Risk Management And Project Scheduling

Unit 2

Red: indicates important

Contents

- Identification of risks
- Management of risks
- Factors affecting the task set for the project
- Scheduling techniques

Identification Of Software Risks

- Risk may be defined as a potential problem. It may or may not occur. But it should be always assumed that it may occur and necessary steps are to be taken.
- Different types of software risks:
  - Skills or knowledge: The persons involved in activities of problem analysis, design, coding and testing have to be fully aware of the activities and various techniques at each phase of the software development cycle. If they have partial knowledge or lacks adequate skill, the products may face many risks at the current stage or at later stage.
  - Interface modules: Complete software contains various modules and each module sends and receives information to other modules and their concerned data types have to match.
  - Poor knowledge of tools: If the team or individual members have poor knowledge of tools used in the software product, then the final product will have many risks.
  - Programming Skills: The code developed has to be efficient, thereby, occupying less memory space and less CPU cycles to compute given task. The software product should be able to implement various object oriented techniques and be able to catch exceptions in case of errors. Various data values have to be checked and in case of improper values, appropriate messages have to be displayed. If this is not, then it leads to risk.

- Management Issues: The management should give proper training to the project staff arrange some recreation activities., give bonus and promotions and interact with all members of the project and try to solve their necessities at the best. It should take care that team members and the project manager have healthy coordination, and in case there are some differences they should solve or make minor shuffles.
- Updates in the hardware resources: The team should be aware of the latest updates in the hardware resources, such as CPU(Intel P4, Motorola series etc..), peripherals etc.. In case the developer makes a product, and later in the market, a new product is released, the product should support minimum feature. Otherwise, it is considered a risk, and may even lead to failure of the project.
- Extra support: The software should be able to support a set of few extra features in the vicinity of the product to be developed.
- Customer Risks: Customer should have proper knowledge of the product needed, and should not be in a hurry to get the work done. He should take care that all the features are implemented and tested. He should take help of few external personnel as needed to test the product and should arrange for demonstrations.
- External risks: The software should have backup in CD, tapes, etc.., fully encrypted with full license facilities. The software can be stored at various important locations to avoid any external calamities like floods, earthquake. Encryption is maintained such that no external persons from the team can tap the source code.

- Commercial risks: The organization should be aware of various competing vendors in the market and various risks involved if their product is not delivered on time. They should have statistics of projects and risks involved from their previous experience and should have skilled personnel.

Monitoring of risks

- Various risks are identified and a risk monitor table with attributes like risk name, module name, team members involved, line of code, codes affecting the risk, hardware resources, etc.. Is maintained. If the project is continued for 2-3 weeks, and then further the risk table is also updated. Its seen whether the ripple effect in the table, due to continuity of old risks. Risk monitors can change the ordering of risks to make the table easy for computation.
- Risk table monitor

Table 2.1: Risk table monitor

Sl. No.	Risk Name	Week 1	Week 2	Remarks
1.	Module compute()	Line 5, 8, 20	Line 5,25	Priority 3
2.	More memory and peripherals	Module f1(), f5() affected	Module f2() affected	Priority 1
.....	.....	.....	.....	.....

- In the table, there is a risk in module compute() where there is a risk in line 5,8, and 20 in week 1. In week2, risks are present in lines 5 and 25. Risks reduced in week 2. The priority is set to 3.

Management of Risks

- Risk management plays an important role in ensuring that the software product is error free.
- **Risk management:**
- A priority is given to risk and the highest priority risk is handled first.
- Factors of the risk are who are the involved team members, what hardware and software items are needed etc..
- The risk manager does scheduling of risks.
- Risk management can be further categorized as follows:
  1. Risk Avoidance:
     a. Risk anticipation
     b. Risk tools

2. Risk Detection
   a. Risk analysis
   b. Risk category
   c. Risk prioritization
3. Risk Control
   a. Risk pending
   b. Risk Resolution
   c. Risk not solvable
4. Risk Recovery
   a. Full
   b. Partial
   c. Extra/alternate features

