EBU6304 – Software Engineering

Project Management

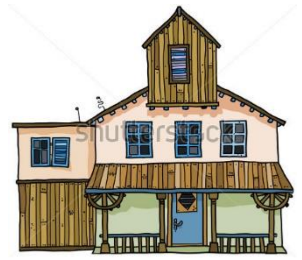
Topics:

- Activities
- Project Planning
- Project Scheduling
- Managing People
- Agile project management

The need

- Software project management is to ensure the software is delivered
  - On time
  - Within budget
  - With Quality
- Good management cannot guarantee project success, but bad management usually results in project failure.
- Project manager is responsible for planning and scheduling project, monitoring progress.

Software Projects vs Other Projects



Software Projects’ Distinctions

- Software products are intangible and flexible.
- Software engineering is not recognised as a sane engineering discipline
  - not standardised
- Many software projects are 'one-off' projects
  - Technologies are changing too fast.
  - Previous experiences may be obsolete.
  - Require a perceptive insight.

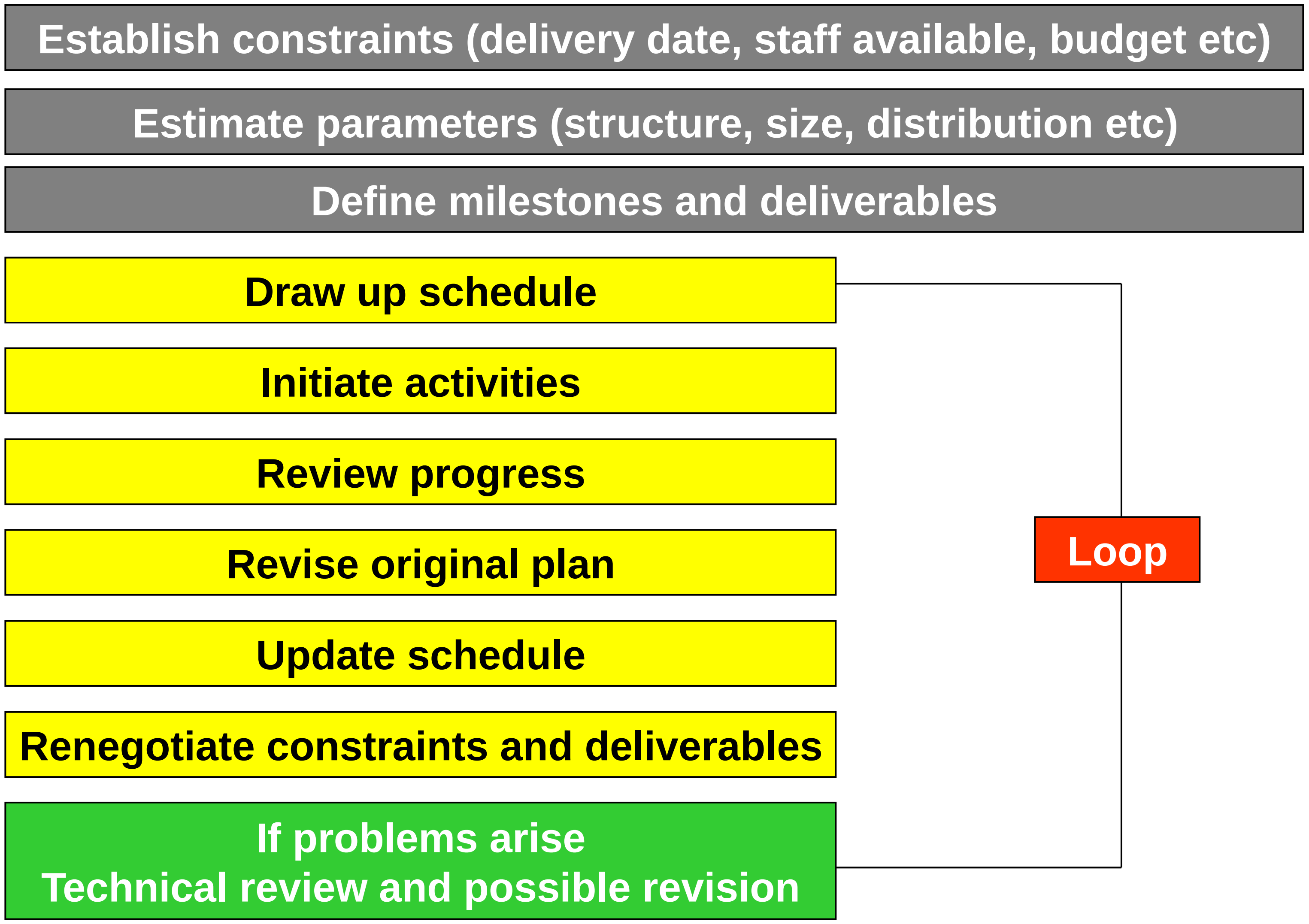
Project Management activities

- Proposal writing
- Project planning
- Project costing
- Time management
- Project monitoring and reviews
- Personnel selection and evaluation
- Report writing and presentations
- Risk management
- Quality management

Project planning

- To make effective management
  - Drawn up at the start of the project.
