(Software) Project-Planning

Poor project-planning is the usual reason for failure.
Project-planning errors are invariably too “aggressive”.

define

estimate

analyse & schedule

track

Task Definition

Project vs. Process tasks

Each task has:

initially:

- defined outcomes (deliverable, work products)
- interdependency
- work assignment (to a team member)
- defined milestone

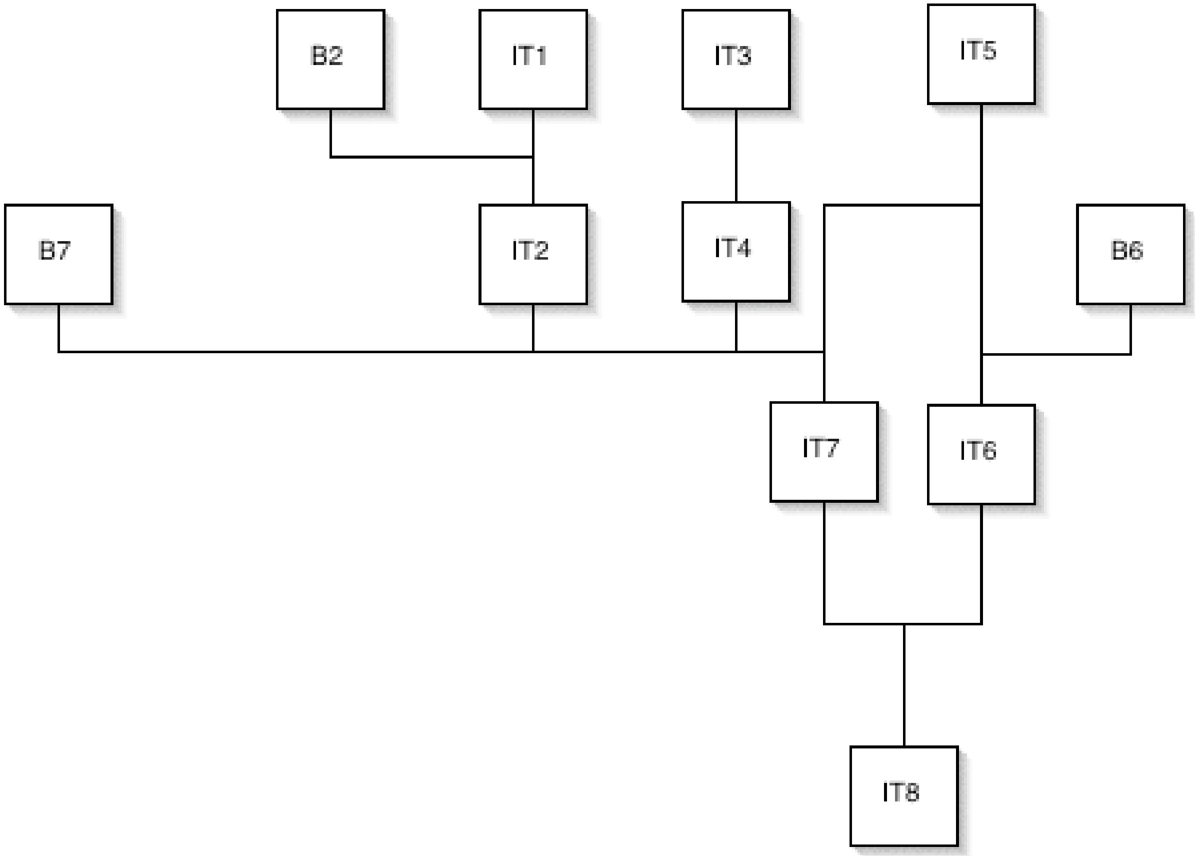
after estimation:

- cost estimate(s)
- time allocation

after analysis:

- boundary times (earliest start, latest start, latest finish, etc.)

example : Priority Constraints



Task Definition
(Example)

Task: IT2
Assigned to: Bill
Description: Second Integration Test for Build 2
Requires: B2, IT1
Required For: IT7
Deliverables:

- Error log for build (if any).
- Test report for integration test (ref: Test Plan sec. 3/B2)

Cost / Duration Table

Records estimates of
time (for scheduling)
cost (for pricing)

task	*est. time*	*est. cost*
	(days)	*($$)*
IT1	6	4800
B2	3	2400
IT2	8	6400
IT3 & IT4	4	3200
IT5	5	4000
B6	2	1600
IT6	4	3200
B7	4	3200
IT7	8	6400
IT8	5	4000
total	49	39 200

Assignment of Tasks

Essential for scheduling and estimation
A social and management activity.
Can’t really be automated much.

