Chapter 34

Project Scheduling

Slide Set to accompany
Software Engineering: A Practitioner's Approach
by Roger S. Pressman

Table of Contents

34.1 Basic Concepts

Why Are Projects Late?

- An unrealistic deadline established by someone outside the software development group
- Changing customer requirements that are not reflected in schedule changes;
- An underestimate of the amount of effort and/or the number of resources;
- Predictable and/or unpredictable risks that were not considered when the project commenced;
- Technical difficulties that could not have been foreseen in advance;
- Human difficulties that could not have been foreseen in advance;
- Miscommunication among project staff that results in delays;
- A failure by project management to recognize that the project is falling behind schedule and a lack of action to correct the problem

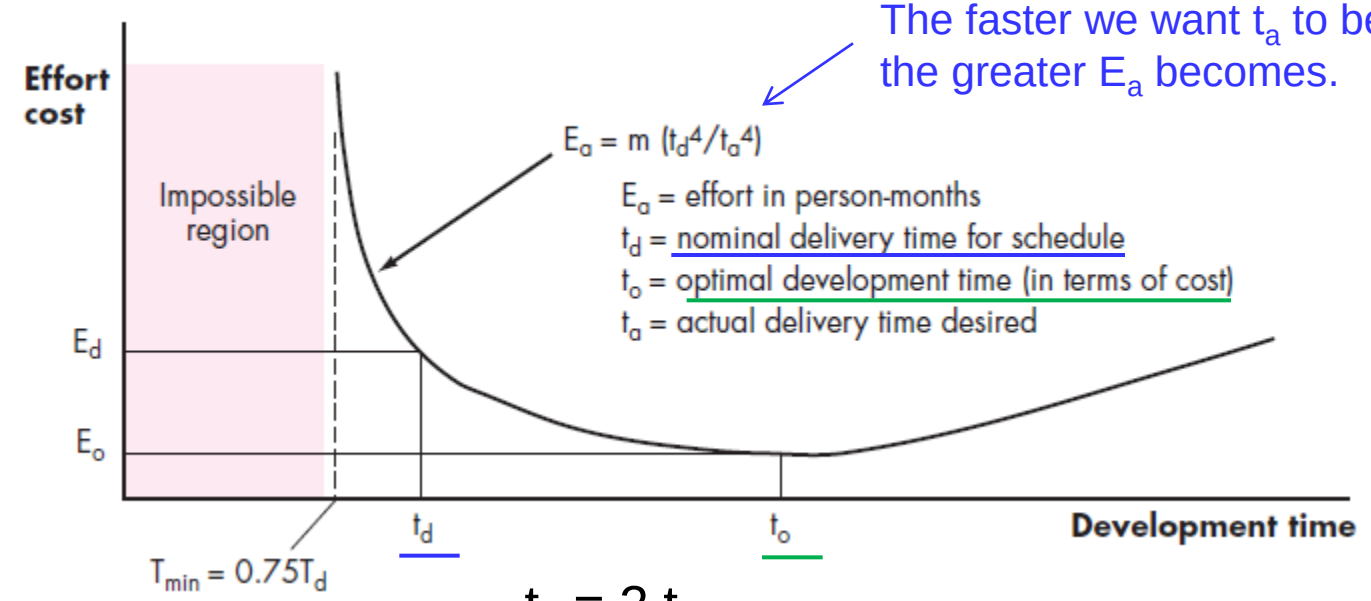
34.2 Project Scheduling

34.2.1 Basic Principles

- Compartmentalization —define distinct tasks
- Interdependency —indicate task interrelationship
- Effort validation —be sure resources are available
- Defined responsibilities —people must be assigned
- Defined outcomes —each task must have an output
- Defined milestones —review for quality

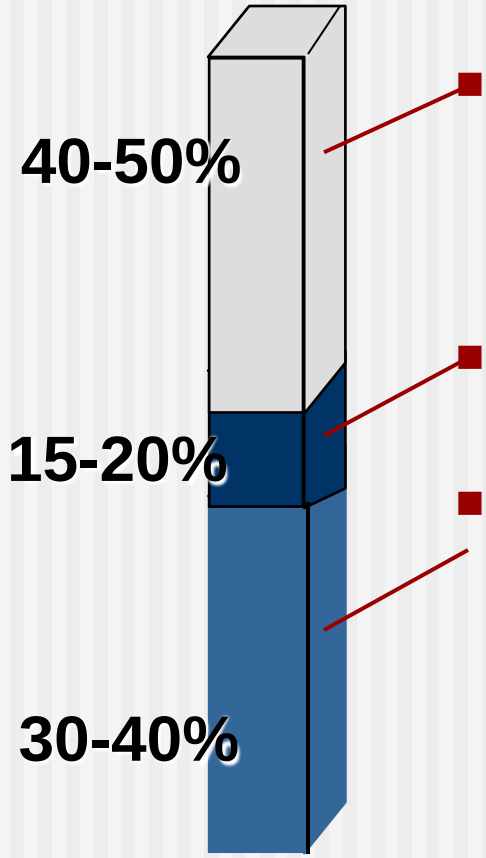
34.2.2 The Relationship Between People and Effort

The relation-ship between effort and delivery time



$t_o = 2\ t_d$

34.2.3 Effort Distribution



- "Front end" activities
  - customer communication
  - analysis
  - design
  - review and modification
- Construction activities
  - coding or code generation
- Testing and installation
  - unit, integration
  - white-box, black box
  - regression

How about the numbers of people needed for different phases?

34.3 Defining a Task Set for the Software Project

1) Determine type of the project:
   - Concept development: explore new business or technology
   - New application development
   - Application enhancement
   - Application maintenance
   - Reengineering
2) Assess the degree of rigor required
3) Identify adaptation criteria
4) Select appropriate software engineering tasks

34.3.1 A Task Set Example

- Tasks for a concept development project

1.1 **Concept scoping** determines the overall scope of the project.

1.2 **Preliminary concept planning** establishes the organization's ability to undertake the work implied by the project scope.

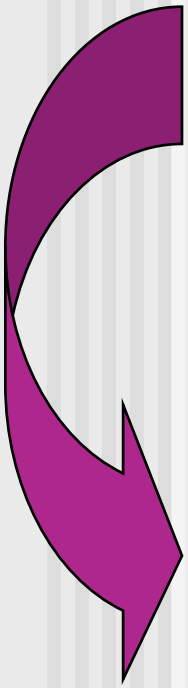
1.3 **Technology risk assessment** evaluates the risk associated with the technology to be implemented as part of the project scope.

1.4 **Proof of concept** demonstrates the viability of a new technology in the software context.

1.5 **Concept implementation** implements the concept representation in a manner that can be reviewed by a customer and is used for "marketing" purposes when a concept must be sold to other customers or management.

1.6 **Customer reaction** to the concept solicits feedback on a new technology concept and targets specific customer applications.

34.3.2 Refinement of Software Engineering Actions



is refined to

1.1 **Concept scoping** determines the overall scope of the project.

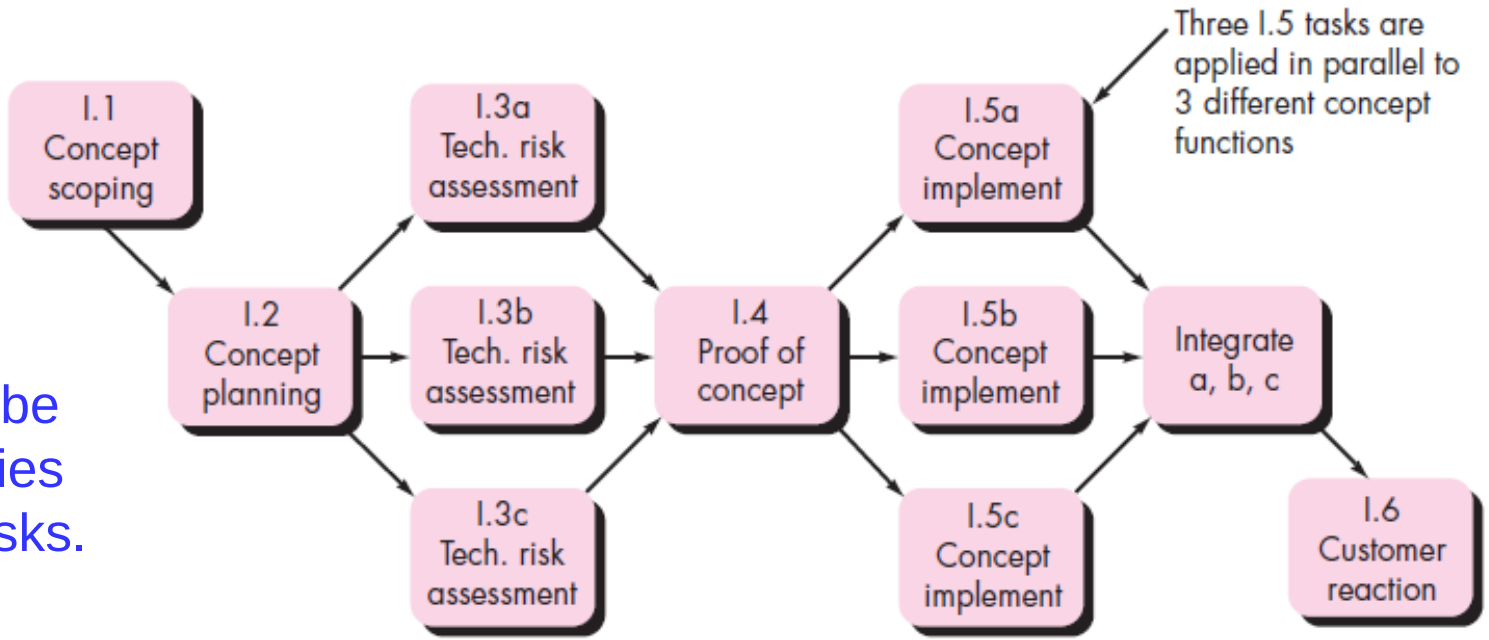
FTR: a formal technical review to be conducted.

