors and the time and cost of identifying and mitigating the risks during the software project life cycle. The stages of risk management (identify, analyze, plan, track, control, communicate) allow management, the project team, and the customer to increase their confidence that there will be both, no nasty surprises and that the risks will be mitigated effectively. Risk categories help in identifying various risks. Risk factors help in studying the impact of various types of risks on risk elements.

The next chapter discusses software metrics and their importance.

CHAPTER 3 SOFTWARE METRICS

Current software management is ineffective because software development is extremely complex, and we have few well-defined, reliable measures of either the product or the process to guide and evaluate development. Improvement of the management process depends upon improved ability to identify, measure and control essential parameters of the development process. This is the goal of software metrics – the identification and measurement of the essential parameters that affect software development.

3.1 Background

The best motivation for metrics comes from a quotation by Lord Kelvin[17]:

"*When you can measure what you are speaking about, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely in your thoughts, advanced to the stage of science.*"

There are two components touched upon in the above sentence: First is that metrics are measurements to measure your progress (in – process metrics). The second is the measurement of how well you have achieved the goals (end – result metrics).

All engineering disciplines have metrics (such as metrics for density, weight, wavelength etc.) to quantify various characteristics of their products. In software projects, the most fundamental question is "how big is the program? ". Without defining what big means, it is obvious that it makes no sense to say, "this program will need more testing than that program" unless we know "how big they are relative to one another". Size is not obvious for the software [17]. Metrics must be objective in the sense that the measurement process is algorithmic and will yields the same results no matter who applies it. The groundwork of software metrics was laid in the seventies. Most of the metrics are aimed at getting empirical laws that relate program size to expected number of bugs, expected number of tests required to find bugs, test technique effectiveness, resource requirement, release instant, reliability and quality requirement etc.

3.2 Definition

Software metrics can be defined as "*The continuous application of measurement based techniques to the software development process and its products to supply meaningful*

and timely management information, together with the use of those techniques to improve that process and its products"[24].

Software metrics are all about measurement which, involve numbers, to improve the process of developing software and to improve all aspects of the management of that process[19]. The application areas of software metrics are: cost and size estimation techniques ,controlling software development ,prediction of quality levels for software, to provide quantitative checks on software design and to provide information to management.

Good metrics should facilitate the development of models that are capable of predicting process or product parameters, not just describing them. Thus, ideal metrics should be[31]:

- Simple, precisely defined- so that it is clear how the metrics can be evaluated
- Objective, to the greatest extent possible
- Easily obtainable (i.e. at reasonable cost)
- Valid – the metrics should measure what it is intended to measure
- Robust – relatively insensitive to changes in the process or product

In addition, for maximum utility in analytic studies and statistical analysis, metrics should have data values that belong to appropriate measurement scales[31].

3.3 Classification of Software Metrics

Software metrics may be broadly classified as [31][16]

- ***Product metrics or Process metrics*** - Product metrics are measures of the software product at any stage of its development, from requirements to installed system. These measure the complexity of the software design, the size of the final program or the number of pages of documentation produced. The process metrics are the measures of the software development process, such as overall development time, type of methodology used or the average level of experience of the programming staff.
- ***Objective or Subjective*** – Objective metrics should always result in identical values for a given metrics as measured by two or more qualified observers.For example, LOC or development time. For subjective metrics, even qualified observers may measure different values for a given metric, since their subjective judgment is involved in arriving at the measured value. Classification of the software as

"organic", "semidetached" or "embedded" as required in COCOMO cost estimation model and level of programmer experience are examples of subjective metrics.

- ***Primitive or Computed*** – Primitive metrics can be directly observed while computed metrics are computed from other metrics. For example, number of defects in unit testing is primitive product metric while LOC per person-month is a computed product metrics.As computed metrics are combinations of other metrics values, they are often more valuable in understanding or evaluating the software process that simple metrics.

3.4 Commonly Used Software Metrics

This section discusses some of the commonly used software metrics:

3.4.1 Size Metrics: A number of metrics attempt to quantify software size. Some of them
are discussed here:

- ***Lines of Code***- LOC is the most widely used metric for program size. It seems to be easily and precisely definable; however, there are a number of different definitions for the number of lines of code in a particular program. These differences involve treatment of blank lines and comment lines, non- executable statements, multiple statements per line, and multiple lines per statement[16].The most common definition of LOC seems to count any line that is not a blank or comment line, regardless of the number of statements per line.
- ***Function Points***- The approach is to compute the total function points (FP) value for the project, based upon the number of external user inputs, inquiries, outputs and master files. The value of FP is the total of these individual values, with the following weights applied: inputs-4, outputs-5, inquiries-4 and master files- 10. Function points are intended to be a measure of program size and thus, effort required for development.
- ***Demarco's Bang Metrics***- DeMarco defines system Bang as a function metric, indicative of size of the system. In effect, it measures the total functionality of the software system delivered to the user. Bang can be calculated from certain algorithm and data primitives available from a set of formal specifications for the software. Since Bang measures the functionality delivered to the user, DeMarco suggests that a reasonable project goal is to maximize "Bang per buck" – Bang divided by the total

project cost.

3.4.2 Complexity Metrics: Numerous metrics have been proposed for measuring program complexity . The examples discussed below are some of the better known complexity metrics:

- ***Cyclomatic Complexity (v(G))***- Given any computer program, we can draw its flow graph G, wherein each node corresponds to a block of sequential code and each arc corresponds to a branch or decision point in the program. The cyclomatic complexity of such a graph can be computed by v(G)= e-n+2 where e is the number of edges and n is the number of nodes in the graph G. McCabe proposed that v(G) can be used as a measure of program complexity and hence, as a guide to program development and testing.
- ***Knots***- The concept of knots is related to drawing the program control flow graph with a node for every statement or block of sequential statements[31]. A know is defined as a necessary crossing of directional lines in the graph. The number of knots is a measure of program complexity.
- ***Information Flow***- The information flow within a program structure may also be used as a metric for program complexity. These are proposed by Kafura and Henry. The method counts the number of local information flows entering (fan-in) and exiting(fan-out) each procedure. Program complexity c is given as: c=[procedure length]. $[\text{fan-in.fan-out}]^2$.

3.4.3 Halstead's Product Metrics: Most of the product metrics proposed have applied to only one particular aspect of the software product[31].Halstead's software science proposed a unified set of metrics that apply to several aspects of programs,as well as to the overall software production effort:

- ***Program Vocabulary*** – Halstead theorized that computer programs can be visualized as a sequence of tokens, each token being classified as either an operator or operand. Vocabulary n of a program is defined as: $n= n_1+n_2$ where n_1 = number of unique operators in program and n_2 = number of unique operands in the program
- ***Program Length***- Program length N can be defined as the count of total number of operators and operands in the program, $N = N_1+N_2$ where N_1 = Total number of operators in program and N_2 = Total number of operands in the program
- ***Program Volume***- Another measure of program size, $V=N. \log_2 n$. Since N is a pure

number, the units of V can be interpreted as bits, so that V is a measure of the storage volume required to represent the program.

3.4.4 Quality Metrics: One can generate long lists of quality characteristics for software- correctness, efficiency, reliability, maintainability etc[35]. Some of the quality metrics by Boehm, McCall and others are discussed below:

- ***Defect Metrics***- The number of defects observed in a software product is a metric of software quality. The indirect measures proposed for finding the no. of defects are number of design changes, number of errors detected by code inspectors and number of errors detected in program tests
- ***Reliability Metrics***- Reliability refers to the probability of software failure, or the rate at which the software failure will occur. Mean Time to Failure (MTTF) is a reliability metric.
- ***Maintainability Metrics***-According to experiments done by Rombach, software complexity metrics can be used to predict the maintainability of a software.

3.5 Various Models for Software Metrics

Metrics are defined or used in conjunction with a particular model of the software development process. Most proposed software models have resulted from a combination of intuition about the basic form of relationships and the use of empirical data to determine the specific quantities involved. Some of them are discussed below:

3.5.1 Theory Based Models: Few of the proposed models have substantial theoretical bases . two of them are discussed below:

- ***Rayleigh Model*** (by L.H. Putnam) developed a model of the software development process based upon the assumption that the personnel utilization during program development is described by a Rayleigh – type curve such as the following[31][16]:

  $y= (K\ t\ e^{-t2/2\ T\ 2}) / T^2$,

  where y= number of persons on the project at time t,

  K = the area under the Rayleigh curve, equal to the total life cycle effort in person-years,

  T = development time.

