IITM GROUP OF INSTITUITIONS
MURTHAL, SONIPAT
CSE404B SOFTWARE PROJECT MANAGEMENT

SOFTWARE PROJECT MANAGEMENT

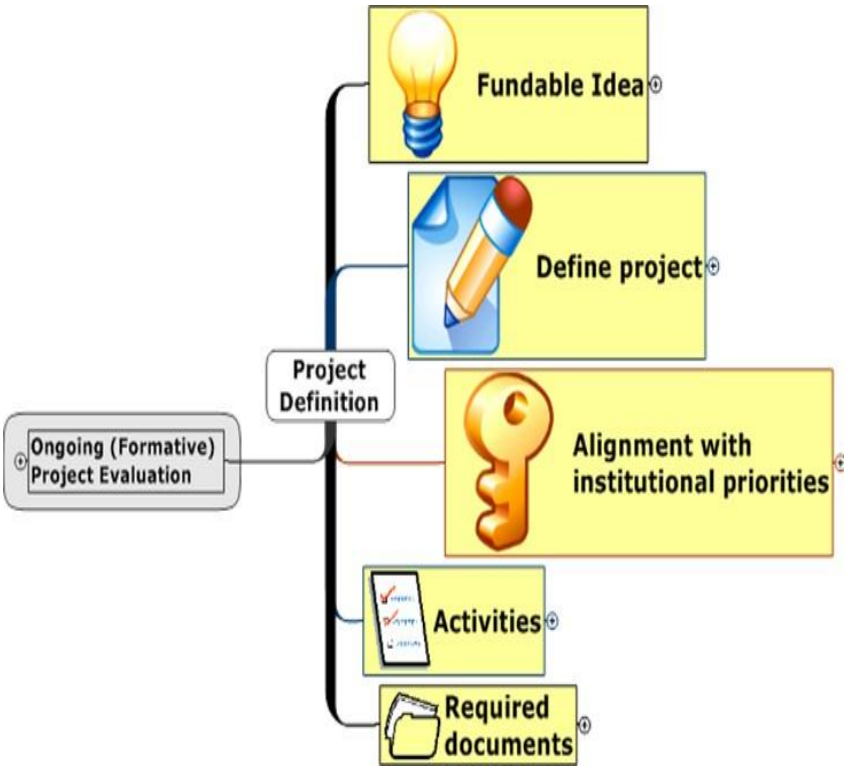
1. What is a Project?

A project is a temporary endeavor undertaken to create a unique product or service

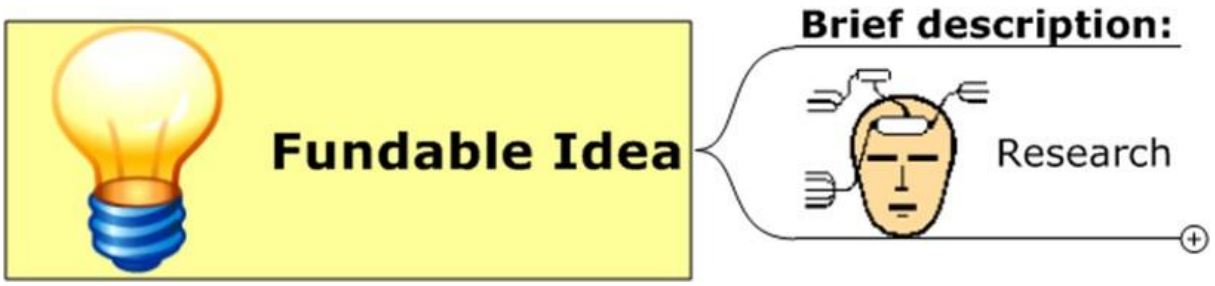
- One time
- Limited funds/time
- Specific resources utilized
- Performed by people - Single or multi-person team
- Planned, controlled
- Specific Deliverables

The Triple Constraint of Projects

- On Time, Budget, Quality = Required Scope


Fundable Idea
Define project
Project Definition
Alignment with institutional priorities
Ongoing (Formative) Project Evaluation
Activities
Required documents

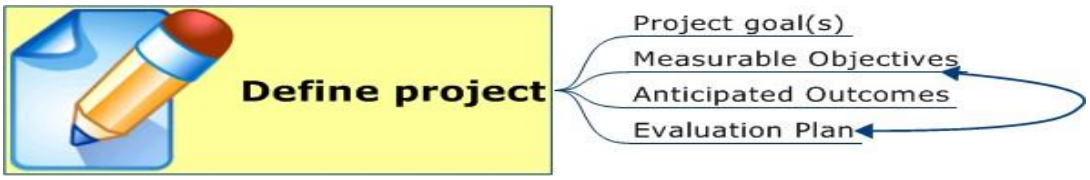
Fundable Idea



Research

- Potential Funding
- Best Practices
- Potential Partners

Define project



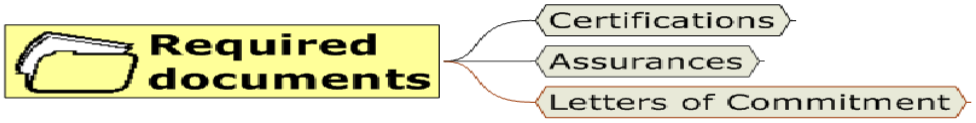
Alignment with institutional priorities

Activities

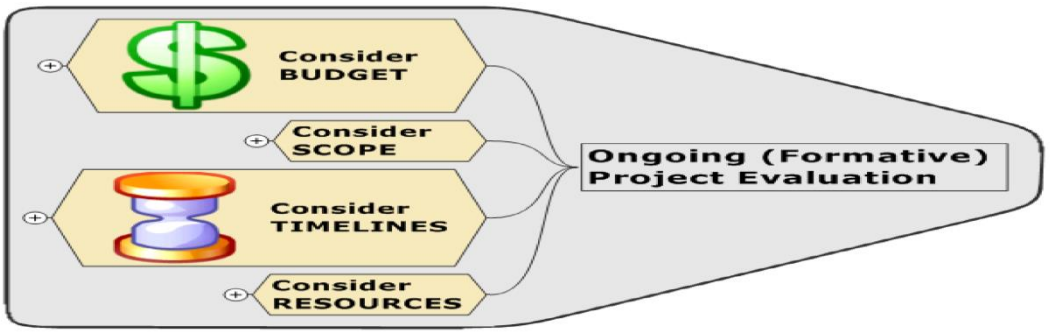
- In support of GOAL
- Realistic and Achievable Objectives
- Identify Resources Required
- Timeline for Implementation
- Criteria for success

- Evaluation Method/Instruments
- Budget Requirements

Required documents



Ongoing (Formative) Project Evaluation



Consider BUDGET

- Going over budget
  - Will result in...
- Having to get more money
- Having to reorganize budget
- Adjusting budget due to scope change
- Adjusting budget due to resources change

Consider SCOPE

- Changing scope
- Increasing scope
- Decreasing scope
- Changing direction

Consider TIMELINES

- Extending deadlines
- Hurrying deadlines
- Changing deadlines for certain project components
- Adjusting schedule due to scope change
- Adjusting timelines due to resource change

Consider RESOURCES

- Losing resources
- Having to replace resources
- Running short on resources

2. Contract management

Contract management or contract administration is the management of contracts made with customers, vendors, partners, or employees. Contract management includes negotiating the terms and conditions in contracts and ensuring compliance with the terms and conditions, as well as documenting and agreeing on any changes or amendments that may arise during its implementation or execution. It can be summarized as the process of systematically and efficiently managing contract creation, execution, and analysis for the purpose of maximizing financial and operational performance and minimizing risk.

Common commercial contracts include employment letters, sales invoices, purchase orders, and utility contracts. Complex contracts are often necessary for construction projects, goods or services that are highly regulated, goods or services with detailed technical specifications, intellectual property (IP) agreements, and international trade.

A study has found that for "42% of enterprises...the top driver for improvements in the management of contracts is the pressure to better assess and mitigate risks" and additionally,"nearly 65% of enterprises report that contract lifecycle management (CLM) has improved exposure to financial and legal risk

A contract is a written or oral legally-binding agreement between the parties identified in the agreement to fulfill the terms and conditions outlined in the agreement. A prerequisite requirement for the enforcement of a contract, amongst other things, is the condition that the parties to the contract accept the terms of the claimed contract. Historically, this was most commonly achieved through signature or performance, but in many jurisdictions - especially with the advance of electronic commerce - the forms of acceptance have expanded to include various forms of electronic signature.

2.1 CONTRACTS

Contracts can be of many types, e.g. sales contracts (including leases), purchasing contracts, partnership agreements, trade agreements, and intellectual property agreements.

- A sales contract is a contract between a company (the seller) and a customer that where the company agrees to sell products and/or services. The customer in return is obligated to pay for the product/services bought.
- A purchasing contract is a contract between a company (the buyer) and a supplier who is promising to sell products and/or services within agreed terms and conditions. The company (buyer) in return is obligated to acknowledge the goods / or service and pay for liability created.
- A partnership agreement may be a contract which formally establishes the terms of a partnership between two legal entities such that they regard each other as 'partners' in a commercial arrangement. However, such expressions may also be merely a means to reflect the desire of the contracting parties to act 'as if' both are in a partnership with common goals. Therefore, it might not be thecommon law arrangement of a partnership which by definition creates fiduciary duties and which also has 'joint and several' liabilities.

3. Activities covered by software project

A Software project is concerned not only with the actual writing of software. In fact, where a software application is bought in 'Off-the-self', there might be no software writing as such. This is still fundamentally a software project because so many of the other elements associated with this type of project are present.

Usually, there are three successive process that bring a new system into being · Feasibility Study
· Planning
· Program Execution

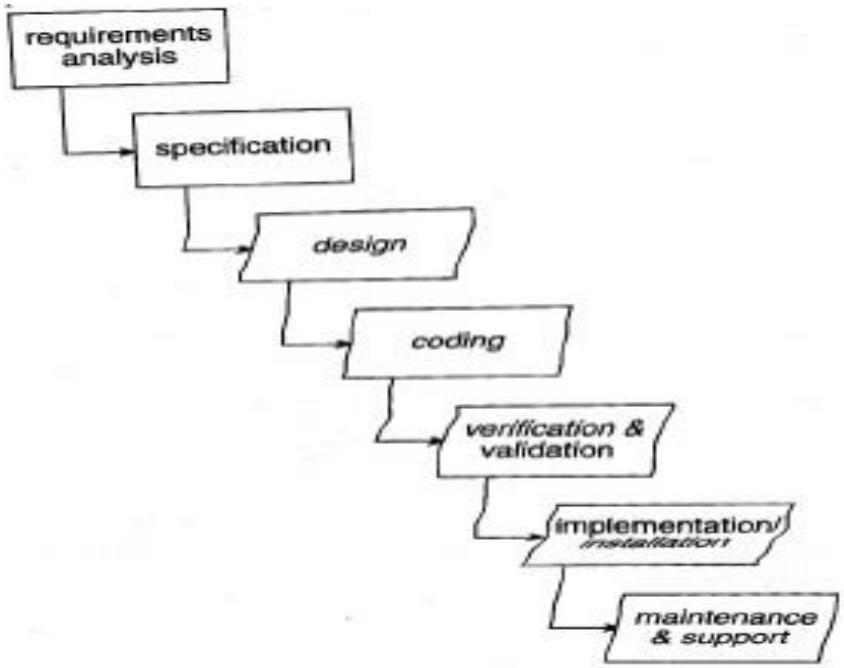
Let us understand what is the mean by 'Feasibility Study'?

- A feasibility study may be carried out before a project commences Its Purpose
  - To establish an outline of what users require
- · To ensure it is feasible to meet their needs · Entirely In part
  - To record initial impressions of how to meet the requirements
  - To give management a feel for costs and timescales of project
  - To provide as much information as possible to project manager for · Estimating
  - Purchasing and accommodation
  - To help management decide whether to proceed with the project Planning
  - Probably the most time-consuming project management activity
  - Continuous activity from initial concept through to system delivery.
- Plans must be regularly revised as new information becomes available
  - Various different types of plan may be developed to support the main software project plan that is concerned with schedule and budget

Project execution:

- The project can now be executed. Individual projects are likely to differ considerably but a classic project life cycle is shown in fig.Requirementsanalysis .

- This is finding out in detail what the users require of the system that the project is to implement.
- Some work along these lines will almost certainly have been carried out when the project was evaluated but now the original information obtained needs to be updated and supplemented.
- Several different approaches to the users' requirements may be explored.



For example, a small system that satisfies some, but not all, of the users' needs at a low price may be compared to a system with more functions but at a higher price. Specification Detailed documentation of what the proposed system is to do. Design A design that meets the specification has to be drawn
up. This design activity will be in two stages. One will be the external or user design. This lays down what the system is to look like to the users in terms of menus, screen and report layouts and so on. The next stage produces the physical design, which tackles the way in which the data and software procedures
are be structured internally.

Coding:

This might refer to writing code in a procedural language such as C or Ada, or might refer to the use of a high level application builder. Even where software is not being built from scratch, some modification to the base application might be required to meet the needs of the new application.

Verification and validation:

Whether software is developed specially for the current application or not, careful testing will be needed to check that the proposed system meets its requirements.

Implementation/installation:

Some system development practitioners refer to the whole of the project after design as

'implementation' (that is, the implementation of the design) while others insist that the term refers to the installation of the system after the software has been developed. In this case it encompasses such things as setting up data files and system parameters, writing user manuals and training users of the new system.

Maintenance and support:

Once the system has been implemented there will be a continuing need for the correction of anyerrors that may have crept into the system and for extensions and improvements to the system. Maintenance and support activities may be seen as a series of minor software projects. Inmany environments, most software development is in fact maintenance.

4. Stepwise: an overview of project planning

Planning is the most difficult process in project management This chapter describes a framework of basic steps in project planning. Many different techniques can be used but this chapter tells the overview of the steps and activities in each step of project planning

A major step in project planning is to plan in outline first and then in more detail.

StepwiseProjectPlanning

Step 0: Select project

Step 1: Identify project scope and objectives

Step 2: Identify project infrastructure

Step 3: Analyze project characteristics

Step 4: Identify project products and activities

Step 5: Estimate effort for each activity.

Step 6: Identify activity risks.

Step 7: Allocate resources

Step 8 Review / Publicize pl\an

Step 9 &10: Execute plan / lower level of planning

Each step of project planning has different activities to perform. Following the description of each step with its activities

Step 0: Select project

This is called step 0 because in a way of project planning, it is outside the main project planning process. Feasibility study suggests us that the project is worthwhile or not.

Step 1: Identify project scope and objectives

The activities in this step ensure that all parties to the project agree on the objectivesand are committed to the success of the project.

Step 1.1: Identify objectives and practical measures of the effectiveness in meeting those objectives

Step 1.2: Establish project authority

Step 1.3: Stakeholders analysis – Identify all stakeholders in the project and their interest.

Step 1.4: Modify objectives in the light of stakeholder analysis.

Step 1.5: Establish method of communication

Step 2: Identify project infrastructure

Projects are rarely carried out in a vacuum. There is usually some kind of infrastructure into which the project must fit. Where the project manager are new to the organization , they must find out the precise nature of this infrastructure.

Step 2.1: Identify relationship between the project and strategic planning

Step 2.2: Identify installation standards and procedures.

Step 2.3: Identify project team organization.

Step 3: Analyze project characteristics.

The general purpose of this part of planning operation is to ensure that the appropriate methods are used for the project.

Step 3.1: Distinguish the project as either objective- product driven

Step 3.2: Analyze other project characteristics (including quality –based ones)

Step 3.3: Identify high level project risks

Step 3.4: Take into account user requirement concerning implementation.

Step 3.5: Select development methodology and life cycle approach.

Step 3.6: Review overall resources estimates

Step 4: Identify project products and activities

The more detailed planning of the individual activities now takes place. The longer term planning is broad and in outline, while the more immediate tasks are planned in some detail.

Step 4.1: Identify and describes project products (ordeliverables)

Step 4.2: Document generic product flows

Step 4.3: Record product instance

Step 4.4: produce ideal activity network

Step 4.5: Modify the ideal to take into account need for stages and checkpoints.

Step 5: Estimate effort for each activity.

Step 5.1: Carry out bottom-up estimates

Step 5.2: Revise plan to create controllable activities.

Step 6: Identify activity risks.

Step 6.1: Identify and quantify activity based risks

Step 6.2: Plan risk reduction and contingency measures where appropriate

Step 6.3: Adjust overall plans and estimates to take account of the risks

Step 7: Allocate resources

Step 7.1: Identify and allocate resources

Step 7.2: Revise plans and estimates to take into account resource constraints

Step 8: Review / Publicize plan

Step 8.1: Review quality aspects of the project plan.

Step 8.2: Document plans and obtain agreement.

Step 9 & 10: Execute plan / lower level of planning

Once the project is underway, plans will need to be drawn up in greater detail for each activity as it becomes due. Detailed and lower level of planning of the later stages will need to be delayed because more information will be available nearer the start of the stage.

Project planning is an iterative process. As the time approaches for the particular activities to be carried out they should be re-planned in more detail.

5. StepwiseProjectPlanning

- Step Wise project planning framework
- Preparation of a software project plan
- Planning and scheduling the activities in software project management
- Various approaches towards activity plan

- Various scheduling techniques such as sequencing and CPM

Aside – When to plan

- Planning is an on-going process of refinement
- Planning at different stages of the project has different emphases and purposes

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

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 availability
    - Resource allocation
    - Staff responsibility
    - Project Monitoring
    - Cash flow forecasting
    - Re-planning of the project towards the pre-defined goal

Other Objectives of Activity Planning

- Feasibility assessment
- Resource allocation
- Detailed costing

- Motivation
- Co-ordination

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
    - how each activity will be undertaken

Project Schedule in 4 Stages

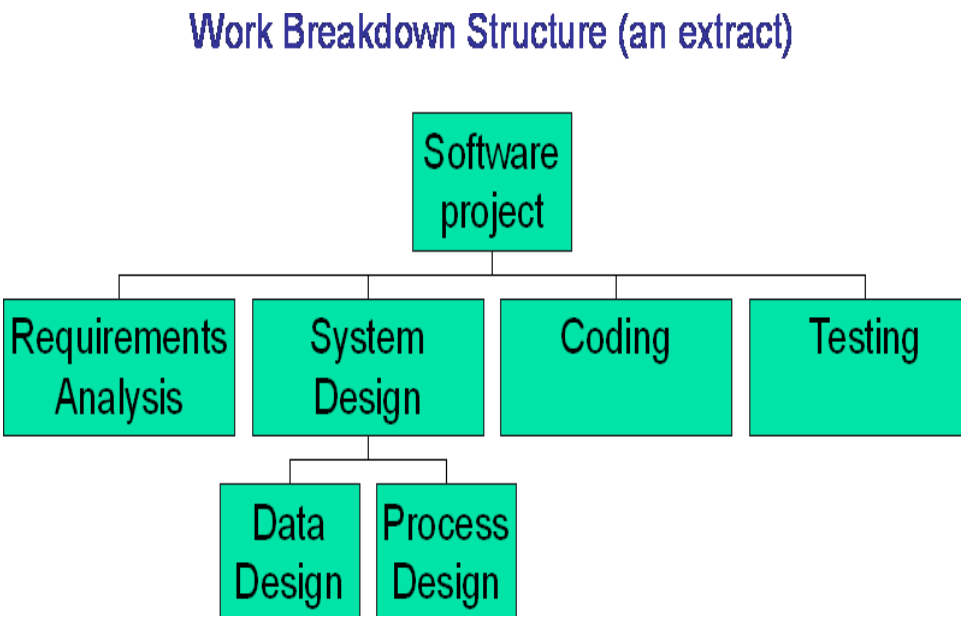
- Ideal Activity Plan
    - An activity plan without any constraints
- Risk consideration for each activity
- Resource consideration for whole project
- Schedule production and publication

Various Approaches towardsIdentifying Activity

- Activity-based approach
- Product-based approach
- Hybrid approach

Activity-based Approach

- Use *Work Breakdown Structure* (WBS) to generate a task list
- WBS involves
    - identifying the main tasks
    - break each main task down into subtasks
    - The subtasks can further be broken down into lower level tasks.



Advantages

- More likely to obtain a task catalogue that is complete and is composed of non-overlapping tasks
- WBS represents a structure that can be refined as the project proceeds
- The structure already suggests the dependencies among the activities

Disadvantage

- Very likely to miss some activities if an unstructured activity list is used

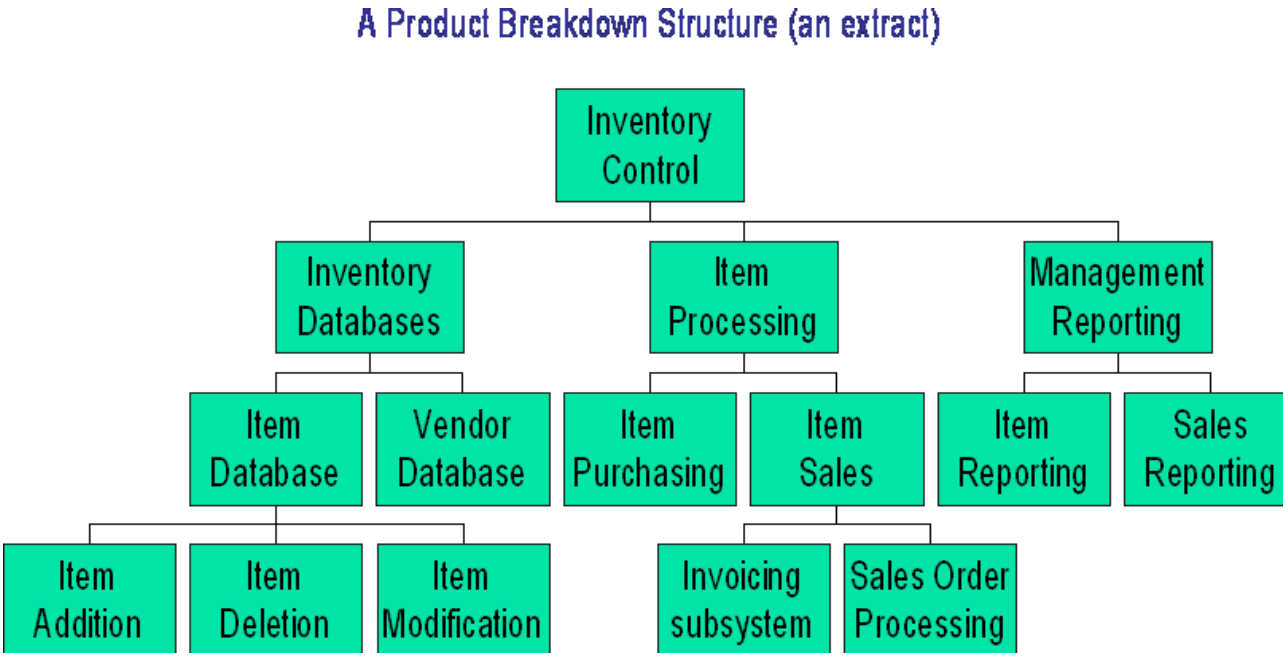
Product-based Approach

- 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'

Advantages

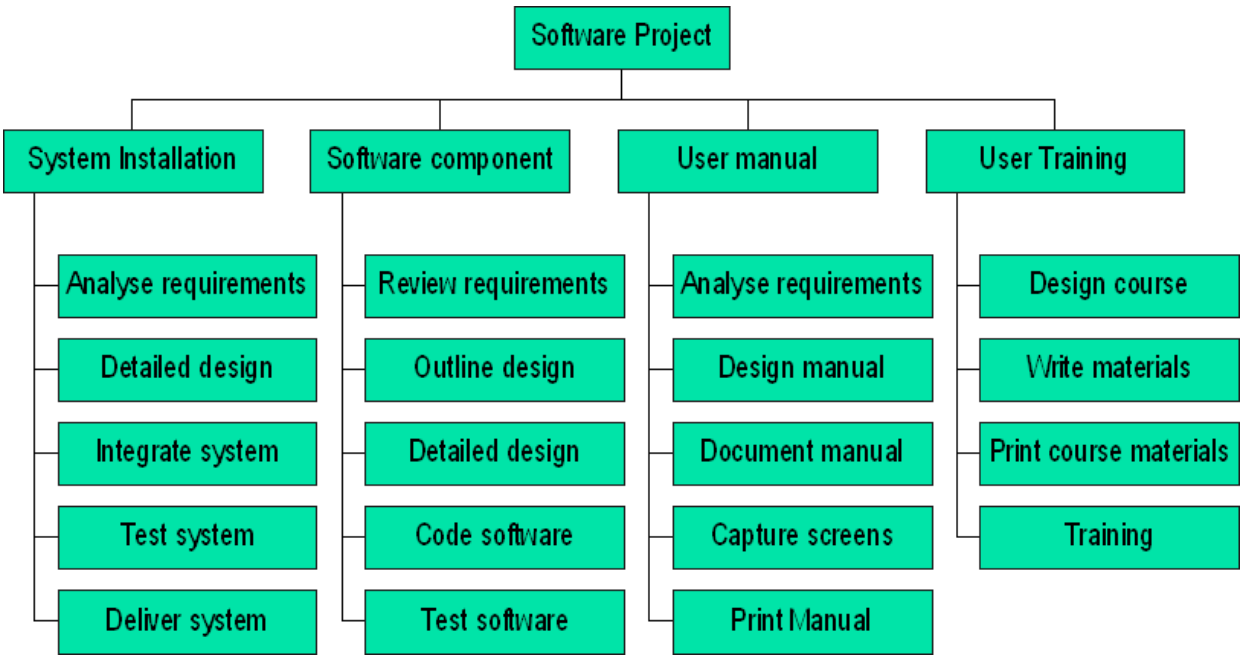
- Less likely to miss a product unexpectedly from a PBS

