SDLS Steps

1- **Investigation Planning: find out the scope of the problem, determine the solutions, costs, time, benifits**

A- determine actor / customer /user (individual, group, organization, another system)

In planning you should make interview with actors to determine the system requirements. Each phase based on the previous

B- System requirements, user requirements, functional requirements non-functional requirements.

C- Design initial plan: what will be the system looks like?

Who will make or develop it?

How the system will be developed?

When the system will be deliverable?

D- Time Plan (design activity network): giving the actor the shortest time for him to deliver the system. Define duration and dependences if activities.

E- Estimate cost: determine the budget for the tools and materials. Calculate the cost for the system.

F- Design proposal system contract: making a contract with the actor dealing on it about how the system will be developed and delivered. Dealing on time, money (cost), and each milestone/

SDLS Steps

2- **Analysis (system requirements, user requirements, usability, functional and non functional requirements)**
3- Designing (UML diagram, DB (tables, views, queries, SQL), Forms, reports)
4- System Implementation (coding and developing)
5- Maintenance of the final system (finishing and ensure verification and validation).
6- Testing the delivered system (running).

System Requirements

- To introduce the concepts of **user** and **system** Requirements
- To describe functional and non-functional requirements
- To explain how system requirements may be organized in a requirements document

Requirements Re-engineering

- The process of **establishing** **Customer Services** **and the Constraints**
- **Requirements: is a high-level abstract statement of a service or of a system constraint to a detailed mathematical functional specification.**

Types of requirement

- **User requirements**
  - **the services provided by the system ,its operational constraints. Written for customers.**
  - **User requirements are defined using natural language, tables and diagrams as these can be understood by all users.**
- **System requirements**
  - **1- A structured document setting out**

  (detailed descriptions of the system's functions SF, services S and operational constraints OC.
  - **2- Defines what should be implemented**

  so may be part of a contract between client and contractor.

System requirements

- **More detailed specifications of system functions, services and constraints** than user requirements.
- They are intended to be **a basis for designing the system**.
- They may be incorporated into the system **contract**.
- System requirements may be defined or illustrated **using system models** discussed in Chapter 8.