  Putnam assumed that either the overall staffing curve or the staffing curves for individual phases of the development cycle can be modeled by an equation of this

form. He developed the following relationship between the size of the software product and the development time , $S= CK^{1/3} T^{4/3}$, where S = number of source LOC delivered;

K = life cycle effort in person-years and C= a state of technology constant

- ***Software Science Model*** (by Halstead)The software science equations can be used as a simple theoretical model of the software development process. The effort required to develop the software is given by the equation: E=V/L which can be approximated by:

  $E=(n_1 n_2 [n_1 \log n_1 + n_2 \log n_2]\log_2 n)/ 2n_2$.

  T = E/S (where T is programming time and S is Stroud number)

  Also, $T= (N^2 \log_2 n)/ 4S$ if only the value of the length, N is known and n can be obtained from $N=n \log_2(n/2)$

3.5.2 Composite Models :As experience has been gained with the previous models, a number of more recent models have utilized some combination of intuition, statistical analyses and expert judgement. These have been labeled as composite models .Three of them are discussed here:

- ***COCOMO*** **Model** (by Boehm)- This is probably the best known and most thoroughly documented of all software cost estimating models. It provides three levels of models: Basic, Intermediate and Detailed[19][31][16]. Boehm identifies three modes of product development- organic, semidetached and embedded - that aid in determining the difficulty of the project. The developmental effort equations are all of the form:

  $E= aS^b m$, where a and b are constants determined for each mode and model level;

  S is the value of source LOC and m is a composite multiplier, determined from 15 cost-driver attributes. Boehm suggests that the detailed model will provide cost estimates that are within 20% of actual values 70% of the time.

- ***SPQR Model-*** (by Jones) T. Capers Jones has developed a software cost estimation model called the Software Productivity, Quality and Reliability model. The basic approach is similar to that of Boehm's COCOMO model. It is based on 20 reasonably well defined and 25 not-so well defined factors that influence software cost and productivity. SPQR requires user responses to more than 100 questions

about the project in order to formulate the input parameters needed to compute development costs and schedules. Jones claims that it is possible for a model such as SPQR to provide cost estimations that will come within 15% of actual values 90% of the time.

- ***ESTIMACS***- (by Rubin) The ESTIMACS model addresses three important aspects of software management - estimation, planning and control. The ESTIMACS system includes- System development effort estimator, Staffing and cost estimator, Hardware configuration estimator, Risk estimator and portfolio analyzer.The ESTIMACS system has been in use for only a short time. In the future, Rubin plans to extend the model to include the maintenance phase of the software life cycle. He claims that estimates of the total effort are within 15% of the actual values.

3.5.3 Reliability Models A number of dynamic models of software defects have been developed. These models attempt to describe the occurrences of defects as a function of time, allowing one to define the reliability, R and mean time to failure(MTTF). An example is discussed here:

- ***Reliability Model by Musa*** – It makes four basic assumptions:
    - Test inputs are random samples from the input environment
    - All software failures are observed
    - Failure intervals are independent of each other
    - Times between failures are exponentially distributed.

  Based upon these assumptions, the following relationships can be derived:

  $d(t)= D(1-e^{-bct})$, where D is the total number of defects, b and c are constants that must be determined from historical data for similar software, d(t) is the number of defects discovered at time t.

  Mean Time to Failure, $MTTF(t) = (e^{-bct})/ cD$.

3.6 How to use Software Metrics

The use of metrics should be an umbrella activity in the software project development because every activity – be it an umbrella activity like software configuration management or in- stream activity like project initiation- will have relevant metrics that need to be identified. Every software project has a certain set of objectives to be met. There are short term objectives, e.g. the specific deliverables for the particular project and there are long term objectives, e.g. how is the project tied to or company vision? Any metrics

program or measurements should provide gauges that measure our progress towards both short term and long term objectives.

There are five questions that one has to answer for a metrics program [17]:

1. What is your goal or where do you want to go?
2. What is your current position?
3. Knowing where you are and where you want to go, what steps should you take?
4. How do you measure your progress?
5. What corrective actions do you take when you see that your progress deviates from the expected course?

The strategy for a successful metrics program falls in line with Deming's PDCA Quality model, shown in figure 3.1

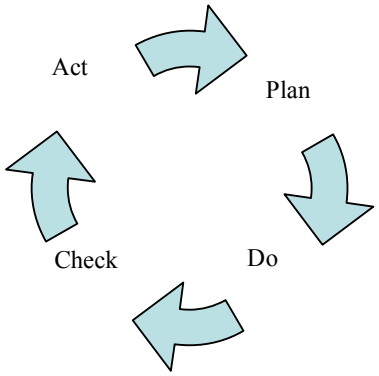


Figure 3.1 : The Deming PDCA cycle for metrics[17]

The acronym P-D-C-A stands for Plan- Do- Check- Act. For any project or activity, you plan what you want to achieve, based on your current position and your eventual destination. While making this plan, you also establish measurement criteria. Then you Do or carry out the plan. This entails carrying out the necessary steps for the actual execution and also carrying out the measurements for the health check. The Check phase compares the measurements made against the expected norms set in the Plan phase. If things are found to be out of line, corrective actions need to be taken. The correction actions are taken in the Act phase and we start all over again by refining the Plan.

3.7 Steps to be taken for a Successful Metrics program

Metrics though used, do not always guarantee success for a software project. Therefore, the metrics require a well planned strategy for implementing the concepts. A specific approach which can be used to plan the software metrics program is Goal/Question?Metric (GQM) paradigm developed by Basili[16]. The steps for a sound metrics program are given below:

1. **Decide what you want to measure and plan how you are going to measure-** The purpose of measurements is that they should lead to achieving goals faster, more economically, with superior quality output and better employee / customer

satisfaction. So, we should identify the entity directly relevant to the goals of the project/organization; easily measurable and controllable and essential to be measured[31]. Select the model and metrics and the method for its measurement[16]. Estimate the costs of implementing a metrics program.

2. **Set Targets and track them-** After identifying the entities to be measured, set some objectives or targets for these i.e. metrics and models available should be compared with respect to their apparent ability to meet the objectives identified[16]. Models identified as capable of meeting the objectives of the organization should be compared in terms of data requirements and associated cost. For tracking the progress, data must be gathered and stored in a database and maintained throughout the life cycle.
3. **Understand variability and work towards minimizing it-** Variability refers to how the key metrics vary over time or over different projects. Certain amount of variability is inevitable, but it is important to ensure that the variability is within "reasonable limits" (lower control limit LCL and upper control limit UCL) and hovers round an expected mean value[17]. When the variability exceeds these limits, we should understand exactly why this happened and how we can minimize such a wide variation in future.

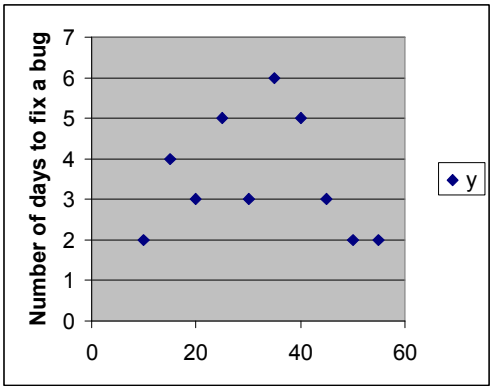


Figure 3.2 Distribution of a metric between UCL (7) and LCL(1): Mean (4) [17]

4. **Act on data and strive for continuous improvement-** The job of metrics collection does not end with metrics collection and analysis. The data collected should be acted upon and corrective action should be taken, resulting in continuous improvement. Conscious business decisions should be taken based on data and analysis of metrics[17]. Most models in use require a calibration process, adapting the values of multiplicative constants, to the empirical data for the environment of application. Review these calibration constants based upon the results achieved [16].

5. **Commitment at all levels in organization-** Implementing a successful metrics program requires whole-hearted co-operation from all levels in the organization. Hence, people issues play a very significant role. Management commitment is essential for the success of metrics, that is, all top managerial people should also consider reporting about metrics essential. Only junior staff should not be forced for metrics. The operational issues of metrics should be understood be everyone. As the metrics related with product may reflect some shortcomings in the performance, the messenger should not be shot for bad results. Also, the privacy laws must also be followed in case of people metrics. Metrics are for monitoring the progress and performance and should be used for correction and not for punishment.
6. **Revise the Metrics program:** To stay successful, metrics should be flexible and growing. For this, the metrics program requires strategic planning[4]. So, periodically revisit the metrics and measurement plan to see whether changes are required due to changes in goals, priorities of the goals, process, audience or maturity of the process. Make corresponding changes in the metrics program to update it.

