Module VI

7

SOFTWARE AND SYSTEM QUALITY MANAGEMENT

Unit Structure

7.1 INTRODUCTION

Definition by ISTQB,

Quality: The degree to which a component, system or process meets specified requirements and/or user/customer needs and expectations.

software quality: The totality of functionality and features of a software product that bear on its ability to satisfy stated or implied needs.

Software quality management (SQM) is a management process that aims to develop and manage the quality of software in such a way so as the best ensure the product meets the quality standards expected by the customer while also meeting any necessary regulatory and developer requirements, if any.

7.2 OVERVIEW OF ISO 9001

- ISO 9000-3, the Guidelines offered by the International Organization for Standardization (ISO), represent implementation of the general methodology of quality management ISO 9000 Standards to the special case of software development and maintenance.

- Both ISO 9001 and ISO 9000-3 are reviewed and updated once every 5–8 years, with each treated separately.
- As ISO 9000-3 adaptations are based on those introduced to ISO 9001, publication of the revised Guidelines follows publication of the revised Standard by a few years.
- At the time of writing, the 2000 edition of ISO 9001 (ISO, 2000a) has been issued, but only the final just-completed draft of ISO 9000-3 (ISO/IEC, 2001) is awaiting approval.
- From the 1997 edition on, the ISO 9000-3 will represent the stand-alone ISO standard for the software industry.
- The 2000 edition of ISO 9001 as well as the new edition of ISO 9000-3 are supported by two additional conceptual standards:

1. ISO 9000 (ISO, 2000b), which deals with fundamental concepts and terminology, and
2. ISO 9004 (ISO, 2000c), which provides guidelines for performance improvement.

ISO 9001 — application to software: the TickIT initiative :

TickIT was launched in the late 1980s by the UK software industry in cooperation with the UK Department for Trade and Industry to promote development of a methodology for adapting ISO 9001 to the characteristics of the software industry known as the TickIT initiative. TickIT is currently authorized to accredit other organizations as certification bodies for the software industry in the UK.

TickIT activities include:

■ Publication of the TickIT Guide, that supports the software industry's efforts to spread ISO 9001 certification. The current guide (edition 5.0, TickIT, 2001), which includes references to ISO/IEC 12207 and ISO/IEC 15504, is distributed to all TickIT customers.

■ Performance of audit-based assessments of software quality systems and consultation to organizations on improvement of software development and maintenance processes in addition to their management.

■ Conduct of ISO 9000 certification audits.

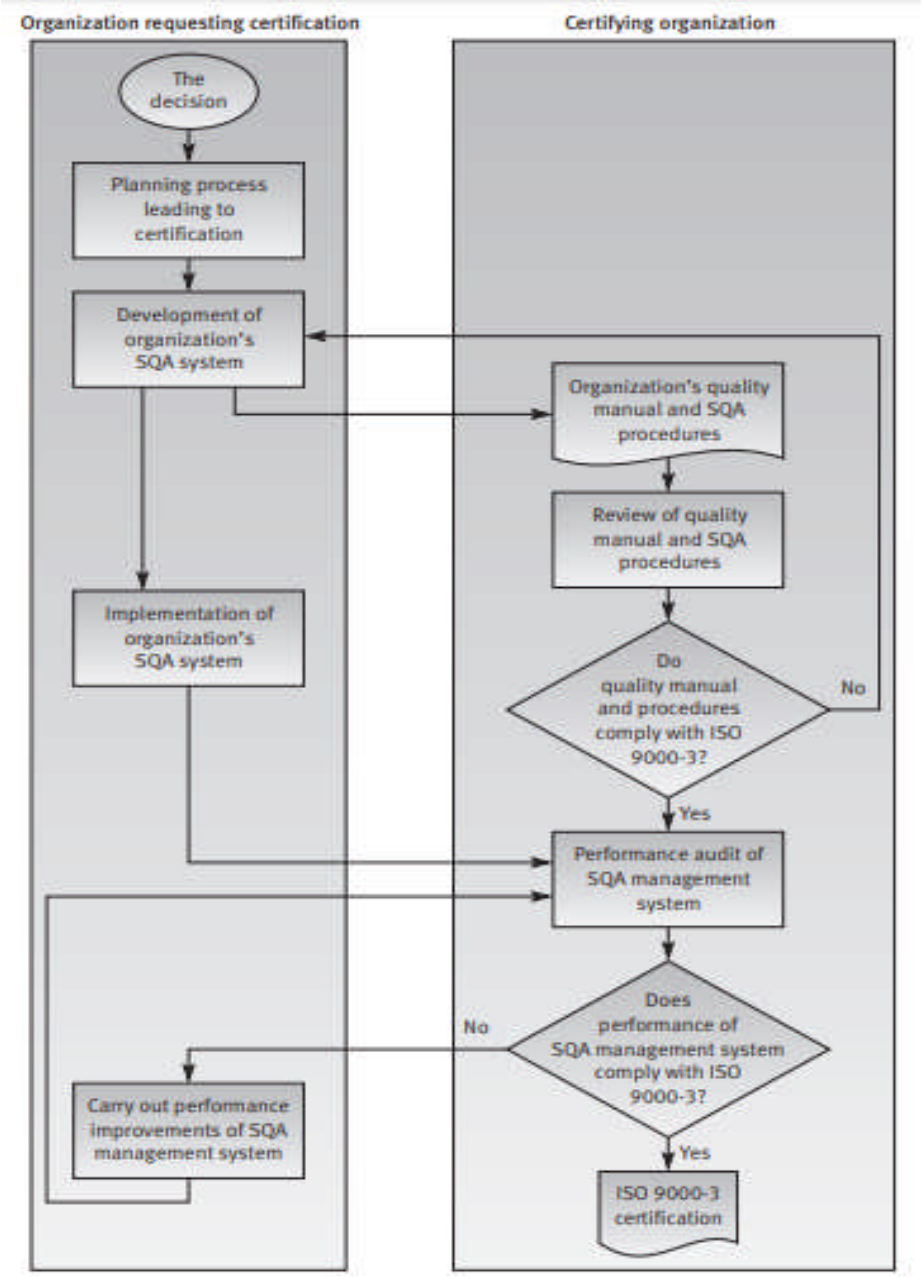


Figure 23.1: The ISO 9000-3 certification process

7.3 SEI CAPABILITY MATURITY MODEL

Carnegie Mellon University's Software Engineering Institute (SEI) took the initial steps toward development of what is termed a capability maturity model (CMM) in 1986.