Product-based Approach – An example



Hybrid Approach

- A mix of the activity-based approach and the product-based approach
- More commonly used approach
- The WBS consists of
    - a list of the products of the project; and
    - a list of activities for each product



IBM in its MITP methodology suggests 5 levels

- Level 1: Project
- Level 2: Deliverables (software, manuals etc.)
- Level 3: Components
- Level 4: Work-packages
- Level 5: Tasks (individual responsibility)

Planning and Scheduling the Activities

- Once we have a project plan (or, project schedule), we need to schedule the activities in a project taking into account the resource constraints

Scheduling Techniques

- Simple sequencing
  - Suitable for small projects
- Critical Path Method (CPM)
  - Suitable for large software projects
  - The most commonly used "networking" technique

Simple sequencing

A simple sequencing of the tasks and the responsible personnel taken into account of the resources

- Easily presented in a simple bar chart

  see figure 6.6 in Hughes book

- Suitable for allocating individuals to particular tasks at an early stage

Critical Path Method (CPM)

Primary objectives:

Planning the project so that it can be completed as quickly as possible

Identifying those activities where their delays is likely to affect the overall project completion date

- Developed by Du Pont Chemical Company and published in 1958
- Capture the activities and their inter-relationships using a graph
  - Lines are used to represent the activities

- Nodes are used to represent the startand stop of activities

- Adding time dimension
  - The forward pass

    Calculate the earliest start dates of the activities

    To calculate the project completion date
  - The backward pass

    Calculate the latest start dates for activities

    Identify the critical path from the graph

- Identifying critical path and critical event
  - Critical event:an event that has zero *slack*
  - Critical path: a path joining those critical events

Example to construct a CPM

Id.	Activity Name	Duration (weeks)	Precedents
A	Hardware selection	7	
B	Software design	4	
C	Hardware Installation	6	A
D	Coding	4	B
E	Data Preparation	5	B
F	User Documentation	9	
G	User Training	5	E,F
H	System Installation	3	C,D

Activity Float

Time allowed for an activity to delay

3 different types:

- Total float (without affecting the completion of the project)

= latest start date – earliest start date

- Free float (without affecting the next activity)

= earliest start date of *next* activity – latest end date of *previous* activity

- Interfering float (= total float - free float)

Significance of critical path

During planning stage

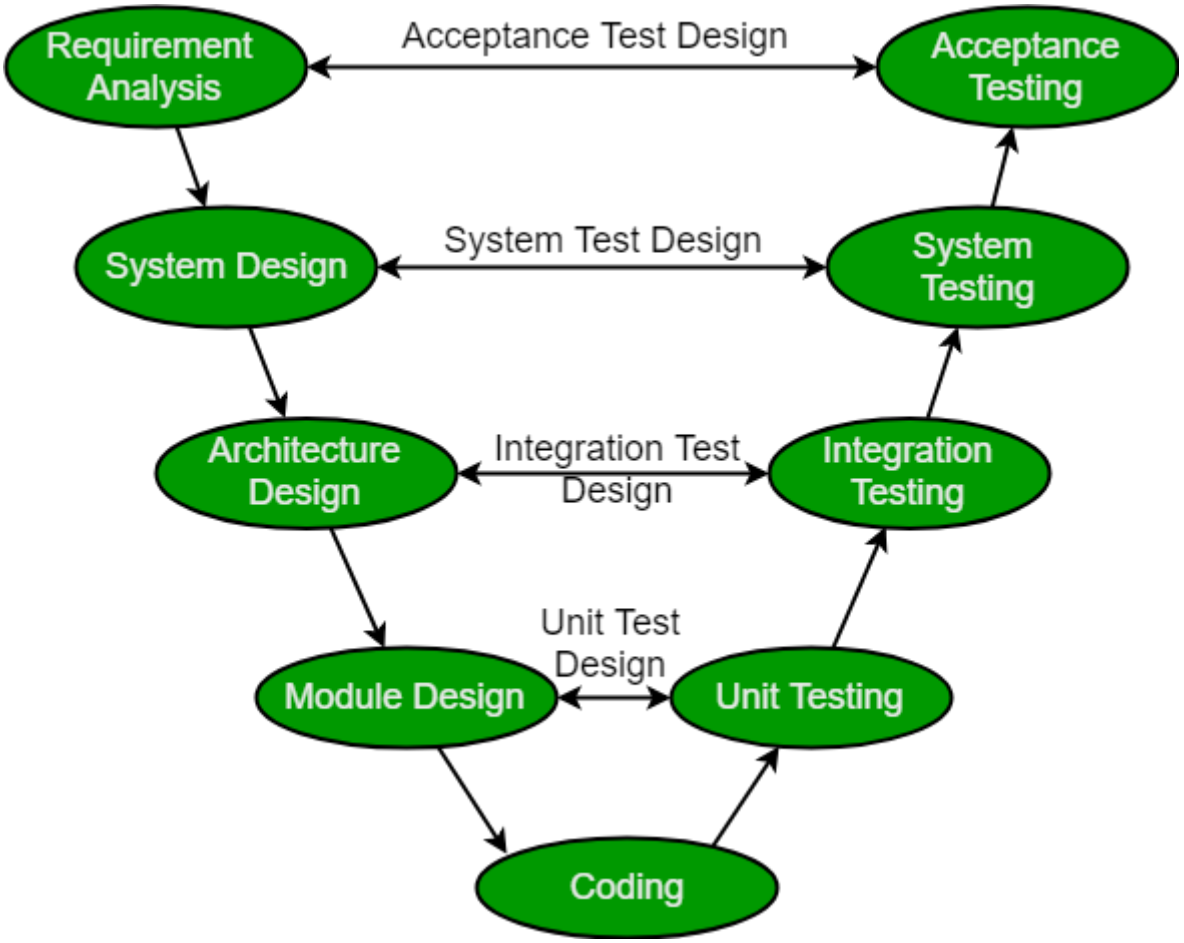
Shortening the critical path will reduce the overall project duration

During management stage

Pay more attention to those activities which fall in the critical path

V MODEL

The V-model is a type of SDLC model where process executes in a sequential manner in V-shape. It is also known as Verification and Validation model. It is based on the association of a testing phase for each corresponding development stage. Development of each step directly associated with the testing phase. The next phase starts only after completion of the previous phase i.e. for each development activity, there is a testing activity corresponding to it.



Verification: It involves static analysis technique (review) done without executing code. It is the process of evaluation of the product development phase to find whether specified requirements meet.

Validation: It involves dynamic analysis technique (functional, non-functional), testing done by executing code. Validation is the process to evaluate the software after the completion of the development phase to determine whether software meets the customer expectations and requirements.

So V-Model contains Verification phases on one side of the Validation phases on the other side. Verification and Validation phases are joined by coding phase in V-shape. Thus it is called V-Model.

Design Phase:

- **Requirement Analysis:** This phase contains detailed communication with the customer to understand their requirements and expectations. This stage is known as Requirement Gathering.
- **System Design:** This phase contains the system design and the complete hardware and communication setup for developing product.
- **Architectural Design:** System design is broken down further into modules taking up different functionalities. The data transfer and communication between the internal modules and with the outside world (other systems) is clearly understood.
- **Module Design:** In this phase the system breaks down into small modules. The detailed design of modules is specified, also known as Low-Level Design (LLD).

Testing Phases:

- **Unit Testing:** Unit Test Plans are developed during module design phase. These Unit Test Plans are executed to eliminate bugs at code or unit level.
- **Integration testing:** After completion of unit testing Integration testing is performed. In integration testing, the modules are integrated and the system is tested. Integration testing is performed on the Architecture design phase. This test verifies the communication of modules among themselves.
- **System Testing:** System testing test the complete application with its functionality, inter dependency, and communication. It tests the functional and non-functional requirements of the developed application.
- **User Acceptance Testing (UAT):** UAT is performed in a user environment that resembles the production environment. UAT verifies that the delivered system meets user's requirement and system is ready for use in real world.

Industrial Challange:
As the industry has evolved, the technologies have become more complex, increasingly faster, and forever changing, however, there remains a set of basic principles and concepts that are as applicable today as when IT was in its infancy.

- Accurately define and refine user requirements.
- Design and build an application according to the authorized user requirements.
- Validate that the application they had built adhered to the authorized business requirements.

Principles of V-Model:

- **Large to Small:** In V-Model, testing is done in a hierarchical perspective, For example, requirements identified by the project team, create High-Level Design, and Detailed Design phases of the project. As each of these phases is completed the requirements, they are defining become more and more refined and detailed.

- **Data/Process Integrity:** This principle states that the successful design of any project requires the incorporation and cohesion of both data and processes. Process elements must be identified at each and every requirements.
- **Scalability:** This principle states that the V-Model concept has the flexibility to accommodate any IT project irrespective of its size, complexity or duration.
- **Cross Referencing:** Direct correlation between requirements and corresponding testing activity is known as cross-referencing.
- **Tangible Documentation:** This principle states that every project needs to create a document. This documentation is required and applied by both the project development team and the support team. Documentation is used to maintaining the application once it is available in a production environment.

Why preferred?

- It is easy to manage due to the rigidity of the model. Each phase of V-Model has specific deliverables and a review process.
- Proactive defect tracking – that is defects are found at early stage.

When to use?

- Where requirements are clearly defined and fixed.
- The V-Model is used when ample technical resources are available with technical expertise.

Advantages:

- This is a highly disciplined model and Phases are completed one at a time.
- V-Model is used for small projects where project requirements are clear.
- Simple and easy to understand and use.
- This model focuses on verification and validation activities early in the life cycle thereby enhancing the probability of building an error-free and good quality product.
- It enables project management to track progress accurately.

Disadvantages:

- High risk and uncertainty.
- It is not a good for complex and object-oriented projects.
- It is not suitable for projects where requirements are not clear and contains high risk of changing.
- This model does not support iteration of phases.
- It does not easily handle concurrent events.

WATERFALL MODEL

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

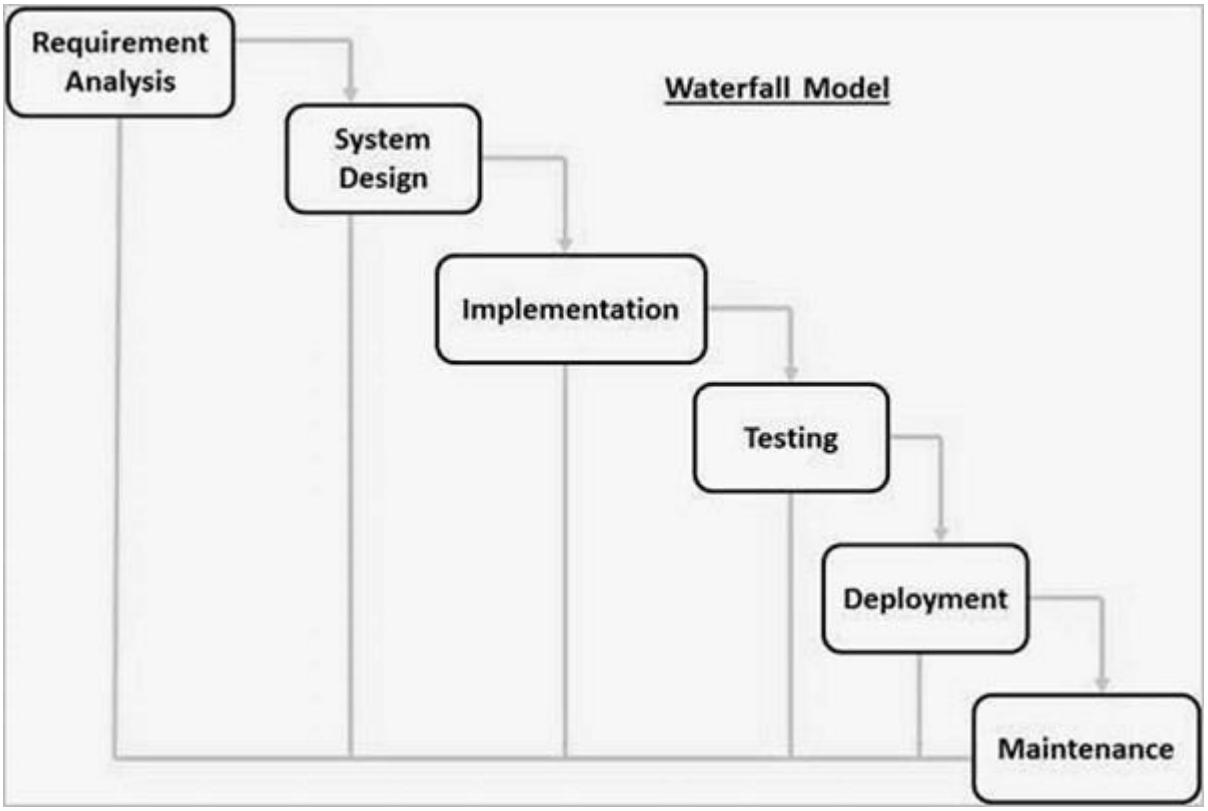
The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

Waterfall Model - Design

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

The following illustration is a representation of the different phases of the Waterfall Model.



The sequential phases in Waterfall model are –

- **Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation** − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model, phases do not overlap.

Waterfall Model - Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are −

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

Waterfall Model - Advantages

The advantages of waterfall development are that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

Some of the major advantages of the Waterfall Model are as follows −

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

Waterfall Model - Disadvantages

The disadvantage of waterfall development is that it does not allow much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

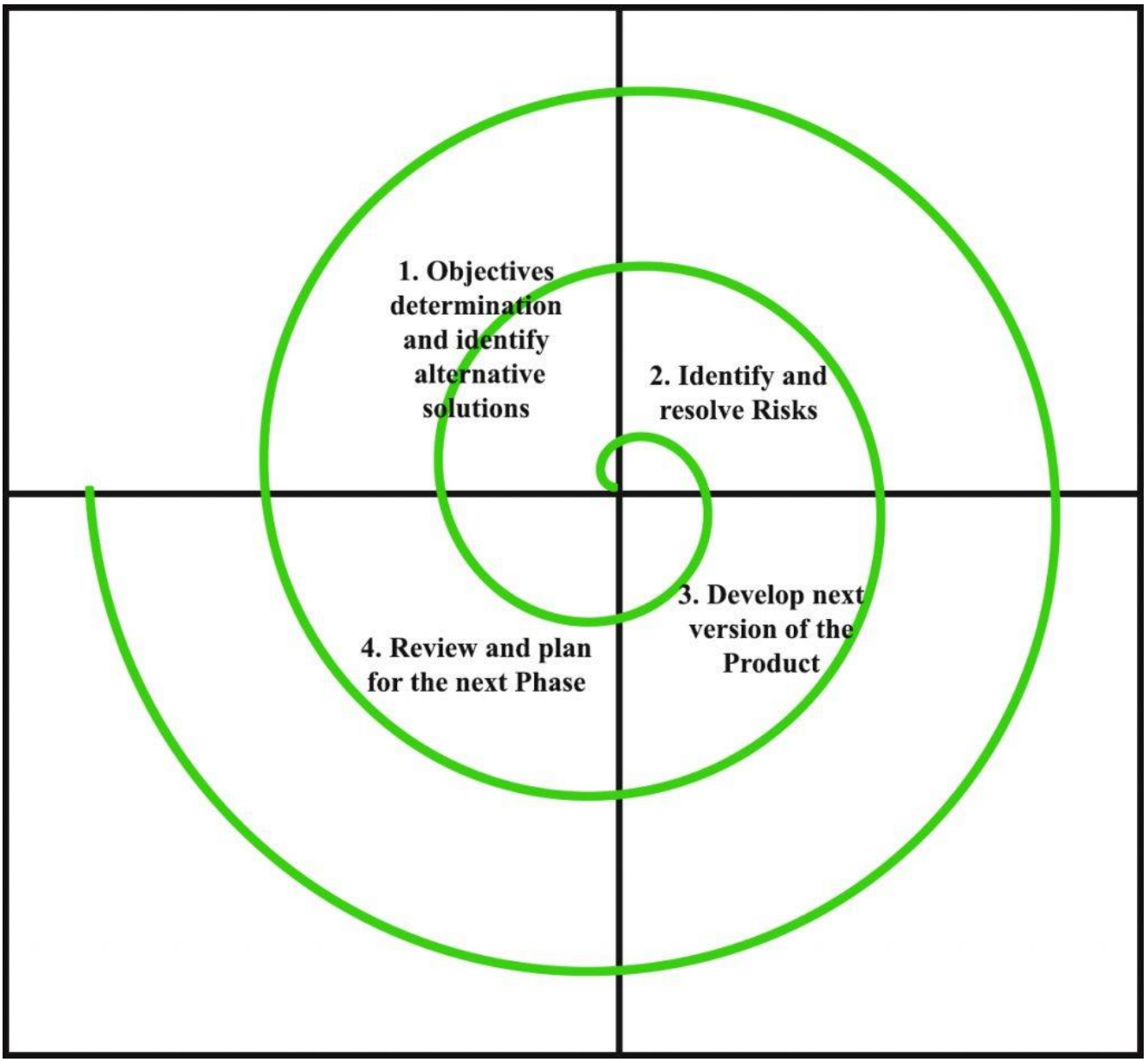
The major disadvantages of the Waterfall Model are as follows −

- No working software is produced until late during the life cycle.
- High amounts of risk and uncertainty.
- Not a good model for complex and object-oriented projects.
- Poor model for long and ongoing projects.
- Not suitable for the projects where requirements are at a moderate to high risk of changing. So, risk and uncertainty is high with this process model.
- It is difficult to measure progress within stages.
- Cannot accommodate changing requirements.
- Adjusting scope during the life cycle can end a project.
- Integration is done as a "big-bang. at the very end, which doesn't allow identifying any technological or business bottleneck or challenges early.

SPIRAL MODEL

Spiral model is one of the most important Software Development Life Cycle models, which provides support for **Risk Handling**. In its diagrammatic representation, it looks like a spiral with many loops. The exact number of loops of the spiral is unknown and can vary from project to project. **Each loop of the spiral is called a Phase of the software development process.** The exact number of phases needed to develop the product can be varied by the project manager depending upon the project risks. As the project manager dynamically determines the number of phases, so the project manager has an important role to develop a product using spiral model.
The Radius of the spiral at any point represents the expenses(cost) of the project so far, and the angular dimension represents the progress made so far in the current phase.

Below diagram shows the different phases of the Spiral Model:



Each phase of Spiral Model is divided into four quadrants as shown in the above figure. The functions of these four quadrants are discussed below-

1. **Objectives determination and identify alternative solutions:** Requirements are gathered from the customers and the objectives are identified, elaborated and analyzed at the start of every phase. Then alternative solutions possible for the phase are proposed in this quadrant.
2. **Identify and resolve Risks:** During the second quadrant all the possible solutions are evaluated to select the best possible solution. Then the risks associated with that solution is identified and the risks are resolved using the best possible strategy. At the end of this quadrant, Prototype is built for the best possible solution.

3. **Develop next version of the Product:** During the third quadrant, the identified features are developed and verified through testing. At the end of the third quadrant, the next version of the software is available.
4. **Review and plan for the next Phase:** In the fourth quadrant, the Customers evaluate the so far developed version of the software. In the end, planning for the next phase is started.

Risk Handling in Spiral Model

A risk is any adverse situation that might affect the successful completion of a software project. The most important feature of the spiral model is handling these unknown risks after the project has started. Such risk resolutions are easier done by developing a prototype. The spiral model supports coping up with risks by providing the scope to build a prototype at every phase of the software development.

Prototyping Model also support risk handling, but the risks must be identified completely before the start of the development work of the project. But in real life project risk may occur after the development work starts, in that case, we cannot use Prototyping Model. In each phase of the Spiral Model, the features of the product dated and analyzed and the risks at that point of time are identified and are resolved through prototyping. Thus, this model is much more flexible compared to other SDLC models.

Why Spiral Model is called Meta Model ?

The Spiral model is called as a Meta Model because it subsumes all the other SDLC models. For example, a single loop spiral actually represents the Iterative Waterfall Model. The spiral model incorporates the stepwise approach of the Classical Waterfall Model. The spiral model uses the approach of **Prototyping Model** by building a prototype at the start of each phase as a risk handling technique. Also, the spiral model can be considered as supporting the evolutionary model – the iterations along the spiral can be considered as evolutionary levels through which the complete system is built.

Advantages of Spiral Model:

Below are some of the advantages of the Spiral Model.

- **Risk Handling:** The projects with many unknown risks that occur as the development proceeds, in that case, Spiral Model is the best development model to follow due to the risk analysis and risk handling at every phase.
- **Good for large projects:** It is recommended to use the Spiral Model in large and complex projects.
- **Flexibility in Requirements:** Change requests in the Requirements at later phase can be incorporated accurately by using this model.
- **Customer Satisfaction:** Customer can see the development of the product at the early phase of the software development and thus, they habituated with the system by using it before completion of the total product.

Disdvantages of Spiral Model:

Below are some of the main disadvantages of the spiral model.

- **Complex:** The Spiral Model is much more complex than other SDLC models.
- **Expensive:** Spiral Model is not suitable for small projects as it is expensive.
- **Too much dependable on Risk Analysis:** The successful completion of the project is very much dependent on Risk Analysis. Without very highly experienced expertise, it is going to be a failure to develop a project using this model.
- **Difficulty in time management:** As the number of phases is unknown at the start of the project, so time estimation is very difficult.

RISK MANAGEMENT

Risk is inevitable in a business organization when undertaking projects. However, the project manager needs to ensure that risks are kept to a minimal. Risks can be mainly divided between two types, negative impact risk and positive impact risk.

Not all the time would project managers be facing negative impact risks as there are positive impact risks too. Once the risk has been identified, project managers need to come up with a mitigation plan or any other solution to counter attack the risk.

Project Risk Management

Managers can plan their strategy based on four steps of risk management which prevails in an organization. Following are the steps to manage risks effectively in an organization:

- Risk Identification
- Risk Quantification
- Risk Response
- Risk Monitoring and Control

Let's go through each of the step in project risk management:

Risk Identification

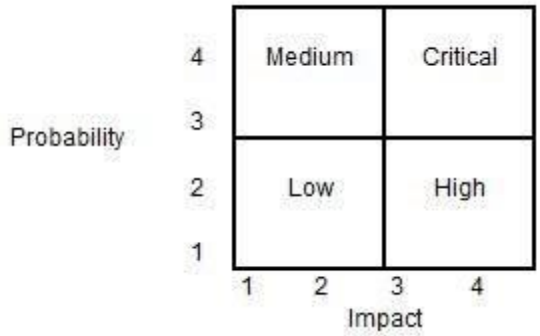
Managers face many difficulties when it comes to identifying and naming the risks that occur when undertaking projects. These risks could be resolved through structured or unstructured brainstorming or strategies. It's important to understand that risks pertaining to the project can only be handled by the project manager and other stakeholders of the project.

Risks, such as operational or business risks will be handled by the relevant teams. The risks that often impact a project are supplier risk, resource risk and budget risk. Supplier risk would refer to risks that can occur in case the supplier is not meeting the timeline to supply the resources required.

Resource risk occurs when the human resource used in the project is not enough or not skilled enough. Budget risk would refer to risks that can occur if the costs are more than what was budgeted.

Risk Quantification

Risks can be evaluated based on quantity. Project managers need to analyze the likely chances of a risk occurring with the help of a matrix.



