	Estimation
...d Gaffney model	E = – 91.4 + 0.355 FP
Kemerer model	E = – 37 + 0.96 FP
Small project regression model	E = – 12.88 + 0.405 FP

Each of these models shows different result for the same value of LOC an FP. Therefore estimation model must be calibrated for local models.

8.5.2 COCOMO Model

COCOMO is one of the most widely used software estimation models in the world. This model is developed in 1981 by **Barry Boehm** to give an estimate of the number of man-months it will take to develop a software product. COCOMO predicts the efforts and schedule of a software product based on **size** of the software. COCOMO stands for "**CO**nstructive **CO**st **MO**del".

COCOMO has three different models that reflect the complexity -

- Basic model
- Intermediate model
- Detailed model.

Similarly there are three classes of software projects.

1) **Organic mode :** In this mode, relatively small, simple software projects with a small team are handled. Such a team should have good application experience to less rigid requirements.

2) **Semi-detached projects :** In this class an intermediate projects in which teams with mixed experience level are handled. Such projects may have mix of rigid and less than rigid requirements.

3) **Embedded projects :** In this class, projects with tight hardware, software and operational constraints are handled.

Let us understand each model in detail.

1) **Basic Model :** The basic COCOMO model estimates the software development effort using only **Lines of Code**. Various equations in this model are -

$$E = a_b \, (KLOC)^{b_b}$$

$$D = C_b \, (E)^{d_b}$$

$$P = E/D$$

Where E is the **effort** applied in person-months.

D is the development time in chronological months.

KLOC means kilo line of code for the project.

P is total number of persons required to accomplish the project.

The coefficients a_b, b_b, c_b, d_b for three modes are as given below.

Software projects	a_b	b_b	c_b	d_b
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Table 8.5.1

Merits of basic COCOMO model

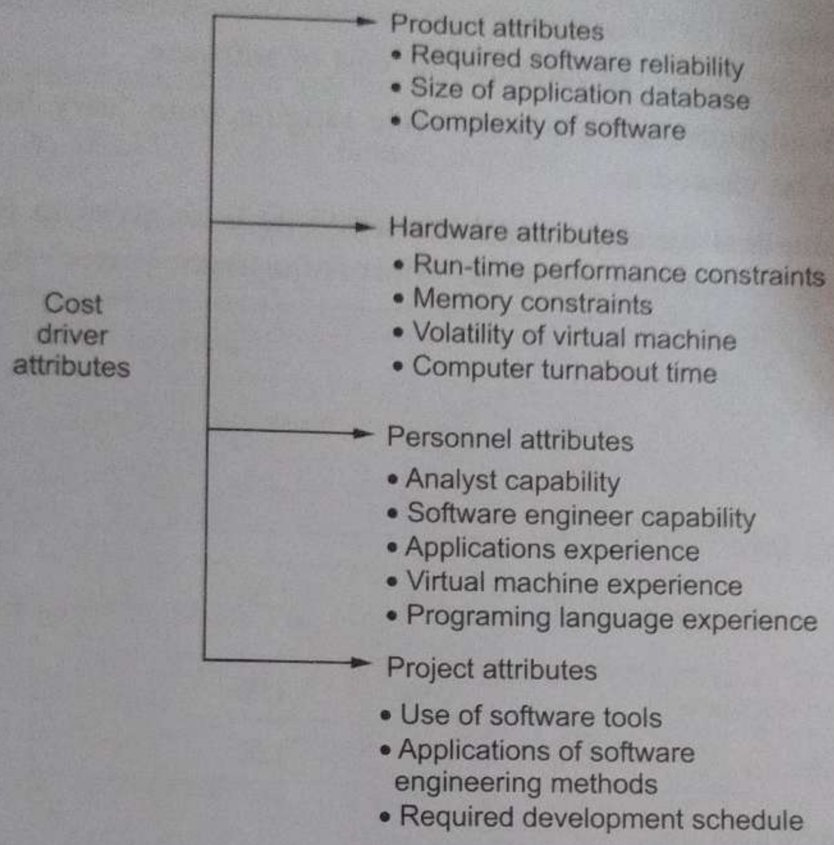
Basic COCOMO model is good for quick, early, rough order of magnitude estimates of software project.

Limitations of basic model

1. The accuracy of this model is limited because it does not consider certain factors for cost estimation of software. These factors are hardware constraints, personal quality, and experience, modern tachniques and tools.

