SOFTWARE PROJECT MANAGEMENT

CSE4016

PRIYANKA SINGH

Course Objectives

- Understand Project Management principles while developing software.
- Gain extensive knowledge about the basic project management concepts, framework and the process models.
- Reflect on how management the principles will improve software projects.
- Demonstrate various software design techniques in software.
- Gauge the applicability of process models for a software development project.

UNIT 1- INTRODUCTION SOFTWARE PROJECT MANAGEMENT

Importance of Software Project Management – Activities – Methodologies – Categorization of Software Projects – Setting objectives – Management Principles – Management Control – Project portfolio Management – Cost-benefit evaluation technology – Risk evaluation – Strategic program Management – Stepwise Project Planning.

Text Books

1. Bob Hughes, Mike Cotterell and Rajib Mall: Software Project Management – Fifth Edition, Tata McGraw Hill, New Delhi, 2012.
2. Text book of production management, Shridhara Bhat.K, 1st Edition, Himalaya Publishing House,2012.
3. Industrial Engineering and Management, Khanna.O.P, 2nd Edition, Dhanpat Rai Publications, 2013.
4. Entrepreneurial Development, Jayshree Suresh, 5th Edition, Margham Publications, 2010.

Reference Books

1. Entrepreneurship, Robert D. Hisrich, 6th Edition, Tata McGraw Hill Publications.,2014.
2. Software Product Management and Pricing: Key Success Factors for Software Organization, Hans-Bernd Kittlaus, Peter N. Clough, 2011, Springer Science &Business Media.

Software

- **Software** is more than just a program code.
- A program is an executable code, which serves some computational purpose.
- Software is considered to be collection of executable programming code, associated libraries and documentations.
- Software, when made for a specific requirement is called **software product.**

Engineering on the other hand, is all about developing products, using well-defined, scientific principles and methods.

- Software engineering- Software engineering is an engineering branch associated with development of software product using well-defined scientific principles, methods and procedures.
- Definition (IEEE)- Software engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and work efficiently on real machines.
- The outcome of software engineering is an efficient and reliable software product.
- Software project management has wider scope than software engineering process.

Characteristics of good software

A software product can be judged by what it offers and how well it can be used. This software must satisfy on the following grounds:

- Operational
- Transitional
- Maintenance

- Operational- This tells us how well software works in operations. It can be measured on:
    - Budget
    - Usability
    - Efficiency
    - Correctness
    - Functionality
    - Dependability
    - Security
    - Safety

- Transitional- This aspect is important when the software is moved from one platform to another:
  - Portability
  - Interoperability
  - Reusability
  - Adaptability

- Maintenance- This aspect briefs about how well a software has the capabilities to maintain itself in the ever-changing environment:
  - Modularity
  - Maintainability
  - Flexibility
  - Scalability

- A Software Project is the complete procedure of software development from requirement gathering to testing and maintenance, carried out according to the execution methodologies, in a specified period of time to achieve intended software product.

- The job pattern of an IT company engaged in software development can be seen split in two parts:
  - Software Creation
  - Software Project Management

Software project management

- What is Project Management?

Ans: The methods and regulation used to define goals, plan and monitor tasks and resources, identify and resolve issues, and control costs and budgets for a specific project is known as project management.

- Software project management includes the tools, techniques, and knowledge essential to deal with the growth of software products.
- In Software Project Management, the end users and developers require to know the cost of the project, duration and length.
- It is a process of managing, allocating and timing resources to develop computer software that meets necessities.

It consists of eight tasks:

- Problem Identification
- Problem Definition
- Project Planning
- Project Organization
- Resource Allocation
- Project Scheduling
- Tracking, Reporting and Controlling
- Project Termination

Importance of software project management

All organizations, business and companies manage project, people, costs and tasks on a daily basis. This can prove to be a complicated task.

Having a solution that helps to organize and manage these aspects proves invaluable to a company.

What is Project?

- A project is a sequence of unique, complex, and connected activities having one goal or purpose and that must be completed by a specific time, within budget, and according to specification.

- A project is well-defined task, which is a collection of several operations done in order to achieve a goal (for example, software development and delivery).

- A Project can be characterized as:
  - Every project may has a unique and distinct goal.
  - Project is not routine activity or day-to-day operations.
  - Project comes with a start time and end time.
  - Project ends when its goal is achieved hence it is a temporary phase in the lifetime of an organization.
  - Project needs adequate resources in terms of time, manpower, finance, material and knowledge-bank.

Sequence of Activities

- A project includes a number of activities that must be completed in some particular order, or sequence.
- An activity is a defined chunk of work.
- The chain of the activities is based on technical requirements, not on management concern. i.e.
  - What is needed as input in order to begin working on this activity?
  - What activities produce those as output?

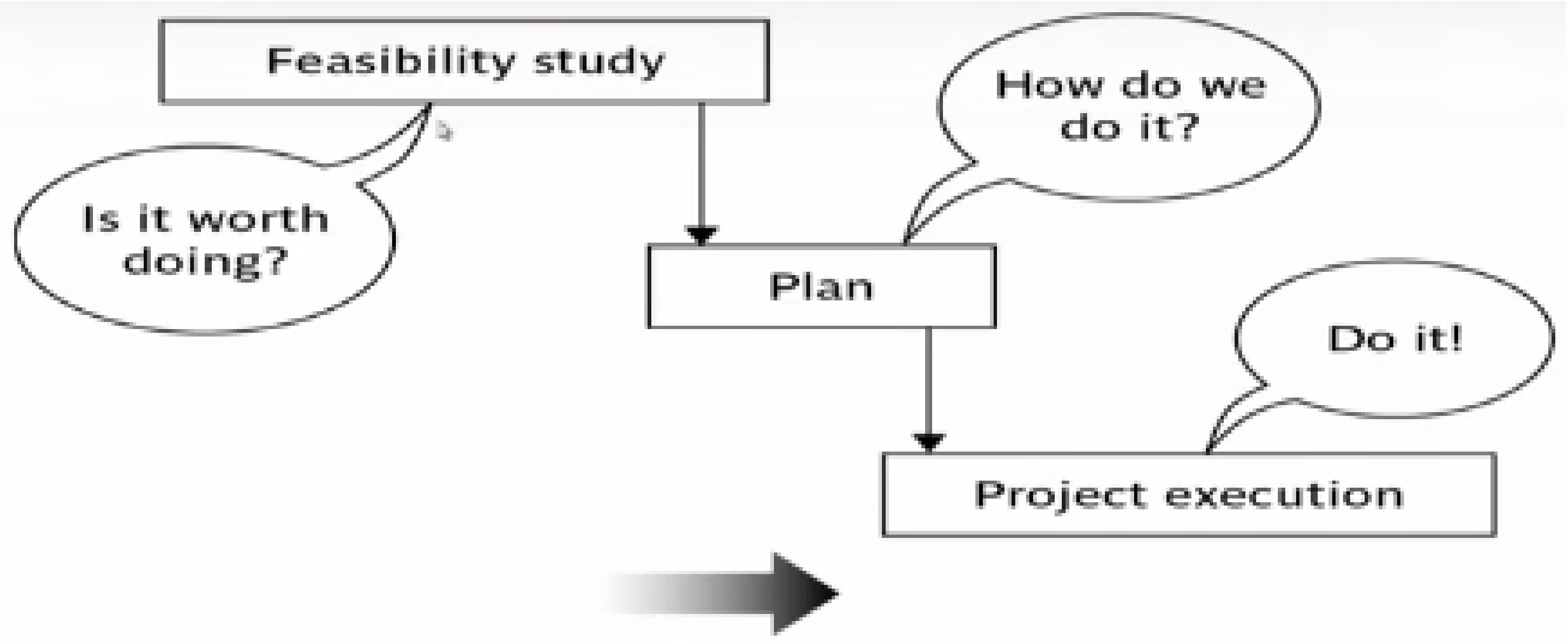
- Unique Activities –
  - The activities in a project must be unique.
  - A project has never happened before, and it will never happen again under the same conditions.

- Complex Activities-
  - The activities that make up the project are not simple, repetitive acts, such as mowing the lawn, painting the house, washing the car, or loading the delivery truck. They are complex.
  - For example, designing an intuitive user interface to an application system is a complex activity.

- Connected Activities-
  - Connectedness implies that there is a logical or technical relationship between pairs of activities.
  - There is an order to the sequence in which the activities that make up the project must be completed. They are considered connected because the output from one activity is the input to another.
  - For example, we must design the computer program before we can program it.

Software Project Manager

- A software project manager is a person who undertakes the responsibility of executing the software project.
- Software project manager is thoroughly aware of all the phases of SDLC that the software would go through.
- Project manager may never directly involve in producing the end product but he controls and manages the activities involved in production.
- A project manager closely monitors the development process, prepares and executes various plans, arranges necessary and adequate resources, maintains communication among all team members in order to address issues of cost, budget, resources, time, quality and customer satisfaction.


Feasibility study
Is it worth doing?
Plan
How do we do it?
Project execution
Do it!

Few responsibilities that a project manager shoulders -

- Managing People
  - Act as project leader
  - Communication with stakeholders
  - Managing human resources
  - Setting up reporting hierarchy etc.
- Managing Project
  - Defining and setting up project scope
  - Managing project management activities
  - Monitoring progress and performance
  - Risk analysis at every phase
  - Take necessary step to avoid or come out of problems
  - Act as project spokesperson

Activities by Software Project Management

- Software project management comprises of a number of activities, which contains planning of project, deciding scope of software product, estimation of cost in various terms, scheduling of tasks and events, and resource management.
- Project management activities may include:
  - Project Planning
  - Scope Management
  - Project Estimation

Project Planning

- Software project planning is task, which is performed before the production of software actually starts.
- It is there for the software production but involves no concrete activity that has any direction connection with software production; rather it is a set of multiple processes, which facilitates software production.

Scope Management

- It defines the scope of project; this includes all the activities, process need to be done in order to make a deliverable software product. Scope management is essential because it creates boundaries of the project by clearly defining what would be done in the project and what would not be done. This makes project to contain limited and quantifiable tasks, which can easily be documented and in turn avoids cost and time overrun.
- During Project Scope management, it is necessary to -
  - Define the scope
  - Decide its verification and control
  - Divide the project into various smaller parts for ease of management.
  - Verify the scope
  - Control the scope by incorporating changes to the scope

Project Estimation

- For an effective management accurate estimation of various measures is a must. With correct estimation managers can manage and control the project more efficiently and effectively.
- Project estimation may involve the following:
  - **Software size estimation-** Software size may be estimated either in terms of KLOC (Kilo Line of Code) or by calculating number of function points in the software. Lines of code depend upon coding practices and function points vary according to the user or software requirement. (The **function point is a "unit of measurement" to express the amount of business functionality** an information system (as a product) provides to a user.)

  - **Effort estimation-** The managers estimate efforts in terms of personnel requirement and man-hour required to produce the software. For effort estimation software size should be known. This can either be derived by managers' experience, organization's historical data or software size can be converted into efforts by using some standard formulae.

- **Time estimation-** Once size and efforts are estimated, the time required to produce the software can be estimated. Efforts required is segregated into sub categories as per the requirement specifications and interdependency of various components of software. Software tasks are divided into smaller tasks, activities or events by ***Work Breakthrough Structure (WBS)***. The sum of time required to complete all tasks in hours or days is the total time invested to complete the project.

- **Cost estimation-** This might be considered as the most difficult of all because it depends on more elements than any of the previous ones. For estimating project cost, it is required to consider -
    - Size of software
    - Software quality
    - Hardware
    - Additional software or tools, licenses etc.
    - Skilled personnel with task-specific skills
    - Travel involved
    - Communication
    - Training and support

Project managers vs Project leader

- Project managers are focused on coordinating the project. Project leaders, on the other hand, are responsible for the overall success and vision for that project. Their vision should guide the rest of the employees in what they should do and how they should do it.
- Project managers are responsible for the tactics that create the end result—a successful project. Their concern is ensuring that the team meets objectives on time and that the project stays on-budget. Project leaders focus on managing the project while inspiring those who work around them to create their vision.
- Project managers have an outline for what they need to do. Project managers manage the budget, staffing and responsibilities of team members to ensure deadlines are met. While project leaders also manage the team's effectiveness, their strategy is to motivate and help generate new ideas to achieve goals.

Activities covered by software project management

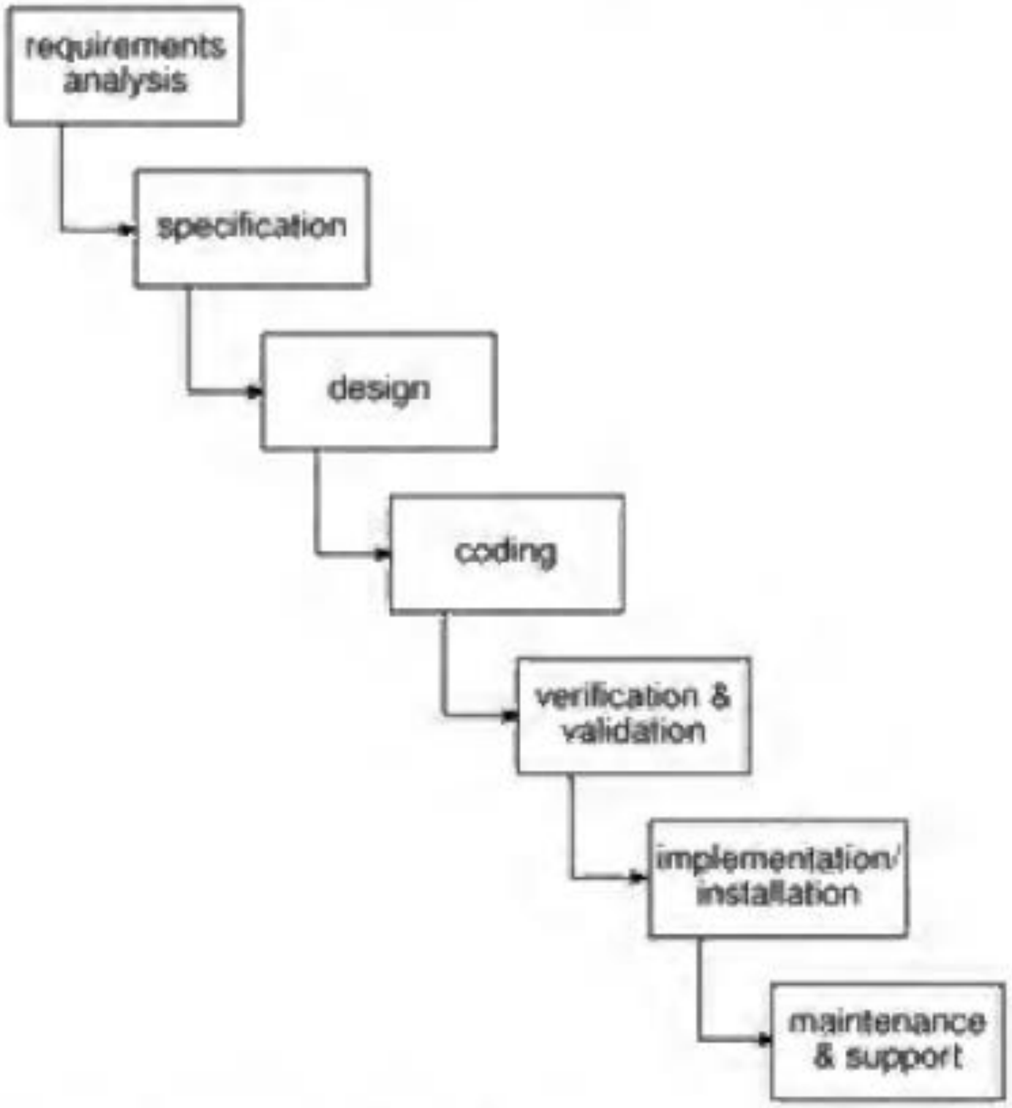
A software project is concerned not only with the actual writing of software. In fact, where a software application is bought in 'off-the-shelf' there might be no software writing as such.

- The feasibility study-This is an investigation to decide whether a prospective project is worth starting.

- Planning- If the feasibility study produces results that indicate that the prospective project appears viable, then plaining of the project can take place. In fact, for a large project, we would not do all our detailed planning right at the beginning.

- Project execution- The project can now be executed, Individual projects are likely to differ considerably but a classic project life-cycle

Typical project life cycle



Some ways of categorizing software projects

Information systems versus embedded systems

A distinction may be made between information systems and embedded systems. Very crudely, the difference is that in the former case the system interfaces with the organization, whereas in the latter case the system interfaces with a machine!

A stock control system would be an information system that controls when the organization reorders stock.

An embedded, or process control, system might control the air conditioning equipment in a building.

Objectives versus products

- Projects may be distinguished by whether their aim is to produce a product or to meet certain objectives.
- A project might be to create a product, the details of which have been specified by the client. The client has the responsibility for justifying the product. On the other hand, the project might be required to meet certain objectives.
- There might be several ways of achieving these objectives in contrast to the constraints of the product-driven project.
- One example of this is where a new information system is implemented to improve some service to users inside or outside an organization. The subject of an agreement would be the level of service rather than the characteristics of a particular information system..

Problems with software projects

- poor estimates and plans;
- lack of quality standards and measure
- lack of guidance about making organizational decision
- lack of techniques to make progress visible
- poor role definition - who does what?
- incorrect success criteria,

Below is a list of the problems identified by a number of students on a degree course in Computing and Information Systems who had just completed a year's industrial placement:

- inadequate specification of work
- management ignorance of IT
- lack of knowledge of application area.
- lack of standards
- lack of up-to-date documentation
- preceding activities not completed on time - including late delivery of equipment
- lack of communication between users and technicians
- lack of communication leading to duplication of work
- lack of commitment - especially when a project is lied to one person who then moves
- narrow scope of technical expertise
- changing statutory requirements
- changing Malware environment
- deadline pressure
- lack of quality control
- remote management
- lack of training.

Purpose of project management and setting objectives

- The purpose of project management is to foresee or predict as many dangers and problems as possible; and to plan, organize and control activities so that the project is completed as successfully as possible in spite of all the risks.
- Project management can involve the following activities: planning - deciding what is to be done; organizing - making arrangements; staffing - selecting the right people for the job; directing - giving instructions; monitoring - checking on progress; controlling - taking action to remedy hold ups; innovation - coming up with new solutions; representing - liaising with users.

Setting Objectives

- Effective objectives in project management are specific.
- A specific objective increases the chances of leading to a specific outcome.
- Therefore objectives shouldn't be vague, such as "to improve customer relations," because they are not measurable.
- Objectives should show how successful a project has been, for example "to reduce customer complaints by 50%" would be a good objective.
- The measure can be, in some cases, a simple yes or no answer, for example, "did we reduce the number of customer complaints by 50%?"

Objectives can often be set under three headings:

1. Performance and Quality

The end result of a project must fit the purpose for which it was intended.

2. Budget

The project must be completed without exceeding the authorized expenditure

3. Time to Completion

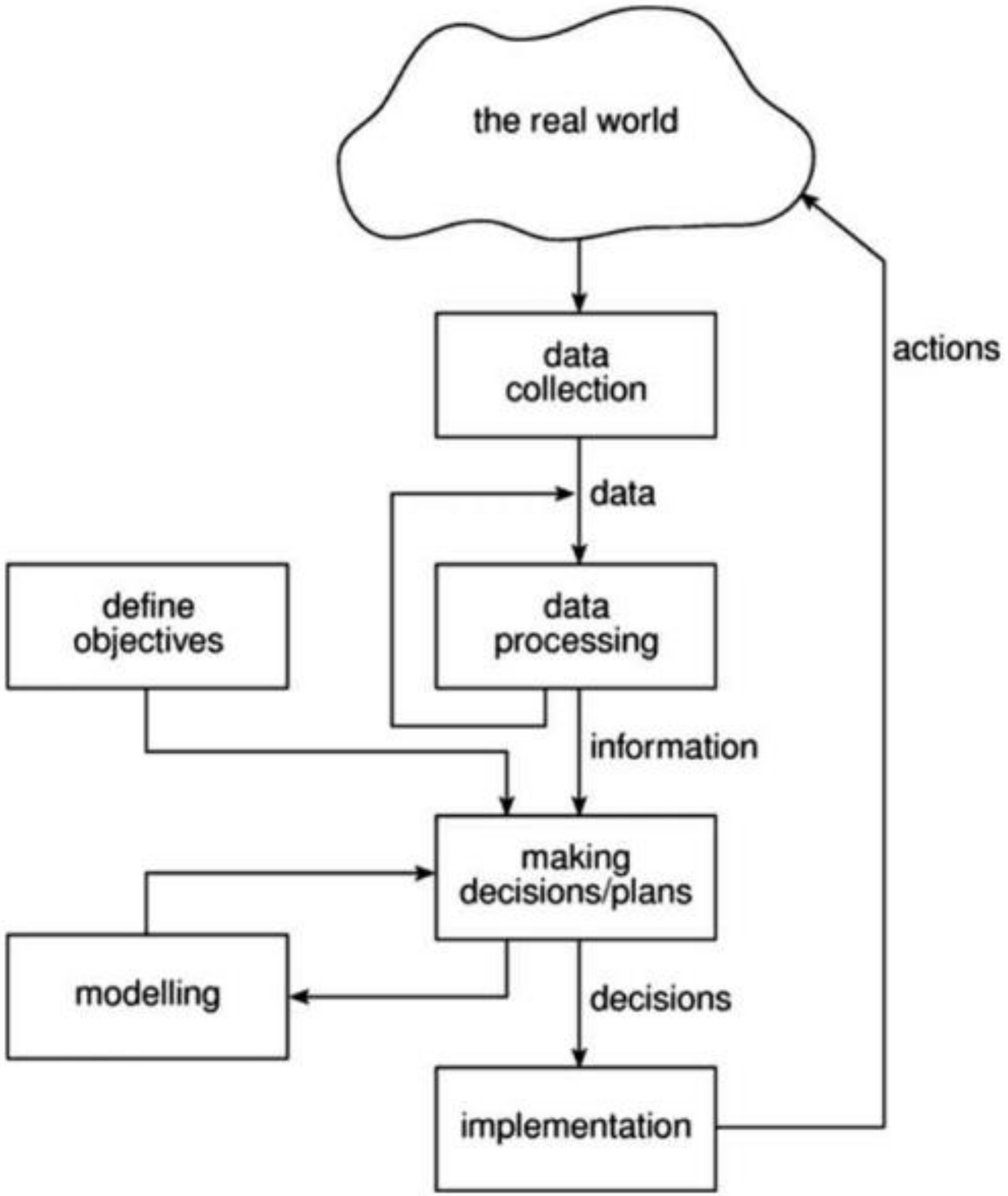
Actual progress has to match or beat planned progress. All significant stages of the project must take place no later than their specified dates, to result in total completion on or before the planned finish date.

Management control

The project control cycle-

- **Control** is a function of management which helps to check errors in order to take corrective actions.
- This is done to minimize deviation from standards and ensure that the stated goals of the organization are achieved in a desired manner.
- According to modern concepts, control is a foreseeing action; earlier concepts of control were only used when errors were detected.
- Control in management includes setting standards, measuring actual performance and taking corrective action and decision making.

Project control cycle



Principles of Project Management

- The primary challenge of project management is to achieve all of the project goals and objectives while honoring the pre-defined constraints.
- The primary constraints are scope, time, quality, and budget. The secondary—and more ambitious—challenge is to optimize the allocation of necessary inputs and integrate them to meet pre-defined objectives.
- For a successful project, the following project management principles are necessary assets when charting a path to completion.
    - Project structure
    - Definition phase
    - Clear goals
    - Transparency about project status
    - Risk recognition
    - Managing project disturbances
    - Responsibility of the project manager
    - Project success

Project Structure-

- Project Goal - An answer to the question “What has to be done” is usually a good starting point when setting a project goal.
- Project Timeline and Order- A flowchart is a powerful tool to visualize the starting point, the endpoint, and the order of work packages in a single chart.
- Project Milestones- Milestones define certain phases of your project and the corresponding costs and results. Milestones represent decisive steps during the project. They are set after a certain number of work packages that belong together. This series of work packages leads to the achievement of a sub-goal.

- **Definition Phase** - The definition phase is where many projects go wrong. This can happen when no clear definition, or when the definition is muddled due to the involvement of too many stakeholders.
- **Clear Goals** - The project manager is responsible for the achievement of all project goals. These goals should always be defined using the SMART paradigm (specific, measurable, ambitious, realistic, time-bound).
- **Transparency About the Project Status-** Your flowcharts, structure plan, and milestone plan are useful tools to help you stay on track. As a project manager, you should be able to present a brief report about the status of the project to your principal or stakeholders at each stage of the project. At such meetings, you should be able to give overviews about the costs, the timeline, and the achieved milestones.

- **Risk Recognition-** It's the duty of the project manager to evaluate risks regularly. You should come into every project with the knowledge that all projects come with a variety of risks. The sooner you identify these risks, the sooner you can address negative developments.

- **Managing Project Disturbances-** It's not very likely that you have enough personal capacity to identify every single risk that may occur. Instead, work to identify the big risks and develop specific strategies to avoid them. Even if you're no visionary, you should rely on your skill set, knowledge, and instincts in order to react quickly and productively when something goes wrong.

- **Responsibility of the Project Manager**

The Project Manager is responsible for communication, including status reporting, risk management, and escalation of issues that cannot be resolved in the team—and generally ensuring the project is delivered within budget, on schedule, and within scope.

- **Project Success**

Project success is a multi-dimensional construct that can mean different things to different people. It is best expressed at the beginning of a project in terms of key and measurable criteria upon which the relative success or failure of the project may be judged.

- For example, some generally used success criteria include:
  - Meeting key project objectives such as the business objectives of the sponsoring organization, owner or user.
  - Eliciting satisfaction with the project management process, i.e., the deliverable is complete, up to standard, is on time and within budget.
  - Reflecting general acceptance and satisfaction with the project's deliverable on the part of the project's customer and the majority of the project's community at some time in the future.

Project portfolio Management

- When there are many projects run by an organization, it is vital for the organization to manage their project portfolio.
- This helps the organization to categorize the projects and align the projects with their organizational goals.
- Project Portfolio Management (PPM) is a management process with the help of methods aimed at helping the organization to acquire information and sort out projects according to a set of criteria.
- Project Portfolio Management is the centralized management of the processes, methods, and technologies used by project managers and project management offices to analyze and collectively manage current or proposed projects based on numerous key characteristics

Objectives of Project Portfolio Management

- The need to create a descriptive document, which contains vital information such as name of project, estimated timeframe, cost and business objectives.
- The project needs to be evaluated on a regular basis to ensure that the project is meeting its target and stays in its course.
- Selection of the team players, who will work towards achieving the project's objectives.

Benefits of Project Portfolio Management

The following benefits can be gained through efficient project portfolio management:

- Greater adaptability towards change.
- Constant review and close monitoring brings about a higher return.
- Identification of dependencies is easier to identify.
- Advantage over other competitors (competitive advantage).
- Helps to concentrate on the strategies, which will help to achieve the targets rather than focusing on the project itself.
- The mix of both IT and business projects are seen as contributors to achieving the organizational objectives.

The answers to these questions will determine the success of the implementation of the project.

Cost Benefit Analysis Methodology

- Cost benefit analysis is a method that facilitates decision makers of companies or institutions to evaluate potential outcomes and choose technologies to achieve these outcomes.

- Cost-benefit analysis (CBA) provides a means for systematically comparing the value of outcomes with the value of resources achieving the outcomes required. It measures the economic efficiency of the proposed technology or project. When there are many options to consider during a decision-making task, it is useful to evaluate the options with a common metric.

- In situations in which large amounts of money are at stake, the presentation of a cost-benefit analysis is the preferred way to demonstrate the reasoning behind investments.

- For the application of CBA, inputs may be divided into parameter values and benefit and cost values. Parameters include the discount rate, the future rates of economic growth, the future rates of inflation and the estimations about the future rates of technological change.

Factors for a reliable Cost Benefit Analysis

- The CBA time period should match the system life cycle.
- The analysis has to include at least three alternative solutions that consider alternative ways of fulfilling the demanded project.
- In the end of the analysis the decision maker should do a sensitivity analysis for the costs and the benefits considered during the previous steps. Sensitivity analysis identifies those input parameters that have the greatest influence on the outcome, repeats the analysis with different input parameter values, and evaluates the results to determine which, if any, input parameters are sensitive. If a relatively small change in the value of an input parameter changes the alternative selected, then the analysis is considered to be sensitive to that parameter.

A CBA application includes the following stages:

- **General description of the project**
- **List of alternative scenarios**
- **Identify Benefits and Costs** (*lists the exact benefits and costs met in each of the alternative scenarios. Cost include- Activities and Resources, Cost Categories, Personnel Costs, Direct and Indirect Costs (Overhead), Depreciation, and Annual Costs. Benefits are the services, capabilities, and qualities of each alternative system*)
- **Schedule Benefits and Costs** (identifies the value of each benefit and cost for each year through the life cycle of the decision beginning from Year 0, which is the start of the decision life)
- **Comparison of alternatives** (with tables and graphs to facilitate decision making.)
- **Sensitivity Analysis** (define how sensitive the results are to changes in the costs and benefits)

Most costs are relatively easy to identify and quantify in approximate monetary terms. It is helpful to categorize costs according to where they originate in the life of the project.

- ***Development costs*** - include the salaries and other employment costs of the staff involved in the development project and all associated costs.
- ***Setup costs*** - include the costs of putting the system into place. These consist mainly of the costs of any new hardware and ancillary equipment but will also include costs of file conversion, recruitment and staff training.
- ***Operational costs*** - consist of the costs of operating the system once it has been installed.

Benefits, on the other hand, are often quite difficult to quantify in monetary terms even once they have been identified. Benefits may be categorized as follows,

- ***Direct benefits*** - these accrue directly from the operation of the proposed system. These could, for example, include the reduction in salary bills through the introduction of a new, computerized system.
- ***Assessable indirect benefits*** - these are generally secondary benefits, such as increased accuracy through the introduction of a more user-friendly screen design where we might be able to estimate the reduction in errors, and hence costs, of the proposed system.
- ***Intangible benefits:*** these are generally longer term or benefits that are considered very difficult to quantify. Enhanced job interest can lead to reduced staff turnover and, hence, lower recruitment costs.

Risk evaluation

Dealing with uncertainty:

- project A might appear to give a better return than B but could be riskier
- Could draw up draw a project risk matrix for each project to assess risks
- For riskier projects could use higher discount rates

Example of a project risk matrix

Risk	*Importance*	*Likelihood*
Software never completed or delivered	H	—
Project cancelled after design stage	H	—
Software delivered late	M	M
Development budget exceeded ≤ 20%	L	M
Development budget exceeded > 20%	M	L
Maintenance costs higher than estimated	L	L
Response time targets not met	L	H

Risk evaluation

- A software risk can be of two types
  (a) internal risks that are within the control of the project manager and
  (2) external risks that are beyond the control of project manager.

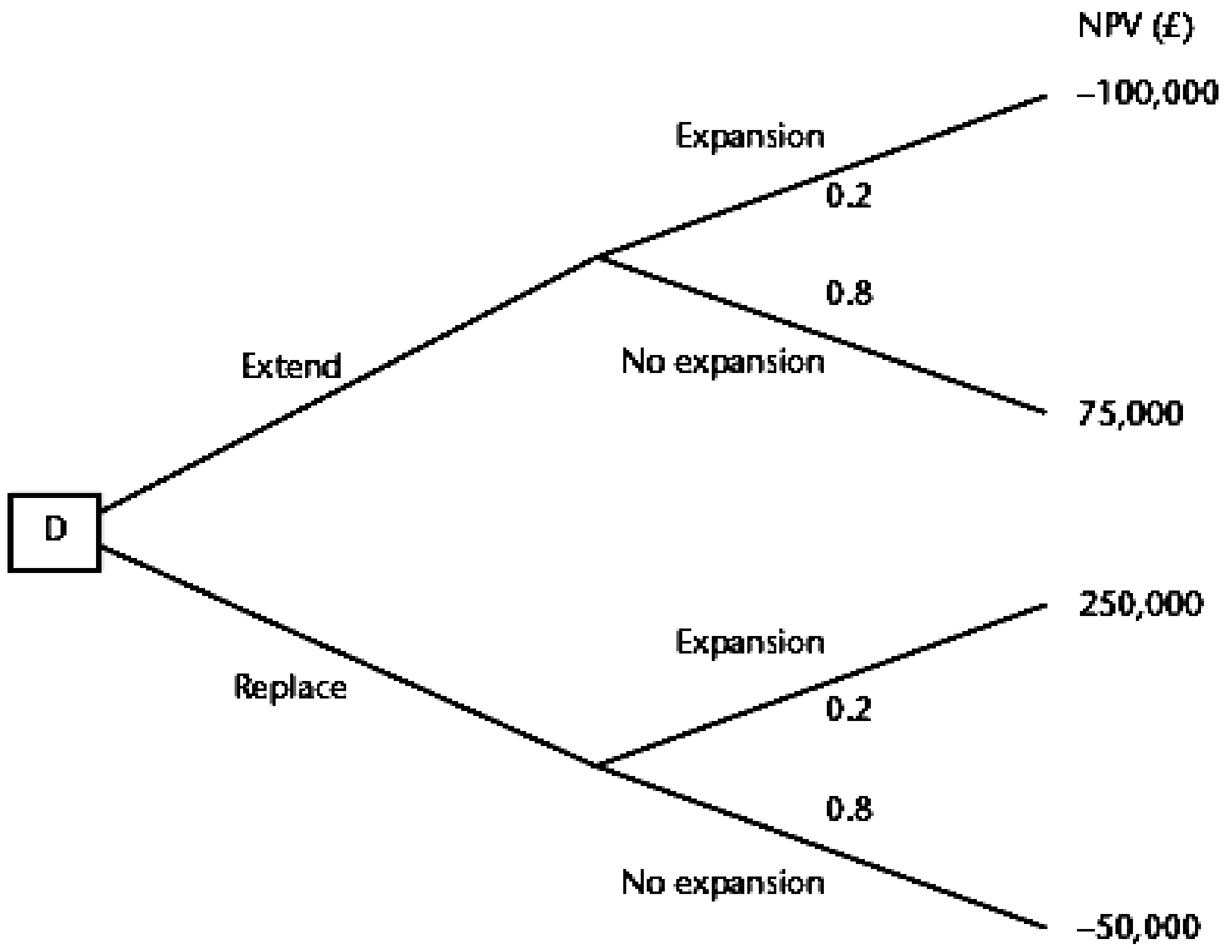
Risk management is carried out to:

- Identify the risk
- Reduce the impact of risk
- Reduce the probability or likelihood of risk
- Risk monitoring

- For risk evaluation:
    - Ranking the risks &
    - Determining the corresponding risk reduction strategies
- Because all risks are not equal in their impact to a project, you need to decide which ones to ignore and which ones to manage. You can prioritize risks by assigning risk priority numbers (RPNs) to them. There are three factors that contribute to the RPN:
    - The probability that the risk may occur
    - The severity of the effect the risk could have on the project
    - The ability to detect ahead of time when the risk happens

Risk evaluation

Decision trees



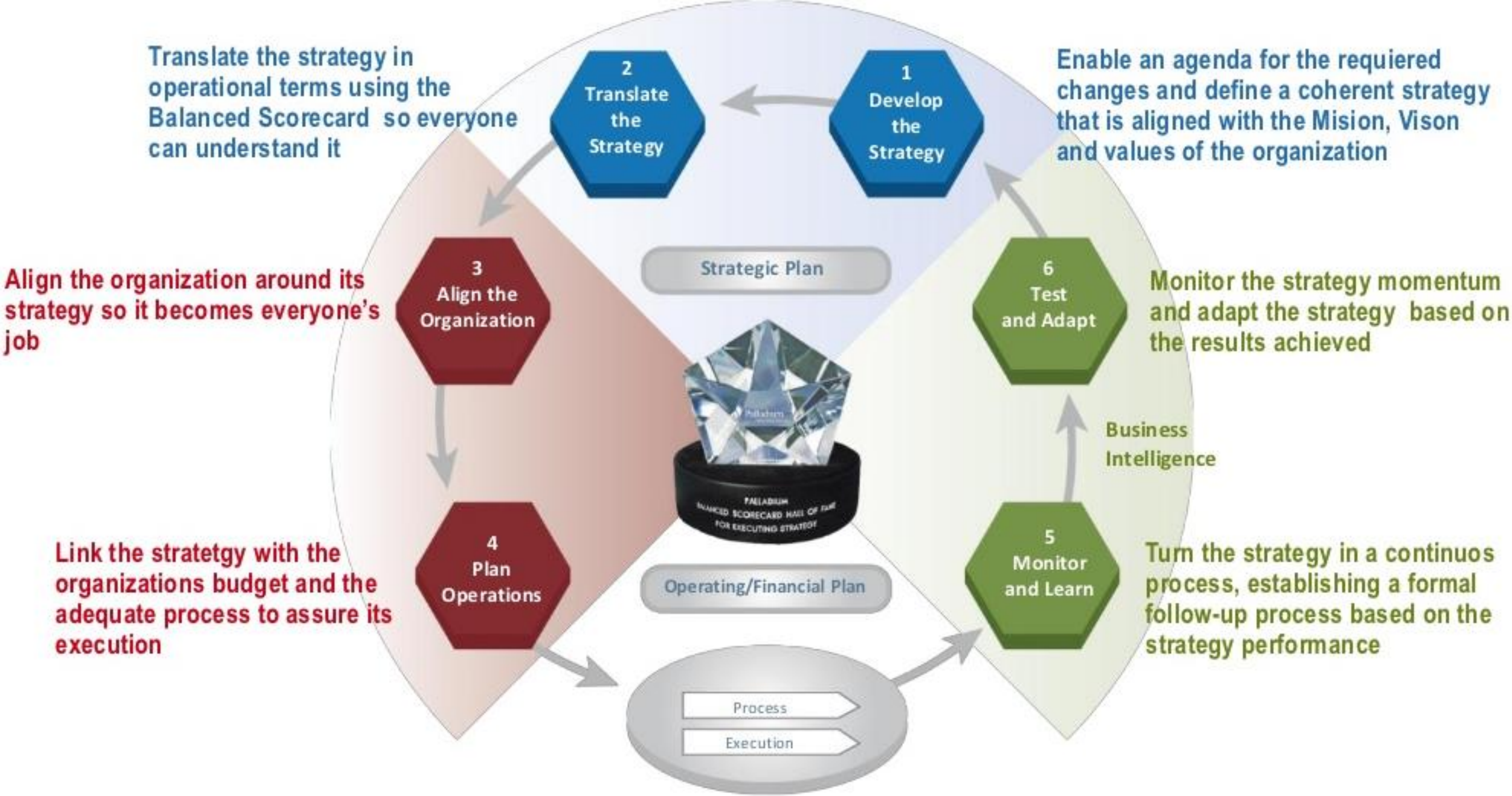
Strategic program Management

Program Management-

- A program consists of interconnected projects that serve some strategic goal.
- Let’s take an example, an organization plans to develop software for accounting purpose and it wants to market it to the right target audience. In this case, there would be two different projects. One project will be for the development of the software and another project will be to market the software. Let’s say, the strategic goal is to increase the customer base of the organization.
- A Program will comprise of these two projects with a goal to increase the customer base.

- Program Management thus refers to the specific techniques, knowledge, and skills to manage such a program.
- In real scenarios, a program could consist of a large number of projects and management of such a program would be highly complicated.
- The essential parts of program management would be to monitor and control the interdependencies among projects in the program
- Strategic Program Management will help you achieve strategic business outcomes and deliver the improved capital efficiency for mega-projects.

- Two thirds of all large engineering and construction projects are significantly under performing, exceeding budgets and schedules approved as part of final investment decisions by 20% or more.
- But, if you prepare your organization, plan your project right and ensure strong, continuous alignment around your strategic business objectives, it doesn't have to be so.
- The challenges of today's mega-projects are largely management and organizational.
- At Strategic Program Management you improve the performance of today's and tomorrow's mega-projects.