Using the matrix, the project manager can categorize the risk into four categories as Low, Medium, High and Critical. The probability of occurrence and the impact on the project are the two parameters used for placing the risk in the matrix categories. As an example, if a risk occurrence is low (probability = 2) and it has the highest impact (impact = 4), the risk can be categorized as 'High'.

Risk Response

When it comes to risk management, it depends on the project manager to choose strategies that will reduce the risk to minimal. Project managers can choose between the four risk response strategies, which are outlined below.

- Risks can be avoided
- Pass on the risk
- Take corrective measures to reduce the impact of risks
- Acknowledge the risk

Risk Monitoring and Control

Risks can be monitored on a continuous basis to check if any change is made. New risks can be identified through the constant monitoring and assessing mechanisms.

Risk Management Process

Following are the considerations when it comes to risk management process:

- Each person involved in the process of planning needs to identify and understand the risks pertaining to the project.
- Once the team members have given their list of risks, the risks should be consolidated to a single list in order to remove the duplications.
- Assessing the probability and impact of the risks involved with the help of a matrix.
- Split the team into subgroups where each group will identify the triggers that lead to project risks.
- The teams need to come up with a contingency plan whereby to strategically eliminate the risks involved or identified.
- Plan the risk management process. Each person involved in the project is assigned a risk in which he/she looks out for any triggers and then finds a suitable solution for it.

Risk Register

Often project managers will compile a document, which outlines the risks involved and the strategies in place. This document is vital as it provides a huge deal of information.

Risk register will often consists of diagrams to aid the reader as to the types of risks that are dealt by the organization and the course of action taken. The risk register should be freely accessible for all the members of the project team.

Project Risk; an Opportunity or a Threat?

As mentioned above, risks contain two sides. It can be either viewed as a negative element or a positive element. Negative risks can be detrimental factors that can haphazard situations for a project.

Therefore, these should be curbed once identified. On the other hand, positive risks can bring about acknowledgements from both the customer and the management. All the risks need to be addressed by the project manager.

Conclusion

An organization will not be able to fully eliminate or eradicate risks. Every project engagement will have its own set of risks to be dealt with. A certain degree of risk will be involved when

undertaking a project.

The risk management process should not be compromised at any point, if ignored can lead to detrimental effects. The entire management team of the organization should be aware of the project risk management methodologies and techniques.

Enhanced education and frequent risk assessments are the best way to minimize the damage from risks

6. **Project Evaluation:**

A high level assessment of the project

- to see whether it is worthwhile to proceed with the project
- to see whether the project will fit in the strategic planning of the whole organization

Project Evaluation

Why

- Want to decide whether a project can proceed before it is too late
- Want to decide which of the several alternative projects has a better success rate, a higher turnover, a higher ...

Is it desirable to carry out the development and operation of the software system

Who

- Senior management
- Project manager/coordinator
- Team leader

When

- Usually at the beginning of the project

  e.g. Step 0 of Step Wise Framework

What

- Strategic assessment
- Technical assessment
- Economic assessment

How

- Cost-benefit analysis
- Cash flow forecasting
- Cost-benefit evaluation techniques

- Risk analysis

7. Strategic Assessment

- Used to assess whether a project fits in the *long-term goal* of the organization
- Usually carried out by senior management
- Needs a strategic plan that clearly defines the objectives of the organization
- Evaluates individual projects against the strategic plan or the overall business objectives

Programme management

suitable for projects developed for use in the organization

Portfolio management

suitable for project developed for other companies by software houses

SA – Programme Management

Individual projects as components of a programme within the organization

Programme as "a group of projects that are managed in a coordinated way to gain benefits that would not be possible were the projects to be managed independently

SA – Programme Management Issues

- Objectives
- How does the project contribute to the *long-term goal* of the organization?
- Will the product increase the market share? By how much?
- IS plan
- Does the product fit into the overall IS plan?
- How does the product relate to other existing systems?
- Organization structure
- How does the product affect the existing organizational structure? the existing workflow? the overall business model?

- MIS
  - What information does the product provide?
  - To whom is the information provided?

- How does the product relate to other existing MISs?

- Personnel
  - What are the staff implications?
  - What are the impacts on the overall policy on staff development?
- Image
  - How does the product affect the image of the organization?

SA – Portfolio Management

- suitable for product developed by a software company for an organization
- may need to assess the product for the client organization

  Programme management issues apply
- need to carry out strategic assessment for the providing software company
- *Long-term goal* of the software company
- The effects of the project on the portfolio of the company (synergies and conflicts)
- Any added-value to the overall portfolio of the company

8. **Technical Assessment**

- Functionality against hardware and software
- Thestrategic IS plan of the organization
- any constraints imposed by the IS plan

Economic Assessment

Why?

- Consider whether the project is the best among other options
- Prioritise the projects so that the resources can be allocated effectively if several projects are underway

How?

- Cost-benefit analysis
- Cash flow forecasting
- Various cost-benefit evaluation techniques

  NPV and IRR

EA – Cost-benefit Analysis

A standard way to assess the economic benefits

Two steps

Identify and estimate all the costs and benefits of carrying out the project

Express the costs and benefits in a common unit for easy comparison (e.g. $)

Costs

- Development costs
- Setup costs
- Operational costs

Benefits

- Direct benefits
- Assessable indirect benefits
- Intangible benefits

EA – Cash Flow Forecasting

What?

- Estimation of the cash flow over time

Why?

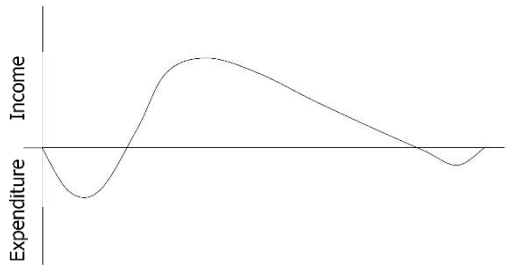
- An excess of estimated benefits over the estimated costs is not sufficient
- Need detailed estimation of benefits and costs versus time

What?

- Estimation of the cash flow over time

Why?

- An excess of estimated benefits over the estimated costs is not sufficient
- Need detailed estimation of benefits and costs versus time



Need to forecast the expenditure and the income

Accurate forecast is not easy

Need to revise the forecast from time to time

Cost-benefit Evaluation Techniques

- **Net profit**

$$= \text{Total income} - \text{Total costs}$$

- **Payback period**

$$= \text{Time taken to break even}$$

- **Return on Investment (ROI)**

$$= \frac{\text{average annual profit}}{\text{total investment}} \times 100\%$$

Cost-benefit Evaluation Techniques – NPV

Net present value (NPV)

- It is the sum of the present values of all future amounts.
- *Present value* is the value which a future amount is worth at present
- It takes into account the profitability of a project and the timing of the cash flows

Let *n* be the number of year and *r* be the discount rate, the present value (PV) is given by

$$\text{PV} = \frac{\text{value in year } n}{(1+r)^n}$$

- **Issues in NPV**
  - Choosingan appropriate discount rate is difficult

    - Ensuring that the rankings of projects are not sensitive to small changes in discount rate

- **Guidelines:**
    - Use the standard rate prescribed by the organization
    - Use interest rate + premium rate
    - Use a target rate of return
    - Rank the projects using various discount rates
- **Disadvantage**
    - May not be directly comparable with earnings from other investments or the costs of borrowing capital
- **Internal Rate of Return (IRR)**
    - The percentage discount rate that would produce a NPV of zero
    - A relative measure
- **Advantages**
    - Convenient
        - Directly comparable with rate of return on other projects and with interest rates
    - Useful
        - Dismiss a project due to its small IRR value
        - Indicate further precise evaluation of a project
    - Supported by MS Excel and Lotus 1-2-3

Estimation

- Why? – to define the project budget and to 'refine' the product to realize the budget
- Who? – the manager
- What? – size and cost
- When? – always
- How? – techniques and models

Issues related to Estimation

- Difficult to make accurate estimation
- Better to have previous data and analyze the actual values against their estimates so that you know how accurate you are
- Even better to have previous data of the whole organization so that you know how accurate the estimation method, if any, used within the organization

Positive Attitude Towards Estimation

- Use your estimation as a guide to manage your project
- From time to time, you need to revise your estimation based on the current status of the project

Estimation Approaches

- Expert judgement
    - Ask the knowledgeable experts
- Estimation by analogy
    - Use the data of a similar and completed project
- Pricing to win
    - Use the price that is low enough to win the contract
- Top-down
    - An overall estimate is determined and then broken down into each component task
- Bottom-up
    - The estimates of each component task are aggregated to form the overall estimate
- Algorithmic model

Estimation is based on the characteristics of the product and the development environment

Size Estimation

- Problems related to size estimation
- Size Estimation Model

Function Point Analysis (FPA)

Problems related to size estimation

- Nature of software
- Novel application of software
- Fast changing technology
- Lack of homogeneity of project experience
- Subjective nature of estimation
- Political implications within the organization

Cost-Benefit Analysis

Cost/benefit analysis, comparing

– Expected costs

– Expected benefits

Issues

– Estimating costs

– Estimating benefits

Use of financial models to evaluate

Cost-Benefit Analysis-Two Steps

Identifying and estimating all of the costs and benefits of carrying out the project and operating the delivered application

Expressing the costs and benefits in common units

Cost-Benefit Analysis-Cost Estimation

Estimate costs to compare with benefits/other investment options

Overall estimation based on

– Estimation of required activities (structure)

– Estimation for each activity

– Estimation of installation/setup cost

– Estimation of operational cost

Difficult, as a lot of these are`estimates';

estimation errors cascade

Cost-Benefit Analysis-Cost Category

Development costs

Setup costs

Operational costs

Cost-Benefit Analysis-Development Costs

- Salaries (base, incentives, and bonuses)
- Equipment for development
  - Hardware
  - Software

Cost-Benefit Analysis-Setup Cost

- Hardware and software infrastructure
- Recruitment/staff training
- Installation and conversion costs

Cost-Benefit Analysis-Operational Costs

Costs of operating the system once it has been installed

- Support costs
- Hosting costs
- Licensing costs
- Maintenance costs
- Backup costs

Cost-Benefit Analysis-Benefit Estimation

Estimate benefits of new system based on– Estimation of cost savings and money generation when deployed– Value of information obtained for objective driven project

- Value of intangibles

Cost Benefits Analysis-Benefits Types

- Direct benefits
- Indirect benefits
- Intangible benefits

Cost Benefits Analysis-Direct Benefits

Directly accountable to new system

– Cost savings (e.g., less staff, less paper, quicker turnaround)

– Money generation (e.g., new revenue stream, new markets)

Measurable after system is operational

Have to be estimated for cost/benefit analysis

Cost Benefits Analysis -Intangible Benefits

Positive side effects of new system

External system (e.g., increase branding, entry to new markets)

Internal system (increased interest in job for users, enabler for other systems)

Often very specific to a project; not measurable even after a system is operational

Part of strategic decision rather than cost/benefit analysis

9. **Cash Flow Forecasting**

Indicates when expenditure and income will take place

Cash Flow Analysis

Typically there are outgoing payments initially and then incoming payments

There might be additional costs at the end of the project life

Cash flow considerations

– Is initial funding for the project available?

– Is timing of incoming/outgoing cash flow in line with financial plans?

– If cash flow is critical, forecasting should be done quarterly or monthly

Risky/expensive projects might be funded using venture capital

10. **Cost-Benefit Evaluation-Techniques**

Costs and benefits have to be expressed using the same scale to be comparable

Usually expressed in payments at certain times (cash flow table)

Payments at different points in time are not comparable based only on the amount

Time of payment should be considered

Techniques

– Net profit

– Payback period

– Return on investment

– Net present value

– Internal rate of return

Cost-Benefit Evaluation Techniques -Net Profit

Difference between total cost and total income

Pros: Easy to calculate

Cons

– Does not show profit relative to size investment (e.g., consider Project 2)

– Does not consider timing of payments (e.g., compare Projects 1 and 3)

Not very useful other than for "back of envelope" evaluations

Cost-Benefit Evaluation Techniques -Payback Period

Time taken to break even

Pros

– Easy to calculate

– Gives some idea of cash flow impact

Cons: Ignores overall profitability

Not very useful by itself, but a good measure for cash flow impact

Costs-Benefit Evaluation Techniques-Return On Investment

Also known as the accounting rate of return (ARR)

Provides a way of comparing the net profitability to the investment required

The common formula– ROI = (average annual profit/total investment) X 100

Cost-Benefit Evaluation Techniques -Return On Investment

Pros: Easy to calculate

Cons

– Does not consider the timing of payments

– Misleading: does not consider bank interest rates

Not very useful other than for "back of envelope" evaluations

Cost-Benefit Evaluation Techniques-Net Present Value

A project evaluation technique that takes into account the profitability of a project and the timing of the cash flows that are produced

Sum of all incoming and outgoing payments, discounted using an interest rate, to a fixed point in time (the present)

Cost-Benefit Evaluation Techniques-Net Present Value

Present value = (value in year t)/(1+r)^t

– r is the discount rate

– t is the number of years into the future that the cash flow occurs

– (1+r)^t is known as discount factor

In the case of 10% rate and one year

– Discount factor = 1/(1+0.10) = 0.9091

In the case of 10% rate and two years

– Discount factor = 1/(1.10 x 1.10) = 0.8294

Pros

– Takes into account profitability

– Considers timing of payments

– Considers economic situation through discount rate

Cons: Discount rate can be difficult to choose

Standard measure to compare different options

Cost-Benefit Evaluation Techniques -Internal Rate of Return

Internal rate of return (IRR) is the discount rate that would produce an NPV of 0 for the project

Can be used to compare different investment opportunities

There is a Microsoft Excel function to calculate IRR

Pros: Calculates figure which is easily comparable to interest rates

Cons: Difficult to calculate (iterative)

Standard way to compare projects

11. **Definition of Risk**

A risk is a potential problem – it might happen and it might not

Conceptual definition of risk

- Risk concerns future happenings
- Risk involves change in mind, opinion, actions, places, etc.
- Risk involves choice and the uncertainty that choice entails

Two characteristics of risk

- Uncertainty – the risk may or may not happen, that is, there are no 100% risks (those, instead, are called constraints)
- Loss – the risk becomes a reality and unwanted consequences or losses occur

Risk Categorization – Approach

Project risks

They threaten the project plan

If they become real, it is likely that the project schedule will slip and that costs will increase

Technical risks

They threaten the quality and timeliness of the software to be produced

If they become real, implementation may become difficult or impossible

Business risks

They threaten the viability of the software to be built

If they become real, they jeopardize the project or the product

Sub-categories of Business risks

Market risk – building an excellent product or system that no one really wants

Strategic risk – building a product that no longer fits into the overall business strategy for the company

Sales risk – building a product that the sales force doesn't understand how to sell

Management risk – losing the support of senior management due to a change in focus or a change in people

Budget risk – losing budgetary or personnel commitment

Known risks

Those risks that can be uncovered after careful evaluation of the project plan, the business and technical environment in which the project is being developed, and other reliable information sources (e.g., unrealistic delivery date)

Predictable risks

Those risks that are extrapolated from past project experience (e.g., past turnover)

Unpredictable risks

Those risks that can and do occur, but are extremely difficult to identify in advance

Reactive vs. Proactive Risk Strategies

Reactive risk strategies

- "Don't worry, I'll think of something"
- The majority of software teams and managers rely on this approach
- Nothing is done about risks until something goes wrong
    - The team then flies into action in an attempt to correct the problem rapidly (fire fighting)
- Crisis management is the choice of management techniques

Proactive risk strategies

- Steps for risk management are followed (see next slide)
- Primary objective is to avoid risk and to have a contingency plan in place to handle unavoidable risks in a controlled and effective manner

Steps for Risk Management

1) Identify possible risks; recognize what can go wrong
2) Analyze each risk to estimate the probability that it will occur and the impact (i.e., damage) that it will do if it does occur
3) Rank the risks by probability and impact
   - Impact may be negligible, marginal, critical, and catastrophic

4) Develop a contingency plan to manage those risks having high probability and high impact

Risk Identification

- Risk identification is a systematic attempt to specify threats to the project plan
- By identifying known and predictable risks, the project manager takes a first step toward avoiding them when possible and controlling them when necessary
- Generic risks
  - Risks that are a potential threat to every software project
- Product-specific risks
  - Risks that can be identified only by those a with a clear understanding of the technology, the people, and the environment that is specific to the software that is to be built
  - This requires examination of the project plan and the statement of scope
  - "What special characteristics of this product may threaten our project plan?"

Risk Item Checklist

- Used as one way to identify risks
- Focuses on known and predictable risks in specific subcategories (see next slide)
- Can be organized in several ways
  - A list of characteristics relevant to each risk subcategory
  - Questionnaire that leads to an estimate on the impact of each risk
  - A list containing a set of risk component and drivers and their probability of occurrence

Known and Predictable Risk Categories

- Product size – risks associated with overall size of the software to be built
- Business impact – risks associated with constraints imposed by management or the marketplace
- Customer characteristics – risks associated with sophistication of the customer and the developer's ability to communicate with the customer in a timely manner
- Process definition – risks associated with the degree to which the software process has been defined and is followed

- Development environment – risks associated with availability and quality of the tools to be used to build the project
- Technology to be built – risks associated with complexity of the system to be built and the "newness" of the technology in the system
- Staff size and experience – risks associated with overall technical and project experience of the software engineers who will do the work

Questionnaire on Project Risk

1) Have top software and customer managers formally committed to support the project?
2) Are end-users enthusiastically committed to the project and the system/product to be built?
3) Are requirements fully understood by the software engineering team and its customers?
4) Have customers been involved fully in the definition of requirements?
5) Do end-users have realistic expectations?
6) Is the project scope stable?
7) Does the software engineering team have the right mix of skills?
8) Are project requirements stable?
9) Does the project team have experience with the technology to be implemented?
10) Is the number of people on the project team adequate to do the job?
11) Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built?

Risk Components and Drivers

- The project manager identifies the risk drivers that affect the following risk components
  - Performance risk - the degree of uncertainty that the product will meet its requirements and be fit for its intended use
  - Cost risk - the degree of uncertainty that the project budget will be maintained
  - Support risk - the degree of uncertainty that the resultant software will be easy to correct, adapt, and enhance
  - Schedule risk - the degree of uncertainty that the project schedule will be maintained and that the product will be delivered on time
- The impact of each risk driver on the risk component is divided into one of four impact levels

- Negligible, marginal, critical, and catastrophic

- Risk drivers can be assessed as impossible, improbable, probable, and frequent

Risk Projection (Estimation)

- Risk projection (or estimation) attempts to rate each risk in two ways
  - The probability that the risk is real
  - The consequence of the problems associated with the risk, should it occur
- The project planner, managers, and technical staff perform four risk projection steps (see next slide)
- The intent of these steps is to consider risks in a manner that leads to prioritization
- Be prioritizing risks, the software team can allocate limited resources where they will have the most impact

Risk Projection/Estimation Steps

1) Establish a scale that reflects the perceived likelihood of a risk (e.g., 1-low, 10-high)
2) Delineate the consequences of the risk
3) Estimate the impact of the risk on the project and product
4) Note the overall accuracy of the risk projection so that there will be no misunderstandings

Contents of a Risk Table

- A risk table provides a project manager with a simple technique for risk projection
- It consists of five columns
  - Risk Summary – short description of the risk
  - Risk Category – one of seven risk categories (slide 12)
  - Probability – estimation of risk occurrence based on group input
  - Impact – (1) catastrophic (2) critical (3) marginal (4) negligible
  - RMMM – Pointer to a paragraph in the Risk Mitigation, Monitoring, and Management Plan

Developing a Risk Table

- List all risks in the first column (by way of the help of the risk item checklists)

- Mark the category of each risk
- Estimate the probability of each risk occurring
- Assess the impact of each risk based on an averaging of the four risk components to determine an overall impact value (See next slide)
- Sort the rows by probability and impact in descending order
- Draw a horizontal cutoff line in the table that indicates the risks that will be given further attention

Assessing Risk Impact

- Three factors affect the consequences that are likely if a risk does occur
  - Its nature – This indicates the problems that are likely if the risk occurs
  - Its scope – This combines the severity of the risk (how serious was it) with its overall distribution (how much was affected)
  - Its timing – This considers when and for how long the impact will be felt
- The overall risk exposure formula is RE = P x C
  - P = the probability of occurrence for a risk
  - C = the cost to the project should the risk actually occur
- Example
  - P = 80% probability that 18 of 60 software components will have to be developed
  - C = Total cost of developing 18 components is $25,000
  - RE = .80 x $25,000 = $20,000

Risk Mitigation, Monitoring, and Management

- An effective strategy for dealing with risk must consider <u>three</u> issues (Note: these are not mutually exclusive)
  - Risk mitigation (i.e., avoidance)
  - Risk monitoring
  - Risk management and contingency planning
- <u>Risk mitigation</u> (avoidance<u>)</u> is the primary strategy and is achieved through a plan

Example: Risk of high staff turnover

- Seven Principles of Risk ManagementMaintain a global perspective

  - View software risks within the context of a system and the business problem that is is intended to solve

- Take a forward-looking view
  - Think about risks that may arise in the future; establish contingency plans
- Encourage open communication
  - Encourage all stakeholders and users to point out risks at any time
- Integrate risk management
  - Integrate the consideration of risk into the software process
- Emphasize a continuous process of risk management
  - Modify identified risks as more becomes known and add new risks as better insight is achieved
- Develop a shared product vision
  - A shared vision by all stakeholders facilitates better risk identification and assessment
- Encourage teamwork when managing risk
  - Pool the skills and experience of all stakeholders when conducting risk management activities.

12. Objectives of activity planning

- Feasibility assessment-Whether project can be finished within specified time scales
- Resource allocation
- Detailed costing-Cost?
- Motivation
- Co-ordination

13. Project schedules

- Steps
  - Ideal activity plan
  - Activity risk analysis
  - Resource allocation
  - Schedule production

Projects and activities

- Defining activities
- Identifying activities

Identifying activities

- The activity based approach
- The product based approach
- The hybrid approach

14. Sequencing and scheduling activities

- Project plan-bar chart
- SSADM

- Take into account availability of staff
- Way of allocation

15. Network-Planning Models

- A project is made up of a sequence of activities that form a network representing a project.
- The path taking longest time through this network of activities is called the "*critical path*."
- The critical path provides a wide range of scheduling information useful in managing a project.
- Critical Path Method (CPM) helps to identify the critical *path(s)* in the project networks.
- CPM with a Single Time Estimate
    - Used when activity times are known with certainty.
    - Used to determine timing estimates for the project, each activity in the project, and slack time for activities.
- CPM with Three Activity Time Estimates (a.k.a. PERT)
    - Used when activity times are uncertain.
    - Used to obtain the same information as the Single Time Estimate model *andprobability information*.
- Time-Cost Models
    - Used when trade-off information cost is a major consideration in planning.
    - Used to determine the least cost in reducing total project time.

Example:. CPM with Single Time Estimate

Consider the following consulting project

Activity	Designation	Immed. Pred.	Time (Weeks)
Assess customer's needs	A	None	2
Write and submit proposal	B	A	1
Obtain approval	C	B	1
Develop service vision and goals	D	C	2
Train employees	E	C	5
Quality improvement pilot groups	F	D, E	5
Write assessment report	G	F	1

Develop a critical path diagram (network) and determine the duration of the critical path and Slacktimes for all activities

1. Draw the network
2. Compute early starts and early finish times (forward pass)
3. Compute late starts and late finish times (backward pass)
4. Compute Slack (LS-ES) per activity and Critical Path(s)

Example2. CPM with Three Activity Time Estimates

Develop a critical path diagram (network) and determine the duration of the critical path and Slacktimes for all activities

1. Draw the network
2. Compute early starts and early finish times (forward pass)
3. Compute late starts and late finish times (backward pass)
4. Compute Slack (LS-ES) per activity and Critical Path(s)

What is the probability of finishing this project in less than 53 days?

What is the probability that the project duration will exceed 56 days?

Time-Cost Models

- Sometimes it is possible to "crash" (expedite) some activities thus reducing the overall completion time for the entire project.
- Crashing an activity implies spending additional funds (e.g., overtime costs, hiring more workers, and so on) to get the task done earlier.

- On many occasions reducing the project completion time that in turn reduces the fixed cost outlays can generate substantial savings.

1. Draw the CPM network, identify the CP
2. Identify the least cost activity(ies) on the critical path(s)
3. Shorten the project completion time (CP) at the least cost

Repeat until no more crashing is possible (or cost exceeds the benefits)

- Assume fixed costs = $1,000 day.
- Find the optimum time-cost schedule.

