EDU

EAST DELTA
UNIVERSITY

CSE 411
Software Engineering and System Analysis and Design

Topic 4: Software Project Management, Planning & Risk Management

Project Management

A project is a group of tasks that need to complete to reach a clear result.

Projects usually described and approved by a project manager or team executive.

Software project management is an art and discipline of planning and supervising software projects.

In software Project Management, the client and the developers need to know the length, period and cost of the project.

Prerequisite of software project management are:

- Time
- Cost
- Quality

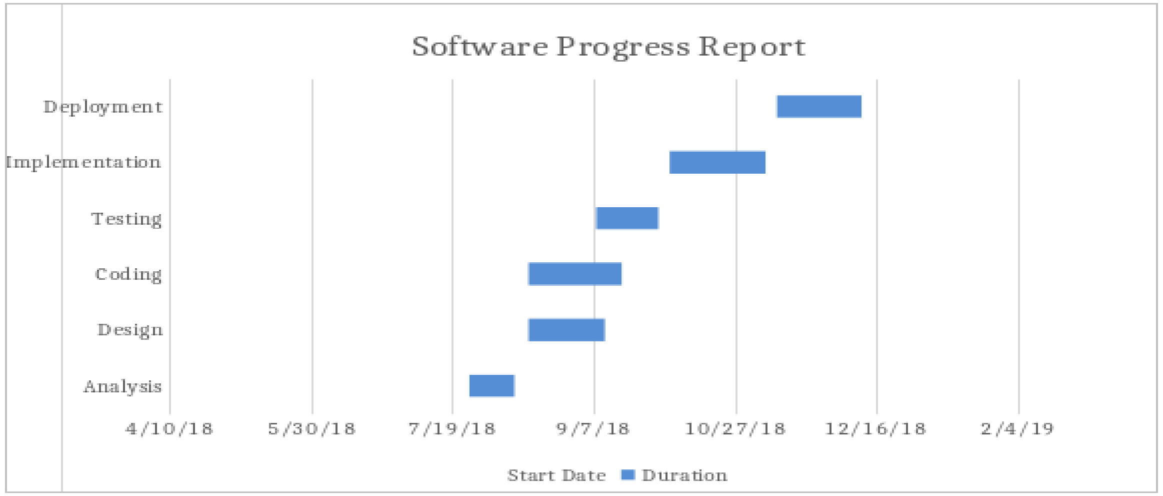
A project manager is a character who has the overall responsibility for the planning, design, execution, monitoring, controlling and closure of a project.

Responsibilities of a Project Manager:

- Managing risks and issues.
- Create the project team and assigns tasks to several team members.
- Activity planning and sequencing.
- Monitoring and reporting progress.
- Modifies the project plan to deal with the situation.

Gantt chart

- Gantt Chart first developed by Henry Gantt in 1917.
- Gantt chart usually utilized in project management, and it is one of the most popular and helpful ways of showing activities displayed against time.