3.8 How to Avoid common pitfalls in Metrics program

When the people from top to bottom are involved in the metrics programs, it may be possible that we are not gaining as much as we intended to while using the metrics program. The common reasons to be watched out for the same are given below [17]:

1. **Do not over engineer -**Metrics collected may be more than required. We are lost in collecting useless metrics just for the sake of collecting these. The important metrics indicating alarms may be just missed because of excess of metrics collected. So, attention should be focused only on important and required metrics. This can be implemented by taking feedback regarding their utility and updating the metrics used over a period of time.
2. **Include cost and time for metrics collection** – Organizations generally do not include the cost and time involved in collecting the metrics and continuing costs to operate the metrics program. This may result in under estimation of budget of the project in the beginning and wrong estimate of total cost of the project in the end. So, initial implementation and continuing costs of the metrics program must be included in budget.
3. **Design metrics before tool-** In order to measure the metrics, the tool is designed first and then it is learned how it will be used for measurement. This is a wrong approach

and results in frustration and wasted efforts [17]. First of all, the way to measure the metrics should be considered while design of the metrics and tool for measurement should be designed accordingly.

4. **Clarify the responsibility for collecting metrics-** Usually the people who collect the metrics are not the same people who analyze the metrics used and present the findings. This may result in ineffective data collection or analysis. Logistical details of how the metrics are collected, by whom, at what frequency, the details of how the metrics are analyzed, at what frequency, who are the recipients of the analysis etc. should be clearly defined and understood by everyone [17]. For this proper documentation of the definition and the purpose of the metrics should be done while designing it.
5. **Metrics as part of the actual work-** Metrics collection should ideally be a natural by-product of work. It is important to set the proper infrastructure in place so that the metrics progress is tightly integrated into the project life cycle activities and does not involve additional steps.

3.9 Benefits of Risk Metrics

The entire development organization also enjoys benefits from risk management and risk metrics:

- Sharing what does and does not work to control risks across multiple projects helps projects avoid repeating the mistakes of the past.
- Members of the organization can pool their experience and identify opportunities to control our most common risks, through education, process improvement, and application of improved software engineering and management techniques.
- Over time, you can build a checklist or database of risk items and mitigation strategies from multiple projects that can help future projects look for the vipers that might be lurking underfoot.

3.10 Conclusion

Metrics is an all-pervasive and all-encompassing activity that takes place throughout the project life cycle. It requires a very delicate and fine balance between engineering expertise and human finesse. A metrics program pays off, but not without some investment of both time and resources. This chapter discussed the definition of a software metrics, characteristics of a good software metric and various software metrics

and models used commonly. How the metrics program should be started so as to execute remaining activities of the project successfully was also discussed. It also discussed the problems faced while using metrics which make the effect less than intended. The chapter concluded with the discussion on benefits of risk metrics which clearly emphasize the need of using them.

The next chapter discusses the research approach used in present work .

CHAPTER 4 RESEARCH APPROACH

This thesis aims at analyzing various risk measurements techniques used currently in different software development companies, by studying the current practices used by them and then proposing new risk metrics for better results. This chapter focuses on the research approach and methodology adopted for the same. Along with that it gives an understanding of the problems faced during the development of present work.

4.1 Study methods

Different techniques and tools can be used to gather information , some of the most popular of them are given below:

1. Documentation reviews: Performing a structured review of project plans and assumptions , prior project files and other information is generally the initial step taken by the researchers.

2. Information gathering techniques: Brainstorming, the Delphi technique and interviewing are the techniques used for the purpose of risk identification. These techniques are discussed below in brief:

- **Brainstorming:** Brainstorming is probably the most frequently used risk identification technique. The goal is to obtain a comprehensive list of risks that can be addressed later in the qualitative risk analysis processes. Using brainstorming, a meeting is organized with a multidisciplinary set of experts. Under the leadership of a facilitator, these people generate ideas about project risk. The brainstorming meeting proceeds without interruption, without expressing judgment or criticism of other's ideas and without regard to individual's status in the organization. Sources of risk are identified in broad scope and posted for all to examine during the meeting. Risks are the categorized by type of risk and their definitions are sharpened .

  Brainstorming can be more effective if participants prepare in advance, the facilitator develops some risks in advance and the meeting is structured by project segment and risk category.

- **Delphi Technique:** The Delphi technique is a way to reach a consensus of experts on a subject such as project risk metrics. Project risk experts are identified but participate anonymously. They do not meet face- to- face.

  A facilitator uses a questionnaire to solicit ideas about the important project risks. The responses are submitted and put into risk categories by the facilitator. These risks are then circulated to the experts for further comments. Consensus on the main project risks may be reached in a few rounds of this process. The Delphi technique helps reduce bias and keeps any person from having under influence on the outcome.

- **Interviewing:** Risks can be identified by interviews of experienced project managers or subject matter experts. The person responsible for risk identification identifies the appropriate individuals, gets brief from them on the project, provides information such as the work breakdown structure (WBS) and the list of assumptions. The interviewees identify risks on the project based on their experience, project information and other resources they find useful.

- **Diagramming techniques:** Diagramming techniques may include:
  - Cause –Effect diagrams (also known as Ishikawa or fishbone diagrams) – useful for identifying causes of risks.
  - System or process flowcharts- show how various elements of a system interrelate and the mechanism of causation
  - Influence diagrams- a graphical representation of a problem showing causal influences, time ordering of events and other relationships among variables and outcomes.

4.2 Method Used

Qualitative research methods employed across a spectrum of organizational types can improve our picture of the criteria for success and the ultimate future state of organizational project risk management maturity in various environments. Such methods not only elaborate our idea of success, they help in identifying prerequisites for different project management outcomes in different organizational environments. Analysis of qualitative data provides a basis for understanding the different drivers and organizational enablers of success.

The survey was conducted using the following steps:

1. Developing a questionnaire for survey purpose
2. Distributing the questionnaire to various software companies
3. Gathering the data
4. Analyzing the data gathered

4.2.1 Developing a Questionnaire for survey

Developing a survey questionnaire is the first step in identifying candidate success criteria, including factors that executives, senior management, and project managers consider indicative of success. The intent of the survey is to explore and identify factors, issues, root causes and perceptions in the language of the person running projects, managing business units and other stakeholders. An organizational assessment factored into the survey tells a story about how people perceive software project risk management is being applied as compared to how they envision it should be applied. The survey also captures hard data that might be quantified; that is, output in terms of elements like quality, cost, time and customer satisfaction. The purpose of collecting this data is to isolate prerequisites for customers. Qualitative research analysis methods applied to the data allows a classification of success criteria by trying to identify hidden variables and reduce a large number of variables into a smaller number of dimensions.

4.2.2 Data Gathering – The data received from the interviewees is to be collected. This can be divided into primary and secondary data gathering.

4.2.2.1 Primary Data Gathering

Of the various information – tools available, interview and questionnaire are effective techniques for surveying and eliciting information about any subject.

Interview and questionnaires are of two types:

1. **Structured** : In this approach, questions are presented with exactly the same language and in the same order to all subjects.

2. **Unstructured** : This is a relatively non-directive information gathering technique. It allows respondents to answer questions freely in their own words. The responses are spontaneous rather than forced.

The alternative used for the present work was structured. Thus, a structure for the questions to be asked was prepared by taking data from different sources such as PMBOK, books, journals and the internet. A questionnaire of 45 questions that concerned the risk management process was designed. As the next step,

the quality managers of various companies were approached. To get a better overall view of the software development process, people were chosen from different organizations.

A questionnaire was sent to the interviewees. The questions were quite open and designed to let the interviewees' start thinking of the different activities performed and the routines that were followed in their companies. The questions covered a number of topics generally and Risk management specifically. The correspondence was through e-mails and phone calls for most of the companies. As the information gathered had confidential data, therefore, the names of the organizations is not being quoted.

The questionnaire was distributed among 20 software developing companies working for different projects at Chandigarh, Gurgaon, Noida, Bangalore and Delhi. The questionnaire has been attached as Appendix-A.

