
UNIT 2 RISK MANAGEMENT AND PROJECT SCHEDULING

Structure	Page Nos.
2.0 Introduction	18
2.1 Objectives	18
2.2 Identification of Software Risks	18
2.3 Monitoring of Risks	20
2.4 Management of Risks	20
2.4.1 Risk Management	
2.4.2 Risk Avoidance	
2.4.3 Risk Detection	
2.5 Risk Control	22
2.6 Risk Recovery	23
2.7 Formulating a Task Set for the Project	24
2.8 Choosing the Tasks of Software Engineering	24
2.9 Scheduling Methods	25
2.10 The Software Project Plan	27
2.11 Summary	28
2.12 Solutions/Answers	28
2.13 Further Readings	30

2.0 INTRODUCTION

As human beings, we would like life to be free from dangers, difficulties and any risks of any type. In case a risk arises, we would take proper measures to recover as soon as possible. Similarly, in software engineering, risk management plays an important role for successful deployment of the software product. Risk management involves monitoring of risks, taking necessary actions in case risk arises by applying risk recovery methods.

2.1 OBJECTIVES

After going through this unit, you should be able to:

- know the meaning of risk;
 - identify risks; and
 - manage the risks
-

2.2 IDENTIFICATION OF SOFTWARE RISKS

A risk may be defined as a potential problem. It may or may not occur. But, it should always be assumed that it may occur and necessary steps are to be taken.

Risks can arise from various factors like improper technical knowledge or lack of communication between team members, lack of knowledge about software products, market status, hardware resources, competing software companies, etc.

Basis for 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. In case, they have partial knowledge or lacks adequate skill, the products may face many risks at the current stage of development or at later stages.

- **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, since it is not thoroughly tested.
- **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 done, then it leads to risk, thereby creating panic in the software computations.
- **Management Issues :** The management of the organisation 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 latest 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 features. Otherwise, it is considered a risk, and may lead to the failure of the project.
- **Extra support:** The software should be able to support a set of a 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 the help of a few external personnel as needed to test the product and should arrange for demonstrations with a set of technical and managerial persons from his office.
- **External Risks:** The software should have backup in CD, tapes, etc., fully encrypted with full licence facilities. The software can be stored at various important locations to avoid any external calamities like floods, earthquakes, etc. Encryption is maintained such that no external persons from the team can tap the source code.
- **Commercial Risks:** The organisation 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.

2.3 MONITORING OF RISKS

Various risks are identified and a risk monitor table with attributes like risk name, module name, team members involved, lines of code, codes affecting this risk, hardware resources, etc., is maintained. If the project is continued further to 2-3 weeks, and then further the risk table is also updated. It is seen whether there is a ripple effect in the table, due to the continuity of old risks. Risk monitors can change the ordering of risks to make the table easy for computation. *Table 2.1* depicts a risk table monitor. It depicts the risks that are being monitored.

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
.....

The above risk table monitor has a risk in module compute () where there is a risk in line 5, 8 and 20 in week 1. In week 2, risks are present in lines 5 and 25. Risks are reduced in week 2. The priority 3 is set. Similarly, in the second row, risk is due to more memory and peripherals, affecting module f1 (), f5 () in week-1. After some modifications in week 2, module f2 () is affected and the priority is set to 1.

Check Your Progress 1

- 1) Define the term risk and how it is related to software engineering.

.....
.....

- 2) List at least two risks involved with the management of team members.

.....
.....

- 3) What are commercial risks?

.....
.....

- 4) What do you mean by monitoring of risks and describe risk table.

.....
.....

- 5) Mention any two ways to prioritise risks.

.....
.....

2.4 MANAGEMENT OF RISKS

Risk management plays an important role in ensuring that the software product is error free. Firstly, risk management takes care that the risk is avoided, and if it not avoidable, then the risk is detected, controlled and finally recovered.