  - Probably the most time-consuming activity.
- Iterative process
  - The plan is only complete when the project itself is complete.
  - Must be regularly revised.
- Various different types of plan may be developed
  - *Quality plan*, *staff development plan*, … etc

Project planning process


Establish constraints (delivery date, staff available, budget etc)

Estimate parameters (structure, size, distribution etc)

Define milestones and deliverables

Draw up schedule

Initiate activities

Review progress

Revise original plan

Update schedule

Renegotiate constraints and deliverables

If problems arise
Technical review and possible revision

Types of Project Plan

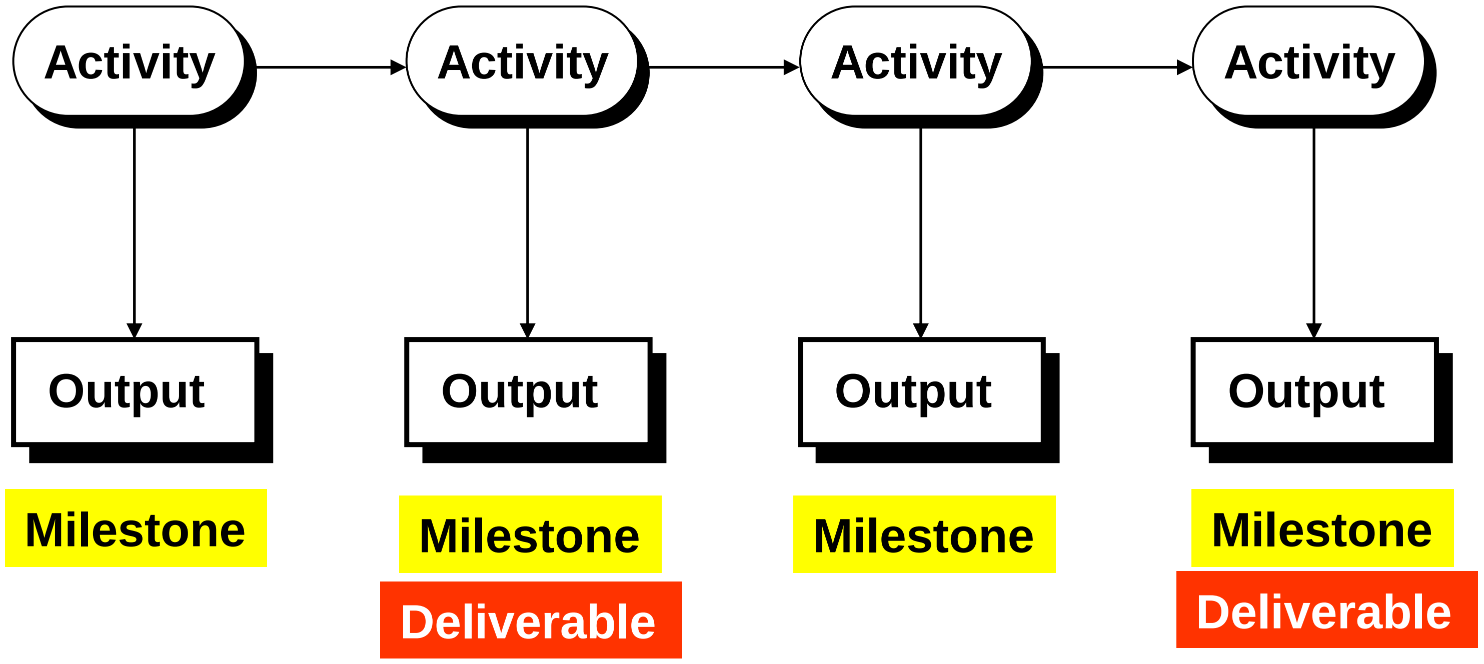
Plan	**Description**
Quality plan	Describes the quality procedures and standards that will be used in a project.
Validation plan	Describes the approach, resources and schedule used for system validation.
Configuration management plan	Describes the configuration management procedures and structures to be used.
Maintenance plan	Predicts the maintenance requirements of the system, maintenance costs and effort required.
Staff development plan.	Describes how the skills and experience of the project team members will be developed.