4.2.2.2 Secondary Data Gathering

Research papers were surveyed, various articles and websites were visited. A number of books were also referred for clearing the risk management concepts and designing the questions for the questionnaire so that every aspect of risks could be taken into consideration while designing new risk metrics. The list of references is attached as Appendix C - references.

4.2.2.3 Analysis of the gathered data

The data gathered was analyzed and results were formulated in the form of tables, graphs , pie charts for easy understanding . The values averaged for some data patterns indicated the general trends followed.

4.3 Problems faced during survey

In the survey, primary and secondary data was required to be gathered and then analyzed. Secondary data was easily available as the internet and library access was available. Primary data gathering was a bit difficulty because of the following reasons:

- Software companies are not ready to reveal their confidential information due to tough competition.
- As software market is very sluggish and due to very frequent lay offs, employees are afraid of revealing any information or techniques used by the company.
- No one wants to spare time especially for research purpose.
- Industry interaction is less being from academic side.

4.4 Conclusion

This chapter focused on the research approach taken up for the study. Different techniques can be used but most practical and effective technique for present work was survey technique. It was accomplished with the help of a questionnaire which was sent to different companies for understanding their risk management techniques and risk metrics used by them.

In the next chapter we will analyze the survey outcome.

CHAPTER 5 RESEARCH FINDINGS

Risk management is dynamic and is carried out throughout the software development process. It requires active participation from the entire team from management down to the lowest working level. Software Companies follow different strategies to combat risk. The purpose of this survey thesis was to study those practices and suggest some new risk metrics that can be even more helpful in managing the risks in software projects. This chapter discusses the outcome of survey conducted, for the study.

5.1 Background

The importance of effective risk management for project success is not disputed. Considerable attention has been given to ensuring comprehensive identification and objective assessment of project risks, to provide a clear understanding of the extent of risk exposure faced by a project. Many techniques have been developed to support these stages in the risk process, which work well when used properly. Software projects fail frequently because companies don't put into practice the established principles and techniques for managing project risk.

The complex process of software acquisition encompasses most, if not all, aspects associated with software risk management. Thus, it seems natural to focus on the entire life cycle of the software acquisition process in developing a holistic vision of risk management. Risk management of software engineering cannot be restricted to any subset or a single phase of the life cycle of software development.

To study the measures taken up by different software companies I had surveyed them with the help of a questionnaire. The survey was successful and most of the Companies did respond to it. Many interesting results came into picture after the analysis of the answers.

5.2 Analysis

The analysis of the data gathered during the survey helps in identifying the key areas of concern. It helps identify the hidden parameters and to provide direct and indirect conclusions for the problem that is studied using the survey. Here, the current risk management practices used by different software companies is studied and analyzed.

5.2.1 Technique used

Questions which required explanation have been summarized as a common outcome for all companies. For questions with 'yes' or 'no' replies have been averaged or percentage has been calculated.

5.2.2 Background of participants

In order to validate the survey results, respondents were asked about their work profile. This information serves two purposes. First, it offers confidence that the survey results represent a well-experienced population. Second, it addresses concerns that may arise regarding bias. The respondents currently represent over 20 software companies located in Chandigarh, Gurgaon, Noida, Bangalore and Delhi. The respondents indicated their experience by software domain.

5.2.3 Analysis result

As stated earlier, the questionnaire was distributed to 20 software companies. On the basis of the answers received , information gathered regarding current development processes, risk management practices and risk metrics is presented in following sections. The questions in the questionnaire had been categorized under four sections as follows:-

1. General category
2. Project development
3. Software Risks
4. Risk Metrics

General Category (Q1 – Q10)

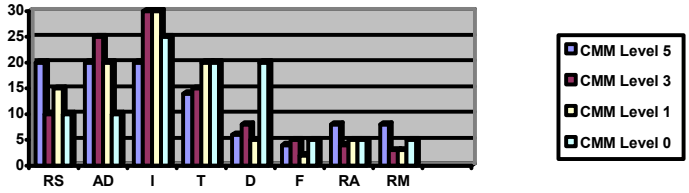
Q1 Specify the maturity level (CMM) followed by the Company.

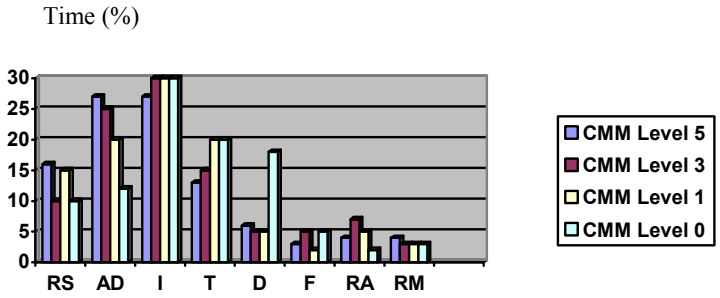
CMM Level 5 – 60%	CMM Level 3 -10%
CMM Level 1- 15%	Non CMM – 15%

Q2 Which process model is used by your company?

Waterfall model is used widely for developing the software(90%). Only 5% companies used prototype or spiral model for software development while rest 5% did not use any process model at all .

Q3 How much time and budget (in percents) is devoted to each of the following activities?





RS - Requirement Specification
AD- Analysis and Design
I - Implementation
T - Testing
D - Documentation
F - Feedback
RA- Risk Analysis
RM- Risk Mitigation

Figure 5.1 Time and Budget devoted to various types of risks(%)

Q4 How is the project schedule being designed in the company? (Specify any tool used)

60% of the companies do not use any tool for scheduling the project. They use experience from previous projects or the customer requirements or priority for scheduling the project. While 40% of the companies used tools for project scheduling. The tool used is Microsoft Project or Some Internally Designed Tool

Q5 How often do projects take longer to complete than estimated (Number of projects that take longer to complete/total amount of projects completed in a specified amount of time)

On an average, 30% of the software projects are delivered late .

Q6 How often are projects finished earlier than estimated? (Number of projects finished earlier than estimated/ total amount of projects completed in a specified amount of time)

Only 5% of the projects finished earlier than estimated as project managers were able to plan them well.

Q7 Are you able to convince your customers about slip of budget or schedule?

Only 60% of the times, the companies could convince the customers about slip of the budget or schedule . In rest of the 40% cases, the companies could not convince the

customer about more cost and time take for development than previously agreed upon .

Q8 Which type of risks frequently occur in software development activity of almost all projects of the company ?

The following table shows the types of risks occurring during software development :

Business	Change in project requirement / Scope and management
Financial	Price hike of software or hardware resources
Personnel	Attrition
Political	Govt. policy changes may effect budget/ custom duties on resources etc.
Resource	Dependency on third party for resources; non familiarity with new resources
Schedule	Under estimation of schedule
Skills	People with required skill set not available
Technical	Hardware software compatibility not tested before buying/using
Requirements	Keep on changing; Misinterpreted; Missing and demanded at time of delivery
Design	Wrong methodology used
Deployment and support	Exhaustive training required to end user; Run time changes required
Integration	Difficulty in integration components from different vendors
Maintenance	Modifications demanded

Table 5.1 Various types of Frequently occurring Risks

Q9 What is the percentage of each risk?

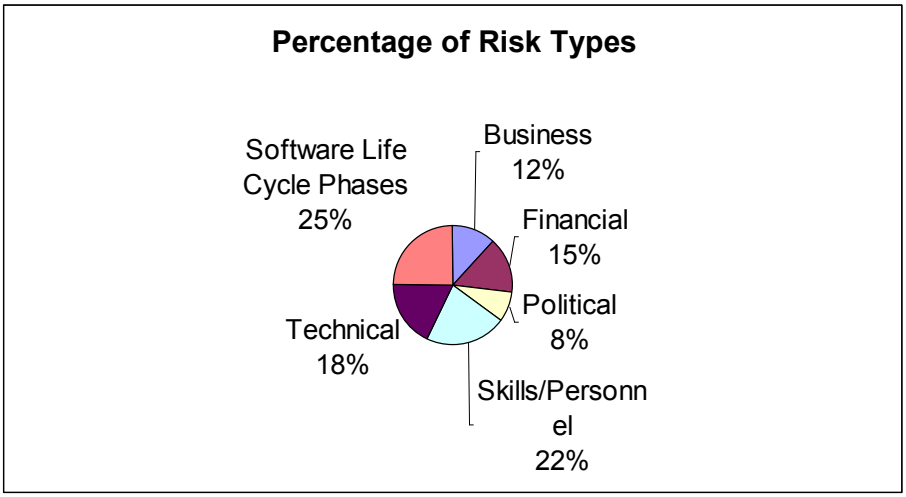


Figure 5.2 Percentage of various types of risks

Q10 In general, are you able to mitigate the Risks for most of the projects?

