UNIT -V

RISK MANAGEMENT: Reactive Versus Proactive Risk Strategies, Risk Identification, Risk Projection, Risk Refinement, RMMM, RMMM Plan.

QUALITY MANAGEMENT: Software Quality, Informal Reviews, Formal Technical Reviews, Statistical Software Quality Assurance, Software Reliability.

RISK MANAGEMENT

Any real or potential condition that can cause injury, illness, or death to personnel; damage to or loss of a system, equipment or property; or damage to the environment. Simpler.....A threat of harm. A hazard can lead to one or several consequences.

A Risk is

- The expectation of a loss or damage (consequence)
- The combined severity and probability of a loss
- The long term rate of loss

A potential problem (leading to a loss) that may - or may not occur in the future.

- Risk Management is A set of practices and support tools to identify, analyze, and treat risks explicitly.
- Treating a risk means understanding it better, avoiding or reducing it (risk mitigation), or preparing for the risk to materialize.
- Risk management tries to reduce the probability of a risk to occur and the impact (loss) caused by risks.

Reactive versus Proactive Risk Strategies

Reactive versus Proactive Risk Strategies :

- The majority of software teams rely solely on reactive risk strategies. At best, a reactive strategy monitors the project for likely risks. Resources are set aside to deal with them, should they become actual problems.
- The software team does nothing about risks until something goes wrong. Then, the team flies into action in an attempt to correct the problem rapidly. This is often called a fire-fighting mode.
- A considerably more intelligent strategy for risk management is to be proactive.
- A proactive strategy begins long before technical work is initiated. Potential risks are identified, their probability and impact are assessed, and they are ranked by importance. Then,
- The software team establishes a plan for managing risk. The primary objective is to avoid risk, but because not all risks can be avoided, the team works to develop a contingency plan that will enable it to respond in a controlled and effective manner.

Risk always involves two characteristics:

- Risk always involves two characteristics: uncertainty—the risk may or may not happen; that is, there are no 100 percent probable risks—and loss—if the risk becomes a reality, unwanted consequences or losses will occur.
- When risks are analyzed, it is important to quantify the level of uncertainty and the degree of loss associated with each risk.
- **Different categories of risks are follows:**

1. Project risks

- ❖ Threaten the project plan. That is, if project risks become real, it is likely that the project schedule will slip and that costs will increase.
- ❖ Project risks identify potential budgetary, schedule, personnel (staffing and organization), resource, stakeholder, and requirements problems and their impact on a software project.

2. 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. Technical risks identify potential design, implementation, interface, verification, and maintenance problems.
- ❖ In addition, specification ambiguity, technical uncertainty, technical obsolescence, and ―leading-edge‖ technology are also risk factors. Technical risks occur because the problem is harder to solve than you thought it would be.

3. Business risks

- ❖ Business risks threaten the viability of the software to be built and often jeopardize the project or the product.
- ❖ Candidates for the top five business risks are

1. building an excellent product or system that no one really wants (market risk)
2. building a product that no longer fits into the overall business strategy for the company (strategic risk)
3. building a product that the sales force doesn‘t understand how to sell (sales risk)
4. losing the support of senior management due to a change in focus or a change in people (management risk)
5. losing budgetary or personnel commitment (budget risks).

Another general categorization of risks has been proposed by Charette.

4. ***Known risks*** are those that can be uncovered after careful evaluation of the project plan, the business and technical environment in which the project is being developed, and other reliable information sources (e.g., unrealistic delivery date, lack of documented requirements or software scope, poor development environment).

5. ***Predictable* risks** are extrapolated from past project experience (e.g., staff turnover, poor communication with the customer, dilution of staff effort as ongoing maintenance requests are serviced).

6. ***Unpredictable risks*** are the joker in the deck. They can and do occur, but they are extremely difficult to identify in advance.