2. The estimates of COCOMO model are within a factor of 1.3 only 29 % of the time and within the factor of 2 only 60 % of time.

Example


Cost driver attributes
Product attributes
• Required software reliability
• Size of application database
• Complexity of software
Hardware attributes
• Run-time performance constraints
• Memory constraints
• Volatility of virtual machine
• Computer turnabout time
Personnel attributes
• Analyst capability
• Software engineer capability
• Applications experience
• Virtual machine experience
• Programing language experience
Project attributes
• Use of software tools
• Applications of software engineering methods
• Required development schedule

Consider a software project using semi-detached mode with 30,000 lines of code. We will obtain estimation for this project as follows -

i) Effort estimation

$$E = a_b \, (\text{KLOC})^{b_b}$$

i.e. $E = 3.0\,(30)^{1.12}$ where lines of code = 30000 = 30 KLOC

$$E = 135 \text{ person-month}$$

ii) Duration estimation

$$D = C_b \, (E)^{d_b}$$

$$= 2.5\,(135)^{0.35}$$

$$D = 14 \text{ months}$$

iii) Persons estimation

$$P = E/D$$

$$= 135/14$$

P = 10 persons approximately

2) Intermediate Model

This is an extension of Basic COCOMO model. This estimation model makes use of set of "Cost driver attributes" to compute the cost of software.

Now these 15 attributes get a 6-point scale ranging from "very low" to "extra high". These ratings can be viewed as

The effort multipliers for each cost driver attribute is as given in following table. The product of all effort multipliers result in "Effort Adjustment Factor" (EAF).

Cost drivers	Ratings					
	Very low	Low	Nominal	High	Very high	Extra high
Product attributes						
Required software reliability	0.75	0.88	1.00	1.15	1.40	
Size of application database		0.94	1.00	1.08	1.16	
Complexity of software	0.70	0.85	1.00	1.15	1.30	1.65
Hardware attributes						
Run-time performance constraints			1.00	1.11	1.30	1.66
Memory constraints			1.00	1.06	1.21	1.56
Volatility of virtual machine		0.87	1.00	1.15	1.30	
Computer turnabout time		0.87	1.00	1.07	1.15	
Personnel attributes						
Analyst capability	1.46	1.19	1.00	0.86	0.71	
Software engineer capability	1.42	1.17	1.00	0.86	0.70	
Applications experience	1.29	1.13	1.00	0.91	0.82	
Virtual machine experience	1.21	1.10	1.00	0.90		
Programming language experience	1.14	1.07	1.00	0.95		
Project attributes						
Use of software tools	1.24	1.10	1.00	0.91	0.82	

Applications of software engineering methods	1.24	1.10	1.00	0.91	0.83	
Required development schedule	1.23	1.08	1.00	1.04	1.10	

Table 8.5.2

The formula for effort calculation can be -

$$E = a_i \, (KLOC)^{b_i} \cdot EAF \text{ person-months}$$

The values for a_i and b_i for various class of software projects are -

Software project	a_i	b_i
Organic	3.2	1.05
Semi-detached	3.0	1.12
Embedded	2.8	1.20

Table 8.5.3

The duration and person estimate is same as in basic COCOMO model. i.e.

$$D = c_b \, (E)^{d_b} \text{ months}$$

i.e. use values of c_b and d_b coefficients that are in Table 8.5.1

$$P = E/D \text{ persons}$$

Merits of Intermediate Model

1. This model can be applied to almost entire software product for easy and rough cost estimation during early stage.
2. It can also be applied at the software product component level for obtaining more accurate cost estimation.

Limitations of Intermediate Model

1. The estimation is within 20 % of actual 68 % of the time.
2. The effort multipliers are not dependent on phases.
3. A product with many components is difficult to estimate.

Example

Consider a project having 30,000 lines of code which is an **embedded** software with critical area hence realiability is high. The estimation can be

$$E = a_i \, (KLOC)^{b_i} \cdot EAF$$

As reliability is high, EAF = 1.15 (product attribute)

$$\left.\begin{array}{l} a_i = 2.8 \\ b_i = 1.20 \end{array}\right\} \text{ for embedded software}$$

$$\therefore \quad E = 2.8\,(30)^{1.20} * 1.15$$

$$= 191 \text{ person-month}$$

$$D = c_b\,(E)^{d_b} = 2.5\,(191)^{0.32}$$

$$= 13 \text{ months approximately}$$

$$P = E/D$$

$$= 191/13$$

$$P = 15 \text{ persons approximately}$$

3) Detailed COCOMO Model

The detailed model uses the same equations for estimation as the Intermediate Model. But detailed model can estimate the effort (E), duration (D) and persons (P) of each of development phases, subsystems, modules.