Activity organisation

- Activities in a project should be organised to
  - Produce tangible outputs
  - Judge progress
- Milestones are the recognisable end-points of a process activity
  - Formal output: reports, documentations
  - Definite: the end of a distinct, logical stage
- Deliverables are project results delivered to customers
  - Usually delivered at the end of some major phase: *specification*, *design* ... etc
- Deliverables *are usually* milestones, but milestones *need not be* deliverables!

Establishing milestones

- To establish milestones, the software process must be broken down into basic activities with associated outputs.
  - Using a diagram is a good way to represent milestones.



Project scheduling

- Estimate time and resources required to complete activities and organise them into a coherent sequence.
- Complicated
  - Previous estimates are uncertain
  - Different design method and implementation language
  - Especially for technically advanced projects

Project scheduling process

- Split project into separate tasks → estimate time and resources required to complete each task.
- Organise tasks concurrently → to make optimal use of workforce.
- Minimise task dependencies → to avoid delays caused by one task waiting for another to complete.
- *Dependent on project managers' intuition and experience!*

Scheduling problems

- Do not assume that every stage will be problem free; the unexpected always happens, always allow for a contingency in planning
  - People, hardware, software
- A good rule
  - Estimate as if nothing will go wrong.
  - Add 30% for anticipated problems.
  - Add further 20% to cover unanticipated problems.

Charts

- Project schedules are usually represented as a set of charts
  - *Work breakdown*, *activities' dependencies*, *staff allocations*