CPM Assumptions/Limitations

- Project activities can be identified as entities. (There is a clear beginning and ending point for each activity.)
- Project activity sequence relationships can be specified and networked.
- Project control should focus on the critical path.
- The activity times follow the beta distribution, with the variance of the project assumed to equal the sum of the variances along the critical path. Project control should focus on the critical path.

MS Project

- MS Project is very popular and inexpensive project management software.
- It is constantly improved (upgraded).
- Many independent software firms have developed "add-ons" to further improve or help users (managers) take full advantage of its capabilities.
- For example, probabilistic analysis (PERT approach) is not directly available in MS Project.
    - CAUTION: "PERT" in MS Project refers to the AON network representation, and simplistic project duration calculations done by using either optimistic or most likely or pessimistic time estimates for all activities.
- Risk+ developed by C/S Solutions "is a comprehensive risk analysis tool that integrates seamlessly with Microsoft® Project to quantify the cost and schedule uncertainty associated with your project plans."

Reliable Construction Company Project

- This is a mini case/group exercise.

- The Reliable Construction Company has just made the winning bid of $5.4 million to construct a new plant for a major manufacturer.
- The contract includes the following provisions:
    - A penalty of $300,000 if Reliable has not completed construction within 47 weeks.
    - A bonus of $150,000 if Reliable has completed the plant within 40 weeks.

Questions:

1. How can the project be displayed graphically to better visualize the activities?
2. What is the total time required to complete the project if no delays occur?
3. When do the individual activities need to start and finish?
4. What are the critical bottleneck activities?
5. For other activities, how much delay can be tolerated?
6. What is the probability the project can be completed in 47 weeks?
7. Is it worth to expedite the activities to finish the project in 40 weeks?
    - Assume activities with 7 or more weeks can be shortened by two weeks and the rest can be reduced by only one week.
    - For simplicity assume that cost per week to expedite any activity is $30,000.

Three Time Estimates for the Project

Activity	**optimistic**	**Most likely**	**pessimistic**
A	**1**	**2**	**3**
B	**2**	**3.5**	**8**
C	**6**	**9**	**18**
D	**4**	**5.5**	**10**
E	1	4.5	5

F	**4**	**4**	**10**
G	**5**	**6.5**	**11**
H	**5**	**8**	**17**
I	**3**	**7.5**	**9**
J	**3**	**9**	**9**
K	**4**	**4**	**4**
L	**1**	**5.5**	**7**
M	**1**	**2**	**3**
N	**5**	**5.5**	**9**

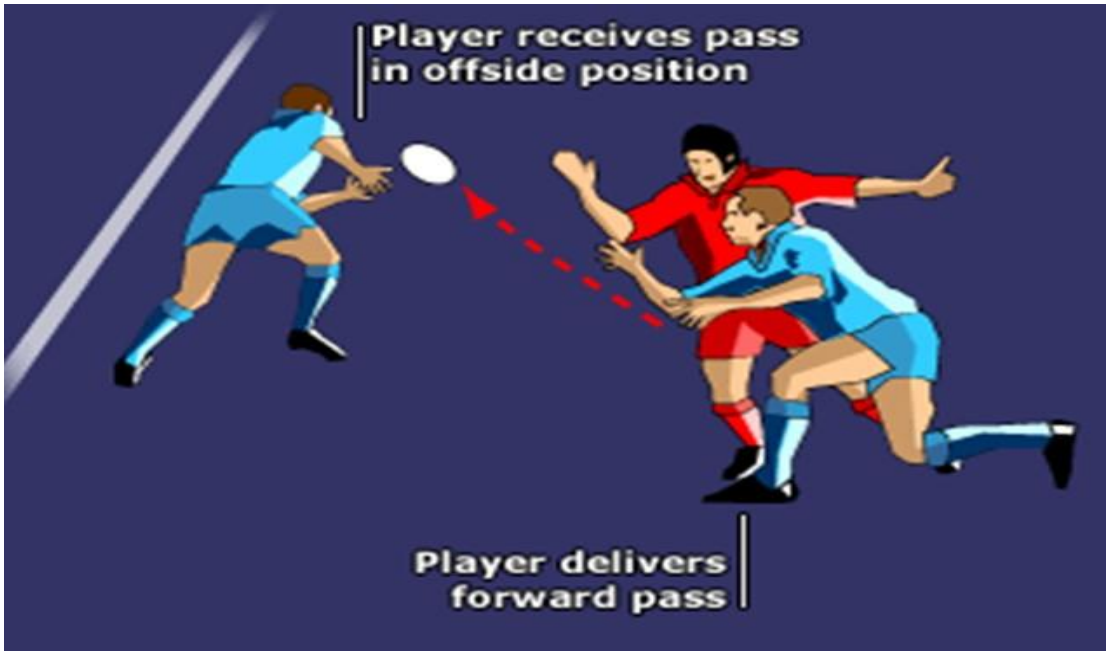
16. **Forward and backward pass**

Why Network Diagrams?

- Splits up the decision making process into
  - Method/logic - the order in which tasks have to be completed
  - Time – estimates for the time to completion can be added to each task
  - Resources – these can be added and then analysis carried out

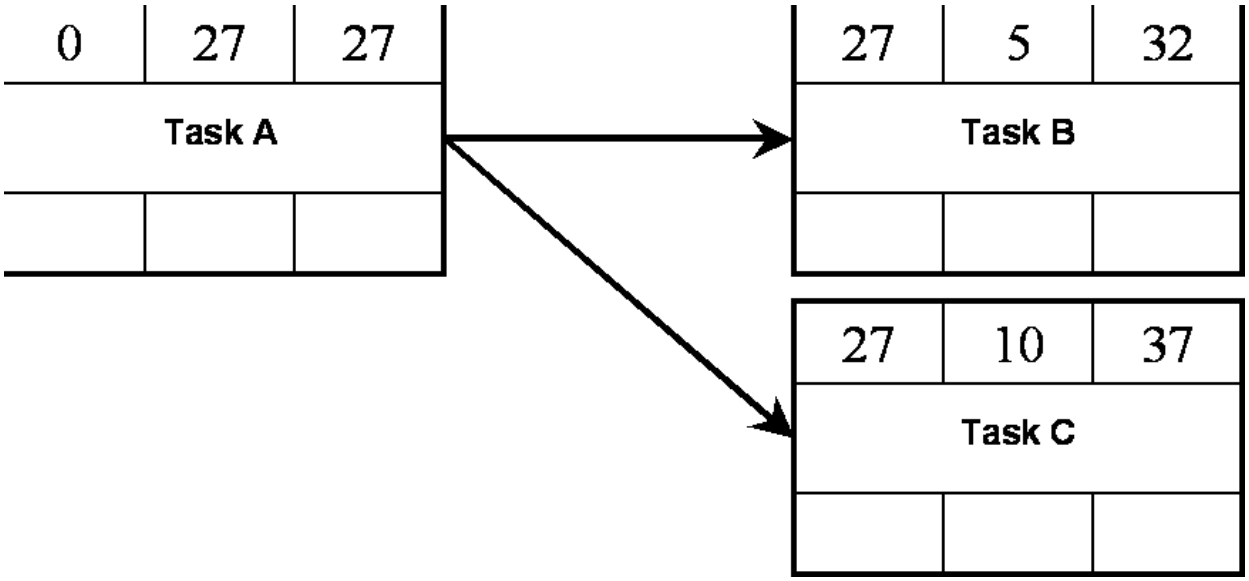
Two Parts to the Analysis

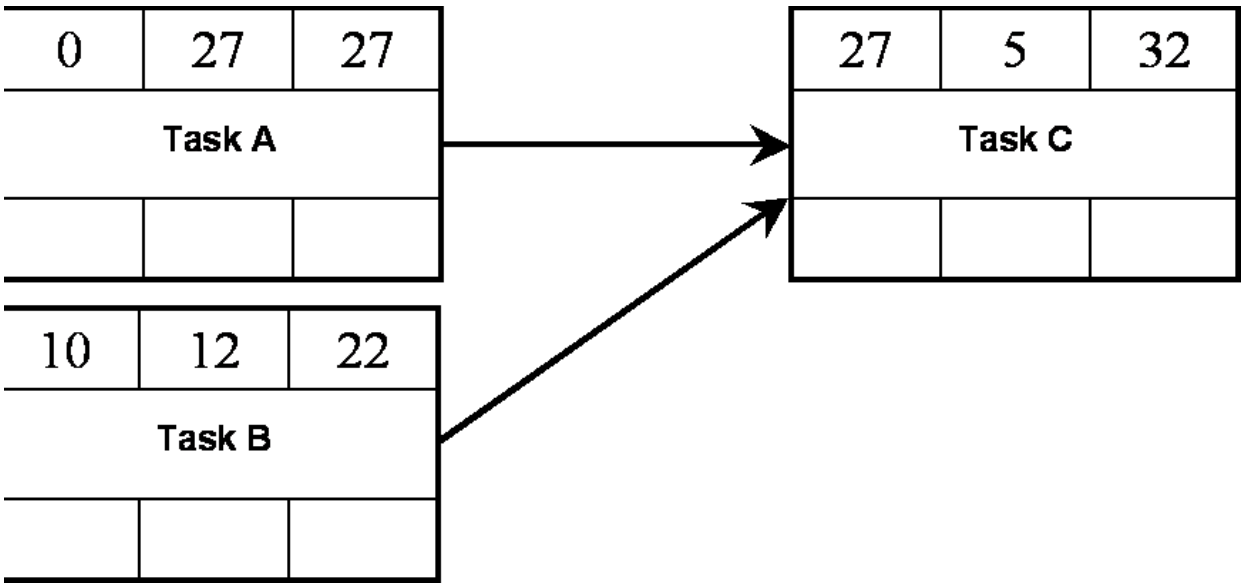
- Forward Pass
  - Calculates the Duration of the Project
- Backward Pass
  - Calculates the slack/float for each task and shows the critical path



- To calculate the total duration of the Project…
- For each task:
  - Take the earliest start time (EST)
  - Calculate the Earliest finish time (EFT):

EFT = EST+Duration

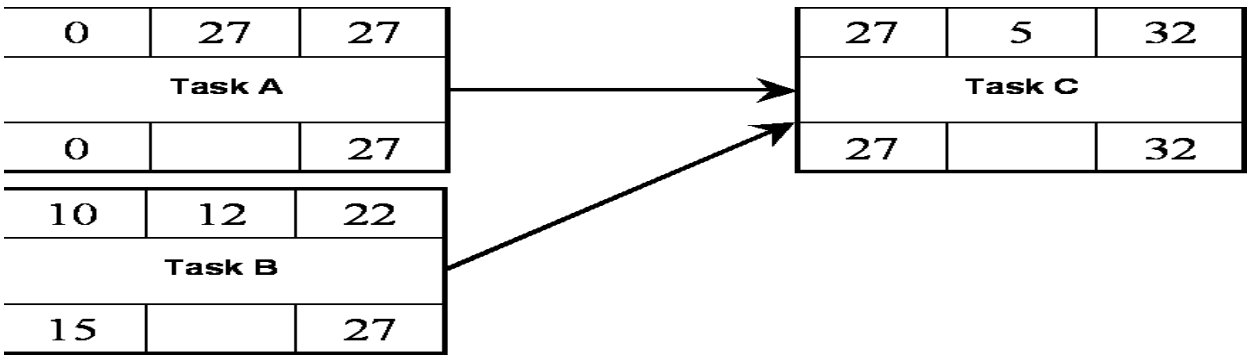


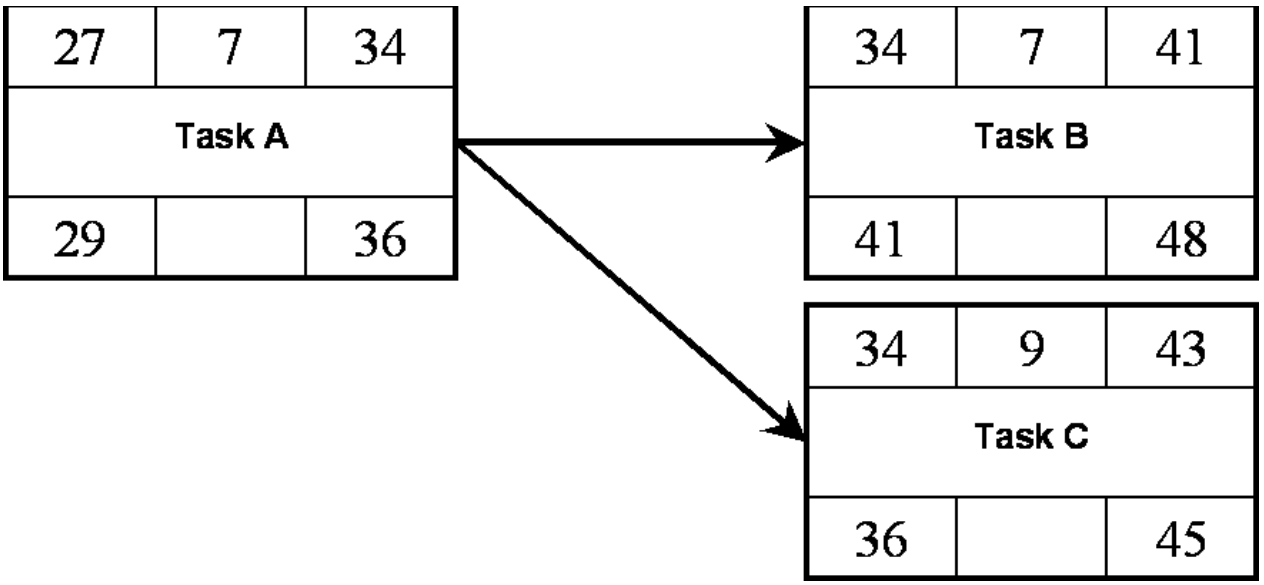


Backward Pass

- To calculate the float for each task?
- For each task:
  - Take the latest start time (LST)
  - Calculate the latest finish time (LFT):

LST = LFT-Duration





17. ACTIVITY FLOAT MEASURES

- Free float
  - The time by which an activity may be delayed without affecting any specific activity
- Interfering float
  - The diff between the total float and free float

18. Reducing Project Duration

- Time Is Money: Cost-Time Tradeoffs
  - Reducing the time of a critical activity usually incurs additional direct costs.
    - Cost-time solutions focus on reducing (crashing) activities on the critical path to shorten overall duration of the project.
  - Reasons for imposed project duration dates:
    - Time-to-market pressures
    - Unforeseen delays
    - Incentive contracts (bonuses for early completion)
    - Imposed deadlines and contract commitments
    - Overhead and public goodwill costs
    - Pressure to move resources to other projects

Options for Accelerating Project Completion

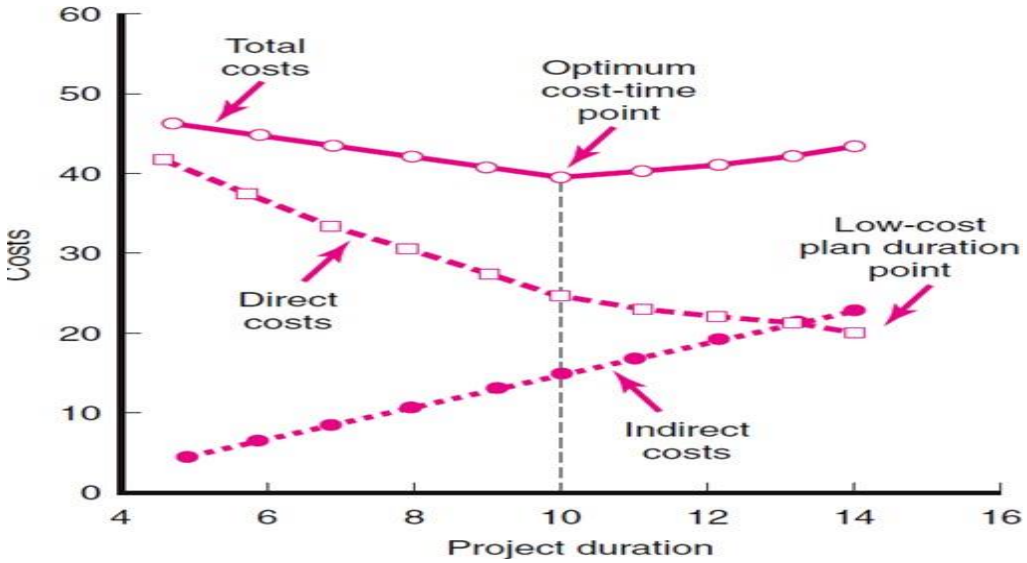
- Resources Not Constrained
  - Adding resources
  - Outsourcing project work
  - Scheduling overtime
  - Establishing a core project team
  - Do it twice—fast and then correctly
- Resources Constrained
  - Fast-tracking
  - Critical-chain
  - Reducing project scope
  - Compromise quality

Explanation of Project Costs

- Project Indirect Costs
  - Costs that cannot be associated with any particular work package or project activity.
    - Supervision, administration, consultants, and interest
  - Costs that vary (increase) with time.
    - Reducing project time directly reduces indirect costs.
- Project Direct Costs
  - Normal costs that can be assigned directly to a specific work package or project activity.
    - Labor, materials, equipment, and subcontractors
  - Crashing activities increases direct costs.

Reducing Project Duration

Project Cost–Duration Graph



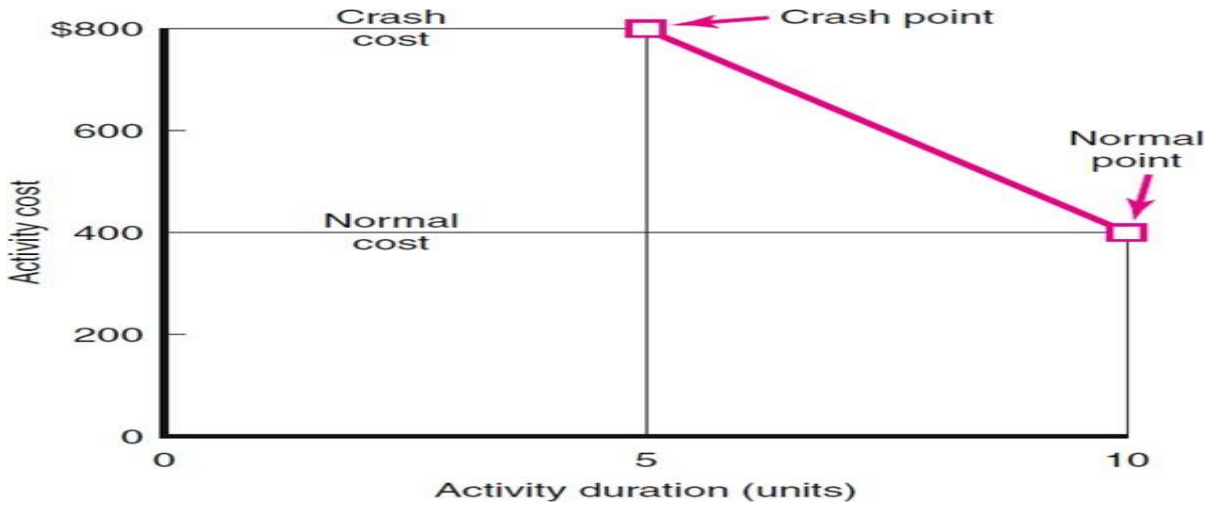
Constructing a Project Cost–Duration Graph

- Findtotal direct costs for selected project durations.
- Find total indirect costs for selected project durations.
- Sum direct and indirect costs for these selected project durations.
- Compare additional cost alternatives for benefits.

Constructing a Project Cost–Duration Graph

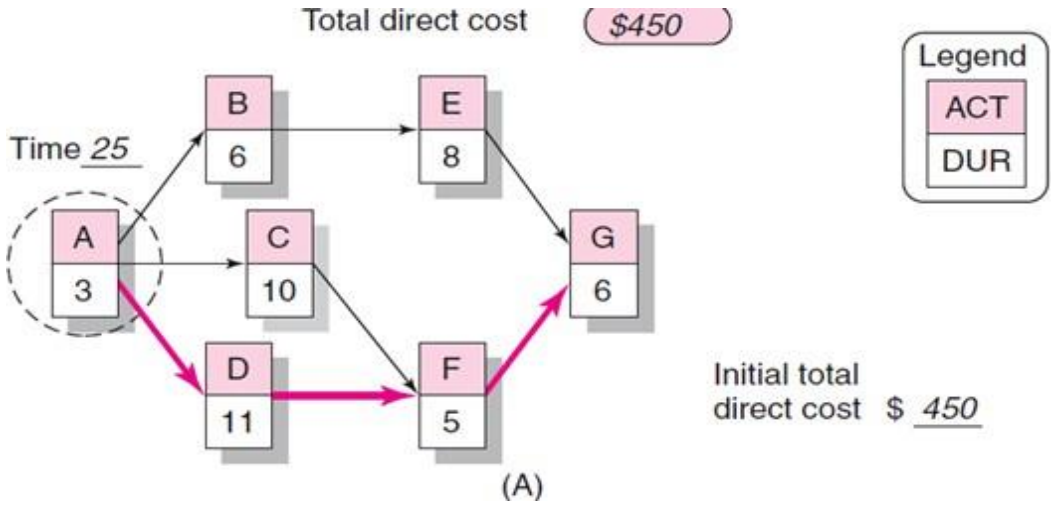
- **Determining Activities to Shorten**
  - Shorten the activities with the smallest increase in cost per unit of time.
  - Assumptions:
    - The cost relationship is linear.
    - Normal time assumes low-cost, efficient methods to complete the activity.
    - Crash time represents a limit—the greatest time reduction possible under realistic conditions.
    - Slope represents a constant cost per unit of time.
    - All accelerations must occur within the normal and crash times.

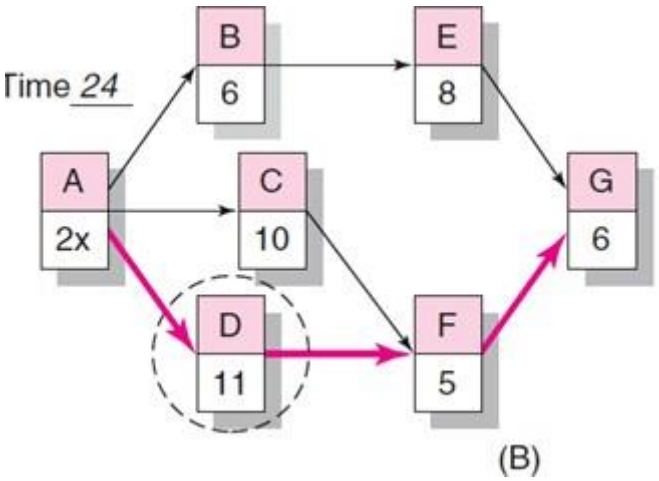
Activity Graph



Cost–Duration Trade-off Example

Activity ID	Slope	Maximum crash time	Direct costs			
			Normal		Crash	
			Time	Cost	Time	Cost
A	$20	1	3	$50	2	$70
B	40	2	6	80	4	160
C	30	1	10	60	9	90
D	25	4	11	50	7	150
E	30	2	8	100	6	160
F	30	1	5	40	4	70
G	0	0	6	70	6	70



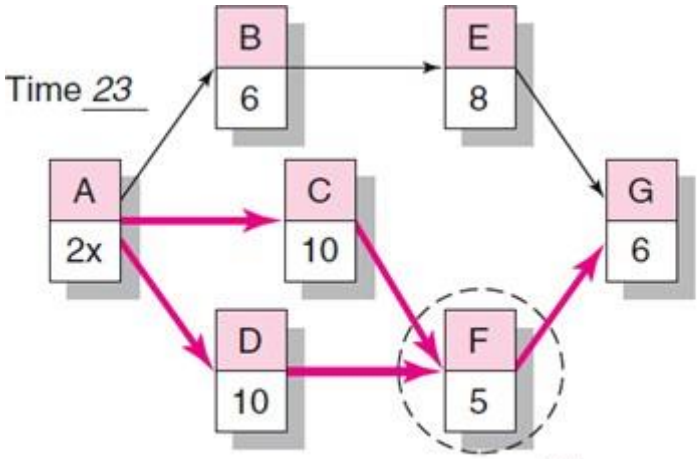


Total direct cost $ 470

Activities changed

A $20 ____ ____

(B)

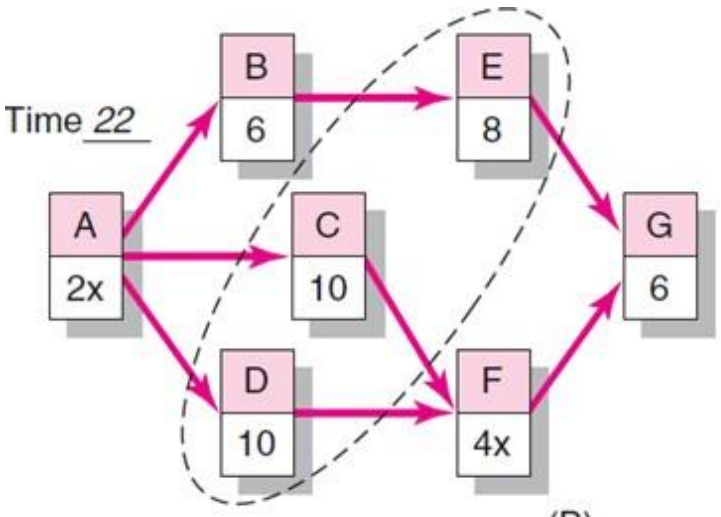


Total direct cost $ 495

Activities changed

D $25 ____ ____

(A)



Total direct cost $ 525

Activities changed

F $30 ____ ____

(B)

Practical Considerations

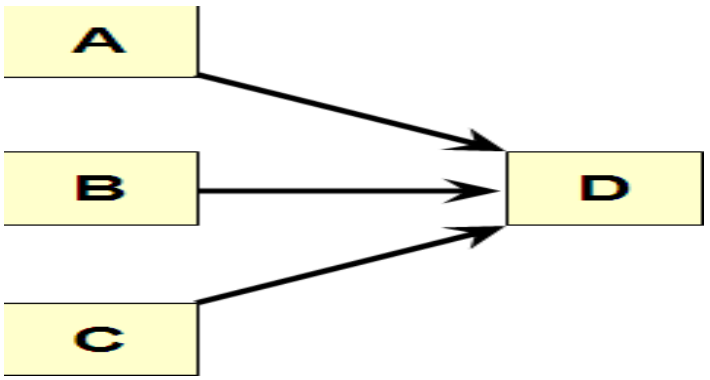
- Using the Project Cost–Duration Graph
- Crash Times
- Linearity Assumption
- Choice of Activities to Crash Revisited
- Time Reduction Decisions and Sensitivity

What if Cost, Not Time Is the Issue?