Only CMM Level 5 companies were able to mitigate the risks for most of the projects while the companies with lower CMM levels or with no maturity level at all were not able to mitigate risks.

Project Development (Q11 – Q20)

Q11 Please specify how are the user requirements interpreted in your company?

- User requirements are gathered by requirements analysis team and User specification

  document is created
- Support engineers take user requirements and make a case study and forward to concerned team
- Through business analysts

Q12 At what stage of development life cycle are the requirements freezed up?

Requirement analysis Phase	50%
Design Phase	30%
Testing Phase	10%
Never freezed (no development model used)	10%

Q13 What control measures exist to manage creeping user requirements?

- Change Control Management procedures

- Sound Configuration Management
- Requirement Stability Index
- Special team look for user requirements which is part of development team
- Designing phase is started only when the final approved and freezed requirement specification is received

Q14 At what level, the failure reports are investigated in your company?

Project Level	70%
Department level	10%
Individual Level	5%
No investigation at all	15%

Q15 List down the areas of uncertainty as applicable to your company/projects. Can you name one of the most probable areas of uncertainty or cause of failure?

Major areas of uncertainty identified are:

- Not much experience with software or hardware
- Political issues
- Personnel (key personnel on leave, key person involved in more than one project)
- Economical issues
- Competition in the market
- Modifications required after delivery of product

Q16-17 Is Risk taken into consideration in project schedule management? Does your company give any formal training for risk handling?

Almost 90% of the companies included risk in project schedule management but only 40% of them provided formal training for handling risk. Thus, though planned and scheduled, it was not properly handled due to lack of training.

Q18 Do you link risk with size of Company/ team/ project?

Only 50% of the cases linked risk with company/project/team size while 50% did not.

Q19 What is the frequency of collecting reports in your company?

Once a week	55%

Once a month	25%
More than 5 per month	20%

Q20 If your key personnel change or leave, what criteria is used to ensure complete hand-over of data?

- Formal Knowledge transfer sessions are held where technical know how is recorded and documented before the key personnel leave
- Training is imparted by the outgoing personnel to the person to whom data is handed over
- Proper backup preparation is done
- Someone whosoever is free is caught and asked to catch all the stuff from the outgoing personnel (non CMM)

Software Risks (Q21 - Q38)

Q21 Does your Company use any tools/ techniques/ processes/strategy for managing Risks? Please Specify.

80% of the companies did not use any tool for managing risks. Only 20% companies used following tools:

- Risk Calculator
- Inhouse developed risk tracking tool
- Risk tracking tool /Template/ Risk ratings
- Post mortem of the finished projects to give feedback

Q22 Is the Strategy used for managing risks proactive or reactive ?

In only 55% of software projects, the strategy used for handling risk was proactive, in 45% cases, it was reactive. Hence more price to combat risk.

Q23 What technique is used by the Company for predicting risks?

Only 35% of the cases, used techniques for predicting risks. Techniques used for predicting risks:

- Historical data
- Business knowledge
- Human Intelligence
- Risks expected are given as input in standard format of tool and reports generated with

  Tool

Q24-25 Do unnecessary features also become a risk factor? Give any unnecessary features which also become a risk factor.

Only in 30% of the cases, unnecessary features like modifiability became risk factors

Q26 Is any risk item checklist created? If yes, what are its contents?

In 60% of the cases, risk item checklist was not created and in rest of the 40% it was created according to taxonomy based questionnaire given by SEI.

Q27 How are team members communicated about project risks?

Regular meetings was the only source used for communicating about project risks where formal reports are used to indicate about risks.

Q28 Do you have a Risk Manager? If yes, what role does he play in your Company?

90% companies did not have the risk manager appointed. Project manager was responsible for handling the risk.

Q29 How does your company define Risk Assessment and Risk Management?

Risks are quantified using probability of occurrence and impact on cost, quality and Schedule and managed according to priority.

Q30 Is any RMMM plan or Risk Information Sheet used?

RMMM Plan	25%
RIS	25%
Neither RMMM plan nor RIS	50%

Q31 What steps are being followed in Risk Management?

Steps followed in risk management (by CMM Level 5 companies)are:

- Risk Identification
- Risk Analysis
- Risk Planning
- Risk Monitoring And Response
- Risk Mitigation Plan
- Regular Follow up With Clients
- Post Mortem Of Finished Projects

Q32 What criteria is used for assessing and analyzing Risks? Do you use any models/processes for assessing risks?

Criteria for assessing and analyzing the risks:

- Deviation from Planned Effort
- Number of bugs caught in different phases of development VS No. of bugs reported after RTM
- Situations specific to projects

Q33 Does high complexity of project mean more risks?

70% of the companies related the risk with project complexity while 30% did not.

Q34,36 Does your Company catalogue data on risks? Does historical data really help in dealing with risks in upcoming Projects? List down the fields of database/risk history table, if used.

80% of the companies (CMM)catalogued data on risk. Risk database fields are:

- Risk Id	- Risk Description
- Probability Of Occurrence	- Time Frame
- Control Factors	- Risk Exposure
- Risk Category	- Risk Profile

- Date Identified	- Risk Duration
- Contingency Plan	

Q35 How are the Risks prioritized? Specify the criteria for identifying the importance/priority of a risk

Risk prioritization was done on the following basis:

- Based on detailed risk prioritization template
- Based on business criticality of risks
- Based on detailed planning to get best possible estimates for risk exposure
- After calculating risk exposure based on it's probability of occurrence and impact on cost quality and schedule
- All the team members vote and prioritize the risks after identification

Q37 How is the team motivated to combat risks ?

Incentives and appreciation letters were used mostly for motivating the team members to combat risks.

Q38 Do you always apply risk mitigation strategies once a risk is known to become a reality?

80% of the companies applied risk mitigation strategy once a risk was known to become a reality

Risk Metrics (Q39 - Q45)

Q39-40 Are any Risk metrics used for measuring risks? What are the various risk metrics used in your company?

Only 60% of the companies used risk metrics some of which are:

- Time spent in different tasks VS Time Planned
- Bugs reported in different tasks VS Estimated number of bugs

- Any new requirement and its overall impact on final deliverables

Q40 What is the frequency of collecting these?

Risk metrics were collected either monthly or at each milestone

Q41 Is there some specific person responsible for collecting risk metrics?

Only in 50% of the cases, a person was held responsible for collecting risk metrics and

he could be either project manager or quality manager

Q42 How much time is spent in collecting data about the risk metrics?

On an average, only one half day a month was spent in collecting metrics.

Q43 At what frequency are the risk metrics updated?

Only in 30% of the companies, which used risk metrics, risks metrics were updated monthly.

Q44 Are you able to predict all types of risks using risk metrics used currently in the company ?

50% of the cases studied could not predict all the risks.

Q45 Specify the areas where risk metrics are not used currently

Uncertain areas

5.3 Conclusion

All the research findings through study of responses to the Questionnaire were discussed in this chapter. In the next chapter we will focus on the results construed during the study and proposing new techniques for better risk management.

CHAPTER 6 PROPOSED TECHNIQUES

The goal of this research is to propose some risk management metrics for predicting software project risks by studying previous research in software project management and different practices taken up by various software companies. This chapter outlines all the inferences drawn from the study and suggests some risk metrics along with risk management techniques that can be used for better risk management. The ME curriculum had been a great source of help and input for the following outcome.

6.1 Background

Risk management can not only help to ensure that the final product is safe, but can also be used to determine whether a development project will be finished on time and in budget. Cost overruns and scheduling delays are two of the largest risks for any software project and can be devastating for a manufacturer. Despite the seeming consensus on the merits of risk management, however, many manufacturers do not include it in their software development projects.

