Fundamental of Software Engineering
CSE 3205

Chapter Six

Software Project Management

School of Electrical Engineering and Computing

Department of Computer Science and Engineering

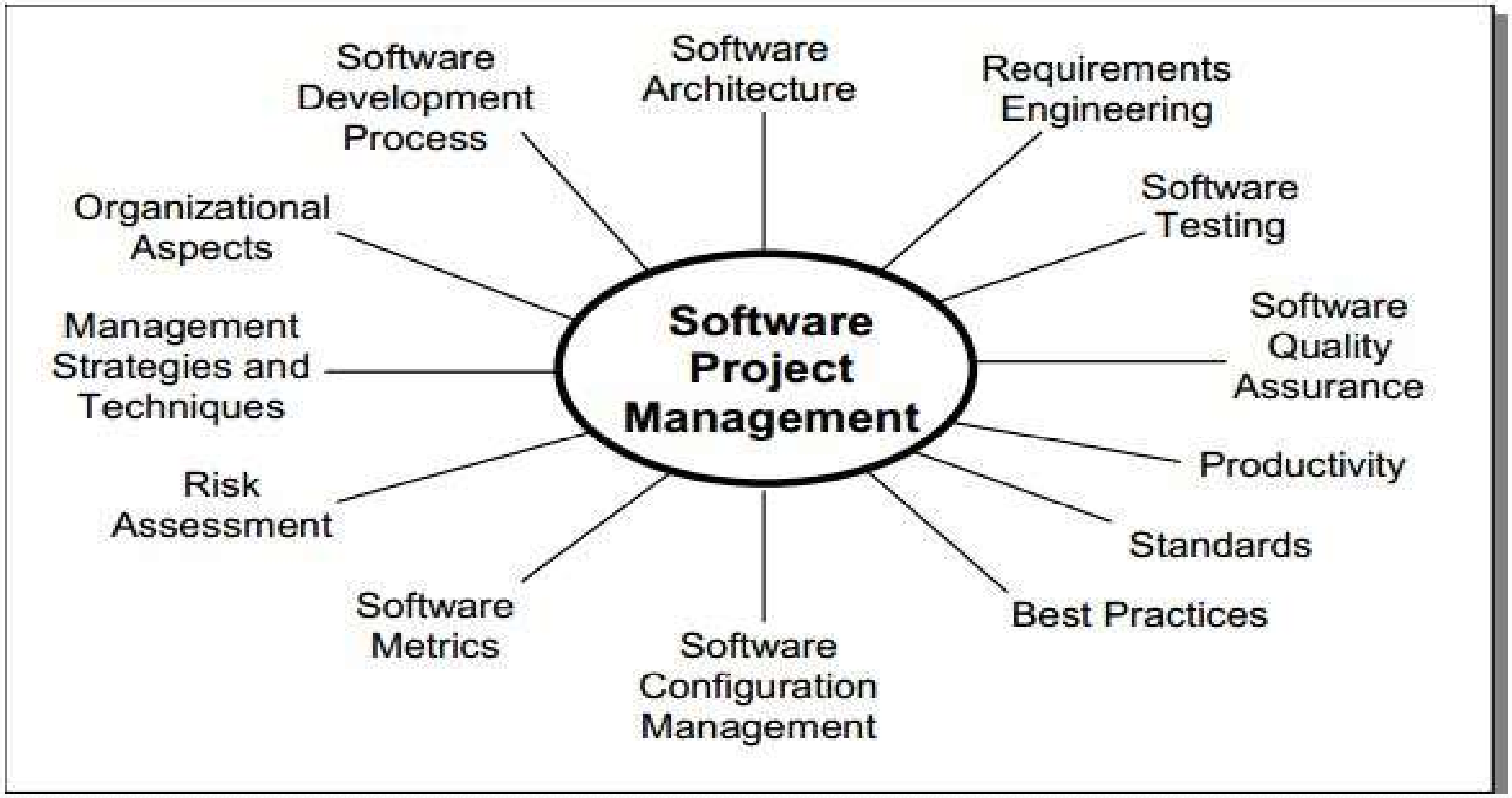
ASTU

February 9, 2024

Software Project Management

- Concerned with activities involved in ensuring that software is delivered on time and on schedule and in accordance with the requirements of the organisations developing and procuring the software.
- Project management is needed because software development is always subject to budget and schedule constraints that are set by the organisation developing the software.