Task definition: Task 1.1 Concept Scoping
1.1.1 Identify need, benefits and potential customers;
1.1.2 Define desired output/control and input events that drive the application;
Begin Task 1.1.2
1.1.2.1 FTR: Review written description of need
1.1.2.2 Derive a list of customer visible outputs/inputs
1.1.2.3 FTR: Review outputs/inputs with customer and revise as required;
endtask Task 1.1.2
1.1.3 Define the functionality/behavior for each major function;
Begin Task 1.1.3
1.1.3.1 FTR: Review output and input data objects derived in task 1.1.2;
1.1.3.2 Derive a model of functions/behaviors;
1.1.3.3 FTR: Review functions/behaviors with customer and revise as required;
endtask Task 1.1.3
1.1.4 Isolate those elements of the technology to be implemented in software;
1.1.5 Research availability of existing software;
1.1.6 Define technical feasibility;
1.1.7 Make quick estimate of size;
1.1.8 Create a Scope Definition;
endTask definition: Task 1.1

34.4 Defining a Task Network

There may be dependencies between tasks.

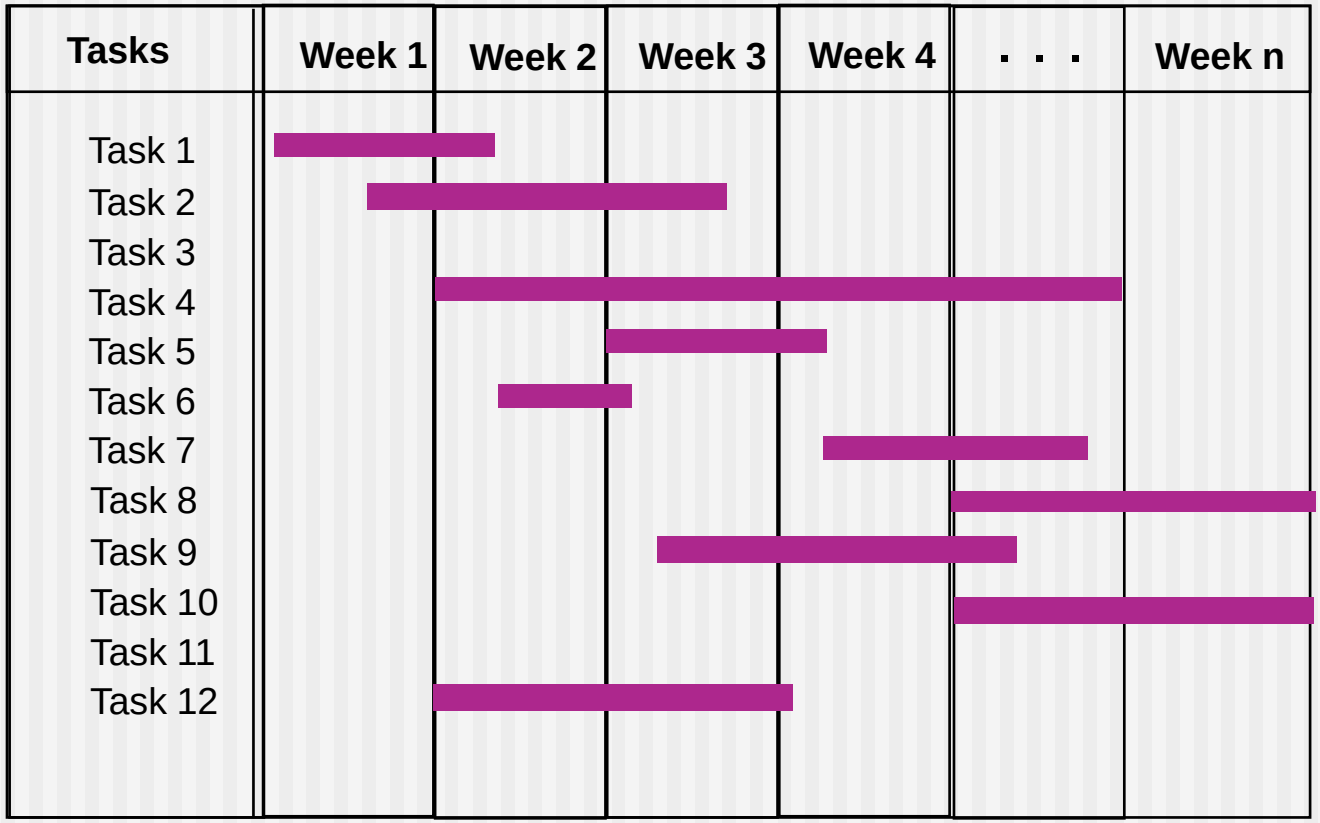
FIGURE 27.2 A task network for concept development



34.5 Scheduling

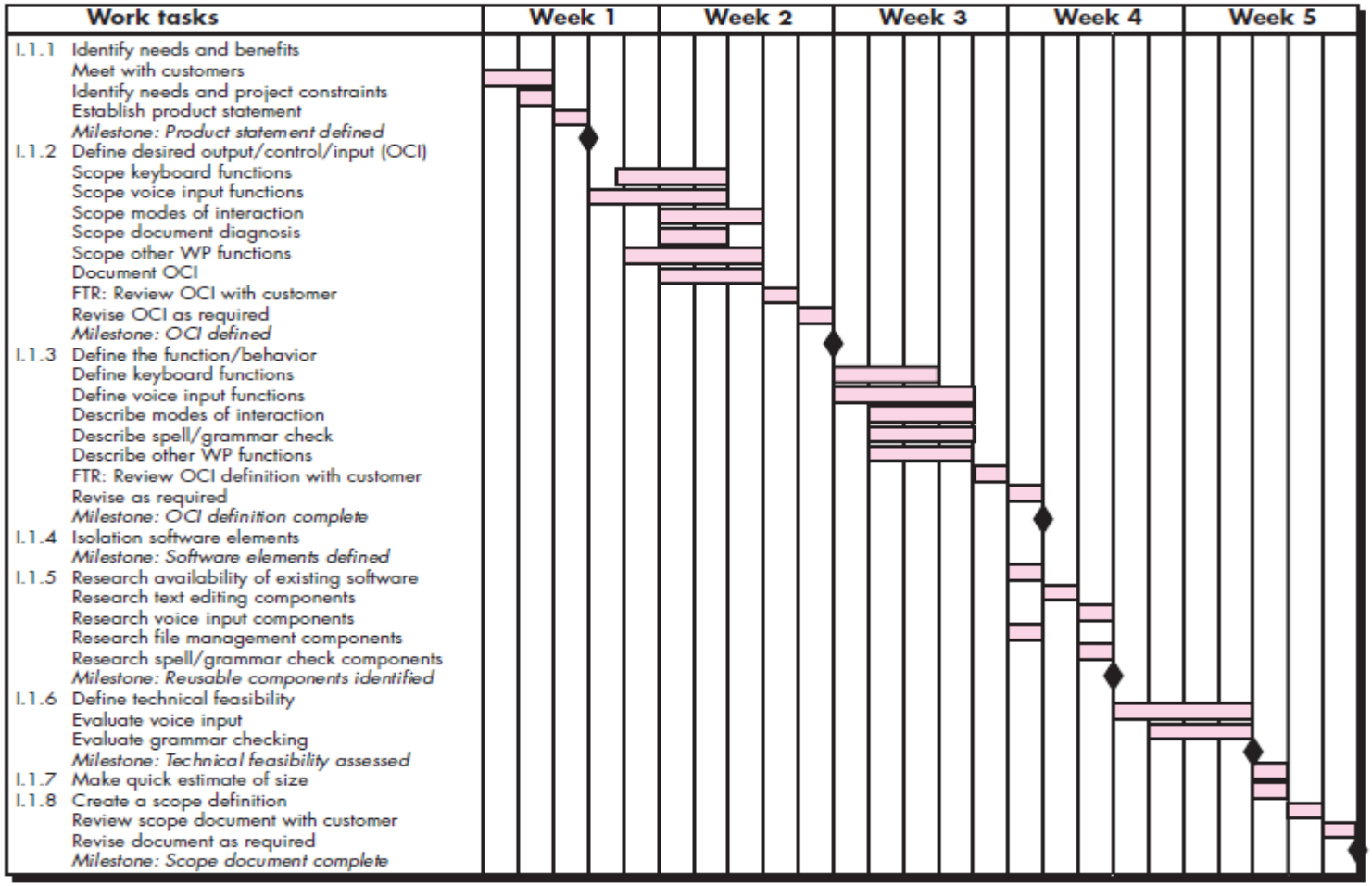
34.5.1 Time-Line Charts

Still we have to decide who will do what.