Definitions and specifications

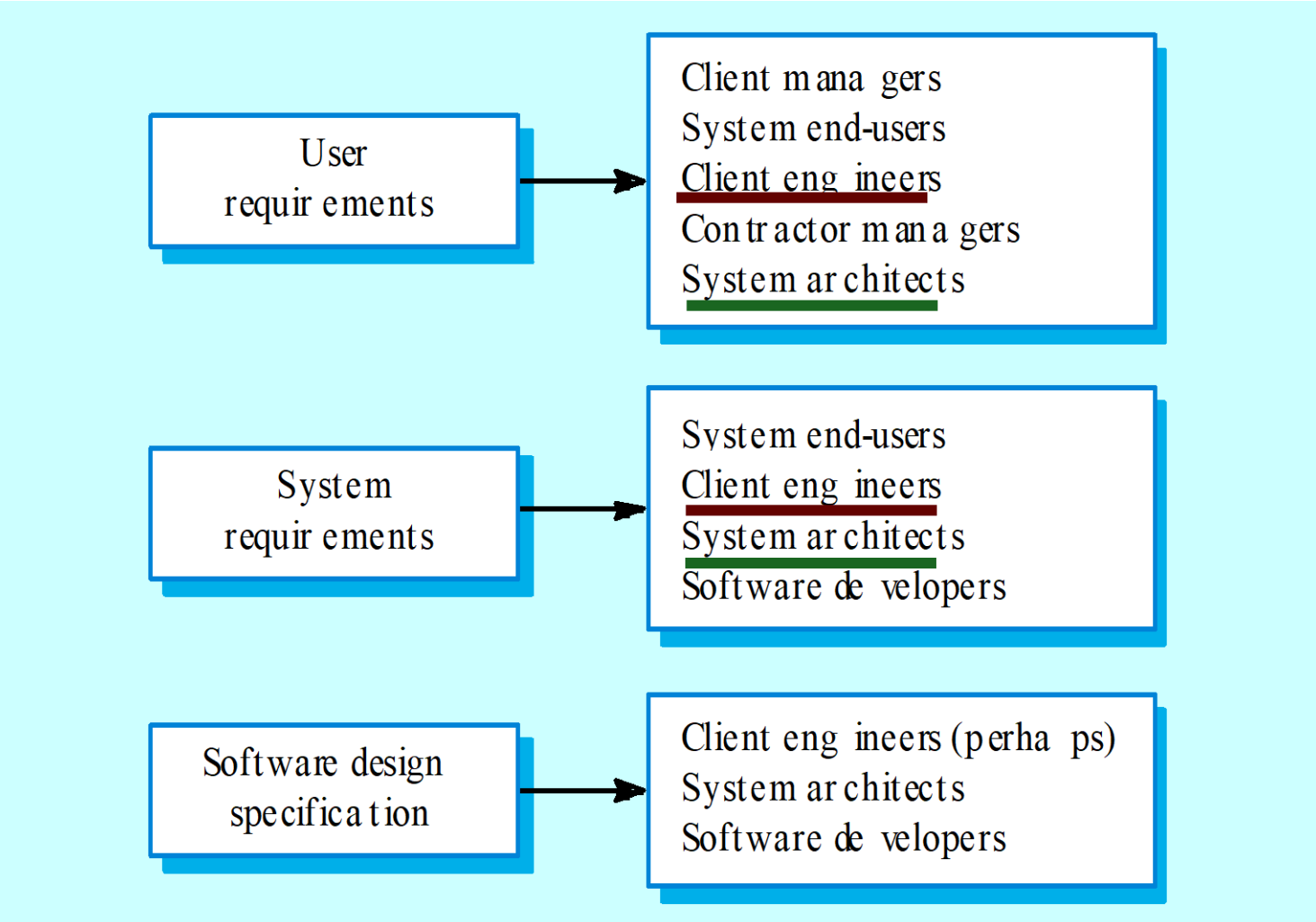
User requirement definition

1. The software must provide a means of representing and accessing external files created by other tools.

System requirements specification

1.1 The user should be provided with facilities to define the type of external files.
1.2 Each external file type may have an associated tool which may be applied to the file.
1.3 Each external file type may be represented as a specific icon on the user's display.
1.4 Facilities should be provided for the icon representing an external file type to be defined by the user.
1.5 When a user selects an icon representing an external file, the effect of that selection is to apply the tool associated with the type of the external file to the file represented by the selected icon.

Requirements readers



Functional and non-functional requirements

- **Functional requirements**
  - **Statements of services the system should provide,**
  - **how the system should react to particular inputs ,**
  - **how the system should behave in particular situations.**
- **Non-functional requirements**
  - **constraints on the services or functions offered by the system**
  - **such as** timing constraints, constraints on the development process, standards, etc.
  - **Every process is functional**
  - **Every Condition is non functional**

Functional requirements

- **Describe functionality or system services.**
- **Depend on the type of :**

software, expected users and the type of system.

- **what the system should do ?**
- **describe the system services in detail.**

The LIBSYS system (library system)

- A library system that provides **a single interface** to **a number of databases** of articles in **different libraries.**

FRs

- Users can **search for**, **download and print** these articles for personal study.

Examples of functional requirements

- The user able to **search** either all of the initial set of databases or **select** a subset from it.
- The system **provide viewers** for the user to read documents in the document store.
- Every order **allocated a unique identifier** (ORDER_ID) which the user shall be able to copy to the account's permanent storage area.

Requirements completeness and consistency

- **In principle, requirements should be both complete and consistent.**
- **Complete**
  - **They should include descriptions of all facilities required.**
- **Consistent**
  - **There should be no conflicts or contradictions in the descriptions of the system facilities.**
- **In practice, it is impossible to produce a complete and consistent requirements document.**