Most software development projects exceed their schedules or budgets. Many studies have identified causes for software project failure. These causes include:

- Poor project planning
- Inadequate documentation of project requirements
- A weak business case
- Lack of support and involvement from senior management
- No written quality plan, or no effective implementation of the plan
- Development team has poor understanding of the business problem to be solved
- Reactive rather than proactive strategies to handle with risks
- Developing the wrong functions and properties
- Straining computer-science capabilities
- Missing or misinterpreted requirement

6.2 How to Plan about Risk

Planning about risk is the first step towards risk management. Until we prepare a plan, we cannot succeed in mitigating risks. Planning for Risk involves designing a RMMM

Plan i.e. Risk Management , Monitoring and Mitigation Plan. Here the plan is divided into two parts to carry out the necessary detailing for better risk management : Risk Management Plan and a Risk Response Plan.

6.2.1 Creating a Risk Management Plan

Following steps must be taken to create a Risk Management Plan [8]:

i. Decide on and document your processes and strategy for risk management to match the

complexity of the project.

ii. Create a risk management team.

iii. Get input from the team and stakeholders and other risk management experts about best

practices.

iv. Assign risk management roles and responsibilities.

v. Determine the budget for the risk process based on all the work that must be done and

include it in the schedule and costs of the project.

vi. Decide how often you will have risk management meetings and work on risk for the

project.

vii. Determine the risk rating approach (scoring and interpretation).

viii Describe in the plan the thresholds for determining risk.

ix Create reporting formats for risk and tracking methods.

x Publish the plan to your team and stakeholders, get approval and make sure everyone

understands it through review and training.

xi Review the plan throughout the project to see whether updates must be made to the plan.

6.2.2 Creating a Risk Response plan

The most important output of the risk management process is the risk response plan. Following steps must be taken to prepare a documented risk response plan:

i. Decide on how you will document your risk response plan

ii. Make sure that the risks and their analysis are documented and gather the team to review the list

iii. Discuss with the team the various risk responses you could sue for the risk. Decide on the best response or responses for each risk.

iv. Document each risk response in detail

v. Add appropriate risk responses into the schedule or budget. Change the project plan or other areas of the project as necessary based on the risk response plan

vi. Add any secondary risk and risk responses based on any risk responses the team decides to implement

vii. Review the risk responses and contingency plans and ensure the owner and any other implementer understands it and knows what to do

viii Review the risk response plan regularly to see if the risk has occurred and if the plans are still applicable

ix Identify new risks based on continued risk identification meetings held with the project team

Once the documented risk management plans are ready, they must now be followed in totality and the actions taken be recorded. This will help in tracking the progress against the plans. Wherever the deviations are found, they must be indicated as red alarms to the concerned management and people so that proper response can be taken.

6.3 Proposed Software Risk Metrics

The time at which a risk is identified is an important factor in deciding the cost associated with that risk. The earlier the risk is detected to become a reality in the development life cycle of the software, the lesser is the cost for handling it. Also, risk occurs in all the phases of the software development. So, it is suggested that risk metrics should be included in all the life cycle phases. Some risk metrics are given below .

During Project Planning

- Risk Reporting timings and format for reports specified?(y/n)
- Time for risk planning included in estimates?(y/n)
- Previous estimates used for estimating time and effort? (y/n)
- Time for testing included in estimates ? (y/n)
- Time spent in planning and tracking activities throughout the project's life

Requirements Analysis Phase

- No. of requirements with ambiguous phrases
- No. of features clearly specified
- User capability specified? (y/n)
- Time and budget clearly specified and mutually agreed upon? (y/n)
- Requirements freezed ?(y/n)
- Number of quality attributes specified
- Work Breakdown Structure clearly specified?(y/n)
- Skill set required specified? (y/n)
- Availability of personnel with required skill sets checked?(y/n)
- Training arranged if required?(y/n)
- Responsibilities clearly specified?(y/n)
- Any prototype created for completely specifying the requirements of the system?(y/n)

Design Phase

- Program architectures defined ? (y/n)
- Any prototype created for checking whether the design is according to the requirements specified? (y/n)
- Number of independent modules identified
- Methodology used specified and mutually agreed upon?(y/n)
- Number of Quality attributes considered while selecting the design

Coding Phase

- Planned Vs Actual Size (at each milestone)
- Planned Vs Actual Time(at each milestone)
- Documentation done?(y/n)

Testing Phase

- No. of errors uncovered during testing Vs Number of errors reported for correction to development team
- Number of errors reported Vs number of errors corrected
- Number of software quality assurance plans and test plans designed.
- Number of formal test executions recorded

Implementation Phase

- Acceptance testing done? (y/n)
- No. of errors uncovered during implementation Vs Number of errors reported for correction to development team

Maintenance Phase

- Number of changes asked by customer after implementation
- Number of new requirements demanded by customer to be added

6.4 Proposed Tools and Techniques

A number of tools and techniques can be used for better risk management. Some of the are suggested below:

6.4.1 Software Quality Assurance

The survey shows higher the maturity level, better the risk management So, every software company should try to adhere to any of the standards to have better quality software development. Commonly used standards are: SEI's CMM or ISO 9000. The Carnegie Mellon SEI suggests recommendations on core software measures namely, software size, time, effort and defects which act as basic fuel elements for Software quality assurance and process improvement. SQA confirms that benchmarking, estimating and Risk assessment , progress control and reporting is used to ensure process improvement.

6.4.2 Prototyping

The study shows that major risks occur during requirements gathering phase of software development. If not properly understood and compiled, lots of effort and time is spent in redesigning and re-planning. Prototyping can improve understanding of the requirements and design, so selecting a prototyping process can reduce many project risks. It is useful to record your decisions in a risk monitoring, mitigation and management plan(RMMM Plan), so that both customer and development team can review how problems are to be avoided, as well as how they are to be handled should they arise.

6.4.3 Formal Methods

Formal methods provide a foundation for specification environments leading to analysis model that are more complete, consistent and unambiguous than those produced using conventional methods. Better the requirements specification document, lesser will be the risk involved in requirements understanding. Formal specification languages like Z and OCL can be used for preparing the requirement specification document. Training for using formal methods may be planned, if required.

6.4.4 Tools for project scheduling

Underestimates for cost and time result in ignoring the risks and results in late delivery of projects. So, it is better to use some tool for project scheduling and include time and cost for handling risks. Microsoft Project is such type of tool which can be used for creating plans for scheduling the software project and tracking the progress of the software development as well.

6.4.5 Risk Database

Risk data warehouse is a repository that provides for collection, maintenance and analysis of data gathered and used in the risk management processes. It shall consist of data from completed projects, with each project providing one data record. It is a centralized data repository that consolidates and organizes large volumes of financial data (regardless of geographic location, hardware platform, system or origin) allowing decision makers to evaluate multiple dimensions of risk, as well as their institution's overall risk. To allow for similarity checking, data captured shall contain general information about all projects such as languages used, platforms, databases used, tools, effort size etc. A sample risk database is shown in Appendix B.

6.4.6 Risk Item Checklist

Checklists used in risk identification may also be used in monitoring and controlling risk. Care must be taken to explore items that do not appear on a standard checklist if they seem relevant to the specific project.

6.4.7 Earned Value Analysis

Earned value is used for monitoring overall project performance against a baseline plan. Results from an earned value analysis may indicate potential deviation of the project at completion from cost and schedule targets. When a project deviates significantly from the baseline, updated risk identification, assessment and quantification should be performed.

6.4.8 Risk Management Tools

Using a risk management tool helps in managing and tracking the risks in an effective way. Following risk management tools can help in better risk management:

- **Risk+** A comprehensive Risk Analysis tool that integrates with Microsoft Project and provides features like Cost and schedule histograms; Data export; Sensitivity analysis; Embedded Graphics; Risk Critical path display; On-line help; Correlated risk; Monte-Carlo or Latin hypercube sampling

- **Ten Risk Manager** – A cost effective software application that helps in identifying the risks, evaluating them using risk registers, identifying risk controls and ownership. It is easy to implement ,can be used at all levels within the organization and has features like tools to gather, monitor and communicate risk information; automatically highlight high risks and associated mitigation actions.

- **Riskman –** A risk evaluation expert system that identifies project-related risks

- **RiskTrak** – A tool that supports the identification, analysis, reporting and management of risks throughout a software project

- **RiskRadar** – it assists project managers in identifying and managing project risks

6.4.9 Formal Training for handling Risk

No one is likely to possess all the skills needed, so it's crucial to train people to develop the necessary skills. A training program should be implemented to help people transition

from engineers to being project leaders and to handle risks, if at all they occur in any project. Before the commencement of any project, orientation of the possible risks(for e.g. system goes down, virus problem, undetectable bugs, sudden change in requirement, changes in team members etc.) which can occur during the project course should be given, it can be in the form of a document (leaflet, booklet etc.) or formal training course. An organization wide standard should be developed for risk communication and feedback of team members shall be taken at the end of each project. This also serves as an indirect lessons learned program or training for the concerned members to handle future risks.