Cont..


Software Project Management
Software Development Process
Software Architecture
Requirements Engineering
Software Testing
Software Quality Assurance
Productivity
Standards
Best Practices
Software Configuration Management
Software Metrics
Risk Assessment
Management Strategies and Techniques
Organizational Aspects

Project Management Skills

- Leadership
- Communications
- Problem Solving
- Negotiating
- Influencing the Organization
- Mentoring
- Process and technical expertise

The Four P’s in Management

- People — the most important element of a successful project
- Product — the software to be built
- Process — the set of framework activities and software engineering tasks to get the job done
- Project — all work required to make the product a reality

Stakeholders in project management

- *Senior managers* who define the business issues that often have significant influence on the project.
- *Project (technical) managers* who must plan, motivate, organize, and control the practitioners who do software work.
- *Practitioners* who deliver the technical skills that are necessary to engineer a product or application.
- *Customers* who specify the requirements for the software to be engineered and other stakeholders who have a peripheral interest in the outcome.
- *End-users* who interact with the software once it is released for production use.

Software Teams

How to lead?

How to organize?

How to collaborate?

How to motivate?

How to create good ideas?

Role of Team Leader

- The MOI Model
  - **Motivation**. The ability to encourage (by “push or pull”) technical people to produce to their best ability.
  - **Organization**. The ability to mold existing processes (or invent new ones) that will enable the initial concept to be translated into a final product.
  - **Ideas or innovation**. The ability to encourage people to create and feel creative even when they must work within bounds established for a particular software product or application.

Software Teams

The following factors must be considered when selecting a software project team structure ...

- the difficulty of the problem to be solved
- the size of the resultant program(s) in lines of code or function points
- the time that the team will stay together (team lifetime)
- the degree to which the problem can be modularized
- the required quality and reliability of the system to be built
- the rigidity of the delivery date
- the degree of sociability (communication) required for the project

Organizational Paradigms

- **closed paradigm**—structures a team along a traditional hierarchy of authority
- **random paradigm**—structures a team loosely and depends on individual initiative of the team members
- **open paradigm**—attempts to structure a team in a manner that achieves some of the controls associated with the closed paradigm but also much of the innovation that occurs when using the random paradigm
- **synchronous paradigm**—relies on the natural compartmentalization of a problem and organizes team members to work on pieces of the problem with little active communication among themselves

suggested by Constantine [Con93]

Software Management Distinctions

- The product is intangible.
- The product is uniquely flexible.
- Software engineering is not recognized as an engineering discipline with the same status as mechanical, electrical engineering, etc.

Cont.

- We don't have much experience.
  - Software engineering is a new discipline, and so we simply don't have much understanding of how to engineer large scale software projects.
- Large software projects are often "bespoke".
  - Most large software systems are one-off, with experience gained in one project being of little help in another.
- The technology changes very quickly.
  - Most large software projects employ new technology; for many projects,

Activities in software project management:

1) Project Planning
2) Project Scheduling
3) Risk Management
4) Managing people

1. Project Planning

- The biggest single problem that afflicts software developing is that of underestimating resources required for a project.
- Developing a realistic project plan is essential to gain an understanding of the resources required, and how these should be applied.

1.1 Types of plan:

- Software development plan.
  - The central plan, which describes how the system will be developed.
- Quality assurance plan.
  - Specifies the quality procedures & standards to be used.
- Validation plan.
  - Defines how a client will validate the system that has been developed.

Cont.

- Configuration management plan.
  - Defines how the system will be configured and installed.
- Maintenance plan.
  - Defines how the system will be maintained.
- Staff development plan.
  - Describes how the skills of the participants will be developed.

Project planning process

```
Establish the project constraints
Make initial assessments of the project parameters
Define project milestones and deliverables
while project has not been completed or cancelled loop
        Draw up project schedule
        Initiate activities according to schedule
        Wait ( for a while )
        Review project progress
        Revise estimates of project parameters
        Update the project schedule
        Re-negotiate project constraints and deliverables
        if ( problems arise ) then
                Initiate technical review and possible revision
        end if
end loop
```

1.2 The Software Development Plan

- This is usually what is meant by a project plan.
- Specifies the order of work to be carried out, resources, responsibilities, and so on.
- Varies from small and relatively informal to large and very formal.
- Developing a project plan is as important as properly designing code:
  - On the basis of a project plan, contracts will be signed and careers made or broken. . .
- Important not to:
  - overestimate your team's ability;
  - simply tell clients what they want to hear;
  - be pressured by developers ("we can do that in an afternoon!")