The flow of risk management is as follows:

2.4.1 Risk Management

A priority is given to risk and the highest priority risk is handled first. Various factors of the risk are who are the involved team members, what hardware and software items are needed, where, when and why are resolved during risk management. The risk manager does scheduling of risks. Risk management can be further categorised as follows:

1. Risk Avoidance
 - a. Risk anticipation
 - b. Risk tools
2. Risk Detection
 - a. Risk analysis
 - b. Risk category
 - c. Risk prioritisation
3. Risk Control
 - a. Risk pending
 - b. Risk resolution
 - c. Risk not solvable
4. Risk Recovery
 - a. Full
 - b. Partial
 - c. Extra/alternate features

Figure 2.1 depicts a risk manager tool.

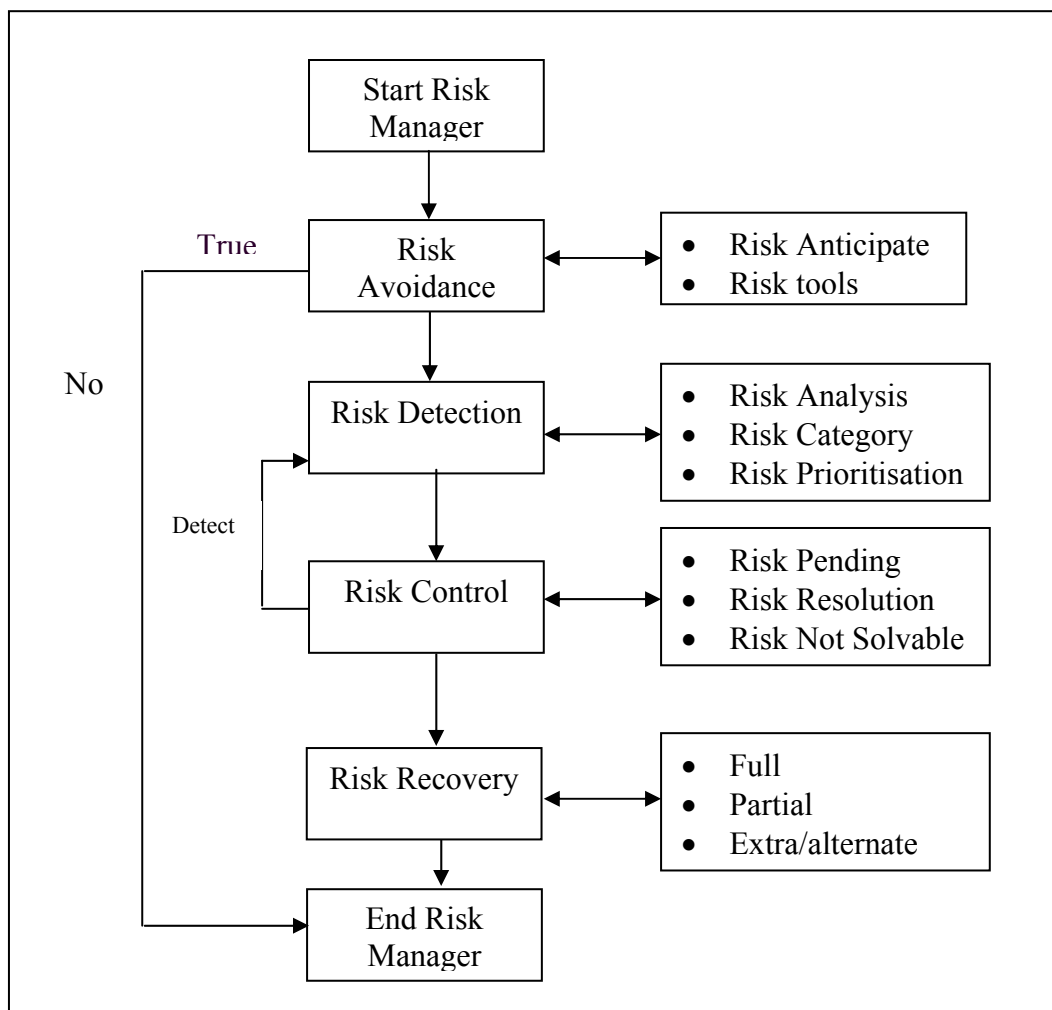


Figure 2.1 : Risk Manager Tool

From the *Figure 2.1*, it is clear that the first phase is to avoid risk by anticipating and using tools from previous project history. In case there is no risk, risk manager halts. In case there is risk, detection is done using various risk analysis techniques and further prioritising risks. In the next phase, risk is controlled by pending risks, resolving risks and in the worst case (if risk is not solved) lowering the priority. Lastly, risk recovery is done fully, partially or an alternate solution is found.