The experimentation with different development strategies is allowed in this model.

Four phases used in detailed COCOMO model are -

1. Requirements planning and product design (RPD)
2. Detailed design (DD)
3. Code and unit test (CUT)
4. Integrate and test (IT)

The effort multipliers for detailed COCOMO are

Exercise 8.5.1 *Using COCOMO, estimate time required for the following :*

1) *A semi-detached model of software project of 2000 lines.*
2) *An embedded model of software of 30,000 lines.*
3) *An organic model of software of one lakh lines.*
4) *An organic model of software of 10 lakh lines.*

Solution : To estimate time using basic model of COCOMO following formula can be used.

$$E = a_b(KLOC)^{b_b}$$

where E is the effort in person-month.

$$D = c_b(E)^{d_b}$$

where D is development time in chronological months.

$$P = E/D$$

where P is total number of persons involved in the project. The constants are

System	a_b	b_b	c_b	d_b
Organic system	2.4	1.05	2.5	0.38
Semidetached system	3.0	1.12	2.5	0.35
Embedded system	3.6	1.20	2.5	0.32

1) Given that, System = Semi detached

Lines of code = 2000 lines = 2 KLOC

$$\therefore \quad E = a_b(KLOC)^{b_b}$$

$$E = 3.0\,(2)^{1.15}$$

$$E = 6.65 \text{ person-month}$$

$$\therefore \quad D = c_b(E)^{d_b}$$

$$D = 4.8 \text{ months}$$

$$\therefore \quad P = E/D$$

$$P = 1.3 \approx 1 \text{ person}$$

Thus 1 person can handle this project within approximately 5 months.

2) Given that, System = Embedded

Lines of code = 30,000 lines = 30 KLOC

$$\therefore \quad E = a_b(KLOC)^{b_b}$$

$$= 3.6\,(30)^{1.20}$$

$$E \approx 213 \text{ person - month}$$

$$D = c_b(E)^{d_b}$$

$$= 2.5\,(213)^{0.32}$$

$$\therefore \quad D \approx 14 \text{ months}$$

$$P = E/D$$

$$\approx 213/14$$

$$\approx 15 \text{ persons.}$$

That means 15 persons can complete this project within approximately 14 months.

3) Given that, System = Organic

Lines of code = 1 lakh = 100 KLOC

$$\therefore \quad E = a_b(KLOC)^{b_b}$$

$$= 2.4\,(100)^{1.05}$$

$$E \approx 302 \text{ person-month}$$

$$D = c_b(E)^{d_b} = 2.5\,(302)^{0.38}$$

$$\therefore \quad \approx 21 \text{ months}$$

$$P = E/D$$

$$\approx 302/21$$

$$\approx 14 \text{ persons.}$$

That means this project can be completed within 21 months by 14 persons, approximately,

4) Given that, System = Organic

Lines of code = 10 lakh = 1000 KLOC

$$\therefore \quad E = a_b(KLOC)^{b_b} = 2.4\,(1000)^{1.05}$$

$$\approx 3390 \text{ person - month}$$

$$D = c_b(E)^{d_b}$$

$$\approx 2.5\,(3390)^{0.38}$$

$$\approx 55 \text{ months}$$

$$\therefore \quad P = E/D$$

$$\approx 3390/55$$

$$\approx 61 \text{ persons}$$

This project can be completed within 55 months by 61 people approximately.

Exercise 8.5.2 *For a project of 100,000 LOC embedded system, compose the efforts if : Highly capable programmers with very little experience in the programming language. Programmers of low quality but a lot of experience with programming language.*

Solution : Given that :

System = Embedded system $\therefore a_i = 2.8$ and $b_i = 1.20$

Line of code = 100 KLOC

The formula for effort calculation can be :

$$E = a_i(KLOC)^{b_i} \times EAF \text{ person-months}$$

Case 1 : Highly capable programmer. Hence EAF = 0.86. Very little experience in programming language. Hence EAF = 1.14.