Translate the strategy in operational terms using the Balanced Scorecard so everyone can understand it
2
Translate the Strategy
1
Develop the Strategy
Enable an agenda for the requiered changes and define a coherent strategy that is aligned with the Mision, Vison and values of the organization
Align the organization around its strategy so it becomes everyone's job
3
Align the Organization
Strategic Plan
6
Test and Adapt
Monitor the strategy momentum and adapt the strategy based on the results achieved
Business Intelligence
Link the stratetgy with the organizations budget and the adequate process to assure its execution
4
Plan Operations
Operating/Financial Plan
5
Monitor and Learn
Turn the strategy in a continuos process, establishing a formal follow-up process based on the strategy performance
Process
Execution

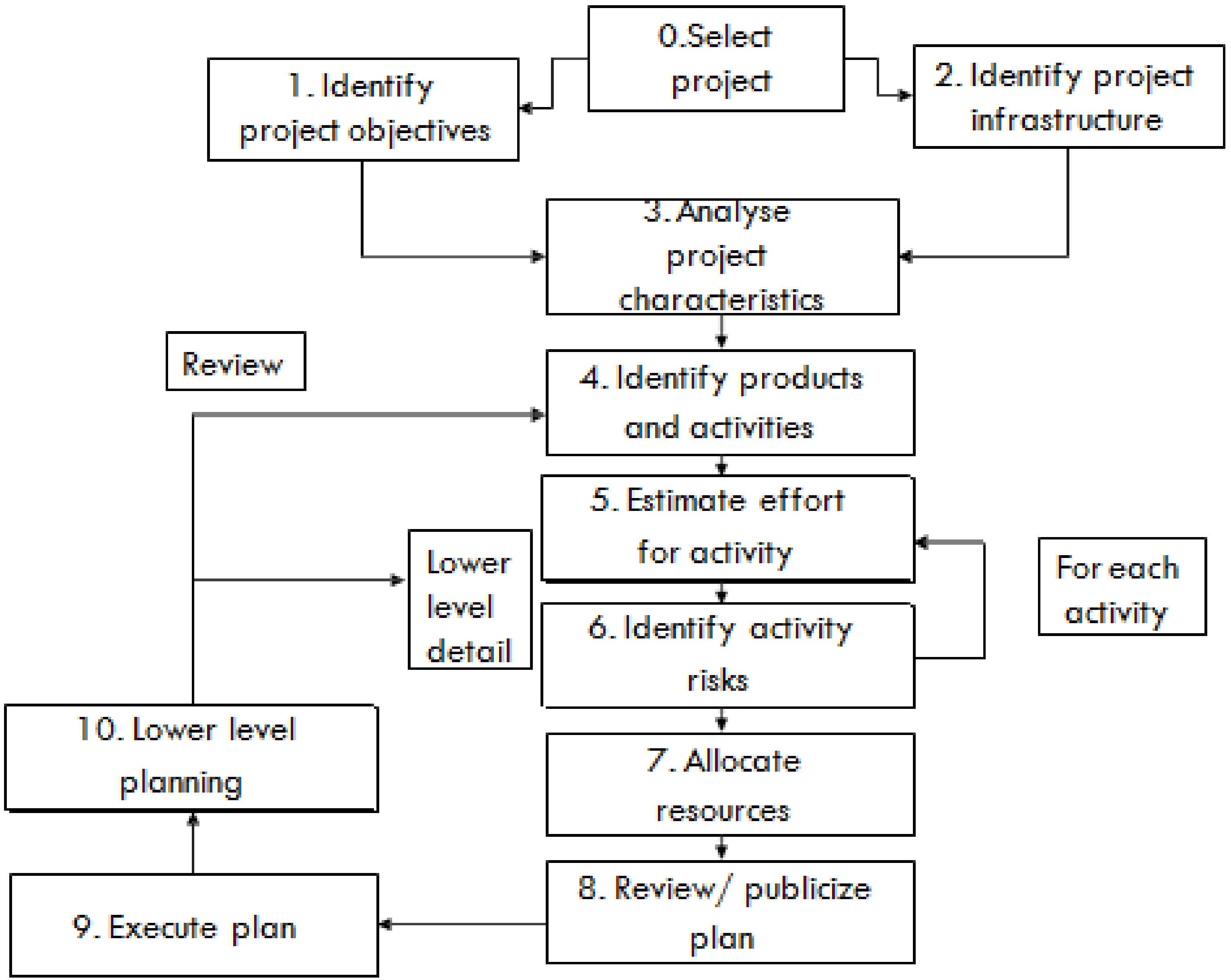
In summary-

- Most organizations have a strategy
- 90% of organizations fail to execute
- Strategy Execution can be a core competence
- Program Management plays a key role in effective Strategy Execution

Step Wise Project Planning

- Practicality
  - tries to answer the question ‘what do I do now?’
- Scalability
  - useful for small project as well as large
- Range of application
- Accepted techniques
  - e.g. borrowed from PRINCE etc

Step Wise' - an overview



A project scenario

- Hardware/software engineering company (C++ language of choice)
- teams are selected for individual projects - some friction has been found between team members
- HR manager suggests psychometric testing to select team
- Software package to be used to test staff
- Visual basic suggested as a vehicle for implementation
- usability is important - decision to carry out usability tests

Step 1 establish project scope and objectives

- 1.1 Identify objectives and measures of effectiveness
  - ‘how do we know if we have succeeded?’
- 1.2 Establish a project authority
  - ‘who is the boss?’
- 1.3 Identify all stakeholders in the project and their interests
  - ‘who will be affected/involved in the project?’
- 1.4 Modify objectives in the light of stakeholder analysis
  - ‘do we need to do things to win over stakeholders?’
- 1.5 Establish methods of communication with all parties
  - ‘how do we keep in contact?’

Back to the scenario

- Project authority
  - should be a project manager rather than HR manager?
- Stakeholders
  - project team members to complete on-line questionnaires: concern about results?
- Revision to objectives
  - provide feedback to team members on results

Step 2 Establish project infrastructure

- 2.1 Establish link between project and any strategic plan
  - ‘why did they want the project?’
- 2.2 Identify installation standards and procedures
  - ‘what standards do we have to follow?’
- 2.3. Identify project team organization
  - ‘where do I fit in?’

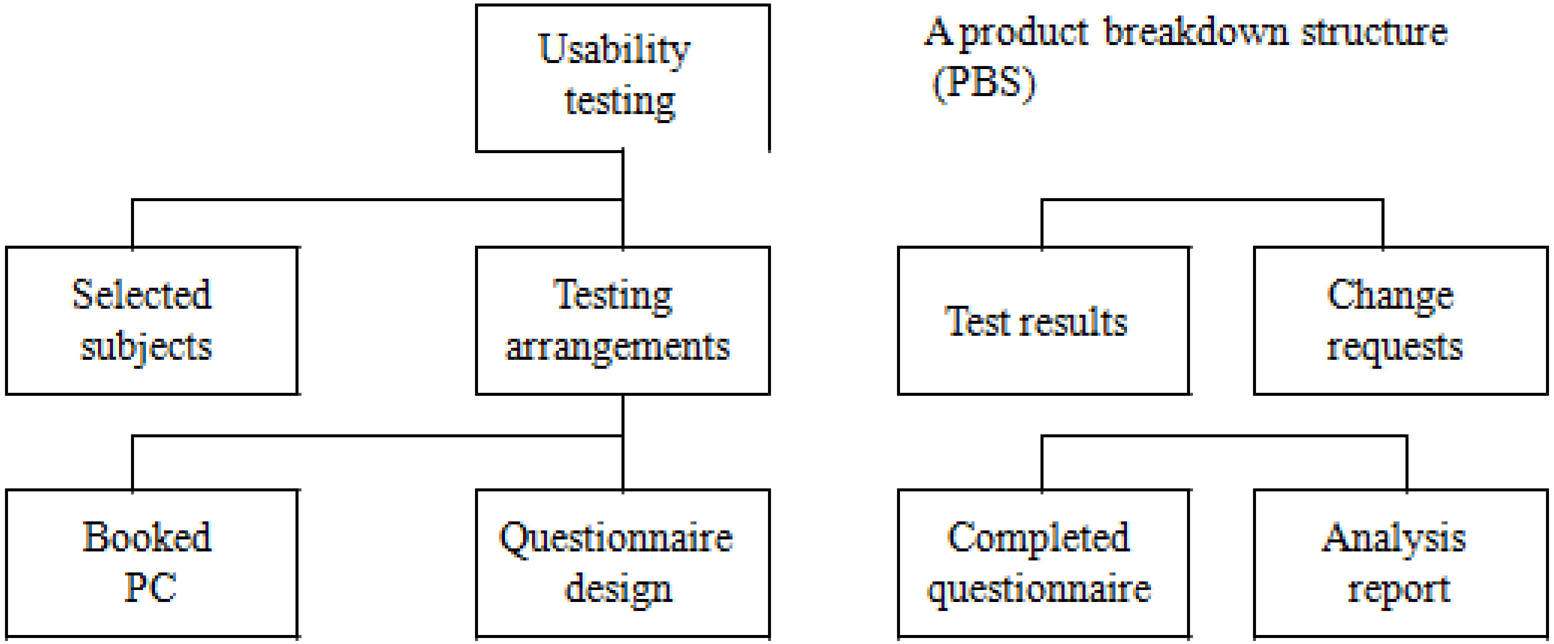
Step 3 Analysis of project characteristics

- 3.1 Distinguish the project as either objective or product-based.
  - Is there more than one way of achieving success?
- 3.2 Analyse other project characteristics (including quality based ones)
  - what is different about this project?
- Identify high level project risks
  - 'what could go wrong?'
  - 'what can we do to stop it?'
- Take into account user requirements concerning implementation
- Select general life cycle approach
  - waterfall? Increments? Prototypes?
- Review overall resource estimates
  - 'does all this increase the cost?'

Back to the scenario

- Objectives vs. products
  - use paper questionnaire then input results of the analysis?
- Some risks
  - team members worried about implications and do no co-operate
  - project managers unwilling to try out application
  - Developer not familiar with features of VB
- Answer? - evolutionary prototype?

Step 4 Identify project products and activities



Products

- The result of an activity
- Could be (among other things)
  - physical thing ('installed pc'),
  - a document ('logical data structure')
  - a person ('trained user')
  - a new version of an old product ('updated software')
- The following are NOT normally products:
  - activities (e.g. 'training')
  - events (e.g. 'interviews completed')
  - resources and actors (e.g. 'software developer') - may be exceptions to this
- Products CAN BE *deliverable* or *intermediate*

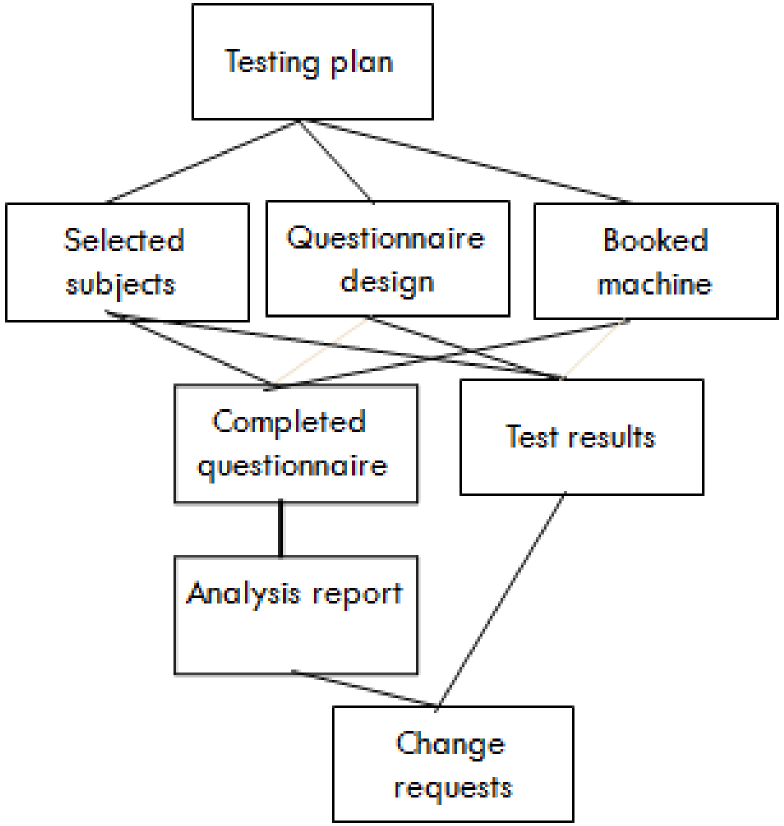
Product description (PD)

- Product identity
- Description - what is it?
- Derivation - what is it based on?
- Composition - what does it contain?
- Format
- Relevant standards
- Quality criteria

Create a PD for 'test data'

Step 4 continued

4.2 document Generic Product flows

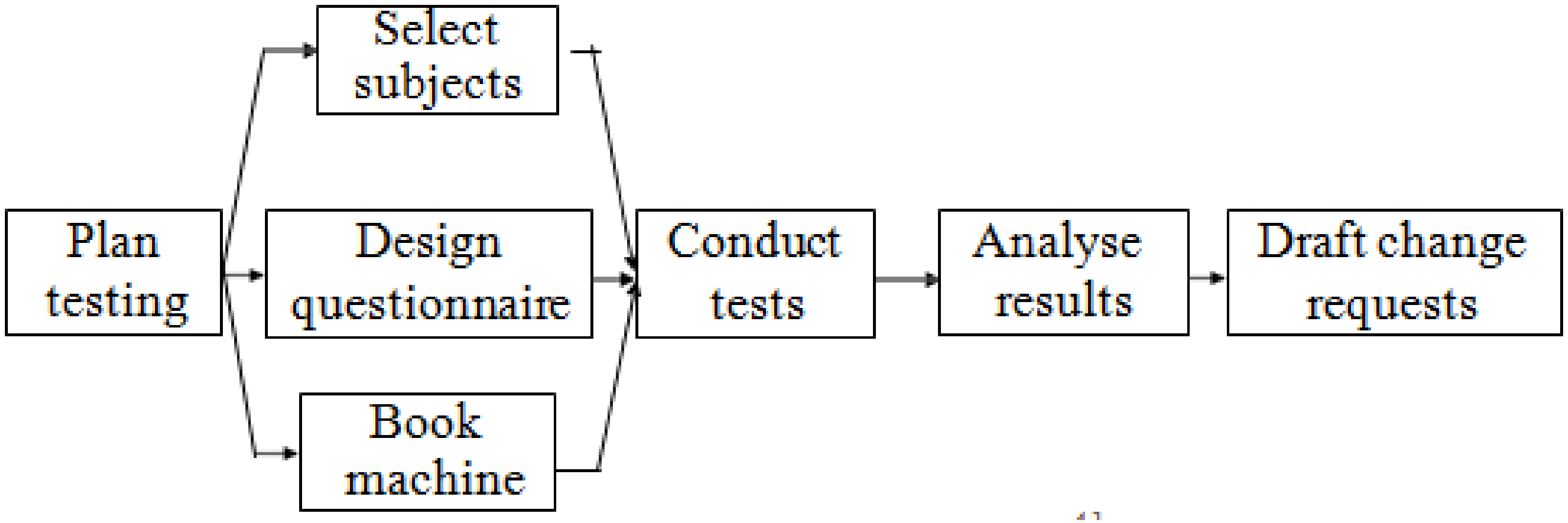


Step 4.3 Recognize product instances

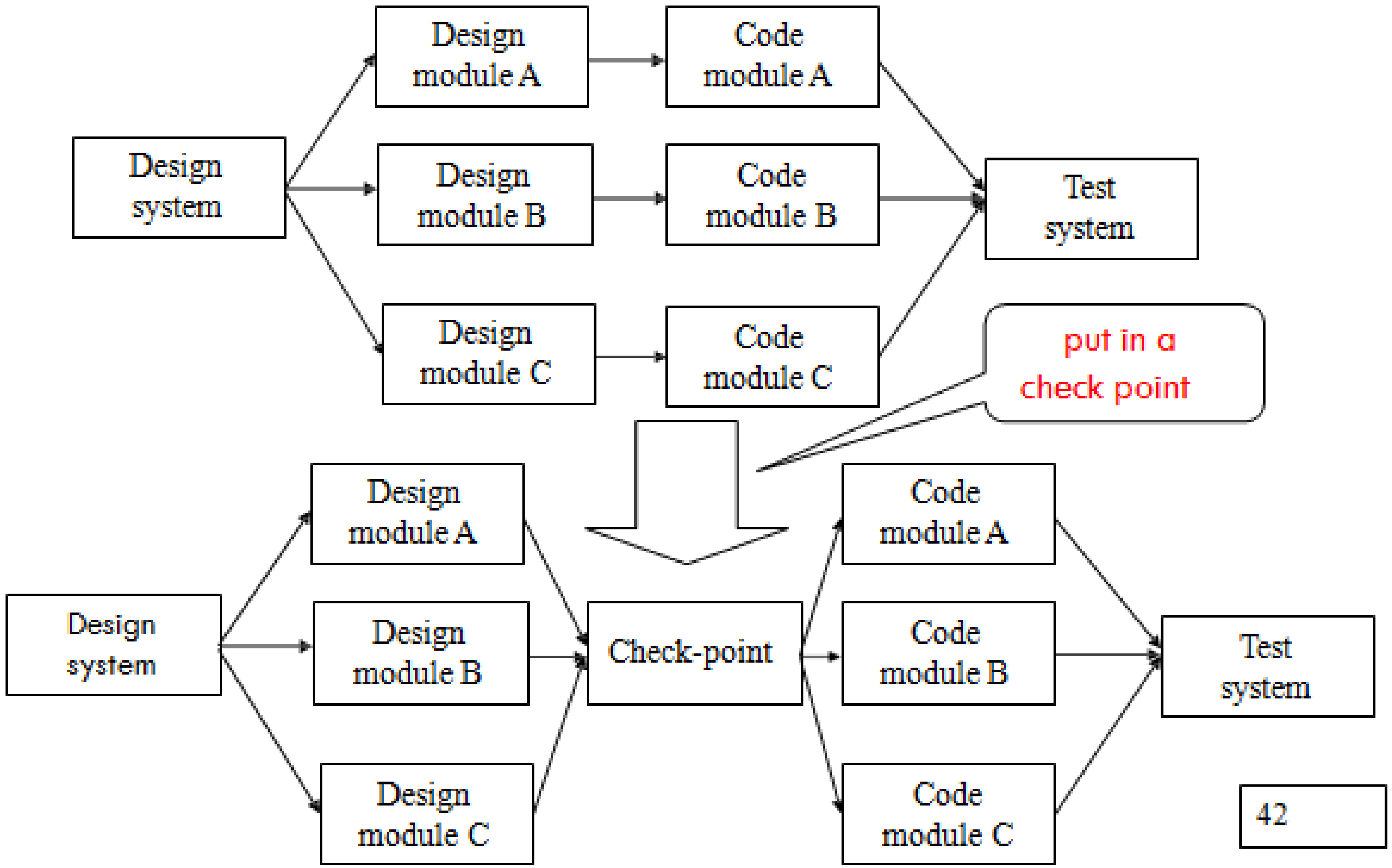
- The PBS and PFD will probably have identified generic products e.g. 'software modules'
- It might be possible to identify specific instances e.g. 'module A', 'module B'…
- But in many cases this will have to be left to later, more detailed, planning

4.4. Produce ideal activity network

- Identify the activities needed to create each product in the PFD
- More than one activity might be needed to create a single product
- Hint: Identify activities by verb + noun but avoid ‘produce…’ (too vague)
- Draw up activity ne



Step 4.5 Add check-points if needed



Step 5:Estimate effort for each activity

- 5.1 Carry out bottom-up estimates
  - distinguish carefully between *effort* and *elapsed* time
- 5.2. Revise plan to create controllable activities
  - break up very long activities into a series of smaller ones
  - bundle up very short activities (create check lists?)

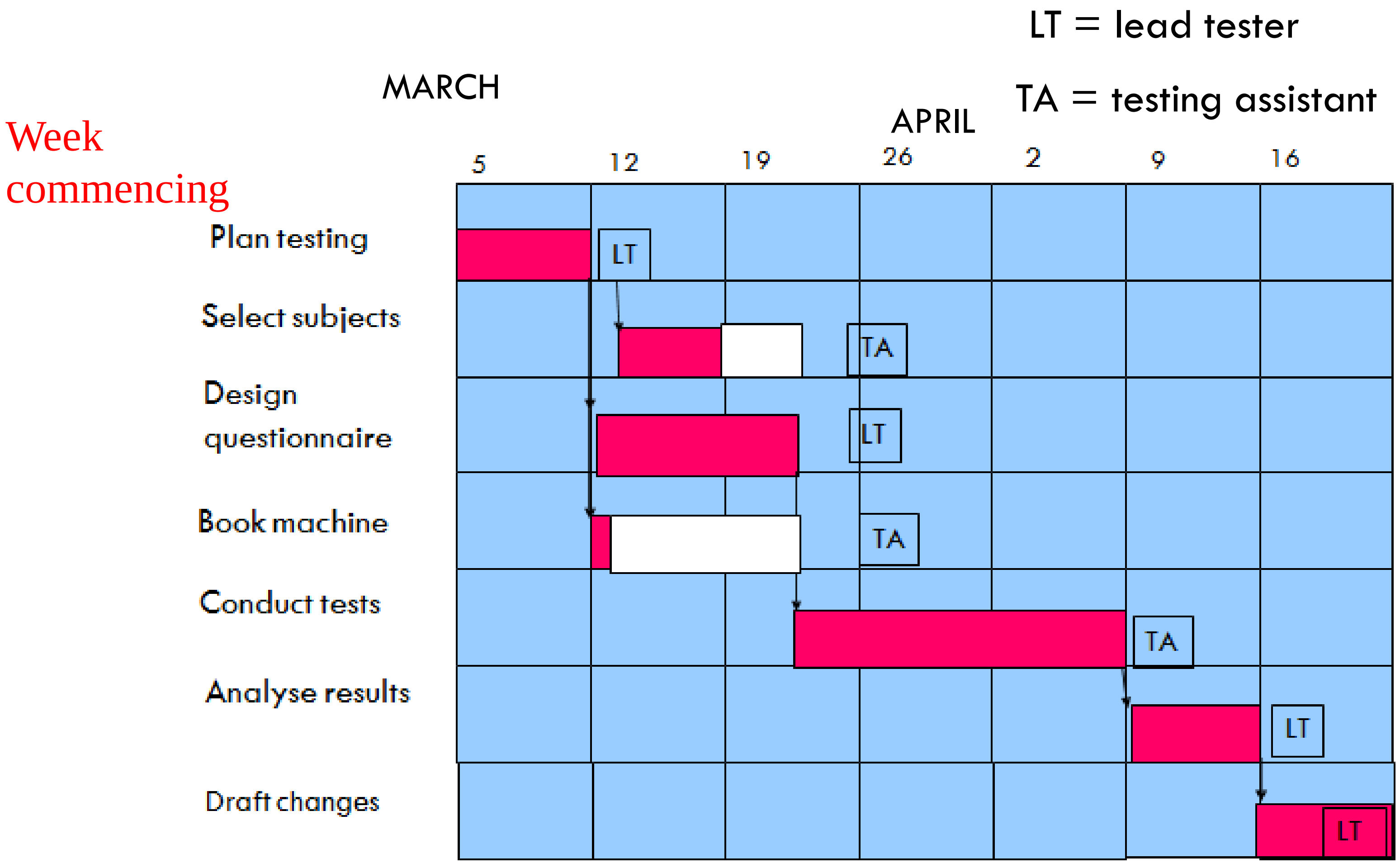
Step 6: Identify activity risks

- 6.1.Identify and quantify risks for activities
  - damage if risk occurs (measure in time lost or money)
  - likelihood if risk occurring
- 6.2. Plan risk reduction and contingency measures
  - risk reduction: activity to stop risk occurring
  - contingency: action if risk does occur
- 6.3 Adjust overall plans and estimates to take account of risks
  - e.g. add new activities which reduce risks associated with other activities e.g. training, pilot trials, information gathering

Step 7: Allocate resources

- Step 7.1 Identify and allocate resources
  - ☐ type of staff needed for each activity
  - ☐ staff availabilities are identified
  - ☐ staff are provisionally allocated to task
- Step 7.2 Revise plans and estimates to take into account resource constraints
  - ☐ staffing constraints
  - ☐ staffing issues

Gantt Chart



LT = lead tester
TA = testing assistant
MARCH
APRIL
Week commencing
5
12
19
26
2
9
16
Plan testing
Select subjects
Design questionnaire
Book machine
Conduct tests
Analyse results
Draft changes
LT
TA
LT
TA
TA
LT
LT

Step 8 : Review/ Publicize plans

- Step 8.1 : Review quality aspects of the project plan
  - To ensure each activity is completed with a quality product
  - Each activity should have "exit requirements".
  - This ensures the quality of the product on each activity.
- Step 8.2 : Document plans and obtain agreement
  - all parties understand and agree to the commitments in the plan

Step 9 and 10 : Execute plan. Lower levels of planning

- Once the project is under way, plans will need to be drawn up in greater detail for each activity as it becomes due.
- Detailed planning of the later stages will have to be delayed because more information will be available nearer the start of the stage.
- Of course, it is necessary to make provisional plans for the more distant tasks because thinking about what has to he done can help unearth potential problems but sight should not be lost of the fact that these plans are provisional.

SUMMARY

- Any planning approach should have the following elements:
  - ✓ the establishment of project objectives;
  - ✓ the analysis of the characteristics of the project;
  - ✓ the establishment of an infrastructure consisting of an appropriate organization and set of standards, methods and tools:
  - ✓ the identification of the products of the project and the activities needed to generate those products;
  - ✓ the allocation of resources to activities:
  - ✓ the establishment of quality controls.
- Project management is an iterative process. As the time approaches for particular activities to be carried out they should be re-planned in more detail.

SOFTWARE PROJECT MANAGEMENT

Module-2: PROJECT LIFE CYCLE AND EFFORT ESTIMATION

CSE4016

PRIYANKA SINGH

Software Process

- A software process is a set of related activities that leads to the production of the software.

- These activities may involve the development of the software from the scratch, or, modifying an existing system.

- Any software process must include the following four activities:
  - ***Software specification (or requirements engineering):*** Define the main functionalities of the software and the constrains around them.
  - ***Software design and implementation:*** The software is to be designed and programmed.
  - ***Software verification and validation:*** The software must conforms to it's specification and meets the customer needs.
  - ***Software evolution (software maintenance):*** The software is being modified to meet customer and market requirements changes.

- In practice, they include sub-activities such as requirements validation, architectural design, unit testing, …etc.
- There are also supporting activities such as configuration and change management, quality assurance, project management, user experience.
- Along with other activities aim to improve the above activities by introducing new techniques, tools, following the best practice, process standardization (so the diversity of software processes is reduced), etc.

- Software process also includes the process description, which includes:
  - Products: The outcomes of the an activity. For example, the outcome of architectural design maybe a model for the software architecture.
  - Roles: The responsibilities of the people involved in the process. For example, the project manager, programmer, etc.
  - Pre and post conditions: The conditions that must be true before and after an activity. For example, the pre condition of the architectural design is the requirements have been approved by the customer, while the post condition is the diagrams describing the architectural have been reviewed.

- Software process is complex, it relies on making decisions. There’s no ideal process and most organizations have developed their own software process.
- For example, an organization works on critical systems has a very structured process, while with business systems, with rapidly changing requirements, a less formal, flexible process is likely to be more effective.

Choice of Process models

- The word “process’ is sometimes used to emphasize the idea of a system in action. In order to achieve an outcome, the system will have to execute one or more activities: this is its process.
- This idea can be applied to the development of computer-based systems where a number of interrelated activities have to be undertaken to create a final product. These activities can be organized in different ways and we can call these process models.
- A major part of the planning method will be the choosing of the development methods to be used and the slotting of these into an overall process model.
- The planner needs not only to select methods but also to specify how the method is to be applied.

Software Process Models

- A software process model is a simplified representation of a software process. Each model represents a process from a specific perspective.
- Definition- A (software/system) process model is a description of the sequence of activities carried out in an SE (software engineering) project, and the relative order of these activities.

- It provides a fixed generic framework that can be tailored to a specific project.
- Project specific parameters will include:
  - Size, (person-years)
  - Budget,
  - Duration.

project plan = process model + project parameters

- There are hundreds of different process models to choose from, e.g:
    - waterfall,
    - code-and-fix
    - spiral
    - rapid prototyping
    - unified process (UP)
    - agile methods, extreme programming (XP)
    - COTS ...
- But most are minor variations on a small number of basic models.

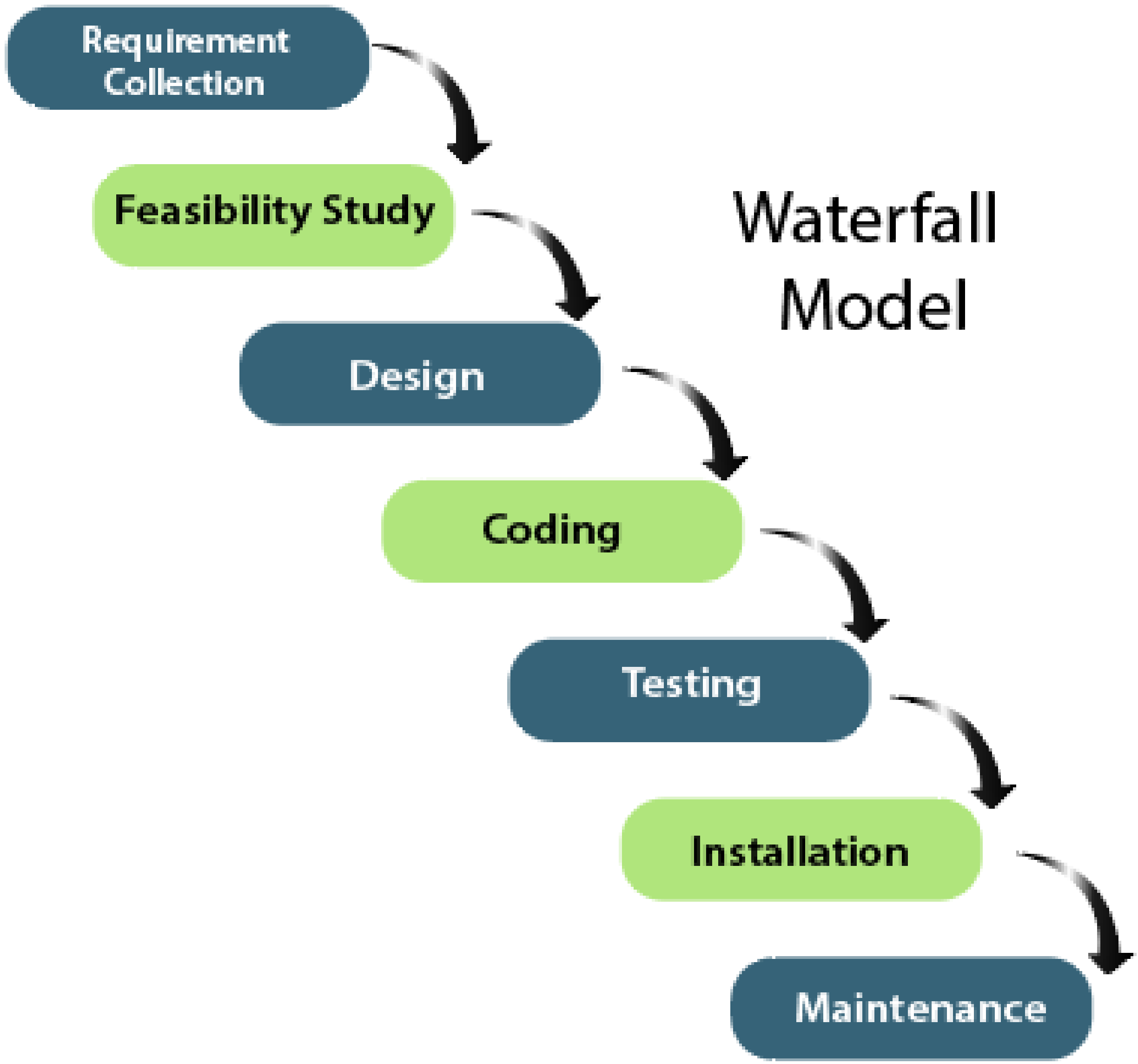
By changing the process model, we can improve and/or tradeoff:

- Development speed (time to market)
- Product quality
- Project visibility
- Administrative overhead
- Risk exposure
- Customer relations, etc, etc.

Normally, a process model covers the entire lifetime of a product. From birth of a commercial idea to final de-installation of last release.

Waterfall Model

- The waterfall model is the classic process model – it is widely known, understood and used.
- In some respect, waterfall is the ”common sense” approach.
- The waterfall model is a sequential approach, where each fundamental activity of a process represented as a separate phase, arranged in linear order.
- In waterfall model, you must plan and schedule all of the activities before starting working on them (plan-driven process).


Requirement Collection
Feasibility Study
Design
Coding
Testing
Installation
Maintenance
Waterfall Model

Advantages

1. Easy to understand and implement.
2. Widely used and known (in theory!)
3. Fits other engineering process models: civil, mech etc.
4. Reinforces good habits: define-before- design, design-before- code
8. Identifies deliverables and milestones
9. Document driven: ***People leave, documents don’t***

   Published documentation standards: URD, SRD, … etc. , e.g. ESA PSS-05.
10. Works well on large/mature products and weak teams.

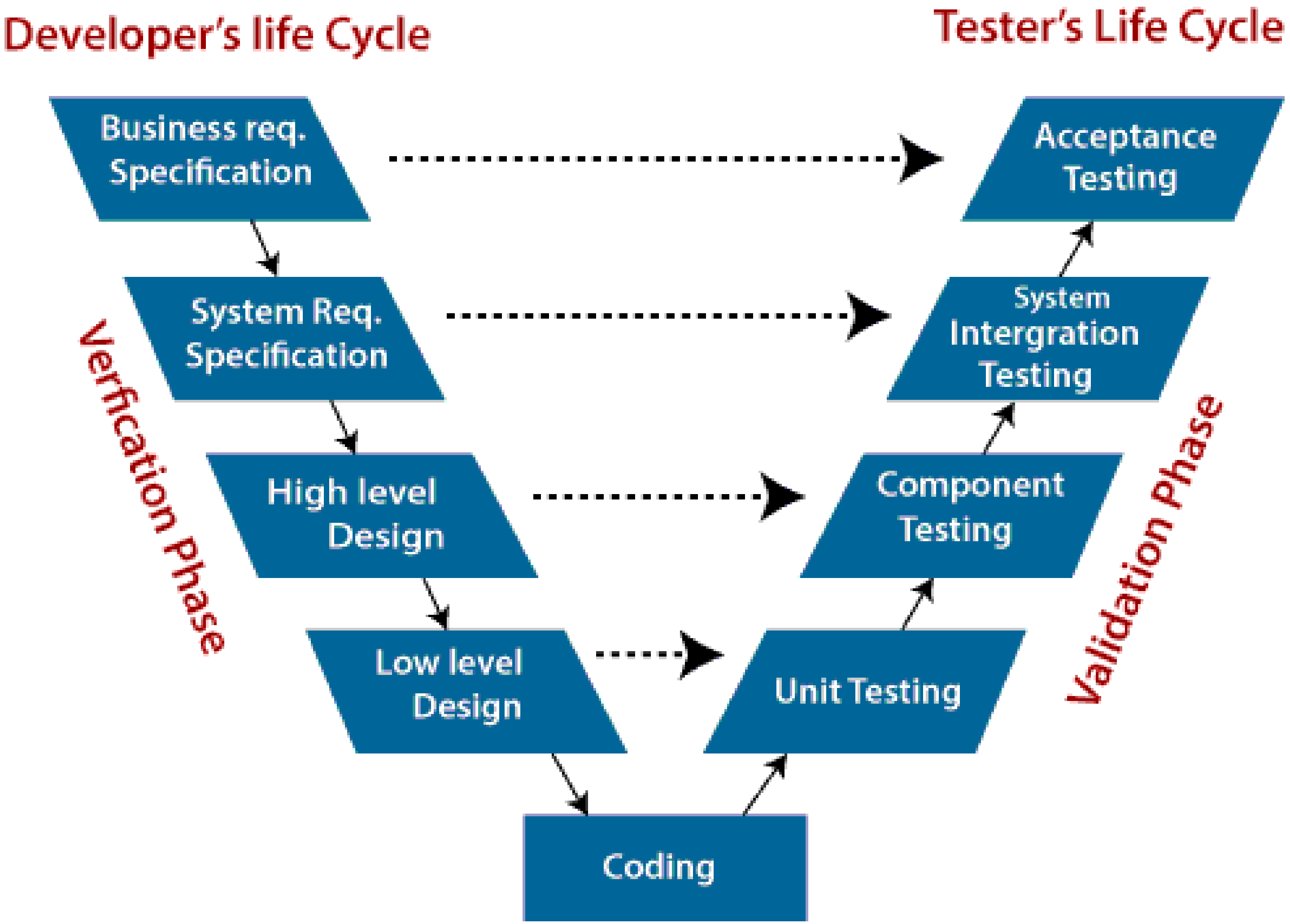
Disadvantages

1. Doesn’t reflect iterative nature of exploratory development.
2. Sometimes unrealistic to expect accurate requirements early in a project
3. Software is delivered late, delays discovery of serious errors.
4. No inherent risk management
5. Difficult and expensive to change decisions, ”swimming upstream”.
6. Significant administrative overhead, costly for small teams and projects.

The V-process model

- This is an elaboration of the waterfall model and stresses the necessity for validation activities that match the activities that create the products of the project.
- 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.

V- Model


Developer's life Cycle
Tester's Life Cycle
Business req. Specification
System Req. Specification
High level Design
Low level Design
Coding
Unit Testing
Component Testing
System Intergration Testing
Acceptance Testing
Verfcation Phase
Validation Phase

- ***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.

Spiral Model

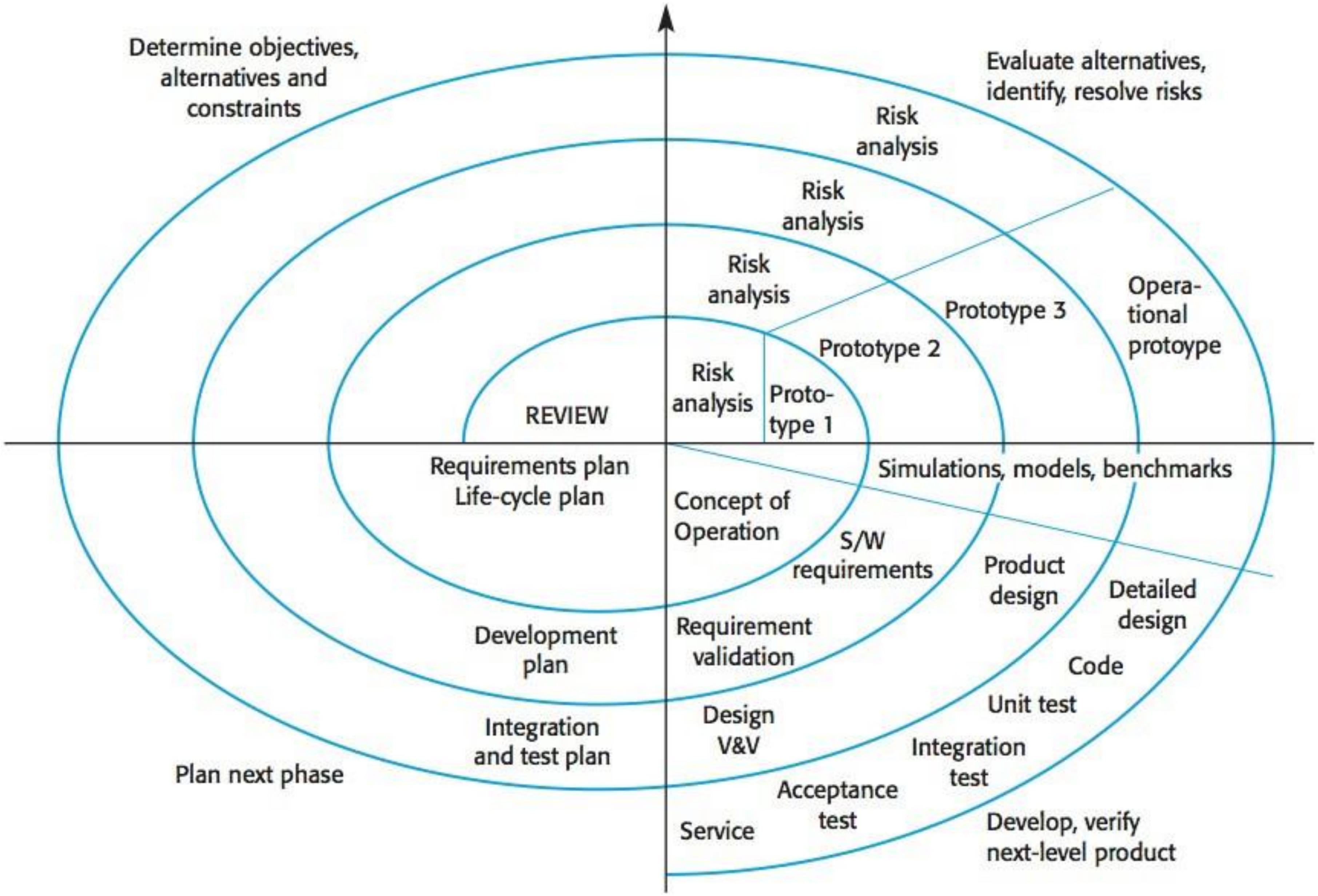
- Extends waterfall model by adding iteration to explore /manage risk
- Project risk is a moving target. Natural to progress a project cyclically in four step phases

1. Consider alternative scenarios, constraints
2. Identify and resolve risks
3. Execute the phase
4. Plan next phase: e.g. user req, software req, architecture

… then goto 1

Key idea: on each iteration identify and solve

the sub-problems with the highest risk


Determine objectives, alternatives and constraints
Evaluate alternatives, identify, resolve risks
Risk analysis
Risk analysis
Risk analysis
Risk analysis
Proto-type 1
Prototype 2
Prototype 3
Opera-tional protoype
REVIEW
Requirements plan Life-cycle plan
Concept of Operation
Simulations, models, benchmarks
S/W requirements
Product design
Detailed design
Code
Unit test
Integration test
Acceptance test
Service
Requirement validation
Design V&V
Development plan
Integration and test plan
Plan next phase
Develop, verify next-level product

Advantages

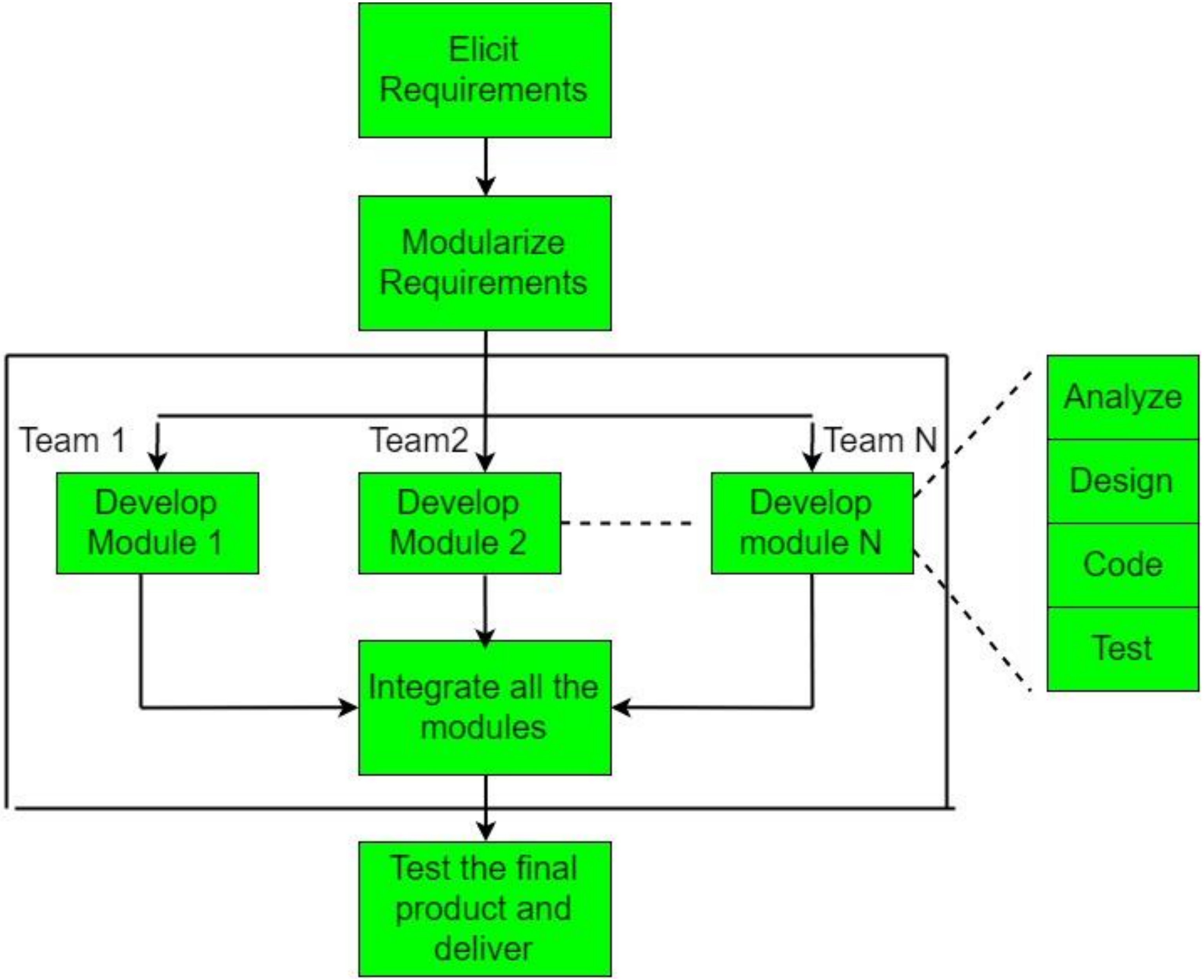
- ***Realism***: the model accurately reflects the iterative nature of software development on projects with unclear requirements
- ***Flexible***: incoporates the advantages of the waterfall and evolutionary methods
- ***Comprehensive model decreases risk***
- ***Good project visibility***.