Risk Manager tool

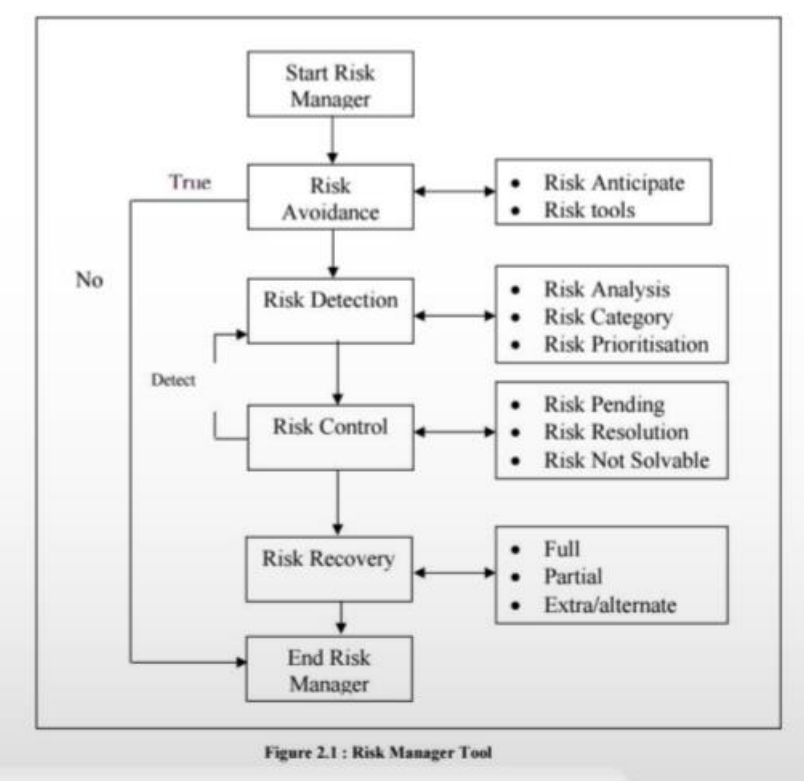

Figure 2.1 : Risk Manager Tool

- Its clear from the above diagram that first phase is to avoid risk by anticipating and using tools from previous project history. If there is no risk, risk manager halts. If there is risk, detection is done using various risk analysis techniques and further prioritizing risks. In the next phase, risk is controlled by pending risks, resolving risks and in the worst case lowering the priority. Lastly risk recovery is done fully, partially or an alternate solution is found.
- Risk Avoidance :
  - Risk anticipation: Various risk anticipation rules are listed according to standards from previous projects experience, and also as mentioned by the project manager.
  - Risk tools: Risk tools are used to test whether the software is risk free. The tools have built-in data base of available risk areas and can be updated depending upon the type of project.

- Risk Detection : Risk detection algorithm detects a risk and it can be categorically stated as:
  - Risk Analysis :

    - In this phase, the risk is analyzed with various hardware and software parameters as probabilistic occurrence(pr), weight factor(wf)(hardware resources, persons, lines of codes),risk exposure (pr * wf)
    - Example: Table 2.1 page 22 block 2(Risk analysis table)
    - Maximum value of risk exposure means the product has to be solved as soon as possible and is given high priority. Risk analysis table is mentioned as example.
  - Risk category:
    - Risk identification can be from various factor like persons involved in the team, management issues, customer specification and feedback, environment, commercial, technology, etc.. Once proper category is identified, priority is given depending upon the urgency of the product.
  - Risk Prioritization:

    - Depending upon the entries of the risk analysis table, the maximum risk exposure is given high priority and has to be solved first.

Risk Control

- Once the prioritization is done, the next step is to control various risks as follows:
  - Risk Pending: According to the priority, low priority risks are pushed at the end of the queue with a view of various resources (hardware, man power, software) and in case it takes more time their priority is made higher.
  - Risk Resolution: Risk manager makes a strong resolve how to solve the risk.
  - Risk elimination: This action leads to serious error in software.
  - Risk transfer: If the risk is transferred to some part of the module, then risk analysis table entries get modified. Thereby, again risk manager will control high priority risk.
  - Disclosures: Announce the risk of less priority to the customer or display message box as a warning. And thereby the risk is left out to the user, such that he should take proper steps during data entry.
  - Risk not solvable: If a risk takes more time and more resources, then it is dealt in its totality in the business aspect of the organization and there by it is notified to the customer, and the team member proposes an alternate solution. There is a slight variation in the customer specification after consultation.