Hence effort can be :

$$E = 2.8(100)^{1.20} \times 0.86 \times 1.14$$

$$E = 689.5 \approx 689 \text{ person-months}$$

$$\text{Duration} = c_b(E)^{d_b} \text{ months}$$

$$c_b = 2.5 \text{ , } d_b = 0.32$$

$$\text{Hence duration} = 2.5\,(689)^{0.32}$$

$$= 20.2 \approx 20 \text{ months}$$

$$\text{Number of persons} \quad P = E/D$$

$$= 689/20$$

$$\therefore \quad P = 34 \text{ persons}$$

Case 2 : Programmers of low quality . Hence EAF = 1.17.

A lot experience with programming language. ∴ EAF = 0.95

Hence total effort :

$$E = 2.8(100)^{1.20} \times 1.17 \times 0.95$$

$$= 781.1 \approx 781 \text{ person-months}$$

$$\text{The duration} = c_b\,(E)^{d_b}$$

$$= 2.5\,(781)^{0.32}$$

$$= 21 \text{ months}$$

$$\text{Person estimation } P = E/D = 781/21$$

$$= 37.1 \approx 37 \text{ persons.}$$

9.1 Risk Management

MU : May-11, Marks 10

Definition of risk : The risk denotes the uncertainty that may occur in the choices due to past actions and risk is something which causes heavy losses.

Definition of risk management : Risk management refers to the process of making decisions based on an evaluation of the factors that threats to the business.

Various activities that are carried out for risk management are -

1. Risk identification
2. Risk projection
3. Risk refinement
4. Risk mitigation, monitoring and management.

9.1.1 Software Risks

There are two characteristics of the risks

1. The risk may or may not happen. It shows the **uncertainty** of the risks.
2. When risks occur, unwanted consequences or **losses** will occur.

Different types of risk

1. Project risk

Project risks arise in the software development process then they basically affect budget, schedule, staffing, resources, and requirements. When project risks become severe then the total cost of project gets increased.

2. Technical risk

These risks affect quality and timeliness of the project. If technical risks become reality then potential design implementation, interface, verification and maintenance problems gets created. Technical risks occur when problem becomes harder to solve.

3. Business risk

When feasibility of software product is in suspect then business risks occur. Business risks can be further categorized as

i) Market risk - When a quality software product is built but if there is no customer for this product then it is called market risk (i.e. *no market for the product)*.

ii) Strategic risk - When a product is built and if it is not following the company's business policies then such a product brings strategic risks.

iii) Sales risk - When a product is built but how to sell is not clear then such a situation brings sales risk.

iv) Management risk - When senior management or the responsible staff leaves the organization then management risk occurs.

v) Budget risk - Losing the overall budget of the project is called budget risk.

