19CS7602 - SOFTWARE PROJECT MANAGEMENT

Unit – I

2- marks

1. Analyze some problems with software projects.

Manager's point of view: Poor estimates and plans.

- Lack of quality standards and measures.
- Lack of techniques to make progress visible.
- Lack of guidance about organizational Decisions.
- Poor role definition. 6.Incorrect success criteria

Student's point of view:

- Inadequate specification of work.
- Lack of knowledge of application area.
- Lack of standards.
- Narrow scope of technical expertise

2. How can you analyse the decision tree in risk evaluation?

For a project manager the Bayesion decision tree analysis is used to mitigate the risk and cost of the decision. It is useful when analyzing a process that is a combination of many decisions as the user is able to calculate the effect of each decision. The decision tree is a model with three types of stages:

- ● A decision node (square)
- ● An event node (circle)
- ● A cost/consequence node

3. Write down the process of Net Present Value for computing a project.

The calculation of net present value is a project evaluation technique that takes into account the profitability of a project and the timing of cash flows that are produced. The NPV for a project is obtained by discounting each cash flow and summing the discounted values.

4. Write down the steps in cost benefit analysis.

Cost –benefit analysis consists of two steps

1. Identifying and estimating all of the costs and benefits of carrying out the project and operating the delivered application. It includes development cost of system, Operating cost of system, Benefits obtained by system.
2. Expressing these costs and benefits in common units.

5. What are the characteristics that make software projects different from other projects?

- ✓ Invisibility - When a physical artifact is being constructed the progress being made can actually be seen. With Software, progress is not immediately visible. Complexity - software products contain more complexity than other engineered artifacts.
- ✓ Conformity - The 'traditional' engineer is usually working with physical. These physical systems can have some complexity, but are governed by physical laws that are consistent. Software developers have to conform to the requirements of human clients. It is not just that individual can be inconsistent.
- ✓ Flexibility - The ease with which software can be changed is usually seen as one of its strengths.

6. List out the main activities of the risk paradigm.

- ❖ Identify
- ❖ Analyze
- ❖ Plan
- ❖ Track
- ❖ Control
- ❖ Communicate

7. Mention the characteristics of Software Projects.

- ❖ Non-routine tasks are involved
- ❖ Planning is required

- ❖ Specific objectives are to be met or a specific product is to be created
- ❖ The project has a predetermined time span
- ❖ Work is carried out for someone other than yourself
- ❖ Work involves several specialism
- ❖ People are formed into a temporary work group to carry out the task
- ❖ Work is carried out in several phases
- ❖ The resources that are available for use on the project are constrained
- ❖ The project is large or complex

8. Illustrate few problems associated with Software Project.

- ❖ People-related problems
- ❖ Process-related problems
- ❖ Product-related problems
- ❖ Technology-related problems

9. Write down the Phases of Project Life Cycle

- ❖ Initiation Phase
- ❖ Project Definition Phase
- ❖ Feasibility Study
- ❖ Project
- ❖ Project Conclusion

10. Which Factor decides the success of a project?

- ❖ The resulting information system is acceptable to the customer.
- ❖ The system was delivered “on time.”
- ❖ The system was delivered “within budget.”
- ❖ The system development process had a minimal impact on ongoing business operations.

Big Questions

1. Paul Duggan is the manager of a software development section. On Tuesday at 10.00 a.m. he and his fellow section heads have a meeting with their group manager about the staffing requirements for the coming year. Paul has already drafted a document 'bidding' for staff. This is based on the work planned for his section for the next year. The document is discussed at the meeting. At 2.00 p.m. Paul has a meeting with his senior staff about an important project his section is undertaking. One of the programming staff has just had a road accident and will be in hospital for some time. It is decided that the project can be kept on schedule by transferring another team member from less urgent work to this project. A temporary replacement is to be brought in to do the less urgent work but this may take a week or so to arrange. Paul has to phone both the human resources manager about getting a replacement and the user for whom the less urgent work is being done, explaining why it is likely to be delayed' Identify which of the eight management responsibilities listed above Paul was responding to at different points during his day.

Project

- ✓ A project is defined as a “temporary endeavor with a beginning and an end and it must be used to create a unique product, service or result”.
- ✓ Further, it is progressively elaborated. What this definition of a project means is that projects are those activities that cannot go on indefinitely and must have a defined purpose.
- ✓ For instance, if your project is less than three months old and has fewer than 20 people working on it, you may not be working in what is called a project according to the definition of the term.
- ✓ It has to be remembered that the term temporary does not apply to the result or service that is generated by the project.
- ✓ The project may be finite but not the result.

Examples

- ✓ Construction of Chennai Airport
- ✓ Computerising Apollo Hospital
- ✓ Conducting unit test

Project Characteristics

A project is not normal day to day activity undertaken by organization rather it is specific, non-routine activity of varying time frame and impact viability of the business in the long run. A typical project has following characteristics:

- ✓ Timeline : A project has a definite timeline with measurable starting and end point.
- ✓ Resources : A project has limited resource of capital and manpower.
- ✓ Tools : Special type of tools and techniques are used for project management (Gantt Charts, etc.)
- ✓ Team : Project management requires diverse team stretching across departments and functions.

Project Life Cycle

A typical project is divided into following phases. Each phase of the project has its own importance and impact on overall success of the project.

- ✓ Initiation Phase : In this phase of the project, feedback received from customers is analyzed and brainstorming is done as to develop new product or modify existing product to meet the new demands.
- ✓ Project Definition Phase : In this phase of the project efforts are made to define the solution for the problem posed by customers.
- ✓ Feasibility Study : In this phase, planning of the project is made and definite milestones are established.
- ✓ Project Execution : In this phase all activities and milestones established in the earlier phase are executed in a timely and orderly manner. This phase utilizes maximum of all resources.
- ✓ Project Conclusion : This is the last phase of the project. In this phase, final product or service is handed over to the operations team for commercial production.

Project Management

A project in any organization is collaboration across departments to achieve a single well defined objective. The process of planning, organizing and managing resources to achieve the organizational objective is called project management. Project management is very important in production of goods and services. Idea generation to final production of product or service, each step can be categorized as individual projects. Any project requires a project manager, who leads the project to its logical conclusion. Project manager is responsible for appointing team members with different background but essential in completion of the project.

Management Activities

Basically, the management involves the following activities:

- ✓ Planning - Deciding what is to be done
- ✓ Organizing - Making arrangements
- ✓ Staffing - Selecting the right people for the job
- ✓ Directing - Giving instructions
- ✓ Monitoring - Checking on progress
- ✓ Controlling - Taking action to remedy hold-ups
- ✓ Innovating - Coming up with new solutions
- ✓ Representing - Liaising with users, etc

2. Write short notes on risk evaluation. How to use the decision trees in risk evaluation? Write down the process of Risk matrix to determine risk level.