The initial version of the CMM was released in 1992, mainly for receipt of feedback from the software community.

The first version for public use was released in 1993 (Paulk et al., 1993, 1995; Felschow, 1999).

The principles of CMM

Following are the concepts and principles of CMM assessment:

- Application of more elaborate management methods based on quantitative approaches increases the organization's capability to control the quality and improve the productivity of the software development process.
- The vehicle for enhancement of software development is composed of the five-level capability maturity model. The model enables an organization to evaluate its achievements and determine the efforts needed to reach the next capability level by locating the process areas requiring improvement.
- Process areas are generic; they define the "what", not the "how". This approach enables the model to be applied to a wide range of implementation organizations because:

➔ It allows use of any life cycle model

➔ It allows use of any design methodology, software development tool and programming language

➔ It does not specify any particular documentation standard.

The CMM and its key process areas (KPAs) are presented in following Figure

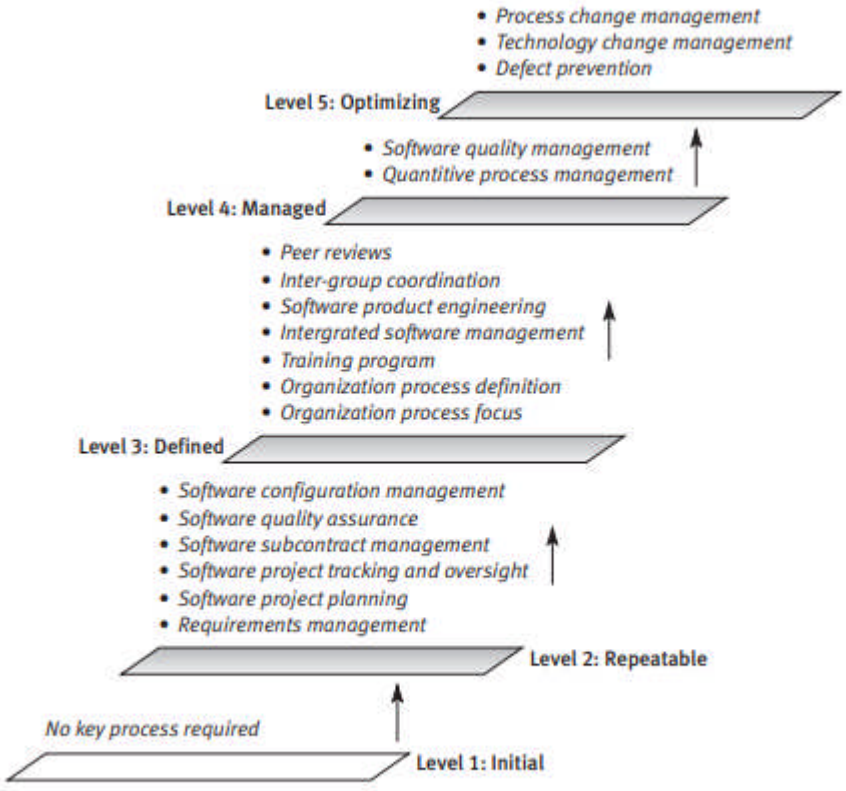


Figure 23.2: The CMM model levels and key process areas (KPAs)

Source: After Paulk *et al.* (1995)

Variety of specialized capability maturity models are as following:

- **System Engineering CMM (SE-CMM)**

➔ It focuses on system engineering practices related to product-oriented customer requirements.

➔ It deals with product development: analysis of requirements, design of product systems, management and coordination of the product systems and their integration.

➔ In addition, it deals with the production of the developed product: planning production lines and their operation.

- **Trusted CMM (T-CMM)**

It was developed to serve sensitive and classified software systems that require enhanced software quality assurance.

- **System Security Engineering CMM (SSE-CMM)**

It focuses on security aspects of software engineering and deals with secured product development processes, including security of development team members.

- **People CMM (P-CMM)**

It deals with human resource development in software organizations: improvement of professional capacities, motivation, organizational structure, etc.

- **Software Acquisition CMM (SA-CMM)**

It focuses on special aspects of software acquisition by treating issues – contract tracking, acquisition risk management, quantitative acquisition management, contract performance management, etc. – that touch on software purchased from external organizations.

- **Integrated Product Development CMM (IPD-CMM)**

It serves as a framework for integration of development efforts related to every aspect of the product throughout the product life cycle as invested by each department.

7.4 MCCALLS QUALITY MODEL

The classic model of software quality factors, suggested by McCall, consists of 11 factors (McCall et al., 1977). The McCall factor model provides a practical, up-to-date method for classifying software requirements (Pressman, 2000).

McCall's Factor Model

The 11 factors are grouped into three categories – product operation, product revision, and product transition factors.

- **Product operation factors −**

Product operation category includes five software quality factors, which deal with the requirements that directly affect the daily operation of the software. They are as follows − **Correctness, Reliability, Efficiency, Integrity, Usability.**

- **Product revision factors −**

According to McCall's model, three software quality factors are included in the product revision category. These factors are as follows − **Maintainability, Flexibility, Testability.**

- **Product transition factors −**

According to McCall's model, three software quality factors are included in the product transition category that deals with the adaptation of software to other environments and its interaction with other software systems. These factors are as follows − **Portability, Reusability, Interoperability.**

1 **Correctness:**

These requirements deal with the correctness of the output of the software system. They include −

- Output mission
- The required accuracy of output that can be negatively affected by inaccurate data or inaccurate calculations.
- The completeness of the output information, which can be affected by incomplete data.
- The up-to-dateness of the information defined as the time between the event and the response by the software system.
- The availability of the information.
- The standards for coding and documenting the software system.

2 **Reliability:**

Reliability requirements deal with service failure. They determine the maximum allowed failure rate of the software system, and can refer to the entire system or to one or more of its separate functions.

3 **Efficiency:**

It deals with the hardware resources needed to perform the different functions of the software system. It includes processing capabilities (given in MHz), its storage capacity (given in MB or GB) and the data communication capability (given in MBPS or GBPS).

It also deals with the time between recharging of the system's portable units, such as, information system units located in portable computers, or meteorological units placed outdoors.