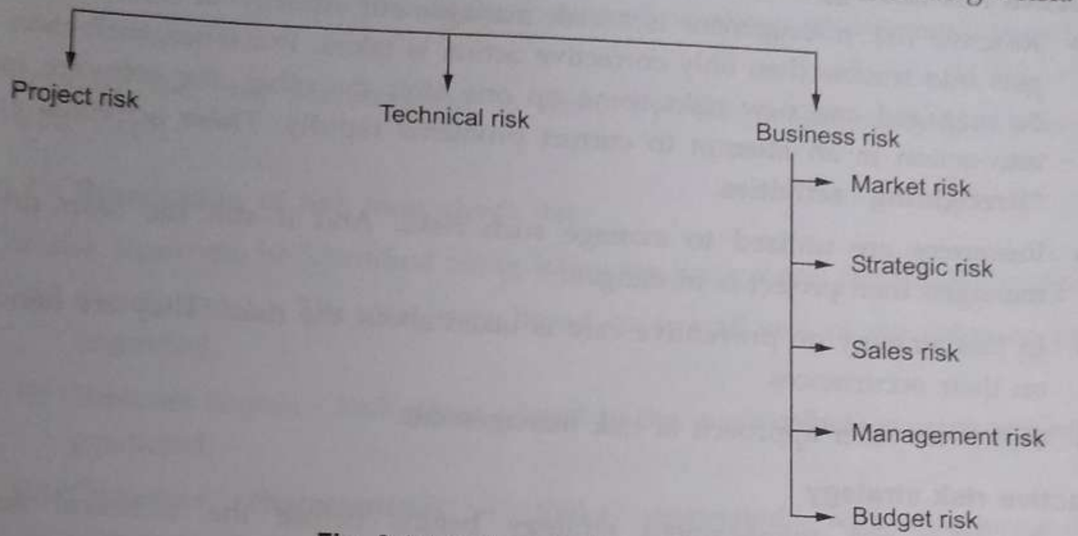


Fig. 9.1.1 Categorization of risk

Another categorization of risk proposed by **Charette** is -

Known risks are those risk that are identified after evaluating the project plan. These risks can also be identified from other sources such as environment in which the product gets developed, unrealistic dead lines, poor requirement specification and software scope. There are two types of known risks - *predictable* and *unpredictable* risks.

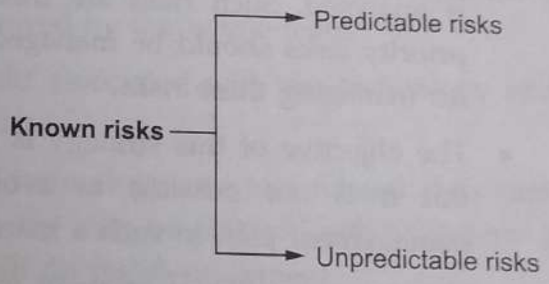


Fig. 9.1.2

Predictable risks are those risks that can be identified in advance based on past project experience. For example : Experienced and skilled staff leaving in between or improper communication with customer resulting in poor requirement specification.

Unpredictable risks are those risks that can not be guessed earlier.

For example certain changes in Government policies may affect the business project.

Review Question

1. *Explain risk analysis and management in detail* **MU : May-11, Marks 10**

9.2 Risk Strategies

Reactive and proactive risk strategies are the approaches used for managing the risks.

Reactive risk strategy

- Reactive risk management is a risk management strategy in which when project gets into trouble then only corrective action is taken. But when such risks can not be managed and new risks come up one after the other, the software team flies into action in an attempt to correct problems rapidly. These activities are called **"firefighting" activities**.
- Resources are utilized to manage such risks. And if still the risks do not get managed then project is in danger.
- In this strategy no preventive care is taken about the risks. They are handled only on their occurrences.
- This is an older approach of risk management.

Proactive risk strategy

- Proactive risk management strategy begins before the technical activity by considering the probable risk.
- In this strategy potential risks are identified first then their probability and impact is analyzed. Such risks are then specified according to their priorities (i.e. high priority risks should be managed first!). Finally the software team prepares a plan for managing these risks.
- The objective of this strategy is to avoid the risks (*prevention is better than cure!!!*). But it is not possible to avoid all the risks, hence team prepares the risk management plan in such a manner that risk controlling can done efficiently.
- This is an intelligent strategy for risk management and now a day it is used by most of the IT industries.

Review Question

1. *What are the types of risks ? Explain in brief.*

9.3 Risk Identification

Risk identification can be defined as the efforts taken to specify threats to the project plan. Risks identification can be done by identifying the known and predictable risks.

The risk identification is based on two approaches

1. Generic risk identification - It includes potential threat identification to software project.

2. Product-specific risk identification - It includes product specific threat identification by understanding people, technology and working environment in which the product gets built.

Normally the risk identification is done by the project manager who follows following steps -

Step 1 : Preparation of risk item check list

The risk items can be identified using following known and predictable components

i) Product size - The risk items based on overall size of the software product is identified.

ii) Business impact - Risk items related to the marketplace or management can be predicted.

iii) Customer characteristics - Risks associated with customer-developer communication can be identified.

iv) Process definition - Risks that get raised with the definition of software process. This category exposes important risks items because whichever is the process definition made, is then followed by the whole team.

v) Development environment - The risks associated with the technology and tool being used for developing the product.

vi) Staff size and experience - Once the technology and tool related risks items are identified it is essential to identify the risk associated with sufficient highly experienced and skilled staff who will do the development.

vii) Technology to be built - complexity of the system should be understood and related risk items needs to be identified.

After preparing a risk item checklist a questionnaire is prepared. These set of questions should be answered and based on these answers the impact or seriousness of particular risk item can be judged.

Step 2 : Creating risk components and drivers list.

The set of risk components and drivers list is prepared along with their probability of occurrence. Then their impact on the project can be analysed.

Let us understand which are the risk components and drivers.