Risk Evaluation

- ✓ Risk is associated with almost every project.
- ✓ Risk can become an important factor when the project is not able to meet its objectives.
- ✓ Every possible risk must be identified, analyzed and minimized during the development of the software system

Risk Identification and Ranking

- ✓ Every projects evaluation involves risk handling issues.
- ✓ The project evaluation are used to identify the risks and quantify their potential effects.
- ✓ The importance and likelihood are classified as high, medium, low.
- ✓ The project risk matrix may be used as a way of evaluating projects or as a means of identifying and ranking the risks for a specific project.

Risk and Net Present Value

- ✓ A higher discount rate are used to calculate net present value.
- ✓ Projects may be categorized as high, medium or low risk using a scoring method and risk premiums designated for each category.

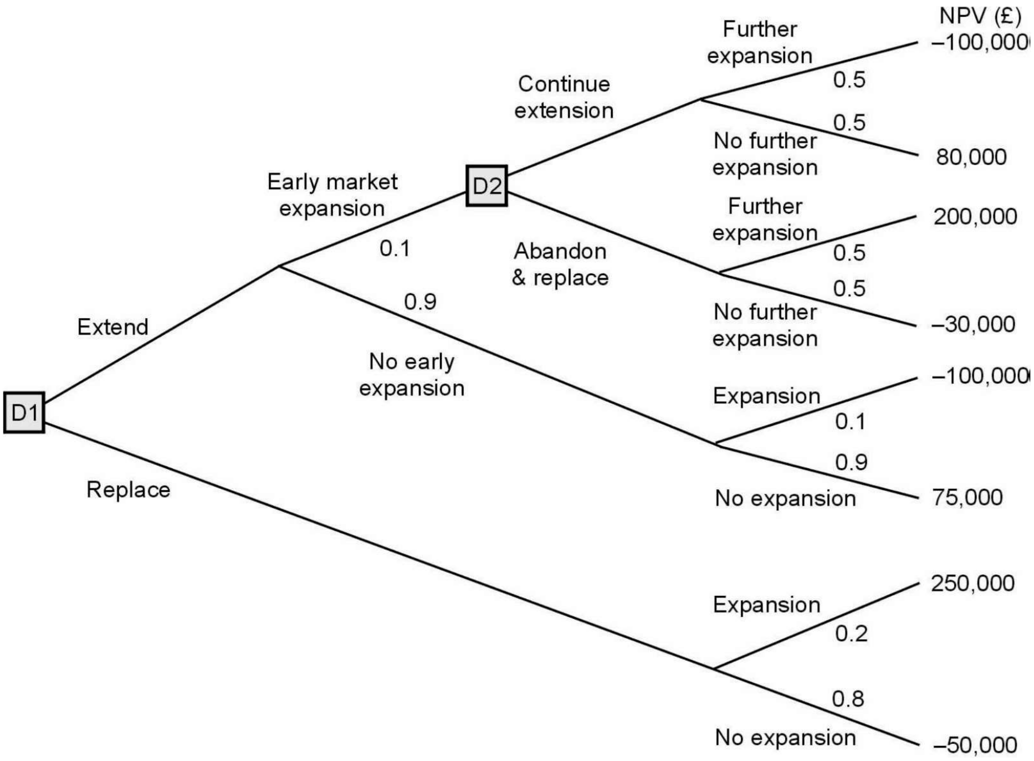
Using Decision Trees

- ✓ Decision trees is a tool which provides evaluation of project's expected outcomes and choosing between the alternative strategies.
- ✓ The analysis of a decision tree consists of evaluating the expected benefit of taking each path from a decision point.
- ✓ The expected value of each path is determined by the sum of the value of each possible outcome multiplied by its probability of occurrence.

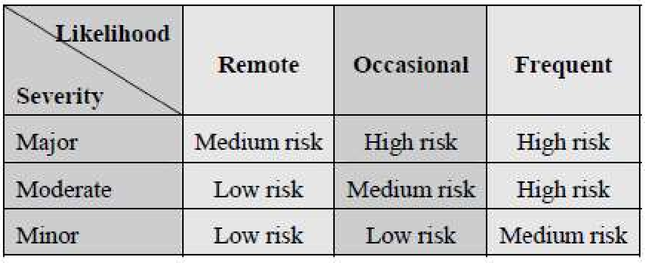
Advantage

- ✓ It will give a precise idea of modeling and analyzing the problems in the project.

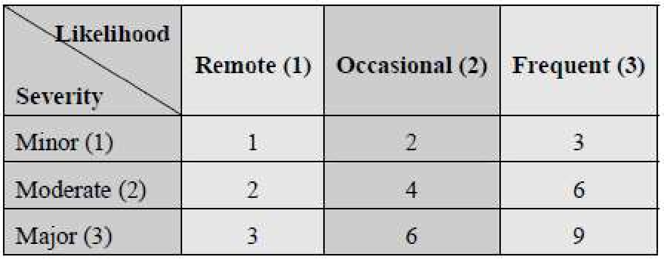
Decision tree



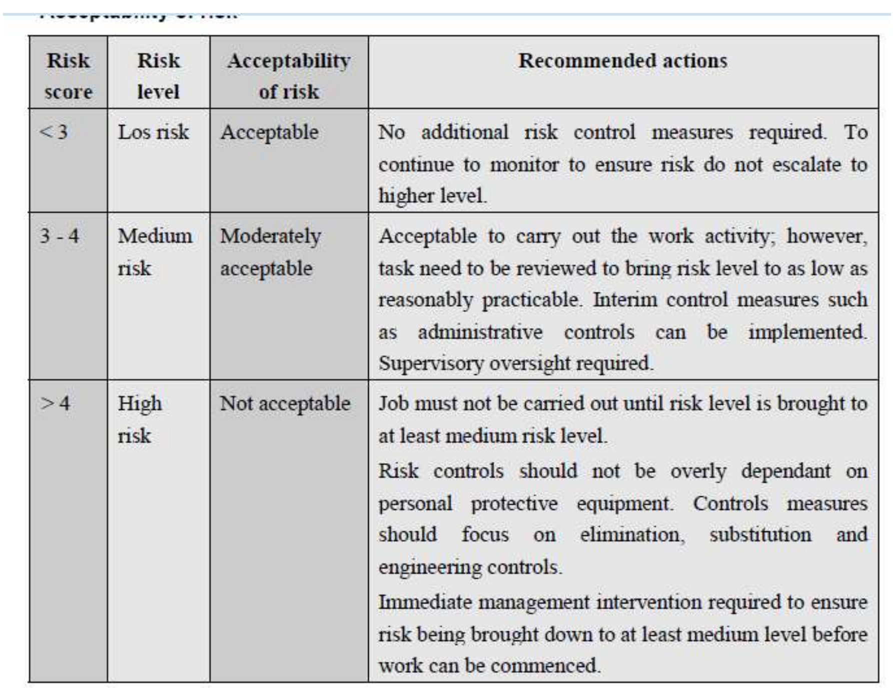
Risk matrix to determine risk level



Likelihood / Severity	Remote	Occasional	Frequent
Major	Medium risk	High risk	High risk
Moderate	Low risk	Medium risk	High risk
Minor	Low risk	Low risk	Medium risk