RISK IDENTIFICATION

- Risk identification is a systematic attempt to specify threats to 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.
- There are two distinct types of risks: generic risks and product-specific risks.
- Generic risks are a potential threat to every software project.
- Product-specific risks can be identified only by those with a clear understanding of the technology, the people, and the environment that is specific to the software that is to be built.
- To identify product-specific risks, the project plan and the software statement of scope are examined, and an answer to the following question is developed: ‒What special characteristics of this product may threaten our project plan?‖
- One method for identifying risks is to create a risk item checklist.
- The checklist can be used for risk identification and focuses on some subset of known and predictable risks in the following generic subcategories:
- Product size—risks associated with the overall size of the software to be built or modified.
- Business impact—risks associated with constraints imposed by management or the marketplace.
- Stakeholder characteristics—risks associated with the sophistication of the stakeholders and the developer‘s ability to communicate with stakeholders 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.

Assessing Overall Project Risk

The following questions have been derived from risk data obtained by surveying experienced software project managers in different parts of the world.

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 its 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. Doestheprojectteamhaveexperiencewiththetechnologytobeimplemented?
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

- The project manager identify the risk drivers that affect software risk components— performance, cost, support, and schedule.

The risk components are defined in the following manner:

- **Performance risk**—the degree of uncertainty that the product will meet its requirements and be fitfor 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 theproduct 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.

Risk Projection

- Risk projection, also called risk estimation, attempts to rate each risk in two ways.
    1. The likelihood or probability that the risk is real and
    2. The consequences of the problems associated with the risk, should it occur

Managers and technical staff to perform four risk projection steps:

1. Establish a scale that reflects the perceived likelihood of a risk.
2. Delineate the consequences of the risk.
3. Estimate the impact of the risk on the project and the product.
4. Assess the overall accuracy of the risk projection so that there will be no misunderstandings.

The intent of these steps is to consider risks in a manner that leads to prioritization. No software team has the resources to address every possible risk with the same degree of rigor. By prioritizing risks, you can allocate resources where they will have the most impact.

Developing a Risk Table

- A risk table provides you with a simple technique for risk projection. A sample risk table is illustrated in Figure.
- List all the risks (no matter how remote) in the first column of the table.
- Each risk is categorized in the second column (e.g., PS implies a project size risk, BU implies a business risk).
- 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.
- Next, the impact of each risk is assessed. Each risk component is assessed, and an impact category is determined.
- The categories for each of the four risk components—performance, support, cost, and schedule—are averaged to determine an overall impact value.
- 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 drop to the bottom.

Sample Risk table prior to sorting

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 than 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	
Σ				
Σ				
Σ				

Impact values:
1—catastrophic
2—critical
3—marginal
4—negligible

2. Assessing Risk Impact

- Three factors affect the consequences that are likely if a risk does occur: its nature, its scope, and its timing.
- The nature of the risk indicates the problems that are likely if it occurs. For example, a poorly defined external interface to customer hardware (a technical risk) will preclude early design and testing and will likely lead to system integration problems late in a project.
- The scope of a risk combines the severity (just how serious is it?) with its overall distribution (how much of the project will be affected or how many stakeholders are harmed?).
- The timing of a risk considers when and for how long the impact will be felt. In most cases, you want the ―bad news‖ to occur as soon as possible, but in some cases, the longer the delay, the better.
- The overall risk exposure RE is determined using the following relationship

$$RE = P * C$$

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

Risk Mitigation, Monitoring, and Management:RMMM

- An effective strategy for dealing with risk must consider three issues (Note: these are not mutually exclusive)

  - Risk mitigation
  - Risk monitoring
  - Risk management and contingency planning

- **Risk mitigation** - is the primary strategy and is achieved through a plan
  - Example: Risk of high staff turnover

- Meet with current staff to determine causes for turnover (e.g., poor working conditions, low pay, competitive job market)
- 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 ensure that documents are developed in a timely manner
- Conduct peer reviews of all work (so that more than one person is "up to speed")
- Assign a backup staff member for every critical technologist.
- During **risk monitoring**, the project manager monitors factors that may provide an indication ofwhether a risk is becoming more or less likely
- **Risk management** and contingency planning assume that mitigation efforts have failed and thatthe risk has become a reality
- RMMM steps incur additional project cost
  - Large projects may have identified 30 – 40 risks
- Risk is not limited to the software project itself
  - Risks can occur after the software has been delivered to the user
- Software safety and hazard analysis
  - These are software quality assurance activities that focus on the identification and assessment of potential hazards that may affect software negatively and cause an entire system to fail
  - If hazards can be identified early in the software process, software design features can be specified that will either eliminate or control potential hazards.
  - It is important to note that risk mitigation, monitoring, and management (RMMM) steps

incur additional project cost

- The RMMM plan may be a part of the software development plan or may be a separate document
- Once RMMM has been documented and the project has begun, the risk mitigation, and monitoring steps begin
  - Risk mitigation is a problem avoidance activity
  - Risk monitoring is a project tracking activity
- Risk monitoring has three objectives
  - To assess whether predicted risks do, in fact, occur
  - To ensure that risk aversion steps defined for the risk are being properly applied
  - To collect information that can be used for future risk analysis
- The findings from risk monitoring may allow the project manager to ascertain what risks caused which problems throughout the project.