Non-functional requirements

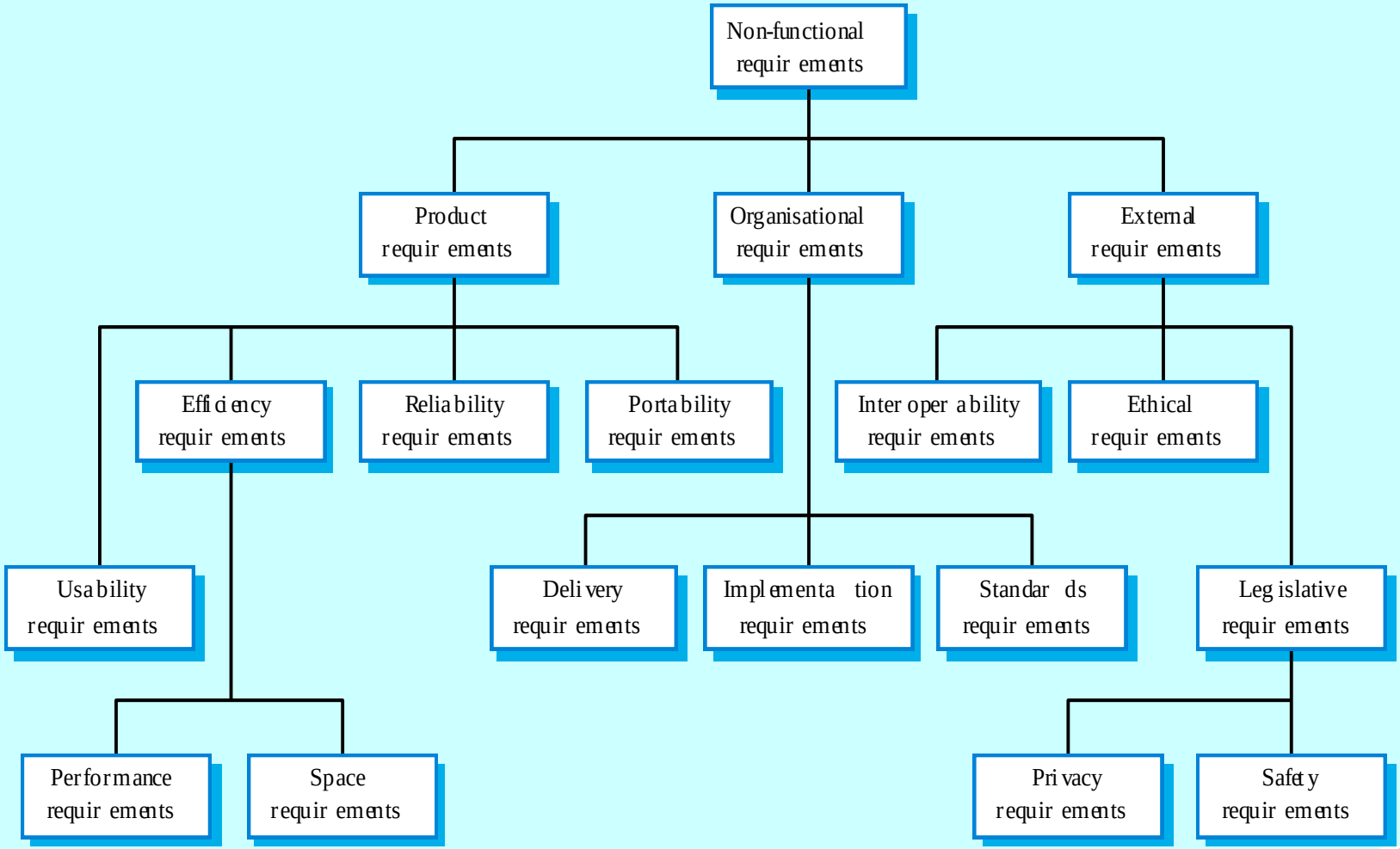
- **These define system properties and constraints**
- reliability,
- response time
- storage requirements.
- Constraints are I/O device capability,
- system representations, etc.
- **Non-functional requirements may be more critical than functional requirements.**

- **If these are not met, the system is useless.**

Non-functional classifications

- **Product requirements**
  - **Requirements which specify that the delivered product must behave in a particular way e.g. execution speed, reliability,..**
- **Organizational requirements**
  - **Requirements which are a consequence of organisational policies and procedures e.g. process standards used, implementation requirements, etc.**
- **External requirements**
  - **Requirements which arise from factors which are external to the system and its development process e.g. interoperability requirements, legislative requirements, etc.**

Non-functional requirement types



Non-functional requirements examples

- Product requirement

  8.1 The user interface for LIBSYS shall be implemented as **simple HTML without frames or Java applets.**

- Organisational requirement

  9.3.2 The system development process and deliverable documents shall conform to the process and deliverables defined in **XYZCo-SP-STAN-95.**

- External requirement

  7.6.5 The system shall not disclose any personal information about customers apart from their name and reference number to the operators of the system.