Likelihood / Severity	Remote (1)	Occasional (2)	Frequent (3)
Minor (1)	1	2	3
Moderate (2)	2	4	6
Major (3)	3	6	9

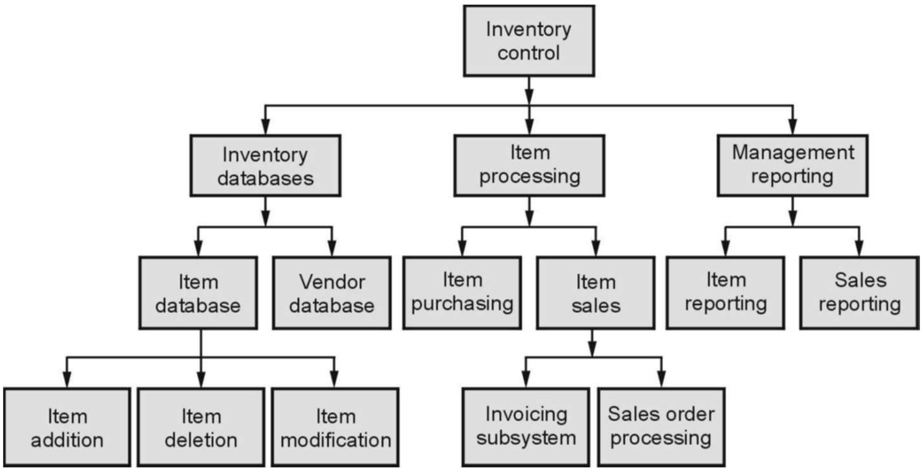


Acceptability of risk

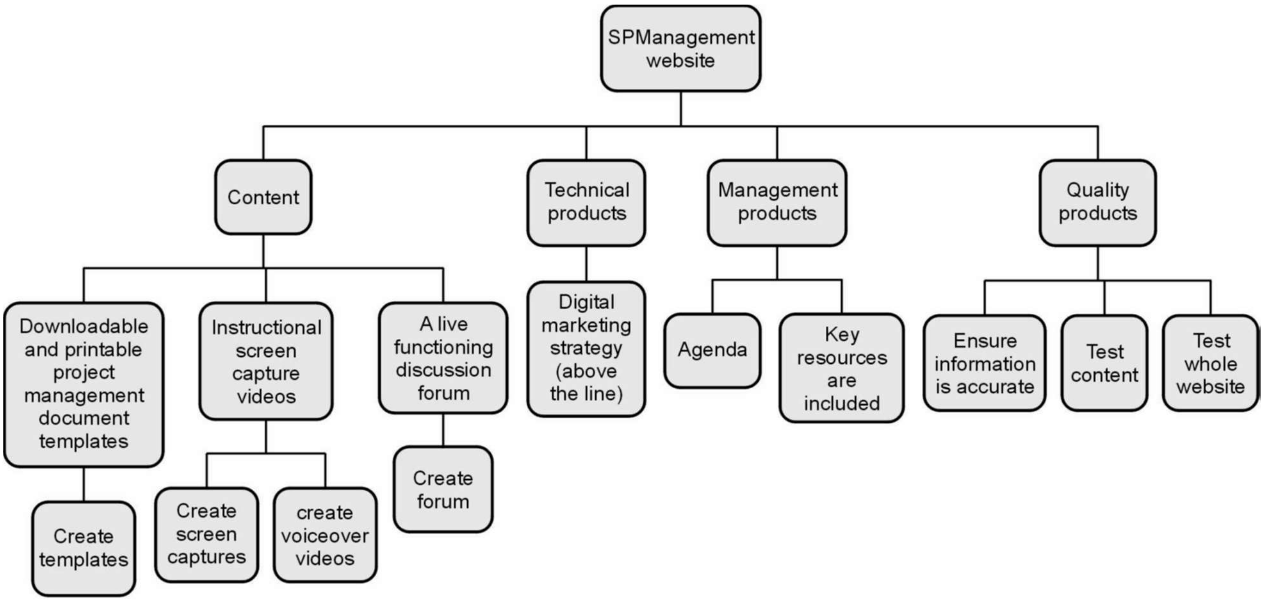
Risk score	Risk level	Acceptability of risk	Recommended actions
< 3	Los risk	Acceptable	No additional risk control measures required. To continue to monitor to ensure risk do not escalate to higher level.
3 - 4	Medium risk	Moderately acceptable	Acceptable to carry out the work activity; however, task need to be reviewed to bring risk level to as low as reasonably practicable. Interim control measures such as administrative controls can be implemented. Supervisory oversight required.
> 4	High risk	Not acceptable	Job must not be carried out until risk level is brought to at least medium risk level. Risk controls should not be overly dependant on personal protective equipment. Controls measures should focus on elimination, substitution and engineering controls. Immediate management intervention required to ensure risk being brought down to at least medium level before work can be commenced.

3. (i) Draw product breakdown structure.
(ii) Sketch the product flow diagram which demonstrates the Project flow.

Product Breakdown structure



Product flow diagram


SPManagement website
Content
Technical products
Management products
Quality products
Downloadable and printable project management document templates
Instructional screen capture videos
A live functioning discussion forum
Digital marketing strategy (above the line)
Agenda
Key resources are included
Ensure information is accurate
Test content
Test whole website
Create templates
Create screen captures
create voiceover videos
Create forum

Product Breakdown Structure (PBS)

- ✓ To show how a system can be broken down into different products for development.

Product Flow Diagram (PFD)

- ✓ To indicate, for each product, which products are required as 'inputs'.

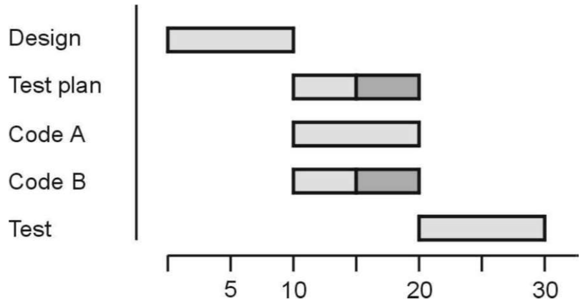
Advantage

- ✓ Less likely to miss a product unexpectedly from a PBS

Hybrid Approach

- ✓ A mix of the activity-based approach and the product-based approach.
- ✓ More commonly used approach.
- ✓ The WBS consists of
  - o a list of the products of the project; and
  - o a list of activities for each product
- ✓ The degree to which the structuring is based on the product or the activity largely depends on the nature of the project and the particular development method.