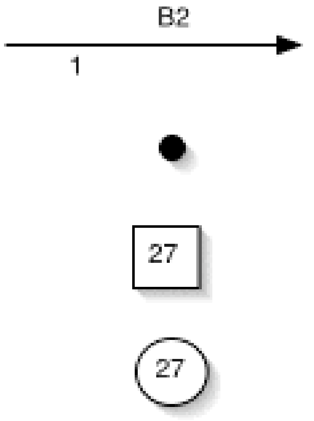
This is where **sequence** is decided on.
“Who does what when”

Coordination and hand-off.
But not collaboration (unless pair-programming).

E.g.

Bill:	IT1, IT3, IT4
Jill:	B2, IT2, IT7, IT8
Phil:	IT5, B7, B6, IT6

Critical Path Method notation

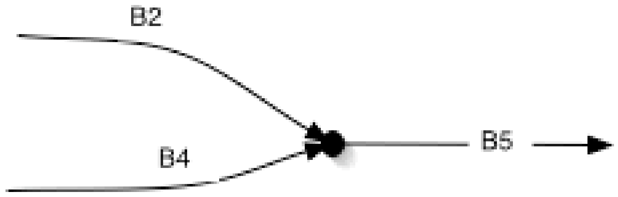


task (name, time estimate)