4 **Integrity:**
This factor deals with the software system security, that is, to prevent access to unauthorized persons, also to distinguish between the group of people to be given read as well as write permit.

5 **Usability:**
Usability requirements deal with the staff resources needed to train a new employee and to operate the software system.

6 **Maintainability:**
This factor considers the efforts that will be needed by users and maintenance personnel to identify the reasons for software failures, to correct the failures, and to verify the success of the corrections.

7 **Flexibility:**

This factor deals with the capabilities and efforts required to support adaptive maintenance activities of the software.

These include adapting the current software to additional circumstances and customers without changing the software.

This factor's requirements also support perfective maintenance activities, such as changes and additions to the software in order to improve its service and to adapt it to changes in the firm's technical or commercial environment.

8 **Testability:**

Testability requirements deal with the testing of the software system as well as with its operation.

It includes predefined intermediate results, log files, and also the automatic diagnostics performed by the software system prior to starting the system, to find out whether all components of the system are in working order and to obtain a report about the detected faults.

Another type of these requirements deals with automatic diagnostic checks applied by the maintenance technicians to detect the causes of software failures.

9 **Portability:**
Portability requirements tend to the adaptation of a software system to other environments consisting of different hardware, different operating systems, and so forth.
The software should be possible to continue using the same basic software in diverse situations.

10 **Reusability:**

This factor deals with the use of software modules originally designed for one project in a new software project currently being developed.

They may also enable future projects to make use of a given module or a group of modules of the currently developed software.

The reuse of software is expected to save development resources, shorten the development period, and provide higher quality modules.

11 **Interoperability:**

Interoperability requirements focus on creating interfaces with other software systems or with other equipment firmware.

For example, the firmware of the production machinery and testing equipment interfaces with the production control software.

7.5 SIX SIGMA

- The process of producing high and improved quality output is known as Six Sigma.
- This can be done in two phases – identification and elimination.
- The cause of defects is identified and appropriate elimination is done which reduces variation in whole processes.
- Six Sigma's aim is to eliminate waste and inefficiency and increase customer satisfaction by delivering what the customer is expecting.
- It follows a structured methodology, and has defined roles for the participants.
- It is a data driven methodology, and requires accurate data collection for the processes being analyzed.
- It is about putting results on Financial Statements.

Following are few characteristics of Six Sigma:

The Characteristics of Six Sigma are as follows:

➔ **Statistical Quality Control:**

Six Sigma denotes Standard Deviation in statistics. Standard Deviation is used for measuring the quality of output.

➔ **Methodical Approach:**

The Six Sigma is a systematic approach of application in DMAIC(Design-Measure- Analyze-Improve-Control) and DMADV (Design- Measure- Analyze-Design-Verify) which can be used to improve the quality of production.

➔ **Fact and Data-Based Approach:**

The statistical and methodical method shows the scientific basis of the technique.

➔ **Project and Objective-Based Focus:**

The Six Sigma process is implemented to focus on the requirements and conditions.

➔ **Customer Focus:**

The customer focus is fundamental to the Six Sigma approach. The quality improvement and control standards are based on specific customer requirements.

➔ **Teamwork Approach to Quality Management:**

The Six Sigma process requires organizations to get organized for improving quality.

7.6 FORMAL TECHNICAL REVIEWS

*"A formal review produces a **report** that identifies the material, the reviewers, and the review team's judgment as to whether the product is acceptable. The **principal deliverable is a summary of the defects** found and the issues raised. The members of a formal review **team share responsibility for the quality of the review**, although authors are ultimately responsible for the quality of the products they create (Freedman and Weinberg 1990)."*

- The purpose of an Formal Technical Review (FTR) is to identify errors in function,logic and implementation of software.
- It is used to verify that the software under review meets its requirements.
- It ensures that the software has been represented according to predefined standards.
- It also helps to achieve software that is developed in a uniform manner and to make projects more manageable.

The FTR is actually a class of reviews that includes walkthroughs and inspections.

- A **review** is '*a process or meeting during which a software product is presented to project personnel, managers, users, customers, user representatives, or other interested parties for comment or approval'*

- An **inspection** is '*a visual examination of a software product to detect and identify software anomalies, including errors and deviations from standards and specifications*'

- A **walkthrough** is '*a static analysis technique in which a designer or programmer leads members of the development team and other*

interested parties through a software product, and the participants ask questions and make comments about possible errors, violation of development standards, and other problems'

Characteristics of FTR are:

- Well-defined process: Phases (orientation, etc.) Procedures (checklists, etc.)
- Well-defined roles: Moderator, Reviewer, Scribe, Author, etc.
- Well-defined objectives: Defect removal, requirements elicitation, etc.
- Well-defined measurements: Forms, consistent data collection, etc.

7.7 TOOLS AND TECHNIQUES FOR QUALITY CONTROL

- The control quality process is defined as the "process of monitoring and recording the results of executing the quality activities to assess performance and recommend necessary changes."
- In other words, quality control focuses on project results ensuring that they comply with the quality standards defined for the project and eliminating any causes of unsatisfactory performance.
- This process measures the details of the product results, such as deliverables or defects, and also of the project management results, such as schedule.
- Many of the techniques under the control quality process assume a working knowledge of statistical quality control, in particular the concepts of sampling and probability.
- The distinctions between attribute and variable sampling, precision and accuracy, and tolerance and control limits are fundamental components of a working knowledge of statistical quality control:

Prevention aids in identifying and avoiding potential problems so that they never enter or impact the process.

Inspection helps to identify and eliminate or correct errors so that they are not delivered to the customer.

Tolerance is a range of acceptable performance or results.