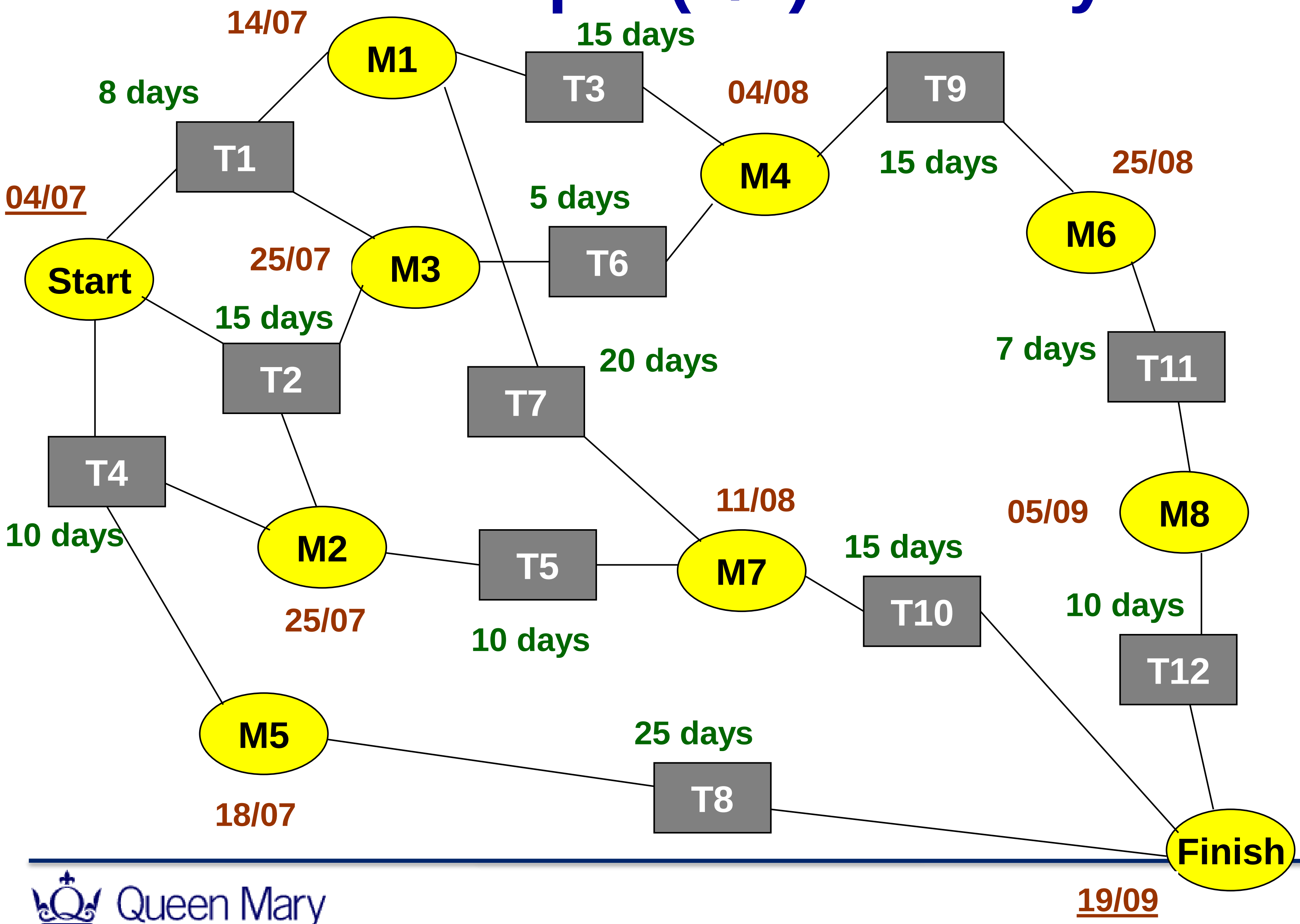
- Charts are graphical notations used to illustrate the project schedule
  - Task chart: show project breakdown *into* tasks; tasks should not be too small.
  - Activity network: show task dependencies *and the* critical path.
  - Bar chart: show schedule *against* calendar time.