9.3.1 Risk Components and Drivers

U.S. Air force has written a guideline for risk identification which is based on identification of risk component and risks drivers. It has suggested following types of risk components -

1. *Performance risk* - It is the degree of uncertainty that the product will satisfy the requirements
2. *Cost risk* - It is the degree of uncertainty that the project will maintain the budget.
3. *Support risk* - It is the degree of uncertainty that the software project being developed will be easy to correct, modify or adapt.

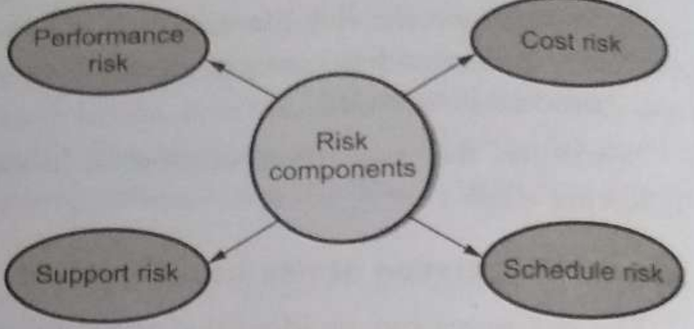


Fig. 9.3.1 Components of risk

4. *Schedule risk* - It is the degree of uncertainty that the software project will maintain the schedule and the project will be delivered in time.

Associated with these components are the risk drivers that are used to analyse the impact of risk. These four risk drivers are listed below

For the risk impact assessment a table is built in which impact of each risk driver on each software component can be specified.

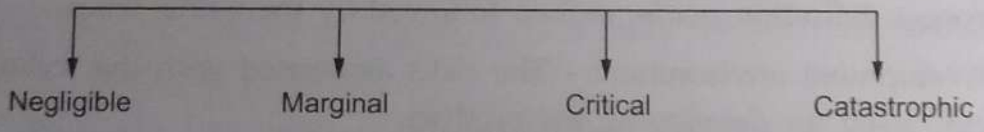


Fig. 9.3.2

9.3.2 How to Asses Overall Project Risk ?

The best approach is to prepare a set of questions that can be answered by project managers in order to asses the overall project risks. These **questions** can be

1. Will the project get proper support by the customer manager ?
2. Are the end-users committed to the software that has been produced ?
3. Is there a clear understanding of requirements ?
4. Is there an active involvement of the customer in requirement definition ?
5. Is that the expectations set for the product are realistic ?
6. Is project scope stable ?
7. Are there team members with required skills ?
8. Are project requirements stable ?
9. Does the technology used for the software is known to the developers ?
10. Is the size of team sufficient to develop the required product ?
11. Is that all the customers know the importance of the product/ requirements of the system to be built ?

Thus the number of negative answers to these questions represents the severity of the impact of the risk on overall project.

9.4 Risk Projection

The risk projection is also called **risk estimation**.

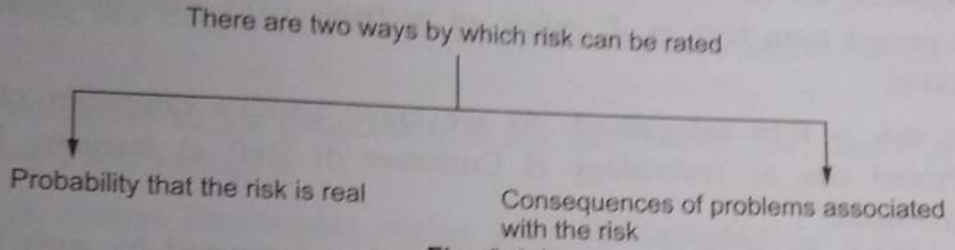

There are two ways by which risk can be rated
Probability that the risk is real
Consequences of problems associated with the risk

Fig. 9.4.1

The project planner, technical staff, project manager performs following steps to perform following steps for risk projection -

- Establish a scale that indicates the probability of risk being real.
- Enlist the consequences of the risk.
- Estimate the impact of the risk on the project and product.
- Maintain the overall accuracy of the risk projection in order to have clear understanding of the software that is to be built.

These steps help to prioritize the risks. Once the risks are prioritized then it becomes easy to allocate the resources for handling them.