There are seven basic quality tools identified as appropriate for use in both the quality management plan and control quality processes. They are known as Ishikawa's seven basic tools of quality:

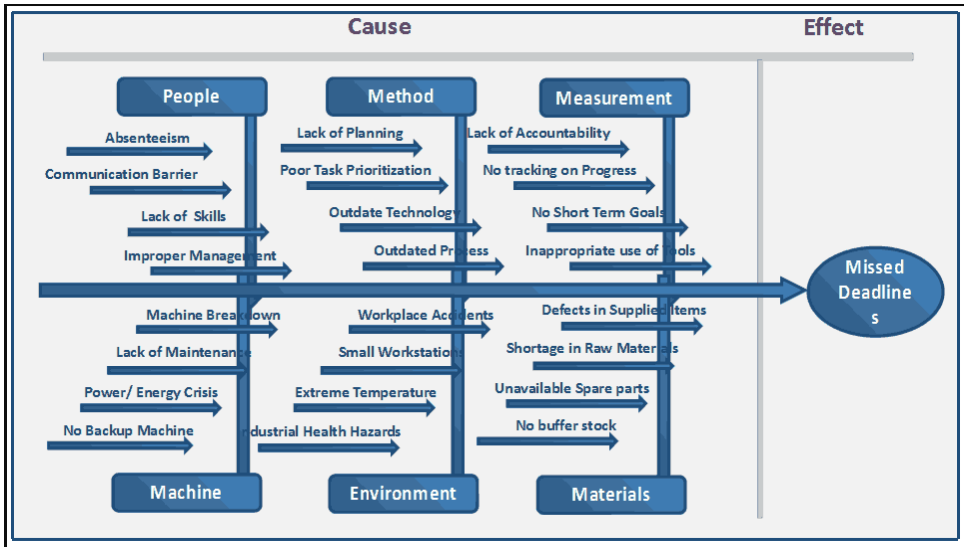
1. cause-and-effect diagrams,
2. flowcharting,
3. check sheets,
4. Pareto diagrams,
5. control charts,

6. histograms and
7. scatter diagrams.

Ishikawa's seven basic tools are also referred to as the 7QC tools.

Cause and Effect Diagrams

- ➔ Cause-and-effect diagrams, or Ishikawa diagrams, were developed by Kaoru Ishikawa to illustrate and help determine how various factors relate to potential problems.
- ➔ Also called fishbone diagrams because they resemble the skeleton of a fish.
- ➔ The head of the fish is the effect and each bone of the fish is a cause that leads to that effect.
- ➔ The bones can branch off into smaller bones as you determine the lowerlevel cause-effect relationships.
- ➔ When all the bones are filled in, the diagram lets you look at all the possible causes (individual or combinations) of the effect (or problem) so that you can develop a solution to mitigate that effect.
- ➔ The diagram allows organized thought and encourages orderly consideration of the factors that result in a certain outcome.



Flowcharts

- ➔ Flowcharts show the logical steps in a process and how various elements within a system are related.
- ➔ They can be used to determine and analyze potential problems in quality planning and quality control.
- ➔ Flowchart outlines the logical steps to complete a process.
- ➔ By documenting these logical steps, the team can identify where quality problems might occur and then develop approaches to proactively manage them.
- ➔ Flowcharting also helps create a process that is repeatable.

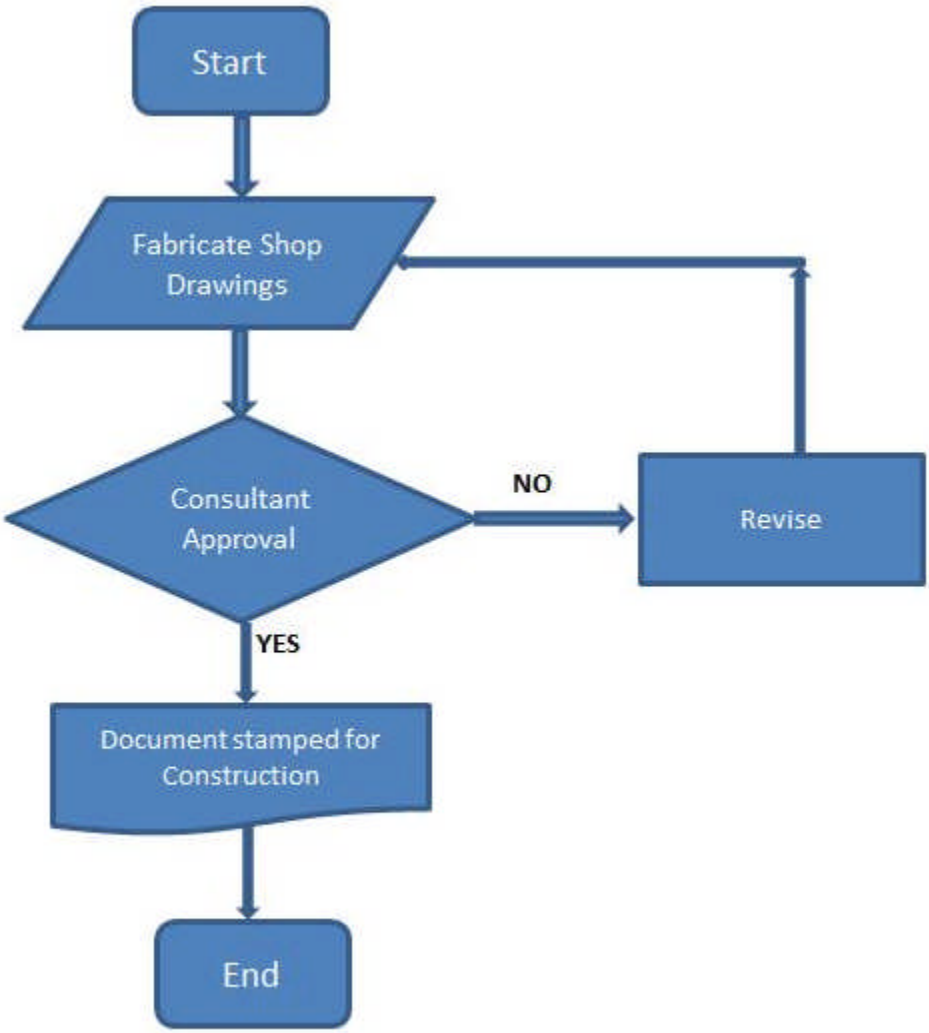


Figure A. Flowchat for drawing approval

Check Sheets

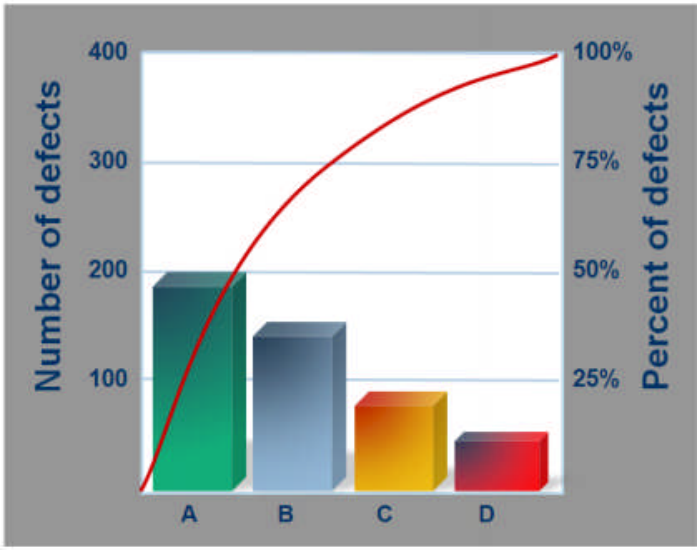
- ➔ Check sheets are used to organize information in order to facilitate data gathering.
- ➔ Check sheets are particularly effective for doing inspections, enabling focus on the particular attributes that may be contributing to potential or identified quality problems.