Examples

- **A system goal**
  - The system should be **easy to use** by experienced controllers and should be organised in such a way that user **errors are minimised.**
- **A verifiable non-functional requirement**
  - Experienced controllers shall be able to use all the system functions after a total of **two hours training**.
  - After this training, the average number of **errors** made by experienced users shall **not exceed two per day.**

- **Every process is functional**
- **Every Condition is non functional**

Requirements measures

Property	Measure
Speed	Processed transactions/second User/Event response time Screen refresh time
Size	M Bytes Number of ROM chips
Ease of use	Training time Number of help frames
Reliability	Mean time to failure Probability of unavailability Rate of failure occurrence Availability
Robustness	Time to restart after failure Percentage of events causing failure Probability of data corruption on failure
Portability	Percentage of target dependent statements Number of target systems

Problems with natural language

- Lack of clarity
  - Precision is difficult without making the document difficult to read.
- Requirements confusion
  - Functional and non-functional requirements tend to be mixed-up.
- Requirements combinations
  - Several different requirements may be expressed together.

Requirement problems

- Database requirements **includes both conceptual and detailed information**
  - **Describes the concept** of a financial accounting system that is to be included in LIBSYS;
  - includes **the detail that managers can configure** this system - this is unnecessary at this level.
- Grid requirement mixes three different kinds of requirement
  - Conceptual functional requirement (the need for a grid);
  - Non-functional requirement (grid units);
  - Non-functional UI requirement (grid switching).

Guidelines for writing requirements

- Invent **a standard format** and use it for all requirements.
- Use language in a consistent way. Use **shall** for **mandatory requirements**, **should** for **desirable requirements.**
- Use **text highlighting** to identify key parts of the requirement.

Q: Name customers and possible **functional requirement and non functional** requirement between the given requirements, for the given scenario.

"You are required to design the software for an automated teller machine (ATM). The ATMs are capable of withdrawal of cash for this example. An ATM accepts a cash card, interacts with the user, and verifies the PIN number provided, carries out the transaction, dispenses cash, and prints receipts. ATMs communicate with a central computer, which clears the transactions with the appropriate bank"

- Solution:
- Process are functional
- Conditions are non functional

The project plan

The project plan sets out:
–The resources available to the project;
–The work breakdown;
–A schedule for the work