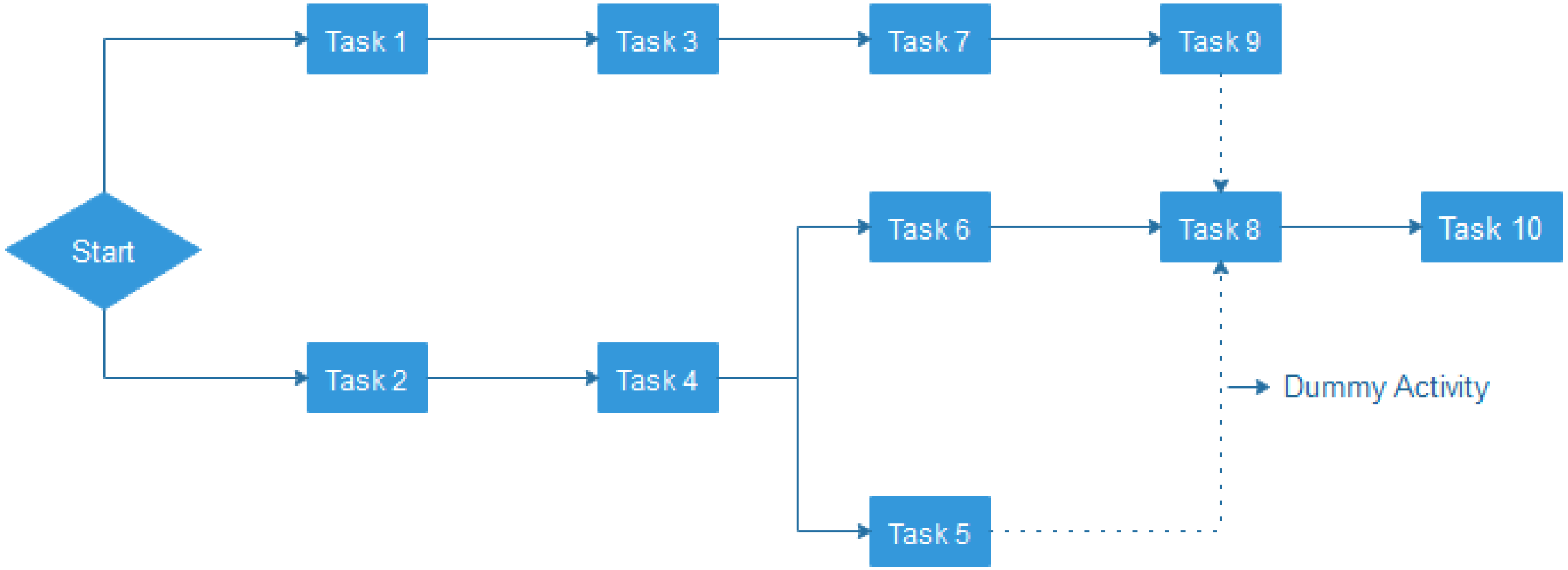
Gantt chart in Excel

https://www.youtube.com/watch?v=qbkL2nlI02Q

PERT chart

- PERT is an acronym of Programme Evaluation Review Technique.
- In the 1950s, it is developed by the U.S. Navy to handle the Polaris submarine missile programme.
- In Project Management, PERT chart represented as a network diagram concerning the number of nodes, which represents events.

PERT chart



- The direction of the lines indicates the sequence of the task.

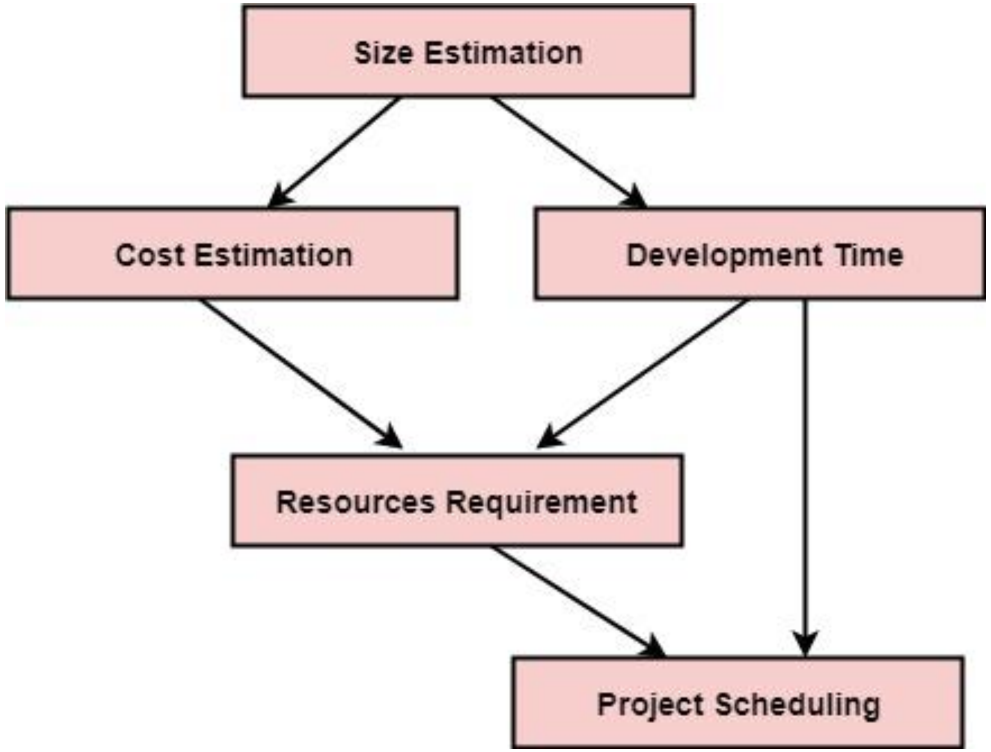
- In the above example, tasks between "Task 1 to Task 9" must complete, and these are known as a dependent or serial task.