Check Sheet

Organization :			
Product Name :			
Date :		Time :	
Inspected no. of Items : 1200		Executive Name :	
Remarks : All Items inspected			
S.No:	Defect	Tally Marks	Count
1.	Lap Mark	𝍸 III	8
2.	Neck Crack	𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 III	38
3.	Finish Check	𝍸 𝍸 𝍸 𝍸 IIII	24
4.	Shoulder Check	𝍸 𝍸 𝍸 II	17
5.	Split Ring	𝍸 𝍸 𝍸 IIII	19
6.	Body Check	𝍸 𝍸 IIII	14
		Total No. of defect count	120
Defectives		𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 𝍸 III	88

Pareto Diagrams

- ➔ A Pareto chart or diagram, is a specific type of histogram that is based on Pareto's principle, which states that a large number of defects or problems are caused by a small number of causes.
- ➔ Pareto's principle, frequently referred to as the 80/20 rule or 80/20 principle.
- ➔ Which means that eighty percent of the cost of defects are caused by twenty percent of the problems.
- ➔ A Pareto diagram is an ordered bar graph showing the number of defects and their causes ranked by frequency.
- ➔ The bars on the diagram graphically show the number and percentage of causes individually and the line shows the cumulative value.
- ➔ Pareto charts help focus attention on the most critical issues to get the most benefit.

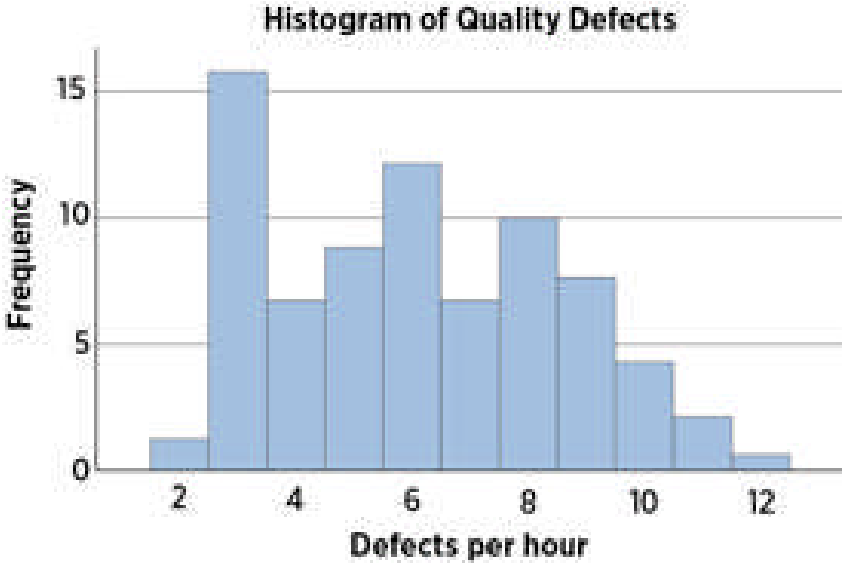


Histograms

➔ A histogram is a vertical bar graph that represents the frequency of each measured category (known as bins) of variable.

➔ The histogram is particularly useful for identifying common causes.

➔ The histogram can be ordered, similar to a Pareto chart, or unordered.



Control Charts

➔ Control charts are used to determine if processes are in or out of statistical control. Most processes experience a degree of normal variation i.e. most processes do not achieve target performance all the time.

➔ Control charts provide a mechanism for establishing a statistically objective range of acceptable variation around the target performance, thereby enabling attention to be focused on special cause variations (those that fall outside of the established performance limits).

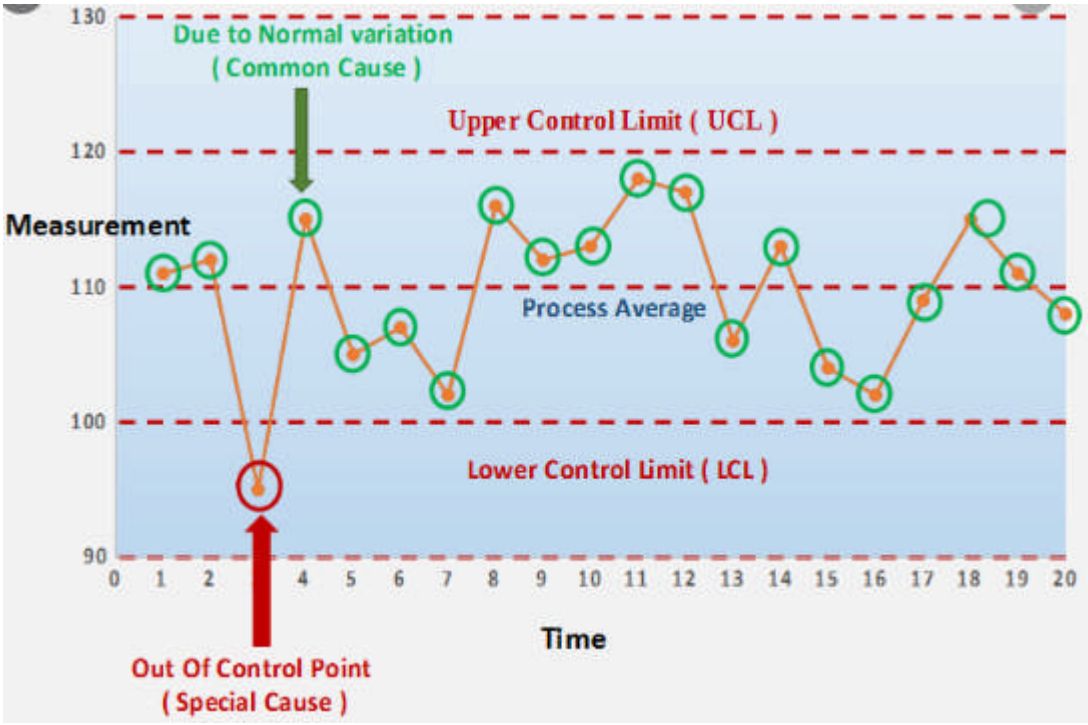
➔ Control chart limits are established on the basis of standard deviations from the mean (target) performance. The upper control limit (UCL) and lower control limit (LCL) are established so that 99.73 percent (three standard deviations above and below the mean) of the measured data points fall within the range.