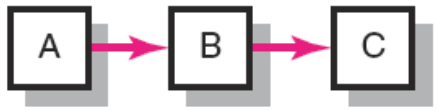
- Commonly Used Options for Cutting Costs
  - Reduce project scope
  - Have owner take on more responsibility
  - Outsourcing project activities or even the entire project
  - Brainstorming cost savings options

Constructing a Project Network

- Terminology
  - Activity: an element of theproject that requires time.
  - Merge activity: an activity that has two or more preceding activities on which it depends.
  - Parallel (concurrent) activities: Activities that can occur independently and, if desired, not at the same time.

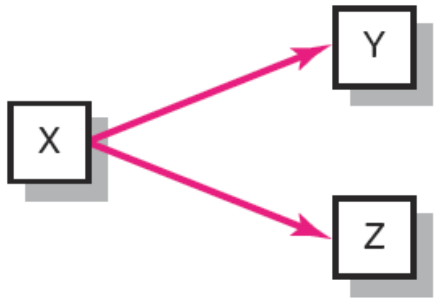


Activity-on-Node Fundamentals



A is preceded by nothing
B is preceded by A
C is preceded by B

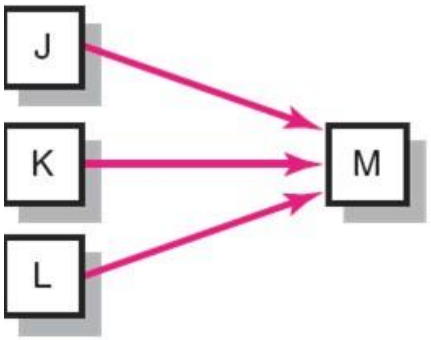
(A)



Y and Z are preceded by X

Y and Z can begin at the same time, if you wish

(B)

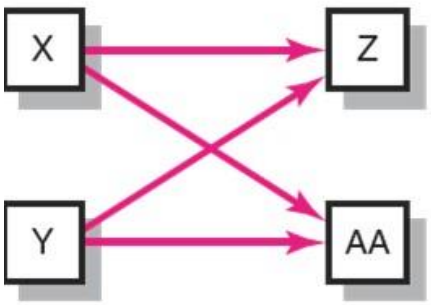


J, K, & L can all begin at the same time, if you wish (they need not occur simultaneously)

but

All (J, K, L) must be completed before M can begin

(C)



Z is preceded by X and Y

AA is preceded by X and Y

(D)

- Path: a sequence of connected, dependent activities.
- Critical path: the longest path through the activity network that allows for the completion of all project-related activities; the shortest expected time in which the

entire project can be completed. Delays on the critical path will delay completion of the entire project.

Forward Pass Computation

- Add activity times along each path in the network (ES + Duration = EF).
- Carry the early finish (EF) to the next activity where it becomes its early start (ES) unless…

The next succeeding activity is a merge activity, in which case the largest EF of all preceding activities is selected.

Backward Pass Computation

- Subtract activity times along each path in the network (LF - Duration = LS).
- Carry the late start (LS) to the next activity where it becomes its late finish (LF) unless...
- The next succeeding activity is a burst activity, in which case the smallest LF of all preceding activities is selected.

Determining Slack (or Float)

- Free Slack (or Float)
  - The amount of time an activity can be delayed without delaying connected successor activities
- Total Slack
  - The amount of time an activity can be delayed without delaying the entire project
- The critical path is the network path(s) that has (have) the least slack in common.

Sensitivity of a Network

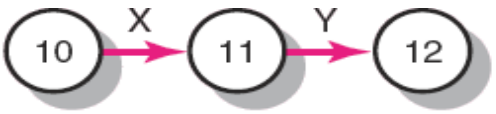
- The likelihood the original critical path(s) will change once the project is initiated.
  - Function of:

    The number of critical paths

    The amount of slack across near critical activities

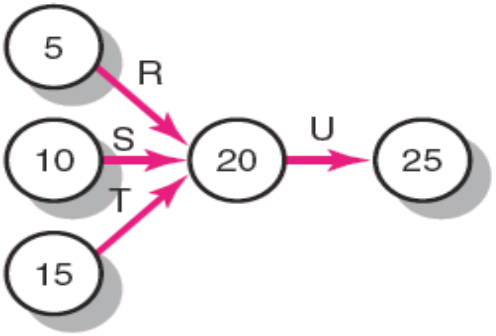
19. **Activity-on-Arrow Network-Building Blocks**

Activity-on-Arrow Network Fundamentals



Y is preceded by X

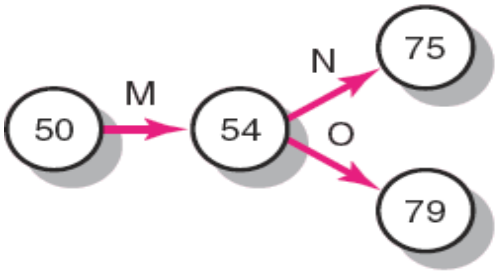
(A)



U is preceded by R, S, T

R, S, T can occur concurrently, if you wish

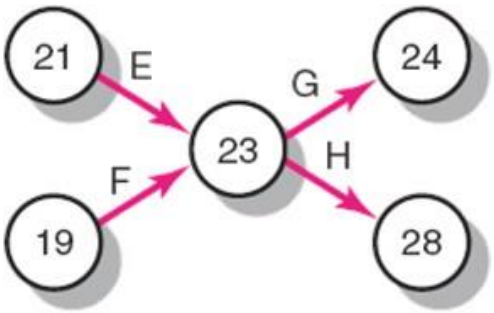
(B)



N and O are preceded by M

When M is complete, N and O can occur concurrently, if you wish

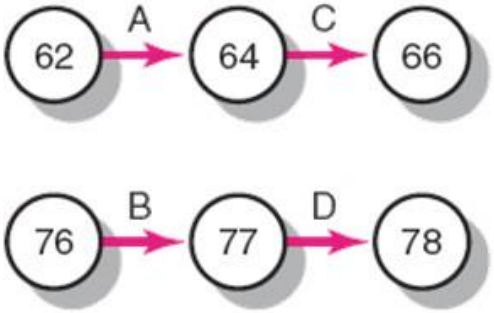
(C)



E and F must precede G and H

E and F can occur together, if you wish
G and H can occur together, if you wish

(D)

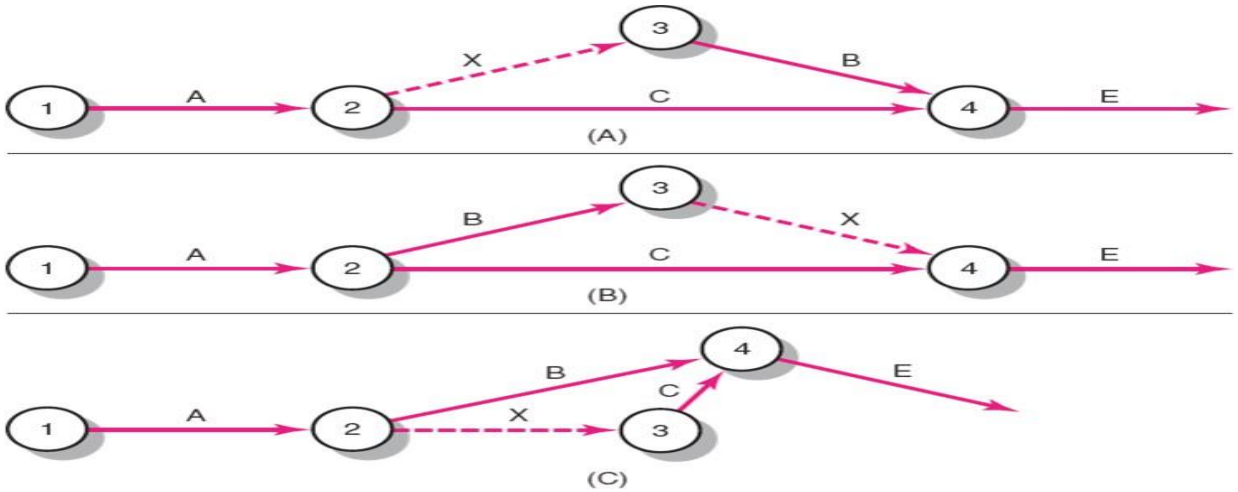


A must precede C
B must precede D

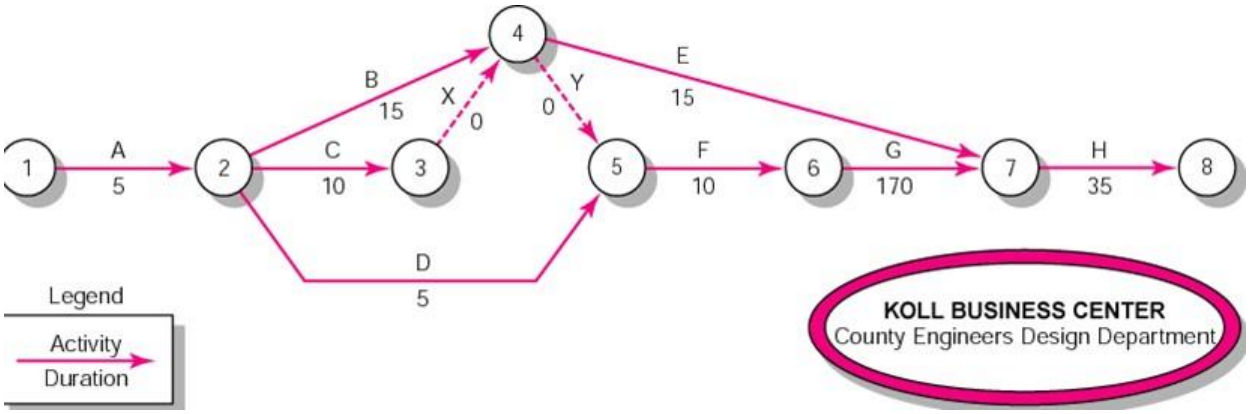
Path A–C is independent of path B–D

(E)

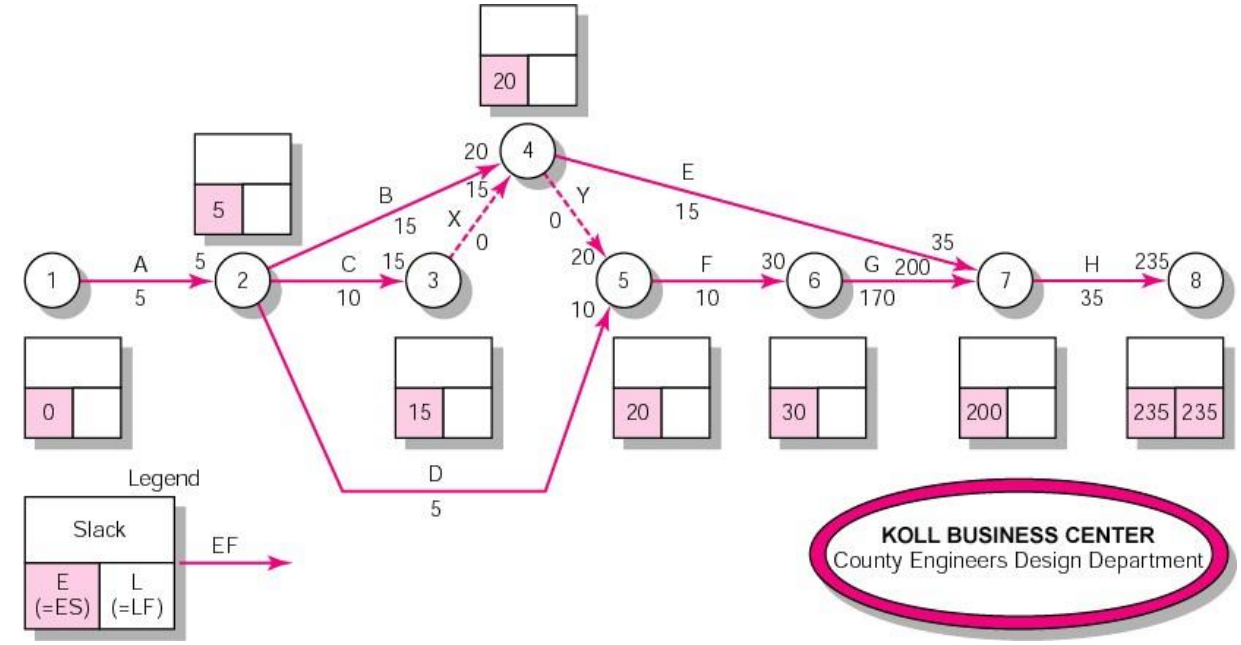
Partial AOA Koll Network



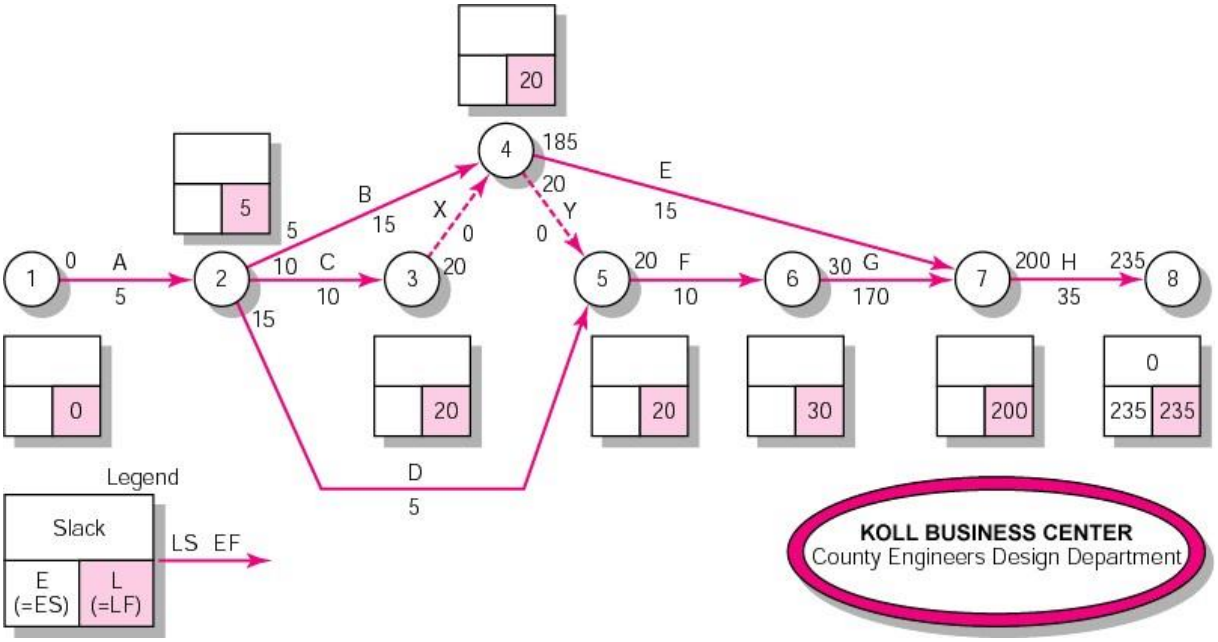
Activity – on-Arrow Network



Activity – on-Arrow Network Forward Pass



Activity-on-Arrow Network Backward Pass



20. RiskManagement

The proactive management of risks throughout the software development lifecycle is important for project success.

- The risk management practice, which involves risk identification, analysis, prioritization, planning, mitigation, monitoring, and communication
- software development risks that seem to reoccur in educational and industrial projects
- a risk-driven process for selecting a software development model

Risk Identification

In the risk identification step, the team systematically enumerates as many project risks as possible to make them explicit before they become problems. There are several ways to look at the kinds of software project risks.

There are some specific factors to consider when examining project, product, and business risks. Some examples of these factors are listed here, although this list is meant to stimulate your thinking rather than to be an all-inclusive list.

People risks are associated with the availability, skill level, and retention of the people on the development team.

Size risks are associated with the magnitude of the product and the product team. Larger products are generally more complex with more interactions. Larger teams are harder to coordinate.

Process risks are related to whether the team uses a defined, appropriate software development process and to whether the team members actually follow the process.

Technology risks are derived from the software or hardware technologies that are being used as part of the system being developed. Using new or emerging or complex technology increases the overall risk.

Tools risks, similar to technology risks, relate to the use, availability, and reliability of support software used by the development team, such as development environments and other Computer-Aided Software Engineering (CASE) tools.

Organizational and managerial risks are derived from the environment where the software is being developed. Some examples are the financial stability of the company and threats of company reorganization and the potential of the resultant loss of support by management due to a change in focus or a change in people.

Customer risks are derived from changes to the customer requirements, customers' lack of understanding of the impact of these changes, the process of managing these requirements changes, and the ability of the customer to communicate effectively with the team and to accurately convey the attributes of the desired product.

Estimation risks are derived from inaccuracies in estimating the resources and the time required to build the product properly.

Sales and support risks involve the chances that the team builds a product that the sales force does not understand how to sell or that is difficult to correct, adapt, or enhance.

Strategies for Risk Management:

During the software development process various strategies for risk management could be identified and defined according to the amount of risk influence. Based upon the amount of risk influence in software development project, risk strategies could be divided into three classes namely careful, typical, and flexible (Boban, M. et.). Generally, careful risk management strategy is projected for new and inexperienced organizations whose software development projects are connected with new and unproven technology; typical risk management strategy is well-defined as a support for mature organizations with experience in software development projects and used technologies, but whose projects carry a decent number of risks; and flexible risk management strategy is involved in experienced software development organizations whose software development projects are officially defined and based on proven technologies (Boban, M. etc.).

Categories of risks:

Schedule Risk:

Project schedule get slip when project tasks and schedule release risks are not addressed properly.

Schedule risks mainly effect on project and finally on company economy and may lead to project failure.

Schedules often slip due to following reasons:

- Wrong time estimation
- Resources are not tracked properly. All resources like staff, systems, skills of individuals etc.
- Failure to identify complex functionalities and time required to develop those functionalities.
- Unexpected project scope expansions.

Budget Risk:

- Wrong budget estimation.
- Cost overruns
- Project scope expansion

Operational Risks:

Risks of loss due to improper process implementation, failed system or some external events risks.

Causes of Operational risks:

- Failure to address priority conflicts
- Failure to resolve the responsibilities
- Insufficient resources
- No proper subject training
- No resource planning
- No communication in team.

Security in System Development

- Risk Analysis & Management needs to be a part of system development, not tacked on afterwards
- Baskerville's three generations of methods

1st Generation: Checklists

Example: BS 7799 Part 1

2nd Generation: Mechanistic engineering methods

Example: this risk analysis method

3rd Generation: Integrated design

Not yet achieved

Definitions:

The meanings of terms in this area are not universally agreed. We will use the following

- **Threat**: Harm that can happen to an asset
- **Impact**: A measure of the seriousness of a threat
- **Attack**: A threatening event
- **Attacker**: The agent causing an attack (not necessarily human)
- **Vulnerability**: a weakness in the system that makes an attack more likely to succeed
- **Risk**: a quantified measure of the likelihood of a threat being realised
- **Risk Analysis** involves the identification and assessment of the levels of risk, calculated from the
    - Values of assets

    - Threats to the assets
    - Their vulnerabilities and likelihood of exploitation
- **Risk Management** involves the identification, selection and adoption of security measures justified by
    - The identified risks to assets
    - The reduction of these risks to acceptable levels

Goals of Risk Analysis:

- All assets have been identified
- All threats have been identified
    - Their impact on assets has been valued
- All vulnerabilities have been identified and assessed

Problems of Measuring Risk

- Businesses normally wish to measure in money, but
- Many of the entities do not allow this
    - Valuation of assets
        - Value of data and in-house software - no market value
        - Value of goodwill and customer confidence
    - Likelihood of threats
        - How relevant is past data to the calculation of future probabilities?
        - The nature of future attacks is unpredictable
        - The actions of future attackers are unpredictable
    - Measurement of benefit from security measures
        - Problems with the difference of two approximate quantities
        - How does an extra security measure affect a $\sim 10^{-5}$ probability of attack?

Risk Levels

- Precise monetary values give a false precision

- Better to use levels, e.g.
  - High, Medium, Low
    - High: major impact on the organisation
    - Medium: noticeable impact (“material” in auditing terms)
    - Low: can be absorbed without difficulty
  - 1 - 10
- Express money values in levels, e.g.
  - For a large University Department a possibility is
    - High
    - Medium
    - Low

Risk Analysis Steps

- Decide on scope of analysis
  - Set the system boundary
- Identification of assets & business processes
- Identification of threats and valuation of their impact on assets (impact valuation)
- Identification and assessment of vulnerabilities to threats
- Risk assessment

Risk Analysis – Defining the Scope

- Draw a context diagram
- Decide on the boundary
  - It will rarely be the computer!
- Make explicit assumptions about the security of neighbouring domains
  - Verify them!

Risk Analysis - Identification of Assets

- Types of asset
    - Hardware
    - Software: purchased or developed programs
    - Data
    - People: who run the system
    - Documentation: manuals, administrative procedures, etc
    - Supplies: paper forms, magnetic media, printer liquid, etc
    - Money
    - Intangibles
        - Goodwill
        - Organization confidence
        - Organisation image

Risk Analysis – Impact Valuation

Identification and valuation of threats - for each group of assets

- Identify threats, e.g. for stored data
    - Loss of **confidentiality**
    - Loss of **integrity**
    - Loss of **completeness**
    - Loss of **availability** (Denial of Service)
- For many asset types the only threat is loss of availability
- Assess impact of threat
    - Assess in levels, e.g H-M-L or 1 - 10
    - This gives the valuation of the asset in the face of the threat

Risk Analysis – Vulnerabilities

- Identify vulnerabilities against a baseline system

For risk analysis of an existing system

- Existing system with its known security measures and weaknesses

For development of a new system

- Security facilities of the envisaged software, e.g. Windows NT
- Standard good practice, e.g. BS 7799 recommendations of good practice

For each threat

- Identify vulnerabilities
  - How to exploit a threat successfully;
- Assess levels of likelihood - High, Medium, Low
  - Of attempt

    Expensive attacks are less likely (e.g. brute-force attacks on encryption keys)
  - Successful exploitation of vulnerability;
- Combine them

Risk Assessment

Assess risk

- If we had accurate probabilities and values, risk would be
  - Impact valuation x probability of threat x probability of exploitation
  - Plus a correction factor for risk aversion
- Since we haven't, we construct matrices such as

Impact valuation

Risk	**Low**	**Med**	**high**
low	**Low**	**Low**	**med**
med	**Low**	**Med**	**High**
high	**Low**	**Med**	**High**

Responses to risk

- Avoid it completely by withdrawing from an activity
- Accept it and do nothing
- Reduce it with security measures

Risk management

- Risk management is concerned with identifying risks and drawing up plans to minimisetheir effect on a project.
- A risk is a probability that some adverse circumstance will occur

    Project risks affect schedule or resources;

    Product risks affect the quality or performance of the software being developed;

    Business risks affect the organisation developing or procuring the software.