Use Automated Tools to Derive a Timeline Chart

FIGURE 27.3 An example time-line chart



PERT Chart

- Two graphical notations used to describe project schedule.
  - Activity charts (PERT) show task dependencies and the critical path.
  - Bar charts (Gantt Chart) show schedule against calendar time.

  - PERT (Program Evaluation and Review Technique)

  - PERT Chart = Activity Network = Task Network = Activity Chart

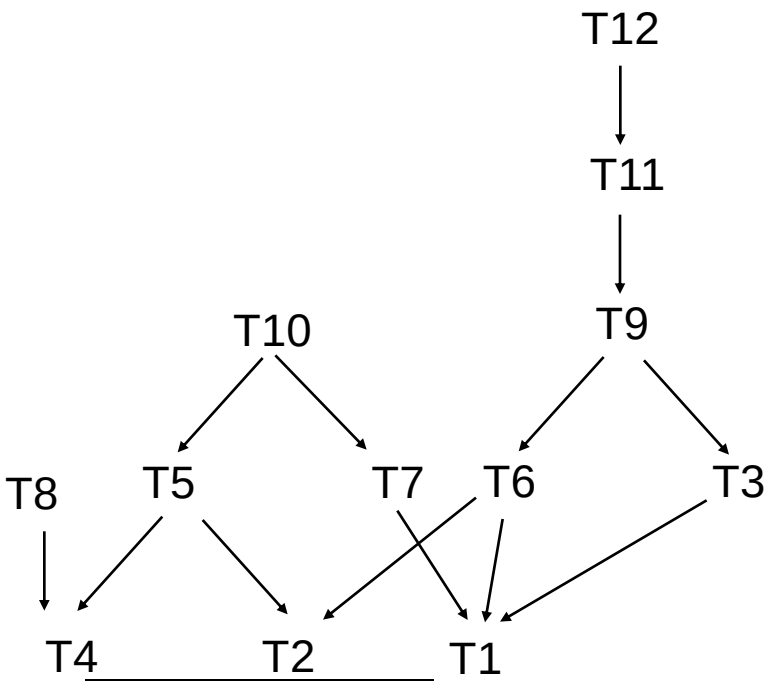
  - Show project breakdown into tasks:
    - Tasks should not be too small.
    - 1 week <= Task duration <= 2 weeks

Task durations and dependencies

Activity	Duration (days)	Dependencies
T1	8	
T2	15	
T3	15	T1 (M 1)
T4	10	
T5	10	T2, T4 (M 2)
T6	5	T1, T2 (M 3)
T7	20	T1 (M 1)
T8	25	T4 (M 5)
T9	15	T3, T6 (M 4)
T10	15	T5, T7 (M 7)
T11	7	T9 (M 6)
T12	10	T11 (M 8)

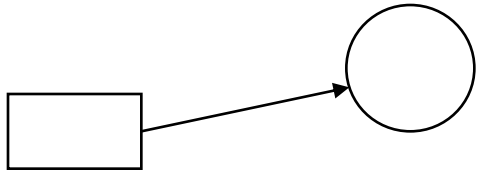
Milestones are M1, ..., M8.

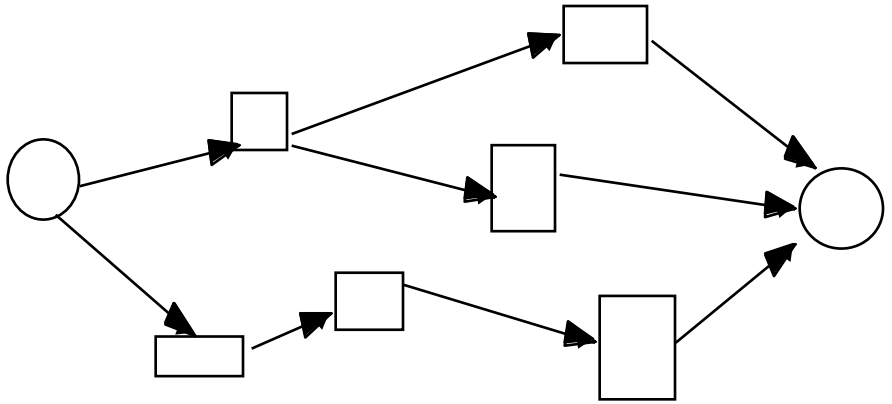
Task Dependency Graph



How to make a PERT Chart?

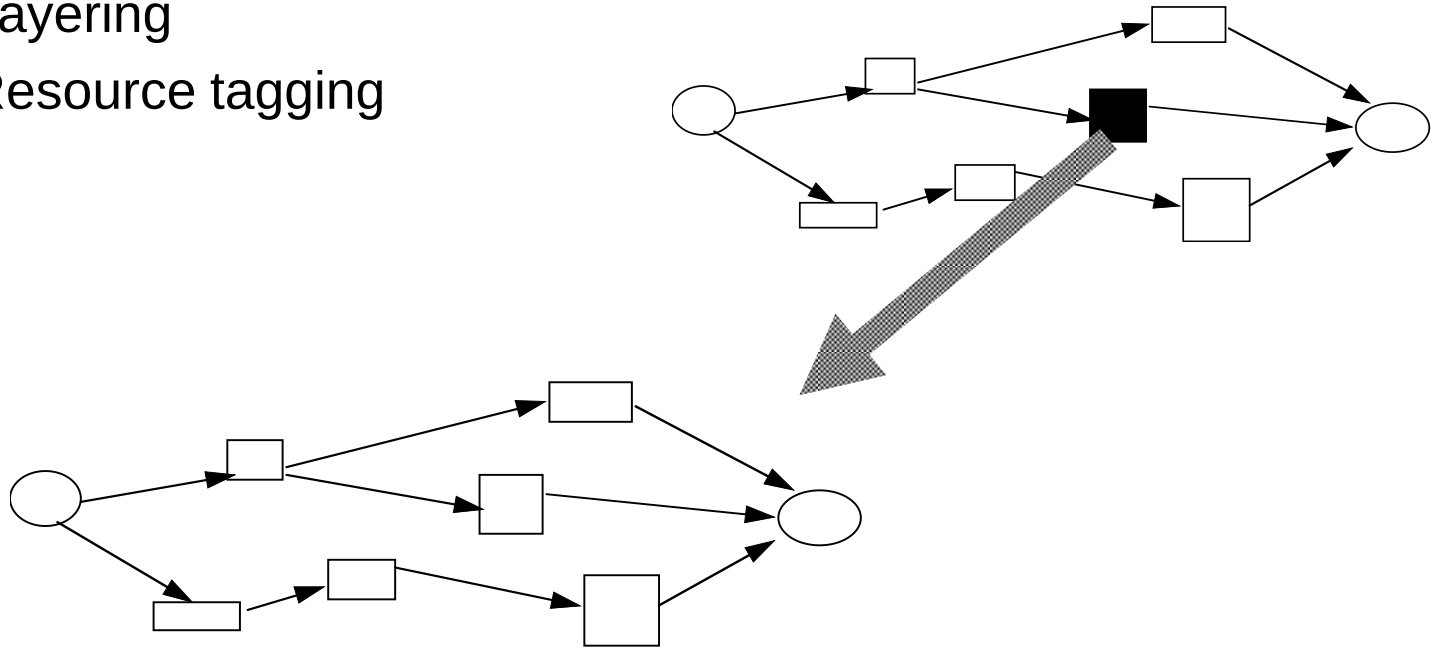
1. Draw the finish node
2. Place nodes that constitute completion
   as immediate predecessors
3. Rethink whether successor nodes exist
4. Add predecessor nodes, if necessary
5. Add successor nodes, if necessary
6. Loop on 4 and 5 until start node is reached