- Between Task 4 and 5, and Task 4 and 6, nodes are not depended and can undertake simultaneously. These are known as Parallel or concurrent tasks.

Project Planning

Software manager is responsible for planning and scheduling project development.

Software Project planning starts before technical work start. The various steps of planning activities are:



Software Cost Estimation

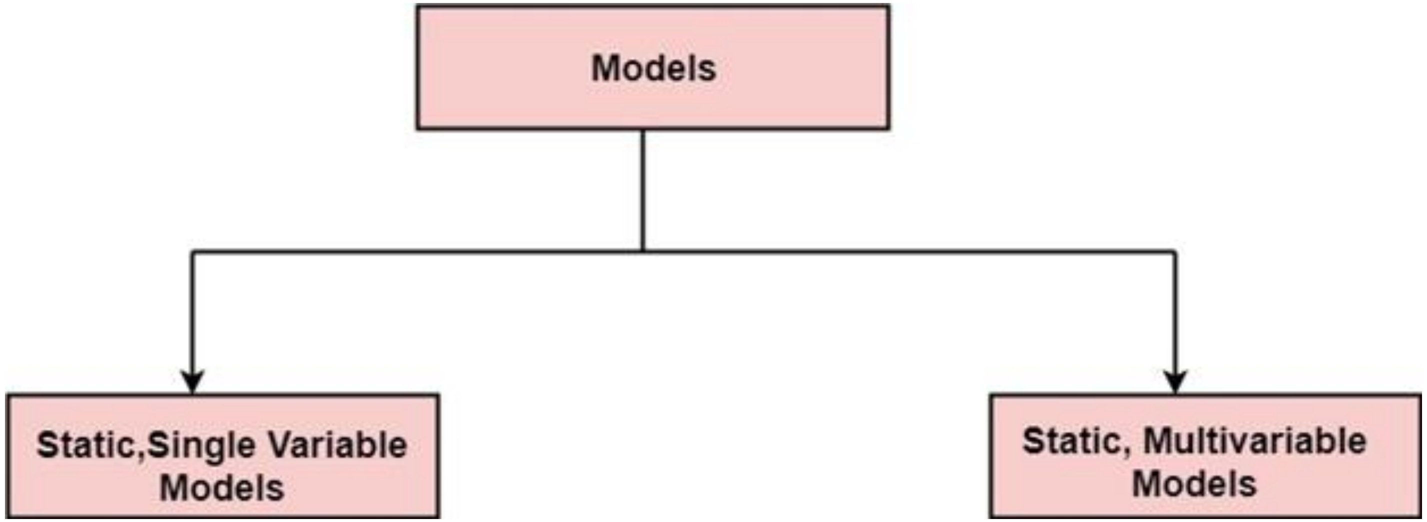
- For any new software project, it is necessary to know how much it will cost to develop and how much development time will it take.