Disadvantages

1. Needs technical expertise in risk analysis and risk management to work well.
2. Model is poorly understood by nontechnical management, hence not so widely used
3. Complicated model, needs competent professional management. High administrative overhead.

Rapid Application Development (RAD)

- The Rapid Application Development Model was first proposed by IBM in 1980's. The critical feature of this model is the use of powerful development tools and techniques.
- A software project can be implemented using this model if the project can be broken down into small modules wherein each module can be assigned independently to separate teams. These modules can finally be combined to form the final product.
- Development of each module involves the various basic steps as in waterfall model i.e analyzing, designing, coding and then testing, etc. as shown in the figure.
- Another striking feature of this model is a short time span i.e the time frame for delivery(time-box) is generally 60-90 days.



Elicit Requirements
Modularize Requirements
Team 1
Team2
Team N
Develop Module 1
Develop Module 2
Develop module N
Analyze
Design
Code
Test
Integrate all the modules
Test the final product and deliver

Advantages –

- Use of reusable components helps to reduce the cycle time of the project.
- Feedback from the customer is available at initial stages.
- Reduced costs as fewer developers are required.
- Use of powerful development tools results in better quality products in comparatively shorter time spans.
- The progress and development of the project can be measured through the various stages.
- It is easier to accommodate changing requirements due to the short iteration time spans.

Disadvantages –

- The use of powerful and efficient tools requires highly skilled professionals.
- The absence of reusable components can lead to failure of the project.
- The team leader must work closely with the developers and customers to close the project in time.
- The systems which cannot be modularized suitably cannot use this model.
- Customer involvement is required throughout the life cycle.
- It is not meant for small scale projects as for such cases, the cost of using automated tools and techniques may exceed the entire budget of the project.

Agile methods

- Agile software development refers to software development methodologies centered round the idea of iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams.
- The ultimate value in Agile development is that it enables teams to deliver value faster, with greater quality and predictablity, and greater aptitude to respond to change. Scrum and Kanban are two of the most widely used Agile methodologies.
- Originally created for software development, it was established as a response to the inadequacies of the Waterfall method, the processes of which did not meet the demands of the highly competitive and constant movement of the software industry.

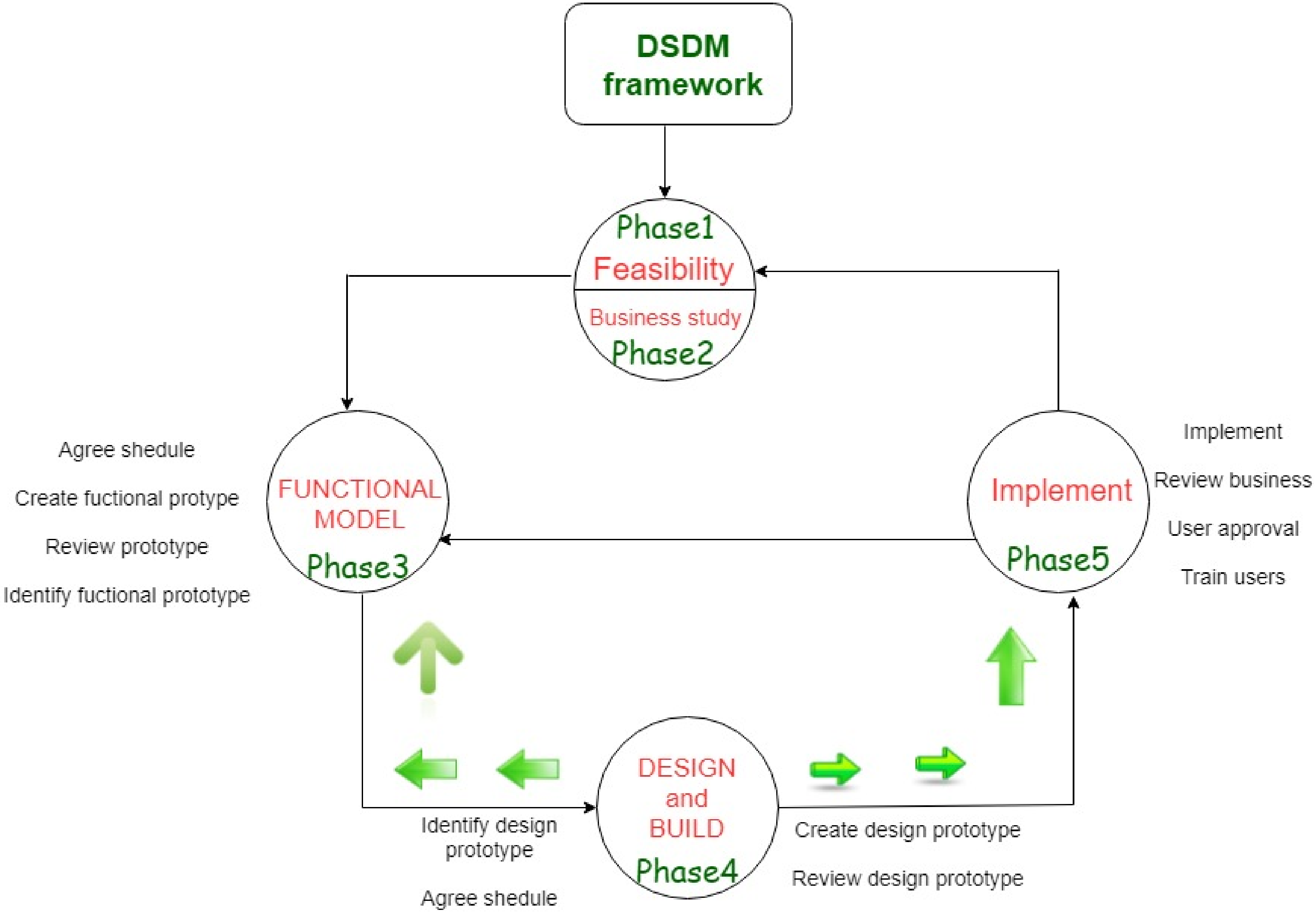
- In Agile development, Design and Implementation are considered to be the central activities in the software process.
- Design and Implementation phase also incorporate other activities such as requirements elicitation and testing into it.
- In an agile approach, iteration occurs across activities. Therefore, the requirements and the design are developed together, rather than separately.
- The allocation of requirements and the design planning and development as executed in a series of increments.
- An agile process focuses more on code development rather than documentation.

Principles:

1. Customer satisfaction through early and continuous software delivery
2. Accommodate changing requirements throughout the development process
3. Frequent delivery of working software
4. Collaboration between the business stakeholders and developers throughout the project
5. Support, trust, and motivate the people involved
6. Enable face-to-face interactions
7. Working software is the primary measure of progress
8. Agile processes to support a consistent development pace
9. Attention to technical detail and design enhances agility
10. Simplicity
11. Self-organizing teams encourage great architectures, requirements, and designs
12. Regular reflections on how to become more effective`

Dynamic System Development Method

- The Dynamic Systems Development technique (DSDM) is an associate degree agile code development approach that provides a framework for building and maintaining systems.
- DSDM is an iterative code method within which every iteration follows the 80% rule that simply enough work is needed for every increment to facilitate movement to the following increment. The remaining detail is often completed later once a lot of business necessities are noted or changes are requested and accommodated.



Dynamic Systems Development Method life cycle

Principles:

1. Focus on the business need
2. Deliver on time
3. Collaborate
4. Never compromise quality
5. Build incrementally from firm foundations
6. Develop iteratively
7. Communicate continuously and clearly
8. Demonstrate control

Extreme Programming

- Extreme programming (XP) is one of the most important software development framework of Agile models.
- It is used to improve software quality and responsive to customer requirements.
- The extreme programming model recommends taking the best practices that have worked well in the past in program development projects to extreme levels.

Good practices needs to practiced extreme programming:

- ***Code Review:*** Code review detects and corrects errors efficiently. It suggests pair programming as coding and reviewing of written code carried out by a pair of programmers who switch their works between them every hour.
- ***Testing***: Testing code helps to remove errors and improves its reliability. XP suggests test-driven development (TDD) to continually write and execute test cases.
- ***Incremental development***: Incremental development is very good because customer feedback is gained and based on this development team come up with new increments every few days after each iteration.
- ***Simplicity***: Simplicity makes it easier to develop good quality code as well as to test and debug it.
- ***Design***: Good quality design is important to develop a good quality software. So, everybody should design daily.
- ***Integration testing***: It helps to identify bugs at the interfaces of different functionalities.

Basic principles of Extreme programming:

- XP is based on the frequent iteration through which the developers implement User Stories. User stories are simple and informal statements of the customer about the functionalities needed. A User story is a conventional description by the user about a feature of the required system. It does not mention finer details such as the different scenarios that can occur.
- On the basis of User stories, the project team proposes Metaphors. Metaphors are a common vision of how the system would work.
- The development team may decide to build a Spike for some feature. A Spike is a very simple program that is constructed to explore the suitability of a solution being proposed. It can be considered similar to a prototype.

Some of the basic activities that are followed during software development by using XP model are given below:

- **Coding**: Here, coding activity includes drawing diagrams (modeling) that will be transformed into code, scripting a web-based system and choosing among several alternative solutions.
- **Testing**: XP model gives high importance on testing and considers it be the primary factor to develop a fault-free software.
- **Listening**: The developers needs to carefully listen to the customers if they have to develop a good quality software.
- **Designing**: Without a proper design, a system implementation becomes too complex and very difficult to understand the solution, thus it makes maintenance expensive. A good design results elimination of complex dependencies within a system.
- **Feedback**: One of the most important aspects of the XP model is to gain feedback to understand the exact customer needs. Frequent contact with the customer makes the development effective.
- **Simplicity**: The main principle of the XP model is to develop a simple system that will work efficiently in present time, rather than trying to build something that would take time and it may never be used.

SOFTWARE PROJECT MANAGEMENT

Module-2: PROJECT LIFE CYCLE AND EFFORT ESTIMATION

CSE4016

PRIYANKA SINGH

Basics of Software estimation

- One definition of a successful project is that the system is delivered 'on time and within budget and with the required quality, which implies that target are set and the project leader then tries to meet those targets.
- Estimating the effort required to implement software is notoriously difficult. Some of the difficulties of estimating are inherent in the very nature of software, especially its complexity and invisibility.
- In addition the intensely human activities that make up system development cannot be treated in a purely mechanistic way.

- Other difficulties include:
    - Novel applications of software
    - Changing technology
    - Lack of homogeneity of project experience

Where are estimates done?

Estimates are carried out at various stages of a software project.

- ***Strategic planning-*** The costs of computerizing potential applications as well as the benefits of doing so might need to be estimated to help divide what priority to give to each project.
- ***Feasibility study*** -This ascertains that the benefits of the potential system will justify the costs.
- ***System specification-*** The *effort needed to implement different design proposals will need to be estimated,* Estimates at the design stage will also confirm that the feasibility study is still valid, taking into account all that has been learnt during detailed requirements analysis.
- **Evaluation of suppliers' proposals**
- **Project planning**- As the planning and implementation of the project progresses to greater levels of detail, more *detailed estimates of smaller work components* will be made.

The basis for software estimating

- ***The need for historical data-*** Nearly all estimating methods need information about how projects have been implemented in the past. However, care needs to be taken in judging the applicability of data to the estimator's own circumstances because of possible differences in environmental factors such as the programming languages used, the software tools available, the standards enforced and the experience of the staff.
- ***Measure of work-*** It is normally not possible to calculate directly the actual cost or time required to implement a project. The time taken to write a program might vary according to the competence or experience of the programmer. Implementation times might also vary because of environmental factors such as the software tools available.

Complexity-

- Two programs with the same KLOC will not necessarily take the same time to write, even if done by the same developer in the same environment.
- One program might be more complex. Because of this, SLOC (Source lines of code) estimates have to be modified to take complexity into account.
- Attempts have been made to find objective measures of complexity, but often it will depend on the subjective judgement of the estimator.

Problems with over- and under-estimates

An over-estimate might cause the project to take longer than it would otherwise. This can be explained by the application of two laws-

- **Parkinson's law-** "Work expands to fill the time available", which implies that given an easy target staff will work less hard.
- **Brook's 1aw-** The effort required to implement a project will go up disproportionately with the number of staff assigned to the project. As the project team grows in size so will the effort that has to go into management, co-ordination and communication.

The danger with the under-estimate is the effect on quality. Staff, particularly those with less experience, might respond to pressing deadlines by producing work which is sub-standard.

Software effort estimation techniques

Barry Boehm, in his classic work on software effort models, identified the main ways of deriving estimates of software development effort as:

- **Algorithmic models** - which use 'effort drivers’ representing characteristics of the target system and the implementation environment to predict effort;
- **Expert judgement** - where the advice of knowledgeable staff is solicited;
- **Analogy** - where a similar, completed, project is identified and its actual effort is used as a basis for the new project;
- **Parkinson** - which identifies the staff effort available to do a project and uses that as the 'estimate’;
- **Price to win** - where the ‘estimate" is a figure that appears to be sufficiently low to win a contract;
- **Top-down** - where an overall estimate is formulated for the whole project and is then broken down into the effort required for component tasks;
- **Bottom-up** - where component tasks are identified and sized and these individual estimates are aggregated.

The top-down approach and parametric models

- The top-down approach is normally associated with parametric (or algorithmic) models. These may be explained using the analogy of estimating the cost of rebuilding a house.
- The effort needed to implement a project will be related mainly to variables associated with characteristics of the final system. The form of the parametric model will normally be one or more formulae in the form:

Effort = (system size) x (productivity rate)

Some parametric models, such as that implied by function points, are focused on system or task size, while others, such are COCOMO, are more concerned with productivity factors.

Estimating by analogy

- The use of analogy is also called case-based reasoning. The estimator seeks out projects that have been completed and that have similar characteristic, to the new project.
- The effort that has been recorded for the matching source case can then be used as a base estimate for the target.
- The estimator should then try to identify any differences between the target and the source and make adjustments to the base estimate for the new project.
- Identify the similarities and differences between the different systems:

distance = square-root of ((target_parameter$_1$ - source_parameter$_1$)^2+ …. + (target_parameter$_n$ - source_parameter$_n$)^2)

Example

- Say that the cases are being matched on the basis of two parameters, the number of inputs to and the number of outputs from the system to be built. The new project B is known to require 7 inputs and 15 outputs. One of the past cases, Project A has 8 inputs and 17 outputs. Is Project B a better analogy with the target than Project A?

Exercise

- Project B has 5 inputs and 10 outputs. What would be the Euclidean distance between this project and the target new project being considered above? Is Project B a better analogy with the target than Project A?

COSMIC Full function points

- The COSMIC method defines the principles, rules and a process for measuring a standard functional size of a piece of software.
- ‘Functional size’ is a measure of the amount of functionality provided by the software, completely independent of any technical or quality considerations.

- The COSMIC method may be used to size software such as business applications; real-time software; infrastructure software such as in operating systems; and hybrids of these.
- The common characteristic of all these types of software is that they are dominated by functions that input data, store and retrieve data, and output data.
- Subject to the above, the method may be applied to measure the Functional User Requirements ('FUR') of software:
  - At any level of decomposition, e.g. a 'whole' piece of software or any of its components, sub-components, etc;
  - In any layer of a multi-layer architecture;
  - At any point in the life-cycle of the piece of software;

COCOMO

- Boehm's COCOMO (Constructive COst MOdel) is often referred to in the literature on software project management, particularly in connection with software estimating. The term COCOMO really refers to a group of models.
- COCOMO (Constructive Cost Model) is a combination of parametric estimation equation and weighting method.
- Based on the estimated instructions DSI (Delivered Source Instructions), the effort is calculated by taking into consideration both the attempted quality and the productivity factors.
- COCOMO is based on the conventional top-down programming and concentrates on the number of instructions.

1. Basic COCOMO:

- Basic COCOMO model is static single-valued model that computes software development effort (and cost) as a function of program size expressed in estimated lines of code.
- By means of parametric estimation equations, the development effort and the development duration are calculated.
- In this connection it is differentiated according to system types (organicbatch, semidetached-on-line, embedded-real-time) and project sizes (small, intermediate, medium, large, very large).

2. Intermediate COCOMO:

- Intermediate COCOMO model computes software development effort as a *function of program size and a set of "cost drivers"* that include subjective assessments of product, hardware, personnel, and project attributes.
- The estimation equations are now taking into consideration 15 influence factors; these are *product attributes* (like software reliability, size of the database, complexity), *computer attributes* (like computing time restriction, main memory restriction), *personnel attributes* (like programming and application experience, knowledge of the programming language), and *project attributes* (like software development environment, pressure of development time).

3. Detailed COCOMO:

- Advanced COCOMO model incorporates all characteristics of the intermediate version with an assessment of the cost driver's impact on each step, like analysis, design, etc.

- In this case the breakdown to phases is not realized in percentages but by means of influence factors allocated to the phases.

- At the same time, it is differentiated according to the three levels of the product hierarchy (module, subsystem, system); product-related influence factors are now taken into consideration in the corresponding estimation equations.

COCOMO II-a Parametric Productivity Model:

- ***Basic COCOMO***:

Boehm originally based his models in the late 1970s on a study of 63 projects.

$$Effort = a(size)^b$$

Generally, information systems were regarded as organic while real-time systems were embedded.

where effort is measured in ***pm***, or the number of ***'person-months'*** consisting of units of 152 working hours, size is measured in ***kdsi*** *(kilo Delivered Source Instructions*) , thousands of delivered source code instructions, and ***a*** and ***b*** are constants.

The first step was to derive an estimate of the system size in terms of ***kdsi***. The constants, ***a*** and ***b*** (see Table), depended on whether the system could be classified, in Boehm's terms, as 'organic', 'semi-detached' or 'embedded'.

These related to the technical nature of the system and the development environment.

- ***Organic mode*** - this would typically be the case when relatively small teams developed software in a highly familiar in-house environment and when the system being developed was small and the interface requirements were flexible.

- ***Embedded mode*** - this meant the product being developed had to operate within very light constraints and changes to the system were very costly.

- ***Semi-detached mode*** - this combined elements of the organic and the embedded modes or had characteristics that came between the two.

System type	a	b
Organic	2.4	1.05
Semi-detached	3.0	1.12
Embedded	3.6	1.20

Examples

Suppose size is 200 KLOC, Calculate effort

- Organic
  - $2.4(200)^{1.05}$ = 626 staff-months
- Semi-Detached
  - $3.0(200)^{1.12}$ = 1,133 staff-months
- Embedded
  - $3.6(200)^{1.20}$ = 2,077 staff-months

Project Duration

Development time= c(Effort)^d

Average Staff Size= Effort/development time

Productivity= Size/Effort

Mode	c	d
Organic	2.5	0.38
Semi-detached	2.5	0.35
Embedded	2.5	0.32

Example

- Suppose size is 200 KLOC, Calculate development time?
- Organic
  - TDEV = $2.5(626)^{0.38}$ = 29 months
- Semi-detached
  - TDEV = $2.5(1133)^{0.35}$ = 29 months
- Embedded
  - TDEV = $2.5(2077)^{0.32}$ = 29 months

Complete Example, Organic

Suppose an organic project has 7.5 KLOC,

- Effort $2.4(7.5)^{1.05}$ = 20 staff–months
- Development time $2.5(20)^{0.38}$ = 8 months
- Average staff 20 / 8 = 2.5 staff
- Productivity 7,500 LOC / 20 staff-months
  = 375 LOC / staff-month

Complete Example, Embedded

Suppose an embedded project has 50 KLOC,

- Effort $3.6(50)^{1.20}$ = 394 staff–months
- Development time $2.5(394)^{0.32}$ = 17 months
- Average staff 394 / 17 = 23 staff
- Productivity 50,000 LOC / 394 staff-months
  = 127 LOC / staff-month

Intermediate COCOMO

$$Effort = a(size)^{b} * C$$

Where

- *E is the effort*
- *a and b are constants (as before)*
- *size is thousands of lines of code in KLOC*
- *C is the effort adjustment factor*

- Intermediate COCOMO introduces Cost Drivers.
    - Product Attributes
    - Computer Attributes
    - Personnel Attributes
    - Project Attributes

- They are used because
    - they are statistically significant to the cost of the project; and
    - they are not correlated to the project size (KLOC).

“The models are just there to help, not to make the management decisions for you.”

-- Barry Boehm

SOFTWARE PROJECT MANAGEMENT

Module-3: ACTIVITY PLANNING AND RISK MANAGEMENT

CSE4016

PRIYANKA SINGH

ACTIVITY PLANNING

- As a project progresses it is unlikely that everything will go according to plan.
- Much of the job of project management concerns recognizing when something has gone wrong, identifying its causes and revising the plan to mitigate its effects.
- The activity plan should provide a means of evaluating the consequences of not meeting any of the activity target dates and guidance as to how the plan might most effectively be modified to bring the project back to target.
- The activity plan may well also offer guidance as to which components of a project should be most closely monitored.

Objectives of activity planning

In addition to providing project and resource schedules, activity planning aims to achieve a number of other objectives which may be summarized as follows.

1-Feasibility assessment- Is the project possible within required timescales and resource constraints? The fact that a project may have been estimated as requiring two work-years effort might not mean that it would be feasible to complete it within, say. three months were eight people to work on it - that will depend upon the availability of staff and the degree to which activities may be undertaken in parallel.

2-Resource allocation -

- What are the most effective ways of allocating resources to the project and when should they be available?
- The project plan allows us to investigate the relationship between timescales and resource availability .

3-Detailed costing- How much will the project cost and when is that expenditure likely to take place? After producing an activity plan and allocating specific resources, we can obtain more detailed estimates of costs and their timing.

4-Motivation- Providing targets and being seen to monitor achievement against targets is an effective way of motivating staff, particularly where they have been involved in setting those targets in the first place.

5-Co-ordination- When do the staff in different departments need to be available to work on a particular project and when do staff need to be transferred between projects?

The project plan, particularly with large projects involving more than a single project team, provides an effective vehicle for communication and coordination among teams. In situations where stuff may need to be transferred between project teams a set of integrated project schedules should ensure that such staff are available when required and do not suffer periods of enforced idleness

- Activity planning and scheduling techniques place an emphasis on completing the project in a minimum time at an acceptable cost or, alternatively, meeting an arbitrarily set target date at minimum cost.
- One effective way of shortening project durations is to carry out activities in parallel. Clearly we cannot undertake all the activities at the same time - some require the completion of others before they can start.

When to plan ?

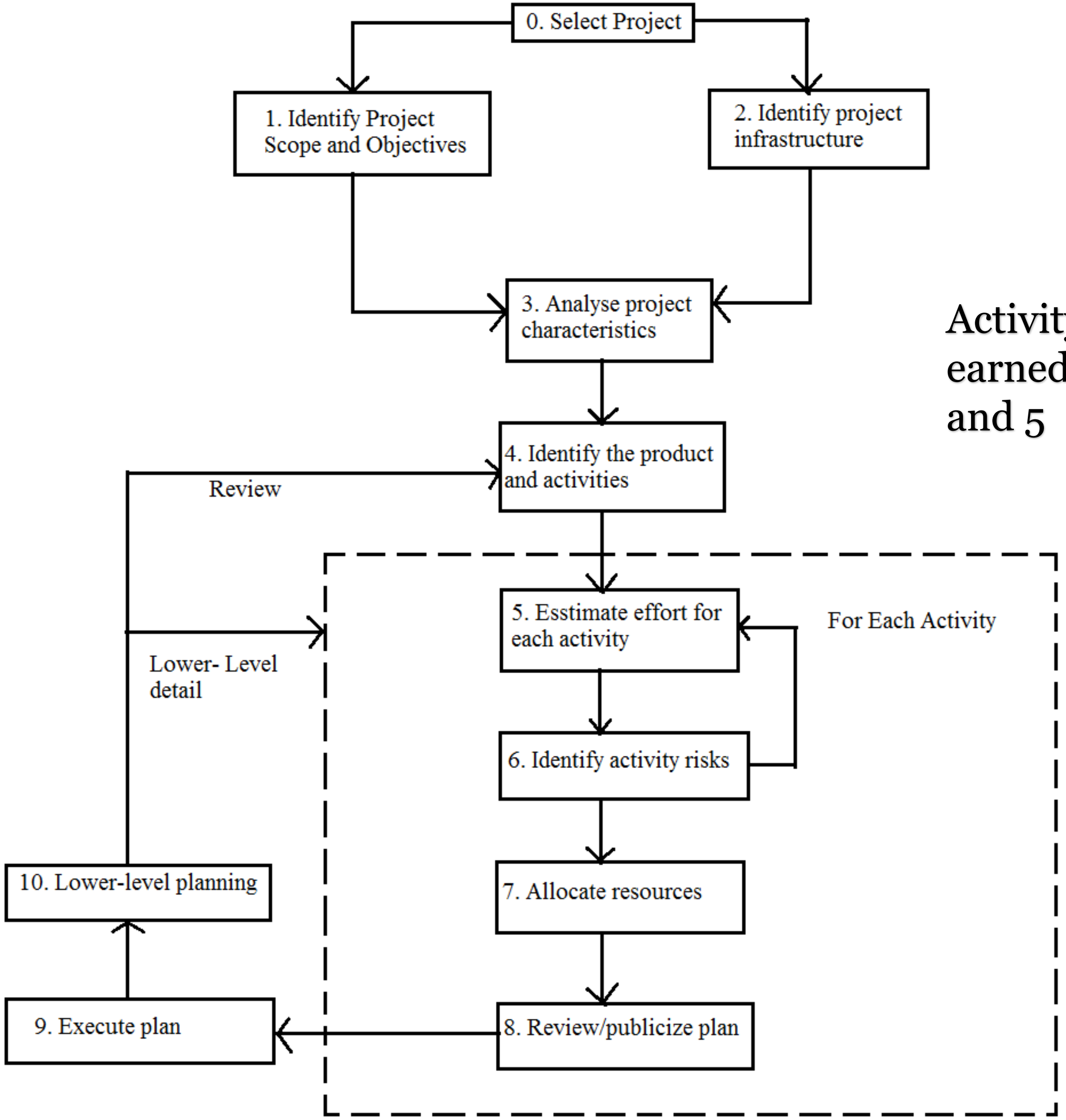
- Planning is an ongoing process of refinement, each iteration becoming more detailed and more accurate than the last.
- Over successive iterations, the emphasis and purpose of planning will shift.
- Throughout the project, until the final deliverable has reached the customer, monitoring and replanning must continue to correct any drift that might prevent meeting the targets.

Project schedules

- Before work commences on a project, the project plan must be developed to the level of showing dates when each activity should start and finish and when and how much of each resource will be required. Once the plan has been refined to this level of detail we call it a *project schedule*.

Creating a project schedule comprises four main stages.

1-The first step in producing the plan is *to decide what activities need to be carried out and in what order they are to be done*. From this we can construct *an ideal activity plan* - that is, a plan of when each activity would ideally be undertaken were resource is not a constraint.



Activity planning is earned out in Steps 4 and 5

2- The ideal activity plan will *then be the subject of an activity risk analysis, aimed at identifying potential problem*. This, might suggest alterations to the ideal activity plan and will almost certainly have implication for resource allocation.

3- The third step is *resource allocation*. The expected availability of resources might place constraints on when certain activities can be carried out, and our ideal plan might need to be adapted to take account of this.

4- The final step is *schedule production*. Once resources have been allocated to each activity, we will be in a position to draw up and publish a project schedule; which indicates planned start and completion dates and a resource requirements statement for each activity.

Projects and activities

- A project is composed of a number of inter-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 the activities it encompasses have been completed
- An activity must have a clearly defined start and a clearly defined end-point, normally marked by the production of a tangible deliverable;
- If an activity requires a resource then that resource requirement must be forecastable and is assumed to be required at a constant level throughout the duration of the activity;
- The duration of an activity must be forecastable - assuming normal circumstances, and the reasonable availability of resources:
- Some activities might require that others are completed before they can begin (these are known as precedence requirement.).

Identifying activities

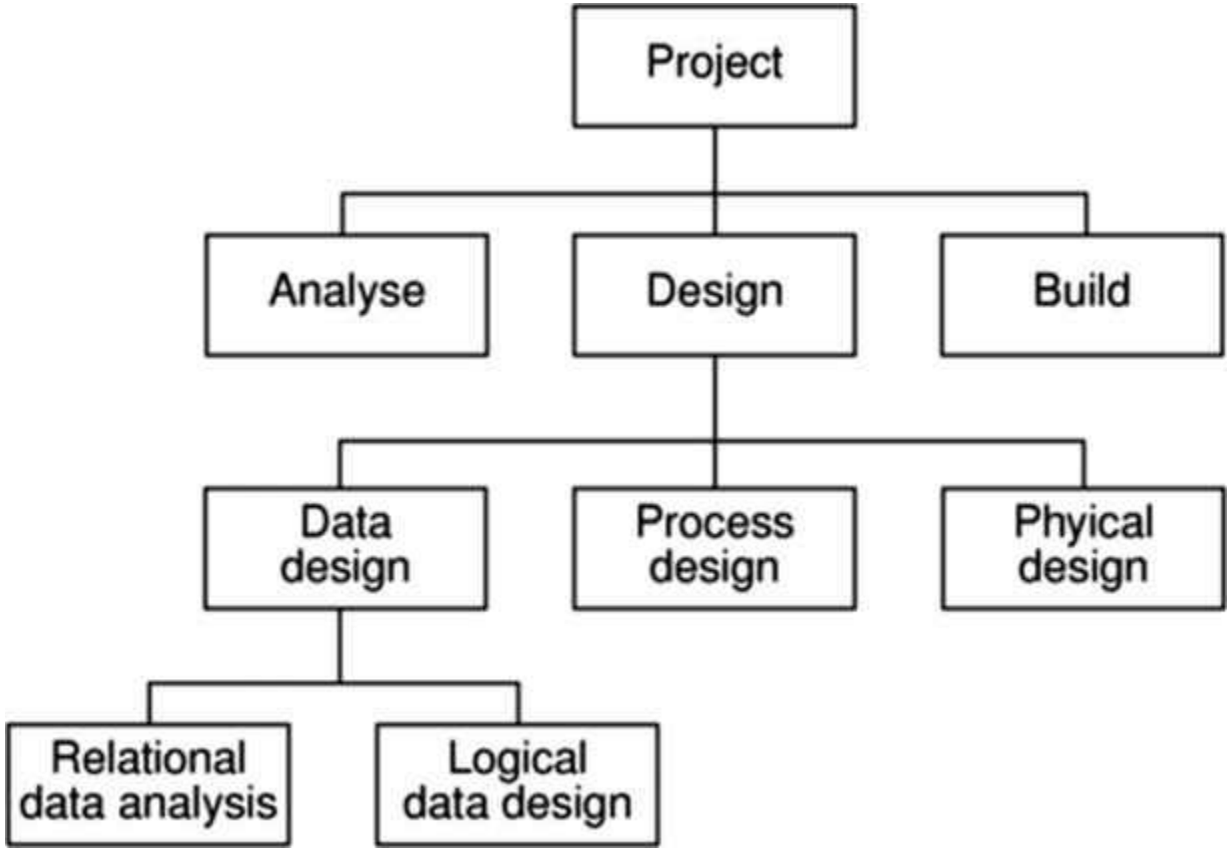
Essentially there are three approaches to identifying the activities or tasks that make up a project -

1. Activity-based approach
2. Product-based approach and
3. Hybrid approach

Activity-based approach

- The activity-bascd approach consists of creating a list of all the activities that the project is thought to involve.
- This might involve a brainstorming session involving the whole project team or it might stem from an analysis of similar past projects.
- When listing activities, particularly for a large project, it might be helpful to subdivide the project into the main life style stages and consider each of thcsc separately.
- Rather than doing this in an ad hoc manner, with the Obvious risks of omitting or double-counting tasks, a much favored way of generating a task list is to create a "Work Breakdown Structure (WBS).
- This involves identifying the main (or high- level) tasks required to complete a project and then breaking each of these down into a set of lower-level tasks.

A fragment of an activity-based Work Breakdown Structure.



- Activities are added to a branch in the structure if they directly contribute to the task immediately above - if they do not contribute to the parent task, then they should not be added to that branch.
- The tasks at each level in any branch should include everything that is required to complete the task at the higher level - if they are not a comprehensive definition of the parent task, then something is missing.
- When preparing a WBS, consideration must he given to the final level of detail or depth of the structure. Too great a depth will result in a large number of small tasks that will be difficult to manage, whereas too shallow structure will provide insufficient detail for project control. Each branch should, however, be broken down at least to a level where each leaf may be assigned to an individual or responsible section within the organization.

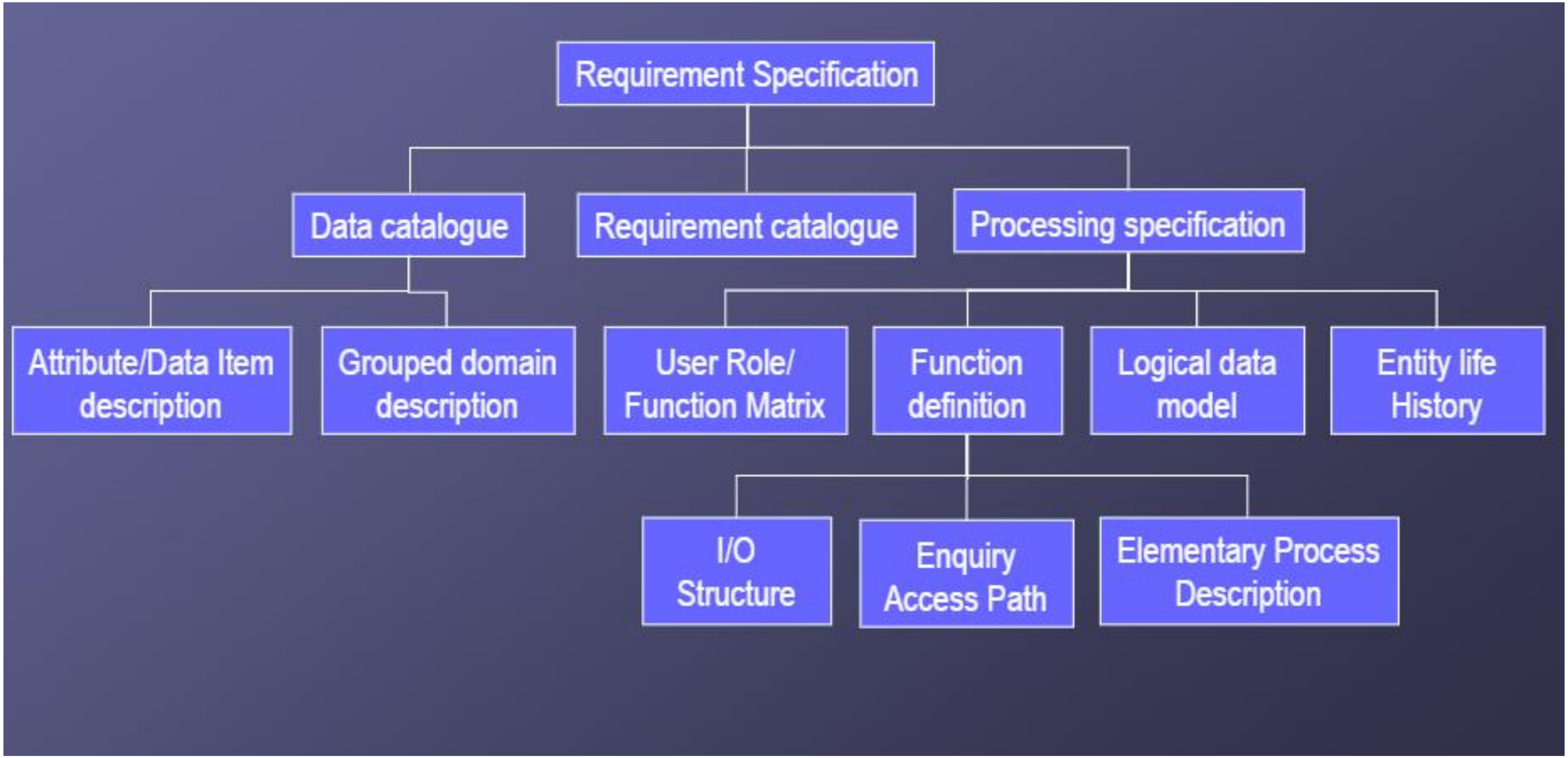
- Advantages claimed for the WBS approach include the belief that it is much more likely to result in a task catalogue that is complete and is composed of non-overlapping activities.
- The WBS also represents a structure that may be refined as the project proceeds.
- Once the project's activities have been identified (whether or not by using a WBS) they need to be sequenced in the sense of deciding which activities need to be completed before others can start.

The product-based approach

- It consists of producing a Product Breakdown Structure (PBS) and a Product Flow Diagram (PFD).
- The PFD indicates, for each product, which other products are required as inputs.
- The PFD can therefore be easily transformed into an ordered list of activities by identifying the transformations that turn some products into others.
- Proponents of this approach claim that it is less likely that a product will be left out of a PBS than that an activity might be omitted from an unstructured activity list.

- The Product Flow Diagram shows which products must be complete before the next can be produced
  - Most items in the diagram are things the customer needs,
    - eg. design documentation, software, manuals
    - indicated by boxes
  - Some items are intermediate products, needed only to help produce other products,
    - eg. first cut database design
    - indicated by boxes
  - Some items will exist already,
    - eg. feasibility study report, terms of reference
    - indicated by ellipses (ovals)
  - Arrows indicate that one product is required in order to produce the next

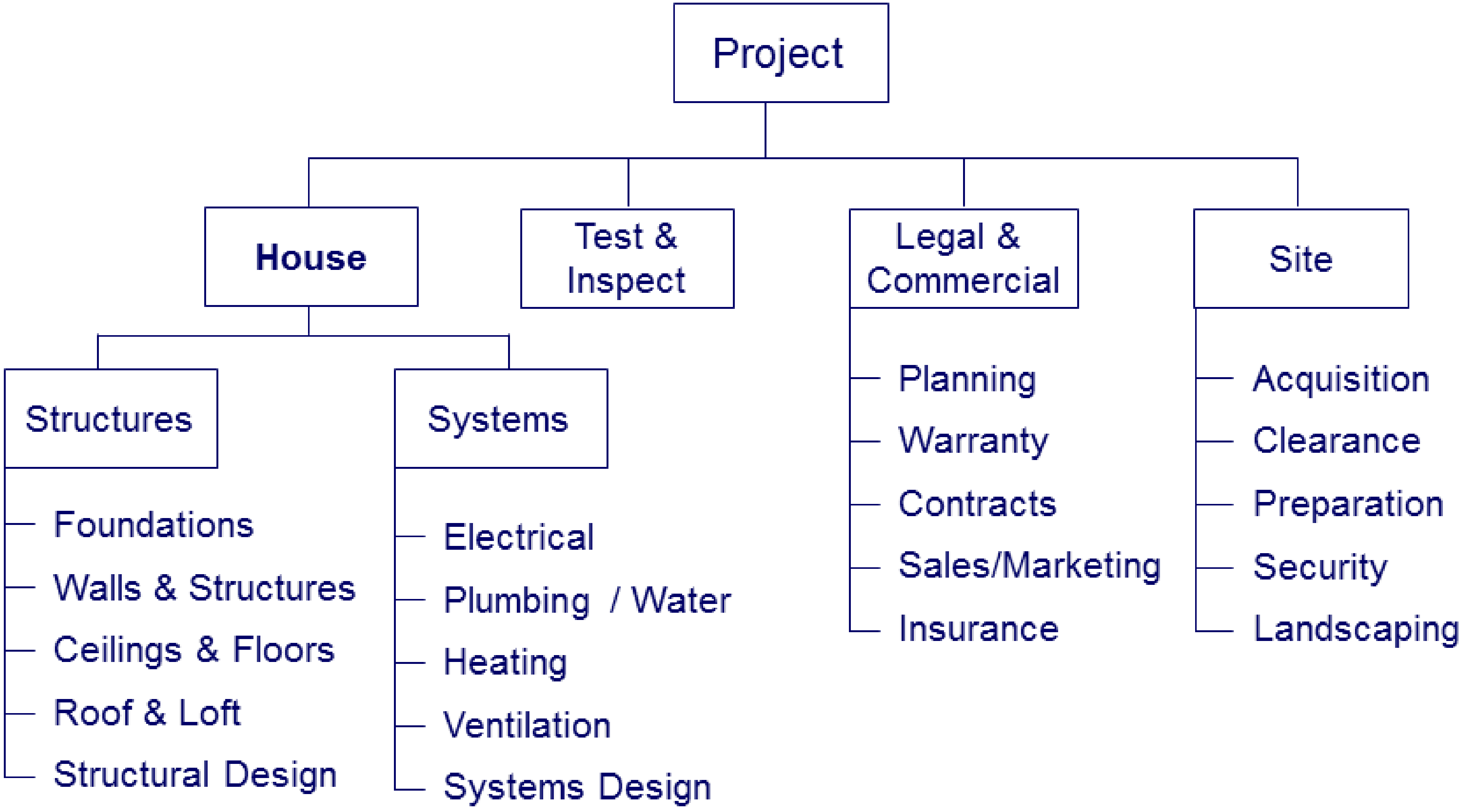
SSADM Product Breakdown Structure for Requirements Specification



- This approach is particularly appropriate if using a methodology such as SSADM (Structured Systems Analysis and Design Method), which clearly specifies, for each step or task, each of the products required and the activities required to produce it.
- The SSADM Reference Manual also supplies generic activity networks and, using the project-specific PBS and derived PFD, these may be used as a basis for developing a project-specific activity network.