2.4.2 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.

2.4.3 Risk Detection

The 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, lines of code, persons), risk exposure ($pr * wf$).

Table 2.1 depicts a risk analysis table.

Table 2.1: Risk analysis table

Sl.No.	Risk Name	Probability of occurrence (pr)	Weight factor (wf)	Risk exposure (pr * wf)
1.	Stack overflow	5	15	75
2.	No Password forgot option	7	20	140
....

Maximum value of risk exposure indicates that the problem has to solved as soon as possible and be given high priority. A risk analysis table is maintained as shown above.

Risk Category: Risk identification can be from various factors 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 Prioritisation: Depending upon the entries of the risk analysis table, the maximum risk exposure is given high priority and has to be solved first.

2.5 RISK CONTROL

Once the prioritisation 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, etc.
- **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 organisation and thereby 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.

2.6 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 features : Sometimes it is 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.

Check Your Progress 2

- 1) Define the term risk management

.....

.....

- 2) What are various phases of risk manager?

.....

.....

- 3) What are the attributes mentioned in risk analysis table?

.....

.....

- 4) What is meant by risk resolution?

.....

.....

- 5) Why do we add extra features to recover from risks?

.....

.....

2.7 FORMULATING A TASK SET FOR THE PROJECT

The objective of this section is to get an insight into project scheduling by defining various task sets dependent on the project and choosing proper tasks for software engineering.

Various static and dynamic scheduling methods are also discussed for proper implementation of 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 might be a communication gap between the customer and the organisation.
- **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 organisation 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 inter-dependencies 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 (according to the market strategy).
- **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.

2.8 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 task is selected. From the above task set, let us consider how to choose tasks for project development (as an example) 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 limitations** : 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.

2.9 SCHEDULING METHODS

Scheduling of a software project can be correlated to prioritising various tasks (jobs) with respect to their cost, time and duration. Scheduling can be done with resource constraint or time constraint in mind. Depending upon the project, scheduling methods can be static or dynamic in implementation.

Scheduling Techniques

The following are various types of scheduling techniques in software engineering are:

- **Work Breakdown Structure** : The project is scheduled in various phases following a bottom-up or top-down approach. A tree-like structure is followed without any loops. At each phase or step, milestone and deliverables are mentioned with respect to requirements. The work breakdown structure shows the overall breakup flow of the project and does not indicate any parallel flow.

Figure 2.2 depicts an example of a work breakdown structure.