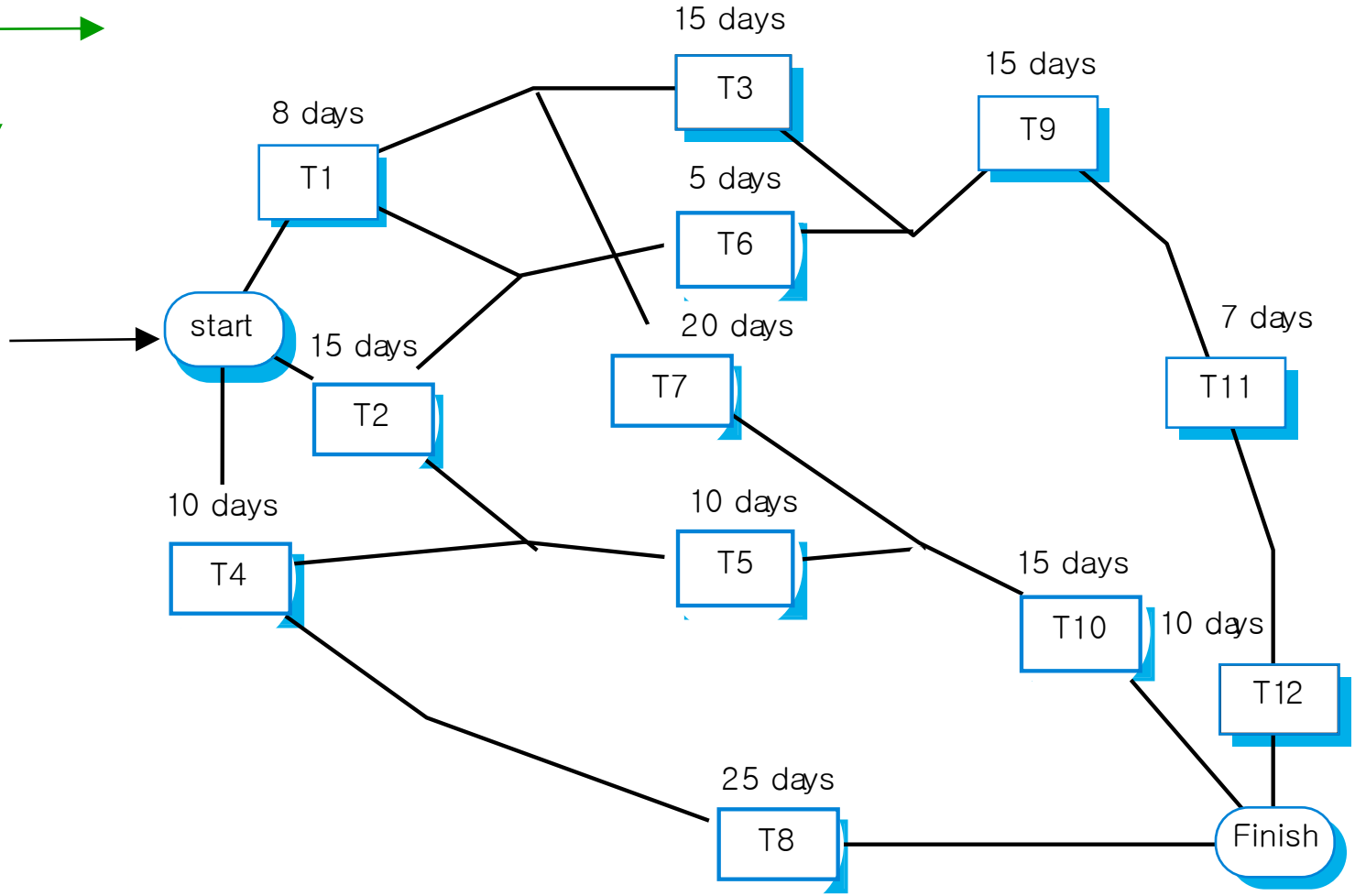


Tool-Supported Additions to PERT

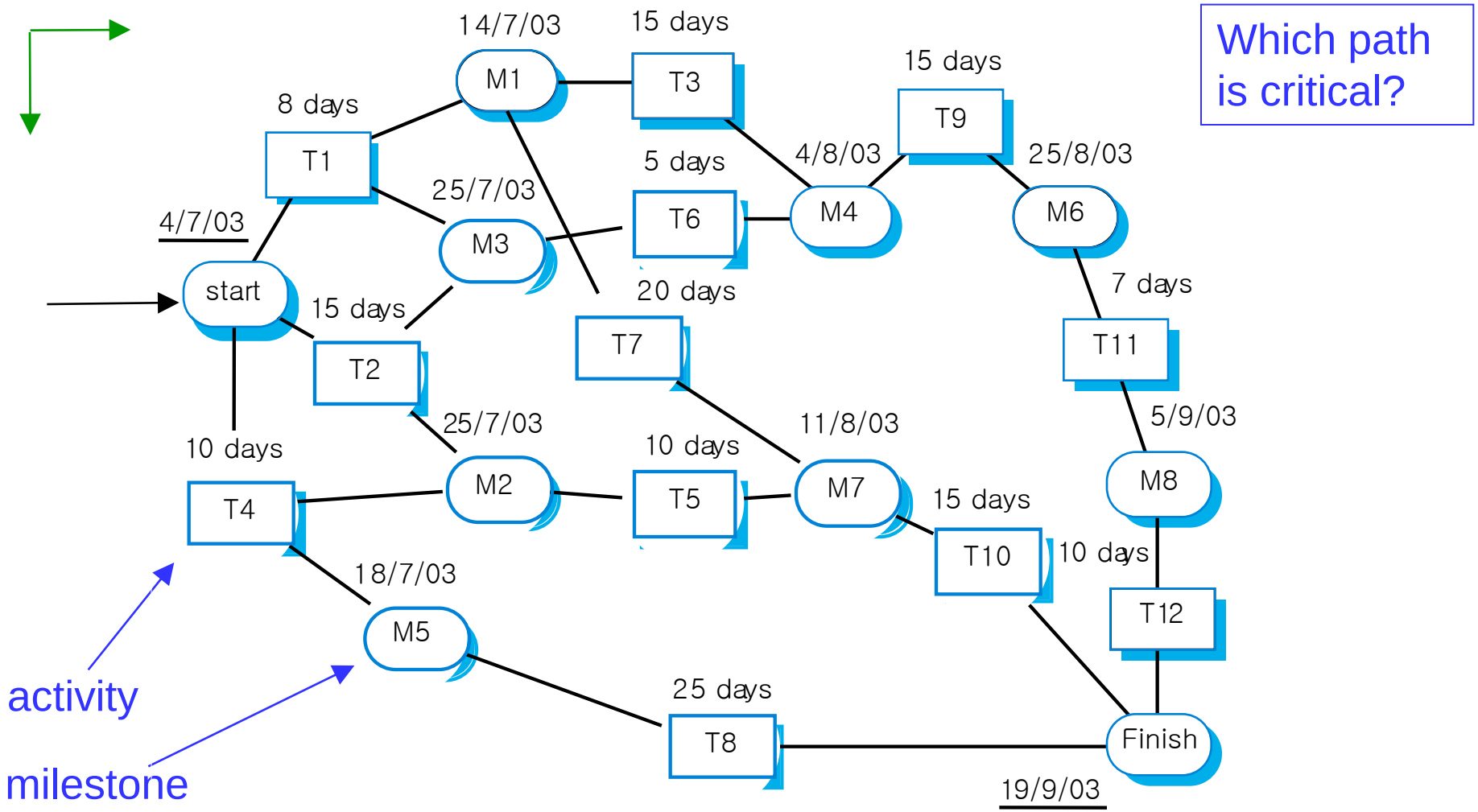
- Layering
- Resource tagging



Activity Network



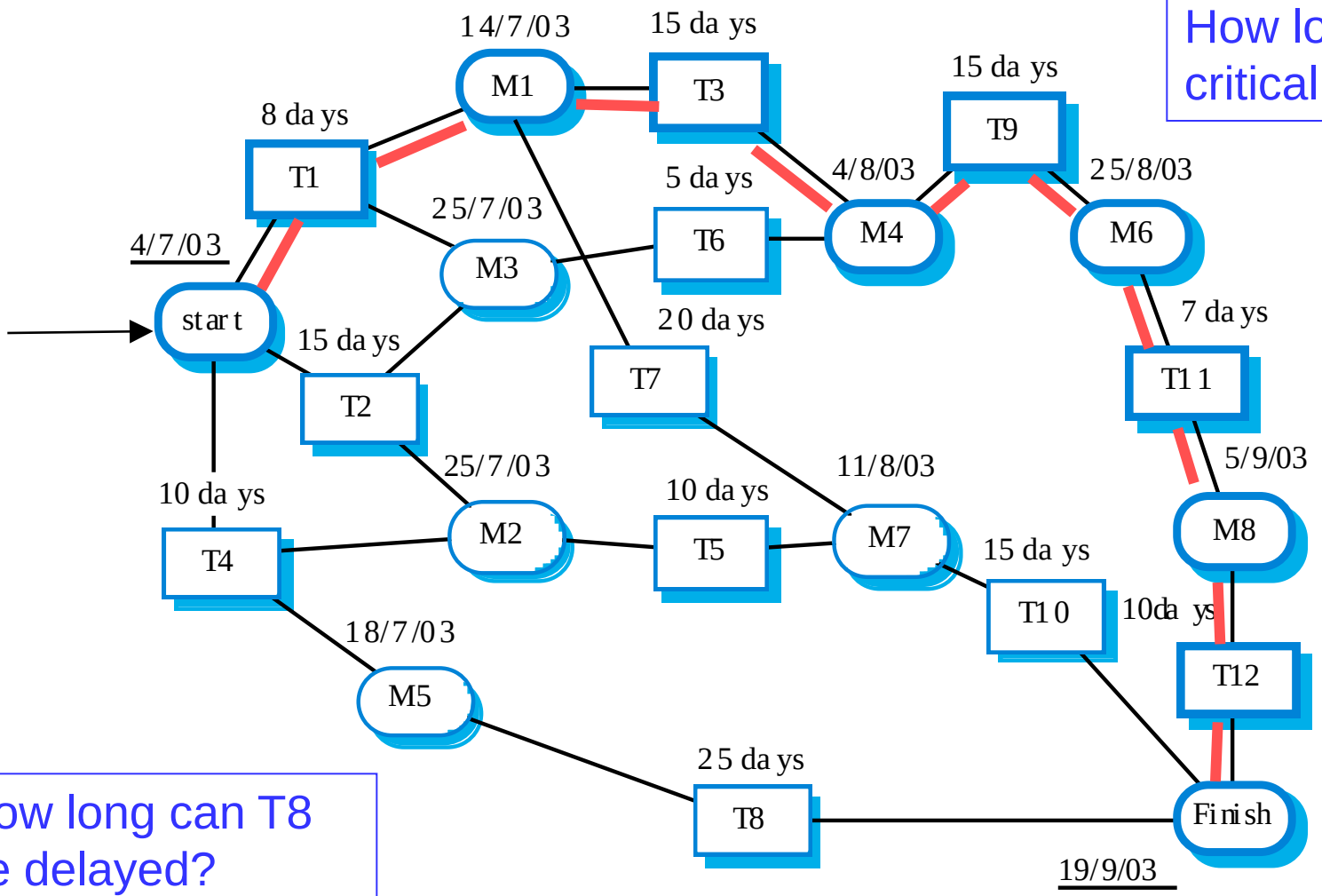
Activity Network



Critical Path

- The longest path through the network in terms of the total duration of tasks:
  - Gives minimum time required to finish the project
  - Delay in a task on the critical path delays the whole project
  - Delays in a noncritical path task may result in a new critical path
  - In complicated projects many “near-critical” tasks and paths may exist

Critical Path



How long is the critical path?