Risk Recovery

- Full: The risk analysis table is scanned and if the risk is fully solved, then corresponding entry is deleted from the table.
- Partial: The risk analysis table is scanned and due to partially solved risks, the entries in the table are updated and thereby priorities are also updated.
- Extra/alternate feature: Sometimes its difficult to remove some risks, and in that case, we can add a few extra features, which solves the problem. Therefore, a bit of coding is done to get away from the risk. This is later documented or notified to the customer.

Formulating a task set for the project

- **Factors affecting the task set for the project:**
  - Technical staff expertise: All staff members should have sufficient technical expertise for timely implementation of the project. Meetings have to be conducted, weekly and status reports are to be generated.
  - Customer satisfaction: Customer has to be given timely information regarding the status of the project. If not, there will be a communication gap between the customer and organization.
  - Technology update: Latest tools and existing tested modules have to be used for fast and efficient implementation of the project.
  - Full or partial implementation of the project: In case, the project is very large and to meet the market requirements, the organization has to satisfy the customer with at least a few modules. The remaining modules can be delivered at a later stage.
  - Time allocation: The project has to be divided into various phases and time for each phase has to be given in terms of person-months, module months, etc..
  - Module binding: Module has to bind to various technical staff for design, implementation and testing phases. Their necessary independencies have to be mentioned in a flow-chart.

- Milestones: The outcome for each phase has to be mentioned in terms of quality, specifications implemented, limitations of the module and latest updates that can be implemented.
- Validation and verification: The number of modules verified according to customer specification and the number of modules validated according to customer's expectations are to be specified.

Choosing the tasks of software engineering

- Once the task set has been defined, the next step is to choose the tasks for software project. Depending upon the software process model like linear sequential, iterative, evolutionary model etc., the corresponding ask is selected. How to choose the tasks for development is as follows:
  - Scope: Overall scope of the project
  - Scheduling and planning: Scheduling of various modules and their milestones, preparation of weekly reports, etc.
  - Technology used: Latest hardware and software used.
  - Customer interaction: Obtaining feedback from the customer.
  - Constraints and limitation: Various constraints in the project and how they can be solved. Limitations in the modules and how they can be implemented in the next phase of the project, etc..
  - Risk Assessment: Risk involved in the project with respect to limitations in the technology and resources.

Scheduling Methods

- Scheduling of a software project means prioritizing various tasks with respect to their cost, time and duration.
- Depending upon the project, scheduling methods can be static or dynamic in implementation.
- Different Scheduling Techniques :
  1. Work Breakdown Structure:
     - its a deliverable-oriented breakdown of a project into smaller components.
     - The project is scheduled in various phases following a bottom-up or top-down approach.
     - A tree like structure is followed without any loops.
     - Aka WBS.
     - At each phase or step, milestones and deliverables are mentioned with respect to requirements.
     - It shows the overall breakup flow of the project and does not indicate any parallel flow.