➔ The following ones are more common:

❑ Rule of Seven, or Seven Run Rule: Seven data points in a row are above or below the mean.

❑ Trend of Seven Rule: Seven data points in a row follow a trend up or down.

❑ Rule of One: Any single data point is outside of the control limits (upper or lower).



Scatter Diagrams

➔ Scatter diagrams plot two variables, the independent variable and the dependent variable, to graphically show the relationship between them.

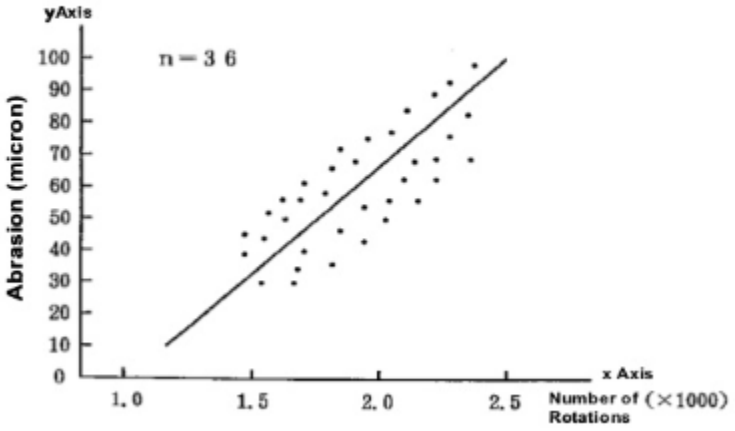
➔ The X-axis in the diagram represents one characteristic (usually the independent variable), and the Y-axis measures the other.

➔ To interpret the diagram, look at two characteristics of the clustering:

❑ Tightness: The closer the cluster is to a diagonal line drawn through the graph, the more the two variables are likely to be linearly correlated. High correlation between the characteristics means that a change in one characteristic will be accompanied by a change in the other.

❑ Direction: If the correlation is positive, then as one variable increases so does the other, and the line will have a positive slope (from lower left to upper right). On the other hand, if the correlation is negative, it implies that as one characteristic increases, the other decreases, and the line will have a negative slope (from lower right to upper left).

Scatter Diagram



7.8 ANSWER THE FOLLOWING :

1. Explain the benefits of the use of SQA standards.

- ❑ The ability to make use of the most sophisticated and comprehensive professional methodologies and procedures
- ❑ Better understanding and cooperation between users of the same standards:
  - – Between team members and between project teams
  - – Between software developers and external participants in the project
  - – Between suppliers and customers.

2. Describe the contributions made by the use of standards.

- ❑ Provision of superior professional methodologies for use in the development process and for its management
- ❑ Provision of SQA certification services based on independent professional quality audits
- ❑ Provision of tools for “self-assessment” of achievements in planning and operating an organization’s SQA system.

3. Describe the general principles underlying quality management according to ISO 9000-3.

- ❑ Customer focus – understanding a customer’s current and future needs
- ❑ Leadership exercised in the creation and maintenance of a positive internal environment in order to achieve the organization’s objectives

- ❑ Involvement of people at all levels to further organizational goals
- ❑ Process approach – activities and related resources perceived and managed as a process
- ❑ Systems approach to management – managing processes as a system
- ❑ Continual improvement of the organization's overall performance
- ❑ Factual approach to decision-making – decisions based on the analysis of data and information
- ❑ Mutually beneficial supplier relationships – emphasis on coordination and cooperation

4. Describe the principles embodied in the Capability Maturity Model (CMM).

- ❑ Application of more highly elaborated software quality management methods increases the organization's capability to control quality and improve software process productivity
- ❑ Application of the five levels of the CMM enables the organization to evaluate its achievements and determine what additional efforts are needed to reach the next capability level
- ❑ Process areas are generic, with the model defining "what" and leaving the "how" to the implementing organizations, i.e., the choice of life cycle model, design methodology, software development tool, programming language and documentation standard.

8

SOFTWARE RISK MANAGEMENT

Unit Structure

8.0 OBJECTIVES

- Define risk identification and causes , effects , and nature off project risks.
- Apply several analysis techniques that can be used to prioritize and analyze various project risks.
- Describe the various risk strategies
- Describe the risk monitoring and control
- Describe risk evaluation in terms of how the entire risk management process should be evaluated in order to identify best practices.

8.1 INTRODUCTION

Project risk management provides an early warning system for problems that need to be addressed or resolved .Although risk has a certain negative connotation , project stakeholders should be vigilant in identifying opportunities. Although many associate uncertainty with

threats, it is important to keep in mind that there is uncertainty when pursuing opportunities, as well.

Plan risk management determines how to approach and plan the project risk management activities. an output of this process is the development of arisk management plan.

Deciding which risks impact the project. Risk identification generally includes many of the project stakeholders and requires an understanding of the project's goal , as well as the project's scope ,schedule, budget , and quality objectives.

Developing procedures and techniques to reduce the threats of risks, while enhancing the likelihood of opportunities.

8.2 RISK ANALYSIS AND MANAGEMENT

Risk analysis and management are series of steps that help a software team to understand and mange uncertainty.

Many problems can plague a software project .A risk is a potential problem –it might happen it might not.

8.2.1 Steps of Risk Analysis and Management

1. Recognizing what can go wrong is the first step , called "risk identification."
2. Each risk is analyzed to determine the likelihood that it will occur and the damage that it will do if it does occur.
3. Risks are ranked, by probability and impact.
4. Finally, a plan is developed to manage those risks with high probability and high impact.