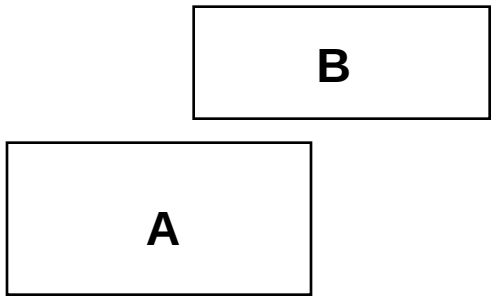
How long can T8 be delayed?

FAQs About Activity Networks

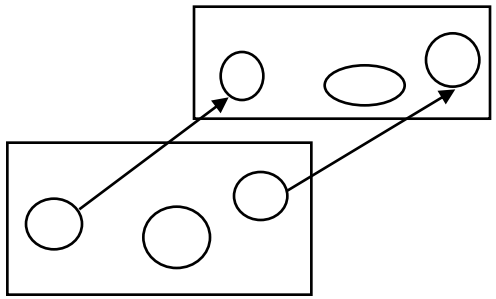
(1) How do I handle overlaps?
(2) How do I show slack time?
(3) How do I handle iterative or evolutionary development models?

(1) Handling Overlaps

Parts of B can begin before A is done



...this is actually a granularity problem



FAQs About Activity Networks

(2) Showing Slack Time

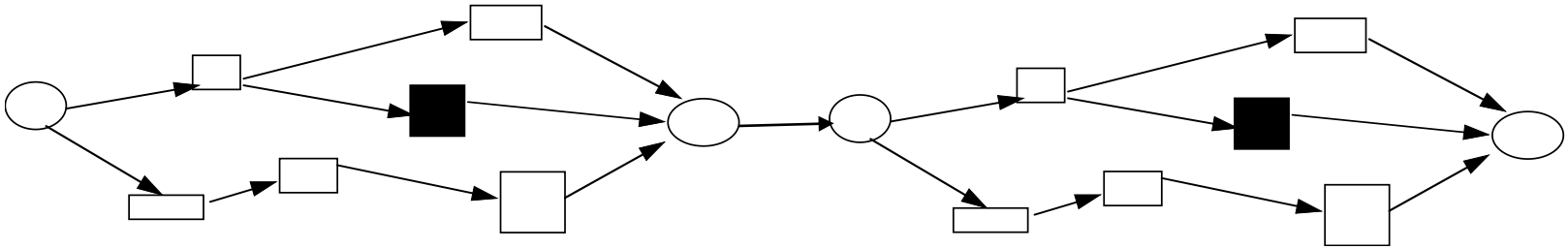
earliest start time

duration

latest finish time

where latest finish minus earliest start <> duration!

(3) Handling Evolutionary Process Model

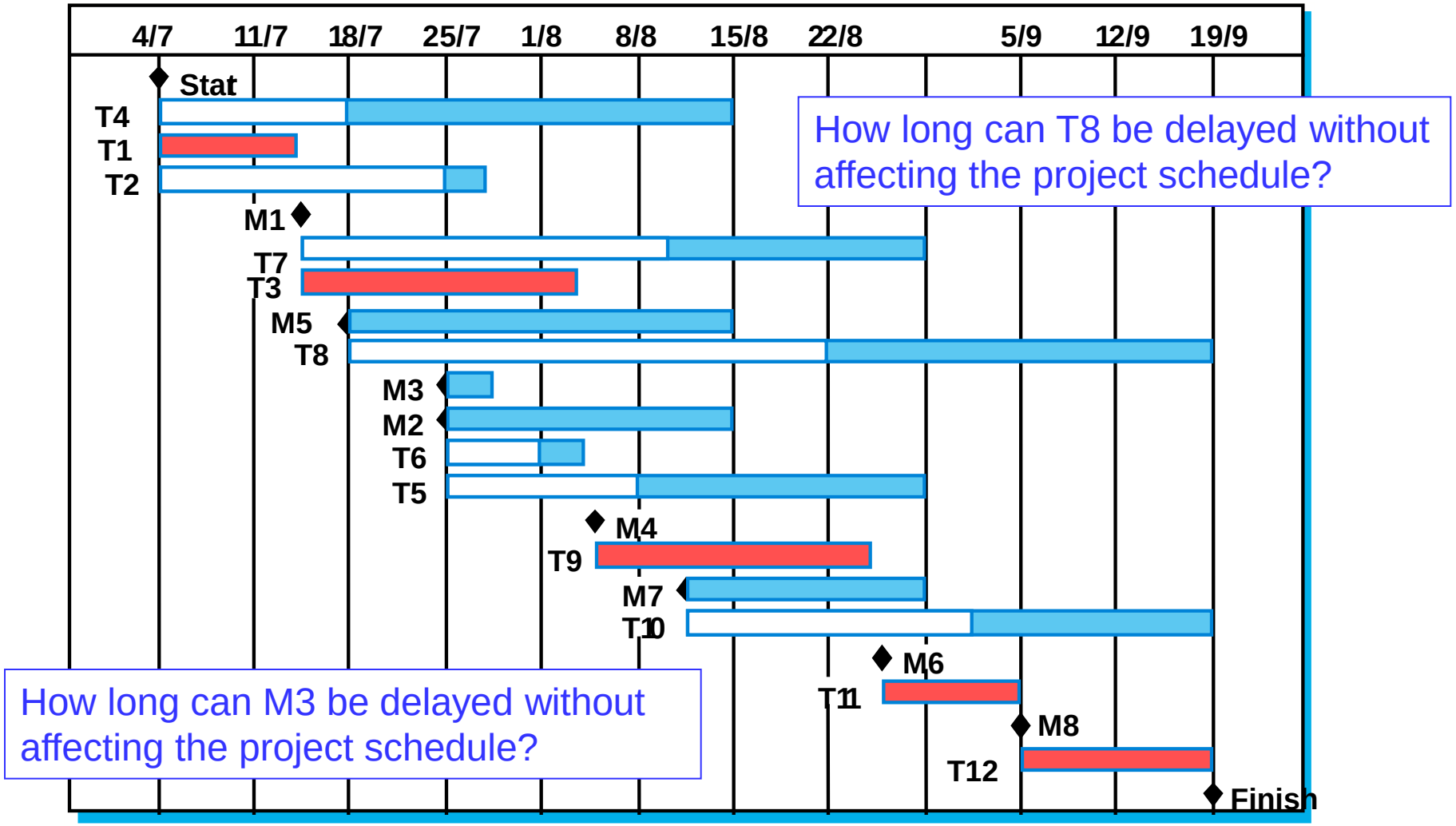


line them up...