1.2.1 Structure of Development Plan

1. Introduction
    - brief intro to project — references to requirements spec
2. Project organization
    - intro to organizations, people, and their roles
3. Risk Analysis
    - what are the key risks to the project?

Cont.

4. Hardware and software resources
   - what h/ware and s/ware resources will be required for the project and when?
5. Work breakdown
   - the project divided into activities , milestones, deliverables; dependencies between tasks etc
6. Project schedule
   - actual time required — allocation of dates
7. Reporting and progress measurement : mechanisms to monitor progress.

1.2.2 Work Breakdown

- There are many ways of breaking down the activities in a project, but the most usual is into:
  - work packages;
  - tasks;
  - deliverables;
  - milestones.

Cont.

- A workpackage is a large, logically distinct section of work:
- typically at least 12 months duration;
- may include multiple concurrent activities;
- independent of other activities;
- but may depend on, or feed into other activities;
- typically allocated to a single team.

Cont.

- **task** is typically a much smaller piece of work:A part of a workpackage.
- typically 3–6 person months effort;
- may be dependent on other concurrent activities;
- typically allocated to a single person.

Cont.

A deliverable is an output of the project that can meaningfully be assessed.

-is a measurable and tangible outcome of the project

Examples: – a report (e.g., requirements spec);

code (e.g., alpha tested product).

- Deliverables are indicators (but only indicators) of progress.

A milestone is a point at which progress on the project may be assessed.

-are checkpoints throughout the life of the project

- Typically a major turning point in the project.

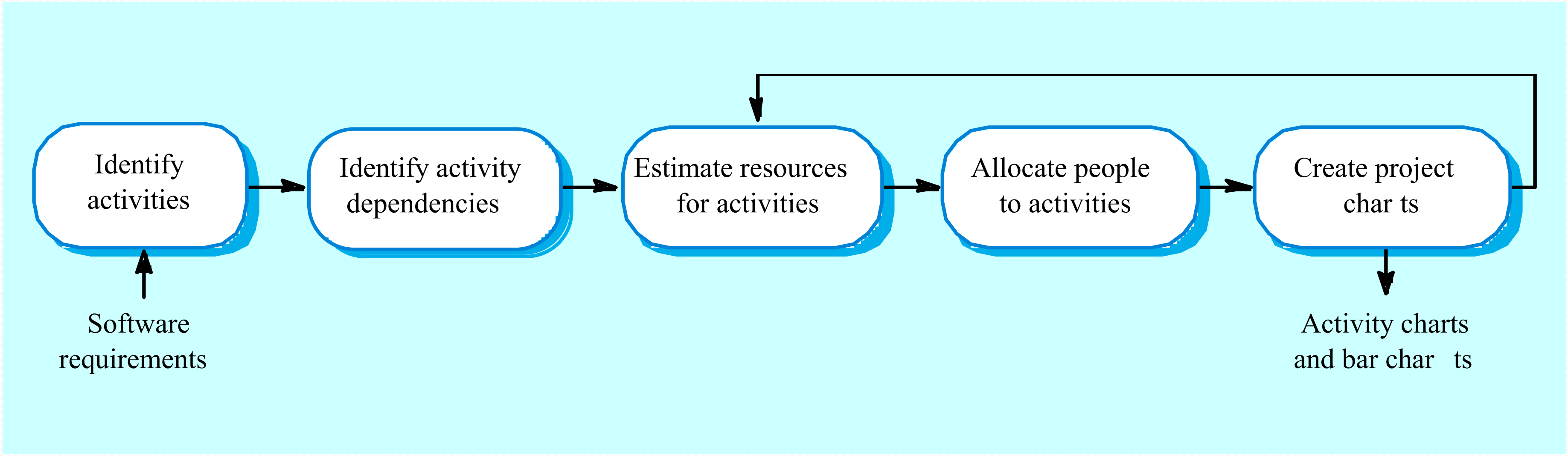
EXAMPLES: – delivery of requirements spec;

delivery of alpha tested code.

2.Project Scheduling

- Split project into tasks and estimate time and resources required to complete each task.
- Organize tasks concurrently to make optimal use of workforce.
- Minimize task dependencies to avoid delays caused by one task waiting for another to complete.
- Dependent on project managers intuition and experience.