Example: Task chart

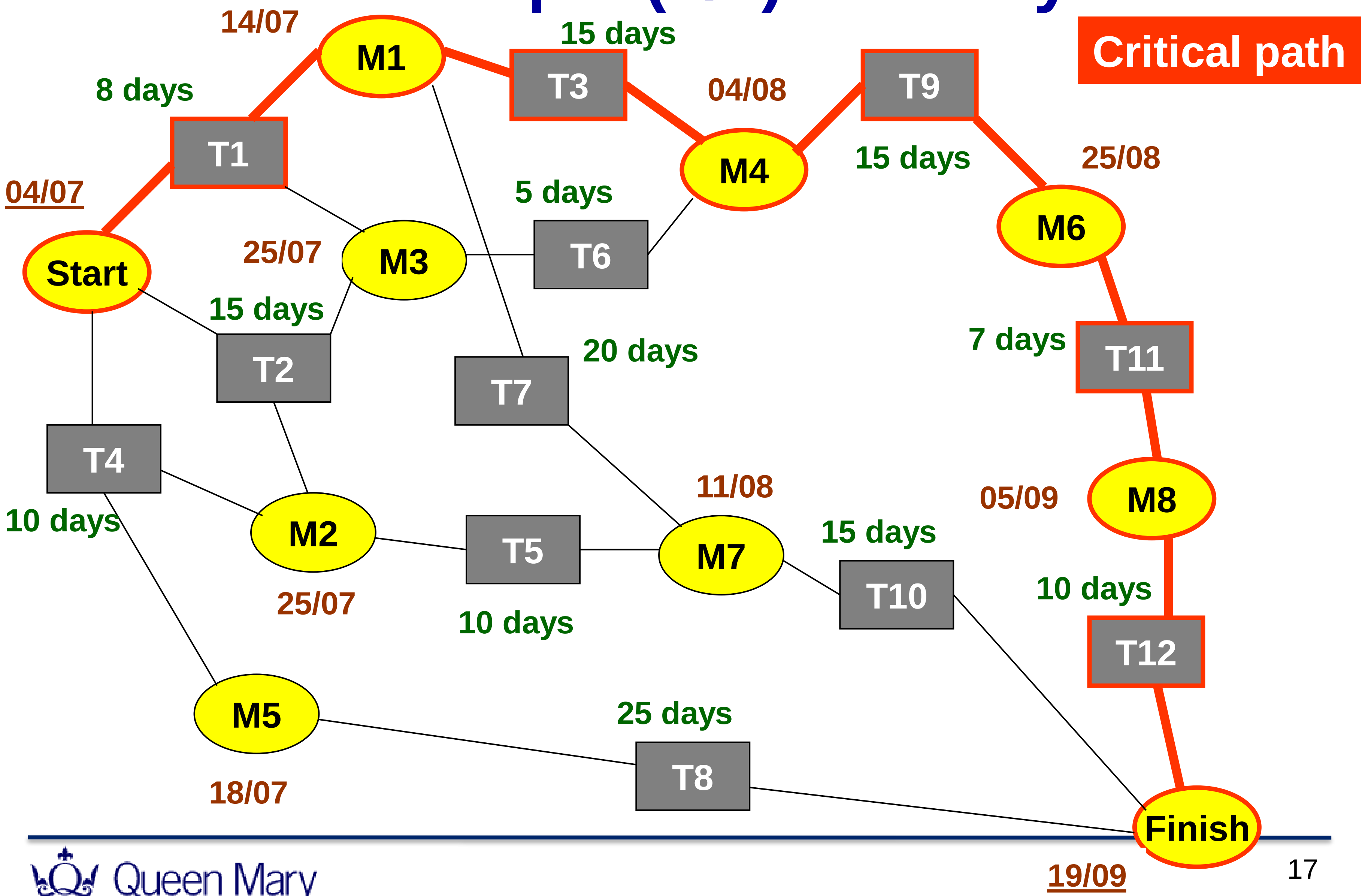
Activity	Duration (days)	Dependencies
T1	8	
T2	15	
T3	15	T1 (M1)
T4	10	
T5	10	T2, T4 (M2)
T6	5	T1, T2 (M3)
T7	20	T1 (M1)
T8	25	T4 (M5)
T9	15	T3, T6 (M4)
T10	15	T5, T7 (M7)
T11	7	T9 (M6)
T12	10	T11 (M8)