The hybrid approach

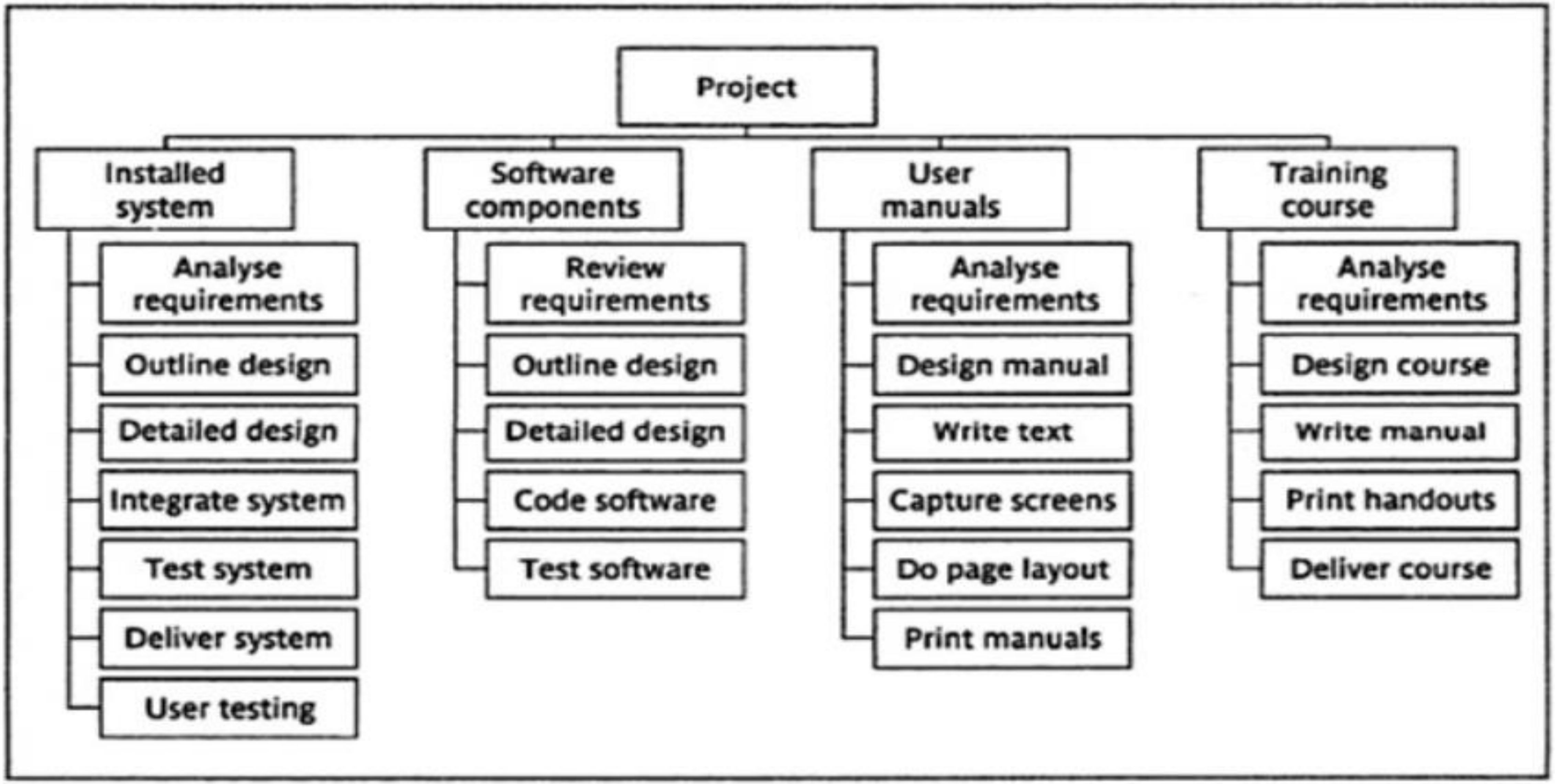
- For each deliverable, a set of activities required to produce that product can be added.
- In a project of any size, it would be beneficial to introduce additional levels - structuring both products and activities.
- The degree to which the structuring is product-based or activity-based might be influenced by the nature of the project and the particular development method adopted.
- As with a purely activity-based WBS, having identified the activities wc are then left with the task of sequencing them.

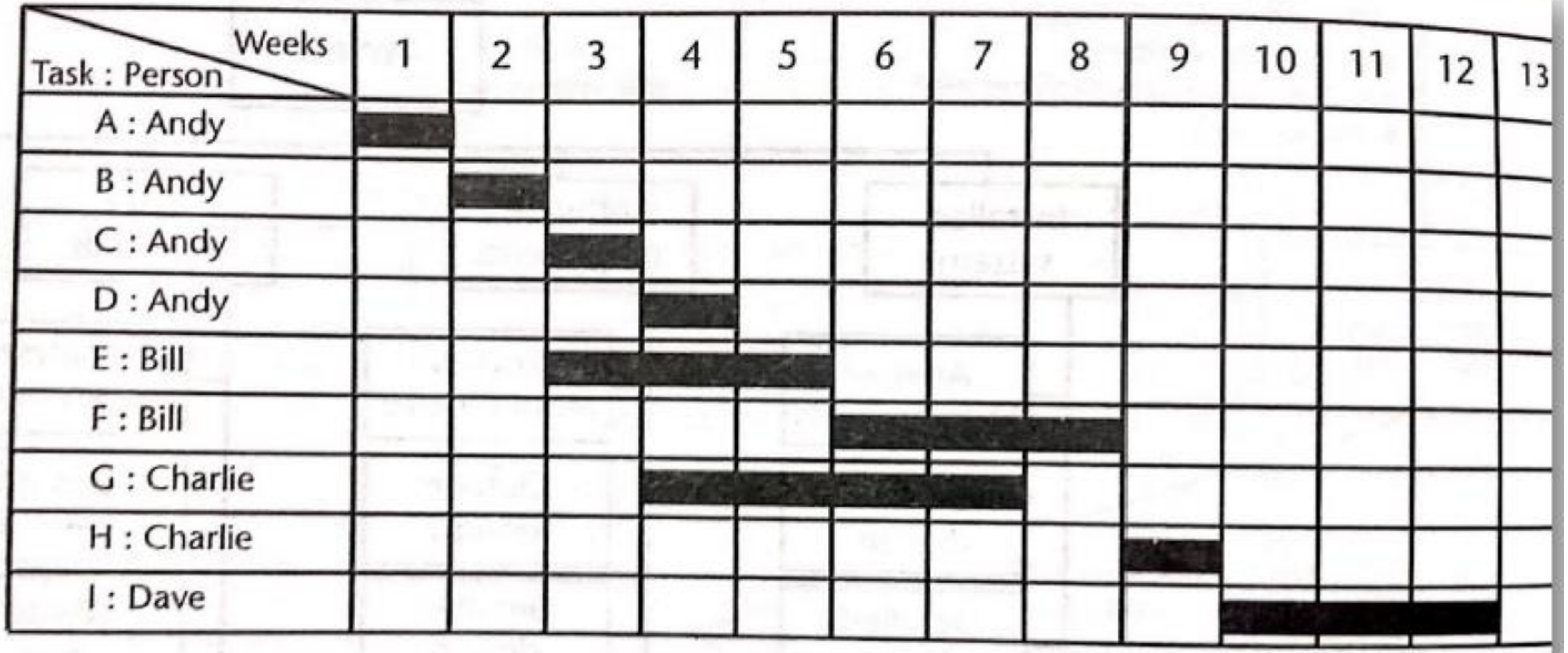

Project
House
Test & Inspect
Legal & Commercial
Site
Structures
Systems
Foundations
Walls & Structures
Ceilings & Floors
Roof & Loft
Structural Design
Electrical
Plumbing / Water
Heating
Ventilation
Systems Design
Planning
Warranty
Contracts
Sales/Marketing
Insurance
Acquisition
Clearance
Preparation
Security
Landscaping

A framework dictating the number of levels and the nature of each level in the structure may be imposed on a WBS. For example, in their MOT (moment of truth) methodology, IBM recommend that the following five levels should be used in a WBS:

- ***Level 1: Project.***
- ***Level 2: Deliverables-*** such as software, manuals and training courses.
- ***Level 3: Components*** -which are the key work items needed to produce deliverables such as the modules and tests required to produce the system software.
- ***Level 4: Work-packages*** which are major work items, or collections of related tasks, required to produce a component.
- ***Level 5; Tasks*** which are tasks that will normally be the responsibility of a single person.

A work breakdown Structure based on deliverables.





Activity key

A : Overall design
B : Specify module 1
C : Specify module 2
D : Specify module 3
E : Code module 1
F : Code module 3
G : Code module 2
H : Integration testing
I : Sytem tesing

- In drawing up the chart, we have done two things:-
    - Sequencing the tasks(i.e., identified the dependencies among activities dictated by the development process)
    - Scheduled them(i.e., specified when they should take place)

- The scheduling has had to take account of availability of staff and the way in which the activities have been allocated to them.

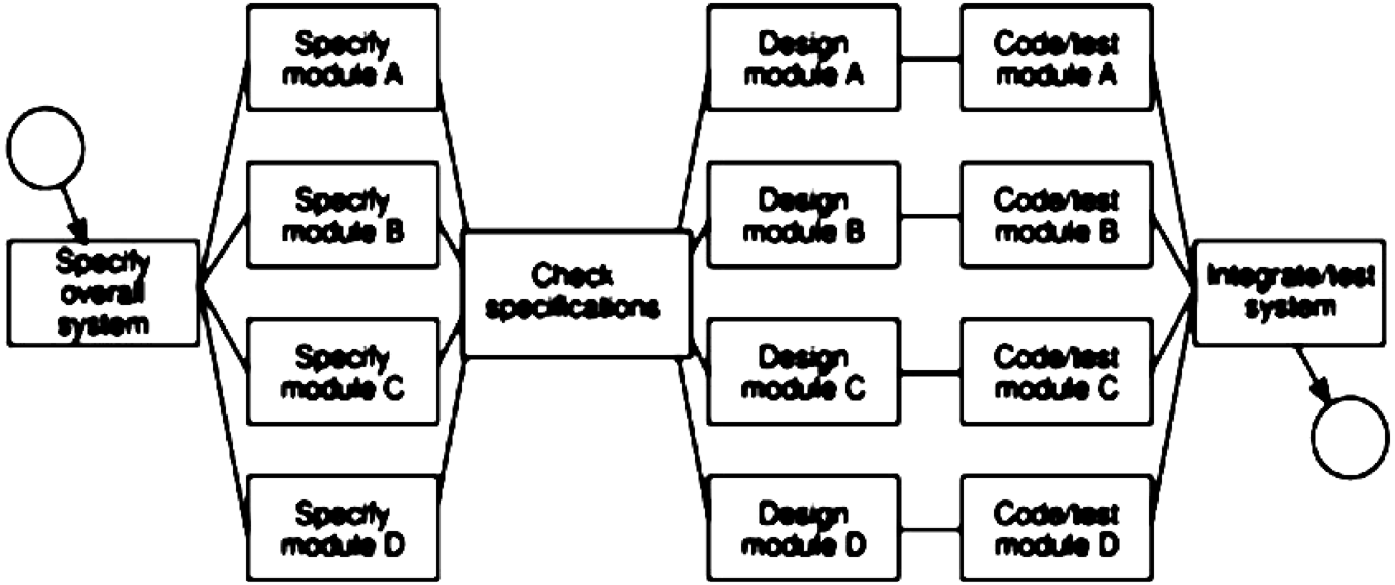
NETWORK PLANNING MODEL

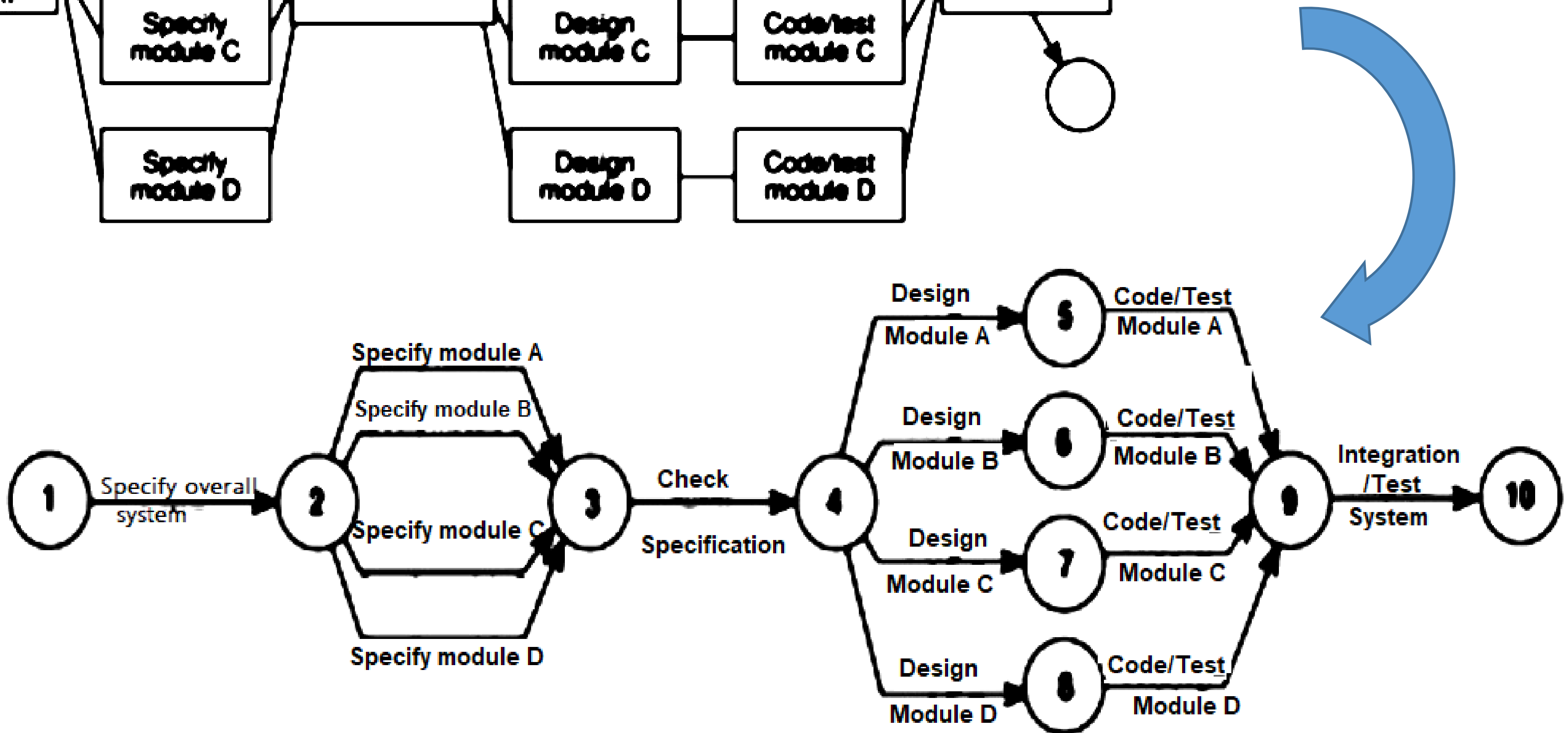
- These project scheduling techniques model the project's activities and their relationships as a network.
- In network, time flows from left to right.
- These technologies were originally developed in the 1950s- the two best known being CPM(critical path method) and PERT (program evaluation review technique).
- Both of these techniques used an activity-on-arrow approach to visualizing the project as a network where activities are drawn as arrows joining circles or nodes, which represent the possible start and/or completion of an activity or set of activities.
- Now, precedence networks has become popular which use activity-on-node networks where activities are represented as nodes and the links between nodes represents precedence (or sequencing) requirements.

Formulating a network model

- The first stage in creating a network model is to represent the activities and their inter-relationships as a graph.
- In CPM we do this by representing activities as links (arrowed lines) in the graph - the nodes (circles) representing the events of activities starting and finishing.

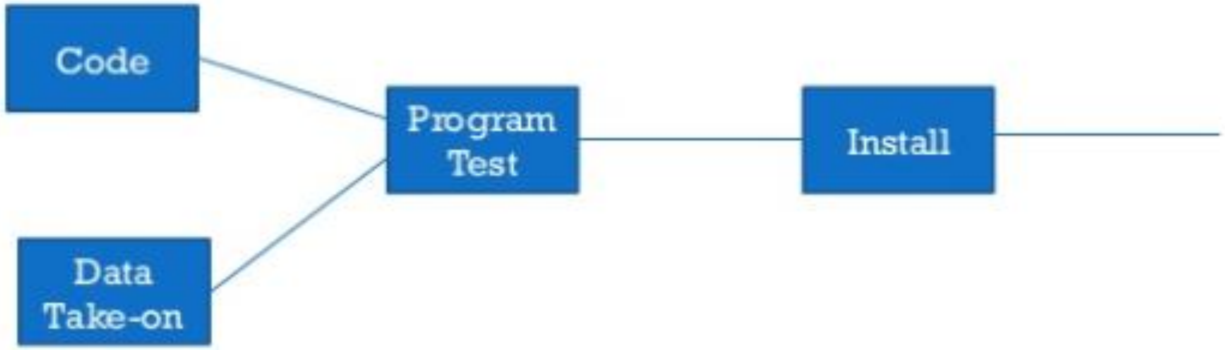
Project activity network fragment to CPM example





Constructing CPM networks

- **A project network should have only one start node**-The start node designates the point at which the project may start. All activities coming from this node may start immediately resources are available - that is, they do not have to wait for any other activities to be completed.
- **A project network should have only one end node**-The end node designates the completion of the project and a project may only finish once.

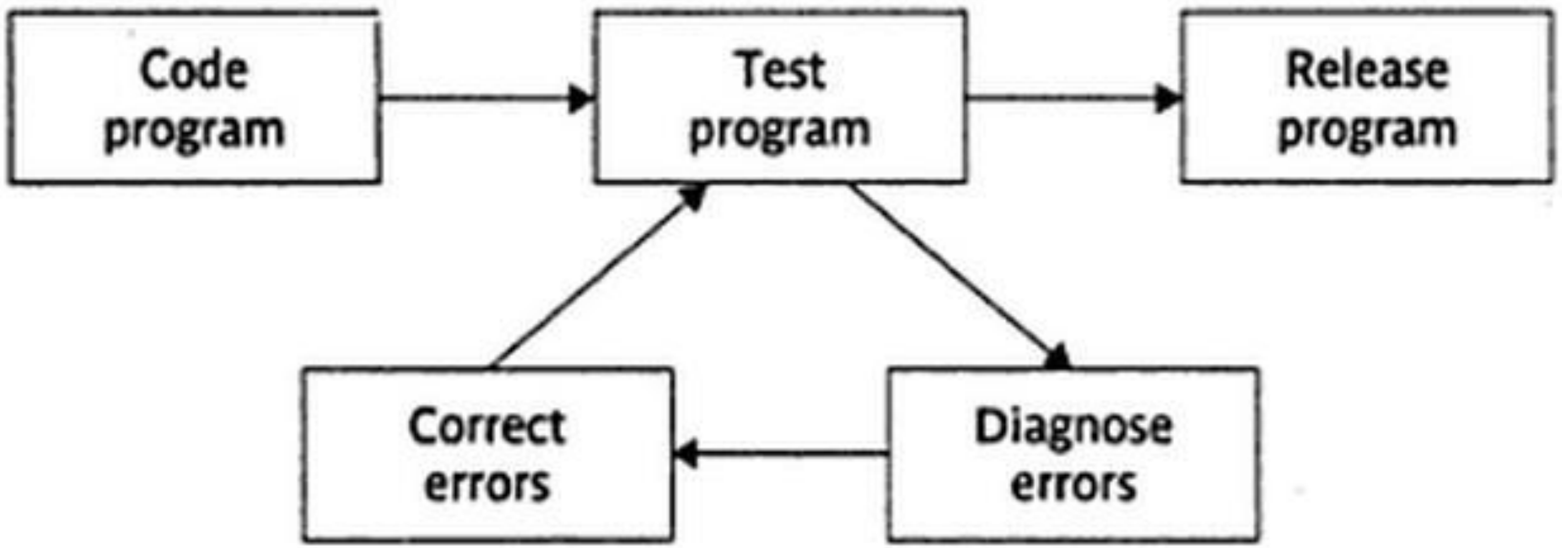


- **Nodes have no duration**- Nodes are events and, as such, are instantaneous points in time. The source node is the event of the project becoming ready to start and the sink node is the event of the project becoming completed.
- Intermediate nodes represent two simultaneous events - the event of all activities leading in to a node have been completed and the event of all activities leading out of that node being in a position to be started.
- **A link has duration**- A link represents an activity and, in general, activities take time to execute.

- **Precedents are the immediate preceding activities- T**he activity Program test cannot start until both Code and Data take-on have been completed and activity Install cannot start until Program test has finished. Code and Data take-on can therefore be said to be precedents of Program test, and Program test is a precedent of Install.

- **Time moves from left to right-**If at all possible, networks are drawn so that time moves from left to right.

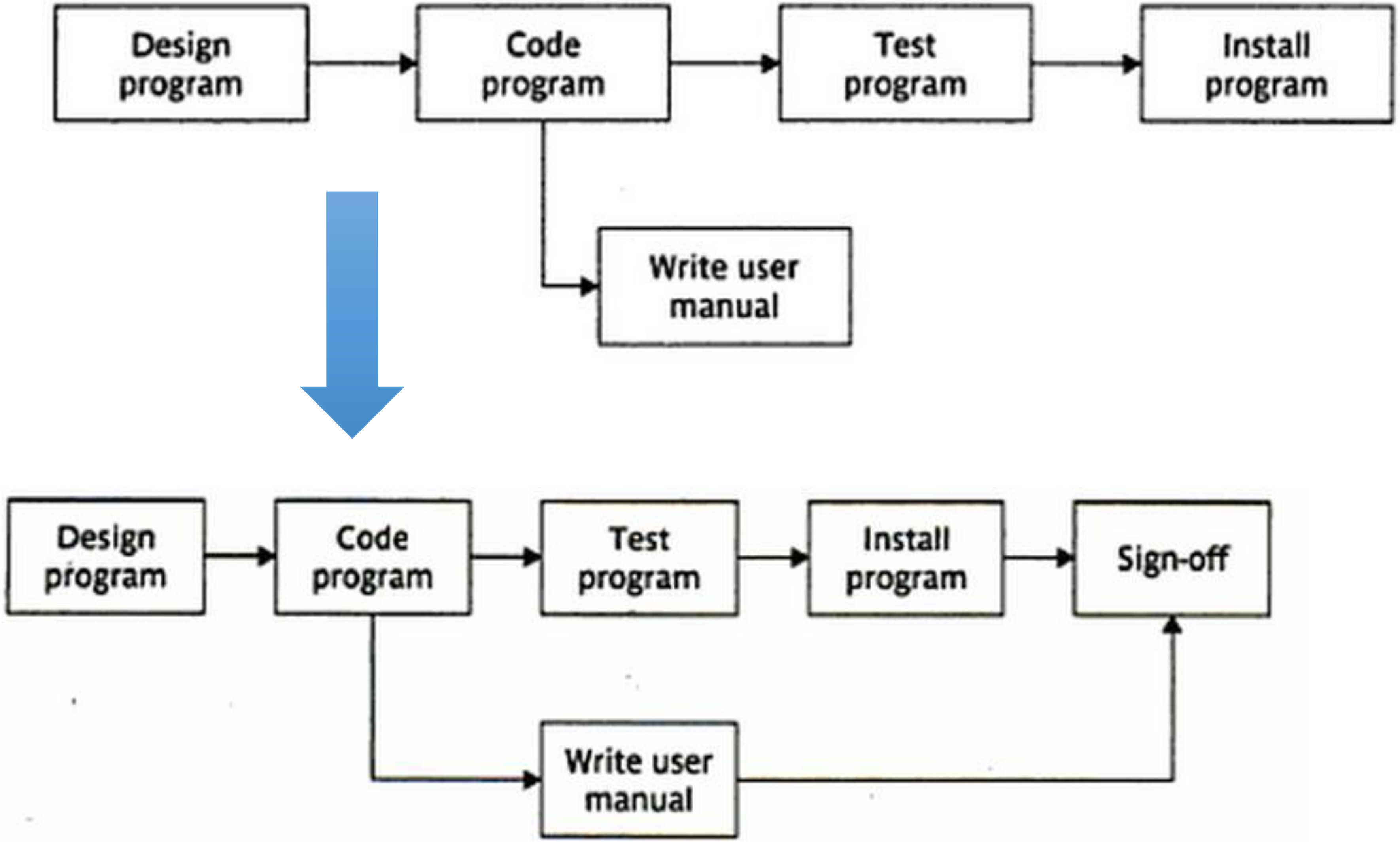
- **A network may not contain loops-** A loop is an error in that it represents a situation that cannot occur in practice. While loops, in the sense of iteration, may occur in practice, they cannot be directly represented in a project network.



- **Nodes are numbered sequentially-** There are no precise rules about node numbering but nodes should be numbered so that head nodes always have a higher number than tail events.

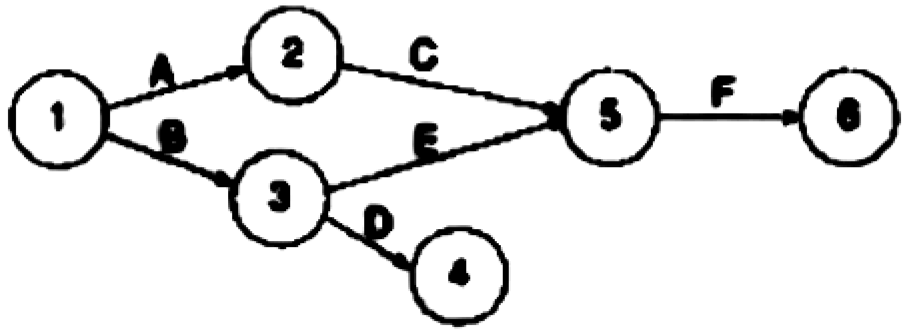
- **A network should not contain dangles.-**A dangling activity such as Write user manual in Figure cannot exist, as it would suggest there are two completion points for the project. In other words, all events, except the first and the last, must have at least one activity entering them and at least one activity leaving them and all activities must start and end with an event.

RESOLVING THE DANGLE

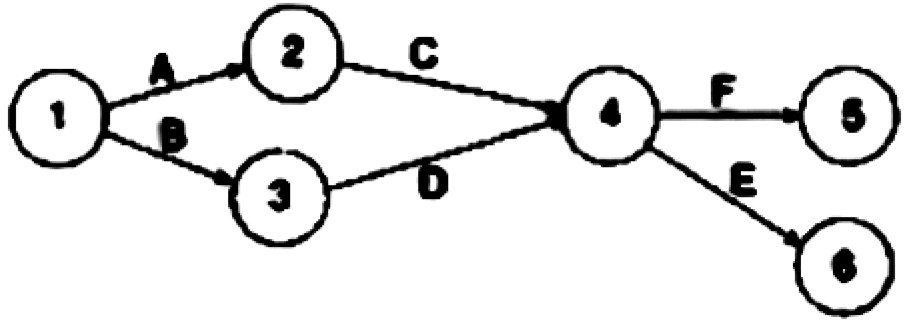


Take a look at network and state what is wrong with each of them and where possible redraw them correctly.

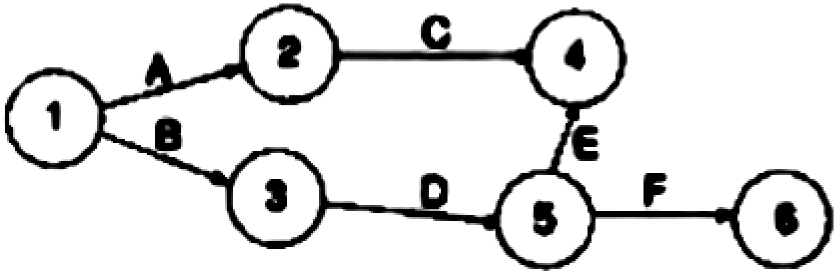
(a)



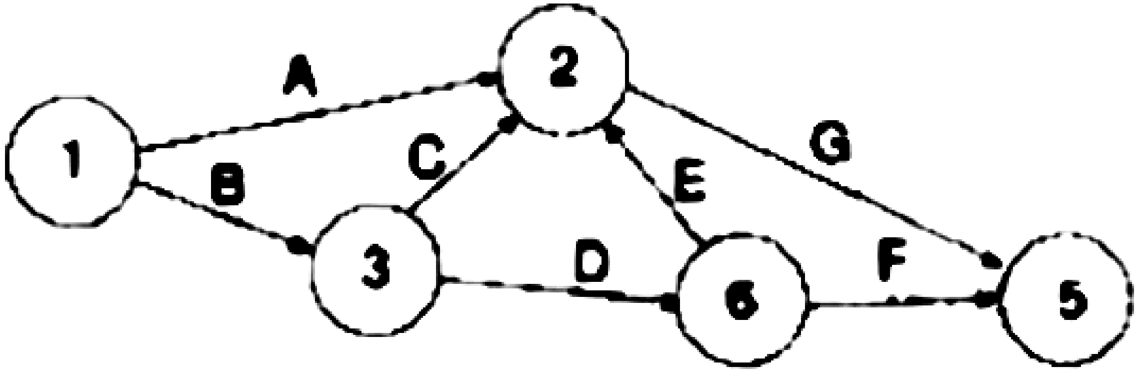
(b)



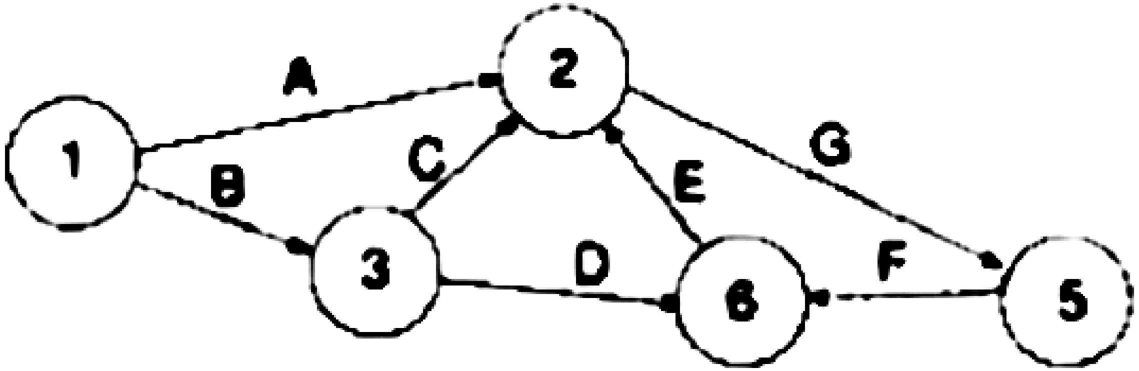
(c)



(d)



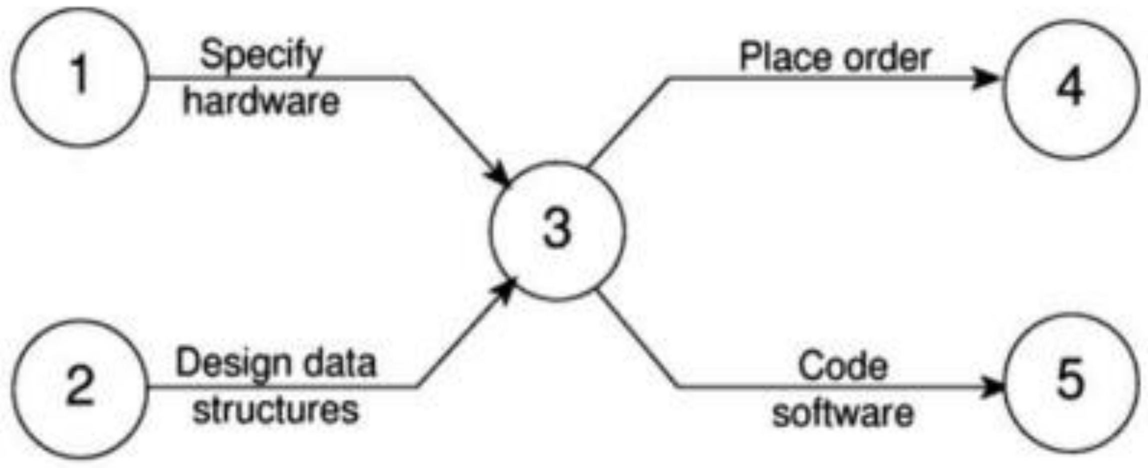
(c)



Using dummy activities

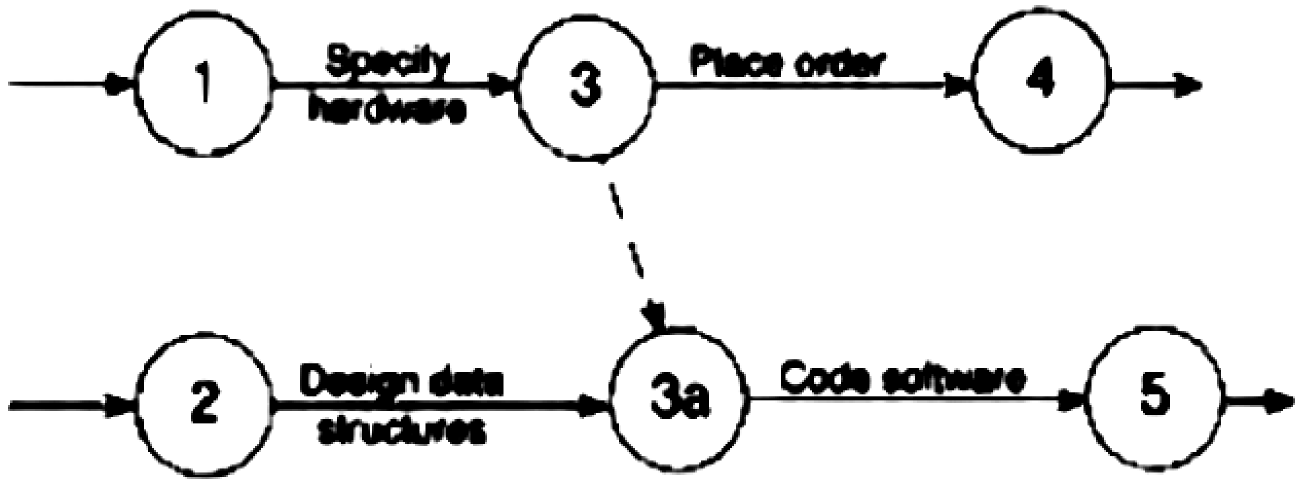
- When two paths within a network have a common event although they are, in other respects independent, a logical error might occur.
- Suppose that, in a particular project, it is necessary to specify a certain piece of hardware before placing an order for it and before coding the software. Before coding the software it is also necessary to specify the appropriate data structures, although clearly we do not need to wait for this to be done before the hardware is ordered.

- Figure is an attempt to model the situation described, although it is incorrect in that it requires both hardware specification and data structure design to be completed before either an order may be placed or software coding may commence.



Two paths with a common node

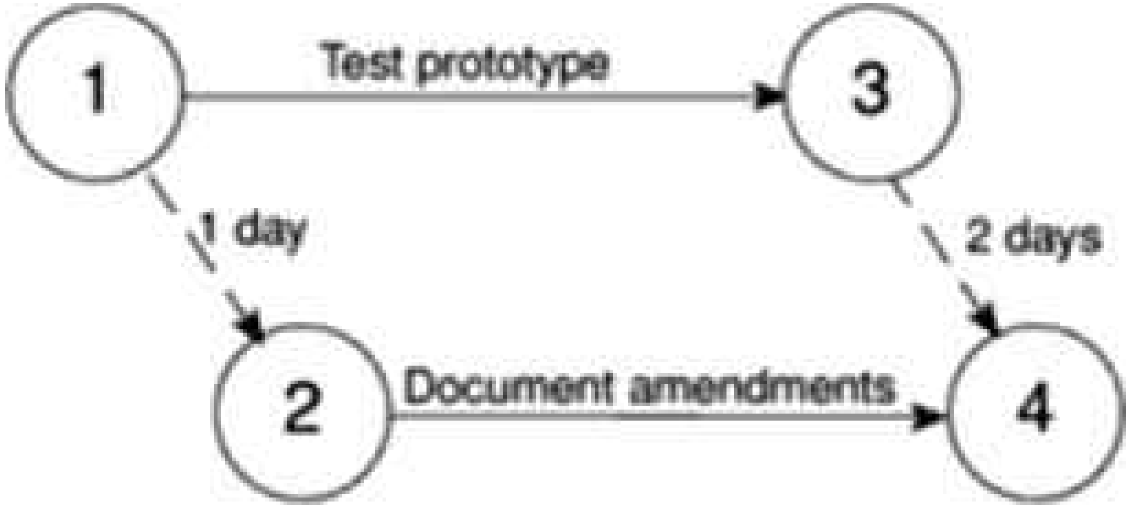
We can resolve this problem by separating the two (more or less) independent paths and introducing a dummy activity to link the completion of data structure design to the start of the activity placing an order. This effectively breaks the link between data structure design and placing the order. Dummy activities shown as dotted lines an the network diagram, have a zero duration and use no resources.



Representing lagged activities

- We might come across situations where we wished to undertake two activities in parallel so long as there is a lag between the two. We might wish to document amendments to a program as it was being tested - particularly if evaluating a prototype.
- In such a case we could designate an activity 'test and document amendments. This would, however, make it impossible to show amendment recording starting after testing had begun and finishing a little after the completion of testing.

- Where parallel activities have a time lag we may show this as a **ladder of activities**. In this case documentation may proceed alongside prototype testing so long as it starts at least a day later. It will finish two days after the completion of prototype testing.



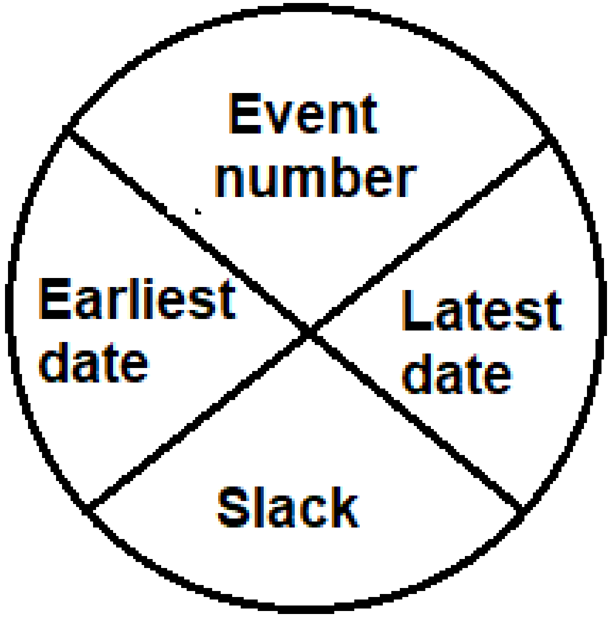
Adding the time dimension

- Having created the logical network model indicating what needs to be done and the interrelationships between those activities, we are now ready to start thinking about when each activity should be undertaken.
- The critical path method is concerned with two primary objectives:
  - planning the project in such a way that it is completed as quickly as possible;
  - identifying those activities where a delay in their execution is likely to affect the overall end date of the project or later activities' start dates.

- The method requires that for each activity we have an estimate of its duration. The network is then analyzed by carrying out a forward pass, to calculate the earliest dates at which activities may commence and the project be completed, and a backward pass, to calculate the latest start dates for activities and the critical path.
- In practice we would use a software application to carry out these calculations for anything but the smallest of projects. It is important, though, that we understand how the calculations are carried out in order to interpret the results correctly and understand the limitations of the method.

CPM conventions

- There are a number of differing conventions that have been adopted for entering information on a CPM network. Typically the diagram is used to record information about the events rather than the activities
.

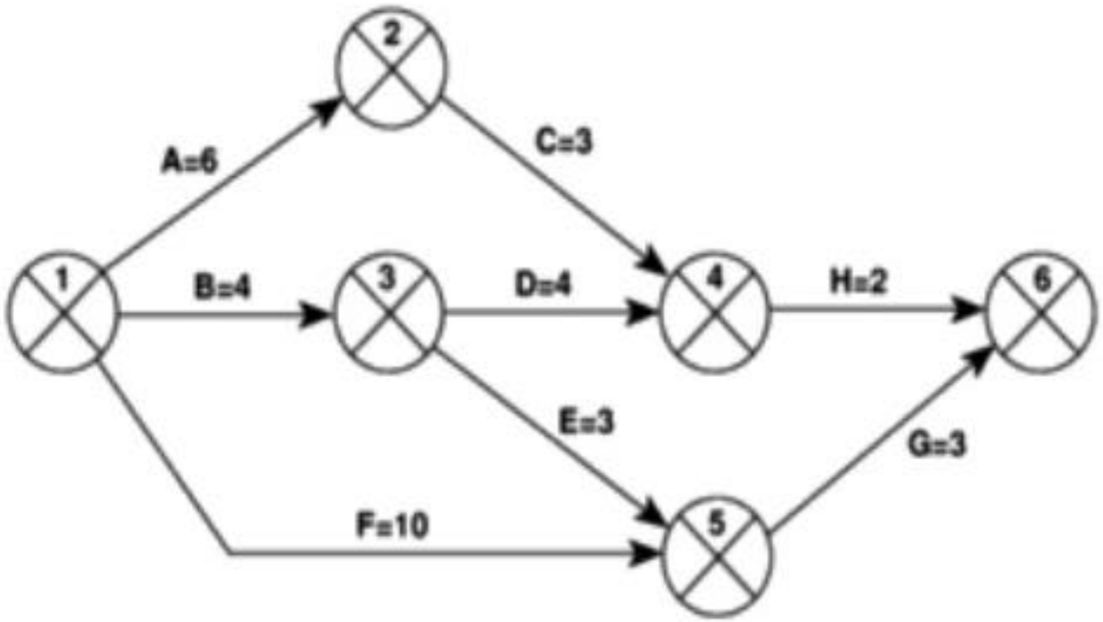


- One of the more common conventions for labelling nodes, and the one adopted here, is to divide the node circle into quadrants and use those quadrants to show the event number, the latest and earliest dates by which the event should occur, and the event slack.