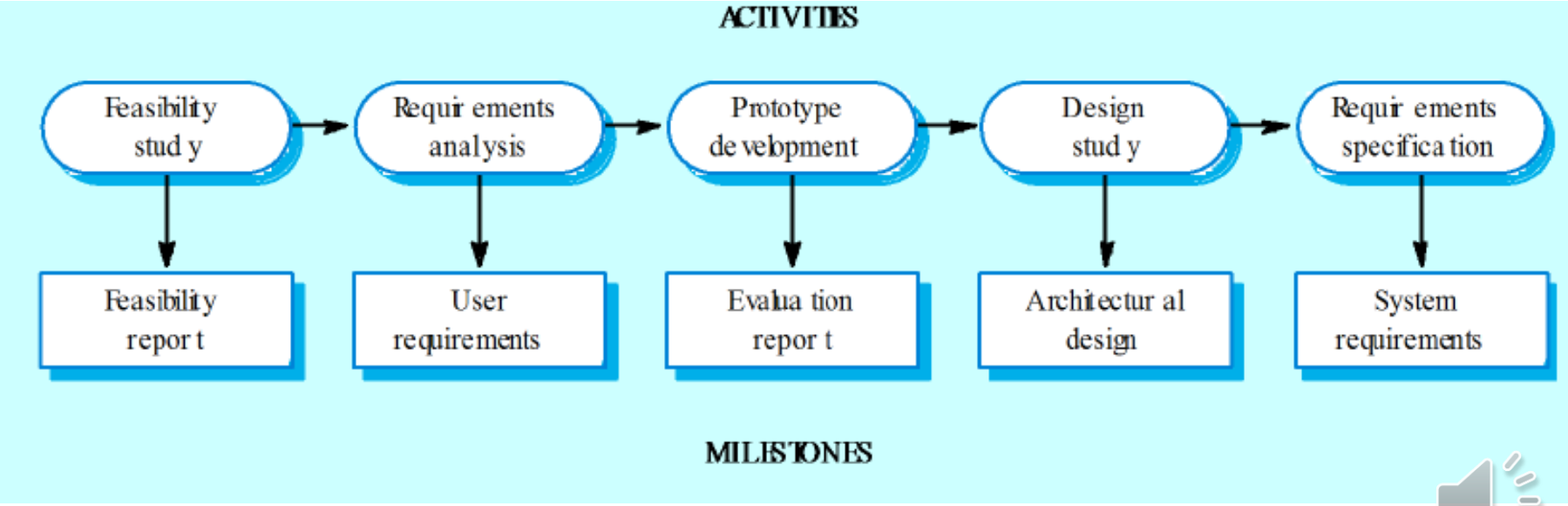
Project plan structure

- Project organization.
- Risk analysis.
- Hardware and software resource requirements.
- Work breakdown.
- Project schedule.
- Monitoring and reporting mechanisms.

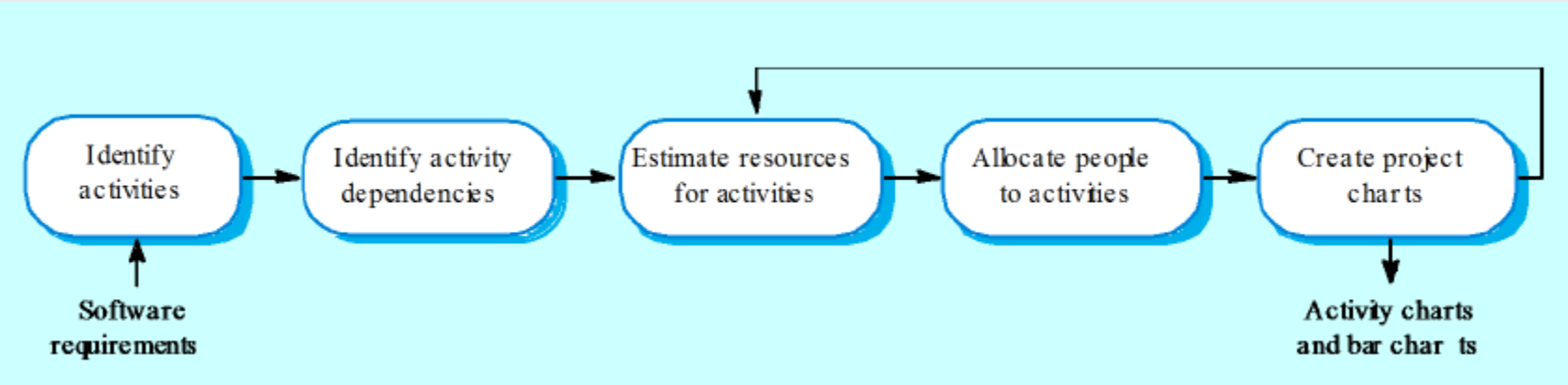
Activity organization

- **Activities in a project should be organized** to produce tangible outputs for management to judge progress.
- ***Milestones* are the end-point of a process activity.**
- ***Deliverables*** are project results delivered to customers.
- **The waterfall process allows for the straightforward definition of progress milestones**

Milestones in the RE process



The project scheduling process



Scheduling problems

- Estimating **the difficulty** of problems
- Estimating **the cost** of developing a solution.
- Productivity **no. of people** working on a task.
- Adding people to a late project makes it later because of **communication overheads.**
- The **unexpected** always happens.