The risk management process

Risk identification

Identify project, product and business risks;

Risk analysis

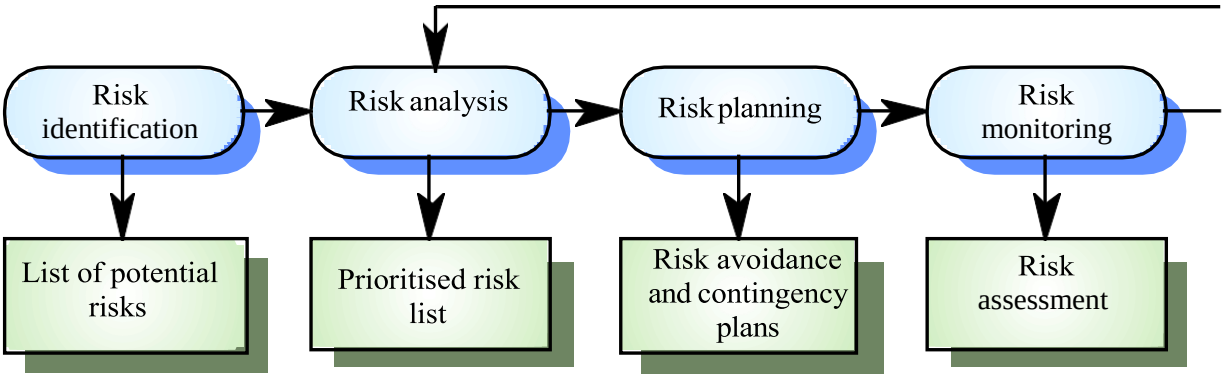
Assess the likelihood and consequences of these risks;

Risk planning

Draw up plans to avoid or minimise the effects of the risk;

Risk monitoring

Monitor the risks throughout the project;



21. **Hazard Identification**

Systematic Processes

What Constitutes a Hazard?

A real or potential condition that, when activated, can transform into a series of interrelated events that result in damage to equipment or property and or injury to people.

Safety Managers View

- Hazard
  - An implied threat or danger, a potential condition waiting to become a loss

- Stimulus
    - Required to initiate action from potential to kinetic
    - May be a:
        - Component out of tolerance
        - Maintenance failure
        - Operator failure
        - Any combination of other events and conditions

When Do We Look for Hazards?

The 5 Common Phases of a Systems Life Cycle

- Conceptual - Research
- Design (Validation & Verification)
- Development (Full-scale engineering & production)
- Operational Deployment
- Termination & Disposal

Hazard Severity

A key factor in establishing a common understanding of a safety programs goal

MIL-STD 882 suggests four categories

- Cat 1: Catastrophic
- Cat 2: Critical
- Cat 3: Marginal
- Cat 4: Negligible

Hazard Analysis Methods

- Failure Modes & Effects Analysis (FMEA)
    - Systematic look at hardware piece by piece
    - Review of how each component could fail
    - Considers how a failure effects other components, sub-systems and systems as a whole
    - Risk assessment accomplished (severity & probability)
- Risk Assessment Code (RAC) assigned
- Fault Tree Analysis (FTA)
    - Detailed review of a specific undesirable event
    - Deductive in nature
    - Top-down effort

    - Normally reserved for critical failures or mishaps
- May be qualitative or quantitative
- Operating Hazard Analysis (OHA)
    - Also known as Operating & Support Hazard Analysis (O&SHA)
    - "What if" tool brings user into the loop
- Integrates people and procedures into the system
    - Diagrams the flow or sequence of events
- Project Evaluation Tree (PET) may be used for OHA accomplishment
    - Systematic evaluation of man, machine, & procedures

22. PURPOSE OF THE RISK MANAGEMENT PLAN

A risk is an event or condition that, if it occurs, could have a positive or negative effect on a project's objectives. Risk Management is the process of identifying, assessing, responding to, monitoring and controlling, and reporting risks. This Risk Management Plan defines how risks associated with the<Project Name> project will be identified, analyzed, and managed. It outlines how risk management activities will be performed, recorded, and monitored throughout the lifecycle of the project and provides templates and practices for recording and prioritizing risks by the Risk Manager and/or Risk Management Team.

Risks related to IT systems or applications must be identified and documented based on the methodology in NIST SP 800-30, Risk Management Guide for Information Technology Systems. IT system or application weaknesses must be identified on an associated plan of action and milestones (POA&M) and tracked in accordance with HHS POA&M guidelines. Appropriate protective measures must be taken to safeguard sensitive IT system or application weaknesses or vulnerabilities from unauthorized disclosure.

RISK RESPONSE PLANNING

Each major risk (those falling in the Red & Yellow zones) will be assigned to a risk owner for monitoring and controlling purposes to ensure that the risk will not "fall through the cracks".

For each major risk, one of the following approaches will be selected to address it:

- **Avoid** – Eliminate the threat or condition or to protect the project objectives from its impact by eliminating the cause
- **Mitigate** – Identify ways to reduce the probability or the impact of the risk
- **Accept** – Nothing will be done
- **Contingency** –Define actions to be taken in response to risks
- **Transfer** – Shift the consequence of a risk to a third party together with ownership of the response by making another party responsible for the risk (buy insurance, outsourcing, etc.)

RISK MONITORING, CONTROLLING, AND REPORTING

The level of risk on a project will be tracked, monitored and controlled and reported throughout the project lifecycle. [Describe the methods and metrics that will be used to track the project's risk status throughout the lifecycle as well as how this status will be reported to the stakeholders/ management.]

Risks will be assigned a risk owner(s) who will track, monitor and control and report on the status and effectiveness of each risk response action to the Project Manager and Risk Management Team on a <insert timeframe>.

A "Top 10 Risk List" will be maintained by the PM/Risk Manager or IPT and will be reported as a component of the project status reporting process for this project.

All project change requests will be analyzed for their possible impact to the project risks.

As Risk Events occur, the list will be re-prioritized during weekly reviews and risk management plan will reflect any and all changes to the risk lists including secondary and residual risks.

Management will be notified of important changes to risk status as a component to the Executive Project Status Report. [State timeframe, i.e., every two weeks]

The Risk Manager (PM) will:

- Review, reevaluate, and modify the probability and impact for each risk item [timeframe, as needed, every two weeks, etc.]
- Analyze any new risks that are identified and add these items to the risk list (or risk database).
- Monitor and control risks that have been identified
- Review and update the top ten risk list [timeframe, as needed, every two weeks, etc.]
- Escalate issues/ problems to management [List factors that would need to be escalated to management. Examples: documented mitigation actions are not effective or producing the desired results; the overall level of risk is rising.]

The Risk Owner will:

- Help develop the risk response and risk trigger and carry out the execution of the risk response, if a risk event occurs.
- Participate in the review, re-evaluation, and modification of the probability and impact for each risk item on a weekly basis.
- Identify and participate in the analysis of any new risks that occur.
- Escalate issues/problems to PM that,

  Significantly impact the projects triple constraint or trigger another risk event to occur.

  Require action prior to the next weekly review

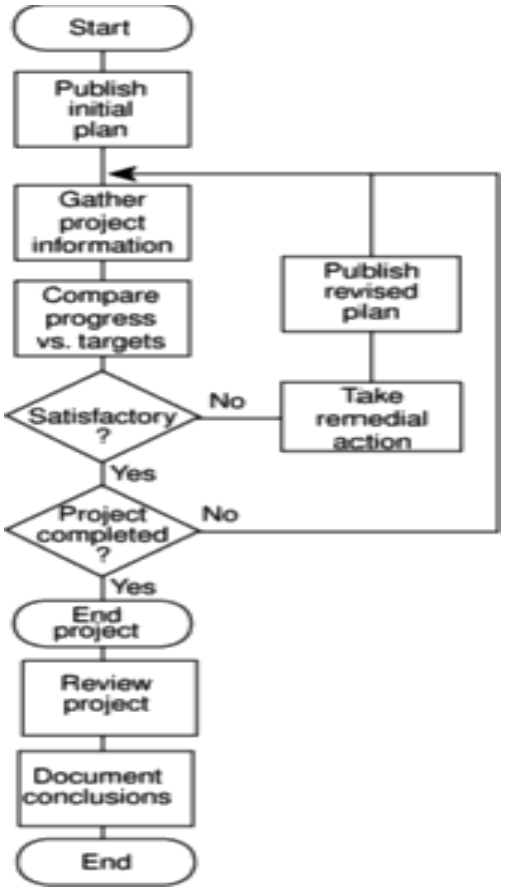
Risk strategy is not effective or productive causing the need to execute the contingency plan.

UNITIV MONITORINGANDCONTROL

CreatingFramework– Collecting TheData–VisualizingProgress –CostMonitoring– EarnedValue–PrioritizingMonitoring–GettingProjectBackToTarget–Change Control– ManagingContracts– Introduction– TypesOfContract–StagesInContract Placement–Typical Terms OfAContract –Contract Management– Acceptance.

23. Creating framework

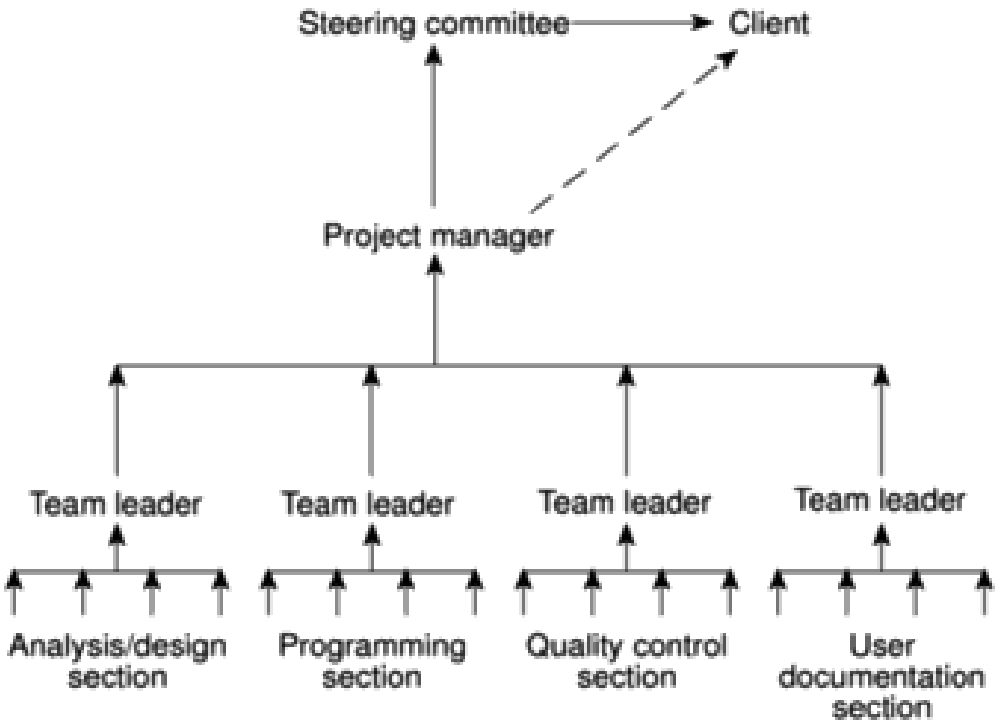
Project control cycle



Responsibility

- Project steering committee
- Project board
- Reporting formal or informal

Assessing progress



Setting checkpoints

- Regular
- Tied to specific events

Taking snapshots

- Review points or control points
- Assess progress daily

24. **Collecting data**

Contents

- Partial completion reporting
- Risk reporting

Time Sheet:

Time Sheet

Staff John Smith **Week ending** 26/3/99

Rechargeable hours

Project	Activity code	Description	Hours this week	% Complete	Scheduled completion	Estimated completion
P21	A243	Code mod A3	12	30	24/4/99	24/4/99
P34	B771	Document take-on	20	90	1/4/99	29/3/99

Total recharged hours	32

Non-rechargeable hours

Code	Description	Hours	Comment & authorization
z99	day in lieu	8	Authorized by RB

Total non-rechargeable hours	8

Activity Assessment Sheet:

Activity Assessment Sheet

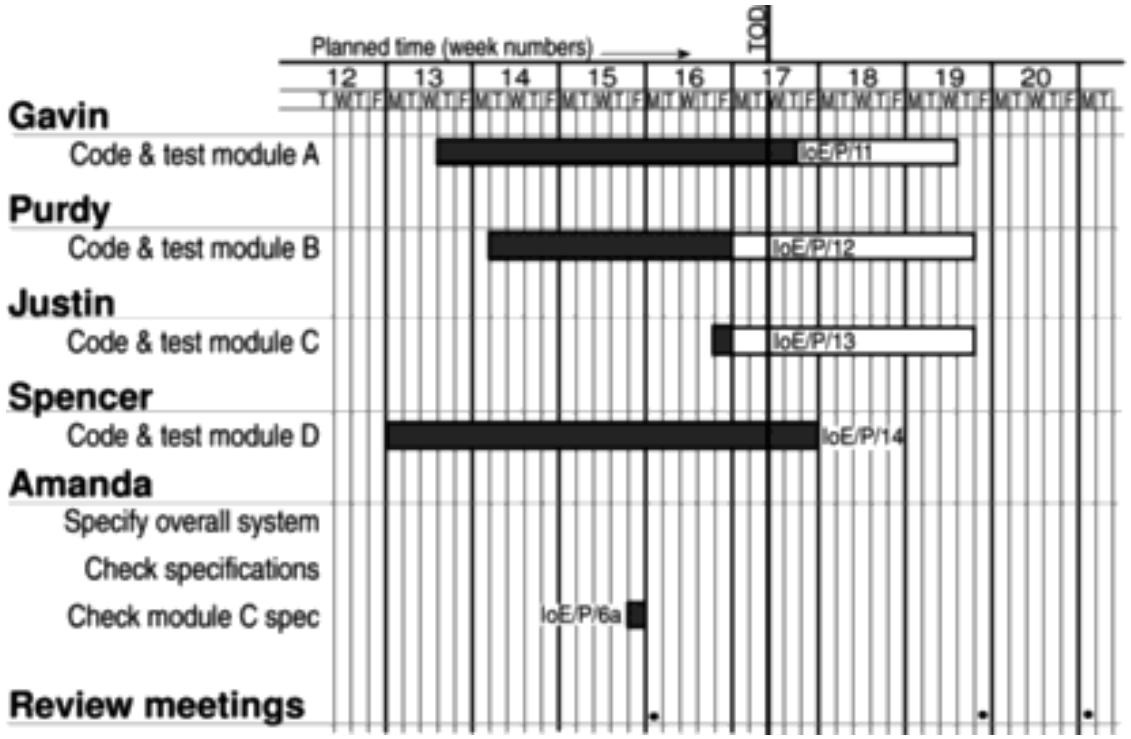
Staff Justin

Ref: IoE/P/13 **Activity: Code & test module C**

Week number	13	14	15	16	17	18	
Activity Summary	C	A	A	R			
Component							*Comments*
Screen handling procedures	C	A	A	C			
File update procedures	C	C	R	A			
Housekeeping procedures	C	C	C	A			
Compilation	C	C	C	R			
Test data runs	C	C	C	A			
Program documentation	C	C	A	R			

25. **Visualizing progress:**

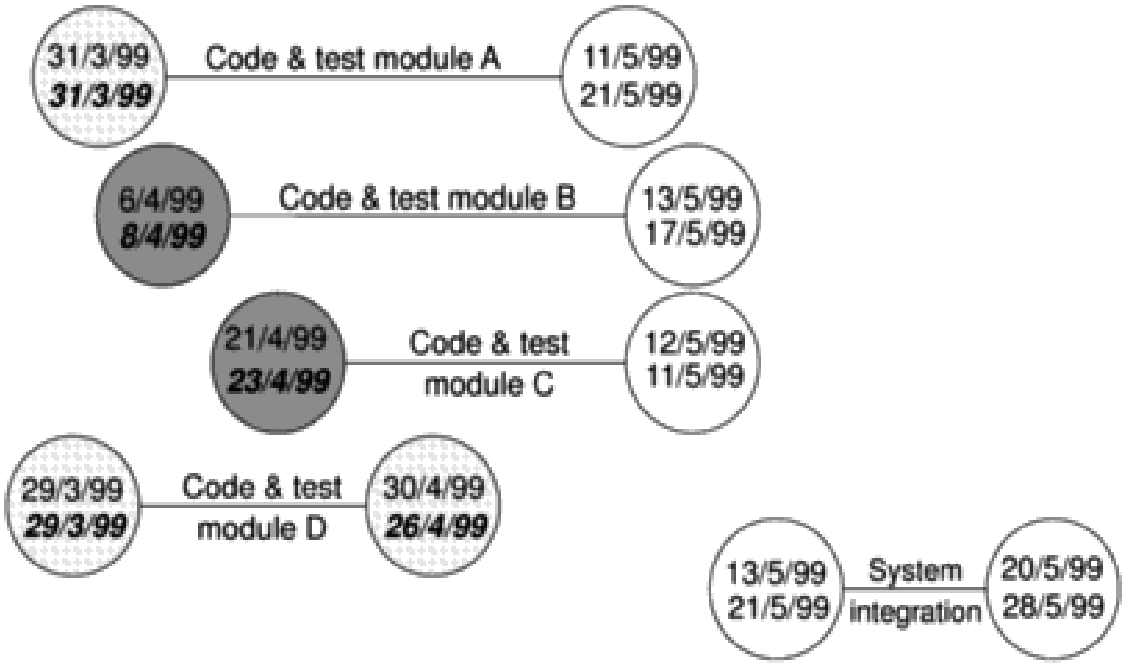
Gantt chart:



Slip chart

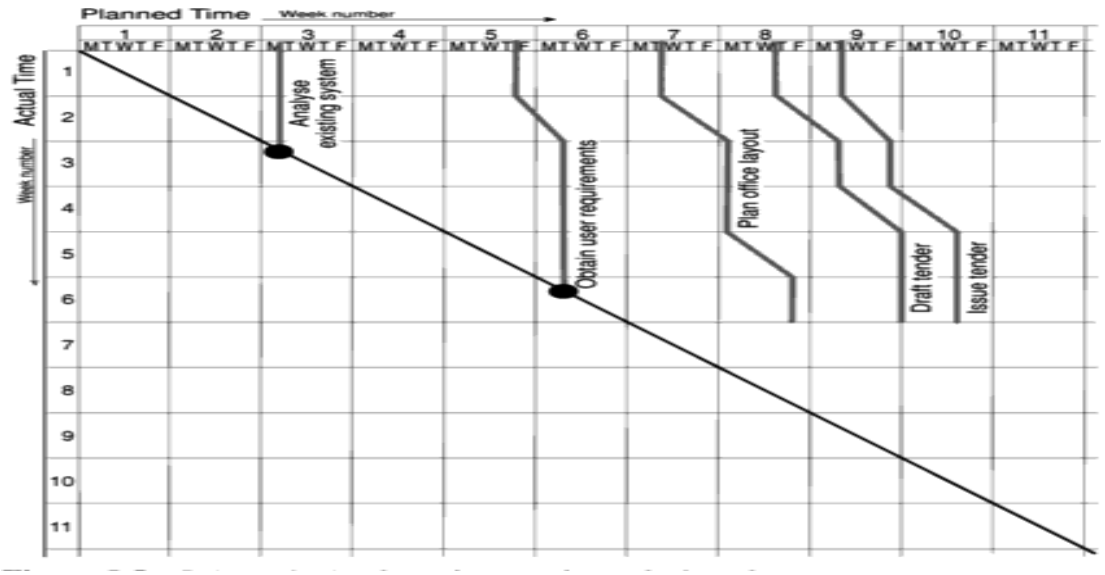
- provides visual indication of activities which are not progressing in schedule

Ball charts -Shows whether or not targets have been met



Time lining:

- Records and displays how the targets have changed throughout the duration of project

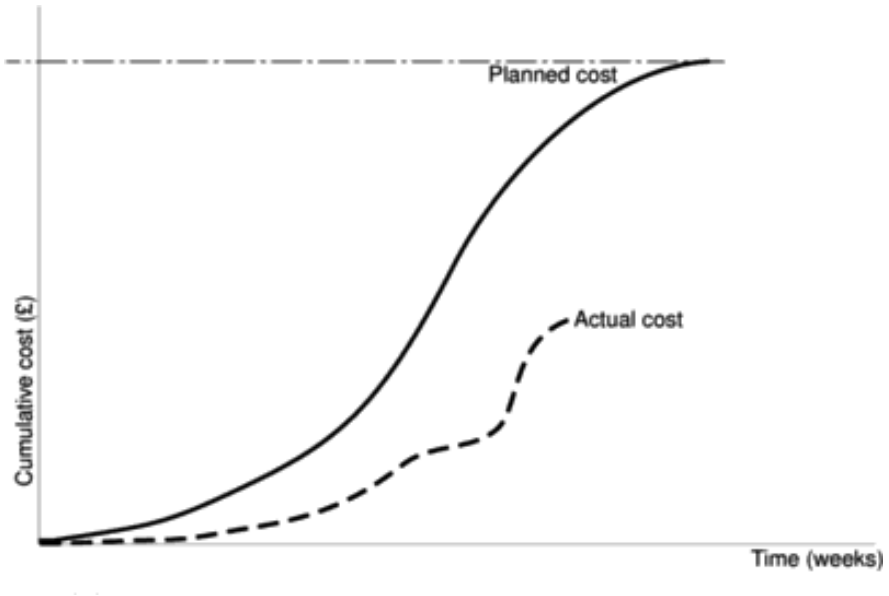


26. **Cost monitoring:**

- Expenditure monitoring
- Framing cumulative expenditure chart
- Projected future costs

- Computer-based planning tool

Tracking expenditure



26. Earned value:

Earned value analysis

- Assigns a value to each task
- BCWS
- Baseline budget
- BCWP-budgeted cost of work performed

Technique

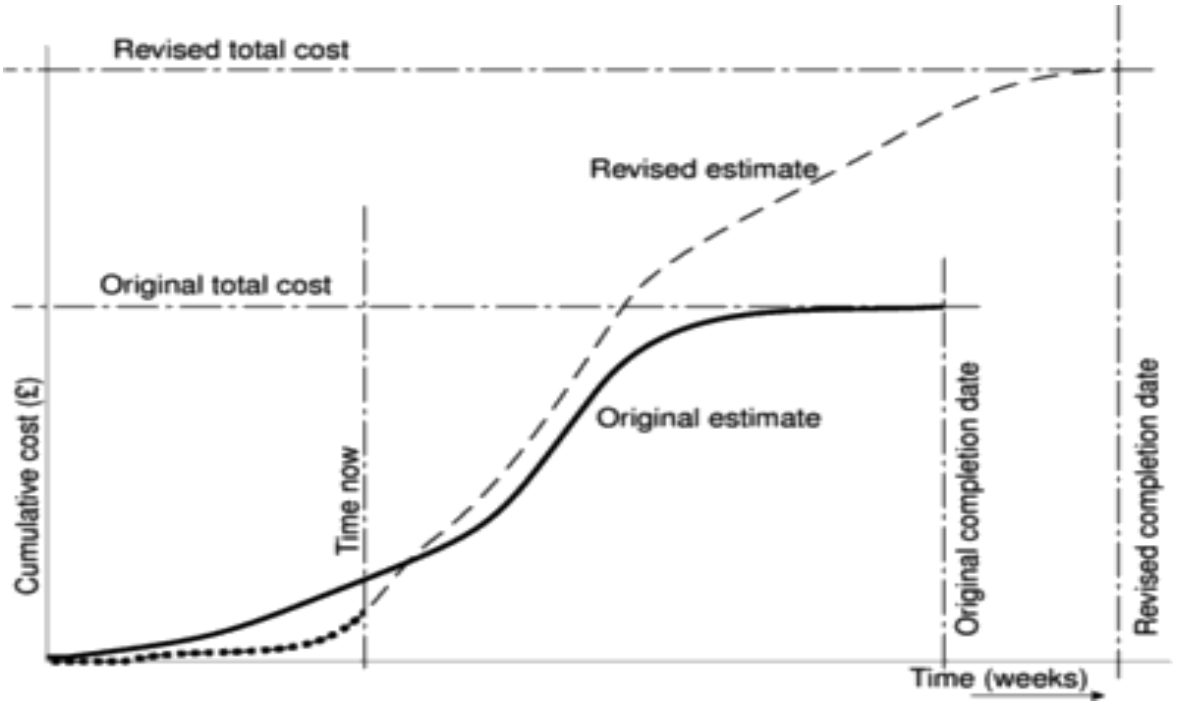
- The 0/100 technique
- The 50/50 technique
- The mile stone technique