2.1 The project scheduling process


Identify activities
Identify activity dependencies
Estimate resources for activities
Allocate people to activities
Create project char ts
Software requirements
Activity charts and bar char ts

2.2 Scheduling problems

- Estimating the difficulty of problems and hence the cost of developing a solution is hard.
- Productivity is not proportional to the number of people working on a task.
- Adding people to a late project makes it later because of communication overheads.
- The unexpected always happens. Always allow contingency in planning.

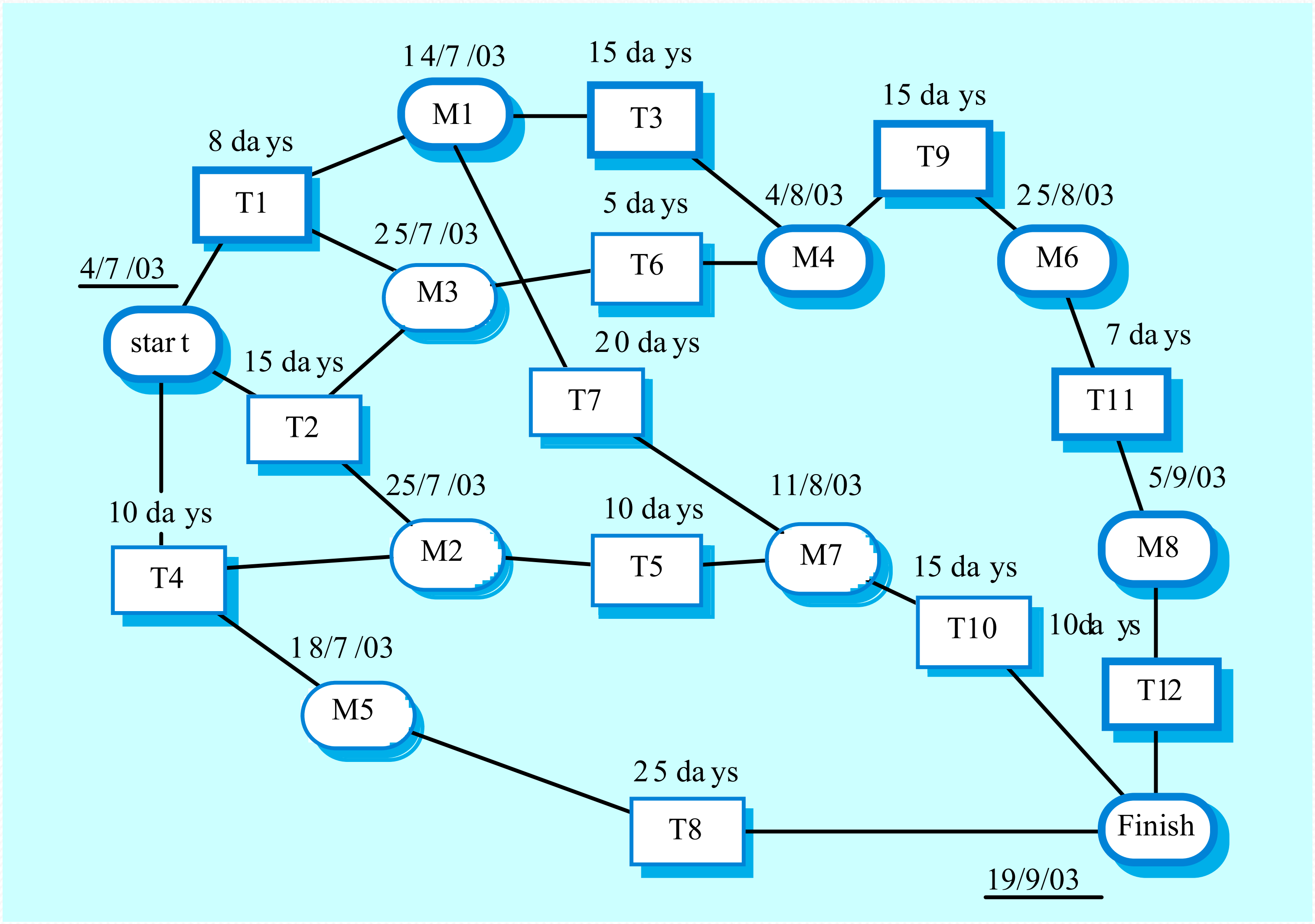
2.3 Bar charts and activity networks

- 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.