An example project specification with estimated activity durations and precedence requirements

Activity		Duration (weeks)	Precedents
A	Hardware selection	6	
B	System configuration	4	
C	Install hardware	3	A
D	Data migration	4	B
E	Draft office procedures	3	B
F	Recruit staff	10	
G	User training	3	E, F
H	Install and test system	2	C, D

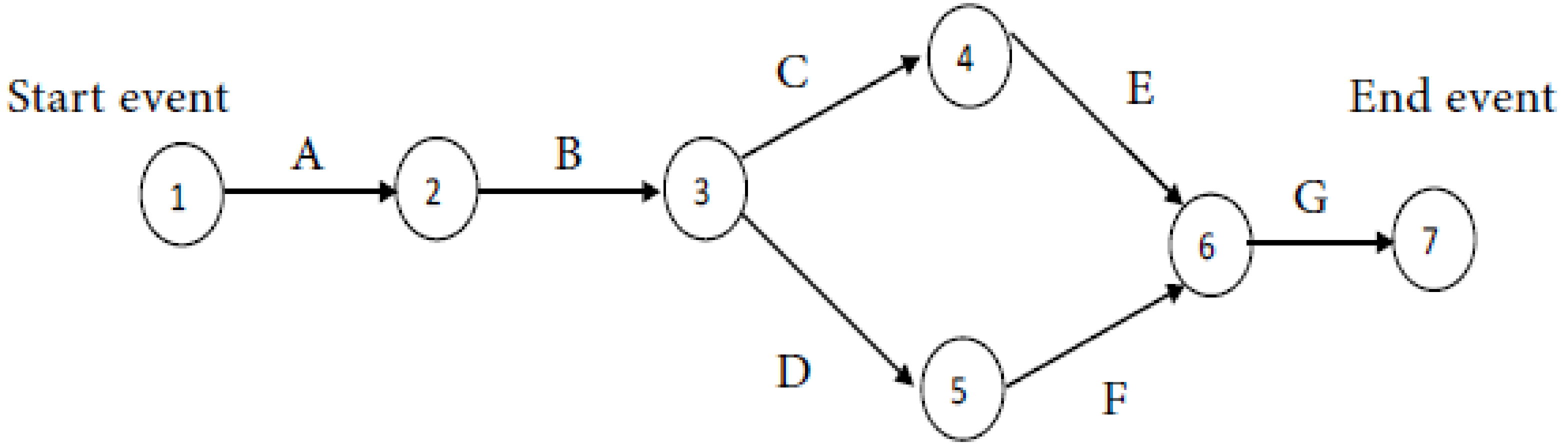
Activity		Duration (weeks)	Precedents
A	Hardware selection	6	
B	System configuration	4	
C	Install hardware	3	A
D	Data migration	4	B
E	Draft office procedures	3	B
F	Recruit staff	10	
G	User training	3	E, F
H	Install and test system	2	C, D



Develop a network diagram for the project specified below

Activity	Immediate Predecessor Activity
A	-
B	A
C, D	B
E	C
F	D
G	E, F

Solution



Construct the network diagram for the following project

Activity	Immediate Predecessor Activity
A	-
B	-
C	A
D	B
E	A
F	C, D
G	E
H	E
I	F, G
J	H, I

Practice questions

Name of Activity	Predecessor Activity	Duration (Weeks)
A	-	8
B	A	13
C	A	9
D	A	12
E	B	14
F	B	8
G	D	7
H	C, F, G	12
I	C, F, G	9
J	E, H	10
K	I, J	7

Activity	Predecessor Activity	Duration (Weeks)
A	-	12
B	A	7
C	A	11
D	A	8
E	A	6
F	B	10
G	C	9
H	D, F	14
I	E, G	13
J	H, I	16

Activity	Predecessor Activity
A	-
B	A
C	A
D	B
E	C
F	C
G	D,F
H	E
I	G,H

Activity	Predecessor Activity
A	-
B	A
C	A
D	A
E	B
F	C
G	D, E, F

SOFTWARE PROJECT MANAGEMENT

Module-3: ACTIVITY PLANNING AND RISK MANAGEMENT
Part-2

CSE4016

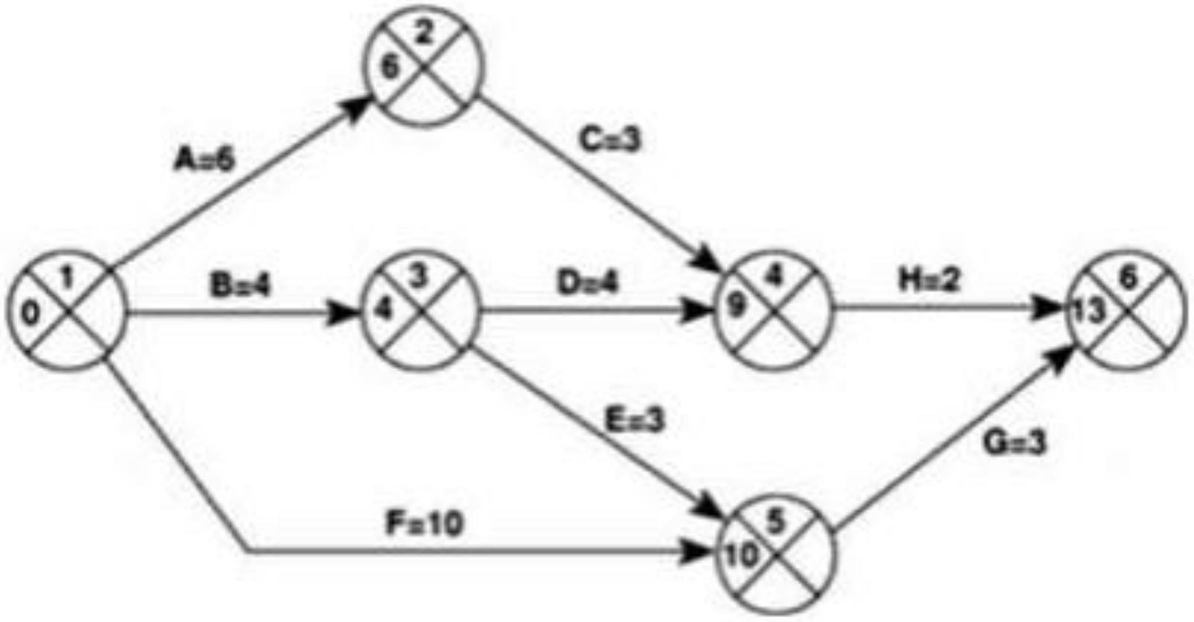
PRIYANKA SINGH

Forward pass

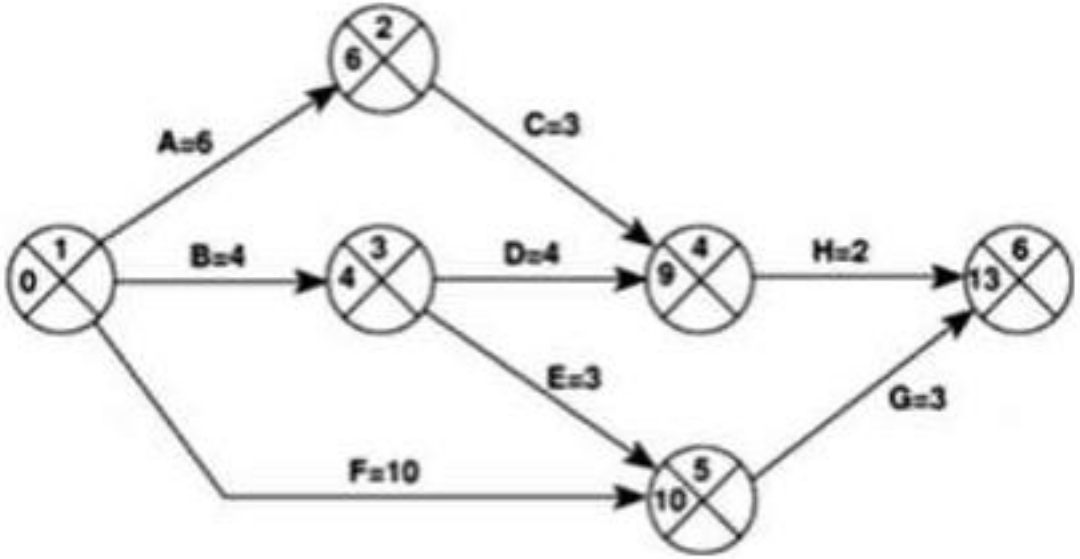
- During the forward pass, earliest dates are recorded as they are calculated. For events, they are recorded on the network diagram and for activities they are recorded on the activity table.
- The forward pass is carried out to calculate the earliest date on which each event may be achieved and the earliest date on which each activity may be started and completed.
- The earliest date for an event is the earliest date by which all activities upon which it depends can be completed.
- By convention, dates indicate the end of the a period and the project is therefore shown as starting in week zero (or the beginning of week I).

Activity		Duration (weeks)	Precedents
A	Hardware selection	6	
B	System configuration	4	
C	Install hardware	3	A
D	Data migration	4	B
E	Draft office procedures	3	B
F	Recruit staff	10	
G	User training	3	E, F
H	Install and test system	2	C, D

A CPM network after the forwardpass



The forward pass and the calculation of earliest start dates is calculated according to the following reasoning.



- Activities A. B and F may start immediately, so the earliest date for event 1 is zero and the earliest start date for these three activities is also zero.
- Activity A will take 6 weeks, so the earliest it can finish is week 6 (recorded in the activity table). Therefore the earliest we can achieve event 2 is week 6.

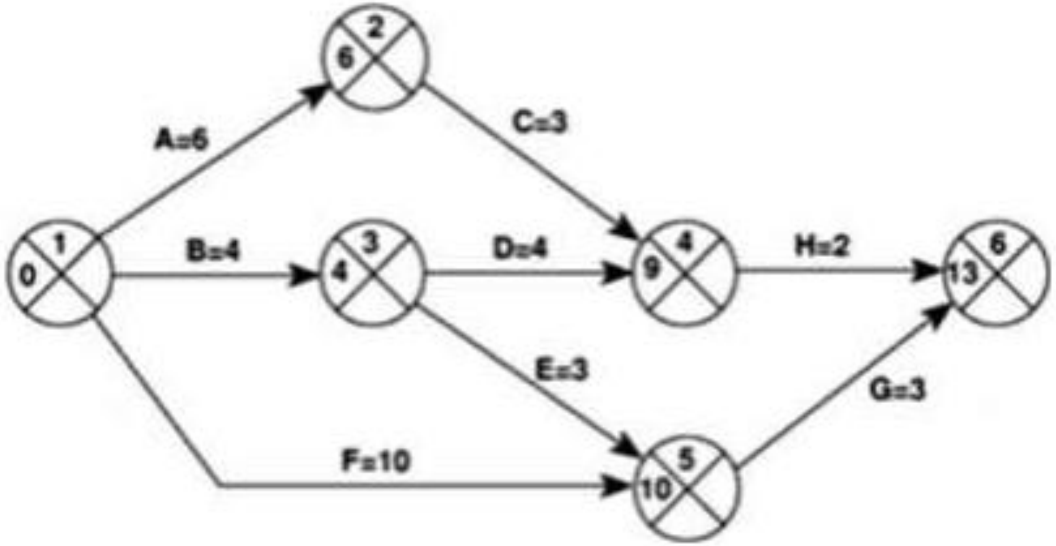
- Activity B w ill take 4 weeks, so the earliest it can finish and the earliest we can achieve event 3 is week 4.
- Activity F will take 10 weeks, so the earliest it can finish is week 10 - we cannot, however, tell whether or not this is also the earliest date that we can achieve event 5 since we have not, as yet, calculated when activity E will finish.

- Activity K can start as early as week 4 (the earliest date for event 3) and, since it is forecasted to take 3 weeks, w ill be completed, at the earliest, at the end of week 7.
- Event 5 may be achieved when both E and F have been completed, that is. week 10 (the later of 7 and 10).
- Similarly we can reason that event 4 will have an earliest date of week 9. This is the later of the earliest finish for activity D (week 8) and the earliest finish for activity C (week 9).
- The earliest date for the completion of the project, event 6, is therefore the end of week 13 - the later of 11 (the earliest finish for H) and 13 (the earliest finish for G).

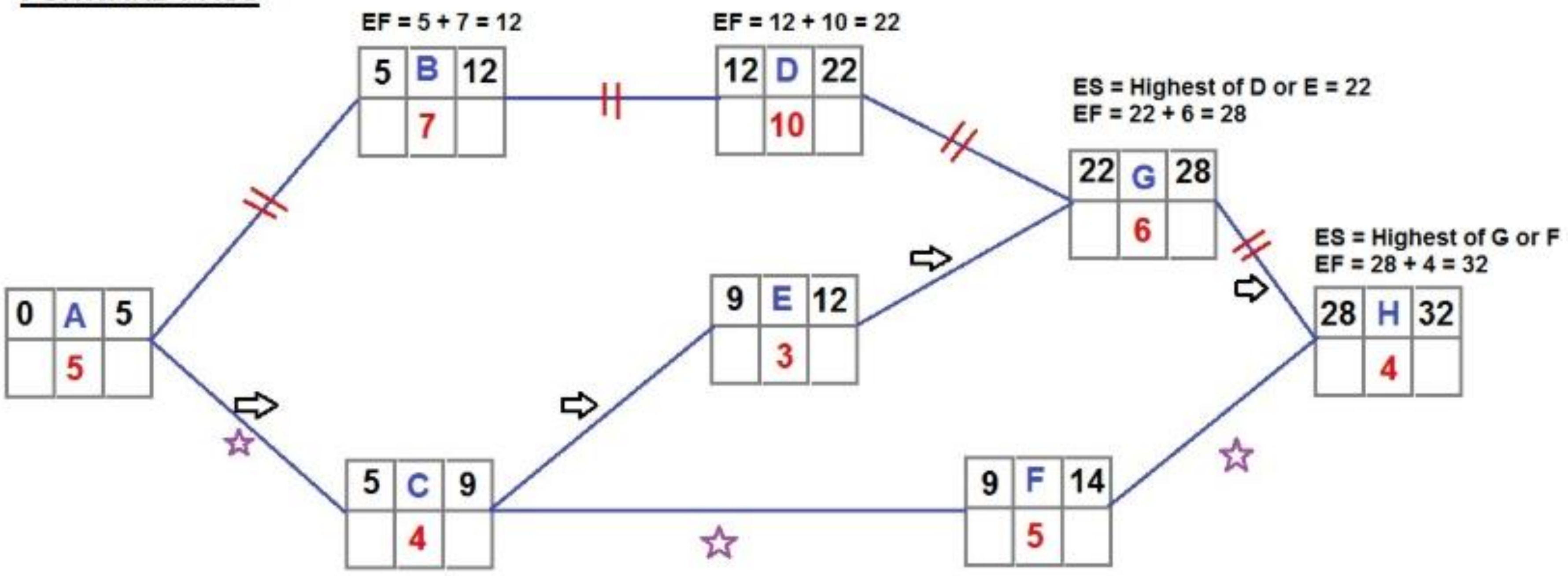
- ***The forward pass rule:*** the earliest date for an event is the earliest finish date for all the activities terminating at that event. Where more than one activity terminates at a common event we take the latest of the earliest finish dates for those activities.

Table 6.2 *The activity table after the forward pass*

Activity	Duration (weeks)	Earliest start date	Latest start date	Earliest finish fate	Latest finish date	Total float
A	6	0		6		
B	4	0		4		
C	3	6		9		
D	4	4		8		
E	3	4		7		
F	10	0		10		
G	3	10		13		
H	2	9		11		



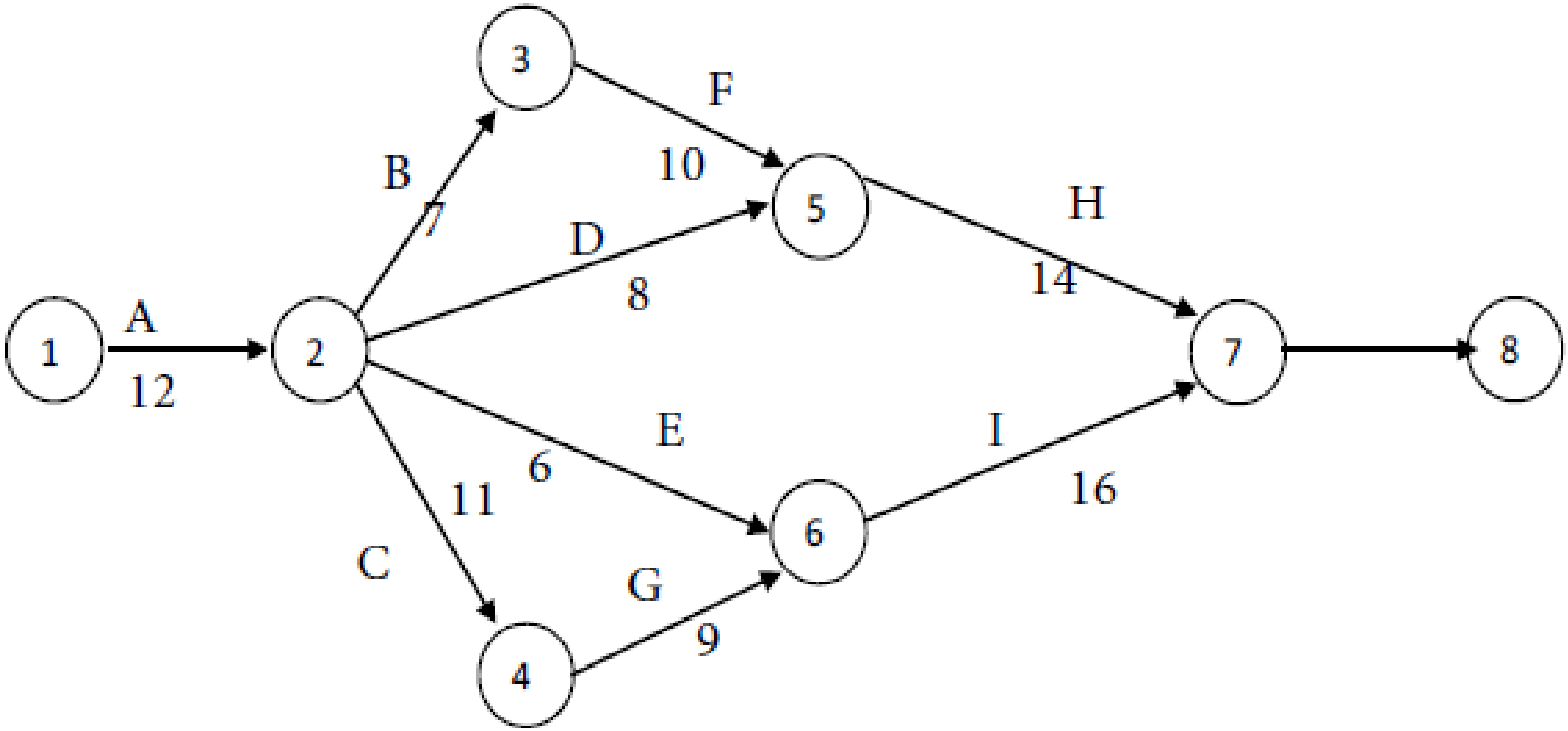
FORWARD PASS



Practice question-1

Activity	Predecessor Activity	Duration (Weeks)
A	-	12
B	A	7
C	A	11
D	A	8
E	A	6
F	B	10
G	C	9
H	D, F	14
I	E, G	13
J	H, I	16

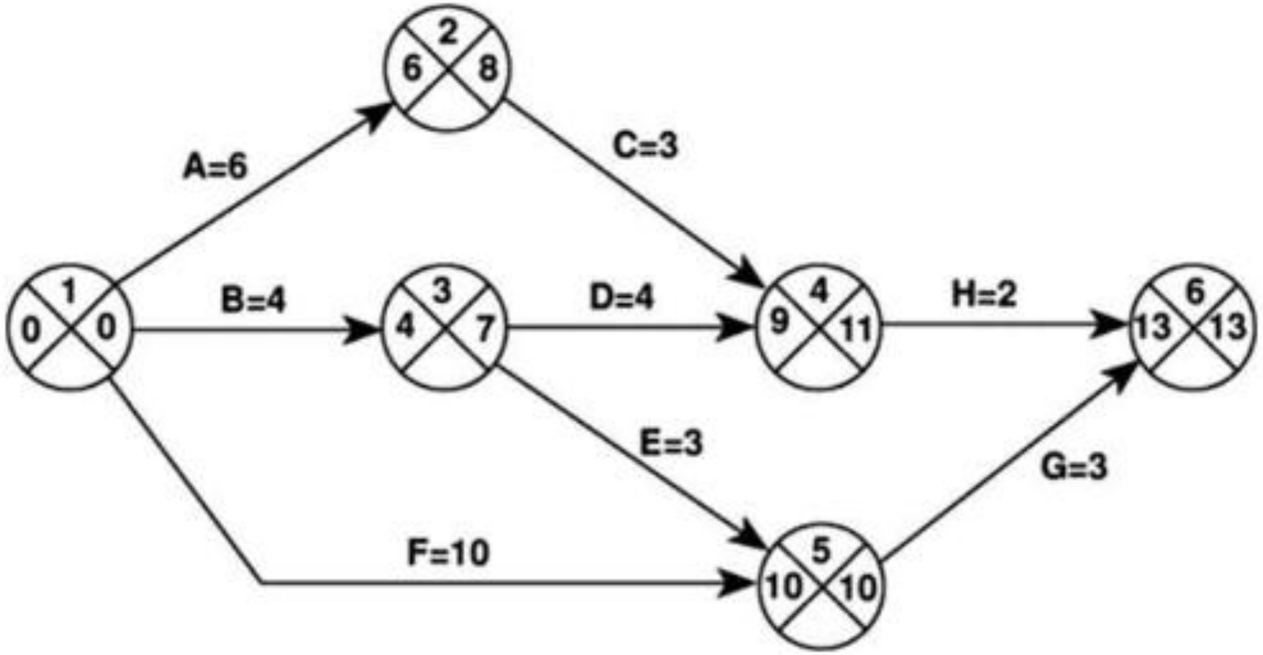
Activity diagram-1

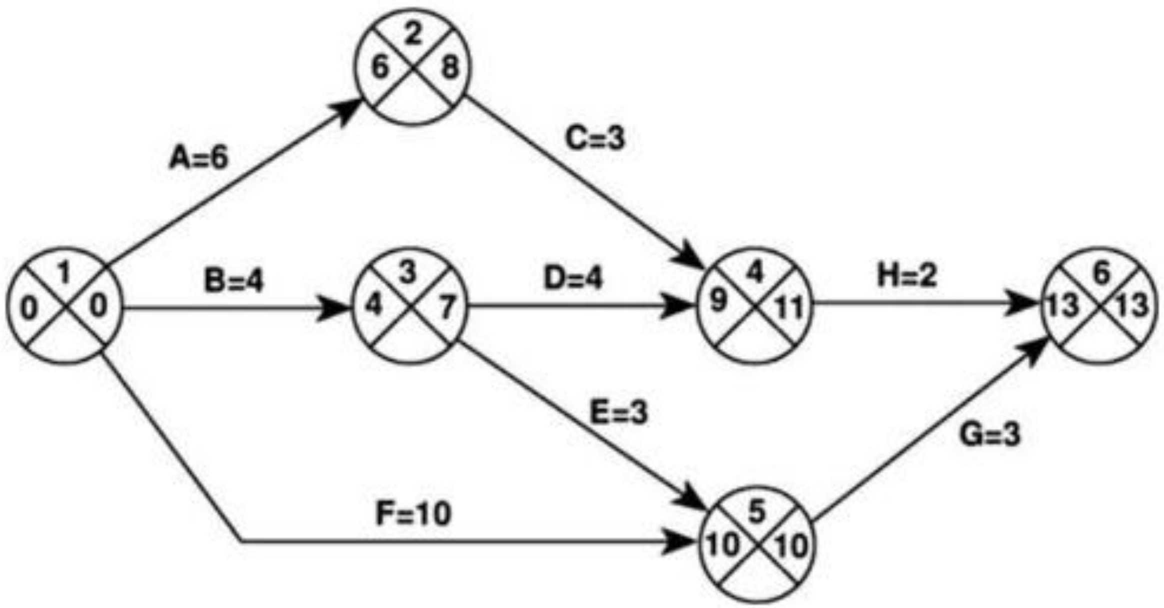


Backward pass

- The second stage is to carry out a backward pass to calculate the latest date at which each event may be achieved, and each activity started and finished, without delaying the end date of the project.
- The latest date for an event is the latest date by which all immediately following activities must be started for the project to be completed on time.
- In calculating the latest dates, we assume that the latest finish date for the project is the same as the earliest finish date - that is, we wish to complete the project as early as possible.

The backward pass rule: the latest date for an event is the latest start date for all the activities that may commence from that event. Where more than one activity commences at a common event we take the earliest of the latest start dates for those activities.





The activity table following the backward pass

Activity	*Duration (weeks)*	*Earliest start date*	*Latest start date*	*Earliest finish fate*	*Latest finish date*	*Total float*
A	6	0	2	6	8	
B	4	0	3	4	7	
C	3	6	8	9	11	
D	4	4	7	8	11	
E	3	4	7	7	10	
F	10	0	0	10	10	
G	3	10	10	13	13	
H	2	9	11	11	13	

The latest event dates are calculated as follows.

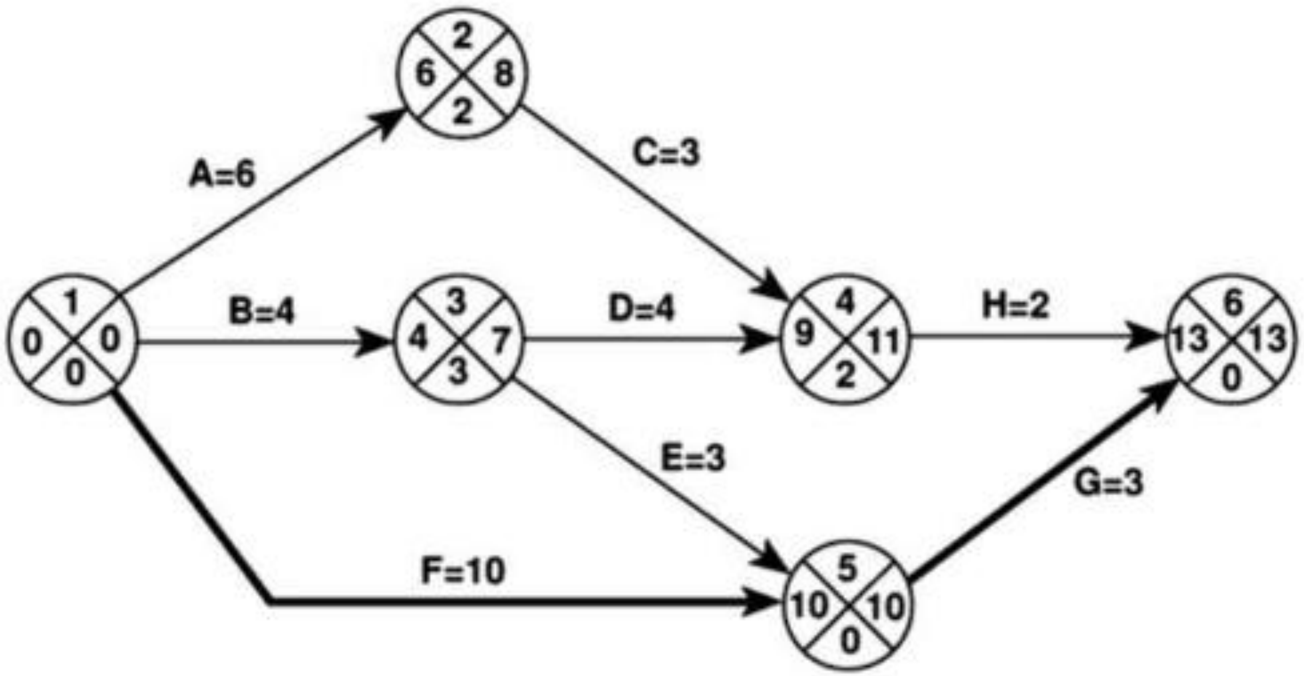
- The latest date for node 6 is assumed to be week 13, the same as the earliest date.
- The latest date for event 5 is week 10, since activity G will take 3 weeks and must be completed by week 13 if the project end date is not to be exceeded.
- The latest date for event 4 is week 11 since activity H does not need to be started until week 11 if it takes 2 weeks and does not need to be completed until week 13.
- The latest date for event 3 is the latest date by which we must be in a position to start both activities D and E. Activity E need not finish until week 10 and need not therefore start until week 7. Activity D need not finish until week 11 and, having a duration of 4 weeks, need not start until week 7. The latest date for event 3 is therefore week 7.
- The latest date for event 2 is week 8 since C, which takes 3 weeks, need not be finished until week 11.
- The earliest and latest dates for the start event must always be the same unless an arithmetic error has occurred.
- The latest date for event 1 is the latest by which we must be in a position to start activity A (which must start by week 2), activity B (which must start by week 3) and activity F (which must start by week 0). This event's latest date is therefore zero. This is, of course, not very surprising since it tells us that if the project does not start on time it won't finish on time.

Identifying the critical path

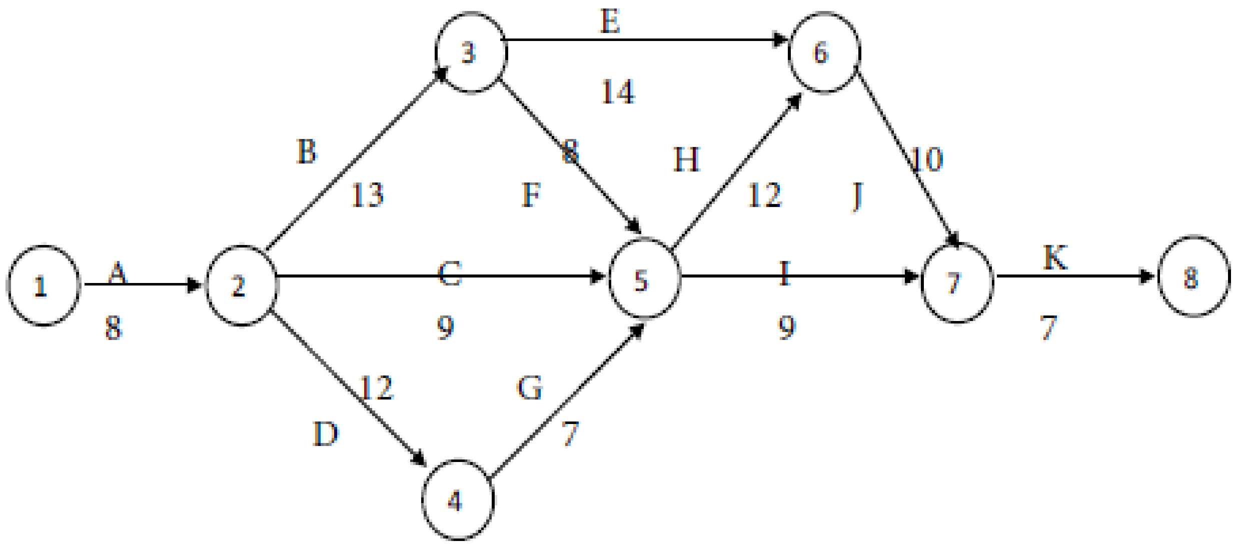
- Any delay on the critical path will delay the project.

- The difference between the earliest date and the latest date for an event is known as the **slack** - it is a measure of how late an event may be without affecting the end date of the project.
- Any event with a slack of zero is critical in the sense that any delay in achieving that event will delay the completion date of the project as a whole.
- There will always be at least one path through the network joining those critical events - this path is known as the **critical path**.

The critical path is the longest path through the network.



Practice question



Determine the earliest and latest times, the total float for each activity, the critical activities, the slacks of the events and the project completion time

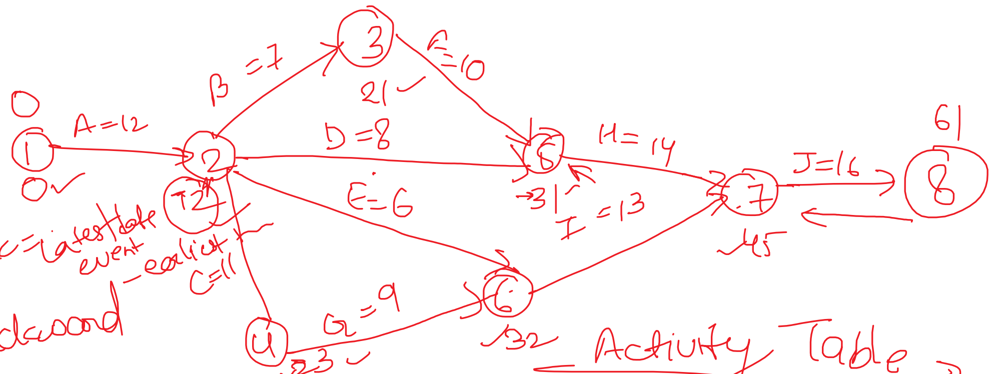
Name of Activity	Predecessor Activity	Duration (Weeks)
A	-	8
B	A	13
C	A	9
D	A	12
E	B	14
F	B	8
G	D	7
H	C, F, G	12
I	C, F, G	9
J	E, H	10
K	I, J	7

The significance of the critical path is two-fold.

- In managing the project, we must pay particular attention to monitoring activities on the critical path so that the ***effects of any delay or resource unavailability are detected and corrected at the earliest opportunity***.

- In planning the project, it is the critical path that we must shorten if we are to reduce the overall duration of the project.

Activity	Predecessor Activity	Duration (Weeks)
A	-	12
B	A	7
C	A	11
D	A	8
E	A	6
F	B	10
G	C	9
H	D, F	14
I	E, G	13
J	H, I	16



Slack = latest date event − earliest

Backward

Node	Latest time
8	61
7	61 − 16 = 45
6	45 − 13 = 32
5	45 − 14 = 31
4	32 − 9 ⇒ 23
3	21
2	min(14, 23, 26, 12) = 12
1	0

Activity Table

Activity	Earliest Start date	Earliest finish date	Latest Start date	Latest finish date	Total float
A	0	12	12−12 = 0	12	0
B	12	19	21−7 = 14	21	2
C	12	23	12	23	0
D	12	20	23	31	11
E	12	18	26	32	14
F	19	29	21	31	2
G	23	32	23	32	0
H	29	43	31	45	2
I	32	45	32	45	0
J	45	61	45	61	0

A, C, G, I, J
↓
critical activity

SOFTWARE PROJECT MANAGEMENT

Module-3: RISK MANAGEMENT Part-3

CSE4016

PRIYANKA SINGH

Estimation errors-

- Some tasks are harder to estimate than others because of the lack of experience of similar tasks or because of the nature of a task.
- Producing a set of user manuals is reasonably straightforward and, given that we have carried out similar tasks previously, we should be able to estimate with some degree of accuracy how long it will take and how much it will cost.
- On the other hand, the time required for program testing and debugging, might be difficult to predict with a similar degree of accuracy - even if we have written similar programs in the past.
- Estimation can be improved by analysing historic data for similar activities and for similar systems.
- Keeping records comparing our original estimates with the final methods of estimation, outcome will reveal the type of tasks that are difficult to estimate correctly.

Planning assumptions

- At every stage during planning, assumptions are made which, if not valid, may put the plan at risk.
- Our activity network, for example, is likely to be built on the assumption of using a particular design methodology - which may be subsequently changed.
- We generally assume that, following coding, a module will be tested and then integrated with others - we might not plan for module testing showing up the need for changes in the original design but, in the event, it might happen.
- At each stage in the planning process, it is important to list explicitly all of the assumptions that have been made and identify what effects they might have on the plan if they are inappropriate.

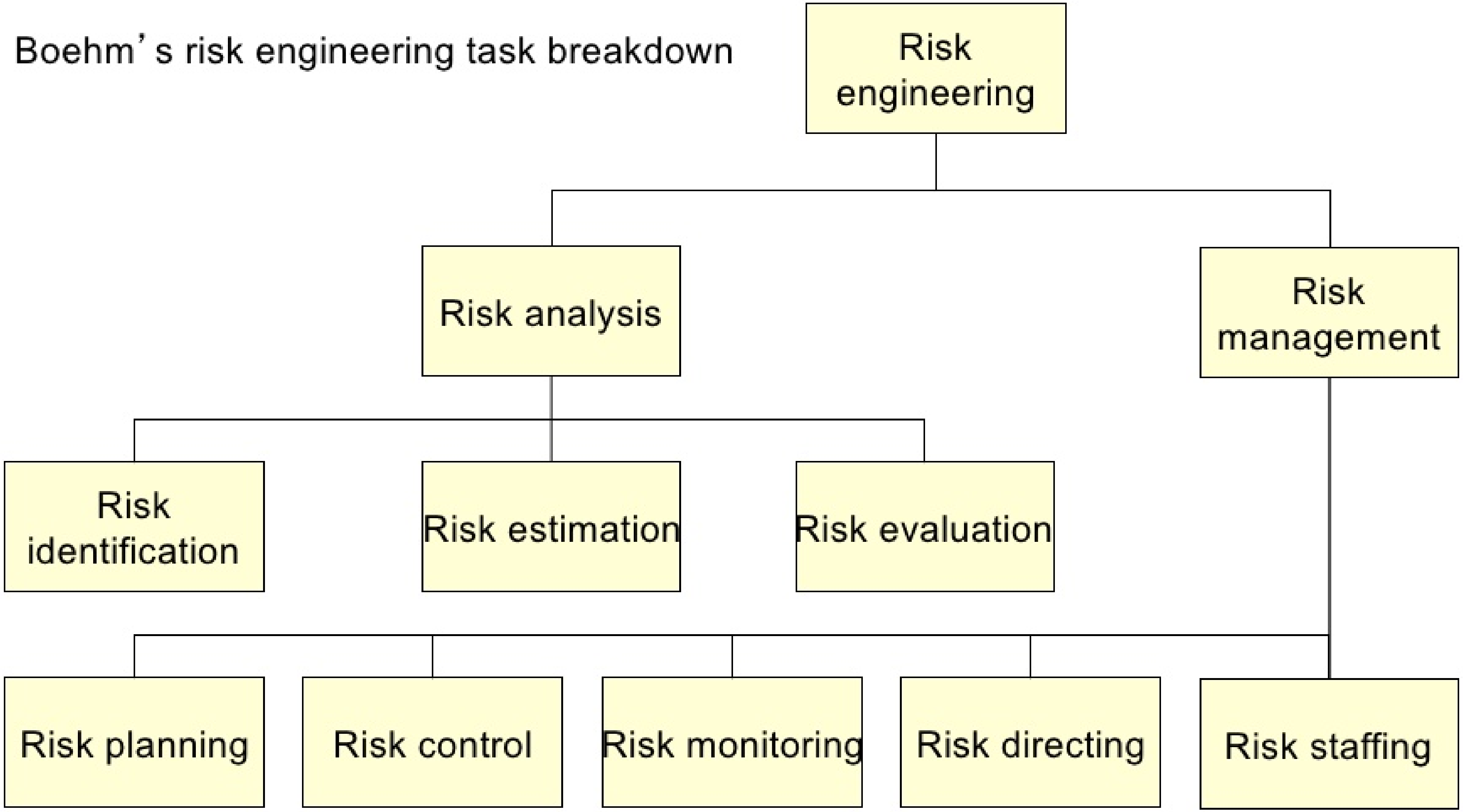
Eventualities

- Some eventualities might never be foreseen and we can only resign ourselves to the fact that unimaginable things do, sometimes, happen.
- They are, however, very rare. The majority of unexpected events can, in fact, be identified - the requirements specification might be altered after some of the modules have been coded, the senior programmer might take maternity leave, the required hardware might not be delivered on time.
- Such events do happen from time to time and, although the likelihood of any one of them happening during a particular project may be relatively low, they must be considered and planned for.

Managing risk

- The objective of risk management is to avoid or minimize the adverse effects of unforeseen events by avoiding the risks or drawing up contingency plans for dealing with them.
- There are a number of models for risk management, but most are similar, in that they identify two main components - risk identification and risk management.
- **Risk identification** consists of listing all of the risks that can adversely affect the successful execution of the project.
- **Risk estimation** consists of assessing the likelihood and impact of each hazard.
- **Risk evaluation** consists of ranking the risks and determining risk aversion strategies.
- **Risk planning** consists of drawing up contingency plans and, where appropriate, adding these to the project's task structure.

Boehm’s risk engineering task breakdown



- **Risk control** concerns the main functions of the risk manager in minimising and reacting to problems throughout the project. This function will include aspects of quality control in addition to dealing with problems as they occur.
- **Risk monitoring** must be an ongoing activity, as the importance and likelihood of particular risks can change as the project proceeds.
- **Risk directing and risk staffing** are concerned with the day-to-day management of risk. Risk aversion and problem solving strategies frequently involve the use of additional staff and this must be planned for and directed.
- Whatever task model or whichever techniques are used, risk management will not be effective unless all project staff are risk-oriented and are provided with an environment where they can freely discuss the risks that might affect a project.

Risk Identification

- The first stage in any risk assessment exercise is to **identify the hazards** that might affect the duration or resource costs of the project.
- A hazard is an event that might occur and will, if it does occur, create a problem for the successful completion of the project.
- In identifying and analyzing risks, we can usefully distinguish between the cause (or hazard), its immediate effect (the problem that it creates) and the risk that it will pose to the project.
- For example, the illness of a team member is a hazard that might result in the problem of late delivery of a component. The late delivery of that component is likely have an effect on other activities and might, particularly if it is on the critical path, put the project completion date at risk.

- A common way of identifying hazards is to use a ***checklist listing all the possible hazards and factors that influence them***. Typical checklists list many, even hundreds, of factors and there are, today, a number of knowledge-based software products available to assist in this analysis.

- Some hazards are **generic risks** - that is, they are relevant to all software projects and standard checklists can be used and augmented from an analysis of past projects to identify them. These will i***nclude risks such as misunderstanding the requirements or key personnel being ill***.