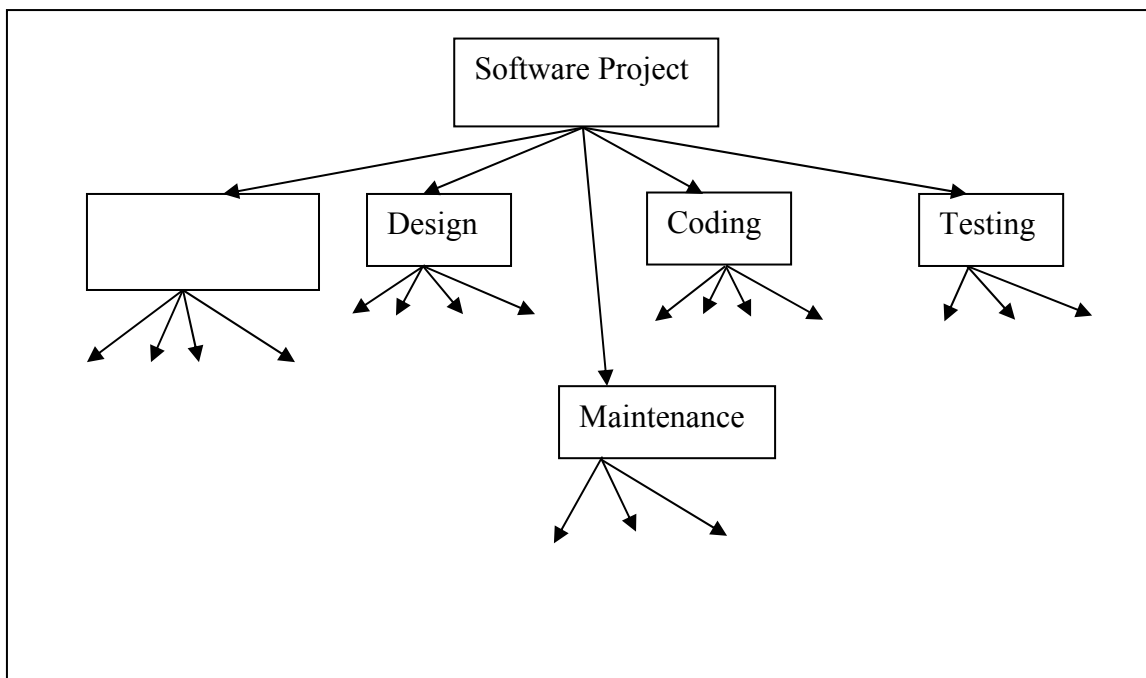
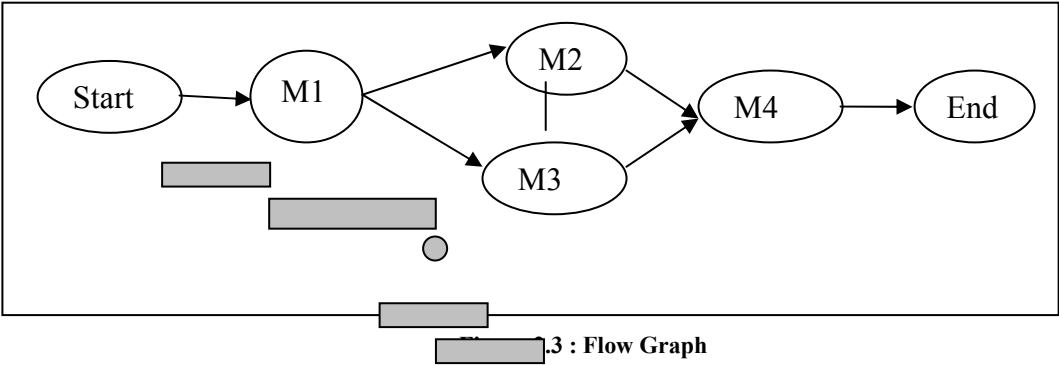


Figure 2.2: An example WBS

The project is split into requirement and analysis, design, coding, testing and maintenance phase. Further, requirement 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 the completion of each stage, integration is done.

- **Flow Graph :** Various modules are represented as nodes with edges connecting nodes. Dependency between nodes is shown by 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. *Figure 2.3* depicts a flow graph.



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. In certain projects, time schedule is also associated with each module. The arrows indicate the flow of information between modules.

- **Gantt Chart or Time Line Charts :** A Gantt chart can be developed for the entire project or a separate chart can be developed for each function. A tabular form is maintained where rows indicate the tasks with milestones and columns indicate duration (weeks/months) . The horizontal bars that spans across columns indicate duration of the task. *Figure 2.4* depicts a Gantt Chart. The circles indicate the milestones.

Tasks	Week1				Week2				Week3				Week4			
A																
i a1																
ii a2																
Milestone																
B																
i b1																
ii b2																
Milestone																
C																
i c1																
Milestone																
D																
i d1																
ii d2																
Milestone																

Figure 2.4: Gantt Chart

- **Program Evaluation Review Technique :** Mainly used for high-risk projects with various estimation parameters. For each module in a project, duration is estimated as follows:
 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 (all resources available), t_{min} .
3. Time taken to complete a project or module with maximum time (resource constraints), t_{max} .
4. Time taken to complete a project from previous related history, $T_{history}$.

An average of t_{normal} , t_{min} , t_{max} and $t_{history}$ is taken depending upon the project. Sometimes, various weights are added as $4*t_{normal}$, $5*t_{min}$, $0.9*t_{max}$ and $2*t_{history}$ to estimate the time for a project or module. Parameter fixing is done by the project manager.