T: Task
M: Milestone

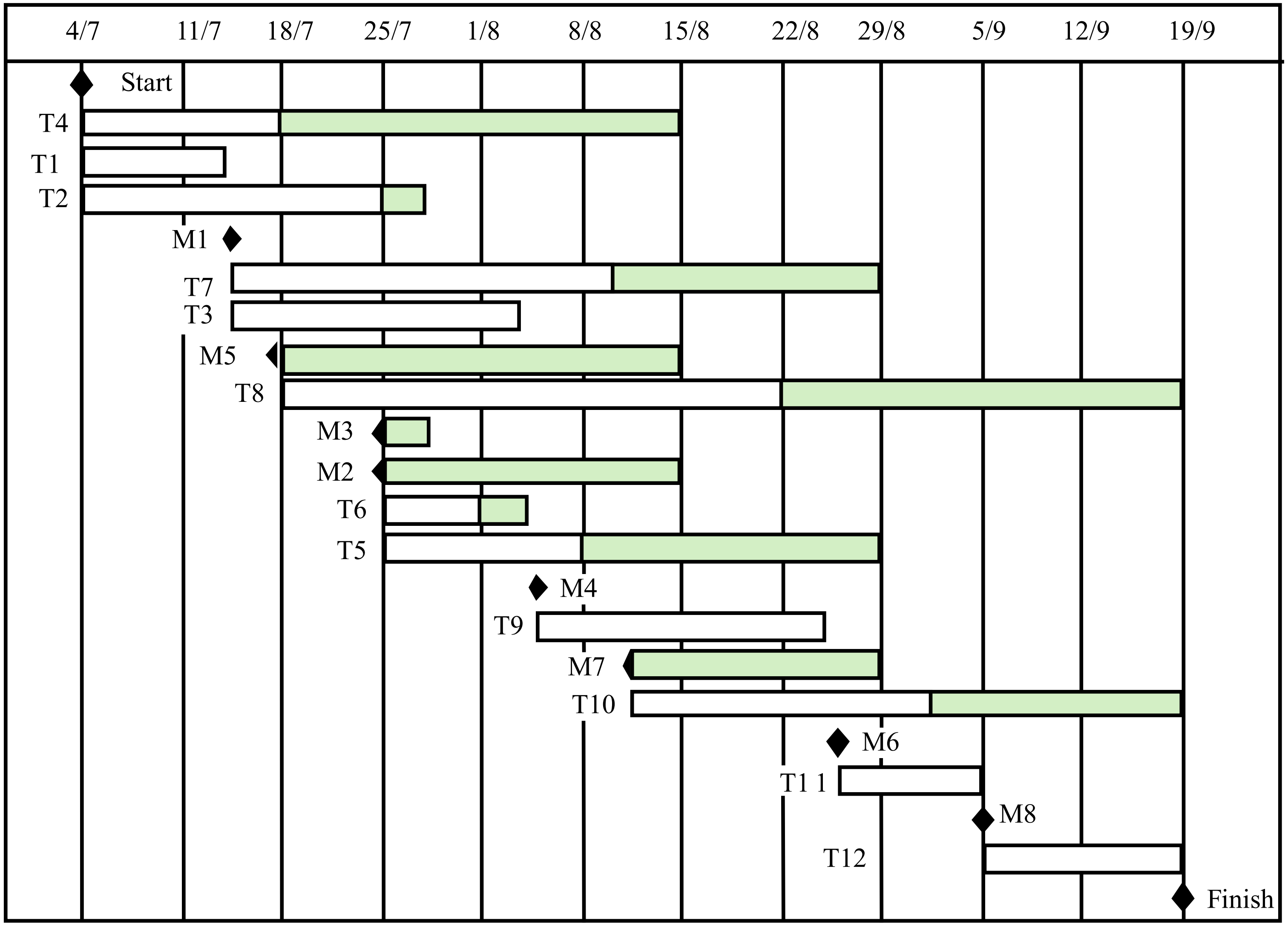
Example (1/2): Activity network