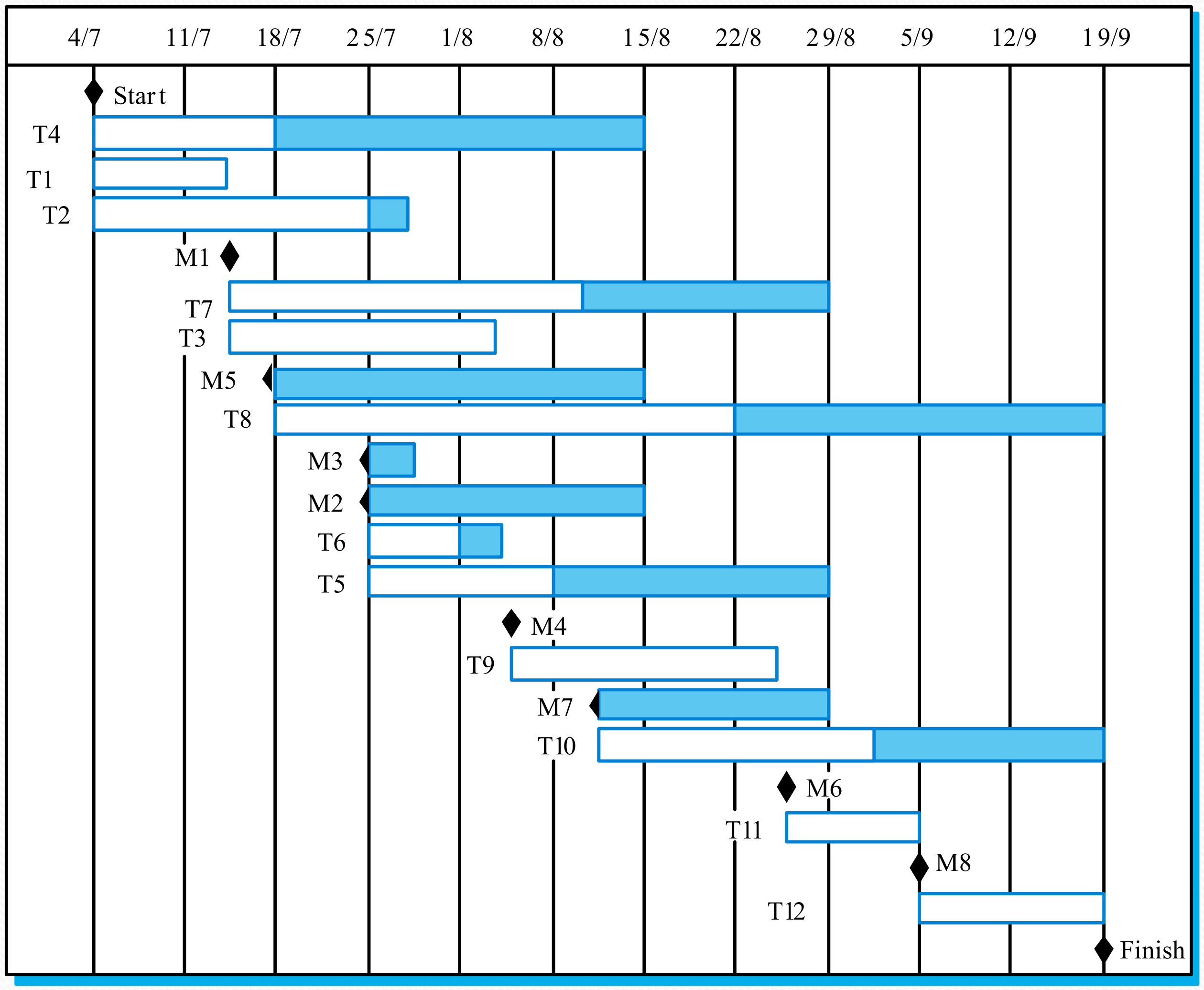
Task durations and dependencies

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)