4. Illustrate the use of Gantt chart in allocation of resources.



The basic steps towards creating a Gantt chart, either by using a sophisticated software, an excel template, or a piece of paper, are the same:

1. Divide the project into manageable tasks, and list them.
2. Set a time definition for each of those tasks. (Start dates and end dates)
3. Set task names.
4. Draw the project timeline, depending on step 2.
5. Observe, identify and mark task dependencies.
6. Use your tasks to fill the bar chart timeline.
7. Assign the tasks.
8. Set milestones for the whole team.
9. Evaluate the critical path.

A graphical visualization of a schedule, where the time span for each activity is depicted by the length of a segment drawn on an adjacent calendar

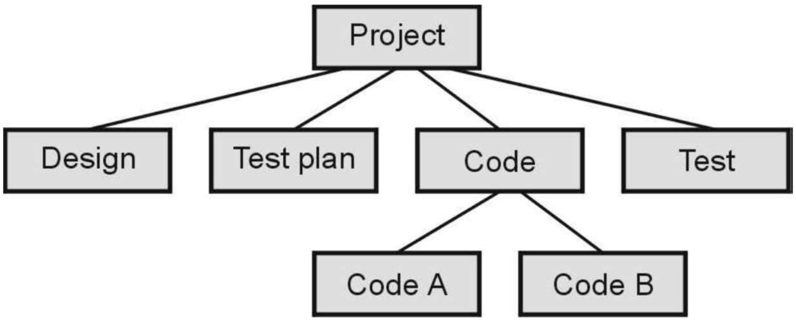
- ✓ Generally, does not show task decomposition
- ✓ Does not show duration, only the time span over which the task is scheduled
- ✓ Does not show precedence relations
- ✓ Can show activity of multiple developers in parallel
- ✓ Makes it easy to monitor a project's progress and expenditures.

Project Plan Structure

- ✓ Introduction
- ✓ Project organisation
- ✓ Risk analysis
- ✓ Hardware and software resource requirements
- ✓ Work breakdown
- ✓ Project schedule
- ✓ Monitoring and reporting mechanisms

Hierarchical decomposition of a project into subtasks

- ✓ Shows how tasks are decomposed into subtasks
- ✓ Does not show duration
- ✓ Does not show precedence relations (e.g. task A must be finished before task B can start)



5. The estimated cost has the estimated income and benefits; the best way of the project evaluation includes the developmental costs in terms of accruing from the system. How the overall estimation of the required activities, the installation cost and the operational cost have been performed? Interpret how cost- benefits evaluation techniques is done & its methods with examples.

Techniques

1. Net profit
2. Payback reriod
3. Return on investment
4. Net present value
5. Internal rate of return

Net profit

- ✓ The net profit of a project is the difference between the total costs and the total income over the life of the project.
- ✓ Net profits do not involve the timing of the cash flows.
- ✓ Project incomes are returned only towards the end of the project.
- ✓ Net profit = Total income – Total costs

Payback period

- ✓ Time taken to break even or payback the initial investment.
- ✓ The project with the shortest payback period will be chosen on the basis that an organization will wish to minimize the time that a project is 'in debt'
- ✓ Minimize the time limit.
- ✓ The payback period is simple to calculate but sensitive to forecasting errors.
- ✓ The limitation of the payback period is that it ignores the overall profitability of the project.
- ✓ Payback period = Time taken to break even

Return On Investment (ROI)

- ✓ ROI = (Average annual profit/total investment) * 100.
- ✓ Also known as Accounting Rate of Return (ARR).
- ✓ Provides a way of comparing the net profitability to the investment required.
- ✓ Return on Investment (ROI) = (Average annual profit / Total investment) * 100 %.

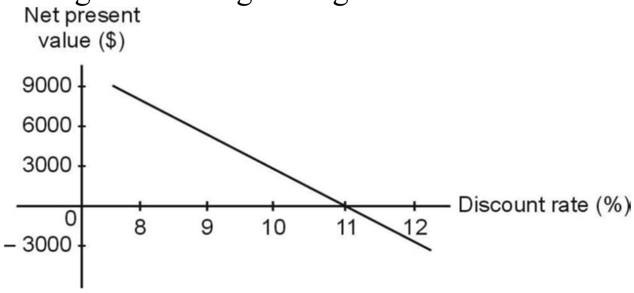
Net Present Value (NPV)

- ✓ A Project evaluation technique that takes into account the profitability of a project and the timing of the cash flow rates are produced.
- ✓ Sum of all incoming and outgoing payments, discounted using an interest rate, to a fixed point in time (the present).
- ✓ Present value = (value in year t) / (1 + r) ^ t'
  - o r is the discount rate.
  - o t is the number of years into the future that the cash flow occurs.
- ✓ Net present value can also be calculated by multiplying the cash flow by the appropriate discount factor.
- ✓ NPV for project is obtained by summing the discounted values and discounting each cash flows.
- ✓ The NPV for project is obtained by discounting each cash flow and summing the discounted values.
- ✓ (1 + r) ^ t is known as discount factor
- ✓ In the case of 10% rate and one year
  - o Discount factor = 1/(1 + 0.10) = 0.9091
- ✓ In the case of 10% rate two years

    - o Discount factor = 1 = (1.10 □ 1.10) 0.8294
- ✓ Discount rate is the annual rate by which we discount future earnings.
    - o e.g. If discount rate is 10% and the return of an investment in a year is $110, the present value of the investment is $100.

Internal Rate of Return (IRR)

- ✓ Attempts to provide a profitability measure as a percentage return that is directly comparable with interest rates.
- ✓ The IRR is calculated as that percentage discount rate that would produce an NPV of zero.
- ✓ A spreadsheet or a small computer program can be used to calculate the IRR is a convenient and useful measure of value of a project.
- ✓ The limitation of IRR is that it does not indicate the absolute size of the return value.
- ✓ Can be used to compare different investment opportunities.
- ✓ May be estimated by plotting a series of guessing.



UNIT – II
2- Marks

1. How to identify the project as either Object – Driven or Product – Driven.

Project whose requirement is to meet certain objectives which could be met in a number of ways, is objective-based project. Project whose requirement is to create a product, the details of which have been specified by the client, is product-based project.