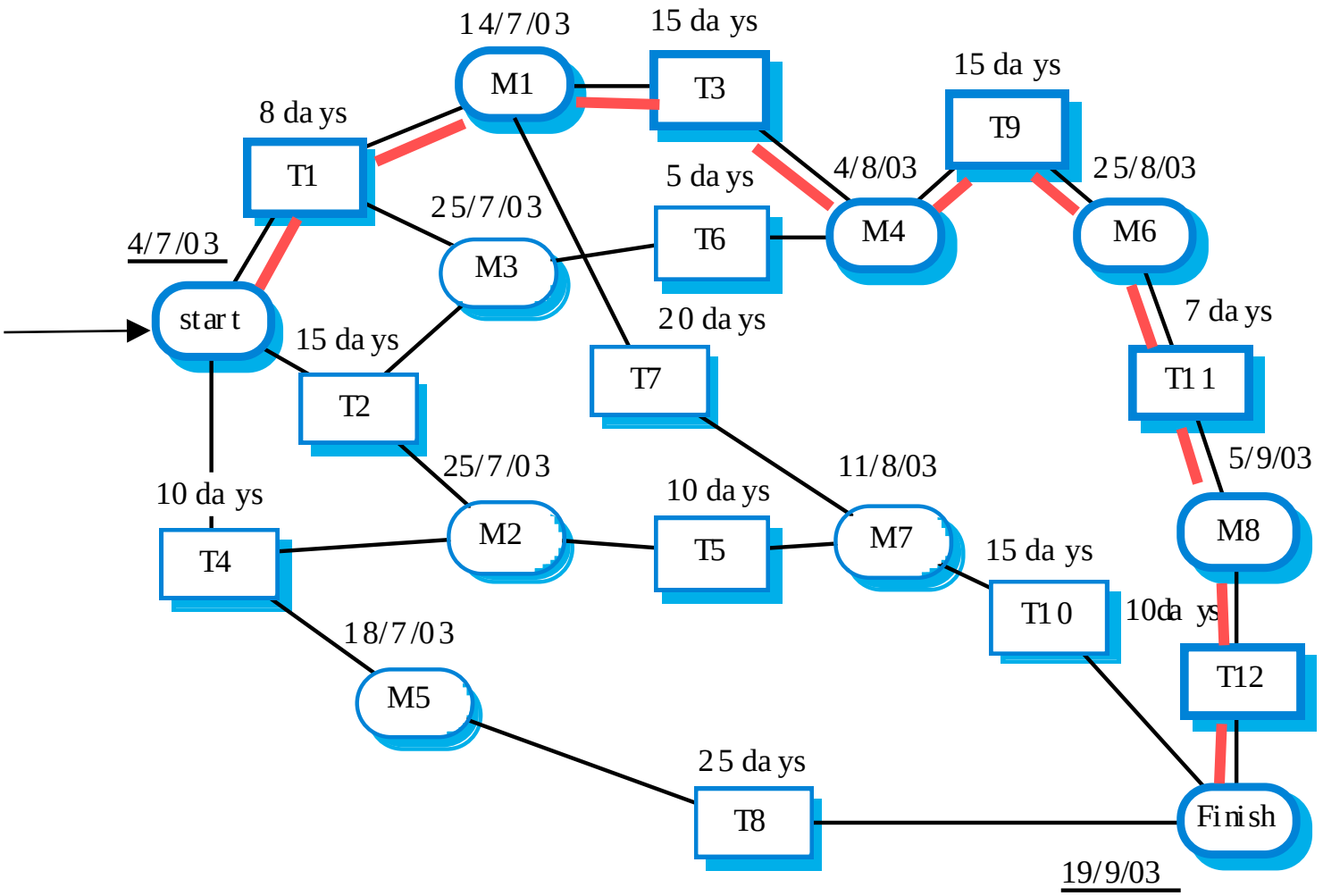
Gantt Chart

- Gantt Chart = Bar Chart = Timeline Chart = Activity Timeline
- Shows schedule against calendar time
- Considered useful for projects with smaller numbers of activities
- Can be derived from activity networks (and vice versa)