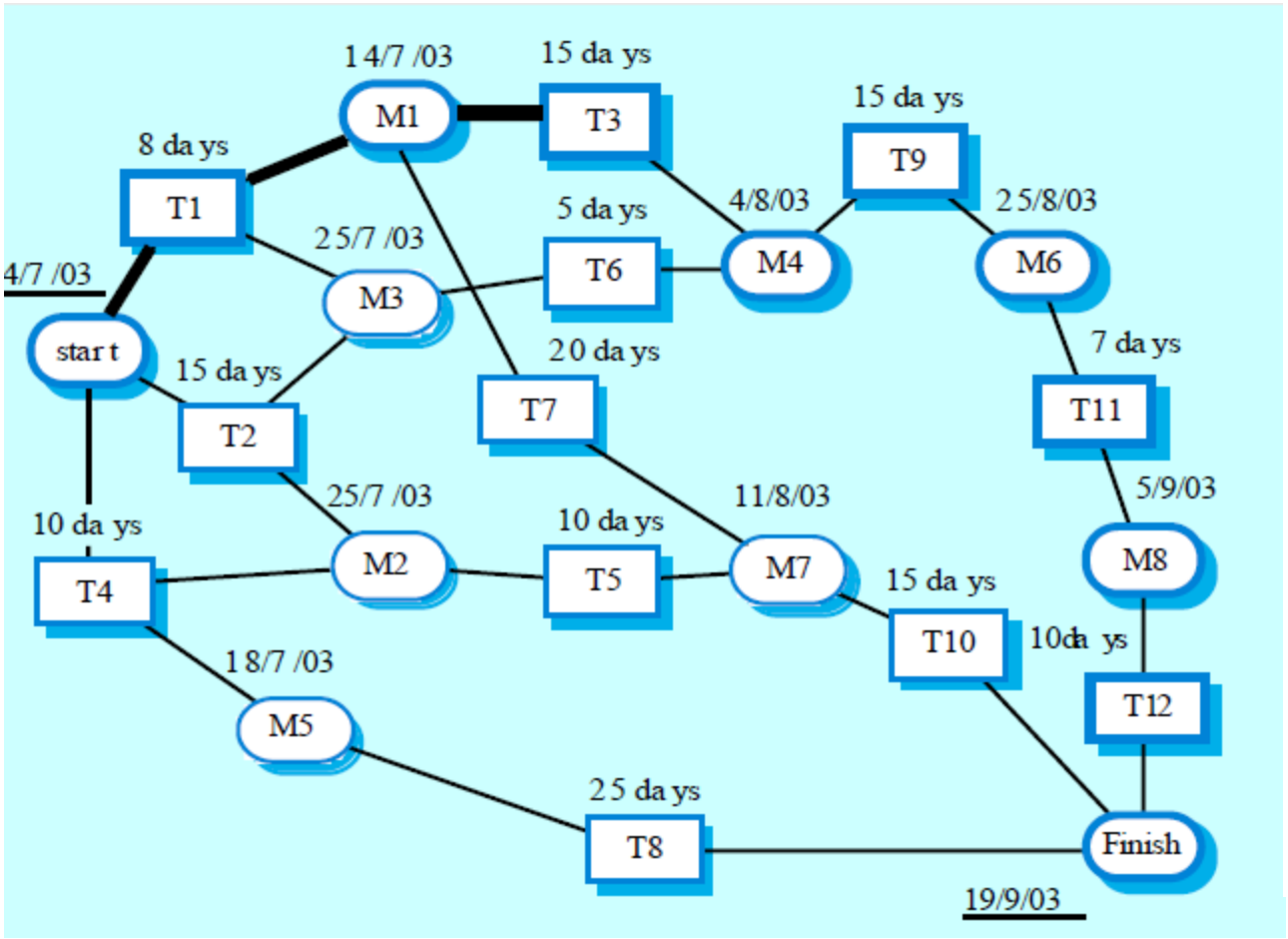
Bar charts and activity networks

- Graphical notations used to **illustrate the project schedule.**
- Show project **breakdown into tasks**. Tasks should not be too small. They should take about a week or two.
- Activity charts **show task dependencies and the critical path.**
- **Bar charts show schedule against calendar time.**