The baseline budget

- First stage in baseline budget
- Forecast growth

- Specify overall system

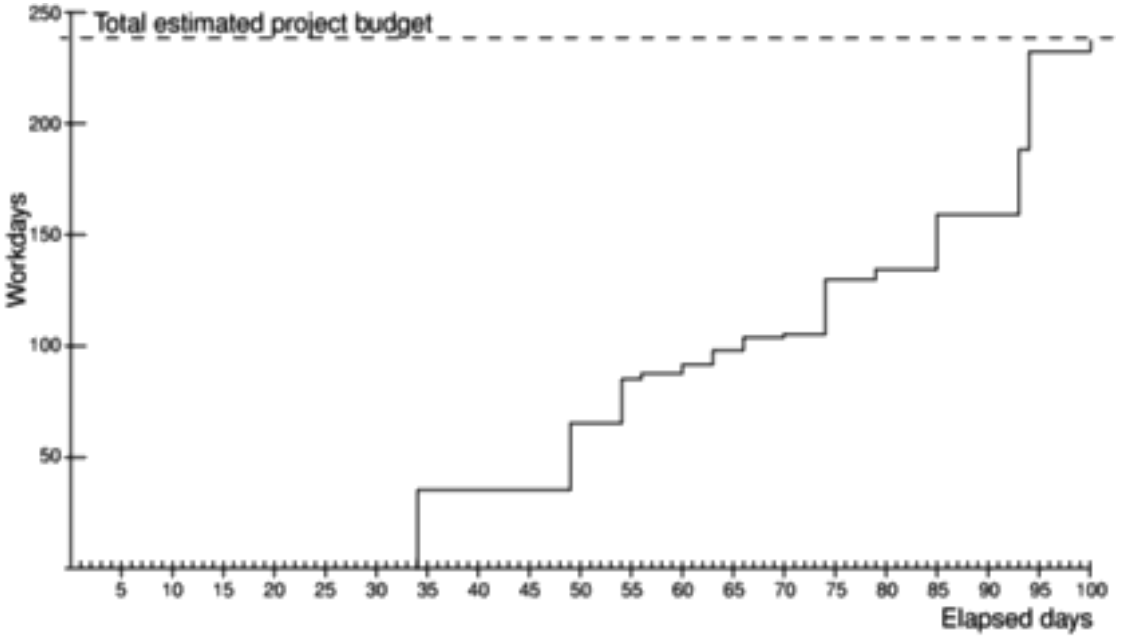
Expenditure chart



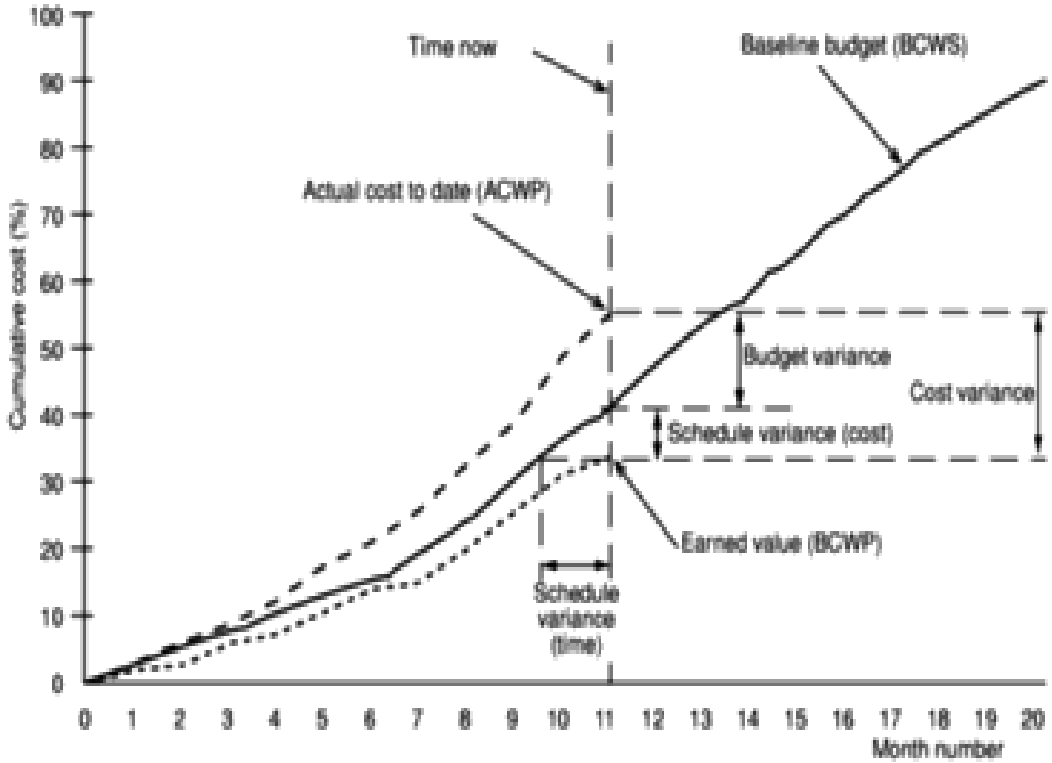
Baseline budget calculation

Task	*Budgeted workdays*	*Scheduled completion*	*Cumulative workdays*	*% cumulative earned value*
Specify overall system	34	34	34	14.35
Specify module B	15	49	} 64	27.00
Specify module D	15	49		
Specify module A	20	54	84	35.44
Check specifications	2	56	86	36.28
Design module D	4	60	90	37.97
Design module A	7	63	97	40.93
Design module B	6	66	103	43.46
Check module C spec	1	70	104	43.88
Specify module C	25	74	129	54.43
Design module C	4	79	133	56.12
Code & test module D	25	85	158	66.67
Code & test module A	30	93	188	79.32
Code & test module B	28	94	} 231	97.47
Code & test module C	15	94		
System integration	6	100	237	100.00

Milestone earned value



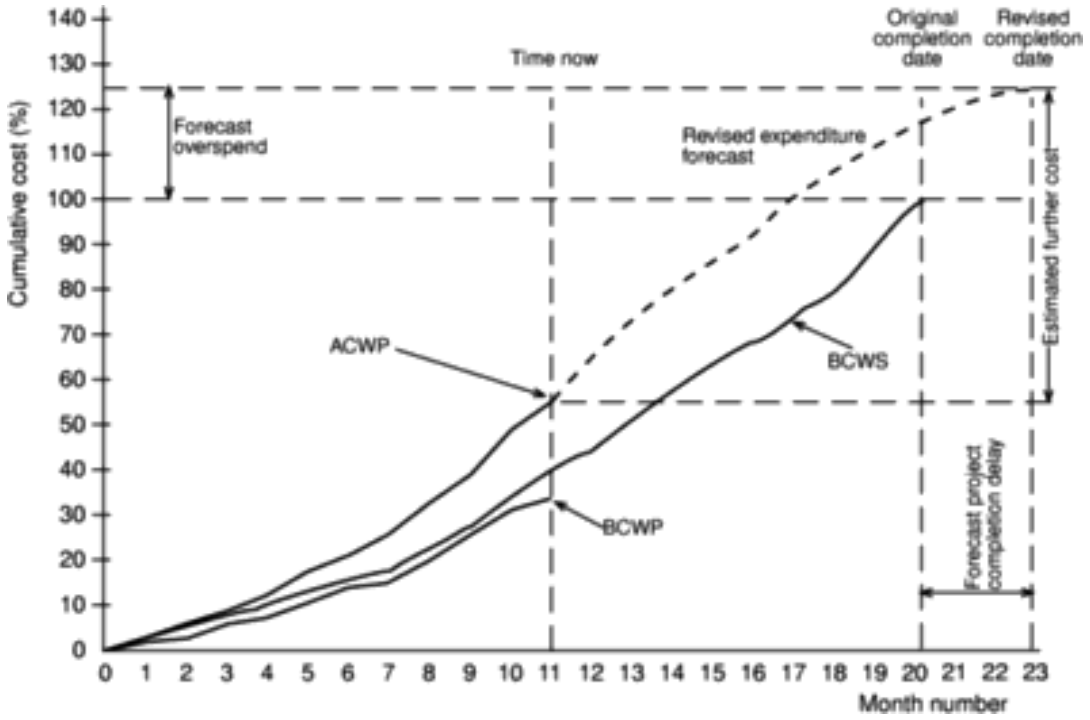
Earned value tracking chart



Performance statistics from earned value chart

- Budget variance
- Schedule variance
- Coast variance
- Performance ratios Schedule performance index SPI=BCWP/BCWS

Earned value charts with revised forecasts



Definition:

Earned value analysis is a method of performance measurement. Earned value integrates cost, schedule and scope and can be used to forecast future performance and project completion dates. It allows projects to be managed better – on time, on budget.

Three quantities form the basis for cost performance measurement using Earned Value Management. They are

1. Budgeted Cost of Work Scheduled (BCWS) or Planned Value (PV)
2. Budgeted Cost of Work Performed (BCWP) or Earned Value (EV) and
3. Actual Cost of Work Performed (ACWP) or Actual Cost (AC).

The above quantities are defined below.

• **Budgeted Cost of Work Scheduled (BCWS)** or **Planned Value (PV)**

– The sum of budgets for all work packages scheduled to be accomplished within a given time period.

• **Budgeted Cost of Work Performed (BCWP)** or **Earned Value (EV)**

– The sum of budgets for completed work packages and completed portions of open work packages.

• **Actual Cost of Work Performed (ACWP)** or **Actual Cost (AC)**

– The actual cost incurred in accomplishing the work performed within a given time period. For equitable comparison, ACWP is only recorded for the work performed to date against tasks for which a BCWP is also reported.

From these three quantities we can determine our total program budget as well as make a determination of schedule and cost performance and provide an estimated cost of the project at its completion. Additional terms are defined to record cost and schedule performance and program budget:

• **Schedule Variance (SV)**

– The difference between the work actually performed (BCWP) and the work scheduled (BCWS). The schedule variance is calculated in terms of the difference in dollar value between the amount of work that should have been completed in a given time period and the work actually completed.

• **Cost Variance (CV)**

– The difference between the planned cost of work performed (BCWP) and actual cost incurred for the work (ACWP). This is the actual dollar value by which a project is either overrunning or under running its estimated cost

Two Performance Ratios:

• **Cost Performance Index (CPI)**

– The ratio of cost of work performed (BCWP) to actual cost (ACWP). CPI of 1.0 implies that the actual cost matches to the estimated cost. CPI greater than 1.0 indicates work is accomplished for less cost than what was planned or budgeted. CPI less than 1.0 indicates the project is facing cost overrun.

• **Schedule Performance Index (SPI)**

– The ratio of work accomplished (BCWP) versus work planned (BCWS), for a specific time period. SPI indicates the rate at which the project is progressing.

• **Estimate at Completion (EAC)**

– It is a forecast of most likely total project costs based on project performance and risk quantification. At the start of the project BAC and EAC will be equal. EAC will vary from BAC only when actual costs (ACWP) vary from the planned costs (BCWP).

4.5.2. Earned Value Management Formula:

Name	Formula
Cost Variance (CV)	EV – AC
Schedule Variance (SV)	EV – PV
Time Variance (TV)	Difference between the time when the achievement of the current earned value was planned to occur and the time now
Cost Performance Index (CPI)	EV / AC
Schedule Performance Index (SPI)	EV / PV

27. Prioritizing monitoring

- Critical path activities
- Activities with no free float
- Activities with less than a specified float
- High risk activities
- Activities using critical resources

28. Getting the project back to target:

2 Strategies

- Shortening the critical path
- Altering the activity precedence requirements

Shorten critical path

- Speed up non-critical path activities
- Fact finding
- Time/cost trade off

Reconsider the precedence requirements

- Normal working practices
- Subdivide to components
- Assess changes

29. Change control:

- A change in program specification
- Change program design and then code

Change control procedures

Change control procedures

A simple change control procedure for operational systems might have the following steps.

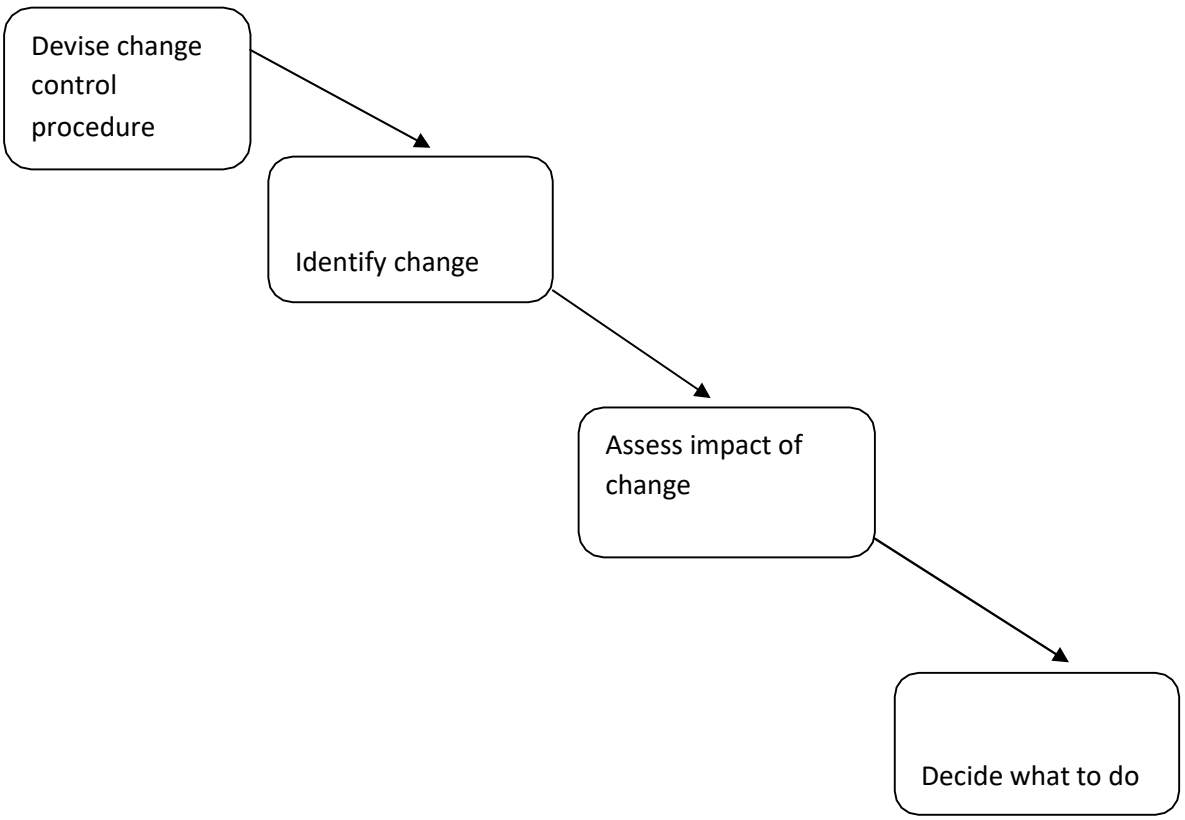
1. One or more users might perceive a need for a modification to a system and ask for a change request to be passed to the development staff.
2. The user management consider the change request and if they approve it pass it to the development management.
3. The development management delegate a member of staff to look at the request and to report on the practicality and cost of carrying out the change. They would, as part of this, assess the products that would be affected by the change.
4. The development management report back to the user management on the findings and the user management decide whether, in view of the cost quoted, they wish to go ahead.
5. One or more developers are authorized to take copies of the master products that are to be modified.
6. The copies are modified. In the case of software components this would involve modifying the code and recompiling and testing it.
7. When the development of new versions of the product has been completed the user management will be notified and copies of the software will be released for user acceptance testing.
8. When the user is satisfied that the products are adequate they will authorize their operational release. The master copies of configuration items will be replaced.

The role of configuration librarian:

- Identifying items that need to be subject to change control
- Management of a central repository of the master copies of software and documentation
- Administering change procedure.
- Maintenance of access records

Typical change control process

- One or more users might perceive the need for a change
- User management decide that the change is valid and worthwhile and pass it to development management
- A developer is assigned to assess the practicality and cost of making the change
- Development management report back to user management on the cost of the change; user management decide whether to go ahead
- One or more developers are authorized to make copies of components to be modified
- Copies modified. After initial testing, a test version might be released to users for acceptance testing .
- When users are satisfied then operational release authorized – master configuration items updated .



Change control and configuration management

- **Change control**

  – Set of procedures to ensure that changes made only after a consideration of the full impacts.

- **Configuration management**

  – Version control to ensure that all changes are properly recorded and managed – and so that knock-on effects on other projects can be identified.

30. Managing contracts:

Contract administration is the management of contracts made with customers, vendors, partners, or employees.The personnel involved in Contract Administration required to negotiate, support and manage effective contracts are expensive to train and retain. Contract management includes negotiating the terms and conditions in contracts and ensuring compliance with the terms and conditions, as well as documenting and agreeing on any changes or amendments that may arise during its implementation or execution. It can be summarized as the process of systematically and efficiently managing contract creation, execution, and analysis for the purpose of maximizing financial and operational performance and minimizing risk.

CHANGE MANAGEMENT:

There may be occasions where what is agreed in a contract needs to be changed later on. A number of bases may be used to support a subsequent change, so that the whole contract remains enforceable under the new arrangement.

A change may be based on:

- A mutual agreement of both parties to vary the contract, outside the framework of the existing contract. This would be an independent basis for changing the contract.
- A unilateral decision to vary the contract, contemplated and allowed for by the existing contract. This would normally have notice periods for fairness and often the right of the other, especially in consumer contracts, to cease the contractual relationship. Be careful that any one-way imposition of change is contractually justified, otherwise it may be interpreted as a repudiation of the original contract, enabling the other party to terminate the contract and seek damages.
- A bilateral decision to vary the contracting, within the variation or change control process outlined in the existing contract. These are often called change control provisions.

31. Types of contract

- Completed software application
  - Bespoke
  - Off-the shelf

        - Customized off-the shelf
- Payment calculation
        - Fixed price contracts
        - Time and material contracts
        - Fixed price per delivered unit contracts

Fixed price contracts:

- Advantage:
    - Known customer expenditure
    - Supplier motivation
- Disadvantage
    - Higher prices to allow for contingency
    - Difficulties in modifying
    - Upward pressure on the cost changes
    - Thread to system quality

Time and material contracts

- Advantage:
    - Ease of changing requirements
    - Lack of price pressure
- Disadvantage:
    - Customer liability
    - Lack of supplier

Fixed price per unit delivered contracts:

Function point count	*Function design cost per FP*	*Implementation cost per FP*	*Total cost per FP*
Up to 2,000	$242	$725	$967
2,001–2,500	$255	$764	$1,019
2,501–3,000	$265	$793	$1,058
3,001–3,500	$274	$820	$1,094
3,501–4,000	$284	$850	$1,134

Fixed price per unit delivered contracts

- Advantage:
    - Customer understanding
        - Comparability
        - Emerging functionality
        - Supplier efficiency
        - Life cycle change
- Disadvantage:
    - Difficulties with s/w size measurement
    - Changing

Based on contractor selection

- Open tendering
- Restricted
- negotiated

Stages in contract placement

- Requirement analysis

- OR
- Mandatory
- Desirable

Sections in requirement document:

	Section name
1	Introduction
2	A description of any existing systems and the current environment
3	The customer's future strategy or plans
4	System requirements • mandatory • desirable
5	Deadlines
6	Additional information required from potential suppliers

Evaluation plan:

- Draw up plan account to proposals
- Opposed to off the shelf application
- Mandatory requirements are identified
- Value for money

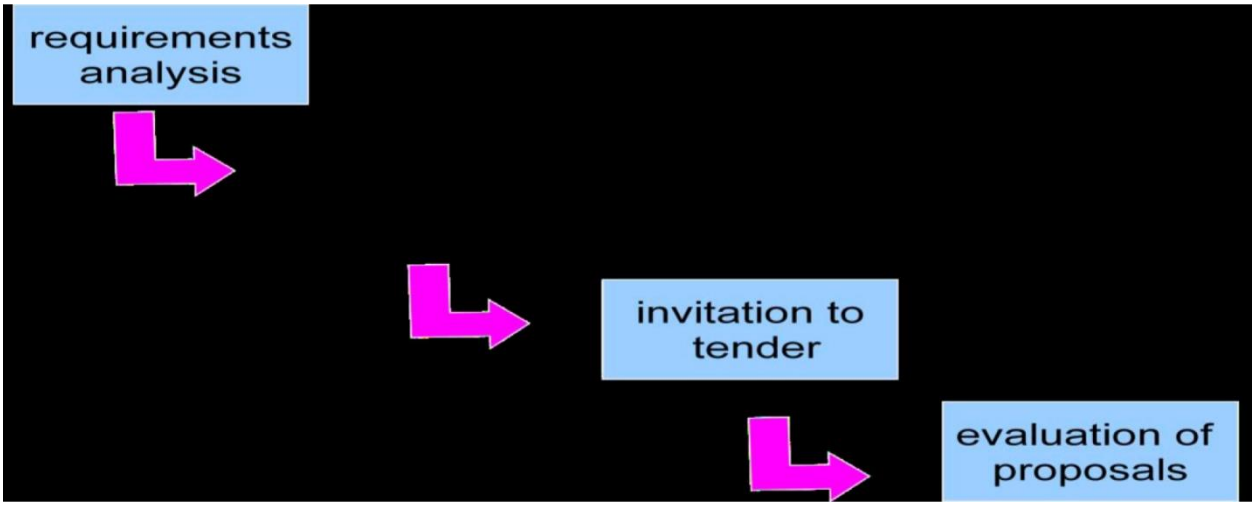
Typical terms of a contract:

- Definitions
- Forms of a agreement
- Goods and services to be supplied
- Services to be provided
- Ownership of the software

Acceptance

- Acceptance tests
- Internal test plans
- Pitfalls
- Very short warranty period

STAGES IN CONTRACT PLACEMENT AND TYPICAL TERMS OF A CONTRACT



Requirements document

• Introduction

• Description of existing system and current environment

• Future strategy or plans
• System requirements -
  – Mandatory/desirable features
• Deadlines
  – Functions in software, with necessary inputs and outputs
  – Standards to be adhered to
  – Other applications with which software is to be compatible
  – Quality requirements e.g. response times
• Additional information required from bidders

Evaluation plan

• How are proposals to be evaluated?
• Methods could include:
  – reading proposals

– Interviews
– Demonstrations
– Site visits
– Practical tests

Invitation to tender (ITT)

• Note that bidder is making an offer in response to ITT
• Acceptance of offer creates a contract
• Customer may need further information
• Problem of different technical solutions to the same problem

Memoranda of agreement (MoA)

• Customer asks for technical proposals
• Technical proposals are examined and discussed
• Agreed technical solution in MoA
• Tenders are then requested from suppliers based in MoA
• Tenders judged on price
• Fee could be paid for technical proposals by customer

32. CONTRACT MANAGEMENT

Contracts should include agreement about how customer/supplier relationship is to be managed e.g.

– Decision points - could be linked to payment
– Quality reviews
– Changes to requirements

33.ACCEPTANCE

• When work is completed, customer needs to carry out acceptance testing.
• Contract may put a time limit to acceptance testing – customer must perform testing bf time expired.
• Part or all payment to the supplier should depend on acceptance testing

Acceptance criteria are defined as “the list of requirements that must be satisfied prior to the customer accepting delivery of the product”.

This document defines the acceptance process, the acceptance criteria, and the review/approval required for customer acceptance of the (Agency name) (project name) project deliverables.

The purpose of this document is to define a standardized Deliverable Review Process, which will provide a structured method to support the Agency Software Verification and Validation Plan (SVVP) to ensure that appropriate, correct, consistent, and complete deliverables are created for the project.

This document describes:

- Goals of the review process;
- Definitions;
- Meeting participants, roles, and responsibilities;
- Review process;
- Dispositions for the review meeting; and
- Review exit criteria.

The primary goal of the Deliverable Review Process is to detect and remove deliverable defects as early as possible in the Software Development Life Cycle (SDLC) process.