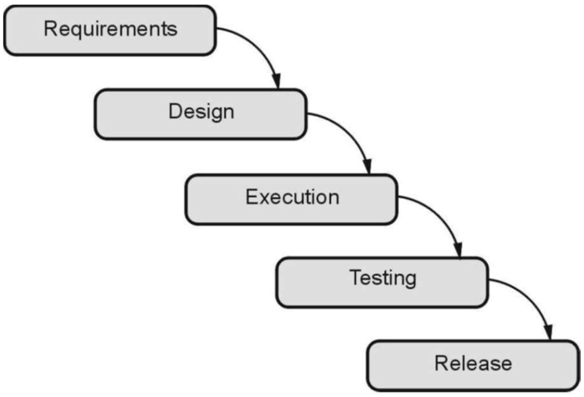
2. Whether RAD is the incremental model? Give the Justification of the incremental model.

RAD model is Rapid Application Development model. It is a type of incremental model. In RAD model the components or functions are developed in parallel as if they were mini projects. The developments are time boxed, delivered and then assembled into a working prototype.

3. How to be aware of the major shortcomings of the SLOC measure.

Source Lines of Code (SLOC): is software metric used to measure the size of software program by counting the number of lines in the text of the program's source code. This metric does not count blank lines, comment lines, and library. SLOC measures are programming language dependent. They cannot easily accommodate nonprocedural languages. SLOC also can be used to measure others, such as errors/KLOC, defects/KLOC, pages of documentation/KLOC, cost/KLOC. ii) Deliverable Source Instruction (DSI): i similar to SLOC. The difference between DSI and SLOC is that s "if then-else" statement, it would be counted as one SLOC but might be counted as several DSI .

4. Draw the Waterfall model with neat diagram.



5. Give some units for measuring the size of the software.

Number of Lines of Code (NLOC)

- number of delivered source instructions (NDSI)
- number of thousands of delivered source instructions (KDSI) Function Point Count

A measure of the functionality perceived by the user delivered by the software developer. A function count is a weighted sum of the number of

- Inputs to the software application
- Outputs from the software application
- Enquiries to the software application
- Data files
- Internal to the software application

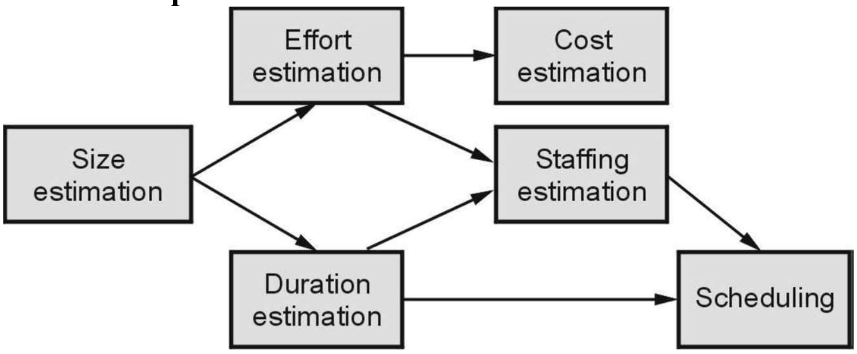
6. List out the types of Software Process Models.

- Waterfall Model
- V-Shaped Model
- Prototyping Model
- Spiral Method (SDM)
- Iterative and Incremental Method
- Rapid Application Development

7. Mention few Agile Methods.

- Adaptive Software Development (ASD)
- Feature Driven Development (FDD)
- Crystal Clear
- Dynamic Software Development Method (DSDM)
- Rapid Application Development (RAD)
- Scrum
- Extreme Programming (XP)
- Rational Unify Process (RUP)

8. Draw the process of Software Cost Estimation



9. Point out the Software Cost Estimation Techniques.

Empirical techniques:

- An educated guess based on past experience.

Heuristic techniques:

- Assume that the characteristics to be estimated can be expressed in terms of some mathematical expression.

Analytical techniques:

- Derive the required results starting from certain simple assumptions.

10. Give some units for measuring the size of the software.

LOC (Lines of Code):

- Simplest and most widely used metric.

- ❖ Comments and blank lines should not be counted.

Function Point Metric

- ❖ Overcomes some of the shortcomings of the LOC metric
- ❖ A set of related inputs is counted as one input.

Big Questions

1.Interpret about the Estimation of Software and Software Effort Estimation Technique.

Estimation of Software

- ✓ Software projects are notorius for going past their deadline, going over budget, or both.
- ✓ The problem lies in the estimation of the amount of effort required for the development of a project.
- ✓ The cost estimation is usually dependent upon the size estimate of the project, which may use lines of code or function points as metrics.
- ✓ There are several different techniques for performing software cost estimation, including expert judgement and algorithmic models. Estimation by expert judgement is a common way of estimating the effort required for a project. Unfortunately, this method of estimation does not emphasize re-estimation during the project life cycle, which is an important part of project tracking, because it allows the estimates to be improved during the project life cycle.
- ✓ The quality of a cost estimation model is not so much attributed to the initial estimate, but rather the speed at which the estimates converges to the actual cost of the project.
- ✓ COCOMO is a popular algorithmic model for cost estimation whose cost factors can be tailored to the individual development environment, which is important for the accuracy of the cost estimates. More than one method of cost estimation should be done so that there is some comparison available for the estimates.
- ✓ This is especially important for unique projects. Cost estimation must be done more diligently throughout the project life cycle so that in the future there are fewer surprises and unforeseen delays in the release of a product.

Basic Steps in Software Effort Estimation

- ✓ Identify project objectives and requirements
- ✓ Plan the activities
- ✓ Estimate product size and complexity
- ✓ Estimate effort, cost and resources
- ✓ Develop projected schedule
- ✓ Compare and iterate estimates
- ✓ Follow up

Basic Algorithmic Form

Effort = constant + coefficient*(size metric) + coefficient*(cost driver 1) + coefficient*(cost driver 2) + coefficient*(cost driver 3) + …

size metric lines of code

    ‘new’ versus ‘old’ lines of code

function points

SLOC as an Estimation Tool

- ✓ Why used ?
  - o early systems emphasis on coding
- ✓ Criticisms
  - o cross-language inconsistencies
  - o within language counting variations
  - o change in program structure can affect count
  - o stimulates programmers to write lots of code
  - o system-oriented, not user-oriented

2. (i) Briefly demonstrate the effort and cost estimation techniques
(ii) Analyze the importance of COCOMO model for software estimation

(i) Effort and Cost estimation techniques

Estimating

- ✓ The process of forecasting or approximating the time and cost of completing project deliverables.
- ✓ The task of balancing the expectations of stakeholders and the need for control while the project is implemented