Example – Assume dependencies in the PERT chart in slide 19

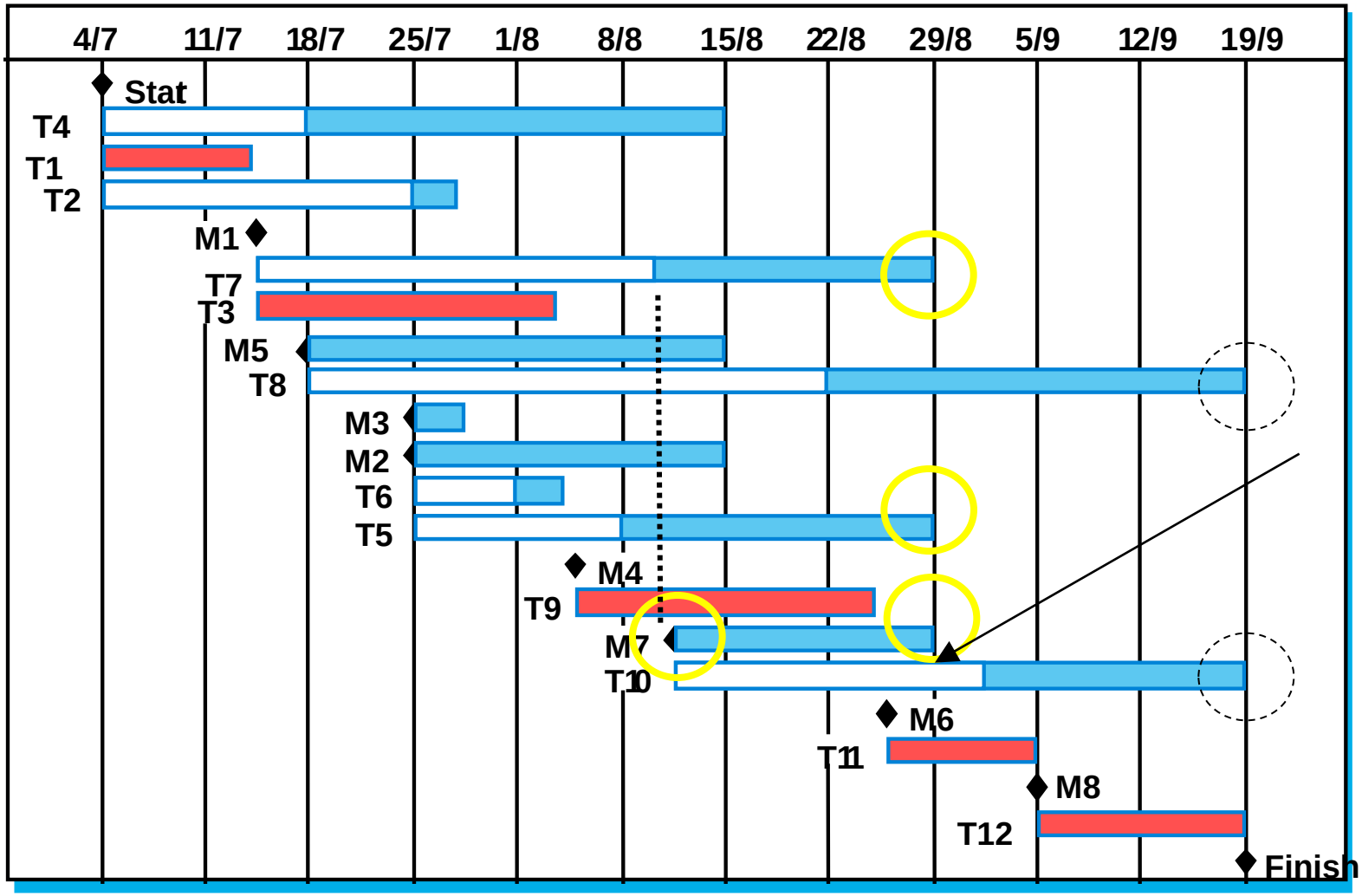


Critical Path

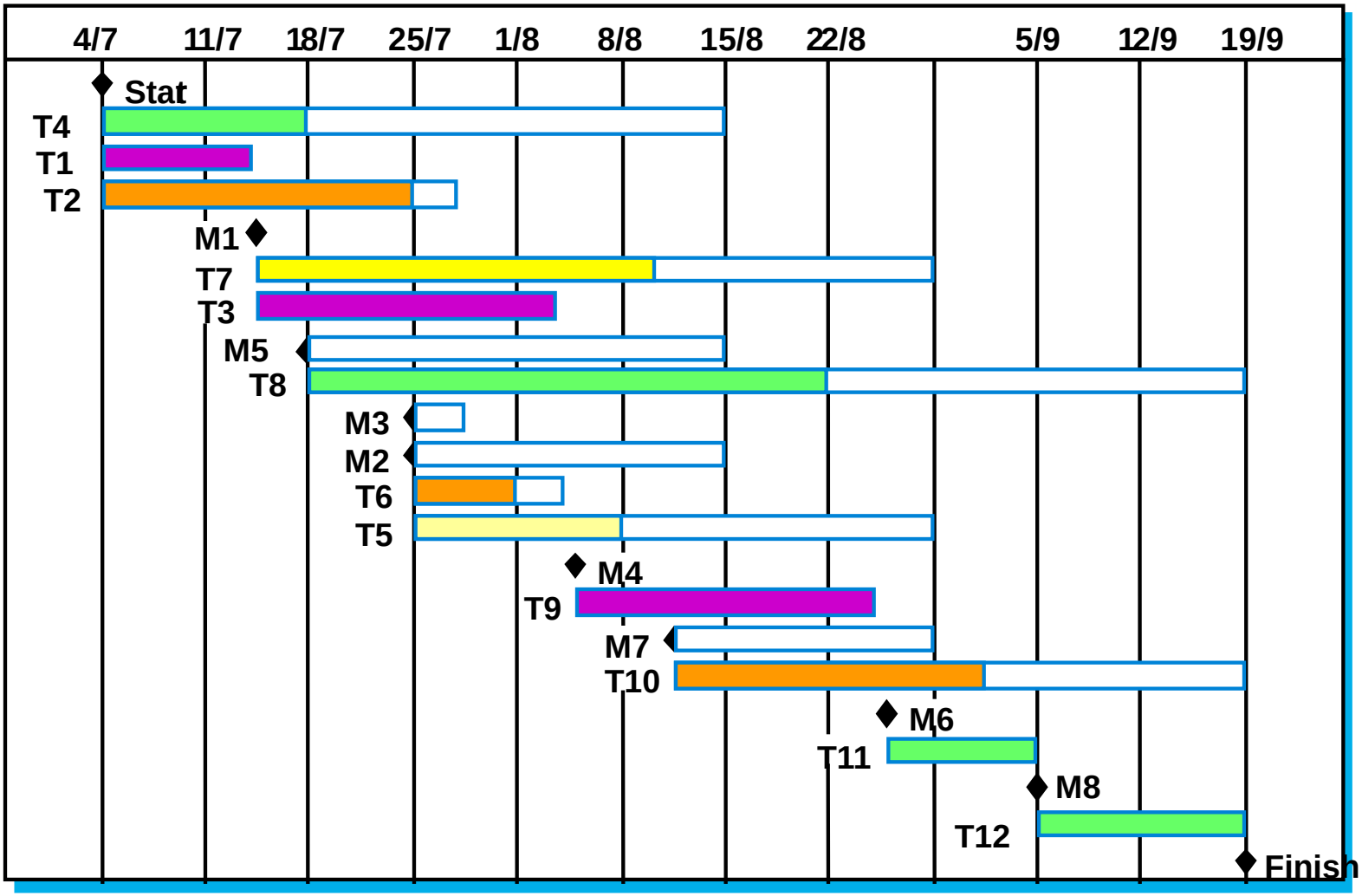


Example

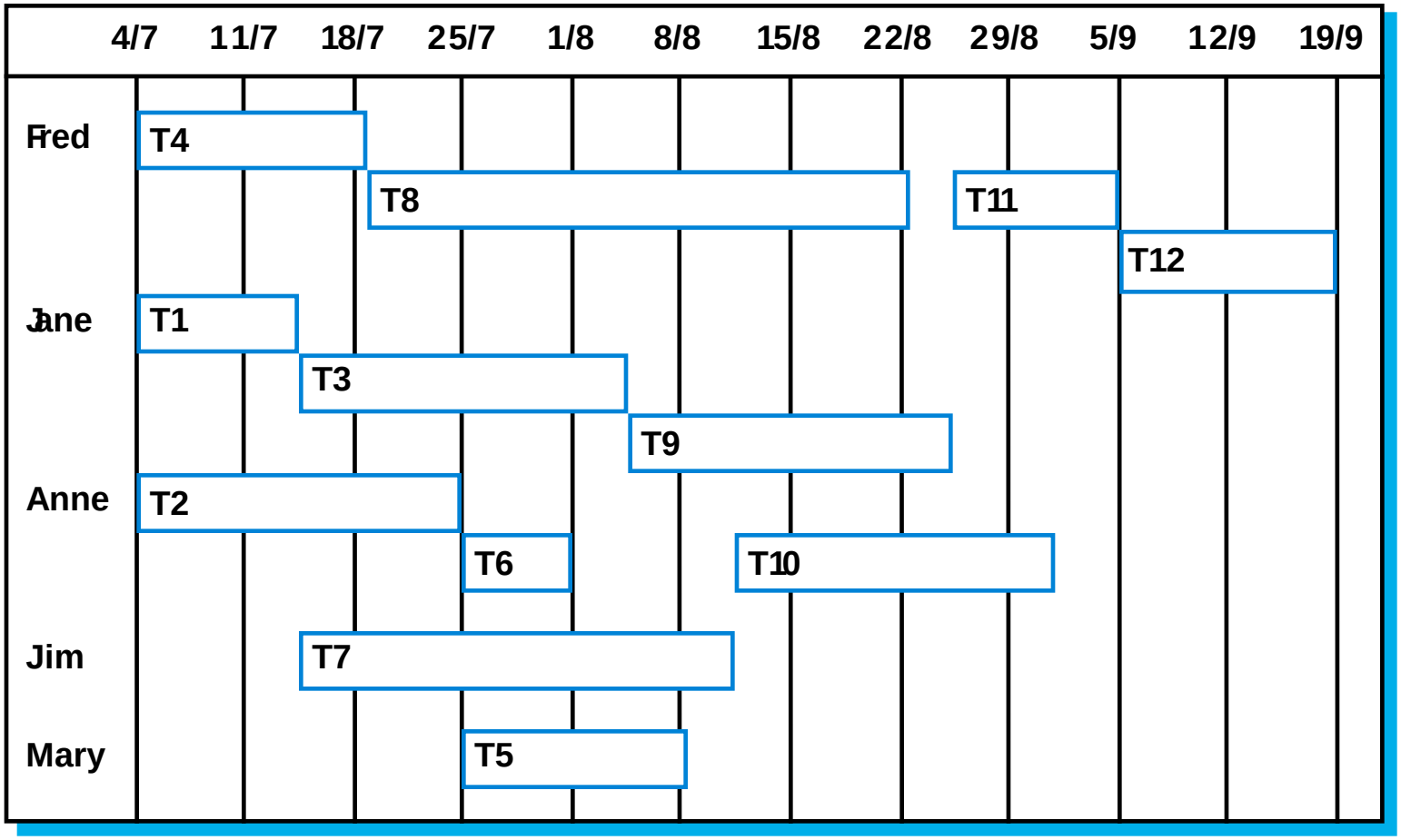
Q. What is the minimum number of people that are required?



Example


4/7
11/7
18/7
25/7
1/8
8/8
15/8
22/8
5/9
12/9
19/9
Stat
T4
T1
T2
M1
T7
T3
M5
T8
M3
M2
T6
T5
M4
T9
M7
T10
M6
T11
M8
T12
Finish

Staff Allocation



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.
  - Sommerville
    1. Estimate as if nothing will go wrong
    2. Add 30% for anticipated problems
    3. Add 20% for unanticipated problems

34.5.2 Tracking the Schedule

- Conduct periodic project status meetings in which each team member reports progress and problems.
- Evaluate the results of all reviews conducted throughout the software engineering process.
- Determine whether formal project milestones have been accomplished by the scheduled date.
- Compare actual start-date to planned start-date for each project task listed in the resource table
- Meet informally with practitioners to obtain their subjective assessment of progress to date and problems on the horizon.
- Use earned value analysis to assess progress quantitatively.