Secondary goals to be attained are:

- Consistency with IEEE Std 1028-1997, Standard for Software Reviews;
- Ensure correctness, completeness, consistency, and accuracy of deliverables and products for all life cycle activities within the development process;

Acceptance Process for Project Deliverables

The acceptance process for (Project Name) provides a roadmap for incremental acceptance by the customer of the software application and associated project deliverables at the following key milestones.

- Project Phase Concept Complete
- Phase Requirements Complete
- Phase Design Complete
- Phase Application Ready For Pilot
- Phase Application Ready For Statewide Rollout
- Phase Complete

Milestone	**Deliverables**
Project Phase Concept Complete	Project Initiation and Implementation Document, Software Project Management Plan
Phase Requirements Complete	Software Requirements Specification Template
Phase Design Complete	Software Design Specification
Phase Application Ready For Pilot	Application, Software Test Plan, Software Transition Plan, Training Plan, User's Handbook, Business

Milestone	Deliverables
Phase Application Ready For Statewide Rollout	Application, Software Test Plan, Software Transition Plan, Training Plan, User's Handbook, Business
Phase Complete	Closeout Review, Lessons-learned

Activity	Individual(s) Responsible
Define acceptance criteria for milestones and deliverables in the current project phase	QA Manager, Project Manager, IS Sponsor, and Business Sponsor(s) for
Identify and plan for verification and validation activities necessary to support acceptance criteria	QA Manager and Project Manager
Complete project deliverables for milestone	Project team members responsible for project deliverable
Ensure completion of any necessary verification and validation activities for deliverables	QA Manager and Project Manager

Acceptance Criteria for Milestones and Deliverables

The acceptance criteria in the table below define the conditions under which the PROJECT Business Sponsor(s), the PROJECT IS Sponsor, and the Project Manager agree that they will accept completion of the milestones and deliverables subject to these acceptance criteria.

Milestone	Deliverable	Acceptance Criteria
Project Phase Concept Complete	Project Initiation and Implementation Document	Document has been reviewed and approved by: Prioritized scope and high level requirements have
Project Phase Concept Complete	Software Project Management Plan	Document has been reviewed and approved by: Prioritized scope and high level requirements have
Phase Requirements Complete	Software Requirements Specification	The Software Requirements Specification describes what capabilities the application should have and includes:

34. INTRODUCTION

- OB = organizational behaviour
- There are 3 main concerns in OB; staff selection, staff development, and staff motivation
- We will look at how the project leader can encourage effective group working and decision making while giving purposeful leadership where needed.
- The issues in this chapter have impact at all stages of project planning and execution, particularly in;
    - Some objectives can address health and safety during the project (step 1: Identify project scope and objectives)
    - Although project leaders might have little control over organizational structure, they need to be aware of its implications (step2: Identify project infrastructure)
    - The scope and nature of activities can be set in a way they will enhance staff motivation (Step 4: Identify the products and activities)
    - Many risks to project success relate to staffing (Step 6: Identify activity risks)
    - The qualities of individual members of staff should be taken into account when allocating staff to activities (Step 7: Allocate resources)

35. Understanding behaviour

Behaviours associated with complex and challenging mental health, dementia or other neurological conditions include aggression, wandering, agitation. These apparent changes in the personality of the person with the disease are a major source of distress both to the person who is presenting the behaviours and to those who experience them – the caregiver, the family members, and the service providers in all sectors of the health-care system.

People differ from each other in their needs and values. Group effort eases their task of achieving organizational goals effectively. Human relations can be defined as motivating people in organizations to work as a team. Although human relationships have existed from quite some time in the past, the study of human relations has developed only recently. Social sciences like sociology, psychology, anthropology, economics and political science have contributed to the development of OB and human relations.

Goal of Human Relations

- Create a win-win situation by:
  - satisfying employee needs
  - while achieving organizational objectives
- Win-win situation:
  - occurs when the organization and the employees get what they want

Four Myths of Human Relations

- Myth 1: Technical skills are more important than human relations skills
- Myth 2: Human relations is just common sense
- Myth 3: Diversity is overemphasized
- Myth 4: Leaders are born not made

The Total Person Approach

- Realizes that an organization employs the whole person, not just his or her job skills
- People play many roles
  - throughout their lives
  - throughout each day
- Organizations view employees as total people
- Organizations are trying to give employees a better quality of work life

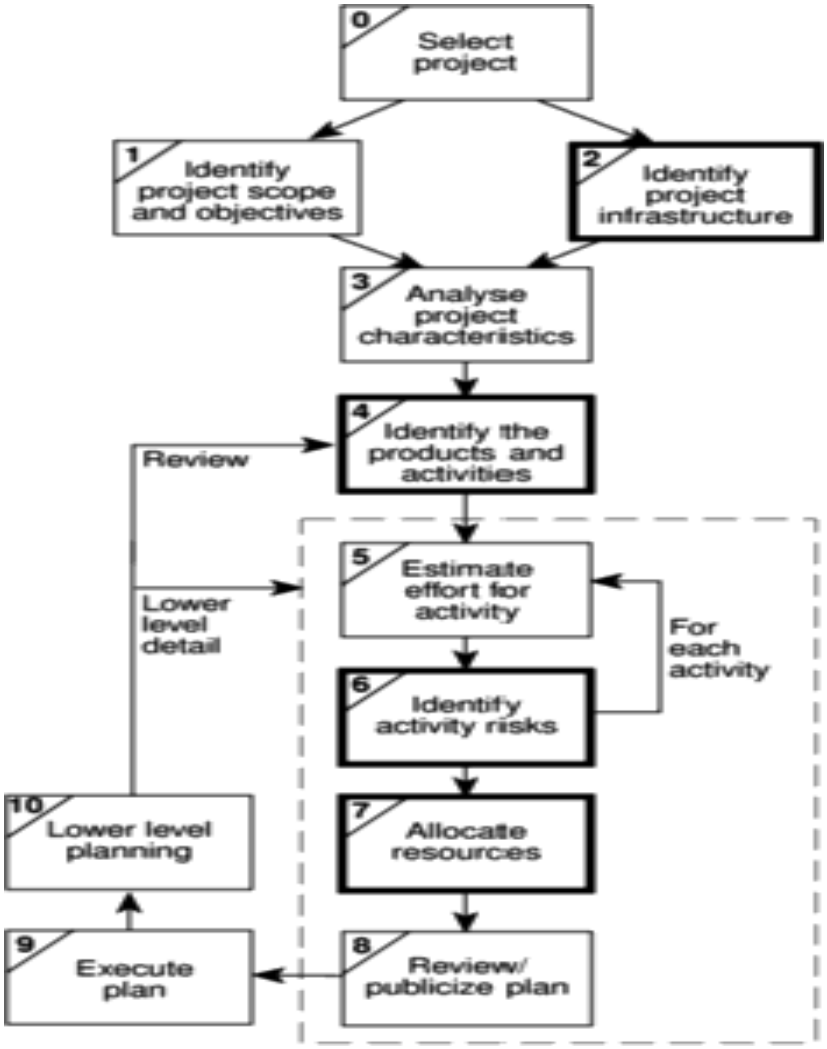
Levels of Behavior

- **Individual behavior –** influences group behavior
- **Group Level Behavior-–** consists of the things two or more people do and say as they interact

Organizational Level Behavior

- Organization – a group of people working to achieve an objective
- Created to produce goods and services for the larger society
- Organizational behavior – the collective behavior of an organization's individuals and groups

STEPWISE FRAMEWORK:



36. ORGANIZATIONAL BEHAVIOUR: A BACKGROUND

- OB was studied by Frederick Taylor in the late 19th and early 20th centuries
- Taylor attempted to analyse the most productive way of doing manual tasks
- Taylor had 3 basic objectives:
    - to select the best people for the job
    - to instruct them in the best methods
    - to give incentives in the form of higher wages to the best workers

- Taylor's view emphasize on the financial basis of staff motivation, however, the other issues of motivation should be encouraged staff not just on such rewards.

- Theory X and Theory Y by Donald McGregordraws attention to the way that expectations influence behaviour

Theory X holds that:

- The average human has an innate dislike of work
- There is a need therefore for coercion, direction and control
- People tend to avoid responsibility

Theory Y holds that:

- Work is as natural as rest or play
- External control and coercion are not the only ways of bringing about effort directed towards and organization's ends
- Commitment to objectives is a function of the rewards associated with their achievement
- The average human can learn to accept and further seek responsibility
- The capacity to exercise imagination and other creative qualities is widely distributed

- One way of judging whether a manager espouses Theory X or Theory Y is to observe how staff react when the boss is absent:
- If there is no discernible change then this is a Theory Y environment;
- If everyone visibly relaxes, it is a Theory X environment

Therefore,

- A "reward" does not have to be a financial reward- it could be something like a sense of achievement
- Theory X and Theory Y illustrated how the state of mind of workers influenced their productivity

37. SELECT HE RIGHT PERSON FOR THE JOB

Taylor stressed " the need for the right person for the job".

Examples of question;

- What sort of characteristics should they be looking for?
- Is an experienced programmer better than a new graduate with a first class mathematics degree?

Recruitment is often an organizational responsibility

- There are 2 types of candidates that are distinguished by Meredith Belbin:

  @eligible candidate

  @suitable candidate

  - Eligible candidates have curriculum vitae (CV) which shows, for example, the 'right' number if years in some previous post and the 'right' paper qualifications.
  - Suitable candidates can actually do the job well
  - A mistake is to select an eligible candidate who is not in fact suitable
  - Thus, Belbin suggests we should try to assess actual skills rather than past experience and provide training to make good minor gaps in expertise
  - And it also has the general approach for recruitment process

38. INSTRUCTION IN THE BEST METHODS

- Create a job specification
- Create a job holder profile
- Obtain applicants
- Examine CVs
- Interviews (e.g. aptitude tests, personality tests and the examination of samples of previous work)
- Other procedures (e.g. medical examination)

What is the purpose of instruction?

- The purpose of instruction is to help people learn. The goal of instructional designers is to make learning easier, quicker, and more enjoyable. Some people view training as a process of finding out who the brightest employees are. But performance in a course is not very highly correlated with the basic ability to be good on the job. We believe that an instructional designer's job is to help everyone to learn and be successful.

Challenge: How to make good instruction?

- The key to improving our instruction is to know what methods of instruction to use when. It's helpful to think of different methods of instruction as different tools for a carpenter. If you only have a hammer, then everything looks like a nail to you. And you won't be able to make a very good piece of furniture. So what we need is a **knowledge base** about methods of instruction to supplement the creative, "art" aspect of training. Such a knowledge base would offer optimal **methods** for given situations.

- But what are the important **situations** that call for different methods? How can we tell what methods (tools) to use when?

What are the relevant kinds of learning?

Perhaps the most important aspect of the situation is the kind of learning that is to be facilitated. Knowing about the kinds of learning helps us to do a better job of teaching them. The most basic distinction is Benjamin Bloom's three domains:

• Cognitive learning (thoughts), such as teaching someone to add fractions.
• Affective learning (feelings, values), such as teaching someone to not want to smoke.
• Physical or motor learning (actions), such as teaching someone to touch type.

39. MOTIVATION

- The taylorist model
- Masloww's hierarchy of needs
- Herzberg's two factor theory
    - Hygiene or maintenance factors
    - Motivators
- The expectancy theory of motivation
    - Expectancy
    - Instrumentality
    - Perceived value

      The third of Taylor's concerns was that of motivating people to work
    - Thus, we are going to look at some models of motivation.
        - The Taylorist model
        - Maslow's hierarchy of needs
        - Herzberg's two-factor theory
        - The expectancy theory of motivation

The Taylorist model:

- Taylor's viewpoint is reflected in the use of piece-rates in manufacturing industries and sales bonuses amongst sales forces.
- Piece-rates can cause difficulties if a new system will change work practices.
- If new technology improves productivity, adjusting piece-rates to reflect this will be a sensitive issue.
- "Piece-rates" are where workers are paid a fixed sum for each item they produce.
- "Day-rates" refer to payment for time worked
- Rewards based on piece-rates need to relate directly to work produced
- So, this model emphasizes on the reward system

Maslow's hierarchy of needs:

- The motivation of individual varies.
- Money is a strong motivator when you are broke
- However, as the basic need for cash is satisfied, other motivations are likely emerge.
- In practice, people are likely to be motivated by different things at different stages of their life .

Herzberg's two-factor theory:

- Some things about a job can make you dissatisfied.
- If the causes of this dissatisfaction are removed, this does not necessarily make the job more exciting
- There are two sets of factors about a job:
    - Hygiene or maintenance factors
    - Motivators

- Hygiene or maintenance factors, which can make you dissatisfied if they are not right, for example the level of pay or the working conditions;

- Motivators, which make you feel that the job is worthwhile, like a sense of achievement or the challenge of the work itself

- A model of motivation developed by Vroom and his colleagues.
- It identifies three influences on motivation:
- expectancy: the belief that working harder will lead to a better performance
- instrumentality: the belief that better performance will be rewarded
- perceived value: of the resulting reward
- Motivation will be high when all three factors are high
- A zero level for any one of the factors can remove motivation

40. HACKMAN JOB CHARACTERISTICS MODEL:

- Managers should group together the elements of tasks to be carried out so that they form meaningful and satisfying assignments.
- Oldham and Hackman suggest that the satisfaction that a job gives is based on 5 factors
- The first three factors make the job 'meaningful' to the person who is doing it
- These three factors:

- **skill variety:** the number of different skills that the job holder has the opportunity to exercise

- **task identify:** the degree to which your work and its results are identifiable as belonging to you

- **task significance**: the degree to which your job has an influence on others

- The other two factors are:

  - **autonomy:** the discretion you have about the way that you do the job

  -**feedback**: the information you get back about the results of your works

- Methods of improving motivation;

  - Set specific goals: these goals need to be demanding and yet acceptable to staff. Involving staff in the setting of goals helps to gain acceptance for them.

  - Provide feedback: Not only do goals have to be set but staff have to have regular feedback about how they are progressing

  - Considering job design: Jobs can be altered to make them more interesting and give staff more feeling of responsibility

- Two measures are often used to enhance job design;

  - job enlargement-> The person doing the job carries out a wider variety of activities. It is opposite of increasing specialization

  - job enrichment -> The job holder carries out tasks that are normally done at a managerial or supervisory level

41. WORKING IN GROUPS:

- A problem with major software projects is that they always involve working in groups, and many people attracted to software development find this difficult.
- It is not easy for people from different backgrounds to work together as a team so it is suggested that teams should go through five basic stages of development.

42. BECOMING A TEAM:

- Forming
- Storming
- Norming
- Performing
- Adjourning

Forming: The members of the group get to know each other and try to set up some ground rules about behaviour

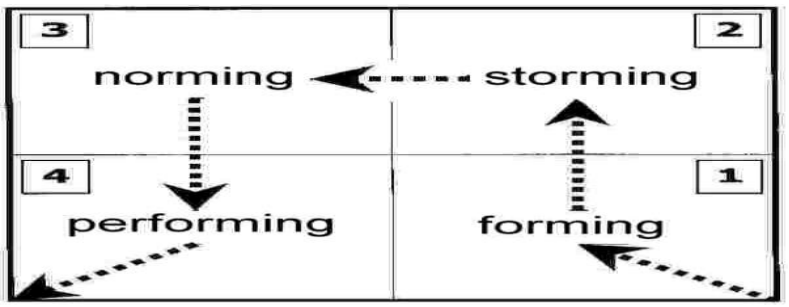
Storming: Conflicts arise as various members of the group try to exert leadership and the group's methods of operation are being established

Norming: Conflicts are largely settled and a feeling of group identity emerges .

Performing: The emphasis is now on the tasks at hand

Adjourning: The group disbands .

Model of Team Development



- Belbin suggests that 'co-ordinator' and 'implementer' are better descriptions than 'chair' and 'team worker'. A new role is added: the 'specialist', the 'techie' who likes to acquire knowledge for its own sake.
- A team need a balance of different types of people:
  - The chair
  - The plant
  - The monitor evaluator
  - The shaper
  - The team worker
  - The resource investigator
  - The complete-finisher
  - The company worker

The chair: not necessarily brilliant leaders but they must be good at running meeting, being calm, strong but tolerant

The plant: someone who is essentially very good at generating ideas and potential solutions to problems

The monitor-evaluator: good at evaluating ideas and potential solutions and helping to selecting the best one

The shaper: rather a worrier, who helps to direct the team's attention to the important issues

The team worker: skilled at creating a good working environment

The resource investigator: adept at finding resources in terms of both physical resources and information

The complete-finisher: concerned with completing tasks

The company worker: a good team player who is willing to undertake less attractive tasks if they are needed for team success

To be a good team member you must be able to:

- time your interventions, e.g. not overwhelm the others in the team;
- be flexible;
- be restrained;
- keep the common goals of the teamin mind all the time.

Group performance:

- Categories
  - Additive tasks
  - Compensatory tasks
  - Disjunctive tasks
  - Conjunctive tasks

43. DECISION MAKING:

- Structured
- Unstructured
  - Some mental obstacles in good decision making
    - Faculty heuristics
    - Escalation of commitment
    - Information overload
  - Group decision making
  - Obstacles

- Measure to reduce disadvantages in group decision making

- Decision can be categorized as being:

  - **structured**: generally relatively simple, routine decisions where rules can be applied in a fairly straightforward way

- **Unstructured:** more complex and often requiring a degree of creativity

Another way to categorize decisions is by the amount if risk and uncertainty that is involved

- To make it more efficient and effective -> training members to follow a set procedure
- Brainstorming techniques can help groups to create more ideas .

44. LEADERSHIP:

- Position power
  - Coercive power
  - Connection power
  - Legitimate power
  - Reward power
- Personal power
  - Expert power
  - Information power
  - Referent power

- Leadership is based on the idea of authority or power
- Power may come either from the person's position (position power), from the person's individual qualities (personal power) or may be a mixture of the two
- Position power;
  - **coercive power**: the ability to force someone to do something by threatening punishment
  - **connection power:** which is based on having access to those who have power
  - **legitimate power:** which is based on a person's title conferring a special status

- **reward power:** where the holder can give rewards to those who carry out tasks to his or her satisfaction

- Personal power;
  - **expert power:** which comes from being the person who is able to do a specialized task
  - **information power:** where the holder has exclusive access to information
  - **referent power**: which is based on the personal attractiveness of the leader

Leadership style:

There are 2 axes: directive vs. permissive and autocratic vs. democratic:

- Directive autocrat
- Permissive autocrat
- Directive democrat
- Permissive democrat

-**directive autocrat:** makes decisions alone, close supervision of implementation

- **permissive autocrat:** makes decision alone, subordinates have latitude in implementation

- **Directive democrat**: makes decisions participatively, close supervision of implementation

- **Permissive democrat:** makes decisions participatively, subordinates have latitude in implementation .

45. Organizational Structures

An **organizational structure** defines how activities such as task allocation, coordination and supervision are directed towards the achievement of organizational aims.[1] It can also be considered as the viewing glass or perspective through which individuals see their organization and its environment.

- **Structure gives members clear guidelines for how to proceed.** A clearly-established structure gives the group a means to maintain order and resolve disagreements.
- **Structure binds members together.** It gives meaning and identity to the people who join the group, as well as to the group itself.
- **Structure in any organization is inevitable** -- an organization, by definition, implies a structure. Your group is going to have some structure whether it chooses to or not. It

might as well be the structure which best matches up with what kind of organization you have, what kind of people are in it, and what you see yourself doing.

It is important to deal with structure early in the organization's development. Structural development can occur in proportion to other work the organization is doing, so that it does not crowd out that work. And it can occur in parallel with, at the same time as, your organization's growing accomplishments, so they take place in tandem, side by side. This means that you should think about structure from the beginning of your organization's life. As your group grows and changes, so should your thinking on the group's structure.

E L E ME N T S O F S T R UC T U RE

While the need for structure is clear, the best structure for a particular coalition is harder to determine. The best structure for any organization will depend upon who its members are, what the setting is, and how far the organization has come in its development.

Regardless of what type of structure your organization decides upon, three elements will always be there. They are inherent in the very idea of an organizational structure.

They are:

- Some kind of governance
- Rules by which the organization operates
- A distribution of work

46. HEALTH AND SAFETY:

- Top-management-safety policy
- Delegation of authorities regarding safety must be clear
- Job descriptions reg-safety
- Deployment of safety officer
- Consultation on safety
- Adequate budgeting for safety costs

Responsibility for safety must be clearly defined at all levels. Some points that will need to be considered include:

- Top management must be committed to the safety policy
- The delegation of responsibilities for safety must be clear
- Those to whom responsibilities are delegated must understand the responsibilities and agree to them

- Deployment of a safety officer and the support of experts in particular technical areas
- Consultation on safety
- An adequate budgeting for safety costs

QUESTION BANK:

Q.1 Explain how cost benefit analysis is evaluated in software project management.

Q.2 How do you access the strategic infrastructure management in project evaluation. Explain with suitable illustration?

Q.3 How does a PERT chart in activity planning while tracking the project. Discuss?

Q.4 Explain the types of contract and their stages in contract placement in detail.

Q.5 How do you prioritize the data collection using earned value analysis? Discuss with suitable illustrations.

Q.6 Discuss with a suitable example "process of selecting the right person for the job" in detail.

Q.7 What is the role of "TEAM" in decision making? Explain it in detail.

Q.8 Explain how risks are handled in project. Give example.

Q.9 Explain the different network planning models. Give example for precedence construction.

Q.10 Discuss the organisational behaviour with example.

Q.11 Discuss the types of contract with example.

Q.12 Describe the steps involved in risk planning.

Q.13 Explain the process of prioritizing monitoring. Give example.

Q.14 Give the methodology used to evaluate risk in a project.

Q.15 Explain the different ways of decision making.

Q.16 Give an example for becoming a team and explain working with in groups with example.

Q.17 Describe the various ways in visualizing the progress of a project.

Q.18 Explain the activities of software project management with example.

Q.19 Illustrate few problems associated with software projects.

Q.20 Discuss step wise project planning with example.

Q.21 Draw and explain typical architecture of project life cycle.

Q.22 Explain the process of constructing a critical path method (network) CPM with example.

Q.23 What is the need of evaluating risk? Explain the risk evaluation process with example.

Q.24 Discuss about risk planning and control.

Q.25 Explain how will you identify the major risk that might affect your project and identify the strategies for minimising the risk.

Q.26 Explain in detail about the types of contract.

Q.27 Explain how the delayed projects back on the track.

Q.28 Explain the Benefits of SCM process and tools.

Explain- Control, Management, Cost savings, Quality.

Q.29 Explain in detail about Building the quality assurance plan.

Q.31 Explain the three COCOMO modes.

Q.32 Briefly explain about software size and reuse estimating

Explain- competencies, SEI CMM and estimating, WBS, size measures, LOC, function points as a unit of size, feature points as a unit of size

Q.33 Explain the following:

A) Work Breakdown Structure (WBS) (8)

B) Short notes on Leader's style (8)

Q.34 Explain the following:

A) Work Breakdown Structure (WBS)-competencies, milestone, approaches to build a WBS.

B) Short notes on Leader's style -Telling, selling, participating, delegating.

Q.35 Briefly explain about PERT and CPM scheduling.

Q.36 Briefly explain about function points to measure the software size. Guidelines for counting Function points

-Count number of functions in each category
-Apply Complexity weighting factors.
-Apply Environmental factors
-Calculate Complexity Adjustment Factor (CAF)
-Compute Adjusted Function Points (AFP)
-Convert to LOC.
-Write the Formulas and explain with example

Q.37 A) Intermediate COCOMO Example:

A10 KLOC embedded mode software product is to perform communications processing functions on a embedded mode. Find effort.-using the formula find effort.

Effort (E)= a*
(Size)b Where a, b
are constants.
Size- thousands of lines of code (KLOC)

E-effort expressed in staff months.

(b). Short notes on Effects of reuse on software size
-new code, modified code, reused code, legacy code

Q.38 Briefly explain about COCOMO: A Regression Model. Explain -COCOMO: A Regression Model with examples

Modes of COCOMO, levels of COCOMO, Advantages and disadvantages of COCOMO.

Q.39 Explain about Project charter and software project Management plan (SPMP). Project charter- definition, Project charter contents.
Software project Management plan- it includes charter, organization, process, work breakdown structure, schedule, budget.
Explain the elements of SPMP

Q.40 Explain in detail about Building the quality assurance plan.

Q.41 Explain about Planning and Organizing for SCM.
Q.42 Briefly explain about the legal issues in software.
HINT:
Product development techniques - Advertising and Consumer, Communications, Contracts, Privacy and Tort.
Project Management skills- Alternative Dispute Resolution, Arbitration, Negotiation, and Mediation.
People Management skills- Contracts, handicap, Employment, Intellectual property