Types of Estimates

- ✓ Top-down (macro) estimates : analogy, group consensus, or mathematical relationships, derived from experience to estimate project duration and total cost. Could be made by a manager with no direct experience of the processes to complete the project.
- ✓ Bottom-up (micro) estimates: estimates of elements of the work breakdown structure, require more effort to develop & rely upon those who understand the work to estimate specific work activities

Macro (Top-down) Approaches

- Consensus methods
- Ratio methods
- Apportion method
- Function point methods for software and system projects
- Learning curves

Micro (Bottom-up) Approaches

- Template method
- Parametric Procedures Applied to Specific Tasks
- Detailed Estimates for the WBS Work Packages
- Phase Estimating: A Hybrid

(ii) COCOMO Model

- COCOMO (COnstructive COst MOdel) proposed by Boehm.
- Divides software product developments into 3 categories :
  - Organic
  - Semidetached
  - Embedded

COCOMO Product classes

- Roughly correspond to:
- Application, utility and system programs respectively.
- Data processing and scientific programs are considered to be application programs.
- Compilers, linkers, editors, etc., are utility programs.
- Operating systems and real-time system programs, etc. are system programs.

Software cost estimation is done through three stages :

- Basic COCOMO,
- Intermediate COCOMO,
- Complete COCOMO.

Basic COCOMO Model

- Gives only an approximate estimation :
  - Effort = a1 (KLOC)a2
  - Tdev = b1 (Effort)b2
- KLOC is the estimated kilo lines of source code,
- a1,a2,b1,b2 are constants for different categories of software products,
- Tdev is the estimated time to develop the software in months,
- Effort estimation is obtained in terms of person months (PMs).

Intermediate COCOMO

- Basic COCOMO model assumes
  - effort and development time depend on product size alone
- However, several parameters affect effort and development time:
  - Reliability requirements
  - Availability of CASE tools and modern facilities to the developers
  - Size of data to be handled
- For accurate estimation,
  - the effect of all relevant parameters must be considered:
  - Intermediate COCOMO model recognizes this fact:

- ❖ Refines the initial estimate obtained by the basic COCOMO by using a set of 15 cost drivers (multipliers).

Complete COCOMO

- ❖ Cost of each sub-system is estimated separately.
- ❖ Costs of the sub-systems are added to obtain total cost.
- ❖ Reduces the margin of error in the final estimate.

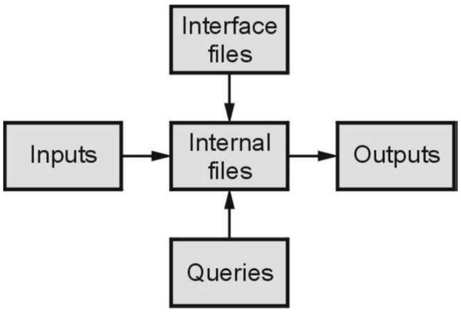
Complete COCOMO Example

- ❖ A Management Information System (MIS) for an organization having offices at several places across the country :
  - o Database part (semi-detached)
  - o Graphical User Interface (GUI) part (organic)
  - o Communication part (embedded)
- ❖ Costs of the components are estimated separately :
  - o summed up to give the overall cost of the system

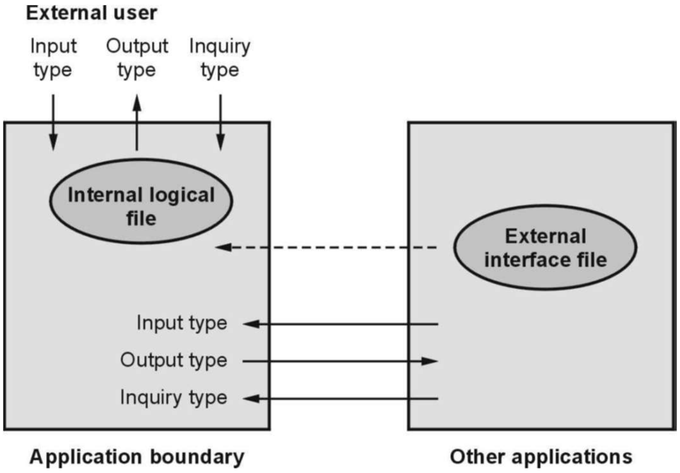
3. How do you find the cosmic full function point measurement method associates the functional user requirements for each piece with a specific layer as each layer possesses an intrinsic boundary for which specific users are identified?

COSMIC Full function point

COSMIC function points are a unit of measure of software functional size. There are other ways to assess software size, but COSMIC Sizing is the most universal, meaningful and useful approach. The functional size is consistent regardless of the technology used to build it. The size can be estimated, or if all requirements are available, measured. Early estimation is very useful for planning and managing software endeavours (projects or product management). The process of measuring software size is called functional size measurement (FSM). COSMIC functional size measurement is applicable to business software, real-time software and infrastructure software at any level of decomposition (from a whole software system down to a single re-usable component or a user story). The functional size is independent of the technology or processes used to develop the system. COSMIC is an ISO standard. It is a refined improvement over its predecessors (IFPUG and Mark II FP). The unit of size is the COSMIC Function Point or CFP.



Functionality Types



Uses

Once you have measured (or estimated) the size in COSMIC Function Points you can then use this as the base metric to:

- Estimate development effort
- Estimate project duration
- Estimate project quality achievement
- Estimate test effort
- Control scope creep
- As the basis for fixed price contracts
- Assess the replacement cost of a software asset
- Estimate maintenance and replacement costs
- Assess the achievement of quality (defect removal rates)

Function Point Calculation

$$\text{Function Counts} = FC = \sum_{i=1}^{5}\sum_{j=1}^{3} x_i\, w_j$$

$$\text{Function Points} = FP = FC\left[.65 + .01\left(\sum_{k=1}^{14} C_k\right)\right]$$

where

$$x_i = \text{function}_i$$

$$w_j = \text{weight}_j$$

$$c_k = \text{complexity factor}_k$$

4. The systems development life cycle (SDLC) is the part of the application development life-cycle in systems engineering, information systems and software engineering to describe a process for planning, creating, testing, and deploying an information system.