In short, the four steps are

- Risk Identification
- Risk Projection
- Risk Assessment
- Risk Management

Risk always involves two characteristics a set of risk information sheets is produced.

- **Uncertainty -** the risk may or may not happen; that is, there are no 100 % probable risks.
- **Loss –** if the risk becomes a reality, unwanted consequences or losses will occur.

8.3 TYPES OF IT PROJECT RISK

What types of risks are we likely to encounter as the software is built??

- **Project risks** threaten the project plan. That is, if project risks become real, it is likely that project schedule will slip and that costs will increase. Project risks identify potential budgetary, schedule, personnel (staffing and organization), resource, customer, and requirements problems and their impact on a software project.
- **Technical risks** threaten the quality and timeliness of the software to be produced. If a technical risk becomes a reality, implementation may become difficult or impossible.
- **Business risks** –threaten the viability of the software to be built. Business risks often jeopardize the project or the product. Candidate for top five business risks are

  - Market risk
  - Strategic risk
  - Management risk and
  - Budget risk
- **Known risks** are those that can be uncovered after careful evaluation of the project plan.
- **Predictable risks** are extrapolated from past project experience (e.g., staffturnover, poor communication with customer, dilution of staff effort as ongoing maintenance requests are serviced).
- **Unpredictable risks –** are the joker in the deck. They can do occur, but they are extremely difficult to identify in advance.

8.4 REACTIVE VS PROACTIVE RISK STRATEGIES

8.4.1 Reactive risk strategies

1. Reactive risk strategies follows that the risks have to be tackled at the time of their occurrence.
2. No precautions are to be taken as per this strategy.
3. They are meant for risks with relatively smaller impact.

8.4.2 Proactive risk strategies

1. Proactive risk strategies follows that the risks have to be identified before start of the project.
2. They have to be analyzed by assessing their probability of occurrence ,their impact after occurrence , steps to be followed for its precaution.
3. They are meant for risks with relatively higher impact.

8.5 RISK IDENTIFICATION

Risk identification is asystematicattemptto the project plan (estimates, schedule, resource loading, etc.) .By identifying known and predictable risks, the project manager takes a first step toward avoiding them when possible and controlling them when necessary.

One method for identifying risks is to create a risk item checklist. The checklistcan be used for risk identification and focuses on some subset of known and predictable risks a in the following generic categories:

- **Product size** – risks associated with overall size of the software to be built or modified.
- **Business impact –** risks associated with constraints imposed by management or the marketplace.
- **Customer characteristics**–risks associated with sophistication of the customer and developer's ability to communicate with the customer in a timely manner.
- **Process definition -** risks associated with the degree to which the software process has been defined and is followed by the development organization.
- **Development environment** – risks associated with the availability and quality of the tools to be used to build the product.
- **Technology to be built** – risks associated with the complexity of the system to be built and the "newness" of the technology that is packaged by the system.
- **Staff size and experience** - risks associated with the overall technical and project experience of the software engineers who will do the work.

8.5.1 Assessing Overall Project Risk

Is the software project we are working on at serious risk?

The questions are ordered by their relative importance to success of a project.

1. Have top software and customer managers formally committed to support the project?
2. Are end –users enthusiastically committed to the project and the system/product to be built?
3. Are requirements fully understood by the software engineering team and their customers?
4. Have customers been involved fully in the definition of requirements?
5. Do end –users have realistic expectations?
6. Is the project scope stable?
7. Does the software engineering team have the right mix of skills?
8. Are project requirements stable?

9. Does the project team have experience with the technology to be implemented?
10. Is the number of people on the project team adequate to do the job?
11. Do all customer/ user constituencies agree on the importance of the project and on the requirements for the system/ product to be built?

If any one of these questions is answered negatively, mitigation, monitoring, and management steps should be instituted without fail.

2.5.2 Risk Components and Drivers

The risk components are defined in the following manner:

- **Performance risk**- the degree of uncertainty that the product will meet its requirements and fit for its intended use.
- **Cost risk** - the degree of uncertainty that the project budget will be maintained.
- **Support risk** – the degree of uncertainty that the resultant software will be easy to correct, adapt, and enhance.
- **Schedule risk**- the degree of uncertainty that the project schedule will be maintained and that the product will be delivered on time.

The impact of each risk driver on the risk component is divided into one of four impact categories– negligible, marginal, critical, or catastrophic.

Components / Category		Performance	Support	Cost	Schedule
Catastrophic	1	Failure to meet the requirement would result in mission failure		Failure results in increased costs and schedule delays with expected values in excess of $ 500K	
	2	Significant degradation to non-achievement of technical performance	Nonresponsive or unsupportable software	Significant financial shortage budget overrun likely	Unachievable IOC
Critical	1	Failure to meet the requirement would degrade system performance to a point where mission success is questionable.		Failure results in operational delays and /or increased costs with expected value of $ 100K to $500K	
	2	Some reduction in technical performance	Minor delays in software modifications	Some shortage of financial resources ,possible overruns	Possible slippage in IOC

Marginal	1	Failure to meet the requirement would result in degradation of secondary mission.		Costs, impacts, and /or recoverable schedule slips with expected value of $1K to $100K	
	2	Minimal to small reduction in technical performance	Responsive software support	Sufficient financial resources	Realistic achievable schedule
Negligible	1	Failure to meet the requirement would create inconvenience or non-operational impact		Error results in minor cost and/ or schedule impact with expected value of less than $ 1K	
	2	No reduction in technical performance	Easily supportable software	Possible budget under run	Early achievable IOC

1) The potential consequence of undetected software errors or faults.
2) The potential consequence if the desired outcome is not achieved.

8.6 RISK PROJECTION

Risk projection , also called risk estimation, attempts to rate each risk in two ways – the likelihood or probability that the risk is real and the consequences of the problems associated with the risk , should it occur.
The project planner, along with other managers and technical staff , performs four risk projection activities.

1. Establishing a scale that reflects the perceived likelihood of a risk.
2. Delineating the consequences of the risk.
3. Estimating the impact of the risk of the project and the product.
4. Noting the overall accuracy of the risk projection so that there will be no misunderstandings.