9.4.1 Building Risk Table

1. Building the risk table is the simplest and most commonly used technique adopted by project managers in order to project the risks. The sample risk table is as given below -

Risk table				
Risk	**Category**	**Probability**	**Impact**	**RMMM**
Is the skilled staff available	Staff	50 %	Catastrophic	
Is that the team size sufficient	Staff	62 %	Critical	
Have the staff received sufficient training	Staff	25 %	Marginal	
Will technology meet the expectations	Technology	30 %	Critical	
Is the software management tool available	Environment	40 %	Negligible	

How much amount of reused software is required?	Project size	60 %	Marginal	
Will customer change the requirement ?	Customer	20 %	Critical	

While building the risk table

- The project team first of all enlists all probable risks with the help of risk item checklist.
- Each risk is then categorized. As we know various categories of risk can be a) Project size b) Technology c) Customer d) Staff e) Business f) Developing environment.
- Probability of occurrence of each risk is then estimated by each team member individually.
- Then impact of each risk is assessed. While calculating the impact of each risk, each using the cost drivers each component of risk (*performance, cost, support, and schedule*) is assessed and it then averaged to quote the overall impact of particular risk.

2. After building this table it is then sorted by probability and impact. The high probability and high impact risks will be at the top of the table. And low probability and low impact risk will be at the bottom of the table. This arrangement of the table is called **first-order prioritization**.
3. Then the project manager goes through this first-order prioritized risk table and draws a horizontal line at some point in the table. This line is called **cut off line**. The risks table above the cut off line is now considered for further risk analysis.
4. The risk table below the cut off line is again sorted and a **second-order prioritization** is applied on this table.
5. The risk table above the cut-off line is having the risks with high probability and high impact and such risks should occupy the significant amount of management time.
6. All the risks that lie above the cut off line should be managed. Using Risk mitigation, monitoring and management plan the last column of the risk table is filled up.

9.4.2 Assessing Risk Impact

While assessing the risks impact three factors are considered

- Nature of risk
- Scope of the risk
- Timing at which risk occurs.

Nature of risk denotes the type or kind of risk. For example if software requirement is poorly understood, the software processes gets poorly designed and ultimately it will create a problem in unit testing. **Scope** of the risk means severity of the risk. And **timing** of risk means determining at which phase of software development life cycle the risk will occur and how long it will persist.

U.S. Air Force has suggested following steps in order to determine the impact of risk -

1. The probability of all the components of risk (*performance, cost, support and schedule*) is calculated and averaged.
2. Using risk drivers (*catastrophic, critical, marginal, negligible*) the impact of risk on each components is determined.
3. Build the risk table and analyse the high impact, high probability risks.

Risk exposure

The risk exposure can be calculated by following formula

Risk Exposure = Probability of occurrence of risk × Cost

For example : Consider a software project with 77 percent of risk probability in which 15 components were developed from the scratch. Each component have on an average 500 LOC and each LOC have an average cost of \$10. Then the risk exposure can be calculated as ,

First of all we will compute

$$\begin{aligned} cost &= \text{Number of components*LOC*cost of each LOC} \\ &= 15*500*10=\$75000 \\ \text{Then Risk Exposure} &= \text{Probability of occurrence of risk} \times \text{Cost} \\ &= 77\ /\ 100\ *75000 \\ &= \$57750 \end{aligned}$$

Thus risk exposure for each risk from risk table is calculated. The total risk exposure of all risks helps in determining the final cost of the project.

Review Question

1. *How risk projection is carried out risk table ?*

9.5 RMMM

MU : Dec.-10, Marks 10

RMMM stands for **risk mitigation, monitoring** and **management**. There are three issues in strategy for handling the risk is

1. Risk avoidance 2. Risk monitoring 3. Risk management.

Risk mitigation

Risk mitigation means preventing the risks to occur (risk avoidance). Following are the steps to be taken for mitigating the risks.

1. Communicate with the concerned staff to find of probable risk.
2. Find out and eliminate all those causes that can create risk before the project starts.
3. Develop a policy in an organization which will help to continue the project even though some staff leaves the organization.
4. Everybody in the project team should be acquainted with the current development activity.
5. Maintain the corresponding documents in timely manner. This documentation should be strictly as per the standards set by the organization.
6. Conduct timely reviews in order to speed up the work.
7. For conducting every critical activity during software development, provide the additional staff if required.

Risk monitoring

In risk monitoring process following things must be monitored by the project manager,

1. The approach or the behaviour of the team members as pressure of project varies.
2. The degree in which the team performs with the spirit of "team-work".
3. The type of co-operation among the team members.
4. The types of problems that are occurring.
5. Availability of jobs within and outside the organization.