The categories of factors that will need to be considered while identifying the risk-

- ***Application factors-*** The nature of the application of project - whether data used is simple while processing application-critical system or a large distributed system - is likely to be a critical factor. The expected size of the application is also important - the larger the system, the greater is the likelihood of errors and communication and management problems.
- ***Staff factors*** The experience and skills of the staff involved are clearly major. An experienced programmer is, one would hope, less likely to make errors experience was than one with little experience. We must, however, also consider appropriateness of the experience.

Such factors as the level of staff satisfaction and the staff turn-over rates are also important to the success of any project - demotivated staff or key personnel leaving unexpectedly have caused many a project to fail.

- ***Project factors-*** It is important that the project and its objectives are well defined and that they are absolutely clear to all members of the project team and all key stakeholders. Any possibility that this is not the case will pose a risk to the success of the project.
- ***Hardware/software factors -***A project that requires new hardware for development is likely to pose a higher risk than one where the software can be developed on existing (and familiar) hardware. Where a system is developed on one type of hardware or software platform to be used on another there might be additional (and high) risks at installation.
- **Changeover factors-** The need for an 'all-in-one' changeover to the new system poses particular risks. Incremental or gradual changeover minimizes the risks involved but is not always practical. Parallel running can provide a safety net but might be impossible or too costly.

- ***Supplier factors-*** The extent to which a project relies on external organizations that cannot be directly controlled often influences the project's success.
- ***Environment factors-*** Changes in the environment can affect a project's success. A significant change in the taxation regulations could, for example, have serious consequences for the development of a payroll application.
- ***Health and safety factors-*** While not generally a major issue for software projects (compared, say, to civil engineering projects), the possible effects of project activities on the health and safety of the participants and the environment should be considered.
- ***BS 6079 states that 'every project should include an audit of these specific risks before work starts' and that 'audit updates should be scheduled as part of the overall project plan'.***

Risk analysis

- Having identified the risks that might affect our project we need some way of assessing their importance.
- ***Some risks will be relatively unimportant*** (for example, the risk that some of the documentation is delivered a day late), ***whereas some will be of major significance*** (such as the risk that the software is delivered late).
- ***Some are quite likely to occur*** (it is quite likely, for example, that one of the software developers in a team will take a few days sick leave during a lengthy project), ***whereas others are relatively unlikely*** (hardware failure causing loss of completed code, perhaps).
- The ***probability of a hazard's occurring is known as the risk likelihood;*** the effect that the resulting problem will have on the project, if it occurs, is ***known as the risk impact*** and t***he importance of the risk is known as the risk value or risk exposure***.

- The risk value is calculated as:

risk exposure = risk likelihood x risk impact

- The risk exposures for various risks can then be compared with each other to assess the relative importance of each risk and they can be directly compared with the costs and likelihoods of success of various contingency plans.
- However, estimation of these costs and probabilities is likely to be d**ifficult, subjective, time-consuming and costly.** In spite of this, it is valuable to obtain some quantitative measure of risk likelihood and impact because, without these, it is difficult to compare or rank risks in a meaningful way.

- Popular approach is to score the likelihood and impact on a scale of, say, 1 to 10 where the hazard that is most likely to occur receives a score of 10 and the least likely a score of 1.

	Hazard	Likelihood	Impact	Risk exposure
R1	Changes to requirements specification during coding	1	8	8
R2	Specification takes longer than expected	3	7	21
R3	Staff sickness affecting critical path activities	5	7	35
R4	Staff sickness affecting non-critical activities	10	3	30
R5	Module coding takes longer than expected	4	5	20
R6	Module testing demonstrates errors or deficiencies in design	1	10	10

- Managing risk involves the use of two strategies:
  - reducing the risk exposure by reducing the likelihood or impact;
  - drawing up contingency plans to deal with the risk should it occur.

In practice, there are generally other factors, when prioritizing risks.

- **Confidence of the risk assessment** - Some of our risk exposure (RE) assessments will be relatively poor. Where this is the case, there is a need for further investigation before action can be planned.
- **Compound risks-** Some risks will be dependant on others. Where this is the case, they should be treated together as a single risk.
- **The number of risks-** There is a limit to the number of risks that can be effectively considered and acted on by a project manager. We might therefore wish to limit the size of the prioritized list.
- **Cost of action-** Some risks, once recognized, can be reduced or avoided immediately with very little cost or effort and it is sensible to take action on these regardless of their risk value. For other risks we need to compare the costs of taking action with the benefits of reducing the risk.

Risk reduction leverage (RRL) = $(RE_{after} - RE_{before}) / \text{risk reduction cost}$

Reducing the risks

Broadly, there are five strategies for risk reduction.

- ***Hazard prevention-*** Some hazards can be prevented from occurring or their likelihood reduced to insignificant levels. The risk of key staff being unavailable for meetings can be minimized by early scheduling.
- **Likelihood reduction-** Some risks, while they cannot be prevented, can have their likelihoods reduced by prior planning. The risk of late changes to a requirements specification can, be reduced by prototyping. Prototyping will not eliminate the risk of late changes and will need to be supplemented by contingency planning.
- **Risk avoidance-** A project can, for example, be protected from the risk of overrunning the schedule by increasing duration estimates or reducing functionality.

- **Risk transfer-** The impact of some risks can be transferred away from the project by, for example, contracting out or taking out insurance.

- **Contingency planning-** Some risks are not preventable and contingency plans will need to be drawn up to reduce the impact of the hazard occur. A project manager should draw up contingency plans for using agency programmers to minimize the impact of any unplanned absence of programming staff.

Evaluating risks to the schedule

- We have seen that not all risks can be eliminated - even those that are classified as avoidable or manageable can, in the event, still cause problems affecting activity durations.
- By identifying and categorizing those risks, and in particular, their likely effects on the duration of planned activities, we can assess what impact they are likely to have on our activity plan.

Using PERT to evaluate the effects of uncertainty

PERT (program evaluation and review technique) was published in the same year as CPM. Developed for the Fleet Ballistic Missiles Program it is said to have saved considerable time in development of the Polaris missile.

The method is very similar to the CPM technique but, instead of using a single estimate for the duration of each task, PERT requires three estimates.

- ***Most likely time(m)*** - the time we would expect the task to take under normal circumstances.
- ***Optimistic time (a)*** - the shortest time in which we could expect to complete the activity, barring outright miracles.
- ***Pessimistic time (b)*** - the worst possible time allowing for all reasonable eventualities but excluding 'acts of God and warfare'.

PERT then combines these three estimates to form a single expected duration, I_e, using the formula

$$t_e = (a+4m+b)/6$$

Using expected durations

- The expected durations are used to carry out a forward pass through a network; using the same method as the CPM technique.
- In this case, however, the calculated event dates are not the earliest possible dates but are the dates by which we expect to achieve those events.

Activity		Duration (weeks)	Precedents
A	Hardware selection	6	
B	System configuration	4	
C	Install hardware	3	A
D	Data migration	4	B
E	Draft office procedures	3	B
F	Recruit staff	10	
G	User training	3	E, F
H	Install and test system	2	C, D

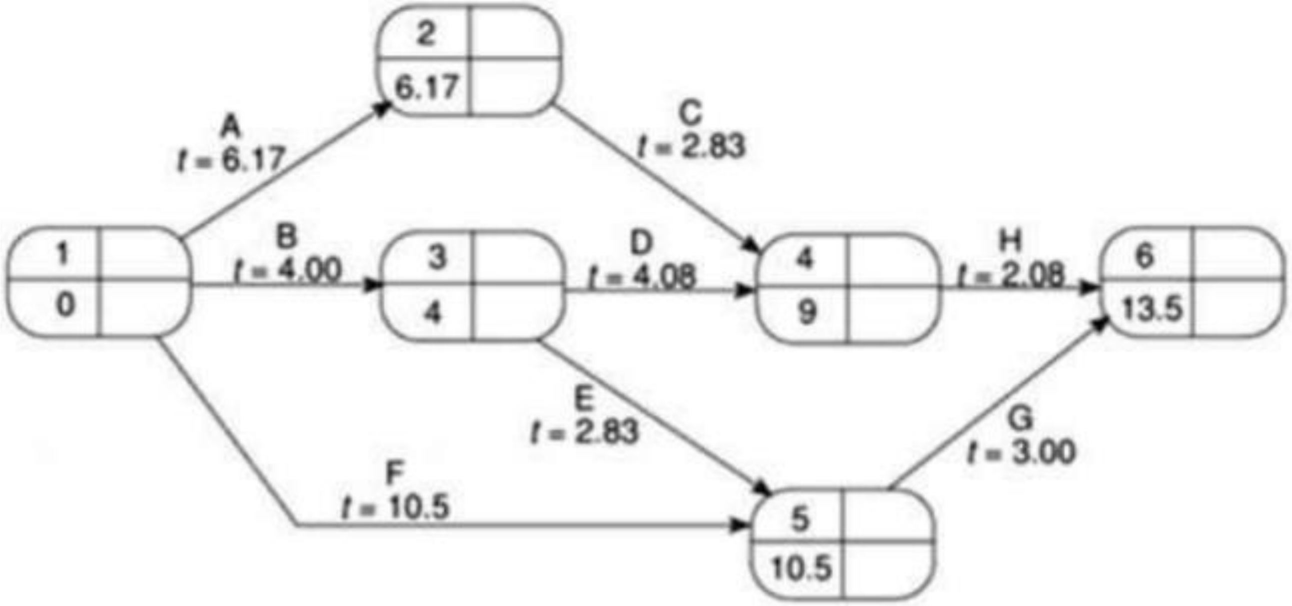
-

	Activity durations (weeks)		
Activity	Optimistic (a)	Most likely (m)	Pessimistic (b)
A	5	6	8
B	3	4	5
C	2	3	3
D	3.5	4	5
E	1	3	4
F	8	10	15
G	2	3	4
H	2	2	2.5

Pert labeling convention

Event Number	Target Date
Expected Date	Standard deviation

The PERT network illustrated in Figure indicates that we expect the project to take13.5 weeks. An advantage of this approach is that it places an emphasis on the uncertainty of the real world.



Activity standard deviations

A ***quantitative measure of the degree of uncertainty of an activity duration*** estimate maybe obtained by calculating the standard deviation s of an activity time, using the formula

$$\mathbf{s=(b-a)/6}$$

- The activity standard deviation is proportional to the difference between the optimistic and pessimistic estimates, and can be used as a ranking measure of the degree of uncertainty or risk for each activity.

Expected durations and standard deviations

Table Expected times and standard deviations

Activity duration (weeks)					
Activity	Optimistic (a)	Most likely (m)	Pessimistic (b)	Expected (t_e)	Standard deviation(s)
A	5	6	8	6.17	0.50
B	3	4	5	4.00	0.33
C	2	3	3	2.83	0.17
D	3.5	4	5	4.08	0.25
E	1	3	4	2.83	0.50
F	8	10	15	10.50	1.17
G	2	3	4	3.00	0.33
H	2	2	2.5	2.08	0.08

The likelihood of meeting targets

- The main advantage of the PERT technique is that it provides a method for estimating the probability of meeting or missing target dates. There might be only a single target date - the project completion - but we might wish to set additional intermediate targets.
- Suppose that we must complete the project within 15 weeks at the outside. We expect it will take 13.5 weeks but it could take more or, perhaps, less. In addition, suppose that activity C must be completed by week 10, as it is to be carried out by a member of staff who is scheduled to be working on another project and that event 5 represents the delivery of intermediate products to the customer. These three target dates are shown on the PERT network in Figure 7.4.
- The PERT technique uses the following three-step method for calculating the probability of meeting or missing a target date:
    - calculate the standard deviation of each project event;
    - calculate the z value for each event that has a target date;
    - convert z values to a probabilities.

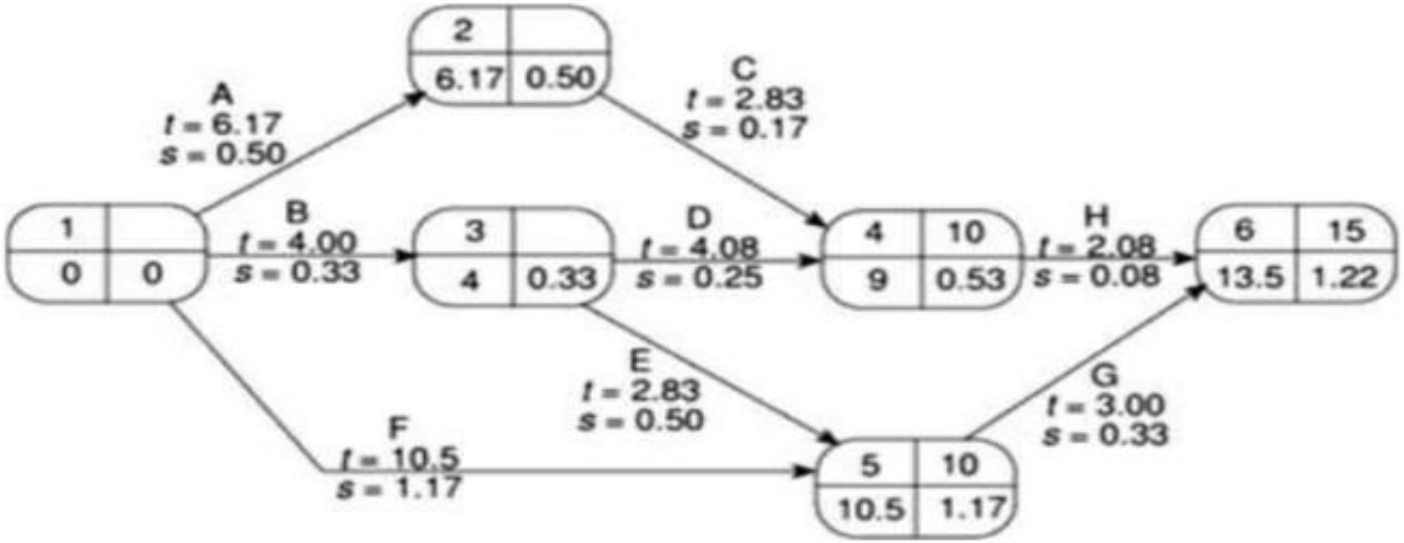
Calculating the standard deviation of each project event

- Standard deviations for the project events can be calculated by carrying out a forward pass using the activity standard deviations in a manner similar to that used with expected durations. There is, however, one small difference - to add two standard deviations we must add their squares and then find the square root of the sum.
- The square of the standard deviation is known as the variance. Standard deviations may not be added together but variances may.
- The standard deviation for event 3 depends solely on that of activity B. For event 5 there are two possible paths, B + E or F. The total standard deviation for path B + E is $\sqrt{(0.33^2 + 0.50^2)} = 0.6$ and that for path F is 1.17; the standard deviation for event 5 is therefore the greater of the two, 1.17.

Table Expected times and standard deviations

	Activity duration (weeks)				
Activity	Optimistic (a)	Most likely (m)	Pessimistic (b)	Expected (t_e)	Standard deviation(s)
A	5	6	8	6.17	0.50
B	3	4	5	4.00	0.33
C	2	3	3	2.83	0.17
D	3.5	4	5	4.08	0.25
E	1	3	4	2.83	0.50
F	8	10	15	10.50	1.17
G	2	3	4	3.00	0.33
H	2	2	2.5	2.08	0.08

The PERT network with three target dates and calculated event standard deviations.



Calculating the z values

- The z value is calculated for each node that has a target date. It is equivalent to the number of standard deviations between the node's expected and target dates. It is calculated using the formula s where t_e is the expected date and T the target date.

$$\mathbf{z = (T - t_e)/s}$$

- The z value for event 4 is (10 - 9.00) / 0.53 = 1.8867.
- PERT focuses attention on the uncertainity of forecasting. PERT can be used to calculatethe standard deviation for each task and use this to rank them according to their degree of risk.

Monte Carlo simulation

- As an alternative to the PERT technique, and to provide a greater degree of flexibility in specifying likely activity durations, we can use Monte Carlo simulation techniques to evaluate the risks of not achieving deadlines.
- Monte Carlo Analysis is a risk management technique that is used for conducting a quantitative analysis of risks. It’s meant to be used to analyze the impact of risks on your project. For example, if this risk occurs, how will it affect our schedule and/or the cost of the project? Monte Carlo gives you a range of possible outcomes and probabilities to allow you to consider the likelihood of different scenarios.
- For example, let’s say you don’t know how long your project will take. You have a rough estimate of the duration of each project task. Using this, you also come up with a best-case scenario (optimistic) and worst case scenario (pessimistic) duration for each task.

- You can then use Monte Carlo to analyze all the potential combinations and give you probabilities on when the project will complete.
- The results would be something like this:
  - 2% chance of completing the project in 12 months.
  - 15% chance of completion within 13 months.
  - 55% chance of completion within 14 months.
  - 95% chance of completion within 15 months.
  - 100% chance of completion within 16 months. (If everything took as long as the pessimistic estimates.)
- Using this information, you can now better estimate your timeline and plan your project.

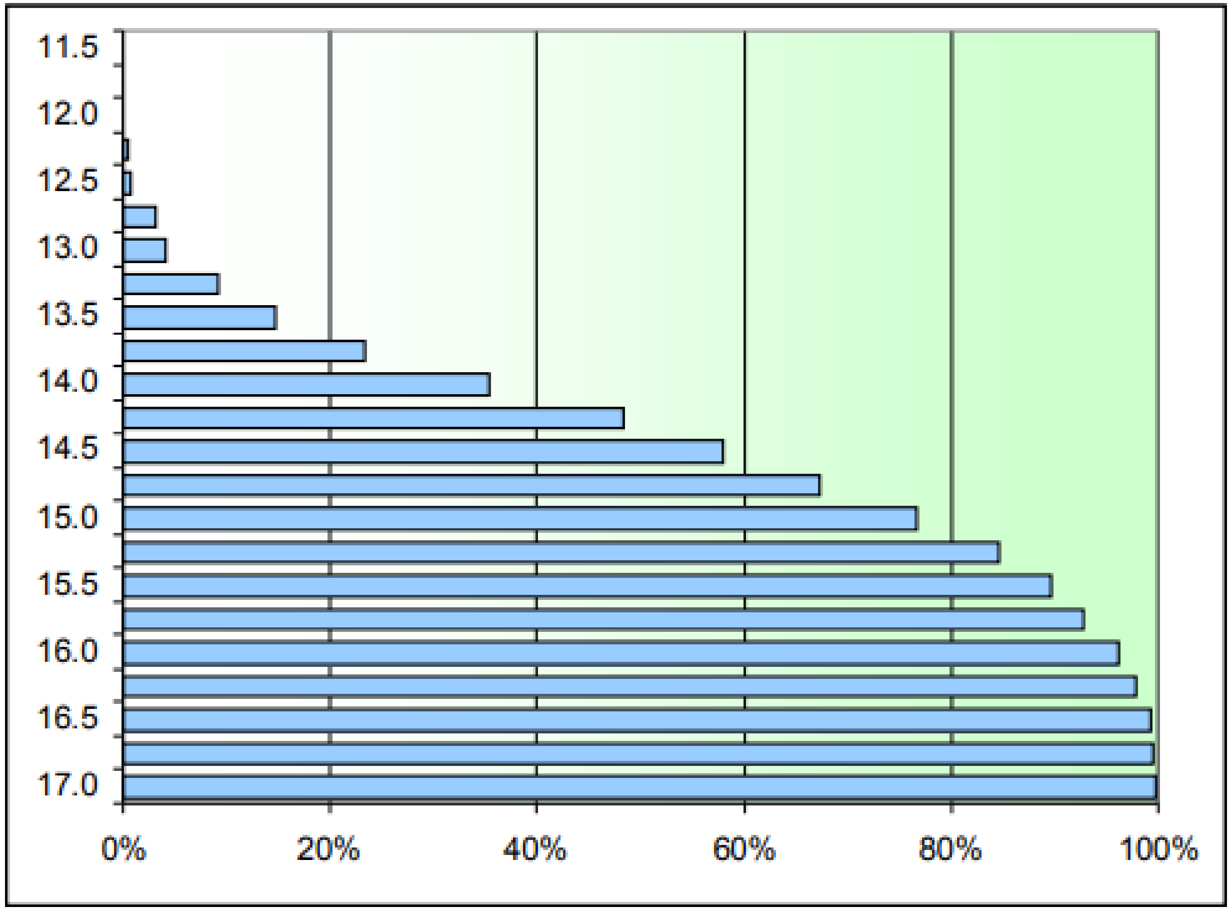


Figure 1: Probability of Completion Within Specified Time (Months)

PROJECT MANAGEMENT AND CONTROL

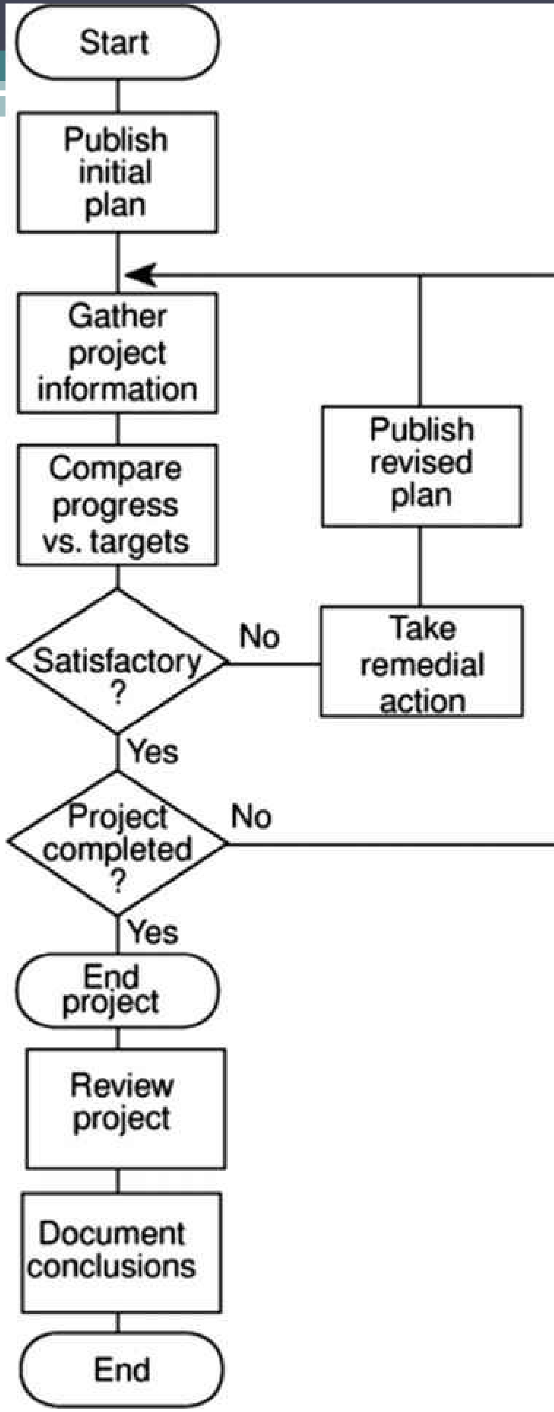
UNIT-4- part 1

Framework for Management and control

- Once work schedules have been published and the project is under way, attention must be focused on ensuring progress.
- This requires monitoring of what is happening, comparison of actual achievement against the schedule and, where necessary, revision of plans and schedules to bring the project as far as possible back on target.

Creating the framework

- Exercising control over a project and ensuring that targets are met is a matter of regular monitoring, finding out what is happening, and comparing it with current targets.
- If there is a mismatch between the planned outcomes and the actual ones then either replanning is needed to bring the project back on target or the target will have to be revised.



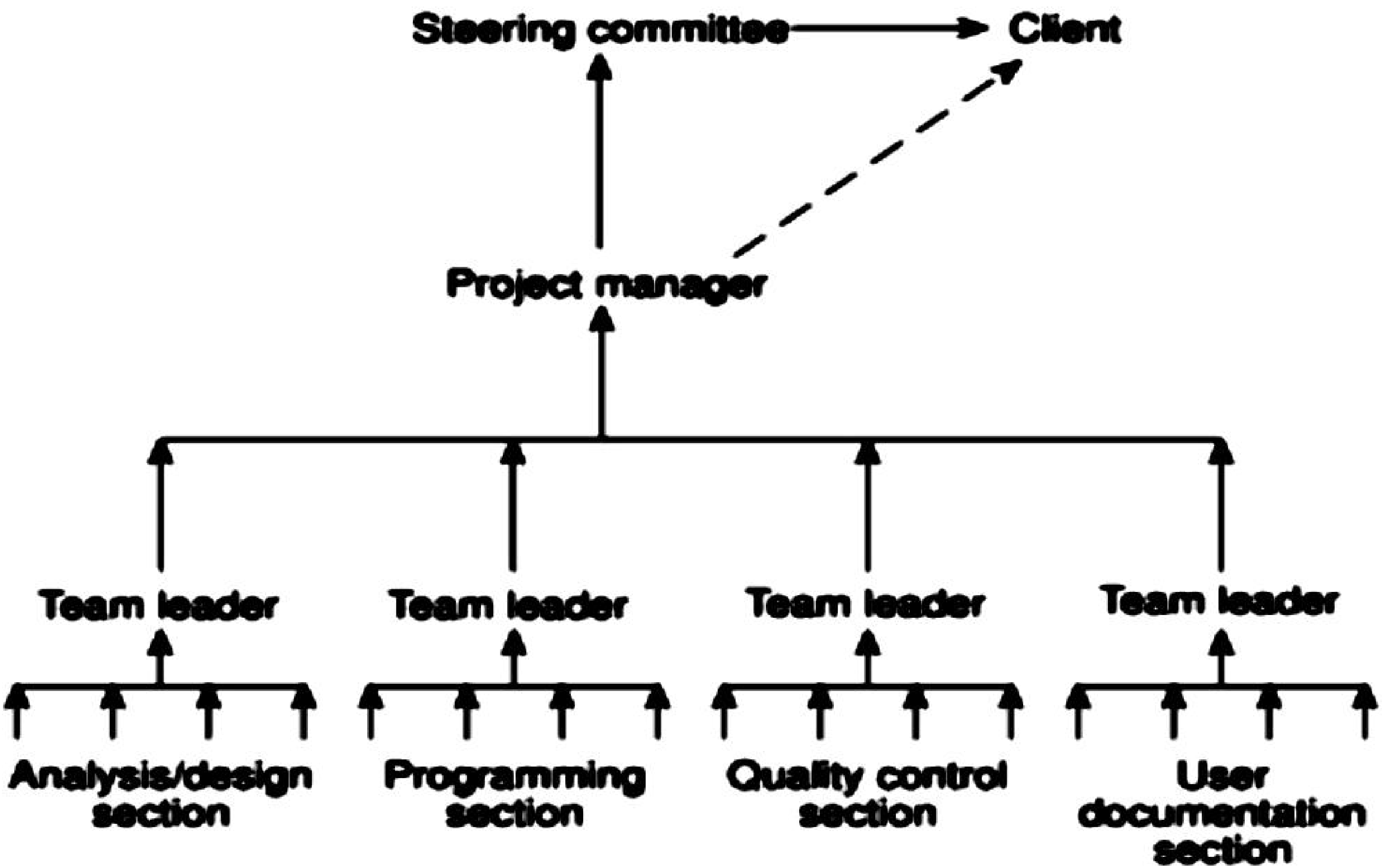
- The project control cycle shows how, once the initial project plan has been published, project control is a continual process of monitoring progress against that plan and, where necessary, revising the plan to take account of deviations.
- It also illustrates the important steps that must be taken after completion of the project so that the experienced gained in any one project can feed into the planning stages of future projects, thus allowing us to learn from past mistakes.
- In practice we are normally concerned with delays in meeting target dates, shortfalls in quality, inadequate quality. functionality, and costs going over target.

Responsibility

The overall responsibility for ensuring satisfactory progress on a project is often the role of the project **steering committee or Project Board**.

- Day-to-day responsibility will rest with the project manager and, in all but the smallest of projects, aspects of this can be delegated to team leaders.
- In most cases team leaders will collate reports on their section's progress and forward summaries to the project manager.
- These, in turn, will be incorporated into project-level reports for the steering committee and, via them or directly, progress reports for the client.

Project Reporting Structure



- Reporting may be oral or written, formal or informal, or regular or ad hoc.

Report type	*Examples*	*Comment*
Oral formal regular	weekly or monthly progress meetings	while reports may be oral formal written minutes should be kept
Oral formal ad hoc	end-of-stage review meetings	while largely oral, likely to receive and generate written reports
Written formal regular	job sheets, progress reports	normally weekly using forms
Written formal ad hoc	exception reports, change reports	
Oral informal ad hoc	canteen discussion, social interaction	often provides early warning; must be backed up by formal reporting

Assessing progress-

- Progress assessment will normally be made on the basis of information collected and collated at regular intervals or when specific events occur.
- Wherever possible, this information will be objective and tangible - for example: whether or not a particular report has been delivered.
- However, such end-of-activity deliverables might not occur sufficiently frequently throughout the life of the project.
- Here progress assessment will have to rely on the judgement of the team members who are carrying out the project activities.

Setting checkpoints-

- It is essential to set a series of checkpoints in the initial activity plan. Checkpoints may be:
  - regular
  - tied to specific events such as the production of a report or other deliverable.

Taking snap-shots-

- The frequency with which the a manager needs to receive information about progress will depend upon the size and degree of risk of the project or that part of the project under their control.
- Team leaders, for example, need to assess progress daily (particularly when employing inexperienced staff) whereas project managers may find weekly or monthly reporting appropriate.
- In general, the higher the level, the less frequent and less detailed the reporting needs to be.

Collecting the data

- As a rule, managers will try to break down long activities into more controllable tasks of one or two weeks duration.
- However, it will still be necessary to gather information about partially completed activities and, in particular, forecasts of how much work is left to be completed.
- It can be difficult to make such forecasts accurately.

Partial completion reporting-

- Weekly time sheets, for example, are frequently adapted by breaking jobs down to activity level and requiring information about work done in addition to time spent.
- Weekly timesheets are a valuable source of information about resources used. They are often used to provide information about what has been achieved.

Time Sheet

Staff: Paul **Week ending**: 14/05/99

Rechargeable hours

Project	Act. code	Desc.	Hours	% done	Sch. date	Est. date
P20	A267	Code mod A7	24	90	01/06/99	20/05/99
P35	B397	Testing mod B8	12	30	24/06/99	24/06/99
		Total	36			

Non-rechargeable hours

Code	Desc.	Hours	Comments and Authorization
L90	hours in Lieu	4	Authorized by Peter
	Total	4	

Risk reporting-

One popular way of overcoming the objections to partial completion reporting is to avoid asking for estimated completion dates, but to ask instead for the team members' estimates of the likelihood of meeting the planned target date.

One way of doing this is the traffic-light method. This consists of the following steps:

- identify the key (first level) elements for assessment in a piece of work;
- break these key elements into constituent elements (second level);

  assess each of the second level elements on the scale green for 'on target', amber for 'not on target but recoverable', and red for 'not on target and recoverable only with difficulty';
- review all the second level assessments to arrive at first level assessments;
- review first and second level assessments to produce an overall assessment.

cont… example

Activity Assessment Sheet

Staff : <u>Zobel</u>

Ref:IoE/P/100 **Activity: Code and test module A**

Week number	13	14	15	16	
Activity summary	G	A	R		
Component					**Comments**
Screen handling procedures	G	G	A		
File updating	G	A	R		
Compilation	G	G	A		
Run test data	G	A	A		

Monitoring progress

- monitoring time of each activity &
- monitoring cost in parallel

- To monitor time we needs tool to visualize the progress project from collected data and present them in way that is easy to understand
- Further helps to indentify the problem activities and areas to be taken care of.

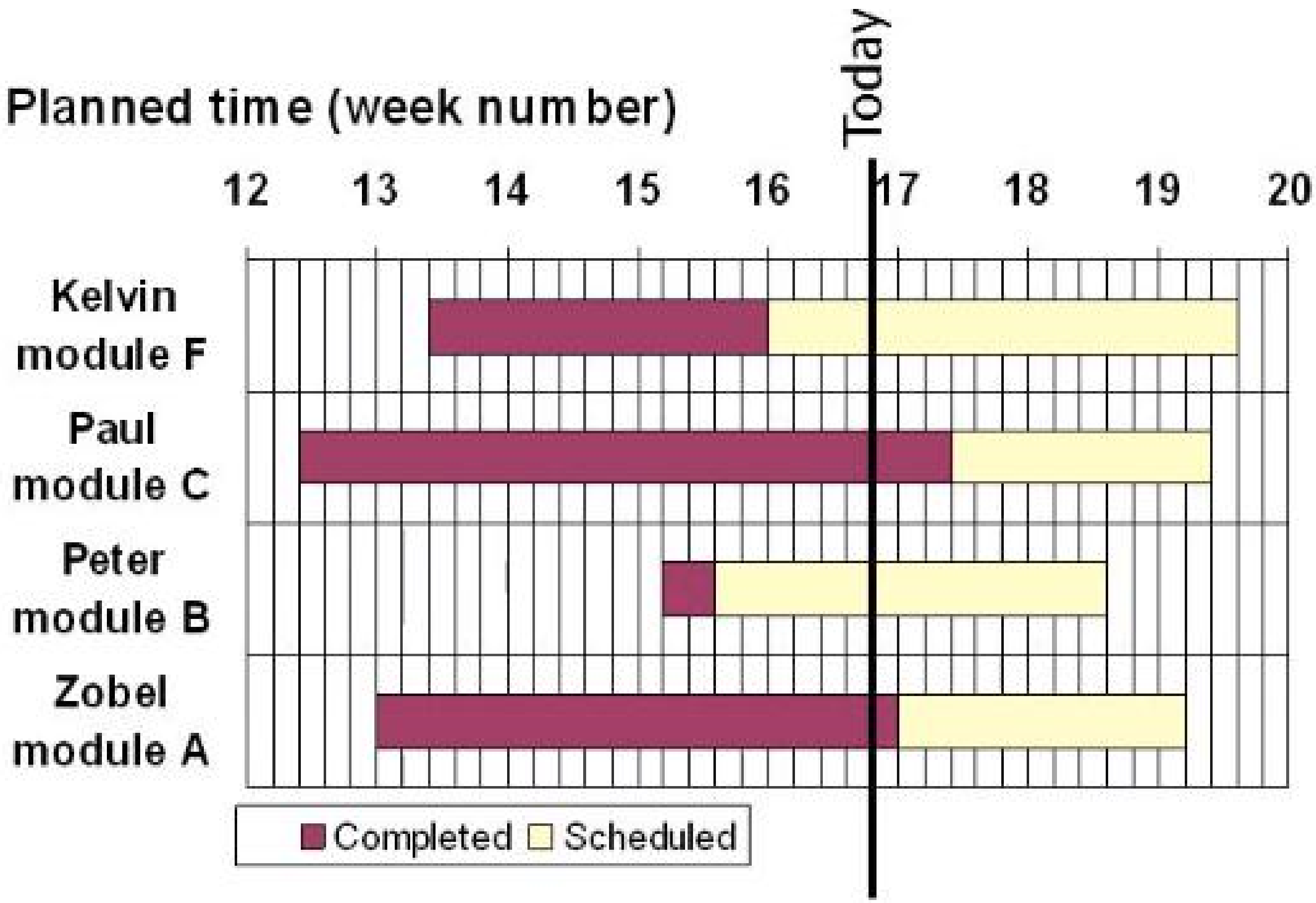
Visualizing progress

- Having collected data about project progress, a manager needs some way of presenting that data to greatest effect.

The Gantt chart

- One of the simplest and oldest techniques for tracking project progress is the Gantt chart.
- This is essentially an activity bar chart indicating scheduled activity dates and durations frequently augmented with activity floats.
- Reported progress is recorded on the chart (normally by shading activity bars) and a 'today cursor' provides an immediate visual indication of which activities are ahead or behind schedule.

Gantt chart

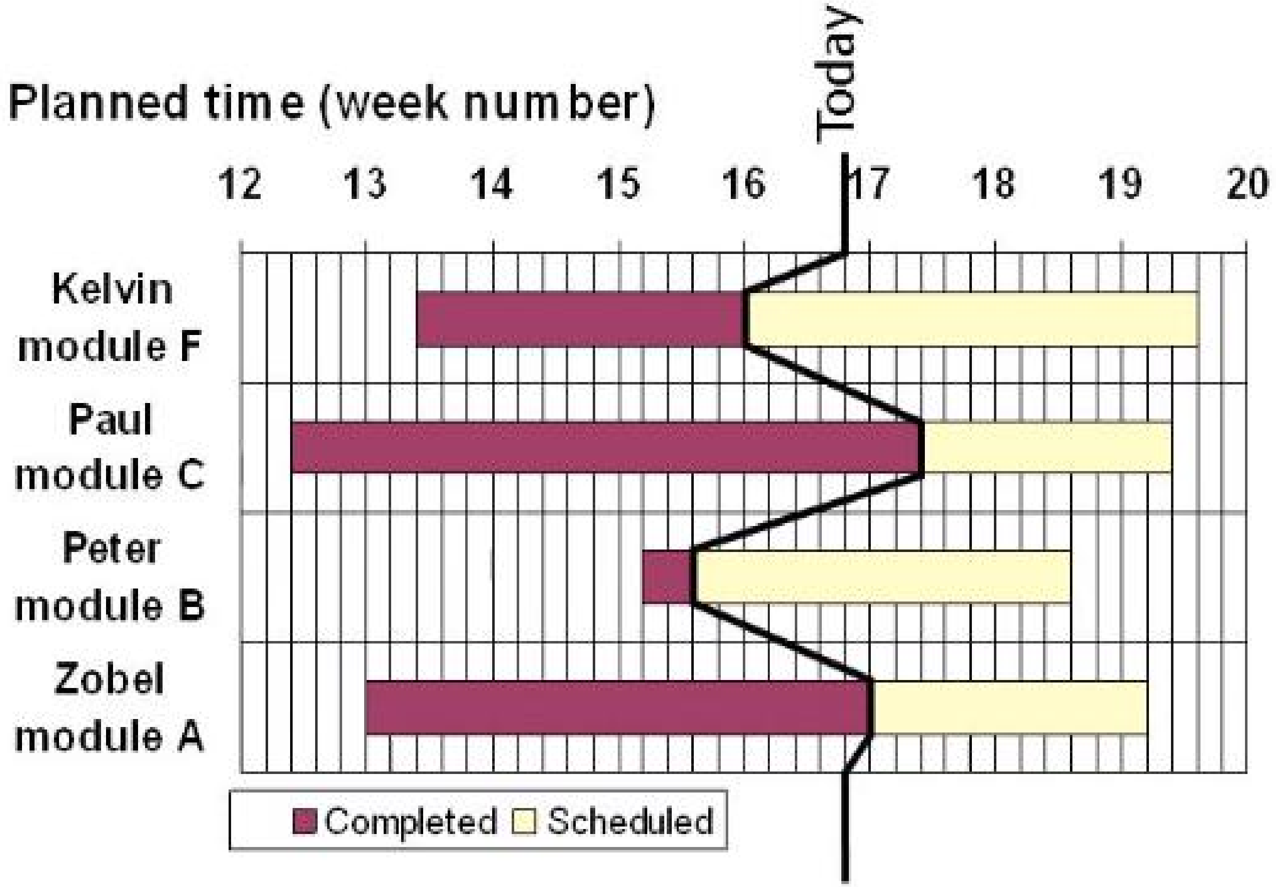


Planned time (week number)
Today
12
13
14
15
16
17
18
19
20
Kelvin module F
Paul module C
Peter module B
Zobel module A
Completed
Scheduled

The slip chart

- A slip chart is a very similar alternative favoured by some project managers who believe it provides a more striking visual indication of those activities that are not progressing to schedule - the more the slip line bends, the greater the variation from the plan.
- Additional slip lines are added at intervals and, as they build up, the project manager will gain an idea as to whether the project is improving (subsequent slip lines bend less) or not.
- A very jagged slip line indicates a need for rescheduling.

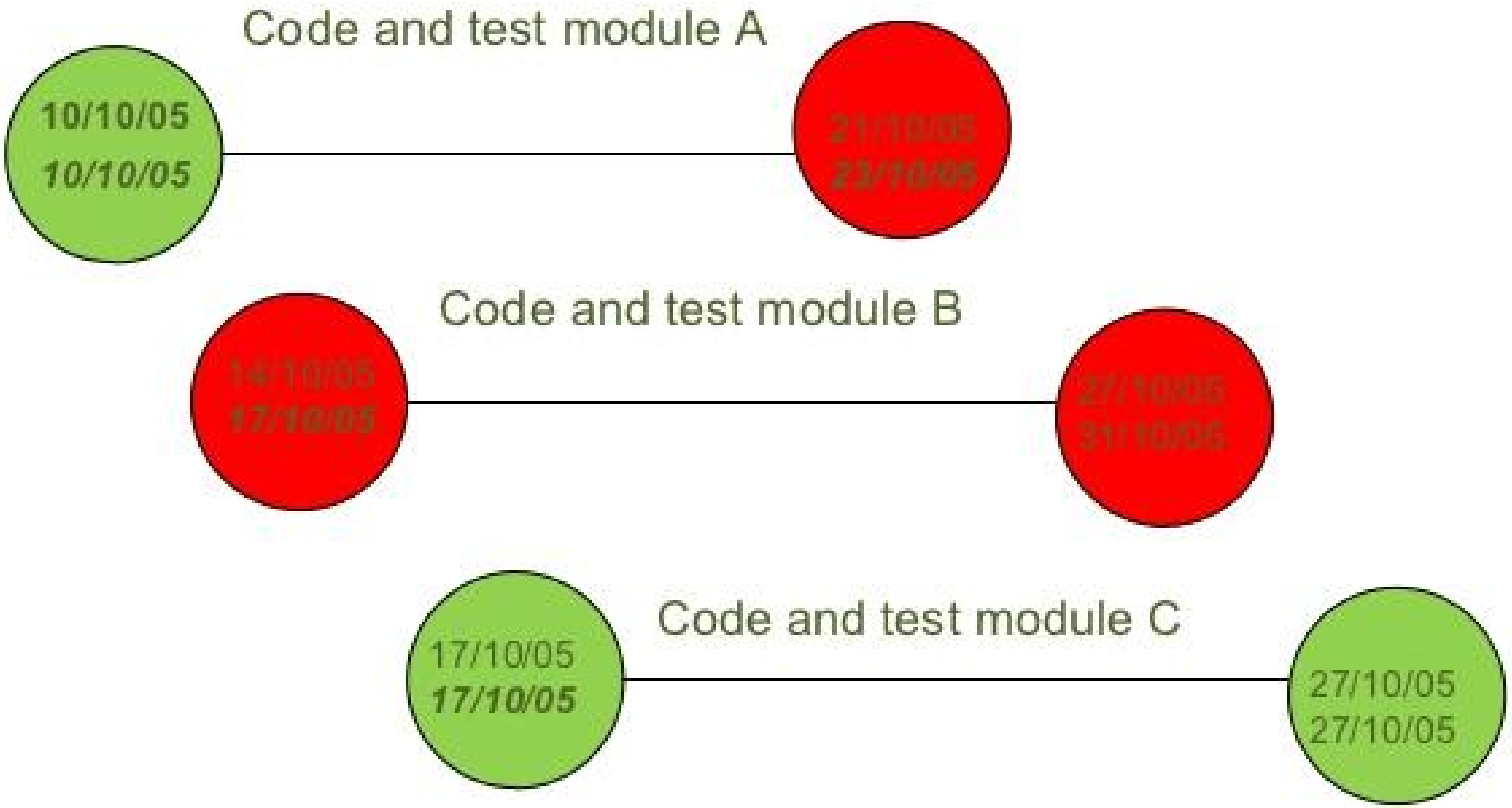
Slip chart


Planned time (week number)
Today
12
13
14
15
16
17
18
19
20
Kelvin module F
Paul module C
Peter module B
Zobel module A
Completed
Scheduled

Ball charts

- A somewhat more striking way of showing whether or not targets have been met is to use a ball chart.
- In this version of the ball chart, the circles indicate start and completion points for activities.
- The circles initially contain the original scheduled dates.
- Whenever revisions are produced these are added as second dates in the appropriate circle until an activity is actually started or completed when the relevant date replaces the revised estimate.
- Circles will therefore contain only two dates, the original and most recent target dates, or the original and actual dates.