- During the planning stage, one needs to choose how many engineers are required for the project and to develop a schedule.

- In monitoring the project's progress, one needs to access whether the project is progressing according to the procedure and takes corrective action, if necessary.

Cost Estimation Models

- A model may be static or dynamic.
- In a static model, a single variable is taken as a key element for calculating cost and time.
- In a dynamic model, all variable are interdependent, and there is no basic variable.



Static, Single Variable Models: When a model makes use of single variables to calculate desired values such as cost, time, efforts, etc. is said to be a single variable model. The most common equation is:

$$C=aL^{b}$$

Where C = Costs ; L= size ; a and b are constants

The Software Engineering Laboratory established a model called SEL model, for estimating its software production. This model is an example of the static, single variable model.

Where,

$E=1.4L^{0.93}$

$DOC=30.4L^{0.90}$

$D=4.6L^{0.26}$

E= Efforts (Person Per Month),

DOC=Documentation (Number of Pages),

D = Duration (D, in months),

L = Number of Lines per code

Static, Multivariable Models: These models are based on method (1), they depend on several variables describing various aspects of the software development environment. In some model, several variables are needed to describe the software development process, and selected equation combined these variables to give the estimate of time & cost. These models are called multivariable models.

WALSTON and FELIX develop the models at IBM provide the following equation gives a relationship between lines of source code and effort:

$$\mathbf{E=5.2L^{0.91}}$$

In the same manner duration of development is given by

$$\mathbf{D=4.1L^{0.36}}$$

Example:

Compare the Walston-Felix Model with the SEL model on a software development expected to involve 8 person-years of effort.

- *Calculate the number of lines of source code that can be produced.*
- *Calculate the duration of the development.*
- *Calculate the productivity in LOC/PY*
- *Calculate the average manning*

The amount of manpower involved :