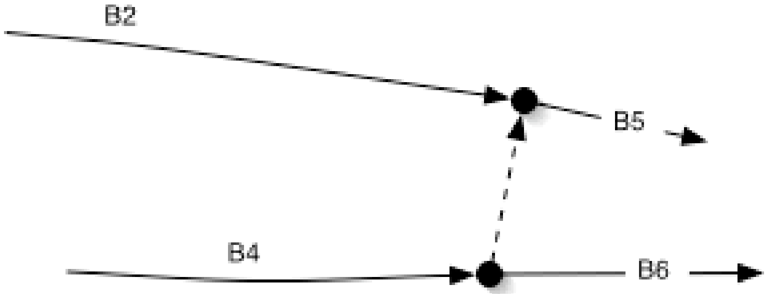
milestone or co-ordination point

earliest possible start date

latest permissible start date

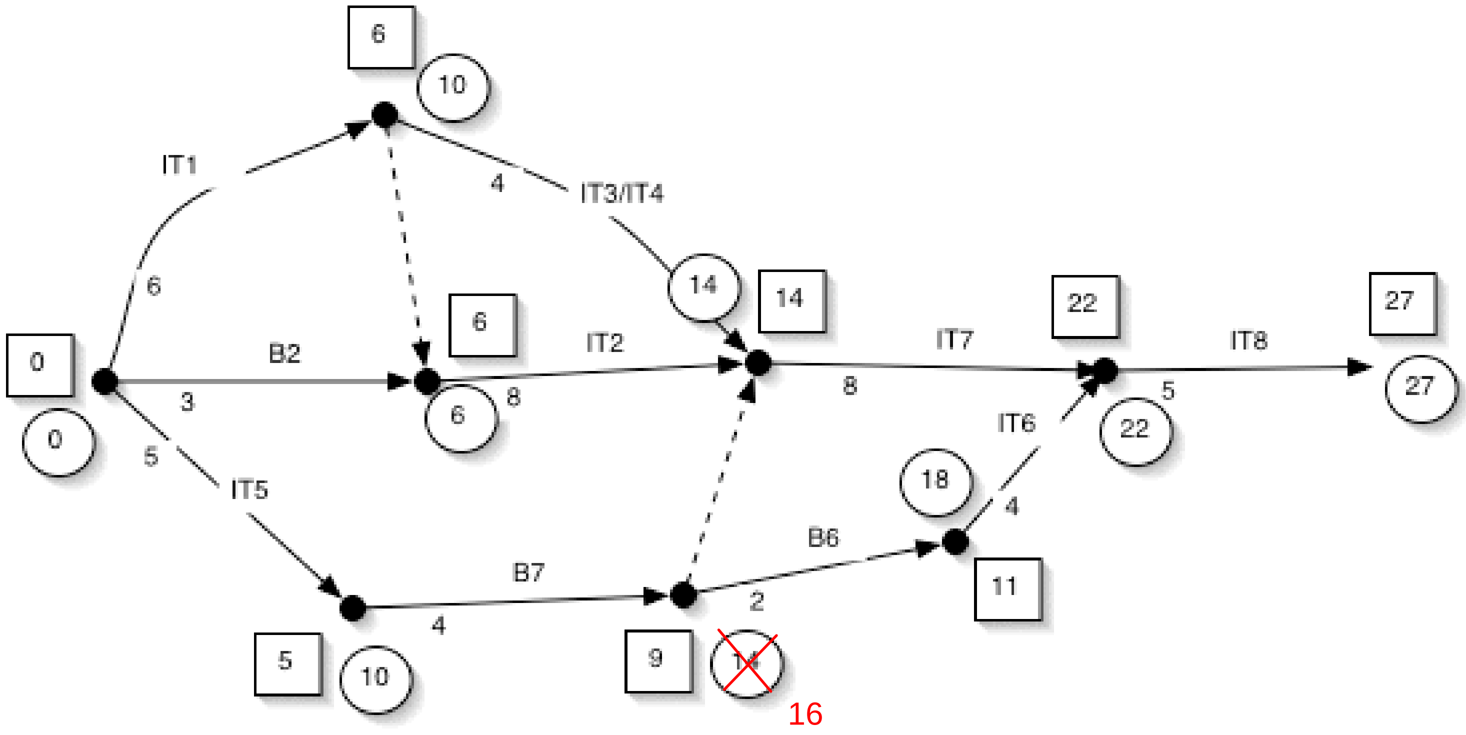


B5 cannot start until B2 and B3 have finished

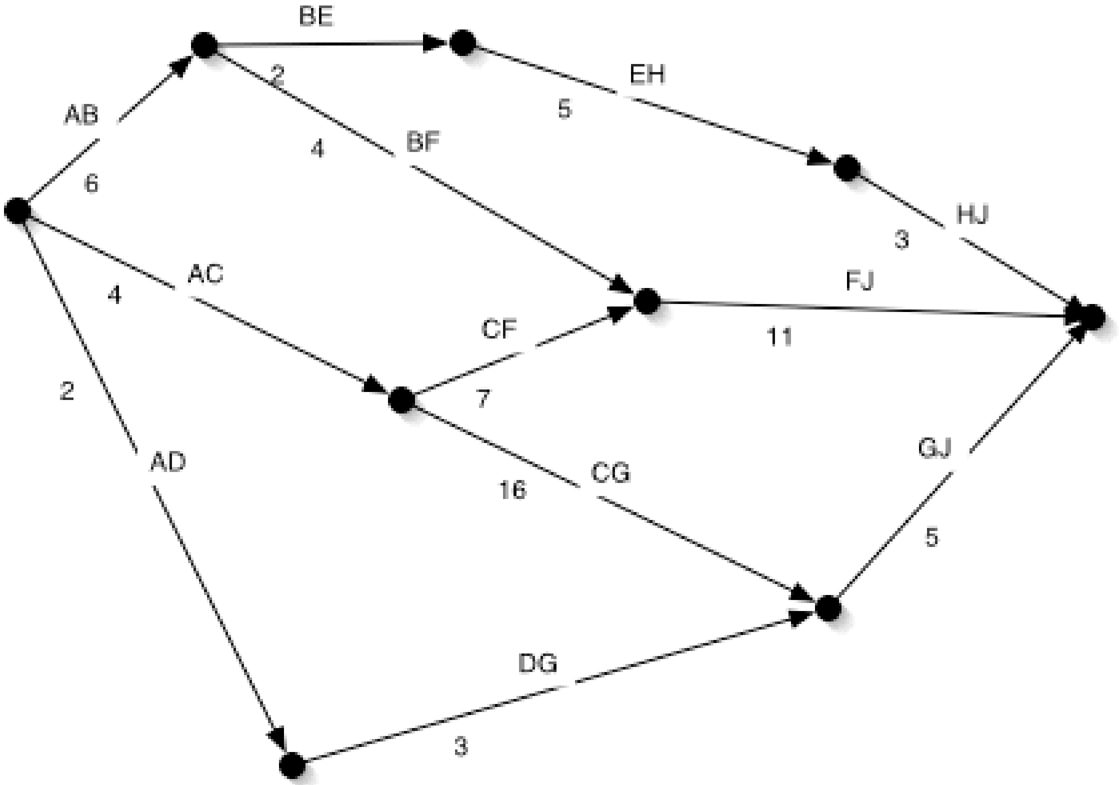


…but B6 can

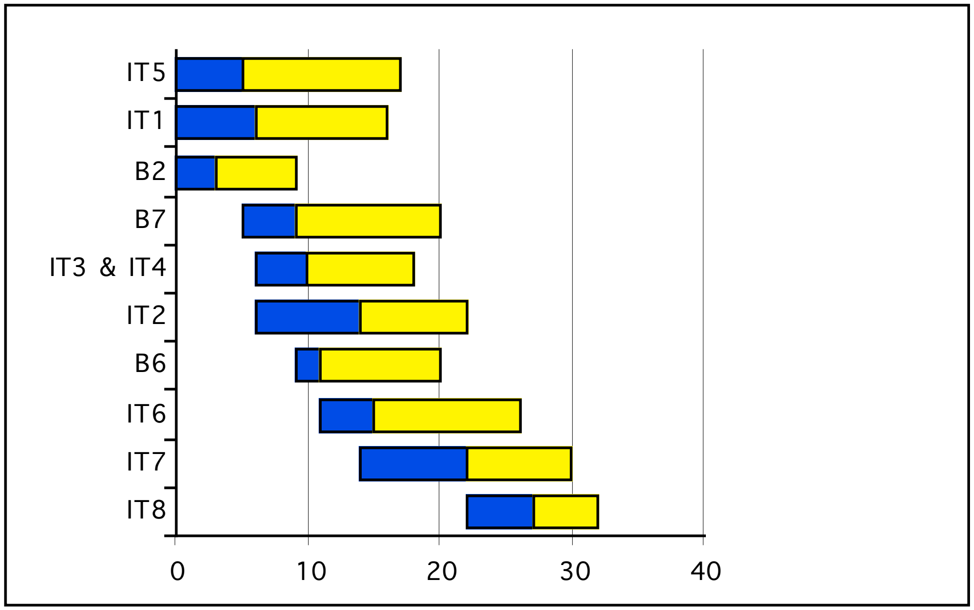
CPM Analysis Graph



CPM Example



Gantt-style chart for sample



Tracking the Schedule

These are standard managers' practices:

- periodic project status meetings: report progress and problems
- evaluating review results
- comparing actual to planned start date
- maintaining timelines for actual elapsed time

These are “agile” practices:

- time-boxing
- iterative development
- “sprints”

Measuring Projects and Products

“Process metrics”

- defect rate
- estimate accuracy (time, resources)
- by developer, module, whole system

“Software metrics” (Product metrics)

- lines of code (LOC)
- speed, memory size, etc.
- various complexity measures on the code
- various complexity measures on the requirements or design
- quality metrics: correctness, maintainability, usability

Goals

- measure improvements
- more accurate estimation
- quantify the unquantifiable

Software Metrics

Reliability:

- mean time to tailure
- availability
- probability of failure on demand

Size and complexity:

- lines of code (LOC), length of identifiers, depth of nesting, …
- fan-in, fan-out, cross-reference (coupling measures)
- pages, readbility (documentation quality)
- actual cost
- function points
- feature points

Goal: improve efficiency and quality of estimation

Estimation methods

Experience

Recorded history (database)

Subtask breakdown

Algorithmic cost models (e.g. COCOMO)