8.6.1 Developing a Risk Table

Risks	Category	Probability	Impact	RMMM
Size estimate may be significantly low	PS	60%	2	
Larger number of users than planned	PS	30%	3	
Less reuse the planned	PS	70%	2	
End –users resist system	BU	40%	3	
Delivery deadline will be tightened	BU	50%	2	

Funding will be lost	CU	40%	1	
Customer will change requirements	PS	80%	2	
Technology will not meet expectations	TE	30%	1	
Lack of training on tools	DE	80%	3	
Staff inexperienced	ST	30%	2	
Staff turnover will be high	ST	60%	2	

I. A risk table provides a project manager with a simple technique for risk projection.

II. A sample risk table is illustrated in Figure. The risk table is sorted by probability and impact to rank risks.

III. A project team begins by listing all risks in the first column of the table. This can be accomplished with the help of risk item checklists referenced. Each risk is categorized in the second column (e.g. PS implies a project size risk, BU implies business risk).

IV. The probability of occurrence of each risk is entered in the next column of the table. The probability value for each risk can be estimated by team members individually. Individual team members are polled in round –robin fashion until their assessment of risk probability begins to converge.

V. Next, the impact of each risk is assessed.

VI. The categories for each of four risk components – performance, support, cost, and schedule- are averaged to determine an overall impact value.

VII. Once the first four columns of the risk table have been completed , the table is sorted by probability and by impact. High –probability, high –impact risks percolate to the top of the table, and low –probability risks drops drop to bottom.

VIII. This accomplishes first- order risk prioritization. The project manager studies the resultant sorted table and defines a cutoff line. The cutoff line implies that only risks that lie above the line will be given further attention.

IX. Risks that fall below the line are re- evaluated to accomplish second- order prioritization.

X. A risk factor that has a high impact but a very low probability of occurrence should not absorb a significant amount of management time.

XI. All risks that lie above the cut off line must be managed.

XII. The column labeled RMMM contains a pointer into a Risk Mitigation, Monitoring and Management Plan or alternatively, a collection of risk information sheets developed for all risks that lie above the cutoff.

How do we assess the consequences of a risk?

The following steps are recommended to determine the overall consequences of a risk.

I. Determine the average probability of occurrence value for each risk component.

II. Using figure determine the impact for each component based on criteria shown.

III. Complete the risk table and analyze the results as described in the preceding sections.

The overall risk exposure RE, is determined using the following relationship
RE = P * C

Where P is probability of occurrence for a risk, and C is cost to the project should the risk occur.

8.7 RISK ASSESSMENT

At this point in the risk analysis process we have established a set of triples of the form:
$[r_i , l_i , x_i]$

Where r_i is risk, l_i is the likelihood (probability) of the risk, and x_i is the impact of the risk.

During risk assessment, we further examine the accuracy of the estimates that were made during risk projection, attempt to rank the risks that have been uncovered, and begin thinking about ways to control and / or avert risks that are likely to occur.

Therefore, during risk assessment we perform the following steps:

I. Define the risk referent levels for the project.

II. Attempt to develop a relationship between each (ri, li,xi) and each of the referent levels.

III. Predict the set of referent points that define a region of termination, bounded by a curve or areas of uncertainty.

IV. Try to predict how compound combinations of risks will affect a referent level.

8.8 RISK MITIGATION, MONITORING, AND MANAGEMENT

- An effective strategy must consider three issues:
- ✓ Risk avoidance
- ✓ Risk monitoring
- ✓ Risk management and contingency planning
- High staff turnover in any organization will have a critical impact on project cost and schedule.
- To mitigate the risk, project management must develop a strategy for reducing turnover.

Among the possible steps to be taken are

- ✓ Meet with current staff to determine causes for turnover * Mitigate those causes that are under our control before the project starts.
- ✓ Once the project commences, assume turnover will occur and develop techniques to ensure continuity when people leave.
- ✓ Organize project teams so that information about each development activity is widely dispersed.
- ✓ Define documentation standards and establish mechanisms to be sure that documents are developed in a timely manner.
- ✓ Conduct peer reviews of all work.
- ✓ Assign a backup staff member for every critical technologist.

8.9 THE RMMM PLAN

1. A risk management strategy can be included in the software project plan or the risk management steps can be organized into a separate Risk Mitigation, Monitoring and Management Plan.
2. The RMMM plan documents all work performed as part of risk analysis and is used by the project manager as part of the overall project plan
3. Some software teams do not develop a formal RMMM document. Rather, each risk is documented individually using a risk information sheet(RIS).
4. Once RMMM has been documented and the project has begun, risk mitigation and monitoring steps commence.

Risk monitoring is a project tracking activity with three primary objectives:

1) To assess whether predicted risks do, in fact, occur
2) To ensure that risk aversion steps defined for the risk are being properly applied
3) To collect information that can be used for future risk analysis. In many cases, the problems that occur during a project.

Risk information sheet			
Risk ID : PO 2-4-32	Date: 5/9/02	Prob: 80%	Impact : high
Description : Only 70 percent of the software components scheduled for reuse will, in fact, be Integrated into the application. The remaining functionality will have to be custom developed.			
Refinement / context : Sub condition 1: certain reusable components were developed by a third party with no knowledge of internal design standards. Sub condition 2: The design standard for component interface has not been solidified and may not conform to certain existing reusable components. Sub condition 3: Certain reusable components have been implemented in a language that is not supported on the target environment.			
Mitigation / monitoring: 1. Contact third party to determine conformance with design standards. 2. Press for interface standards completion; consider component structure when deciding on interface protocol. 3. Check to determine number of components in sub condition 3 category; check to determine if language support can be acquired.			
Management / contingency plan / trigger : RE computed to be $ 20,200. Allocate this amount within project contingency cost. Develop revised schedule assuming that 18 additional components will have to be custom built; allocate staff accordingly. Trigger : Mitigation steps unproductive as of 7/ 1 /02			
Current status : 5/12/02: Mitigation steps initiated.			
Originator: D. Gagne		Assigned : B.Laster	

8.10 SUMMARY

Risk identification should include identifying both threats and opportunities. Since most risks are interrelated and can affect the project in different ways, the project stakeholders should take a broad view of project risks.

Risk assessment allows the project stakeholders to determine what risks require a response. The goal of project risk management is not to avoid each and every risk at all costs, but to make well- informed decisions as to which risks are worth taking and which risks require a response. A well informed decision requires an analysis of the probability of a particular risk occurring and its likely impact.