8PY = 96 persons-months

(a)Number of lines of source code can be obtained by reversing equation to g

$$L = \left(\frac{E}{a}\right)1/b$$

Then

L (SEL) = (96/1.4)1⁄0.93=94264 LOC

L (W-F) = (96/5.2)1⁄0.91=24632 LOC

(b)Duration in months can be calculated by means of equation

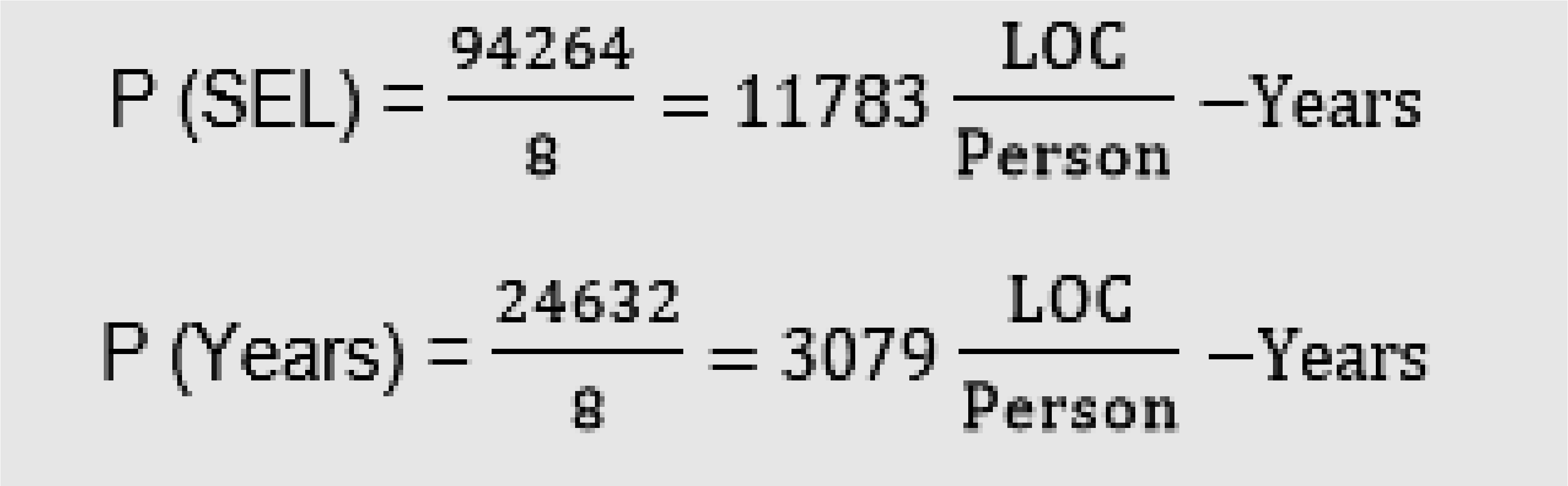
$$\mathbf{D\ (SEL) = 4.6\ (L)^{0.26}}$$

$$\mathbf{= 4.6\ (94.264)^{0.26} = 15\ months}$$

$$\mathbf{D\ (W\text{-}F) = 4.1\ L^{0.36}}$$

$$\mathbf{= 4.1\ (24.632)^{0.36} = 13\ months}$$

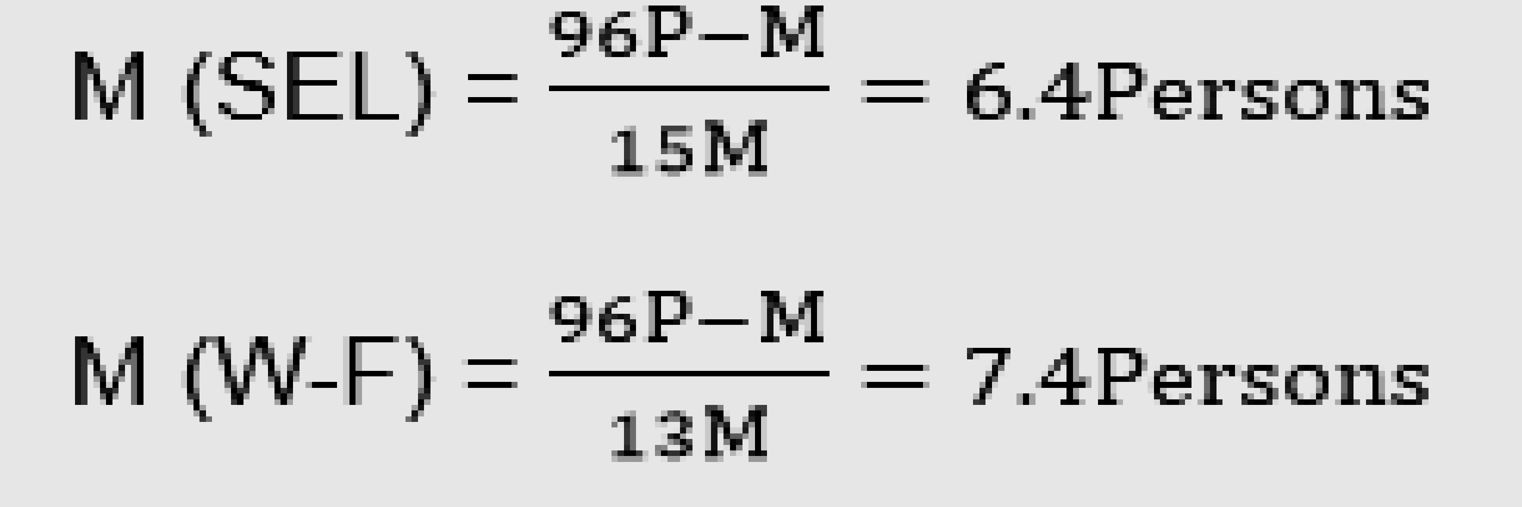
(c)Productivity is the lines of code produced per persons/month (year)



$$P\,(\text{SEL}) = \frac{94264}{8} = 11783\ \frac{\text{LOC}}{\text{Person}}\ -\text{Years}$$

$$P\,(\text{Years}) = \frac{24632}{8} = 3079\ \frac{\text{LOC}}{\text{Person}}\ -\text{Years}$$