- The systems development life cycle (SDLC), also referred to as the application development life-cycle, is a term used in systems engineering, information systems and software engineering to describe a process for planning, creating, testing, and deploying an information system.
- The systems development life-cycle concept applies to a range of hardware and software configurations, as a system can be composed of hardware only, software only, or a combination of both.
- A systems development life cycle is composed of a number of clearly defined and distinct work phases which are used by systems engineers and systems developers to plan for, design, build, test, and deliver information systems.
- Like anything that is manufactured on an assembly line, an SDLC aims to produce high-quality systems that meet or exceed customer expectations, based on customer requirements, by delivering systems which move through each clearly defined phase, within scheduled time frames and cost estimates.
- Computer systems are complex and often (especially with the recent rise of service oriented architecture) link multiple traditional systems potentially supplied by different software vendors.
- To manage this level of complexity, a number of SDLC models or methodologies have been created, such as "waterfall"; "spiral"; "Agile software development"; "rapid prototyping"; "incremental"; and "synchronize and stabilize".
- SDLC can be described along a spectrum of agile to iterative to sequential. Agile methodologies, such as XP and Scrum, focus on lightweight processes which allow for rapid changes (without necessarily following the pattern of SDLC approach) along the development cycle.
- Iterative methodologies, such as Rational Unified Process and dynamic systems development method, focus on limited project scope and expanding or improving products by multiple iterations.

- ❖ Sequential or big-design-up-front (BDUF) models, such as waterfall, focus on complete and correct planning to guide large projects and risks to successful and predictable results. Other models, such as anamorphic development, tend to focus on a form of development that is guided by project scope and adaptive iterations of feature development.
- ❖ In project management a project can be defined both with a project life cycle (PLC) and an SDLC, during which slightly different activities occur. According to Taylor (2004), "the project life cycle encompasses all the activities of the project, while the systems development life cycle focuses on realizing the product requirements".
- ❖ SDLC is used during the development of an IT project, it describes the different stages involved in the project from the drawing board, through the completion of the project.

Phases

- ❖ The system development life cycle framework provides a sequence of activities for system designers and developers to follow.
- ❖ It consists of a set of steps or phases in which each phase of the SDLC uses the results of the previous one.
- ❖ The SDLC adheres to important phases that are essential for developers, such as planning, analysis, design, and implementation, and are explained in the section below.
- ❖ It includes evaluation of present system, information gathering, feasibility study and request approval. A number of SDLC models have been created: waterfall, fountain, spiral, build and fix, rapid prototyping, incremental, synchronize and stabilize.
- ❖ The oldest of these, and the best known, is the waterfall model: a sequence of stages in which the output of each stage becomes the input for the next. These stages can be characterized and divided up in different ways, including the following:
    - o Preliminary analysis
    - o Systems analysis, requirements

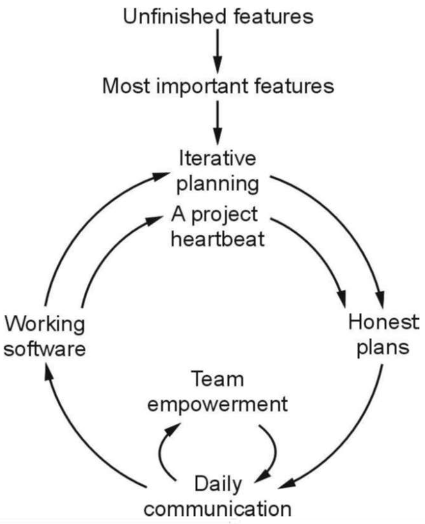
A series of steps followed by the developer are:

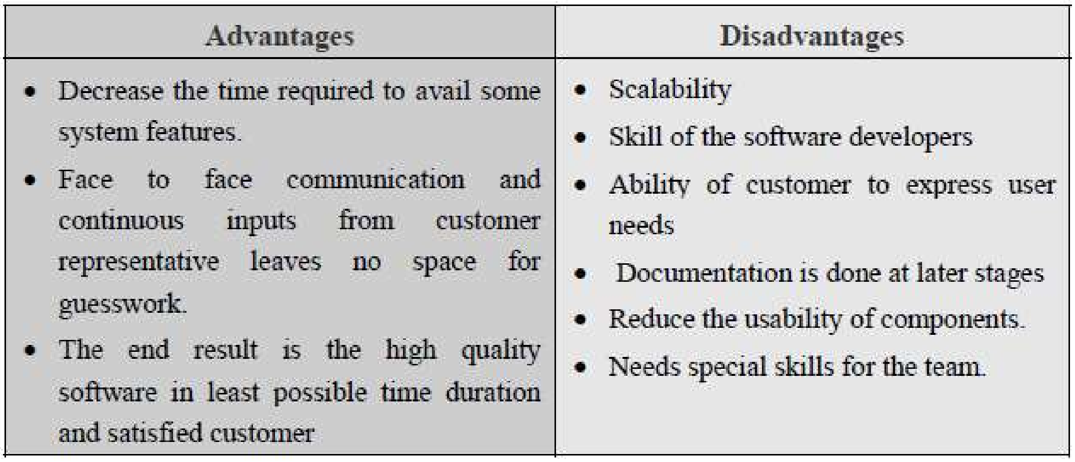
1. Collection of Facts: End user requirements are obtained through documentation, client interviews, observation and questionnaires,
2. Scrutiny of the existing system: Identify pros and cons of the current system in-place, so as to carry forward the pros and avoid the cons in the new system.
3. Analyzing the proposed system: Solutions to the shortcomings in step two are found and any specific user proposals are used to prepare the specifications.
    - o Systems design
    - o Development
    - o Integration and testing
    - o Acceptance, installation, deployment
    - o Maintenance
    - o Evaluation
    - o Disposal

5. Analyze the core values XP.

Extreme programming (XP)

- ❖ It is based on iterative and incremental development, where requirements and solutions evolve through collaboration between cross-functional teams.
- ❖ For small-to-medium-sized teams developing software with vague or rapidly changing requirements
- ❖ Coding is the key activity throughout a software project
- ❖ Communication among teammates is done with code
- ❖ Life cycle and behavior of complex objects defined in test cases – again in code
- ❖ It can be used with any type of the project, but it needs more involvement from customer and to be interactive. Also, it can be used when the customer needs to have some functional requirement ready in less than three weeks





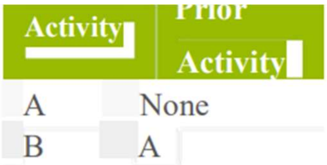
Advantages	Disadvantages
• Decrease the time required to avail some system features. • Face to face communication and continuous inputs from customer representative leaves no space for guesswork. • The end result is the high quality software in least possible time duration and satisfied customer	• Scalability • Skill of the software developers • Ability of customer to express user needs • Documentation is done at later stages • Reduce the usability of components. • Needs special skills for the team.

Unit – III
2 marks

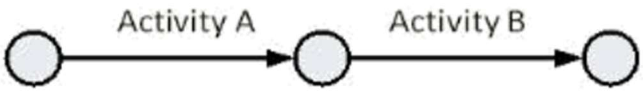
1. **List out the basis constraint to draw bar chart.**

- ❖ The chart has been drawn up taking account of the nature of the development process and the resources that are available.
- ❖ Sequenced the tasks- identify the dependencies.
- ❖ Scheduled them- specify when they should take place.