The project manager should monitor certain mitigation steps. For example.

If the current development activity is monitored continuously then everybody in the team will get acquainted with current development activity.

The **objective** of risk monitoring is

1. To check whether the predicted risks really occur or not.
2. To ensure the steps defined to avoid the risk are applied properly or not.
3. To gather the information which can be useful for analyzing the risk.

Risk management

Project manager performs this task when risk becomes a reality. If project manager is successful in applying the project mitigation effectively then it becomes very much easy to manage the risks.

For example, consider a scenario that many people are leaving the organization then if sufficient additional staff is available, if current development activity is known to everybody in the team, if latest and systematic documentation is available then any 'new comer' can easily understand current development activity. This will ultimately help in continuing the work without any interval.

Exercise 9.5.1 *What are the risks associated with delayed projects ? How do project managers manage such risk ?*

MU : Dec.-10, Marks 10

Solution : With delayed project the project risk occurs. Following are the problems that may get occur due to delayed projects -

1. The cost of overall project will get increased.
2. The resources need to be engaged for a long time
3. The technology may get changed over a period. And project may no longer be compatible with the current trends and technology.
4. The requirements might get changed by the period of completion of the period.
5. Due to delayed project work the development staff may loose interest in the project and hence there will be dilution of effort.

The project managers assist the project team in developing the strategy for dealing with delayed project risk. An effective strategy considers three important issues

1. Risk avoidance
2. Risk monitoring
3. Risk monitoring and Contingency planning

The first and foremost activity is to avoid the risk of getting the project a delayed one. This is called risk mitigation. To mitigate this risk project management must develop a strategy for handling the delay in projects. The possible steps can be

1. Meet the current staff and determine the causes of delay in the project.
2. Communicate with the customer to convince him/her about elimination/reduction of unrealistic requirements.
3. Mitigate those causes that are under the control of project manager.
4. Organize the project team in such a way that development activity is widely dispersed.

5. Develop a technique to ensure the continuity when people leave the ongoing project in-between.
6. Conduct peer reviews of all work periodically.
7. Assign backup staff member for critical activities.
8. Define documentation standards and develop the documents in timely manner. The documents will be useful for all the team members during the development process.

As project proceeds risk monitoring activities start and then it could be identified whether or not the risk is becoming more or less likely. During this phase following factors can be monitored -

1. The general attitude of team members in high project pressures.
2. Interpersonal relationship among the team members
3. The degree to which the team has jelled.

If the risk mitigation efforts are failed and if the risk of delayed project becomes the reality then if backup is available, information is documented and knowledge is dispersed across the team then project manager can add-up the new-comers to speed up the work. Some members then can transfer the knowledge to the new-comers and amount of work can be shared with them.

By following above discussed strategies, the delay in the project can be handled to some extent.

Review Questions

1. *What is risk mitigation, monitoring and management (RMMM) ? Write a note on it.*
2. *Write short note on : RMMM.*

9.6 RMMM Plan

MU : Dec.-09,11, May-12, Marks 10

The RMMM plan is a document in which all the risk analysis activities are described. Sometimes project manager includes this document as a part of overall project plan. Sometimes specific RMMM plan is not created, however each risk can be described individually using risk information sheet. Typical template for RMMM plan or Risk information sheet can be,

Risk information sheet
Project name *<enter name of the project for which risks can be identified>*

Risk id <#>	**Date** *<date at which risk is identified >*	**Probability** *<risk probability>*	**Impact** *<low/medium/high>*
Origin *<the person who has identified the risk>*		**Assigned to** *<who is responsible for mitigating the risk>*	
Description *<Description of risk identified>*			
Refinement/Context *<associated information for risk refinement>*			
Mitigation/Monitoring *<enter the mitigation/monitoring steps taken>*			
Trigger/Contingency plan *<if risk mitigation fails then the plan for handling the risk>*			
Status *<Running status that provides a history of what is being done for the risk and changes in the risk. Include the date the status entry was made>*			
Approval <name and signature of person approving closure>.		**Closing date** *<date>*	

The risk information sheet can be maintained by database systems. After documenting the risks using either RMMM plan or Risk information sheet the risk mitigation, monitoring and analysis activities are stopped.

Review Question