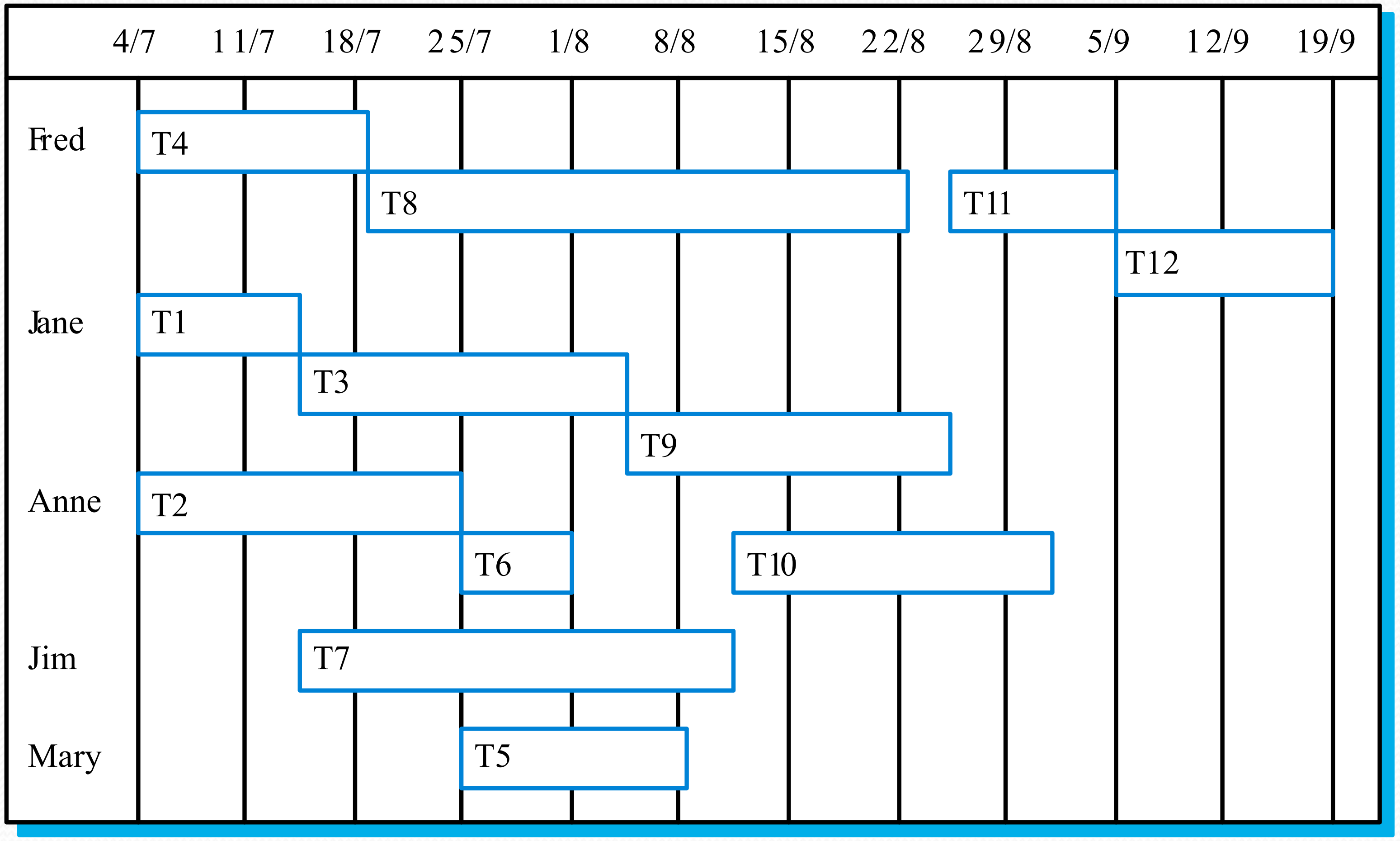
Activity network



Activity timeline



Staff allocation



3. Risk Management

- Risk management is concerned with identifying risks and drawing up plans to minimise their effect on a project.
- A risk is a probability that some adverse circumstance will occur
  - Project risks affect schedule or resources;
  - Product risks affect the quality or performance of the software being developed;
  - Business risks affect the organisation developing or procuring the software.