Ball charts



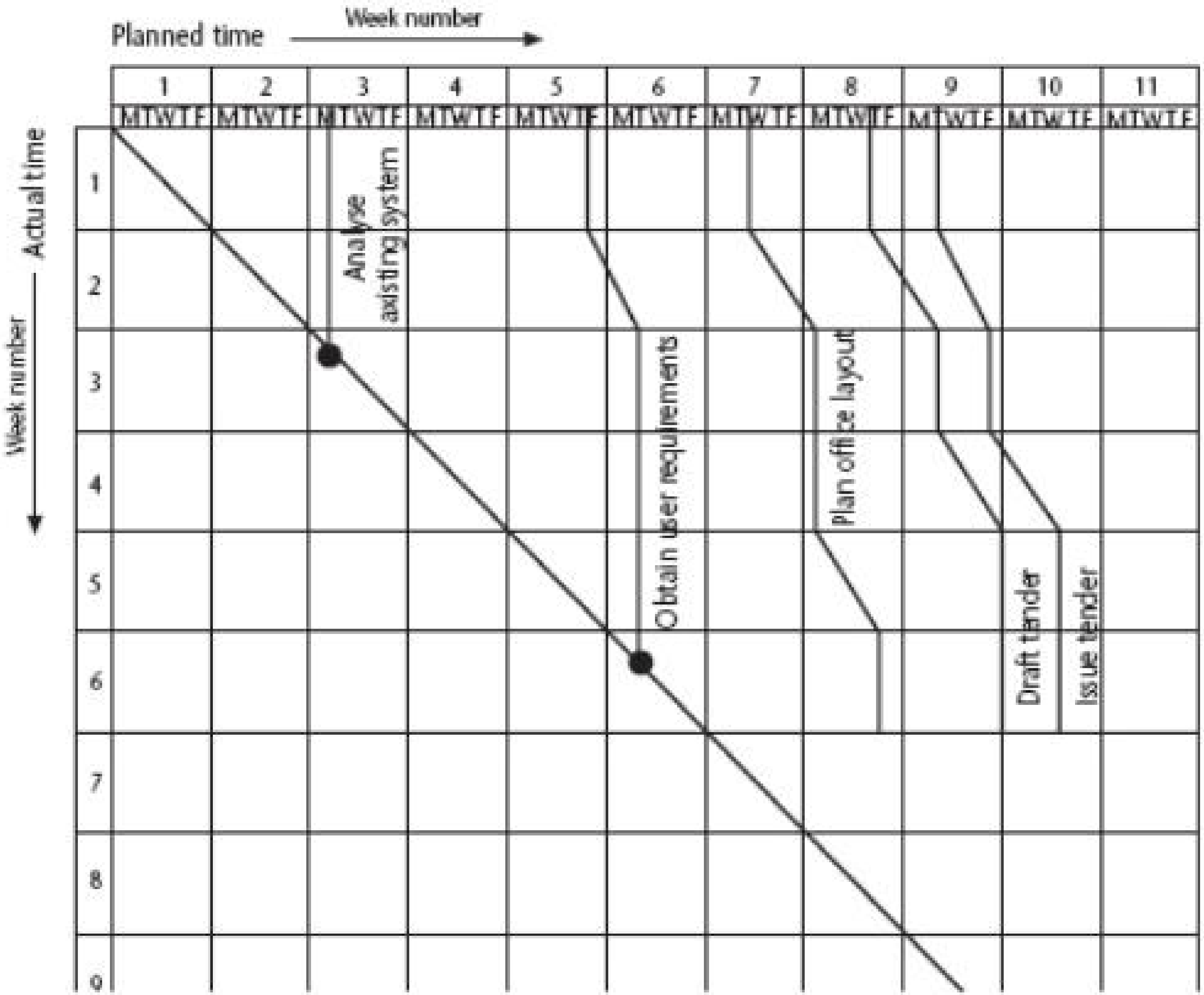
Green: On time
Red: Missed the target

Timeline

- One disadvantage of the charts described so far is that they do not show clearly the slippage of the project completion date through the life of the project.
- Knowing the current state of a project helps in revising plans to bring it back on target, but analysing and understanding trends helps to avoid slippage in future projects.
- The timeline chart is a method of recording and displaying the way in which targets have changed throughout the duration of the project.
- The timeline chart is useful both during the execution of a project and as part of the post-implementation review. Analysis of the timeline chart, and the reasons for the changes, can indicate failures in the estimation process or other errors that might, with that knowledge, be avoided in future.

The timeline

Records the way targets have changd throughout the project



Change control

- Requirements are modified because of changing circumstances or because the users get a clearer idea of what is really needed.
- The payroll system that is implementing might, for instance, need to be adjusted if the staffing structure at the college is reorganized.
- Careful control of these changes is needed because an alteration in one document often implies changes to other documents and the system products based on that document.
- A change in a program specification will normally be carried through into changes to the program design and then changed code.

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.

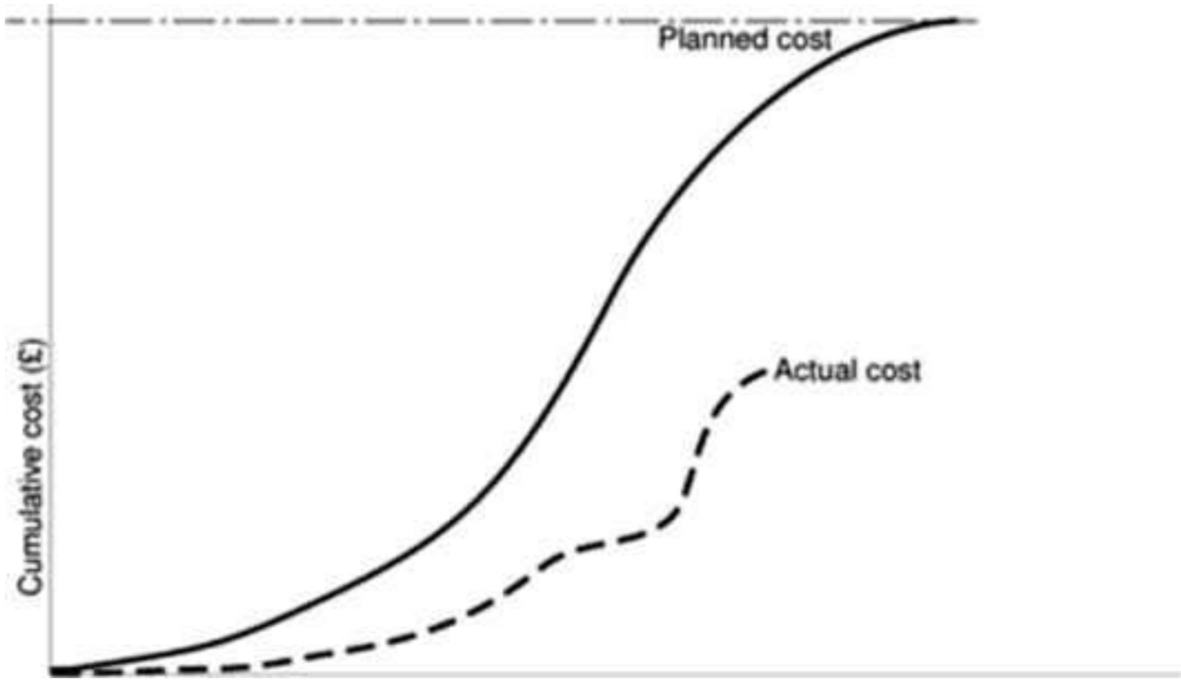
8. When the user is satisfied that the products are adequate they will authorize their operational release. The master copies of configuration items will be replaced.

Cost monitoring

Unit4-part-2

Cost monitoring

- Expenditure monitoring is an important component of project control. Not only in itself, but also because it provides an indication of the effort that has gone into (or at least been charged to) a project.
- A project might be on time but only because more money has been spent on activities than originally budgeted.
- A cumulative expenditure chart such as that shown in Figure provides a simple method of comparing actual and planned expenditure.



- By itself it is not particularly meaningful - Figure could, for example, illustrate a project that is running late or one that is on time but has shown substantial costs savings!
- Cost charts become much more useful if we add projected future costs calculated by adding the estimated costs of uncompleted work to the costs already incurred.
- Where a computer-based planning tool is used, revision of cost schedules is generally provided automatically once actual expenditure has been recorded.

Earned Value

- Earned Value Analysis, also known as Budgeted Cost of Work Performed.
- Earned Value Analysis is based on assigning a 'value' to each task or work package (WBS) based on the original expenditure forecasts.
- The assigned value is the original budgeted cost for the item and is known as the ***baseline budget or planned value or budgeted cost of work scheduled (BCWS).***
- A task that has not started is assigned the value zero and when it has been completed, it, and hence the project, is credited with the value of the task.
- The ***total value credited to a project at any point is known as the earned value or budgeted cost of work performed (BCWP)*** and this can be represented as a value or as a percentage of the BCWS.

Where tasks have been started but are not yet complete, some consistent method of assigning an earned value must be applied. Common methods in software projects are:

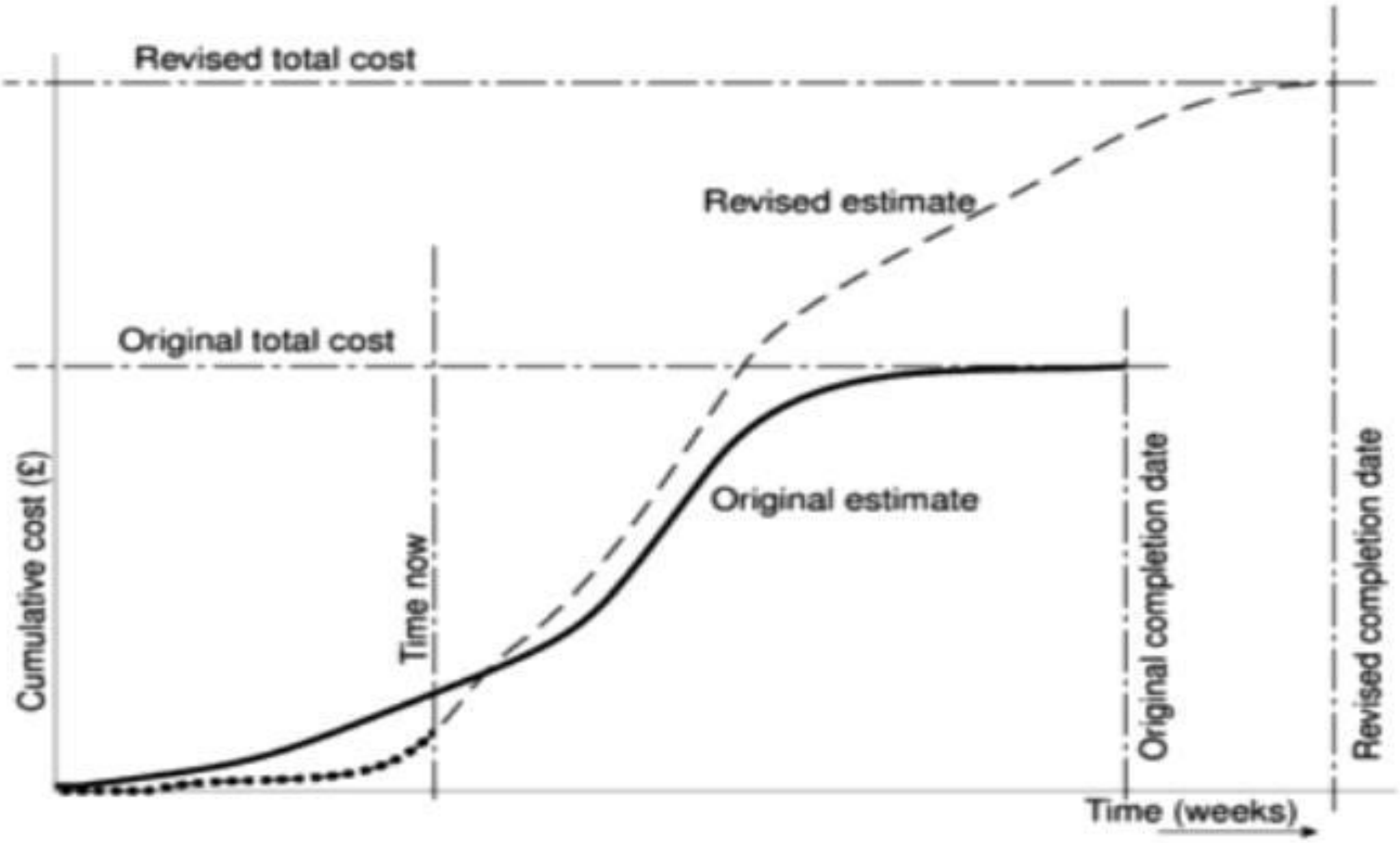
• **the 0/100 technique-** Where a task is assigned a value of zero until such time that it is completed when it is given a value of 100% of the budgeted value;

• **the 50/50 technique-** Where a task is assigned a value of 50% of its value as soon as it is started and then given a value of 100% once it is complete;

• **the milestone technique-** Where a task is given a value based on the achievement of milestones that have been assigned values as part of the original budget plan.

Of these, we prefer the 0/100 technique. The 50/50 technique can give a false sense of security by over-valuing the reporting of activity starts. The milestone technique might be appropriate for activities with a long duration estimate but, in such cases, it is better to break that activity into a number of smaller ones.

Cumulative cost expenditure



Baseline budget

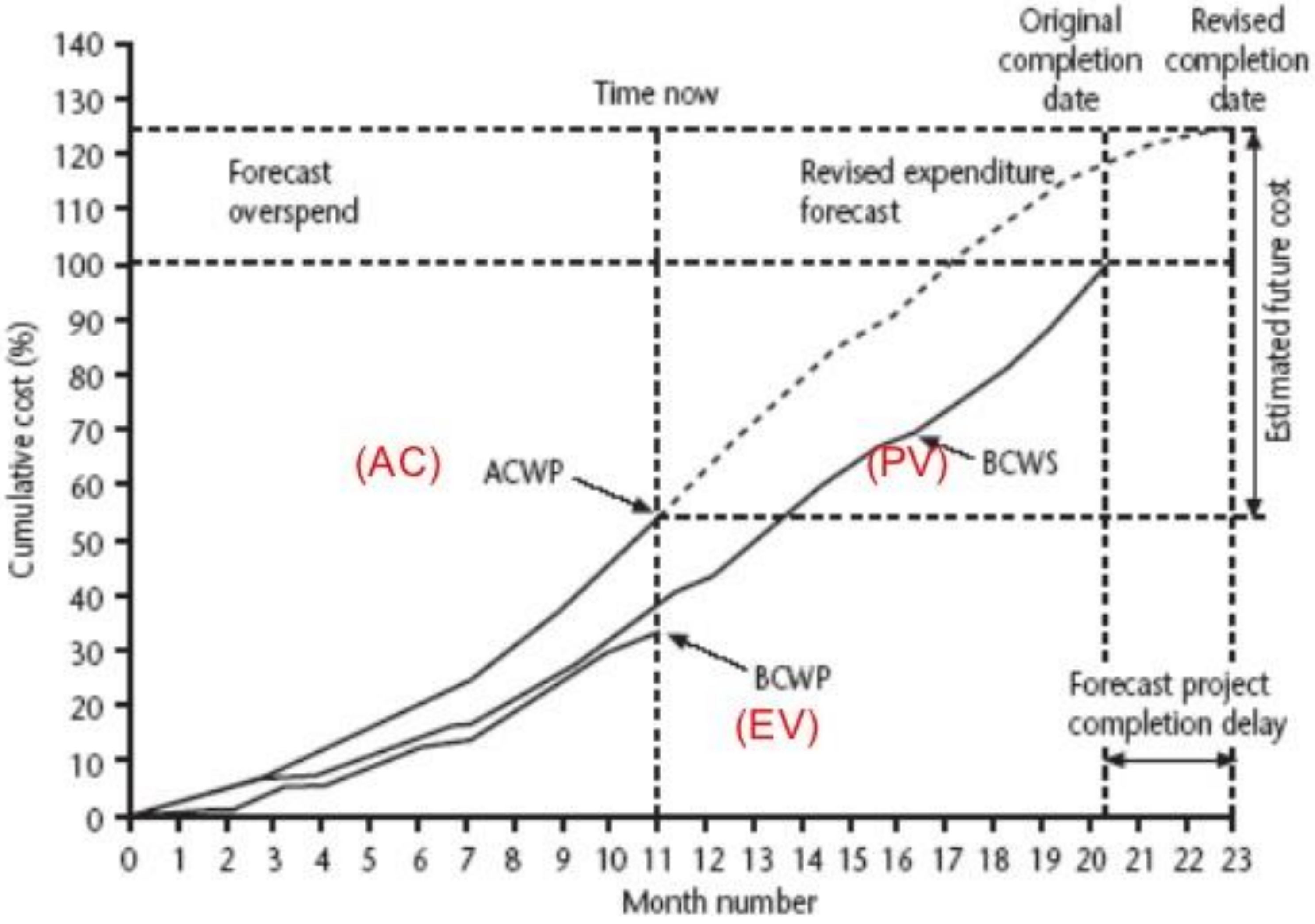
- The first stage in setting up an earned value analysis is to create the baseline budget.
- The baseline budget is based on the project plan and shows the forecast growth in earned value through time.
- Earned value may be measured in monetary values but, in the case of staff-intensive projects such as software development, it is common to measure earned value in person-hours or workdays.

Monitoring earned value

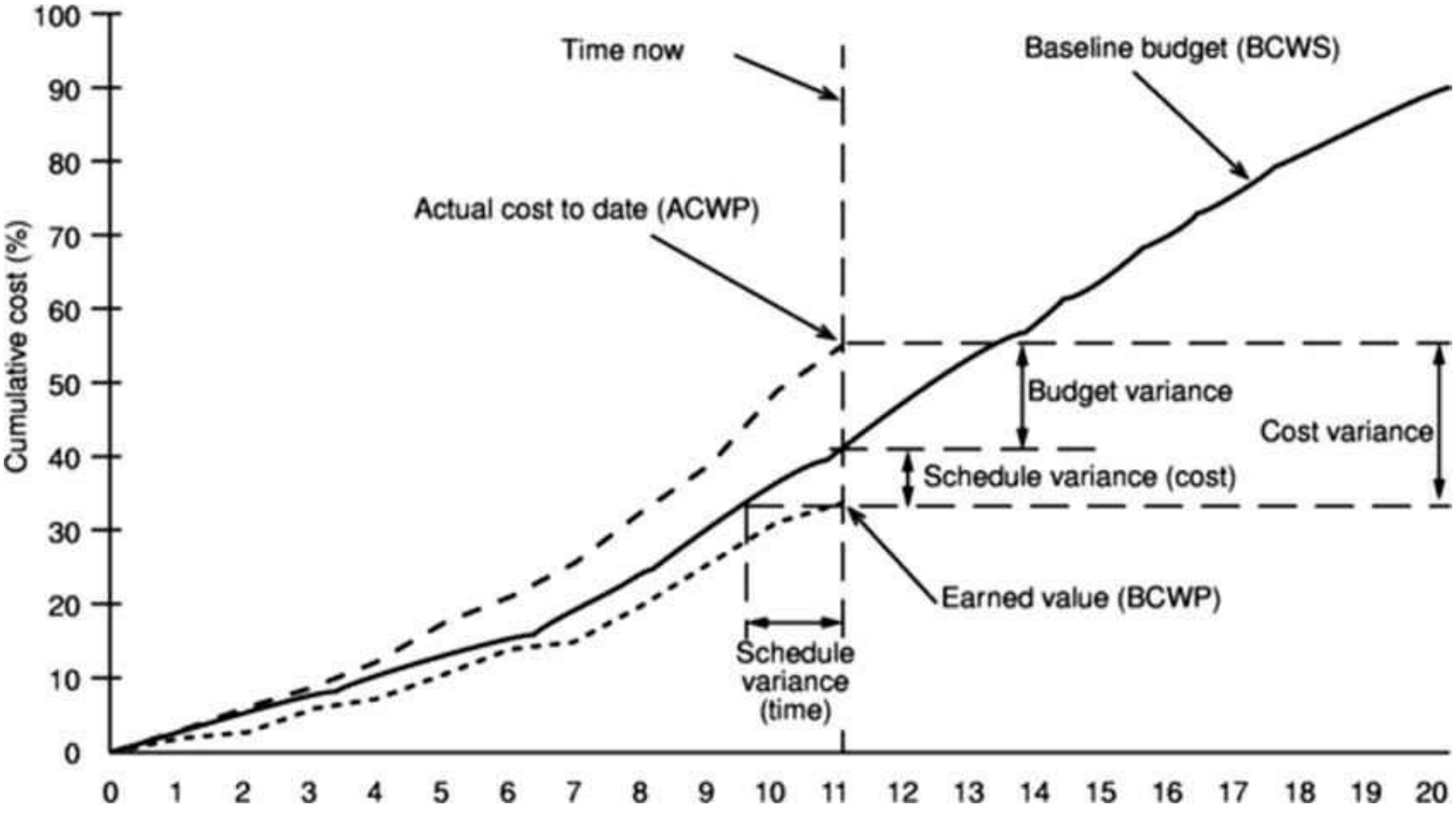
- Having created the baseline budget, the next task is to monitor earned value as the project progresses. This is done by monitoring the completion of tasks (or activity starts and milestone achievements in the case of the other crediting techniques). The actual cost of each task can be collected as actual cost of work performed, ACWP.
- **Budget variance= ACWP - BCWS** and indicates the degree to which actual costs differ from those planned.
- **Schedule variance= BCWP -BCWS** and indicates the degree to which the value of completed work differs from that planned.
- **Cost variance= BCWP - ACWP** and indicates the difference between the budgeted cost and the actual cost of completed work. It is also an indicator of the accuracy of the original cost estimates.
- **Performance ratios-**
  1. **cost performance index (CPI = BCWP/ACWP) and**
  2. **schedule performance index (SPI = BCWP/ BCWS)**

They can be thought of as a 'value-for-money* indices. A value greater than one indicates that work is being completed better than planned whereas a value of less than one means that work is costing more than and/or proceeding more slowly than planned.

Earned value chart with revised forecasts



Cont..



Time now
Baseline budget (BCWS)
Actual cost to date (ACWP)
Budget variance
Cost variance
Schedule variance (cost)
Earned value (BCWP)
Schedule variance (time)
Cumulative cost (%)
0
10
20
30
40
50
60
70
80
90
100
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Earned value – an example

- Tasks
  - Specify module 5 days
  - Code module 8 days
  - Test module 6 days
- At the beginning of day 20, PV = 19 days
- If everything but testing completed, EV = 13 days
- Schedule variance = EV-PV i.e. 13-19 = -6
- Schedule performance indicator (SPI) = EV/PV

  i.e 13/19 = 0.68

Earned value analysis – actual cost

- Actual cost (AC) is also known as Actual cost of work performed (ACWP)
- In previous example, if
  - 'Specify module' actually took 3 days (planned 5 days)
  - 'Code module' actually took 4 days (planned 8 days)
- Actual cost = 7 days
- Cost variance (CV) = EV-AC

  i.e. 13-7 = 6 days
- Cost performance indicator (CPI) = EV/AC

  i.e = 13/7 = 1.86
- Positive CV or CPI > 1.00 means project under budget or the work is completed better than planned

Prioritizing monitoring

We might focus more on monitoring certain types of activity e.g.

- **Critical path activities-** Any delay in an activity on the critical path will cause a delay in the completion date for the project. Critical path activities are therefore likely to have a very high priority for close monitoring.
- **Activities with no free float** –Free float is the amount of time an activity may be delayed without affecting any subsequent activity. If delayed later dependent activities are delayed
- **Activities with less than a specified float- I**f any activity has very little float it might use up this float before the regular activity monitoring brings the problem to the project manager's attention. It is common practice to monitor closely those activities with less than, say, one week free float.

Cont..

- **High risk activities-** A set of high risk activities should have been identified as part of the initial risk profiling exercise. If we are using the PERT three-estimate approach we will designate as high risk those activities that have a high estimated duration variance. These activities will be given close attention because they are most likely to overrun or overspend.
- **Activities using critical resources-** Activities can be critical because they are very expensive (as in the case of specialized contract programmers). Staff or other resources might be available only for a limited period, especially if they are controlled outside the project team. In any event, an activity that demands a critical resource requires a high level of monitoring.

Getting the project back to target

- Almost any project will, at one time or another, be subject to delays and unexpected events. One of the tasks of the project manager is to recognize when this is happening (or, if possible, about to happen) and, with the minimum delay and disruption to the project team, attempt to mitigate the effects of the problem.
- In most cases, the project manager tries to ensure that the scheduled project end date remains unaffected. This can be done by shortening remaining activity durations or shortening the overall duration of the remaining project.
- This might not always be the most appropriate response to disruptions to a plan. There is little point in spending considerable sums in overtime payments in order to speed up a project if the customer is not overly concerned with the delivery date and there is no other valuable work for the team members once this project is completed.
- There are two main strategies to consider when drawing up plans to bring a project back on target –

1. shortening the critical path
2. altering the activity precedence requirements.

1- **Shorten the critical path**

- The overall duration of a project is determined by the current critical path, so speeding up non-critical path activities will not bring forward a project completion date.
- Extolling staff to 'work harder' might have some effect, although frequently a more positive form of action is required, such as increasing the resources available for some critical activity. Fact-finding, for example, might be speeded up by allocating an additional analyst to interviewing users. It is unlikely, however, that the coding of a small module would be shortened by allocating an additional programmer - indeed, it might be counterproductive because of the additional time needed organizing and allocating tasks and communicating.
- Resource levels can be increased by making them available for longer. Thus, staff might be asked to work overtime for the duration of an activity and computing resources might be made available at times when they might otherwise be inaccessible.
- Where these do not provide a sufficient solution, the project manager might consider allocating more efficient resources to activities on the critical path or swapping resources between critical and non-critical activities.

2-**Reconsider the precedence requirements**

- If attempting to shorten critical activities proves insufficient, the next step is to consider the constraints by which some activities have to be deferred pending completion of others.
- Example- The original project network would most probably have been drawn up assuming 'ideal' conditions and 'normal' working practices. It might be that, to avoid the project delivering late, it is now worth questioning whether as yet unstarted activities really do have to await the completion of others. It might, in a particular organization, be 'normal' to complete system testing before commencing user training. In order to avoid late completion of a project it might, however, be considered acceptable to alter 'normal' practice and start training earlier.
- One way to overcome precedence constraints is to subdivide an activity into a component that can start immediately and one that is still constrained as before.

Managing Contracts

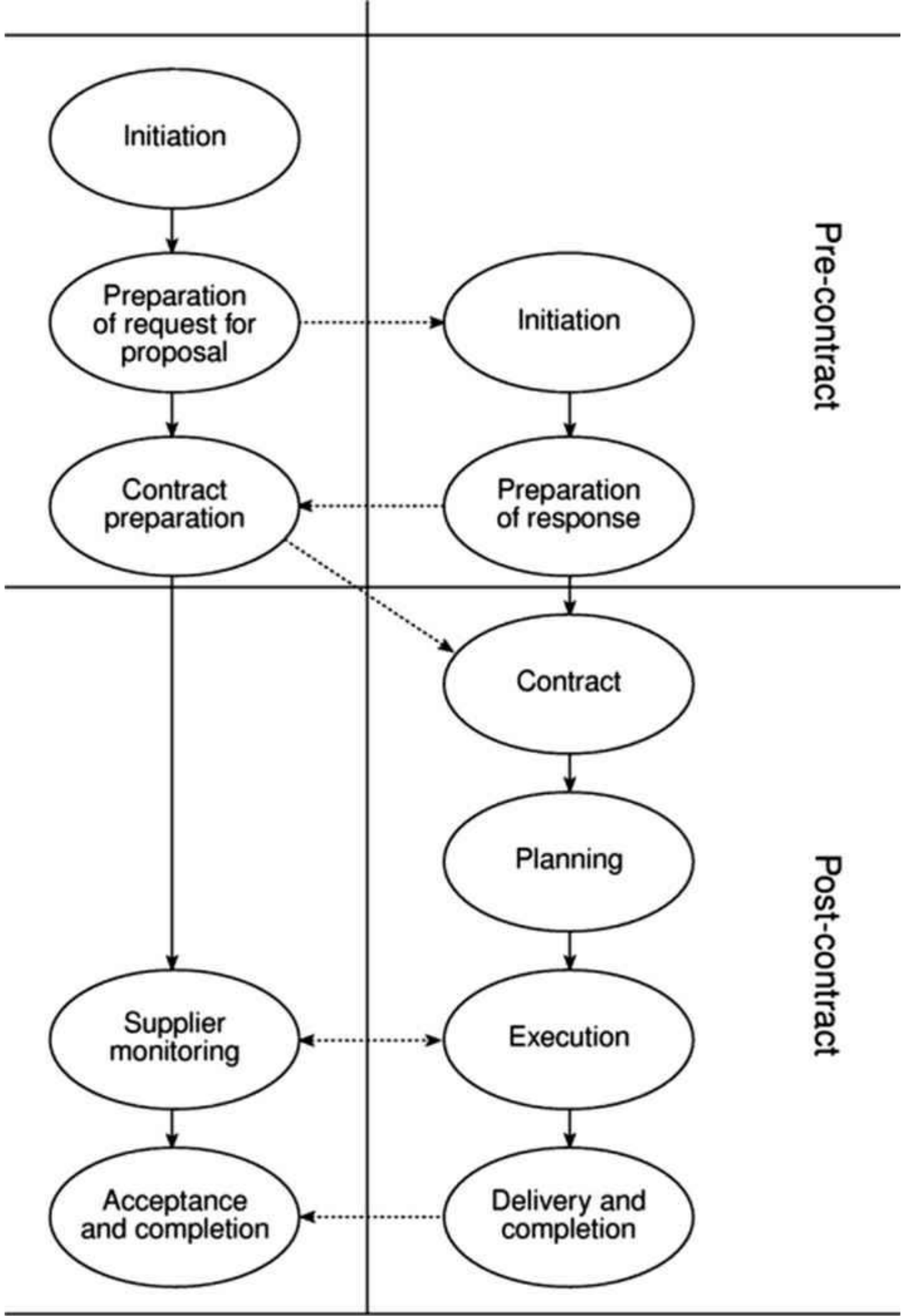
Module4-part-3

Acquiring software from external supplier

This could be:

- a bespoke system - created specially for the customer
- off-the-shelf - bought ‘as is’
- customized off-the-shelf (COTS) - a core system is customized to meet needs of a particular customer

- ISO 12207 acquisition and supply process



Types of contract

- Fixed price contracts
- Time and materials contracts
- Fixed price per delivered unit

Fixed Price Contracts

- With fixed price contracts, also known as lump sum contracts, the buyer and service provider agree on a fixed price for the services in question.
- This type of contract is low-risk for the buyer, but high-risk for the seller since the time and costs of the project could exceed the fixed price.
- For this reason, a fixed price contract should include a detailed scope of work that clearly outlines what the buyer can expect for the agreed-upon price.
- When the contract is signed, the seller must complete the task or deliver the goods as agreed or risk being in breach of contract.

Advantages to customer:

- known expenditure
- supplier motivated to be cost-effective

Disadvantages:

- supplier will increase price to meet contingencies
- difficult to modify requirements
- upward pressure on the cost of changes
- threat to system quality

Time and materials

- This contract is used when labor is the main deliverable and typically provides the seller an hourly rate.
- This is one of the most beautiful engagements that can get into by two or more parties. This engagement type is the most risk-free type where the time and material used for the project are priced.
- The contractor only requires knowing the time and material for the project in order to make the payments. This type of contract has short delivery cycles, and for each cycle, separate estimates are sent of the contractor.
- Once the contractor signs off the estimate and Statement of Work (SOW), the service provider can start work.
- Unlike most of the other contract types, retainer contracts are mostly used for long-term business engagements.

Advantages to customer:

- easy to change requirements
- lack of price pressure can assist product quality

Disadvantages:

- Customer liability - the customer absorbs all the risk associated with poorly defined or changing requirements
- Lack of incentive for supplier to be cost-effective

Fixed price per unit delivered contracts

FP count	*Design cost/FP*	*implement-ation cost/FP*	*total cost/FP*
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/unit example

- Estimated system size 2,600 FPs
- Price
  - 2000 FPs x \$967 *plus*
  - 500 FPs x \$1,019 *plus*
  - 100 FPs x \$1,058
  - i.e. \$2,549,300
- What would be charge for 3,200 FPs?

Advantages for customer

- customer understanding of how price is calculated
- comparability between different pricing schedules
- emerging functionality can be accounted for
- supplier incentive to be cost-effective

Disadvantages

- difficulties with software size measurement - may need independent FP counter
- changing (as opposed to new) requirements: how do you charge?

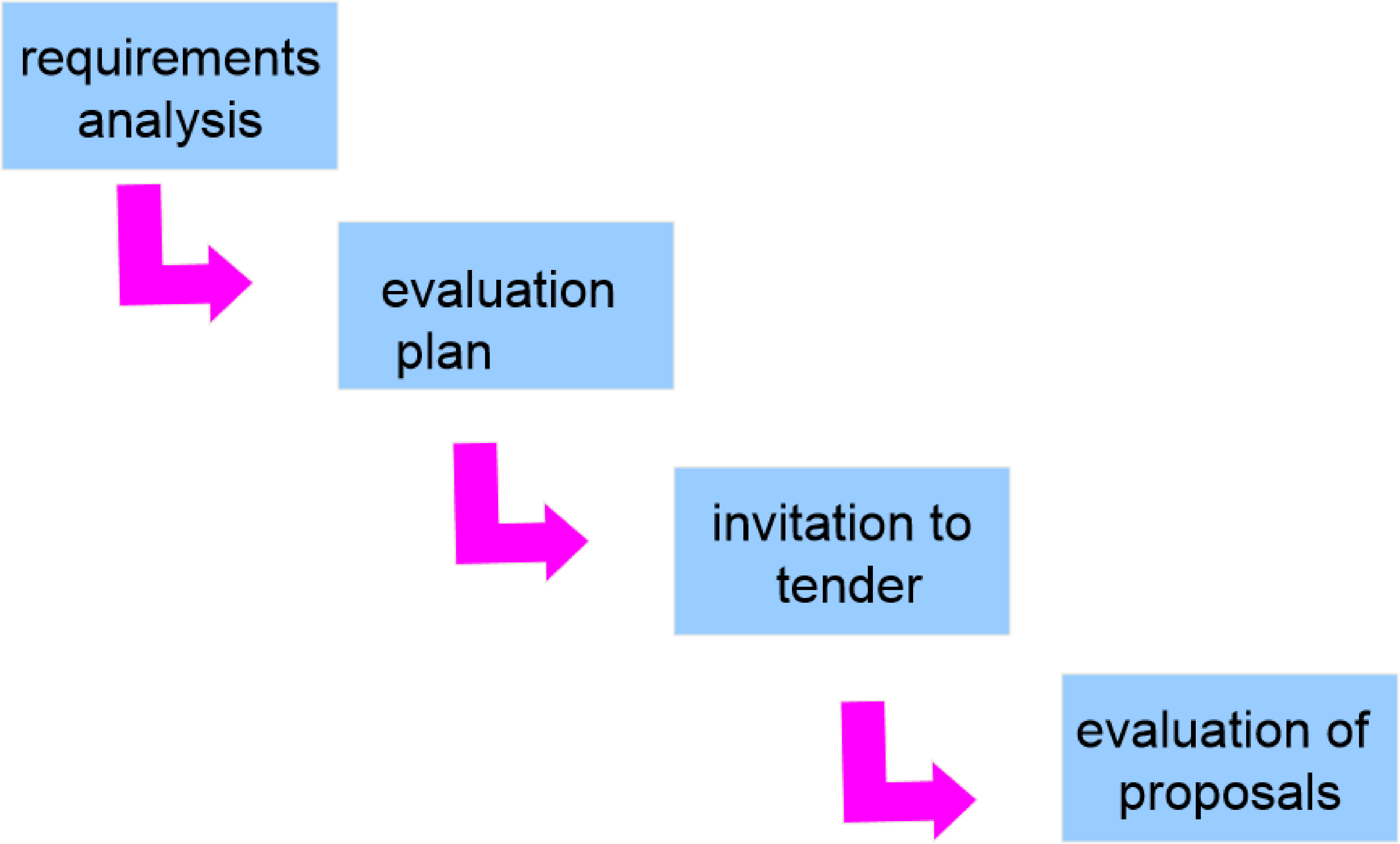
The tendering process

- Open tendering
  - any supplier can bid in response to the *invitation to tender*
  - all tenders must be evaluated in the same way
  - government bodies may have to do this by local/international law

The tendering process

- Restricted tendering process
  - bids only from those specifically invited
  - can reduce suppliers being considered at any stage
- Negotiated procedure
  - negotiate with one supplier e.g. for extensions to software already supplied

Stages in contract placement



Stages in contract placement

- **Requirements analysis**
- Before potential supplier can be approached, you need to have a clear set of requirements.
- Two points need to be emphasized here.
- The first is that it is easy for this step to be skimped where the user has many day-to-day pressures and not much time to think about future developments. In this situation, it can be useful to bring in an external consultant to draw up a requirements document. Even here, users and their managers need to look carefully at the resulting requirements document to ensure that it accurately reflect their needs.

- Main sections in a requirements document

  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 features

  5 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

  6 Additional information required from potential suppliers

- The requirements define carefully the functions that need to be carried out by the new application and all the necessary inputs and outputs for these functions. The requirements should also state any standards with which there should be compliance, and the existing systems with which the new system needs to be compatible.

- **Invitation to tender (ITT)**
- Having produced the requirements and the evaluation plan, it is now possible to issue the invitation to tender to prospective suppliers.
- Essentially, this will be the requirement document with a supporting letter, which may have additional information about how responses to the invitation are to be lodged.
- A deadline will be specified and it is hoped that by then a number of proposals with price quotations will have been received.
- In English law, for a contract to exist there must be an offer on one side that must be accepted by the other side. The invitation to tender is not an offer itself but an invitation for prospective suppliers to make an offer.
- 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

Evaluation of proposals

- This needs to be done in a methodical and planned manner. An evaluation plan, which will describe how each proposal will be checked to see if it meets each requirement. This reduces risks of requirements being missed and ensures that all proposals are treated consistently.
- Otherwise, there is a risk that a proposal might be unfairly favoured because of the presence of a feature that was not requested in the original requirement.
- It will be recalled that an application could be either bespoke, off-the-shelf, or customized. In the case of off-the-shelf packages, it would be the software itself that would be evaluated and it might be possible to combine some of the evaluation with acceptance testing.

Cont...

- How are proposals to be evaluated?
- Methods could include:
  - reading proposals
  - interviews
  - demonstrations
  - site visits
  - practical tests
- Need to assess value for money for each desirable feature
- Example:
    - feeder file saves data input
    - 4 hours a month saved
    - cost of data entry at RM20 an hour
    - system to be used for 4 years
    - if cost of feature RM1000, would it be worth it?

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

How would you evaluate the following?

- **Usability of existing package**
  Could try out a demo or ask existing users
- **Usability of application to be built**
  You would have to make stipulation about the process e.g. on the development of interface prototypes; you could also specify performance requirements
- **Maintenance costs of hardware**
  this could be incorporated in a maintenance agreement and you could compare the terms offered by different potential suppliers; another approach is ask to current users of the hardware about their experience of it.

How would you evaluate the following? con't

- **Time taken to respond to support requests**
  this could once again be made a contractual matter and the terms offered by different suppliers could be compared; suppliers could be asked to supply evidence of their past performance (but they might refuse, or not tell the truth); you could ask for references from current customers of the supplier;
- **Training**
  once again references could be taken up; you could ask for the CV of the trainer; you could even get them to give a short presentation

Typical terms of a contract

The terminology used in the contract document may need to be defined, for example, who is meant by the words 'client' and 'supplier'.

Form of agreement

- For example, is it a contact of sale, a lease, or a licence? Also, can the subject of the contract, such as a licence to use a software application, be transferred to another party?

Goods and services to be supplied

- Equipment and software to be supplied This includes an actual list of the individual pieces of equipment to be delivered, complete with the specific model numbers.

Services to be provided- This covers such things as:

- documentation;
- installation;
- conversion of existing files;
- maintenance agreements;
- transitional insurance arrangements.

Cont…

- **Ownership of the software**
- Who has ownership of the software? There are two key issues here: firstly, whether the customer can sell the software to others and, secondly, whether the supplier can sell the software to others.
- Where off-the-shelf software is concerned, the supplier often simply grants a license for you to use the software.
- Where the software is being written specially for a customer, then that customer will normally wish to ensure exclusive use of the software - they may object to software which they hoped would give them a competitive edge being sold to their rivals.