(d)Average manning is the average number of persons required per month in the project



$$M\ (SEL) = \frac{96P-M}{15M} = 6.4Persons$$

$$M\ (W\text{-}F) = \frac{96P-M}{13M} = 7.4Persons$$

✓

COCOMO Model

- Boehm proposed COCOMO (Constructive Cost Estimation Model) in 1981.

- COCOMO is one of the most generally used software estimation models in the world.

- COCOMO predicts the efforts and schedule of a software product based on the size of the software.

The necessary steps in this model are:

- Get an initial estimate of the development effort from evaluation of thousands of delivered lines of source code (KDLOC).
- Determine a set of 15 multiplying factors from various attributes of the project.
- Calculate the effort estimate by multiplying the initial estimate with all the multiplying factors i.e., multiply the values in step1 and step2.

To determine the initial effort Ei in person-months the equation used is of the type is shown below

$$\mathbf{E_i=a*(KDLOC)^b}$$

The value of the constant a and b are depends on the project type.

In COCOMO, projects are categorized into three types:

- Organic
- Semidetached
- Embedded

✓ **1.Organic:** A development project can be treated of the organic type, if the project deals with developing a well-understood application program, the size of the development team is reasonably small, and the team members are experienced in developing similar methods of projects. Examples of this type of projects are simple business systems, simple inventory management systems, and data processing systems.

✓ **2. Semidetached:** A development project can be treated with semidetached type if the development consists of a mixture of experienced and inexperienced staff. Team members may have finite experience in related systems but may be unfamiliar with some aspects of the order being developed. Example of Semidetached system includes developing a new operating system (OS), a Database Management System (DBMS), and complex inventory management system.

✓ **3. Embedded:** A development project is treated to be of an embedded type, if the software being developed is strongly coupled to complex hardware, or if the stringent regulations on the operational method exist. For Example: ATM, Air Traffic control.

According to Boehm, software cost estimation should be done through three stages:

- Basic Model
- Intermediate Model
- Detailed Model

Basic COCOMO Model:

The basic COCOMO model provide an accurate size of the project parameters. The following expressions give the basic COCOMO estimation model:

$$\textbf{Effort} = a_1 * (\text{KLOC})^{a2}\ \text{PM}$$

$$\textbf{Tdev} = b_1 * (\text{efforts})^{b2}\ \text{Months}$$

Where -

- KLOC - size of the software product indicate in **Kilo Lines of Code**,
- a_1, a_2, b_1, b_2 are constants for each group of software products,
- Tdev is the estimated time to develop the software in months,
- Effort is the total effort required to develop the software product, expressed in person months (PMs).

Estimation of development effort

For the three classes of software products, the formulas for estimating the effort based on the code size are shown below:

Organic: Effort = 2.4(KLOC) $^{1.05}$ PM

Semi-detached: Effort = 3.0(KLOC) $^{1.12}$ PM

Embedded: Effort = 3.6(KLOC) $^{1.20}$ PM

Estimation of development time

For the three classes of software products, the formulas for estimating the development time based on the effort are given below:

Organic: Tdev = 2.5(Effort) $^{0.38}$ Months

Semi-detached: Tdev = 2.5(Effort) $^{0.35}$ Months

Embedded: Tdev = 2.5(Effort) $^{0.32}$ Months

Example1: Suppose a project was estimated to be 400 KLOC. Calculate the effort and development time for each of the three model i.e., organic, semi-detached & embedded.