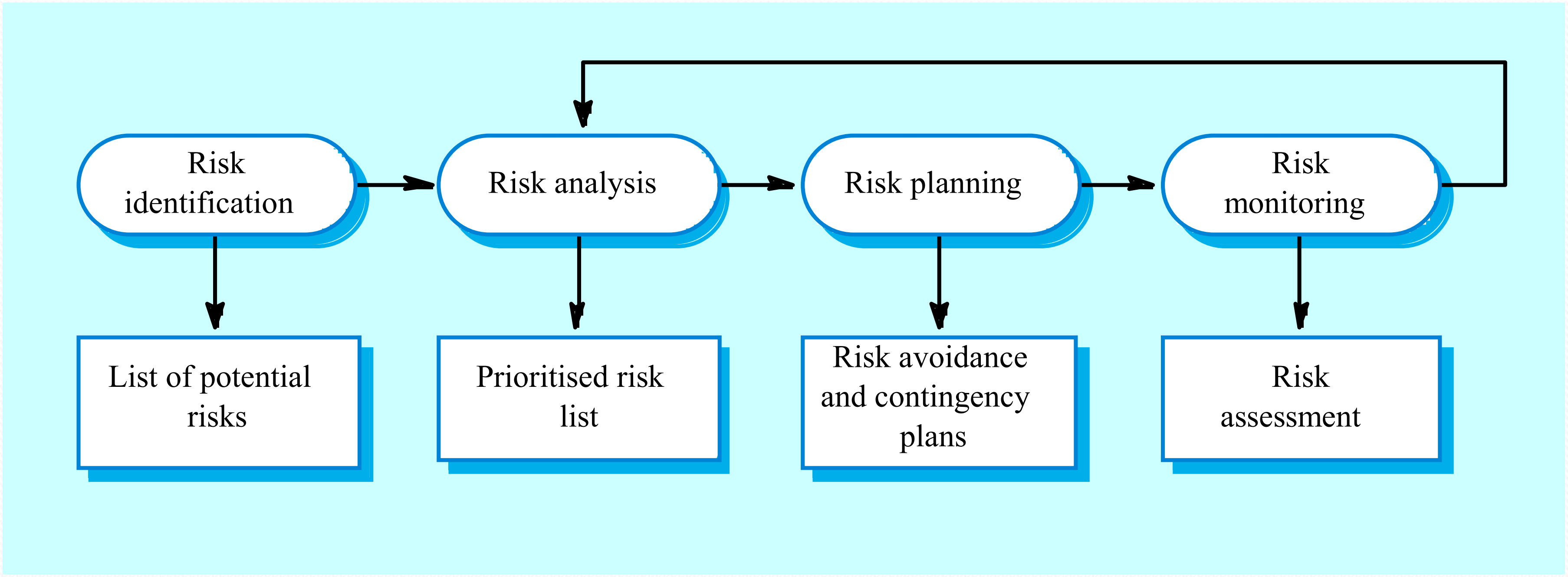
3.1 Software risks

Risk	Affects	Description
Staff turnover	Project	Experienced staff will leave the project before it is finished.
Management change	Project	There will be a change of organisational management with different priorities.
Hardware unavailability	Project	Hardware that is essential for the project will not be delivered on schedule.
Requirements change	Project and product	There will be a larger number of changes to the requirements than anticipated.
Specification delays	Project and product	Specifications of essential interfaces are not available on schedule
Size underestimate	Project and product	The size of the system has been underestimated.
CASE tool under-performance	Product	CASE tools which support the project do not perform as anticipated
Technology change	Business	The underlying technology on which the system is built is superseded by new technology.
Product competition	Business	A competitive product is marketed before the system is completed.

3.2 The risk management process

- Risk identification
  - Identify project, product and business risks;
- Risk analysis
  - Assess the likelihood and consequences of these risks;
- Risk planning
  - Draw up plans to avoid or minimise the effects of the risk;
- Risk monitoring
  - Monitor the risks throughout the project;

3.3 The risk management process



3.3.1 Risk identification

- Technology risks.
- People risks.
- Organisational risks.
- Requirements risks.
- Estimation risks.

Risks and Risk types

Risk type	Possible risks
Technology	The database used in the system cannot process as many transactions per second as expected. Software components that should be reused contain defects that limit their functionality.
People	It is impossible to recruit staff with the skills required. Key staff are ill and unavailable at critical times. Required training for staff is not available.
Organisational	The organisation is restructured so that different management are responsible for the project. Organisational financial problems force reductions in the project budget.
Tools	The code generated by CASE tools is inefficient. CASE tools cannot be integrated.
Requirements	Changes to requirements that require major design rework are proposed. Customers fail to understand the impact of requirements changes.
Estimation	The time required to develop the software is underestimated. The rate of defect repair is underestimated. The size of the software is underestimated.