2.10 THE SOFTWARE PROJECT PLAN

Planning is very important in every aspect of development work. Good managers carefully monitor developments at various phases. Improper planning leads to failure of the project. Software project plan can be viewed as the following :

1. Within the organisation: How the project is to be implemented? What are various constraints(time, cost, staff) ? What is market strategy?
2. With respect to the customer: Weekly or timely meetings with the customer with presentations on status reports. Customer feedback is also taken and further modifications and developments are done. Project milestones and deliverables are also presented to the customer.

For a successful software project, the following steps can be followed:

- 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 Technique
- People
 - Staff Recruitment
 - Team management
 - Customer interaction
- Quality control and standard

All of the above methods/techniques are not covered in this unit. The student is advised to study references for necessary information.

Check Your Progress 3

- 1) Mention at least two factors to formulate a task set for a software project.

.....

.....

- 2) What are the drawbacks of work breakdown structure?

.....

.....

- 3) What are the advantages of Gantt chart?

.....

.....

- 4) What is the purpose of software project plan?

.....

.....

2.11 SUMMARY

This unit describes various risk management and risk monitoring techniques. In case, major risks are identified, they are resolved and finally risk recovery is done. Risk manager takes care of all the phases of risk management. Various task sets are defined for a project from the customer point of view, the developer's point of view, the market strategy view, future trends, etc. For the implementation of a project, a proper task set is chosen and various attributes are defined. For successful implementation of a project, proper scheduling (with various techniques) and proper planning are done.

2.12 SOLUTIONS/ANSWERS

Check Your Progress 1

- 1) Any problem that occurs during customer specification, design, coding, implementation and testing can be termed as a risk. If they are ignored, then they propagate further down and it is termed ripple effect. Risk management deals with avoidance and detection of risk at every phase of the software development cycle.

- 2) Two risks involved with team members are as follows:
 - Improper training of the technical staff.
 - Lack of proper communication between the developers.
- 3) The organisation should be aware of various competing vendors in the market, and various risks involved if the product is not delivered on time. It should have statistics of projects and risks involved from their previous experience or should have expertise personnel. The organisation should take care that the product will be able to support upcoming changes in hardware and software platforms.
- 4) Monitoring of risks means identifying problems in software functions, hardware deficiencies (lack of memory , peripherals, fast cpu and so on), etc. Risk table has entries for all the risk types and their timely weekly solution . Priorities of various risks are maintained.
- 5) Risks can be prioritised upon their dependencies on other modules and external factors. If a module is having many dependencies then its priority is given higher value compared to independent modules. If a module often causes security failure in the system, its priority can be set to a higher value.

Check Your Progress 2

- 1) Risk management means taking preventive measures for a software project to be free from various types of risks such as technical, customer, commercial, etc.
- 2) Various phases of risk management are risk avoidance, risk detection, risk analysis, risk monitoring, risk control and risk recovery.
- 3) Attributes mentioned in the risk analysis table are risk name, probability of occurrence of risk, weight factor and risk exposure.
- 4) Risk resolution means taking final steps to free the module or system from risk. Risk resolution involves risk elimination, risk transfer and disclosure of risk to the customer.
- 5) Some times, it is difficult to recover from the risk and it is better to add extra features or an alternate solutions keeping in view of customer specification with slight modifications in order to match future trends in hardware and software markets.

Check Your Progress 3

- 1) The two factors to formulate a task set for a software project are as follows:
 - Customer satisfaction
 - Full or partial implementation of the project
- 2) Work breakdown structure does not allow parallel flow design.
- 3) Gantt chart or time line chart indicates timely approach and milestone for each task and their relevant sub tasks.
- 4) Software project plan indicates scope of the project, milestones and deliverables, project estimates, resource allocation, risk management, scheduling techniques and quality control and standard .

2.13 FURTHER READINGS

- 1) *Software Engineering*, Ian Sommerville; *Sixth Edition, 2001*, Pearson Education.
- 2) *Software Engineering – A Practitioner’s Approach*, Roger S. Pressman; McGraw-Hill International Edition.

Reference websites

<http://www.rspa.com>

<http://www.ieee.org>

<http://www.ncst.ernet.in>