Solution:

The basic COCOMO equation takes the form:

$\text{Effort}=a_1*(\text{KLOC})^{a2}$ PM

$\text{Tdev}=b_1*(\text{efforts})^{b2}$ Months

Estimated Size of project= 400 KLOC

(i)Organic Mode

$E = 2.4 * (400)^{1.05} = 1295.31$ PM

$D = 2.5 * (1295.31)^{0.38}=38.07$ Months

(ii)Semidetached Mode

$E = 3.0 * (400)^{1.12}=2462.79$ PM

$D = 2.5 * (2462.79)^{0.35}=38.45$ Months

(iii) Embedded Mode

$E = 3.6 * (400)^{1.20} = 4772.81$ PM

$D = 2.5 * (4772.8)^{0.32} = 38$ Months

Example2: A project size of 200 KLOC is to be developed. Software development team has average experience on similar type of projects. The project schedule is not very tight. Calculate the Effort, development time, average staff size, and productivity of the project.

Solution:

The semidetached mode is the most appropriate mode, keeping in view the size, schedule and experience of development time.

Hence $E=3.0(200)^{1.12}=1133.12PM$

$D=2.5(1133.12)^{0.35}=29.3Months$

$$\text{Average Staff Size (SS)} = \frac{E}{D}\ \text{Persons}$$

$$= \frac{1133.12}{29.3} = 38.67\ \text{Persons}$$

$$\text{Productivity} = \frac{KLOC}{E} = \frac{200}{1133.12} = 0.1765\ KLOC/PM$$

COCOMO Model

P = 176 LOC/PM

RISK MANAGEMENT

Risk Management is the system of identifying addressing and eliminating these problems before they can damage the project.

There are three main classifications of risks which can affect a software project:

- Project risks
- Technical risks
- Business risks

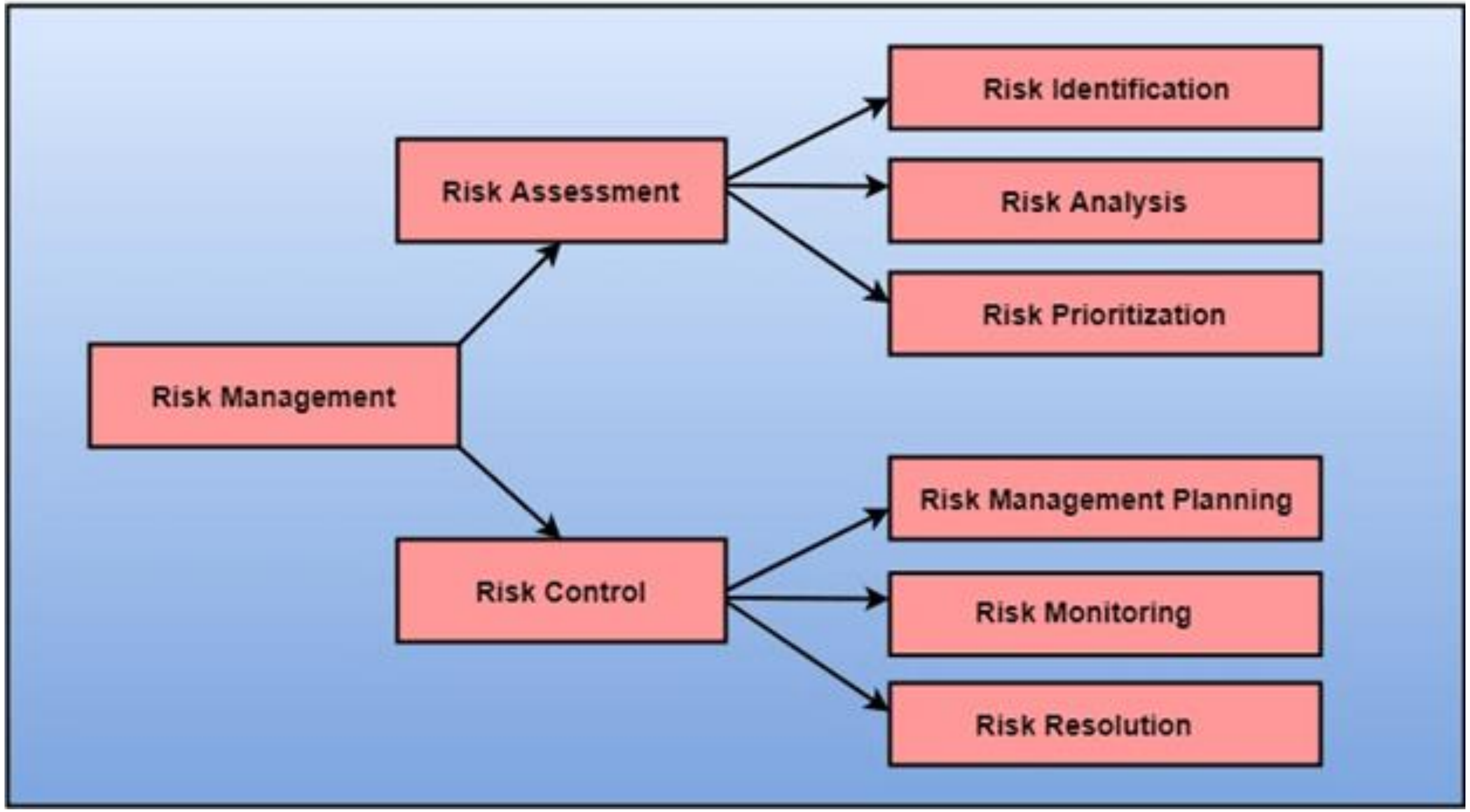
1. **Project risks:** Project risks concern differ forms of budgetary, schedule, personnel, resource, and customer-related problems. A vital project risk is schedule slippage. Since the software is intangible, it is very tough to monitor and control a software project. It is very tough to control something which cannot be identified. For any manufacturing program, such as the manufacturing of cars, the plan executive can recognize the product taking shape.