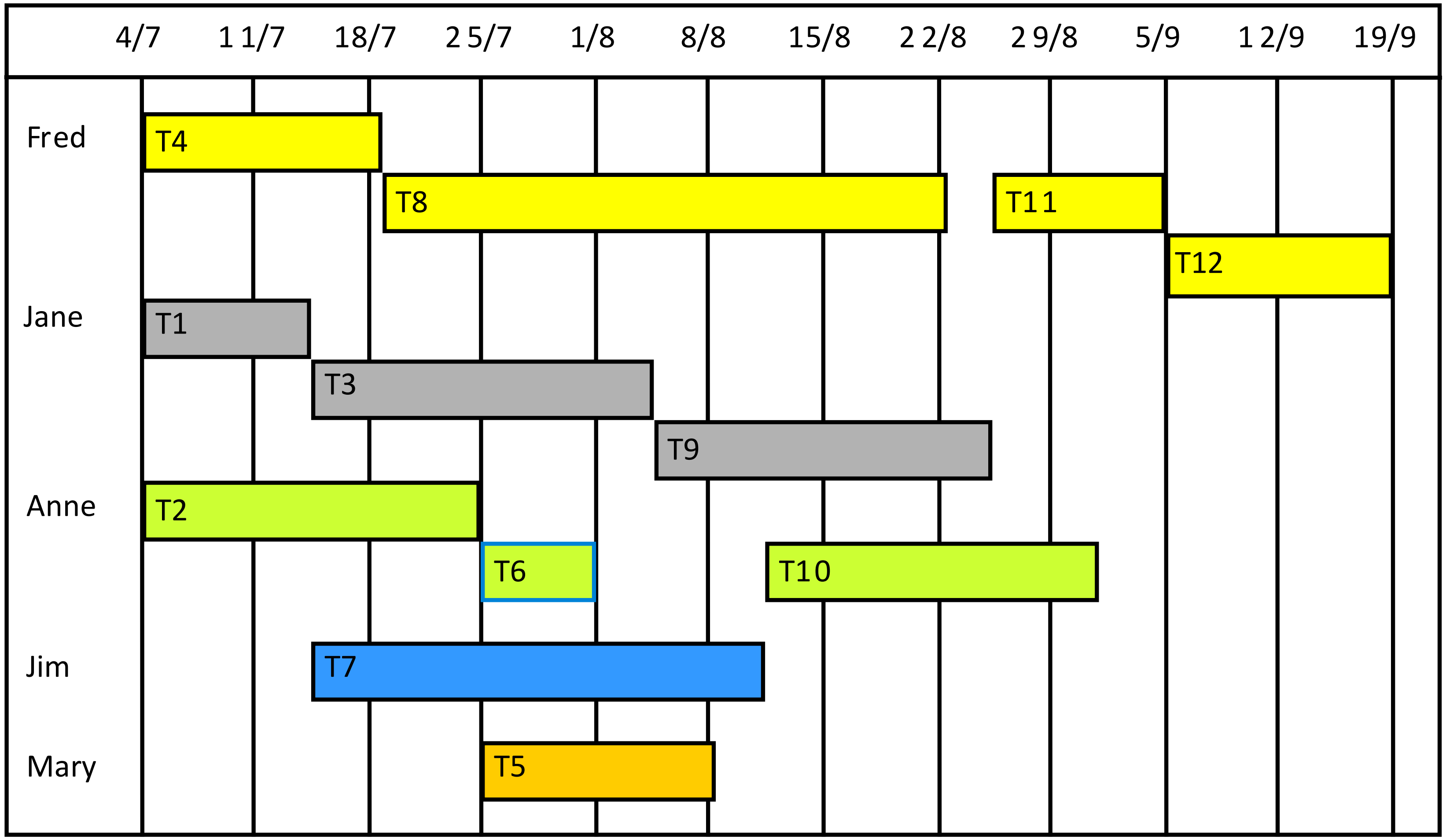
Example (2/2): Activity network



Example: Bar chart (*Gantt Chart*)