Task durations and dependencies

Activity	Duration (days)	Dependencies
T1	8	
T2	15	
T3	15	T1 (M1)
T4	10	
T5	10	T2, T4 (M2)
T6	5	T1, T2 (M3)
T7	20	T1 (M1)
T8	25	T4 (M5)
T9	15	T3, T6 (M4)
T10	15	T5, T7 (M7)
T11	7	T9 (M6)
T12	10	T11 (M8)

Activity network



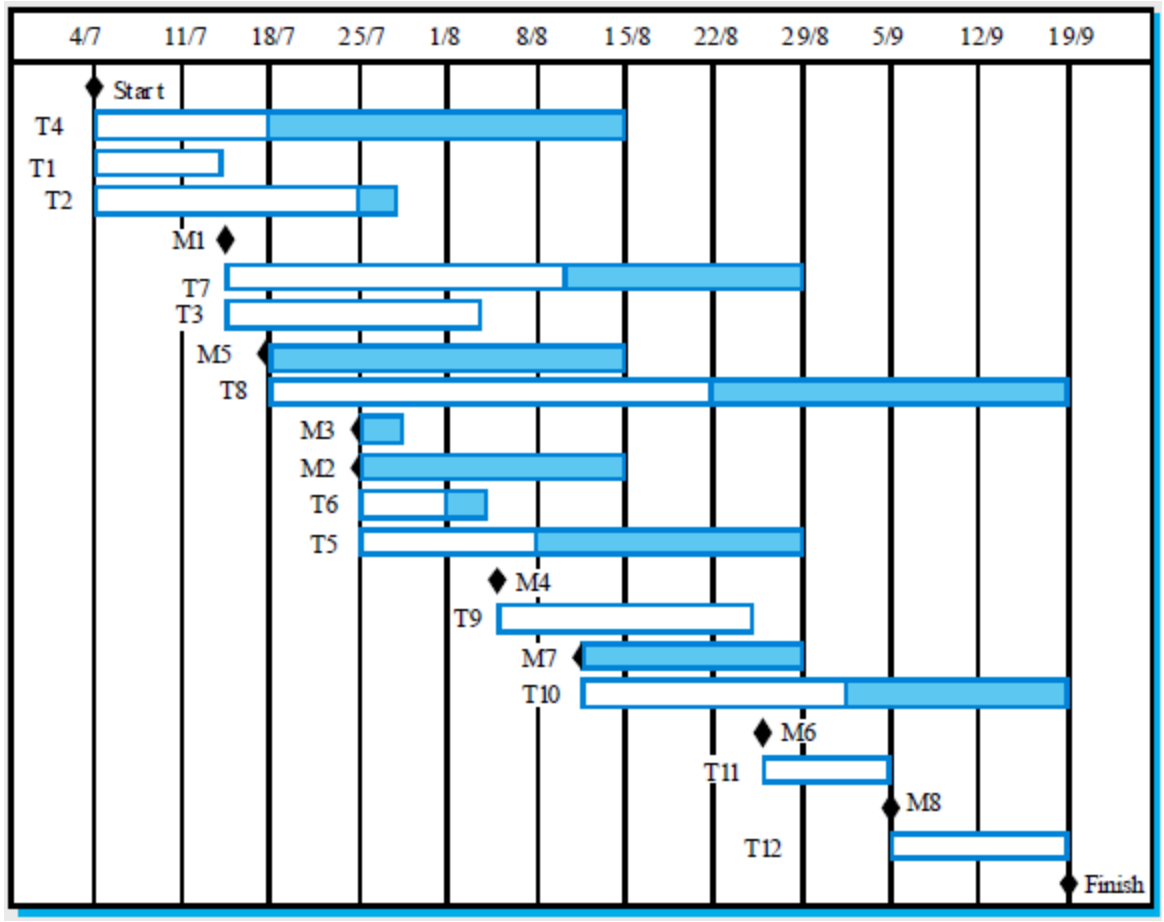
Critical path

Minimum time to finish project = longest path

T8 delayed 2 weeks = not affect on CP