6.4.10 Risk Manager

"The ability to manage projects and their constituent processes effectively is critical to the success of any business".

A risk manager is recommended if a program is large enough to afford one. The role for this position will be to capture and formalize risk management activities and results. This role includes being spokesperson for the program for risks for major reviews and reports. The risk manager can describe the common elements of the risk management program before specifics are discussed on a subsystem-by-subsystem basis, and later presentations can focus on details of specific elements of the system. If a company does not want to opt for a risk manager, the project manager can be the in charge. For this it's crucial to train him to develop necessary skills. Training may include general aspects of project management, technical and soft-skills programs and specifically risk handling.

6.4.11 Risk Reporting

The reporting criteria should be decided at the time of project scheduling itself. The risk item checklist should be designed and time and frequency of reporting it to the concerned people should be decided and properly documented. The Risk manager can be a person responsible for collecting the status reports from all the team members .The status reports can be of great help in checking for risks becoming a reality. The status reports based on the amount of functionality delivered, versus functionality planned to be delivered, at each stage of completeness provides quantitative monitoring of project status. The increased visibility of the project status gives early warning of project slippage [19].

6.4.12 Motivate team members for risk mitigation

Team members who help in risk management and risk mitigation should be motivated by giving incentives such as raise in salary, bonus, appreciation letters or anything according to company strategy. This sets an example and causes ripple effect in the peers, motivating all for using proactive strategies and better risk handling.

6.5 Conclusion

This chapter proposed risk management techniques and risk metrics after thorough study of data gathered from 20 software companies. These techniques and risk metrics can be applied to all projects of varying sizes. It discussed how to plan about risks of a software project. Some risk metrics that can be used in various project life cycle phases have been proposed. Techniques for better risk management such as following standards for maturity, prototyping, using formal methods, earned value analysis and tools for project scheduling are proposed. Along with these, risk manager, risk reporting method, formal training for handling risks, motivating people for combating risks, creating risk item checklist and creating and maintaining a risk database can provide measures for better risk handling. Some Risk management tools are also suggested which, if used, produce better results.

The next chapter finally concludes the study.

CHAPTER 7
CONCLUSIONS

Software Project Management includes Risk Management techniques and tools since years and a considerable improvement has been attained in this area but still the scene on Software Project development side is not very encouraging. The projects are still late, over budget and missing many desired aspects.

7.1 Background

Peter Kulik say,
"Tomorrow's problems are today's risks - Software risk management is becoming more and more widely implemented because of its proven results to identify and eliminate risks before they become problems"

In the work presented here, the first chapter focused on the brief introduction of Risk Management and its importance in Software development activities. It also defined the problem taken up in the thesis.

In the second chapter , the focus was on theoretical background of Software Project Management and Risk Management concepts. All software projects face the problem of quality, schedule and cost being affected by risks that are unexpected, unplanned or simply ignored. The stages of risk management (identify, analyze, plan, track, control and communicate) allow management, the project team, and the customer to increase their confidence that there will be both, no nasty surprises and that the risks will be mitigated effectively.

Software metrics were discussed in third chapter. How to define , classify and use the software metrics is focused primarily. The problems faced in metrics program are also discussed. Benefits of risk metrics emphasis the need for using the risk metrics.

Fourth chapter focused on the research approach taken up for the study. Different techniques can be used but most practical and effective technique for present work was survey technique. It was accomplished with the help of a questionnaire which was sent to different companies for understanding their Risk Management techniques and the risk metrics used currently by them.

In chapter 5, we construed from the survey findings that risk data is available in some of the companies but that data pertains to general risks, data pertaining to product specific risk is still missing or is not taken care of . Companies use checklists for some of the projects but mostly very less serious consideration is paid to risk history because risk handling is done on general basis. Many of the Software Engineers which

were interviewed were not aware of any risk history or probable fields. No risk manager is appointed in most of the projects, the absence of which lack the responsibility and seriousness in collecting and reporting about risks. CMM level 5 companies are more efficient in handling with risk than other less mature ones.

The last chapter proposed risk management techniques and risk metrics after thorough study and surveying the data from 20 software companies of different maturity levels. These techniques and risk metrics can be applied to all projects of varying sizes. Project risks can also be assessed by life cycle phases. Along with that it also discussed how to plan about Risks of a project by deciding deliverables at each milestone.

7.2 Proposed Risk Management Tools and Techniques

This thesis construes that by following some general techniques like creating Risk Management and Risk Response Plan, risk database, prototyping, creating checklists, earned value analysis, reviewing risks and risk metrics periodically, appointing risk Manager in large and complex projects, proper requirement change and configuration management, giving formal training and incentives for proper risk handling and an organization wide communication about risk, software risks can be handled. We can plan the risk deliverables at each milestone. Risk metrics should be taken care of for managing risks effectively.

By following all or some of these techniques and risk metrics we can ensure minimum risk occurrence and if, at all a risk occurs, we are ready with mitigation plans. The emphasis is on proactive rather than reactive approach for handling risk After every project, the team members should be asked to fill a risk feedback form for the lessons learned from the project. The data thus collected can be added to risk database. It will help in handling future risks and creating a proactive vision.

7.3 Future Scope

There are enough techniques for better risk management but the need of the hour is there implementation in Project development by different companies. As a future scope, more risk metrics in the life cycle phases of a project can be developed. Better risk management tools can be designed which can be used by any type of software development. Also specific types of risks related with particular category of software (for

example web based applications or client server applications) can also be studied separately in detail .Another development can be the implementation of risk database on Internet as a common repository available to all projects globally. These techniques need comprehensive study and numerous reviews conducted by different software companies.

7.4 Conclusion

To construe, there are many curative risk management tools and techniques and work is in progress towards proactive approach for handling risks in software projects. But until the software companies put these techniques , tools and metrics into practice ,not much benefits can be achieved and the projects will keep on suffering. The companies at CMM level 5 have leaped forward in this direction by combating most of the risks. Still, if these include some more risk management techniques and risk metrics as discussed in the present work, risk disasters can be assuaged further though its impossible to eliminate the risks completely.

APPENDIX A

QUESTIONNAIRE

Name of the company: Address: Company Profile:	Name of the Person: Designation: Work profile:

Note: This data would be used for research purpose only (in survey for M.E. Software Engineering Thesis)

General

1. Does your Company follow CMM? (Yes/No) If yes, indicate its maturity level :
   i) CMM Level 1 ii) CMM Level 2 iii) CMM Level 3
   iv) CMM Level 4 v) CMM Level 5

2. Which of the following process model is used by your company?

   1. Waterfall 2.Spiral 3. RAD 4. Prototype
   5. Any other? Please specify_____

3. How much time and budget (in percent) is devoted to each of the following activities?

Activity	**Time**	**Budget**
Requirement Specification		
Analysis and design		
Implementation		
Testing		
Documentation		
Feedback		
Risk Analysis		
Risk Mitigation		

4. How is the project schedule being designed in the company?
   (Specify any tool used)

5. How often do projects take **longer** to complete than estimated
   (Number of projects that take longer to complete/total amount of projects completed in a specified amount of time)?
6. How often are projects finished **earlier** than estimated?

(Number of project finished earlier than estimated/ total amount of projects completed in a specified amount of time)

7. Are you able to convince your customers about slip of budget or schedule? (Yes / No)

8. Which type of risks frequently occur in software development activity of almost all projects of the Company ?(Please tick mark)

Business	
Financial	
Personnel	
Political	
Resource	
Schedule	
Skills	
Technical	
Requirements	
Design	
Deployment and Support	
Integration	
Maintenance	

9. What is the percentage of each risk?

Type of Risk	**Percentage**
Business	
Financial	
Political	
Skills/ Personnel	
Technical	
Software life cycle phases	

10. In general, are you able to mitigate the Risks for most of the projects?(Yes/No)

Project Development

11. Please specify how are the user requirements interpreted in your company?

12. At what stage of development life cycle are the requirements freezed up?
    i) Requirement Analysis
    ii) Design
    iii) Implementation
    iv) Coding
    v) Testing

13. What control measures exist to manage creeping user requirements?

14. At what level, the failure reports are investigated in your company?
    i) Department
    ii) Project
    iii) Individual
    iv) Not at all

15. List down the areas of uncertainty as applicable to your company/projects. Can you name one of the most probable areas of uncertainty or cause of failure? (Skill set unavailability, Changing requirements, etc.)