Cont…

Environment-

- Where physical equipment is to be installed, the demarcation line between the supplier's and customer's responsibilities with regard to such matters as accommodation and electrical supply needs to be specified. Where software is being supplied, the compatibility of the software with the existing hardware and operating system platforms would need to be confirmed.

Customer commitments

- Even when work is carried out by external contractors, a development project still needs the participation of the customer. The customer will have to provide accommodation for the suppliers and perhaps other facilities such as telephone lines.

Acceptance procedures

- Good practice would be to accept a delivered system only after it has undergone user acceptance tests.

Cont…

Standards

- This covers the standards with which the goods and services should comply. For example, a customer can require the supplier to conform to the ISO 12207 standard relating to the software life cycle and its documentation.

Project and quality management

- The arrangements for the management of the project must be agreed. Among these would be frequency and nature of progress meetings and the progress information to be supplied to the customer. The contract could require that appropriate ISO 9000-series standards be followed.

Timetable

- This provides a schedule of when the key parts of the project should be completed. This timetable will commit both the supplier and the customer.

Cont…

- **Price and payment method**
- Obviously the price is very important! What also needs to be agreed is when the payments are to be made. The supplier's desire to be able to meet costs as they are incurred needs to be balanced by the customer's requirement to ensure that goods and services are satisfactory before parting with their money.
- **Miscellaneous legal requirements**
- This is the legal small print. Contracts often have clauses that deal with such matters the legal jurisdiction that will apply to the contract, what conditions would apply to the sub-contracting of the work, liability for damage to third parties, and liquidated damages. Liquidated damages are estimates of the financial losses that the customer would suffer if the supplier were to fall short of their obligations.

Contract management

- We need to consider the communications between the supplier and the customer while the work contracted for is being carried out. It would probably suit all concerned if the contractor could be left to get on with the work undisturbed.
- However, at certain decision points, the customer needs to examine work already done and make decisions about the future direction of the project. The project will require representatives of the supplier and customer to interact at many points in the development cycle - for example, users need to be available to provide information needed to carry out effective detailed interface design.
- This interaction, or other external factors, often leads to changes being needed, which effectively vary the terms of the contract and so a careful change control procedure is needed. Each of these topics will now be tackled in a little more detail.

- When a the contract is being negotiated, certain key points in the project can be identified where customer approval is needed before the project can proceed. For incremental delivery, example, a project to develop a large system can be divided into increments. For each increment there could be an interface design phase, and the customer needs to approve the designs before the increment is built. There could also be a decision point between increments.
- For each decision point, the deliverables to be presented by the suppliers, the decisions to be made by the customer and the outputs from the decision point all need to be defined. These decision points have added significance if payments to the supplier are based on them. Not only the supplier but also the customer has responsibilities with respect to these decision points - for example, the supplier should not be unnecessarily delayed while awaiting customer approval of some interim deliverable.
- Where work is contracted out there will be a general concern about the quality of that work.

Acceptance

- When the work has been completed, the customer needs to take action to carry out acceptance testing. The contract might put a time limit on how long acceptance testing can take, so the customer must be organized to carry out this testing before the time limit for requesting corrections expires.
- We have already noted that some software houses are rather cursory with their pre-acceptance testing: the implication seeming to be that they would rather the users spent their time on testing than they themselves. This imposition can be reduced by asking to approve the supplier's internal test plans.
- Part or all of the payment to the supplier will depend on this acceptance testing. Sometimes part of the final payment will be retained for a period of operational running and is eventually paid over if the levels of reliability are as contracted for.

Cont…

- There is usually a period of warranty during which the supplier should fix any errors found for no charge.
- The supplier might suggest a very short warranty period of say 30 days. It is in the customer's interests to negotiate a more realistic period of say at least 120 days.

SOFTWARE PROJECT MANAGEMENT

STAFFING IN SOFTWARE PROJECTS

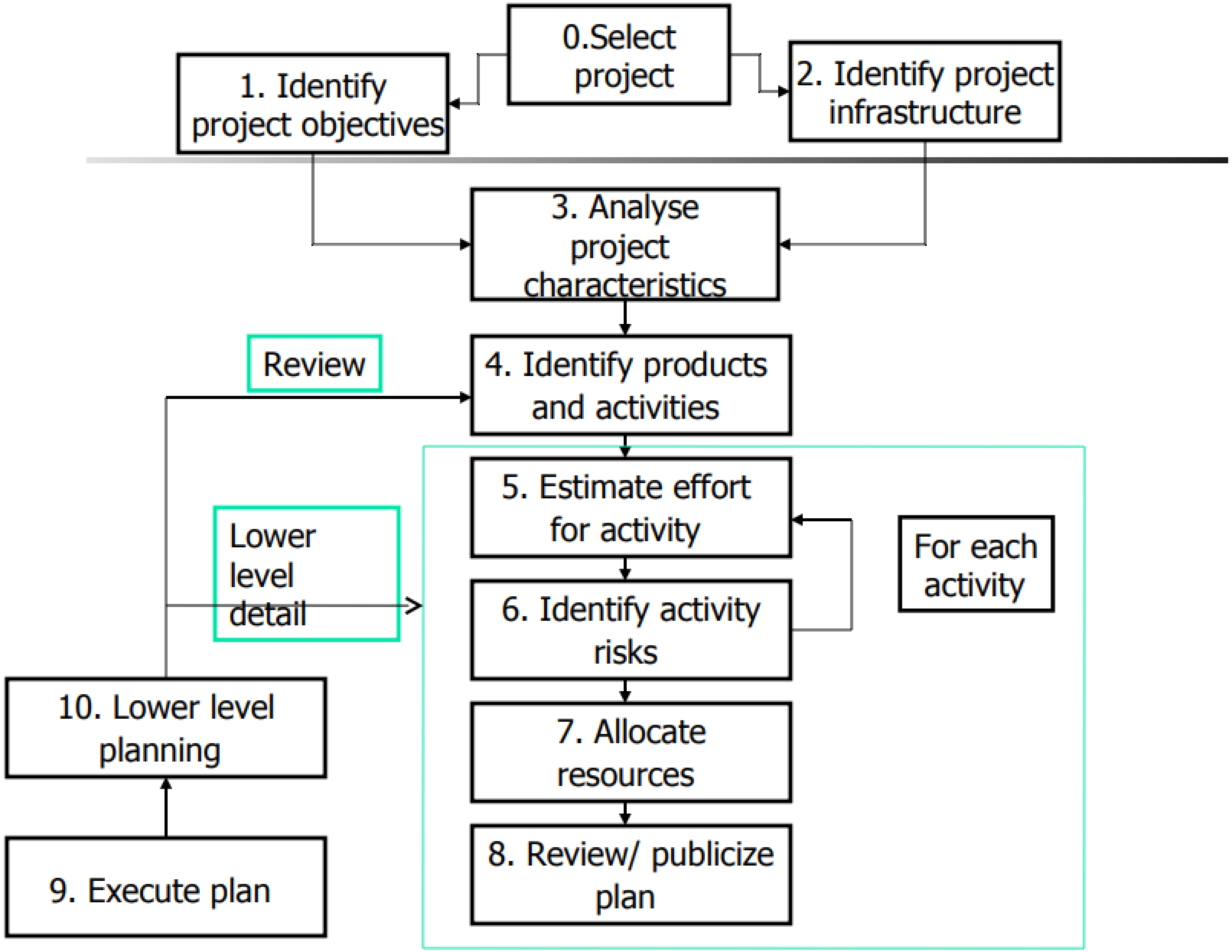
Module 5

PRIYANKA SINGH

Introduction

- We will look at some aspects of organizational behaviour (OB) research. There will be three concerns: staff selection, staff development and, which will be dealt with in more detail, staff motivation.
- We will look at how the project leader can encourage effective group working and decision making while balancing this, where needed, by purposeful leadership. The final part of the chapter looks at some of the more formal aspects of organizational structures.
- The issues raised in this chapter have impacts at all stages of project planning and execution but in particular at the following points (shown in figure on next page)
  - although perhaps having little control over organizational structure, the project leader needs to be aware of its implications (Step 2);
  - the scope and nature of activities can be set in a way that will enhance staff motivation (Step 4);
  - many risks to project success relate to staffing (Step 6);
  - the qualities of individual members of staff should be taken into account when allocating staff to activities (Step 7)

‘Step Wise’ - an overview



Understanding behaviour

- People with practical experience of working on projects invariably identify the handling of people as one of the must important aspects of project management. What project manager will want to know is whether the effective and sensitive management of staff comes only from experience or whether guidance can be useful. The field of social science known as organizational behaviour (OB) helps.
- This has evolved theories that try to explain people's behaviour and that tend to be structured 'If A is the situation then B is likely to result'. Attempts are then made to observe behaviour or to conduct experiments where variables for A and B are measured and a statistical relationship between the two variables is sought. Unlike physical science, it is rarely, if ever, the case that it can be said that B must always follow A.
- A major problem is that in the real world there is bound to be a very wide range of influences on a situation, many of which will not be apparent to the observer. It is therefore difficult to decide which set of research findings is relevant. A danger is that we end up with a set of maxims that are little better than superstitions.

Organizational behaviour a background

- The roots of studies in OB can be traced back to work done in the late 19th and early 20,th centuries by Frederick Taylor.
- By studying the way that manual workers did tasks, he attempted to work out the most productive way of doing these tasks. The workers were then trained to do the work in this way.
- Taylor had three basic objectives:
  - to select the best person for the job;
  - to instruct such people in the best methods;
  - to give incentives in the form of higher wages to the best workers.
- The conditions under which the staff worked also affects productivity.
- OB researchers discovered that the state of the minds of the people influenced productivity.

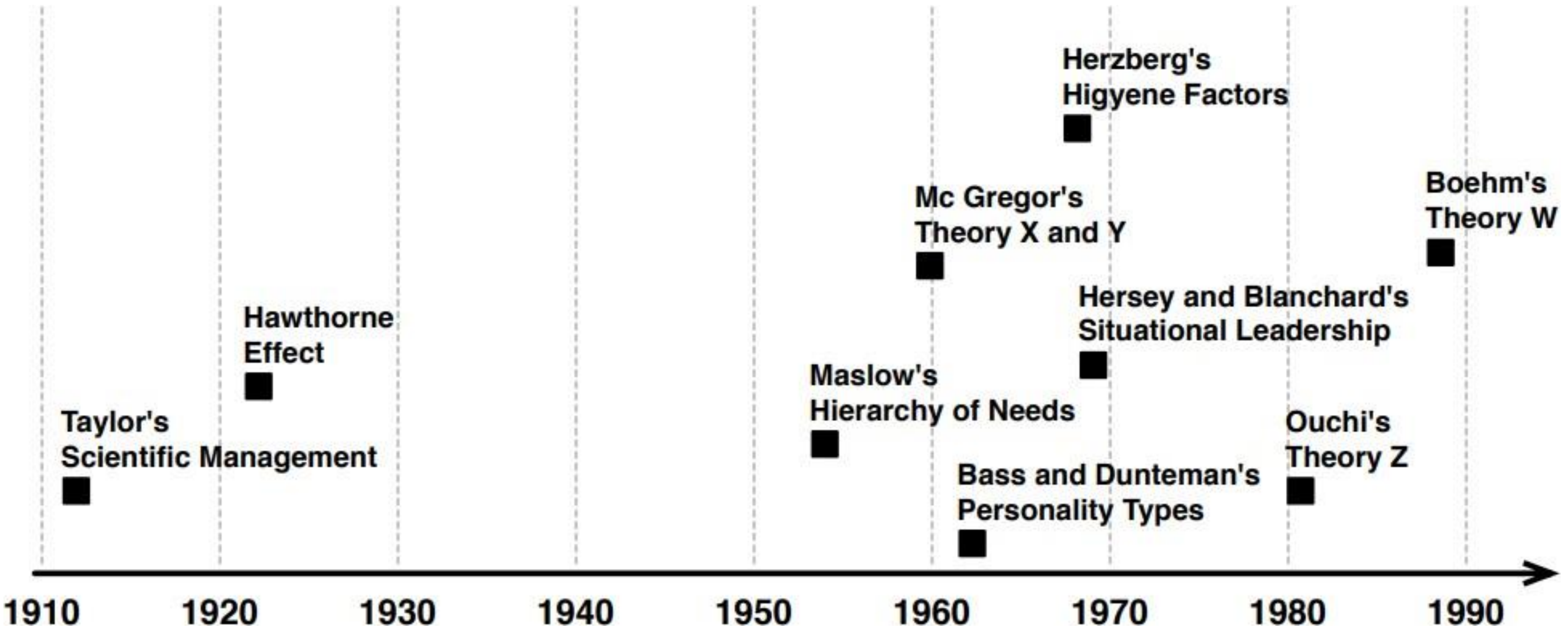
Cont…

- Taylorism' is often represented as crude and mechanistic these days. Interestingly though, the Taylorist approach is one that is adopted, in pan, in modem sports coaching.
- Understanding what motivates people at work is difficult (but with a great potential impact).
- During the 1920s, OB researchers discovered, while causing out a now famous set of tests on the conditions under which staff worked best, that not only did a group of workers for whom conditions were improved increase their work-rates, but a control group, for whom conditions were unchanged, also increased their work-rates.
- Simply showing a concern for what workers did increased productivity. This illustrated how the state of mind of workers influenced their productivity.

Cont…

- The cash-oriented view of work of some managers can thus be contrasted with a more rounded vision of people in their place of work. The two attitudes were labelled Theory X and Theory Y by Donald McGregor.
- The Hawthorne effect (studies about the effect of lighting conditions on productivity) demonstrates how difficult it is to come out with accurate theories
- Organized in:
  – Motivational factors
  – Personality traits/Interaction types
  – Management strategies

Timeline



Herzberg's
Higyene Factors
Boehm's
Theory W
Mc Gregor's
Theory X and Y
Hersey and Blanchard's
Situational Leadership
Hawthorne
Effect
Maslow's
Hierarchy of Needs
Taylor's
Scientific Management
Ouchi's
Theory Z
Bass and Dunteman's
Personality Types
1910
1920
1930
1940
1950
1960
1970
1980
1990

Theory X - Donald McGregor.

- The average human has an innate dislike of work.
- There is a need therefore for coercion, direction and control.
- People tend to avoid responsibility.
- Human beings are poorly creative in solving organizational problems.
- Motivation is mainly related to satisfying physical/security needs.

Theory Y - Donald McGregor

- People are, on average, very creative
- Motivation is often self-realization and self-esteem
- Work is as natural as rest or play
- External control and coercion are not the only ways of bring
effort directed towards an organization's end
- 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.

Cont…

- The two theories entail different styles of leaderships:
  - Theory X entails an autocratic leadership
  - Theory Y entails a climate of trust
- Managers will apply theory X or Y management style according to their opinion of the people they deal with.
- One way of judging whether a manager espouses Theory X or Theory Y is to observe how the manager's 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. McGregor's distinction between the two theories also draw attention to the way that expectations influence behavior.
- If a manager or teacher assumes that you are going to work diligently and do well, then you are likely to try and meet these expectations.

1- Selecting the right person for the job

- Taylor stressed the need for the right person for the job. Many factors, such as the use of software tools and methodologies, affect programming productivity. However, one of the biggest differences in software development performance is among individuals.
- What sort of characteristics should they be looking for? Should they go, for example, for the experienced programmer or the new graduate with the first class mathematics degree?
- It is extremely dangerous to generalize but looking specifically at behavioral characteristics, the American researcher Cheney found that the most important influence on programmer productivity seemed to be experience.
- Besides the s/w tools the individuals selected for a job affect the programming productivity.
- Person who can communicate well and more importantly; with users. .

The American researchers Conger and Zawacki found that computing people would appear to have much weaker ‘social needs’ than people in other professions. They quote Gerald Weinberg; 'If asked most programmers preferably say they prefer to work alone where they wouldn’t get disturbed by other people.' This is reflected in the problem that people attracted to writing software, and are good at it, will not make good managers later in their careers.

The recruitment process

- Recruitment is often an organizational responsibility.
  - Eligible candidates- have a curriculum vitae which shows the right and required details
  - Suitable candidates- who can actually do the job well.
- A general approach might be the following.
  - ***Create a job specification-*** Advice is needed, as there will be legal implications in an official document. However, formally or informally, the requirements of the job should be documented and agreed.
  - ***Create a job holder profile-*** Using the job specification, a profile of the person needed to carry out the job is constructed. The qualities, qualifications, education and experience required would be listed.
  - ***Obtain applicants-*** Typically, an advertisement would be placed, either within the organization or outside in the trade or local press. The job holder profile would be examined carefully to identify the medium most likely to reach the largest number of potential applicants at least cost. For example, if a specialist is needed it would make sense to advertise in the relevant specialist journal.

- ***Examine CVs-*** These should be read carefully and compared to the job holder profile- nothing is more annoying for all concerned than when people have CVs which clearly indicate that they are not eligible for the job and yet they are called for interview.

- ***Interviews etc***.- A number of different selection techniques can be tried, including aptitude tests, personality tests, and the examination of samples of previous work. All these methods must be related to specific qualities detailed in the job holder profile. Interviews are the most commonly used method. It is better if there is more than one interview session with an applicant and with each session there should not be more than two interviewers because a greater number reduces the possibility of follow-up questions and discussion.

- ***Other procedures-*** References will need to be taken up where necessary, and a medical examination might be needed.

2- Instruction in the best methods

- When a new member of the team is recruited, the team leader will need to plan that person's induction into the team very carefully. Where a project is already well under way, this might not be easy. However, the effort should be made - it should pay off eventually as the new recruit will become a fully effective member of the team more quickly.
- The team leader should also be aware of the need to assess continually the training needs of their team members. Just as you formulate a user requirement before considering a new system, and you construct a job holder profile before recruiting a member of staff, so a training needs profile is drawn up for each staff member before you consider specific courses. Some training can be provided by commercial training companies. Where money is tight, other sources of training should be considered but training should not be abandoned altogether even if it consists only of a team member's being told to find out about a new software tool and then demonstrating it to colleagues.
- The methods learnt need, of course, to be actually applied. Reviews and inspections should help to ensure this.

3- Motivation

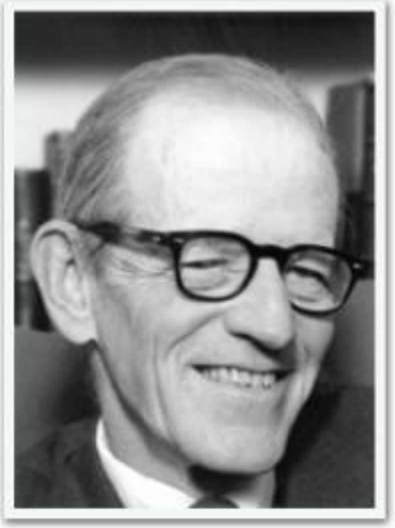
- Motivation and application can often make up for shortfalls in innate skills
- Taylor's approach - financial incentives
- Abraham Maslow (1908-1970)
  - motivations vary from individual to individual. People will be motivated by different things at different times. People will always feel dissatisfied, but the focus of the dissatisfaction changes over time.
  - hierarchy of needs – as lower ones fulfilled, higher ones emerge

Taylor (1911)

- First systematic theory on management. Most of it obsolete by the 30's, but seminal
- Main objective is improving economic efficiency and labor productivity

- Considerations about workforce:
  * Most workers who are forced to perform repetitive tasks tend to work at **the slowest rate that goes unpunished**.
  * Workers could not be relied upon for talent or intelligence and all workers behave similarly: high control is needed
  * Need for better pays (linked to outputs)
  * Breaks are necessary, but simply as a way to improve efficiency

Hawthorne Effect



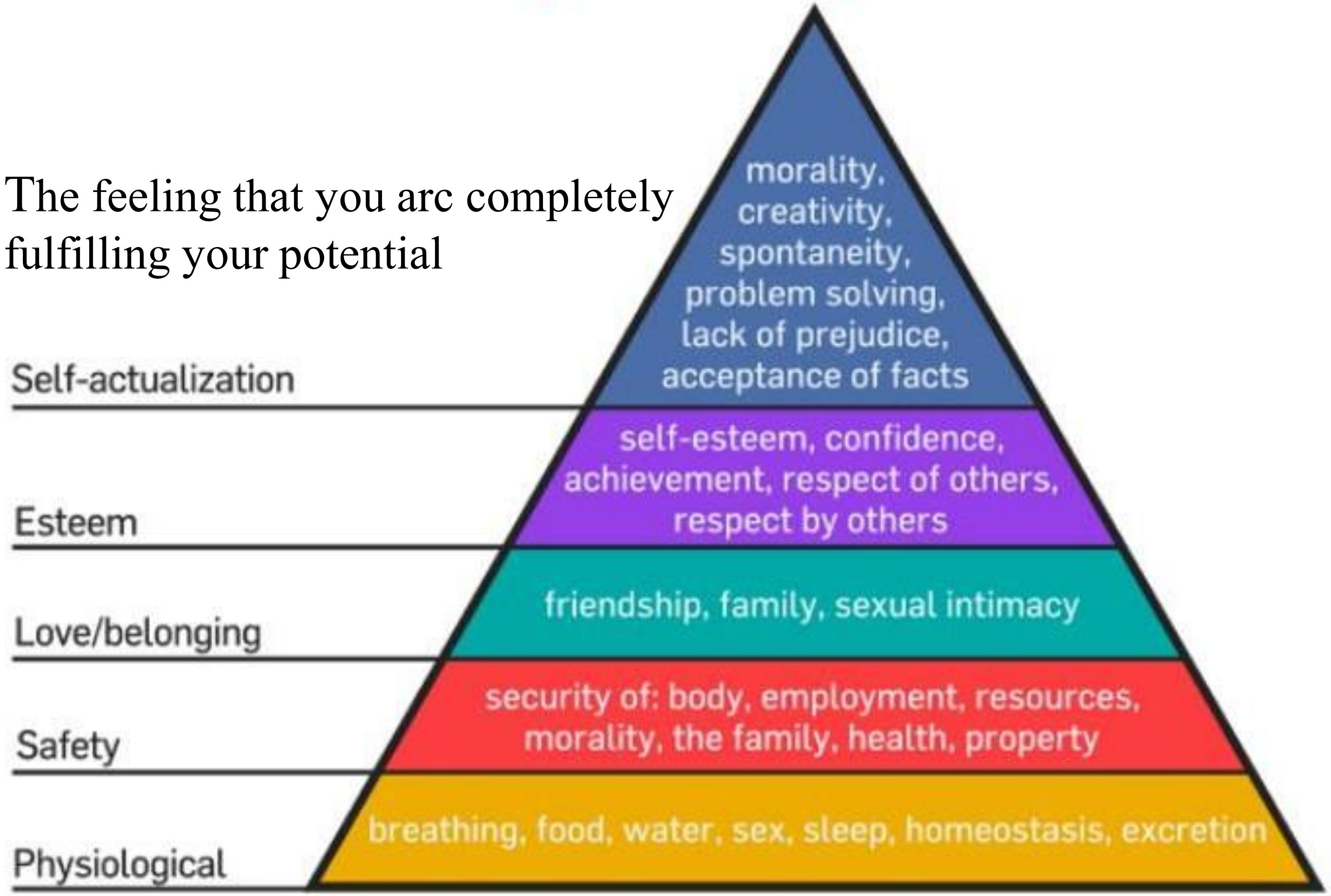
George Elton Mayo

- Studies conducted in the twenties to understand the effect of different lighting conditions on productivity
- Results indicated that increasing or decreasing lighting conditions improved workers' performances
- Conclusion: attention and feeling to be part of something important is what really matters
- Seminal work which would be later criticized (small numbers of workers involved)

Maslow’s Hierarchy of Needs

Maslow (1954)

The feeling that you arc completely
fulfilling your potential



Maslow's Hierarchy of Needs

- Lower level of the hierarchy: d-needs (deficiency needs). If not met, distress
- Higher level of the hierarchy: fulfilment
- A sound management ensures that:
  - d-needs are satisfied
  - Higher levels of the hierarchy are achieved

Frederick Herzberg
(1968)

–Three prevailing philosophies:

* Organizational theorist think humans are irrational. The only way is organizing work
* Industrial engineers believe that humans are mechanistically oriented and economically motivated
* Behavioral scientists believe in creating a cozy environment

–Different positive and negative KITAs (Kick In the Ass) have been tried: physical and phycological

–They do not work!

Certain things about a job might make you dissatisfied, if the causes of this dissatisfaction are removed, this does not necessarily make the job more etching.

Frederick Herzberg proposed the “Two Factor Theory”

1. Motivational Factors: which make you feel that the job is worthwhile, like a sense of achievement or the nature of the work itself .
2. Hygiene Factors: which can make you dissatisfied if they are not right, for example, the level of pay or the working conditions;

Motivational and Hygiene Factors

Motivational Factors	**Hygiene Factors**
Achievement	Company Policy and Administration
Recognition	Supervision
Work itself	Relationship with supervisor
Responsibility	Work Conditions
Advancement	Salary
Growth	Relationship with peers
	Personal life
	Relationship with subordinates
	Status
	Security

Vroom -The expectancy theory of motivation

- Vroom and colleagues identified three influences on motivation
  - Expectancy – the belief that working harder leads to better performance
  - Instrumentality – the belief that better performance will be rewarded
  - Perceived value- of the resulting reward
- Note: If any of the above factors has a zero value, then motivation will also be zero. Motivation will be high when all three factors are high. A zero level for any one of the factors can lead to a lack of motivation.
- If you are wording on a package for a user and, although you think you can get it to work, you discover that the user has started employing an alternative package and no longer needs this one then you will probably feel you are wasting your lime and give up {zero instrumentality).
- Given that the users really do want the package, your reward in this is set of circumstances might simply be a warm feeling that you have helped your colleagues and that they are grateful to you.

Oldham-Hackman Job Characteristics

- Managers should try to group together the elements of the tasks that need 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 five factors. The first three factors make the job 'meaningful' to the person who is doing it:
  - skill variety, the number of different skills that the job holder has the opportunity to exercise;
  - task identity, 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 work.

Methods of improving motivation

- Setting 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.
- Providing feedback- Not only do goals have to be set but staff have to have regular feedback about how they are progressing.

•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 and job enrichment.
  - •Job enlargement -The scope of the job is increased so that the member of Job enlargement and job staff carries out a wider range of activities. It is the opposite of increasing enrichment are based on specialization. For example, a programmer in a maintenance group might be the work of F.
  - •Job enrichment- In this case, the job is changed so that the holder carries out tasks that are normally done at a higher, managerial, level. Staff might be given responsibility for ordering consumables, for scheduling their work or for quality control. With a programmer in a maintenance team, they might be given authority to accept requests for changes which involved less than five days' work without the need for their manager's approval.

Becoming a team

- Simply throwing people together does not mean that they will immediately be able to work together as a team. Group feelings develop over a period of time. One suggestion is that teams go through five basic stages of development:
- Five basic stages of development:
  - forming - the members of the group get to know each other and try to set up some ground rules about behavior;
  - 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.

Belbin came to the conclusion that teams needed a balance of different types of people.

- **The chair-** Not necessarily a brilliant leader but must be good at running meetings, 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 select 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, for example by 'jollying people along'.
- **The resource investigator-** Adept at finding resources in terms of both physical resources and information.
- **The completer-finisher** Good at completing tasks.
- **The company worker-** A good team player who is willing to undertake less attractive tasks if they are needed for team success.

Group performance

- Are groups more effective than individuals working alone? Given the preference of many people attracted to software development for working on their own, this is an important question. In many projects, judgement's need to be made about which tasks are best carried out collectively and which are best delegated to individuals to do on their own.
- One way of categorizing group tasks is into:
  - Additive tasks
  - Compensatory tasks;
  - Disjunctive tasks;
  - Conjunctive tasks.
- Additive tasks are where the efforts of each participant are added together to get the final result, as in a gang of people clearing snow.
- With disjunctive tasks there is only one correct answer. The effectiveness of the group depends on someone coming up with the right answer and the others recognizing it as being correct.
- Conjunctive tasks are where progress is governed by the rate of the slowest performer.

  Software production where different staff are responsible for different modules seems to be a prime example of this.

Decision making

- Decisions 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 of categorizing decisions is by the amount of risk and uncertainty that is involved.
- Yet another distinction is between the rational-economic model and the satisficing model. The rational-economic model of decision making is the basis of classical economics. It predicts, for example, that a prospective buyer of personal computer equipment will purchase goods at the lowest possible price. This assumes that the decision maker has a complete knowledge of the state of the market. In order to achieve this, days, weeks, or months could be spent phoning dealers.
- Sensible people probably follow a satisficing approach and would look at a limited number of representative outlets to get a general idea of prices. Any potential loss of money through having missed an even lower offer would probably be offset by the savings in time, phonecalls, travel and so on.

Cont…

Some mental obstacles to good decision making

We have to accept the role of intuition but must be aware that there are some mental obstacles to effective intuitive thinking, for example:

- ***Faulty heuristics-*** Heuristics mean rules of thumb. Rules of thumb can be useful but there are dangers:
    - they are based only on the information that is to hand and this can be misleading;
    - they are based on stereotypes, such as accepting a Welshman into a male voice choir without an audition because of the 'well-known fact' that the Welsh are a great singing nation.
- ***Escalation of commitment-*** This refers to the way that once you have made a decision it is increasingly difficult to alter it even in the face of evidence that it is wrong.
- ***Information overload-*** It is actually possible to be presented with too much information so that you 'cannot see the wood for the trees'.

Cont…

Group decision making

- Assuming that the meetings are genuinely collectively responsible and have been properly briefed, research shows that groups are better at solving complex problems where the members of the group have complementary skills and expertise. The meeting allows them to communicate freely and to get ideas accepted.
- Groups are less effective when dealing with poorly structured problems, which need creative solutions. Brainstorming techniques have been developed to help groups in this situation but research shows that people often come up with more ideas individually than in a group.

Cont…

Obstacles to good group decision making

- Group decision making has some disadvantages: it is time consuming; it can in some cases stir up conflicts within the group; and decisions can be unduly influenced by dominant members of the group.
- Conflict could, in fact, be less than might be expected. Experiments have shown that people will modify their personal judgements to conform to group norms. These are common attitudes that are developed by a group over a period of time.
- You might think that this would tend to moderate the more extreme views that some individuals in the group might hold. In fact, people in groups often make decisions that carry more risk than where they have to make the decision on their own. This is known as the risky shift.

Cont…

Measures to reduce the disadvantages of group decision making

- One method of making group decision making more efficient and effective is by training members to follow a set procedure. The Delphi technique endeavours to collate the judgements of a number of experts without actually bringing them face-to-face.
- The big problem with this approach used to be that because the experts could be
geographically dispersed the process was time consuming.

Leadership

- Leadership is the ability to influence others in a group to act in a particular way in order to achieve group goals. A leader is not necessarily a good manager or vice versa, because managers have other roles to play, such as those of organizing, planning and controlling. Leadership is based on the idea of some kind of authority or power, although leaders do not necessarily have much formal authority. This power comes from either the person's position (position power) or from the person's individual qualities (personal power) or can be a mixture of the two.
- Position power has been further analysed into:
  - 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 confer rewards on those who carry out tasks to their satisfaction.
- Personal power, on the other hand, can be further analysed into:
  - expert power, which comes from being the person who is able to do a specialized task;
  - information power, where the holder has access to information that others do not;
  - referent power, which is based on the personal attractiveness of the leader.

Leadership styles

- Attempts have been made to measure leadership styles on two axes: directive vs. permissive and autocratic vs. democratic:
  - directive autocrat makes decisions alone with close supervision of their implementation;
  - permissive autocrat makes decision alone but gives subordinates latitude in implementation;
  - directive democrat makes decisions participatively but closely supervises their implementation;
  - permissive democrat makes decisions participatively and gives subordinates latitude in implementation.
- Another axis on which there have been attempts to measure management qualities has been on the degree to which a manager is task-oriented, that is, the extent to which the execution of the task at hand is paramount, and the degree to which the manager is concerned about the people involved (people orientation).

Cont…

- Work environments vary according to the amount of control that can be exerted over the work.
- Some jobs are routine and predictable (as when dealing with batched computer output). Others may be driven by outside factors (as in the case of a help-desk) or are situations where future direction is uncertain (for example, at the early stages of a feasibility study).
- Where there is a high degree of uncertainty, subordinates will seek guidance from above and welcome a task-oriented management style. As uncertainty is reduced, the task-oriented manager is likely to relax and to become more people-oriented and this will have good results.
- People-oriented managers are better where staff can control the work they do and know what to do without referring matters to their line managers.

Organizational structures

1- Formal versus informal structures

- While organizational structures can have an enormous impact on the way a project is conducted, it is something that project leaders such as Amanda at IOE can often do little to change.
- The formal structure is the one that is expressed in the staff hierarchy chart. It is basically concerned with authority, about who has which boss. It is backed by an informal structure of contacts and communication that grows up spontaneously among members of staff during the course of work. When the unexpected happens it is often this system that comes into play. Over a period of time, the advantages and disadvantages of different organizational structures tend to even out - the informal organization gets built up and staff find unofficial ways of getting around the obstacles posed by the formal structure.

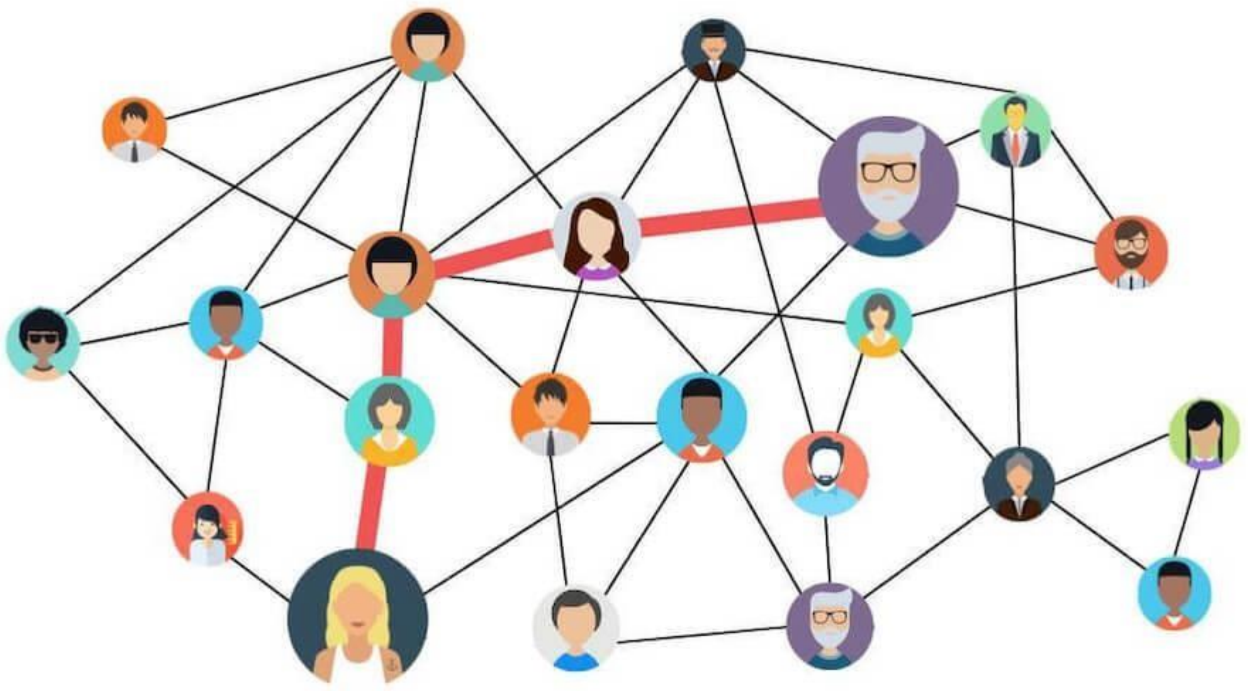
2- Hierarchical approach

The 'traditional' management structure is based on the concept of the hierarchy -each member of staff has only one manager, while a manager will have responsibility for several members of staff. Authority flows from the top down through the structure. A traditional concern has been with the span of control - the number of people that a manager can effectively control.

3- Staff versus line

Staff in organizations can often be divided into line workers who actually produce the end product and support staff who carry out supporting roles. In some organizations that produce software for the market or as a component of a larger product which is sold, the software specialists might be seen as part of the line. In a financial organization, on the other hand, the information systems department would probably be seen as part of the support staff.

- The definition of a virtual team is relatively new to project management and collaboration.
- A **virtual team** is a group of people who participate in common projects by making collaborative efforts to achieve shared goals and objectives. These people perform tasks and jobs in a virtual work environment created and maintained through IT and software technologies.
- This term often refers to teleworking teams or those groups of people who use telecommunication technologies – mobile IoT and Skype, for example – to share information and co- operate and collaborate in real-time.
- **Virtual team management** is the foundation for managing remote/telecommunication-based teamwork.



Virtual team

There are two types of virtual teams, such as follows:

- **Global virtual team**. As a rule, these teams are located in different countries and cities all over the world. They can be employees of several companies which join their efforts and resources (incl. people, technology, money) to perform shared outsourced projects and achieve common goals.
- **Local virtual team**. Members of a local virtual workgroup usually belong to the same company. That company is either big or small, and it has enough resources (technology is essential) to establish and maintain virtual team workplaces and organize its employees into a productive remote group.

What is virtual team management?

The combination of activities for assembling, building, organizing, controlling, and supervising the virtual teamwork is called **virtual team management**. It is an essential part of project team collaboration.

- Virtual team management includes, but not limited to, the following processes:

  - **Assembling**. Probation periods are the first measurements to be applied when starting with remote teamwork organization. The team leader should decide on those people who meet all the requirements of probation periods.
  - **Training**. During this process, the team leader sets expectations as to future virtual teaming and then develops and applies a group training methodology to teach the team members how to meet the expectations.
  - **Managing**. This process means using telecommunication technologies to manage ongoing tasks and jobs of remote group members.
  - **Controlling**. The team leader establishes performance measures to assess and evaluate team performance. This person needs to find out whether the team is on the right track and can achieve project goals on schedule.

Project communication plan

- A project communication plan is a simple tool that enables you to communicate effectively on a project with your client, team, and other stakeholders. It sets clear guidelines for how information will be shared, as well as who's responsible for and needs to be looped in on each project communication.

Why a project communication plan is important

- As the project manager, you've already mapped out every task and deliverable to get you across the finish line. Why not do the same for project communications? After all, your project plan needs a steady stream of communication to stay on track.
- A communication plan plays an important role in every project by:
  - Creating written documentation everyone can turn to
  - Setting clear expectations for how and when updates will be shared
  - Increasing visibility of the project and status
  - Providing opportunities for feedback to be shared
  - Boosting the productivity of team meetings
  - Ensuring the project continues to align with goals

Types of Communication

- Completing a complex project successfully requires good communication among team members. If those team members work in the same building, they can arrange regular meetings, simply stop by each other's office space to get a quick answer, or even discuss a project informally at other office functions. Many projects are performed by teams that interact primarily through electronic communication and are, therefore, called *virtual teams*.
- To avoid miscommunication that can harm trust and to include team members in a project culture, the project team needs a plan for communicating reliably and in a timely manner. This planning begins with understanding two major categories of communication.

1- Synchronous Communications

If all the parties to the communication are taking part in the exchange at the same time, the communication is **synchronous**. A telephone or Skype conference call is an example of synchronous communication. The following are examples of synchronous communications:

- *Live meeting:* Gathering of team members at the same location
- *Conference call:* A telephone call in which several people participate
- *Audio conference:* Like a conference call, but conducted online using software like Skype
- *Computer-assisted conference:* Audio conference with a connection between computers that can display a document or spreadsheet that can be edited by both parties
- *Video conference:* Similar to an audio conference but with live video of the participants. Some laptop computers have built-in cameras to facilitate video conferencing
- *IM (instant messaging):* Exchange of text or voice messages using pop-up windows on the participants' computer screens
- *Texting:* Exchange of text messages between mobile phones, pagers, or personal digital assistants (PDAs)—devices
  that hold a calendar, a contact list, a task list, and other support programs

2-Asynchronous Communications

Getting a team together at the same time can be a challenge—especially if they are spread out across time zones. Many types of communication do not require that the parties are present at the same time. This type of communication is asynchronous. There are several choices of asynchronous communications.

- Mail and Package Delivery
- Fax
- Email
- Project Blog
- Really Simple Syndication (RSS)

How to write a project communication plan

Ready to put your communication plan to paper? Writing a project management communication plan is as simple as following these 5 steps:

- List your project's communication needs. Every project is different. Take the size of the project, the nature of work being done, and even the client's unique preferences into account as you determine which types of communication this project needs to succeed.
- Define the purpose. Bombarding people with too many emails or unnecessary meetings can interfere with their ability to get work done and cause them to overlook important updates. Be purposeful in your plan, and ensure every communication you include has a reason for being. If you're feeling really ambitious, go ahead and outline a basic agenda for the topics that will be covered in each meeting or report.
- Choose a communication method. Do you really need a meeting to share weekly updates, or is your project discussion board enough? Think through how your team works best, so they can stay in the loop while still being productive. If your client prefers the personal touch of a phone call, build that into your plan too.
- Set a cadence for communication. Establishing a regular frequency for communication streamlines the process by setting clear expectations from the get-go. This not only frees you from fielding random requests for status updates. It also enables project members to carve out space for important meetings and reports ahead of time.
- Identify the owner and stakeholders. Assigning ownership creates accountability so your carefully crafted plan can reach its full potential. As the project manager, you'll be responsible for most communications, but there may be some you want to delegate to others. While you're naming names, list the audience or stakeholders for each communication type too. That way key players come prepared to provide updates when needed.

Example

Communication	Frequency	Goal	Owner
Project team			
Project status report	Weekly	Review project status and discuss potential issues or delays	Project manager
Team standup	Daily	Discuss what each team member did yesterday, what they'll do today, and any blockers	Project manager
Task progress updates	Daily	Share daily progress made on project tasks	Project manager
Project review	At milestones	Present project deliverables, gather feedback, and discuss next steps	Project manager
Post-mortem meeting	At end of project	Assess what worked and what did not work and discuss actionable takeaways	Project manager
Project sponsor			
Project status report	Weekly	Review project status and discuss potential issues or delays	Project manager
Project review	At milestones	Present project deliverables, gather feedback, and discuss next steps	Project manager

END