2. **Technical risks:** Technical risks concern potential method, implementation, interfacing, testing, and maintenance issue. It also consists of an ambiguous specification, incomplete specification, changing specification, technical uncertainty, and technical obsolescence. Most technical risks appear due to the development team's insufficient knowledge about the project.

3. **Business risks:** This type of risks contain risks of building an excellent product that no one need, losing budgetary or personnel commitments, etc.

Risk management consists of three main activities

Risk Management Activities



Risk Management

Risk Assessment

Risk Identification

Risk Analysis

Risk Prioritization

Risk Control

Risk Management Planning

Risk Monitoring

Risk Resolution

The objective of risk assessment is to division the risks in the condition of their loss, causing potential.

Risk Identification: The project organizer needs to anticipate the risk in the project as early as possible so that the impact of risk can be reduced by making effective risk management planning.

There are different types of risks which can affect a software project:

Technology risks: Risks with the technologies that are used to develop the system.

People risks: Risks with the person in the development team.

Organizational risks: Risks from the organizational environment where the software is being developed.

Tools risks: Risks from the software tools and other support software used to create the system.

Requirement risks: Risks from the changes to the customer requirement and the process of managing the requirements change.

Estimation risks: Risks from the management estimates of the resources required to build the system

Risk Analysis: During the risk analysis process, you have to consider every identified risk and make a perception of the probability and seriousness of that risk.

There is no simple way to do this. You have to rely on your perception and experience

Risk Control is the process of managing risks to achieve desired outcomes.

There are three main methods to plan for risk management:

Avoid the risk: This may take several ways such as discussing with the client to change the requirements to decrease the scope of the work, giving incentives to the engineers to avoid the risk of human resources turnover, etc.

Transfer the risk: This method involves getting the risky element developed by a third party, buying insurance cover, etc.

Risk reduction: This means planning method to include the loss due to risk. For instance, if there is a risk that some key personnel might leave, new recruitment can be planned.

End of Topic 4