Quality Concepts

Variation control is the heart of quality control

Form one project to another, we want to minimize the difference between the predictedresources needed to complete a project and the actual resources used, including staff, equipment, and calendar time

Quality of design

Refers to characteristics that designers specify for the end product

Quality Management

Quality of conformance

Degree to which design specifications are followed in manufacturing the product

Quality control

Series of inspections, reviews, and tests used to ensure conformance of a work product to its specifications

Quality assurance

Consists of a set of auditing and reporting functions that assess the effectiveness andcompleteness of quality control activities

Cost of Quality

Prevention costs

Quality planning, formal technical reviews, test equipment, training

Appraisal costs

In-process and inter-process inspection, equipment calibration and maintenance, testing

Failure costs

rework, repair, failure mode analysis

External failure costs

Complaint resolution, product return and replacement, help line support, warranty work.

Software Quality Assurance

1, Software quality assurance (SQA) is the concern of every software engineer to reducecost and improve product time-to-market.

2. A Software Quality Assurance Plan is not merely another name for a test plan, thoughtest plans are included in an SQA plan.

3. SQA activities are performed on every software project.

4. Use of metrics is an important part of developing a strategy to improve the quality ofboth software processes and work products.

Software Quality Assurance

1. Definition of Software Quality serves to emphasize:
2. Conformance to software requirements is the foundation from which software quality ismeasured.
3. Specified standards are used to define the development criteria that are used to guide themanner in which software is engineered.
4. Software must conform to implicit requirements (ease of use, maintainability, reliability,etc.) as well as its explicit requirements

.

SQA Activities

1. Prepare SQA plan for the project.
2. Participate in the development of the project's software process description.
3. Review software engineering activities to verify compliance with the defined softwareprocess.
4. Audit designated software work products to verify compliance with those defined as partof the software process.
5.Ensure that any deviations in software or work products are documented and handledaccording to a documented procedure.
6. Record any evidence of noncompliance and reports them to management.

Software Reviews

1. Purpose is to find errors before they are passed on to another software engineeringactivity or released to the customer.
2. Software engineers (and others) conduct formal technical reviews (FTRs) for softwarequality assurance.
3. Using formal technical reviews (walkthroughs or inspections) is an effective means forimproving software quality.

Formal Technical Review

A FTR is a software quality control activity performed by software engineers andothers.
The objectives are:

1. To uncover errors in function, logic or implementation for any representation of thesoftware.
2. To verify that the software under review meets its requirements.
3. To ensure that the software has been represented according to predefined standards.
4. To achieve software that is developed in a uniform manner and
5. To make projects more manageable.

Review meeting in FTR

1. The Review meeting in a FTR should abide to the following constraints
2. Review meeting members should be between three and five.
3. Every person should prepare for the meeting and should not require more than two hours ofwork for each person.
4. The duration of the review meeting should be less than two hours.
5. The focus of FTR is on a work product that is requirement specification, a detailedcomponent design, a source code listing for a component.
6. The individual who has developed the work product i.e, the producer informs the projectleader that the work product is complete and that a review is required.
7. The project leader contacts a review leader, who evaluates the product for readiness, generates copy of product material and distributes them to two or three review members foradvance preparation

8. Each reviewer is expected to spend between one and two hours reviewing the product,making notes The review leader also reviews the product and establish an agenda for the review meeting
9. The review meeting is attended by review leader, all reviewers and the producer.
10. One of the reviewer act as a recorder,who notes down all important points discussed inthe meeting.

The meeting(FTR) is started by introducing the agenda of meeting and then **the producer introduces his product. Then the producer "walkthrough" the** product, thereviewers raise issues which they have prepared in advance.