16. Is Risk taken into consideration in project schedule management? (Yes/No)

17. Does your company give any formal training for risk handling? (Yes/No)

18. Do you link risk with size of Company/ team/ project? (Yes/No)

19. What is the frequency of collecting reports in your company (for example the time report system, the status report system etc)?

    i) Once a month
    ii) 2-3 times per month
    iii) 4-5 times per month
    iv) More than 5 times per month

20. If your key personnel change or leave, what criteria is used to ensure complete hand-over of data?

Software Risks

21. Does your Company use any tools/ techniques/ processes/strategy for managing Risks? Please Specify.

22. Is the Strategy used for managing risks proactive or reactive ?

23. What technique is used by the Company for predicting risks?

24. Do unnecessary features also become a risk factor? (Yes/No)

25. Give any unnecessary features which also become a risk factor.

26. Is any risk item checklist created? If yes, what are its contents (Product size, customer characteristics…)?

27. How are team members communicated about project risks?

28. Do you have a Risk Manager? If yes, what role does he play in your Company?

29. How does your company define Risk Assessment and Risk Management?

30. Is any RMMM plan or Risk Information Sheet used? (yes/no)

31. What steps are being followed in Risk Management (Planning, Investigation…)?

32. What criteria is used for assessing and analyzing Risks? Do you use any models/processes for assessing risks?

33. Does high complexity of project mean more risks?(Yes/No)

34. Does your Company catalogue data on risks? Does historical data really help in dealing with risks in upcoming Projects?

35. How are the Risks prioritized? Specify the criteria for identifying the importance/priority of a risk

36. List down the fields of database/risk history table, if used.

37. How is the team motivated to combat risks ?
(i)Incentives
(ii) appreciation letters
(iii) promotion
(iv) any other, please specify.

38. Do you always apply risk mitigation strategies once a risk is known to become a reality ? (Yes/ No)

Risk Metrics

39. Are any Risk metrics used for measuring risks? (Yes/ No)

40. What are the various risk metrics used in your company? What is the frequency of collecting these?

41. Is there some specific person responsible for collecting risk metrics? (Yes/ No)

42. How much time is spent in collecting data about the risk metrics?

43. At what frequency are the risk metrics updated?

44. Are you able to predict all types of risks using risk metrics used currently in the company ?

45. Specify the areas where risk metrics are not currently used in your

company?

APPENDIX B EXAMPLE RISK DATABASE

A sample entry in risk database is shown as follows:-

General Table

Field Name	Values
Project Code	P1005
Project Name	Jaysons Billing system
Business Domain	Brokerage/Finance
Project Leader	Mr. John Smith
Peak Team size	12
Life Cycle Phases	Full
Platform used	SQL server and Windows 2000 workstations
Language used	Java and Visual Basic 6.0
Start Date	01 Jan 02
End Date	25 Dec 02
Tools Used	VAJ for source code, Microsoft Word for documentation
Problem definition	Project includes automation of company accounts

Risk Table

Risk Number	Risk Category	Risk Mitigation steps
R101	Too many requirement changes	1.Obtain sign-off for the initial requirements specification 2.Convince the client about impact 3. Define a procedure to handle change 4. Negotiate payment on actual effort
R102	Unavailability of skill sets	1.Targeted and focused training 2.Ascertaining the skill set availability before signing up the project
…	….	…..

Project Risk Table

Project Code	Risk Number
P1005	R101, R105, R450
P1008	R102, R678, R234, R450

A temporary table containing Risk impact can be created for each project, the values will vary with each project. An example is as follows:-

Possible risks for the Project Code : P1090

Project Name : Jaysons Billing System

Risk Category	Probability	Impact	Risk Exposure	Mitigation Plan
Personnel attrition: Team members may leave at short notice	0.4	3	1.2	Assign tasks so that more than one person is aware of the units and use cases in the project
Size estimate significantly low	0.6	2	1.2	Use effective estimation techniques and also use previous project's information
End users resist system	0.2	3	0.6	Sign up the criteria for acceptance testing

Above table would be helpful in prioritizing the risks and making the team members aware of the possible occurrences of risks for the current project. This table/ report should be generated before the start of the project by visualizing and taking in account all the previous risks from the risk repository and circulated to all the concerned members. Each team member should be given an orientation for handling all possible risks before hand for a proactive approach to risks. A mitigation strategy can be planned and executed simultaneously with the project management plan. Such tables should be generated before the start of each phase of life cycle. Risks should be monitored and reevaluated periodically, perhaps at milestones for the assurance that the risk mitigation steps are having an effect and to revisit risk perception.

APPENDIX C

REFERENCES

1. Williams, R.C., J.A. Walker, and A.J.Dorofee, "Putting Risk Management into practice" IEEE Software, May 1997, pp. 75-81
2. Thomsett R., "The Indiana Jones School of Risk Management" American Programmer Vol 5 no. 7, September 1992, pp. 10-18
3. "Taxonomy based Risk Identification" Software Engineering Institute, CMU/SEI-93-TR-6-1993
4. Leveson, N.G., Safeware: System Safety and Computers, Addison Wesley,1995
5. Keil, M.,et. Al., "A framework for identifying software project risks" CACM Vol 41, no.1, November 1998, pp.76-83.
6. Karolak D.W., Software Engineering Risk Management, IEEE computer society press, 1996
7. Higuera, R.P., "Team risk management" Cross talk, U.S. Dept. of defence, January 1995, pp. 2-4.
8. Hall E.M. Managing risk: Methods for software systems development, Addison Wesley, 1998
9. Glutch, D.P., " A construct for describing software developmet risks" CMU/SEI-94-TR-14, Software Egineering Institute, 1994.
10. Gilb, T. Principles of software Engineering Management, Addison Wesley, 1998
11. Drucker P.,Management, WH Heinemann, 1975
12. Charette RN, "building btidges over intelligent rivers" American programmer, vol. 5 no. 7, September, 1992, pp. 2-9.
13. Charette RN, software engineering risk analysis and management, McGraw Hill/ intertext, 1989
14. Boehm, BW, Software Risk management, IEEE computer society press, 1989
15. Software risk abatement, AAFCS/AFLC pamphlet 800-45, US Air force, September 30, 1988
16. Pressman R.S, 'Software Engineering: A Practitioner's Approach', Sixth edition, Tata McGraw Hill.
17. Ramesh G, 'Managing Global Software Projects', Tata McGraw Hill.
18. Humphrey W., 'Managing the Software Process', Addison-Wesley.

19. Fenton E, Pleeger, 'Software Metrics – A Rigorous & Practical Approach' Thomson Brooks/Cole
20. Ghezzi C, Jazayeri M, Madrioli D, "Fundamentals of Software Engineering", Prentice Hall India.
21. P.Jalote, "Software Project Management in Practice", first edition 2002.
22. SWEBOK, Software Engineering Book of Knowledge
23. PMP: Project Management Professional Workbook, Sybex publications
24. Aggarwal K.K. and Singh Y.,'Software Engineering- Programs, Documentation and Operating Procedures', New Age International Publishers
25. www.pmbok.org
26. www.sei.cmu.edu
27. www.pmi.org
28. www.infogoal.com/pmc/pmchome.htm
29. www.computerworld.com/managementtopics/management
30. www.ManagementScience.org
31. Research paper on Software Metrics at: ftp://ftp.sei.edu/pub/education/cm12.pdf
32. Wiegers Karl E., "Lessons from software work effort" at www.processimpact.com
33. www.rspa.com/spi/metrics-primer.html
34. Morris Pam, "Metrics based Project Governance" in IWSM/Metrikon 2004 at www.totalmetrics.com/cms/servletpark
35. The SEI Core measures Anita D. Carleton, Robert E. and Wolfart B. Goethert – The Journal of the Quality Assurance Institue July 2004
36. www.projectmanagement.com
37. www.risksig.com
38. www.computer.org
39. Tghezzi C, McDermid J.A, ESEE'89: Proceedings of 2nd European SE Conference
40. www.CS-solutions.com
41. www.qsm.com
42. www.risktrac.com
43. www.spmn.com