34.6 Earned Value Analysis (EVA)

- Earned value
  - is a measure of progress
  - enables us to assess the "percent of completeness" of a project using quantitative analysis rather than rely on a gut feeling
  - "provides accurate and reliable readings of performance from as early as 15 percent into the project." [Fle98]

Computing Earned Value (1/3)

- The **budgeted cost of work scheduled (BCWS)** is determined for each work task represented in the schedule.
  - $BCWS_i$ is the effort planned for work task *i.*
  - To determine progress at a given point along the project schedule, the value of BCWS is the sum of the $BCWS_i$ values for all work tasks that should have been completed by that point in time on the project schedule.
- The BCWS values for all work tasks are summed to derive the budget at completion, BAC. Hence,

$$BAC = \sum (BCWS_k) \text{ for all tasks } k$$

Computing Earned Value (2/3)

- Next, the value for **budgeted cost of work performed (BCWP)** is computed.
  - The value for BCWP is the sum of the BCWS values for all work tasks that have actually been completed by a point in time on the project schedule.
- "the distinction between the BCWS and the BCWP is that the former represents the budget of the activities that were planned to be completed and the latter represents the budget of the activities that actually were completed." [Wil99]
- Given values for BCWS, BAC, and BCWP, important progress indicators can be computed:
  - Schedule performance index, SPI = BCWP/BCWS
  - Schedule variance, SV = BCWP – BCWS
  - SPI is an indication of the efficiency with which the project is utilizing scheduled resources.

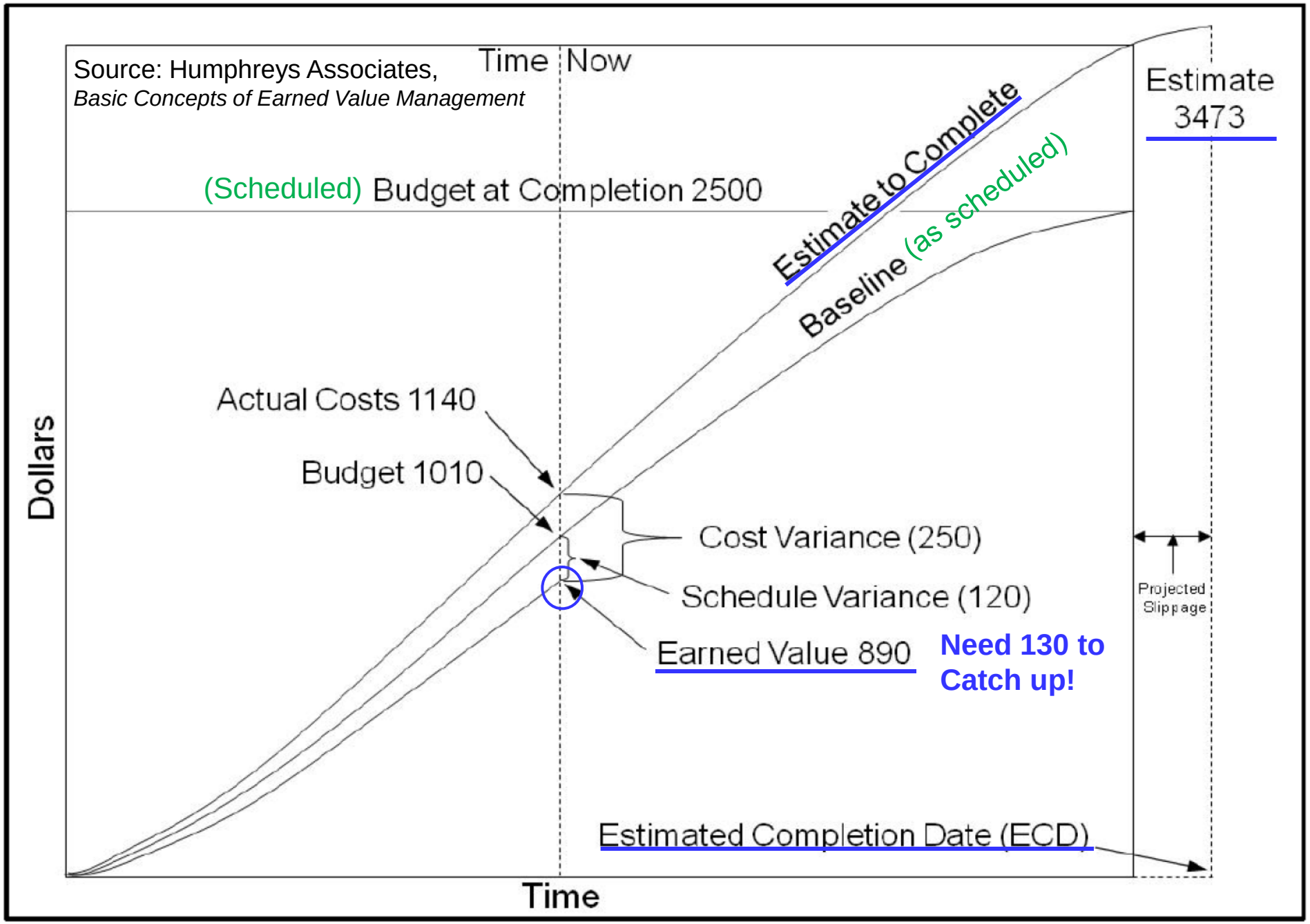
Computing Earned Value (3/3)

- Percent scheduled for completion = BCWS/BAC
  - provides an indication of the percentage of work that should have been completed by time *t.*
- Percent complete = BCWP/BAC
  - provides a quantitative indication of the percent of completeness of the project at a given point in time, *t.*
- Actual cost of work performed, ACWP, is the sum of the effort actually expended on work tasks that have been completed by a point in time on the project schedule. It is then possible to compute
  - Cost performance index, CPI = BCWP/ACWP
  - Cost variance, CV = BCWP – ACWP

Work Breakdown Structure

Source: www.pierdesign.ca

		Budget	Actuals		Earned value		
			Cost to date	Weeks of 12/Mar to 19/Mar	% complete	Earned value	Status
1	Phase 1 - Requirements	$17,280	$16,470		100%	$17,280	early
1.1	meetings, site visits and interviews	$7,200	$7,110		100%	$7,200	early
1.2	market and cost analysis	$3,600	$3,960		100%	$3,600	late
1.3	patent search	$1,440	$810		100%	$1,440	early
1.4	competitor analysis	$1,440	$1,350		100%	$1,440	early
1.5	documentation	$3,600	$3,240		100%	$3,600	early
2	Phase 2 - Design concepts	$36,000	$36,720		100%	$36,000	late
2.1	concept 1 - instrument	$7,200	$7,470		100%	$7,200	late
2.2	concept 2 - instrument	$7,200	$7,290		100%	$7,200	late
2.3	concept 3 - instrument and accessories	$7,200	$7,290		100%	$7,200	late
2.4	concept 4 - instrument and accessories	$7,200	$5,850		100%	$7,200	early
2.5	concept 5 - instrument derivatives	$7,200	$8,820		100%	$7,200	late
3	Phase 3 - Assessment and review	$19,440	$15,570		100%	$19,440	early
3.1	prepare presentation packages	$5,400	$3,960		100%	$5,400	early
3.2	arrange meetings	$1,440	$810		100%	$1,440	early
3.3	focus group, needs analysis	$3,600	$4,050		100%	$3,600	late
3.4	marketing, engineering risk analysis	$3,600	$2,430		100%	$3,600	early
3.5	documentation	$5,400	$4,320		100%	$5,400	early
4	Phase 4 - Concept refinement and definition	$32,400	$19,440	$19,440	86%	$27,900	early
4.1	concept 1 - creative design, user feature solutions	$7,200	$5,670	$5,670	100%	$7,200	early
4.2	concept 1 - engineering feasibility, manufacturing, cost assessment	$7,200	$3,240	$3,240	100%	$7,200	early
4.3	concept 2 - creative design, user feature solutions	$7,200	$7,830	$7,830	75%	$5,400	late
4.4	concept 2 - engineering feasibility, manufacturing, cost assessment	$7,200	$2,340	$2,340	75%	$5,400	early
4.5	documentation	$3,600	$360	$360	75%	$2,700	early
5	Phase 4 - Prototype & validation	$23,400					OK
6	Phase 5 - Design engineering implementation	$48,600					OK
7	Project management	$17,712	$9,630	$1,260	27%	$4,782	late
8	**Total**	**$194,832**	**$97,830**	**$20,700**	**54%**	**$105,402**	**early**



Estimate based on combined cost and schedule performance