If errors are found the recorder notes down.

Review reporting and Record keeping

During the FTR, a reviewer(recorder) records all issues that have been raisedA review summary report answers three questions

What was reviewed?
Who reviewed it?
What were the findings and conclusions?
Review summary report is a single page form with possible attachments

The review issues list serves two purposes

1. To identify problem areas in the product
2. To serve as an action item checklist that guides the producer as corrections are made

Review Guidelines

1. Review the product, not the producer
2. Set an agenda and maintain it
3. Limit debate and rebuttal
4. Limit the number of participants and insist upon advance preparation.
   Develop a checklist for each product i.e likely to be reviewed
   Allocate resources and schedule time for FTRS
5. Conduct meaningful training for all reviewer
   Review your early reviews

Software Defects :

A. Industry studies suggest that design activitiesintroduce 50-65% of all defects or errors during the software process
B. Review techniques have been shown to be upto 75% effective in uncovering design flaws which ultimately reduces the cost of subsequent activities in the software process

Statistical Quality Assurance

1. Information about software defects is collected and categorized.
2. Each defect is traced back to its cause
3. Using the Pareto principle (80% of the defects can be traced to 20% of the causes)isolate the "vital few" defect causes.

Six Sigma for Software Engineering:

The most widely used strategy for statistical quality assurance

Three core steps:

1. Define customer requirements, deliverables, and project goals via well-definedmethods of customer communication.
2. Measure each existing process and its output to determine current qualityperformance (e.g., compute defect metrics)
3. Analyze defect metrics and determine vital few causes.

For an existing process that needs improvement

1. Improve process by eliminating the root causes for defects
2. Control future work to ensure that future work does not reintroduce causes ofdefects

If new processes are being developed

1. Design each new process to avoid root causes of defects and to meetcustomer requirements
2. Verify that the process model will avoid defects and meet customerrequirements

Software Reliability

Defined as the probability of failure free operation of a computer program in a specified environment for a specified time period Can be measured directly and estimated using historical anddevelopmental data.

Software reliability problems can usually be traced back to errors in design or implementation.

Measures of Reliability

Mean time between failure (MTBF) = MTTF + MTTR

MTTF = mean time to failure

MTTR = mean time to repair

Availability = [MTTF / (MTTF + MTTR)] x 100%

ISO 9000 Quality Standards

ISO (International Standards Organization) is a group or consortium of 63 countries established to plan and fosters standardization. ISO declared its 9000 series of standards in 1987. It serves as a reference for the contract between independent parties. The ISO 9000 standard determines the guidelines for maintaining a quality system. The ISO standard mainly addresses operational methods and organizational methods such as responsibilities, reporting, etc. ISO 9000 defines a set of guidelines for the production process and is not directly concerned about the product itself.

Types of ISO 9000 Quality Standards

The ISO 9000 series of standards is based on the assumption that if a proper stage is followed for production, then good quality products are bound to follow automatically. Thetypes of industries to which the various ISO standards apply are as follows.

1. **ISO 9001:** This standard applies to the organizations engaged in design, development, production, and servicing of goods. This is the standard that applies to most software development organizations.

2. **ISO 9002:** This standard applies to those organizations which do not design products but are only involved in the production. Examples of these category industries contain steel and car manufacturing industries that buy the product and plants designs from external sources and are engaged in only manufacturing those products. Therefore, ISO 9002 does not applyto software development organizations.

3. **ISO 9003:** This standard applies to organizations that are involved only in the installationand testing of the products. For example, Gas companies.

An organization determines to obtain ISO 9000 certification applies to ISO registrar office for registration. The process consists of the following stages:

1. **Application:** Once an organization decided to go for ISO certification, it applies to the registrar forregistration.

2. **Pre-Assessment:** During this stage, the registrar makes a rough assessment of the organization.

3. **Document review and Adequacy of Audit:** During this stage, the registrar reviews the documentsubmitted by the organization and suggest an improvement.

4. **Compliance Audit:** During this stage, the registrar checks whether the organization has compiled thesuggestion made by it during the review or not.

5. **Registration:** The Registrar awards the ISO certification after the successful completion of all thephases.

6. **Continued Inspection:** The registrar continued to monitor the organization time by time.