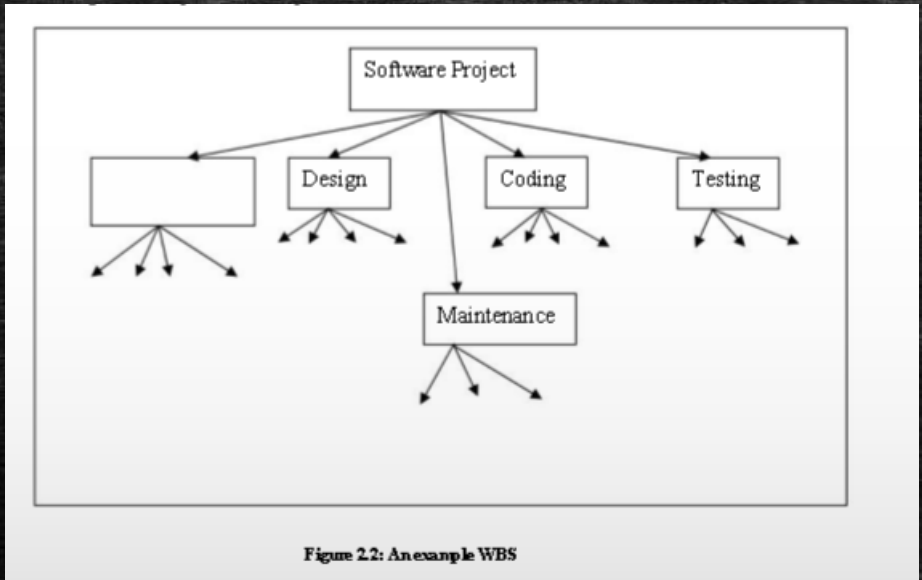

Figure 2.2: An example WBS

- In the above example project is divided into requirement and analysis, design, coding, testing and maintenance phase. Further requirements and analysis is divided into R1,R2,…Rn; Design is divided into D1,D2,….Dn; Coding is divided into C1,C2,…Cn; Testing is divided into T1,T2…Tn; and maintenance is divided into M1,M2…Mn; If the project is complex, then further sub division is done. Upon completion of each stage, integration is done.

2. Flow Graph:
   - Modules are represented as nodes with edges connecting nodes.
   - Dependency between nodes is shown by the flow of data between nodes.
   - Nodes indicate milestones and deliverables with the corresponding module implemented.
   - Cycles are not allowed in the graph.
   - Start and end nodes indicate the source and terminating nodes of the flow.
   - Example: Fig 2.3 page 26 block 2
   - In the above example, M1 is the starting module and the data flows to M2 and M3. The combined data from M2 and M3 flow to M4 and finally the project terminates. Arrows indicate the flow of information between modules.

3. Gantt Chart or Time Line Charts:
   - It can be developed for the entire project or separate chart can be developed for each function.
   - A tabular form is maintained where rows indicate the tasks with milestones and columns indicate duration. The horizontal bars across the columns indicate duration of task and the circles indicate milestones.
   - Example: Fig 2.4 Page 26 block 2
4. Program Evaluation Review Technique:
   - Used for high-risk projects with various estimation parameters.
   - For each module, duration is estimated as:
     1. Time taken to complete a project or module under normal conditions, tnormal.
     2. Time taken to complete a project or module with minimum time, tmin.
     3. Time taken to complete a project or module with maximum time, tmax.
     4. Time taken to complete a project from previous related history, Thistory
   - An average of tnormal, tmin, tmax, Thistory is taken depending upon the project. Sometimes various weights are added as 4*tnormal,5*tmin,0.9* tmax and 2* Thistory to estimate the time for a project or module. Parameter fixing is done by the project manager.

Software Project Plan

- Planning is very important in every aspect of development work.
- Software project plan can be viewed as the following:
  1. Within the organization: how the project is implemented? What are various constraints(time, cost, staff)? What is market strategy?
  2. With respect to customer: Weekly or timely meetings with the customer with presentations on status reports. Customers feedback is also taken and further modifications and developments are done. Project milestones and deliverables are also presented to the customer.
- Steps for successful software project:
  - Select a project
    - Identifying project's aims and objectives
    - Understanding requirements and specification
    - Methods of analysis, design and implementation
    - Testing techniques
    - Documentation

- Project milestones and deliverables
- Budget Allocation
  - Exceeding limits within control
- Project estimates
  - Cost
  - Time
  - Size of code
  - Duration
- Resource Allocation
  - Hardware
  - Software
  - Previous relevant project information
  - Digital Library

- Risk Management
  - Risk avoidance
  - Risk Detection
  - Risk Control
  - Risk Recovery
- Scheduling techniques
  - Work Breakdown Structure
  - Activity Graph
  - Critical path method
  - Gantt Chart
  - Program Evaluation Review Techniques
- People
  - Staff Recruitment
  - Team management
  - Customer interaction
- Quality control and standard