Managers used to allocate project work

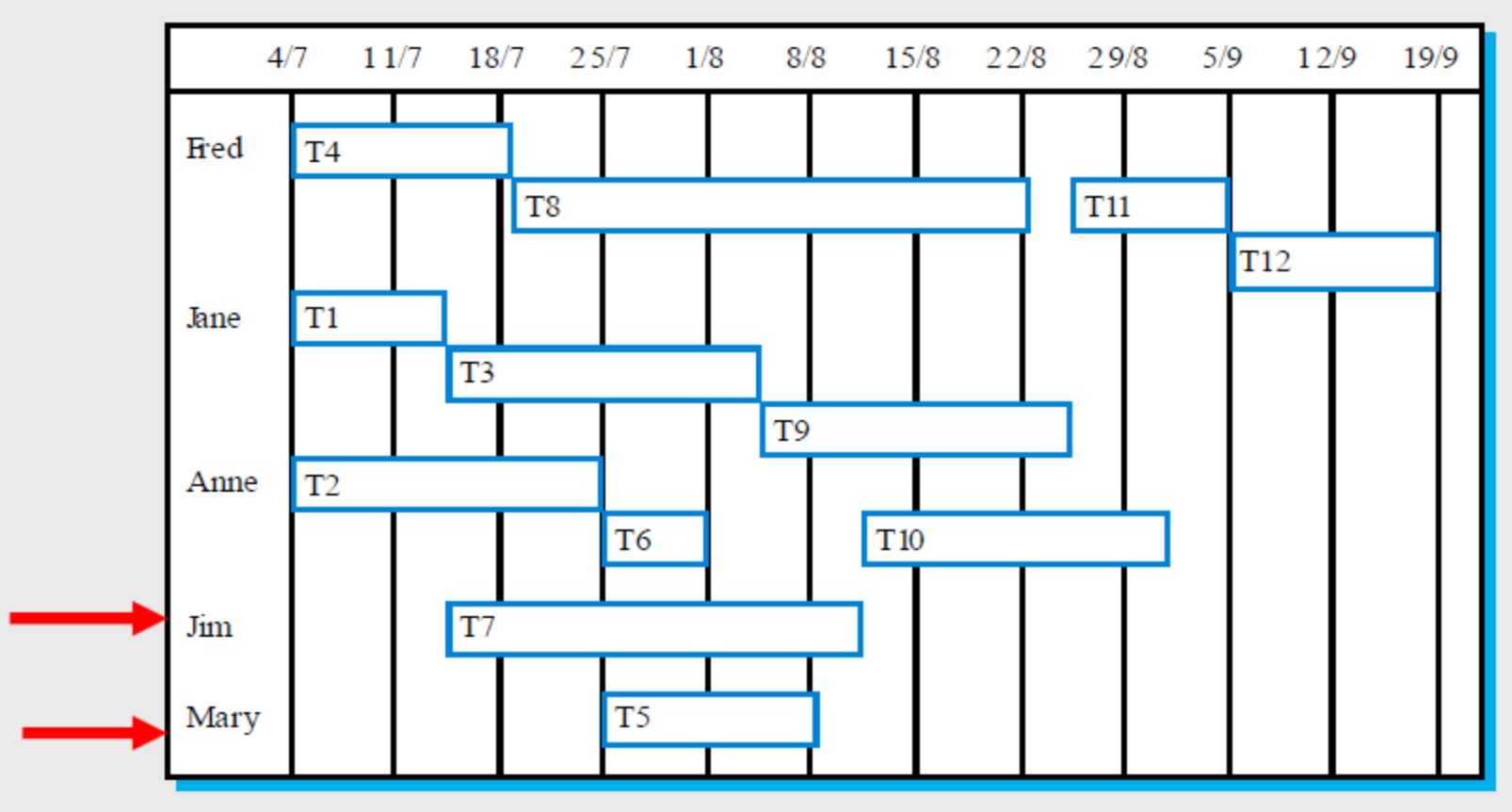
Activity timeline (Bar Chart), (Gantt chart)



Shaded bar = incomplete activities on time = doesn't affect on CP

A on CP = no errors = no delay

Staff allocation



Attention

Q:What is the critical distinction between a milestone and a deliverable?

Task	Duration (days)	Dependencies
T1	15	
T2	20	T1
T3	14	T1, T2 (M1)
T4	15	T3(M2)
T5	25	T4(M3)