Staff allocation



Monitoring/Reporting

- Project scheduling and project budgeting is often based on prior project experiences and on various product and process measurements (metrics) collected in the past.
- Daily stand up
- Weekly Reports, Weekly Meetings
- Customer Meetings
- Demos

Metrics

- Examples of metrics or measurements include:
  - Number of lines of code
  - Number of defects in code (in particular at defined points in development)
  - Test cases completed and in what time frame
  - Test cases passed / failed
- Metrics on a 'live' project are also used to *assess progress of development against project plans*.

People in the process

- People are the most important assets.
- The tasks of a manager:
  - Select staff
  - Motivate people
  - Manage groups
  - Solve technical and non technical problems in the most effective way
- Poor people management is an important contributor to project failure.

People management factors

- Consistency → Team members should all be treated in a comparable way without favourites or discrimination.
- Respect → Different team members have different skills and these differences should be respected.
- Inclusion → Involve all team members and make sure that people’s views are considered.
- Honesty → You should always be honest about what is going well and what is going badly in a project.

Managing groups

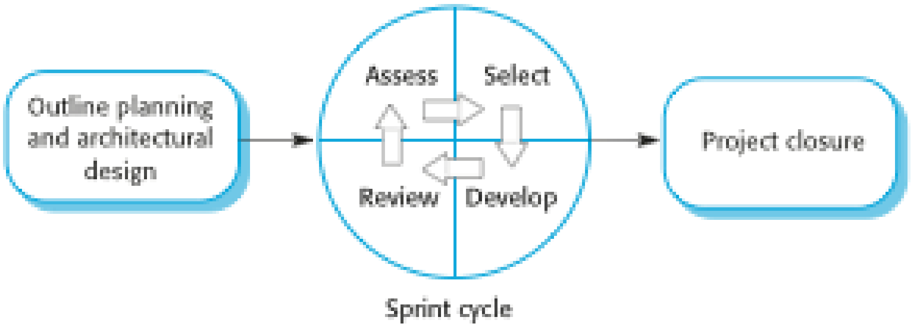
- Most software engineering is a group activity.
- Group interaction is a key determinant of group performance.
- *A group is more than a collection of individuals!*

Group working

- Group composition
  - The right balance of technical skills, experiences and personalities.
- Group cohesiveness
  - Group spirit: members consider the group to be more important than any individual in it.
  - Openness, learn from each other, social events.
- Group communications
  - Size, structure, composition, environment etc.
- Group organisation
  - There may be a hierarchical structure for large projects.

Agile project management

- The standard approach to project management is plan-driven.
- Agile project management requires a different approach, which is adapted to incremental development and the particular strengths of agile methods. (Scrum approcah)



Teamwork

- The ‘Scrum master’ is a facilitator who
  - arranges short daily stand up meetings
  - tracks the backlog of work to be done
  - records decisions
  - measures progress against the backlog
  - communicates with customers and management outside of the team.
- Agile team
  - In most Agile teams, the golden number is said to be between 5 and 9 members

Scrum approach benefits

- The product is broken down into a set of manageable and understandable chunks.
- Unstable requirements do not hold up progress.
- The whole team have visibility of everything and consequently team communication is improved.
- Customers see on-time delivery of increments and gain feedback on how the product works.
- Trust between customers and developers is established and a positive culture is created in which everyone expects the project to succeed.

Summary

- Project management
  - Activities
  - Project Planning
  - Project Scheduling
  - Managing People
  - Agile project management

References

- Chapter 3 and 22 – “Software Engineering” textbook by Ian Sommerville
- Introduction to Agile by Sondra Ashmore