2. **Draw an arrow network with suitable examples**



Activity	Prior Activity
A	None
B	A



3. **What are the objectives of activity planning?**

- ❖ Feasibility assessment
- ❖ Resources allocation
- ❖ Detailed costing
- ❖ Motivation
- ❖ Co-ordination

4. **What do you understand by work breakdown structure.**

- ❖ Work Breakdown Structure (WBS) provides a notation for representing task structure:
  - o Activities are represented as nodes of a tree.

- The root of the tree is labelled by the problem name.
- Each task is broken down into smaller tasks and represented as children nodes.

Big Questions

1. The project schedule is the core of the project plan, it is used by the project manager to commit people to the project and show the organization how the work will be performed? Explain various steps involved & the objectives of activity planning in detail.

Activity Planning

- ❖ A project plan is a schedule of activities indicating the start and stop for each activity
- ❖ Also provide the project and resource schedules
- ❖ The start and stop of each activity should be visible and easy to measure
- ❖ Each activity should have some 'deliverables' for ease of monitoring
- ❖ During planning, managers consider :
  - ✓ Resource available : Make sure the resources are there when needed.
  - ✓ Resource allocation : Make sure there are no competing resources.
  - ✓ Staff responsibility : Schedule showing which staff carry out each activity
  - ✓ Project Monitoring : Measure the actual achievement.
  - ✓ Cash flow forecasting : Produce a timed cash flow forecast.
  - ✓ Re-planning of the project towards the pre-defined goal : Re-plan the project so that it will correct drift from the target.

Objectives of Activity Planning

- ❖ Once a detailed activity plan is finished, it can be used to achieve the following:
- ❖ Feasibility assessment : Can the project be delivered on time and within budget (constraints) ?
- ❖ Resources allocation :
  - ✓ How to allocate the resources with best results ?
  - ✓ When should those resources be ready ?
- ❖ Detailed costing :
  - ✓ A detailed estimates on the project cost and the timings.
  - ✓ A detailed forecast on when the expenditure is likely to take place.
- ❖ Motivation :
  - ✓ Providing targets and being able to monitor the achievement of the targets at the end of the activity can be a good strategy to motivate staff.
- ❖ Co-ordination :
  - ✓ Help to set the time and requirements of staff from different departments to work together in the project, if necessary.
  - ✓ Provide a good way for the project teams to communicate, cooperate and collaborate among themselves.

Different Levels of Plans

- ❖ Project Schedule : A plan that shows
  1. the dates when each activity should start and stop
  2. when and how much of the resources will be required
- ❖ Activity Plan : A plan that describes
  1. how each activity will be undertaken
  2. the activity plan is done in steps 4 and 5 of step Wise framework.

Project Vs Activity

- ❖ A project is composed of a number of related activities.
- ❖ A project may start when at least one of its activities is ready to start.
- ❖ A project will be completed when all of its activities have been completed.
- ❖ An activity must have a clear start and a clear stop.
- ❖ An activity should have a duration that can be forecasted.
- ❖ Some activities may require that other activities are completed before they can begin.

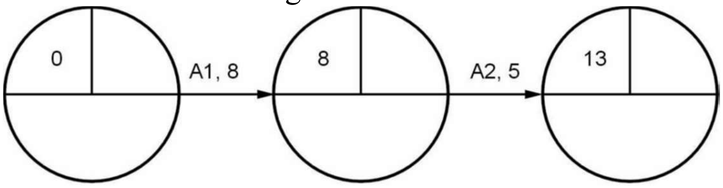
2. Demonstrate the Network planning model, with an example how critical path can be identified in precedence networks.

Critical Path Method (CPM):

- Developed to co-ordinate maintenance projects in the chemical industry.
- A complex undertaking, but individual tasks are routine (tasks' duration = deterministic)

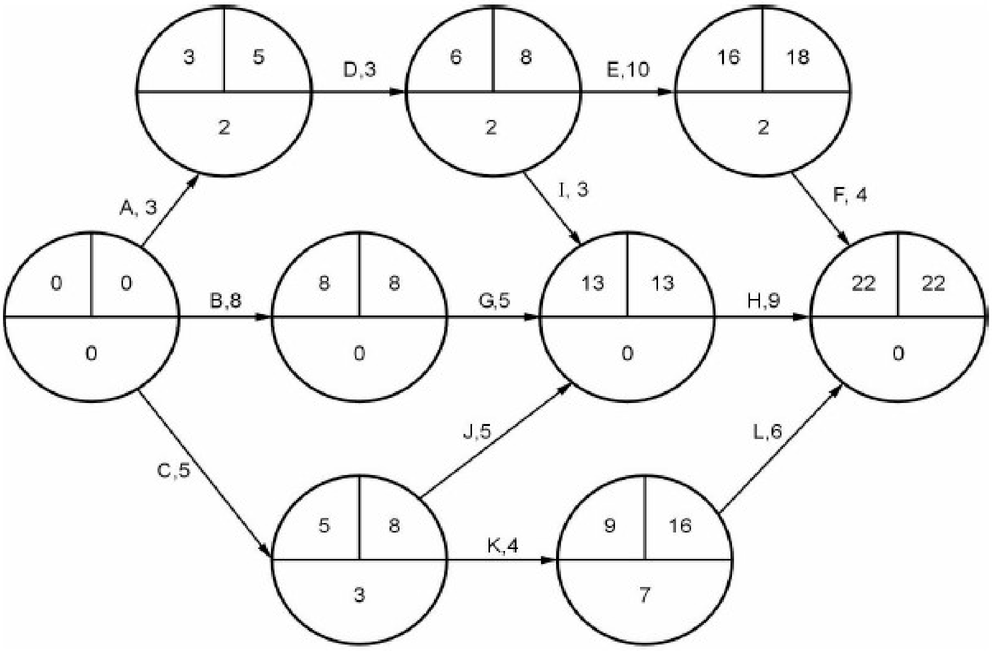
The forward pass

- Calculate the earliest date on which each event may be achieved and the earliest dates on which each activity may be started and completed.
- Steps involved in forward pass are,
  - Start at the start node.
  - Compute the top pair of numbers.
  - Add the duration to the connecting node's earliest finish time.



The backward pass

- Calculate the latest date at which each event may be achieved and each activity started and finished, without delaying the end date of the project.
- The steps involved in backward pass are,
  - Start at the end node
  - Compute the bottom pair of numbers
  - Subtract the duration from the connecting node's earliest start time.



Identifying the critical path

- The critical path is identified by using the activity-on-node networks.
- Slack is used to identify the path.
- Slack is the difference between the earliest date and the latest date of an event. A – B – G – H is the critical path.