3.3.2 Risk analysis

- Assess probability and seriousness of each risk.
- Probability may be very low, low, moderate, high or very high.
- Risk effects might be catastrophic, serious, tolerable or insignificant.

Risk analysis (i)

Risk	Probability	Effects
Organisational financial problems force reductions in the project budget.	Low	Catastrophic
It is impossible to recruit staff with the skills required for the project.	High	Catastrophic
Key staff are ill at critical times in the project.	Moderate	Serious
Software components that should be reused contain defects which limit their functionality.	Moderate	Serious
Changes to requirements that require major design rework are proposed.	Moderate	Serious
The organisation is restructured so that different management are responsible for the project.	High	Serious

Risk analysis (ii)

Risk	Probability	Effects
The database used in the system cannot process as many transactions per second as expec ted.	Moderate	Serious
The time required to deve lop the software is underestimated.	High	Serious
CASE tools cannot be integrated.	High	Tolerable
Customers fail to understand the impact of requirements changes.	Moderate	Tolerable
Required training for staff is not available.	Moderate	Tolerable
The rate of defect repair is underestimated.	Moderate	Tolerable
The size of the software is underestimated.	High	Tolerable
The code generated by CASE tools is inefficient.	Moderate	Insignificant

3.3.3 Risk planning

- Consider each risk and develop a strategy to manage that risk.
- Avoidance strategies
  - The probability that the risk will arise is reduced;
- Minimisation strategies
  - The impact of the risk on the project or product will be reduced;
- Contingency plans
  - If the risk arises, contingency plans are plans to deal with that risk;

3.3.4 Risk management strategies (i)

Risk	Strategy
Organisational financial problems	Prepare a briefing document for senior management showing how th e project is making a very important contribution to the goals of the business.
Recruitment problems	Alert customer of potential difficulties and the possibility of delays, investigate buying-in components.
Staff illness	Reorganise team so that there is more overlap of work and people therefore understand each other's jobs.
Defective components	Replace potentially defective components with bought-in components of known reliability.

Risk management strategies (ii)

Risk	**Strategy**
Requirements changes	Derive traceability information to assess requirements change impact, maximise information hiding in the design.
Organisational restructuring	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business.
Database performance	Investigate the possibility of buying a higher-performance database.
Underestimated development time	Investigate buying in components, investigate use of a program generator

3.3.5 Risk monitoring

- Assess each identified risks regularly to decide whether or not it is becoming less or more probable.
- Also assess whether the effects of the risk have changed.
- Each key risk should be discussed at management progress meetings.

3.3.6 Risk Indicators

Risk type	**Potential indicators**
Technology	Late delivery of hardware or support software, many reported technology problems
People	Poor staff morale, poor relationships amongst team member, job availability
Organisational	Organisational gossip, lack of action by senior management
Tools	Reluctance by team members to use tools, complaints about CASE tools, demands for higher-powered workstations
Requirements	Many requirements change requests, customer complaints
Estimation	Failure to meet agreed schedule, failure to clear reported defects

4. Project Size Estimation Metrics

- Size is one of the most important attributes of a software product

CODE-BASED SIZING METRICS

- Code-based sizing metrics measure the size or complexity of software using the programmed source code. Because a significant amount of effort is devoted to programming, it is believed that an appropriate measure correctly quantifying the code can be a perceivable indicator of software cost.

Cont..

- source lines of code (SLOC) among others have been proposed and used as code-based sizing metrics.
- SLOC is the most popular. It is used as a primary input by most major cost estimation models, such as SLIM, SEER-SEM, PRICE-S,COCOMO, and Knowledge Plan

Cont….

- The Constructive Cost Model (COCOMO) is a formula-based method for estimating the total cost of developing software .
- The fundamental input to COCOMO is the estimated number of lines of source code.

Cont.

- Function Point Analysis(FPA) estimates the cost of a project from the estimates of the delivered functionality
- FPA can therefore be applied at the end of the software requirements definition phase to make cost estimates.

Key points

- Good project management is essential for project success.
- The intangible nature of software causes problems for management.
- Managers have diverse roles but their most significant activities are planning, estimating and scheduling.
- Planning and estimating are iterative processes which continue throughout the course of a project.

Key points

- A project milestone is a predictable state where a formal report of progress is presented to management.
- Project scheduling involves preparing various graphical representations showing project activities, their durations and staffing.
- Risk management is concerned with identifying risks which may affect the project and planning to ensure that these risks do not develop into major threats.

Thank You!

Q?