# Software Development Project Management

# Software Development Process?

1) Announce availability.
2) Order the T-shirts for the Development team.
3) Write the code.
4) Write the manual.
5) Hire a Product Manager.
6) Spec the software (Writing the specs after the code helps to ensure that the software meets the specifications).
7) Ship.
8) Test (the customers are a big help here).
9) Identify bugs as potential enhancements.
10) Announce the upgrade program.

# Cynical Views or Reality?

# Cynical Views or Reality?

- Projects progress quickly until they are 90% complete. Then they remain at 90% complete forever.

# Cynical Views or Reality?

- Projects progress quickly until they are 90% complete. Then they remain at 90% complete forever.

- When things are going well, something will go wrong. When things just can't get worse, they will. When things appear to be going better, you have overlooked something.

# Cynical Views or Reality?

- Projects progress quickly until they are 90% complete. Then they remain at 90% complete forever.
- When things are going well, something will go wrong. When things just can't get worse, they will. When things appear to be going better, you have overlooked something.
- If project content is allowed to change freely, the rate of change will exceed the rate of progress.

# Cynical Views or Reality?

- Projects progress quickly until they are 90% complete. Then they remain at 90% complete forever.
- When things are going well, something will go wrong. When things just can't get worse, they will. When things appear to be going better, you have overlooked something.
- If project content is allowed to change freely, the rate of change will exceed the rate of progress.
- Project teams detest progress reporting because it manifests their lack of progress.

# Software Project *Elements*

- All *technical* and *managerial* activities required to deliver the deliverables to the client.
- A software project has a specific duration, consumes resources and produces ***work products***.
- Management categories to complete a software project:
  - Tasks, Activities, Functions

# Software Project Management Plan (SPMP)

- The controlling document for a software project.
- Specifies the technical and managerial approaches to develop the software product.
- Companion document to **SRS**: Changes in either may imply changes in the other document.
- SPMP may be part of project agreement.

# Project Agreement

- Document written for a client that defines:
  - scope, duration, cost and deliverables for the project.
  - exact items, quantities, delivery dates, delivery location.
- Can be a contract, a statement of work (SOW), a business plan, or a project charter.
- Client: Individual or organization that specifies the requirements and accepts the project deliverables.
- Deliverables:
  - Software
  - Documents
  - Demonstrations of function
  - Demonstration of nonfunctional requirements
  - Demonstrations of subsystems

# Terms

- **Work product**: Something that needs to be done or get created in the project:
  - Applies to activities as well as tangible products

# Phases in the Unified Process

- Inception: Vision, requirements
- Elaboration: Design
- Construction: Coding, testing
- Transition: Delivery, testing

# Work-Products by Phase

- **Inception phase**
  - Business case/justification
  - Problem statement
  - Requirements Specification
  - Use cases
  - Risk analyses
  - Priorities
  - Acceptance plan

# Work-Products by Phase

- **Design phase**
  - System architecture
  - Various diagrams (class, sequence, …)
  - Documentation plan
  - Identify third-party components
  - Configuration management plan
  - Testing plan
  - Maintenance plan

# Work-Products by Phase

- **Construction phase**
  - Product source code
  - Test code
  - Documentation
    - Internal
    - User

# Work-Products by Phase

- **Transition phase**
  - The software
  - Installation instructions
  - User manual
  - Customer support

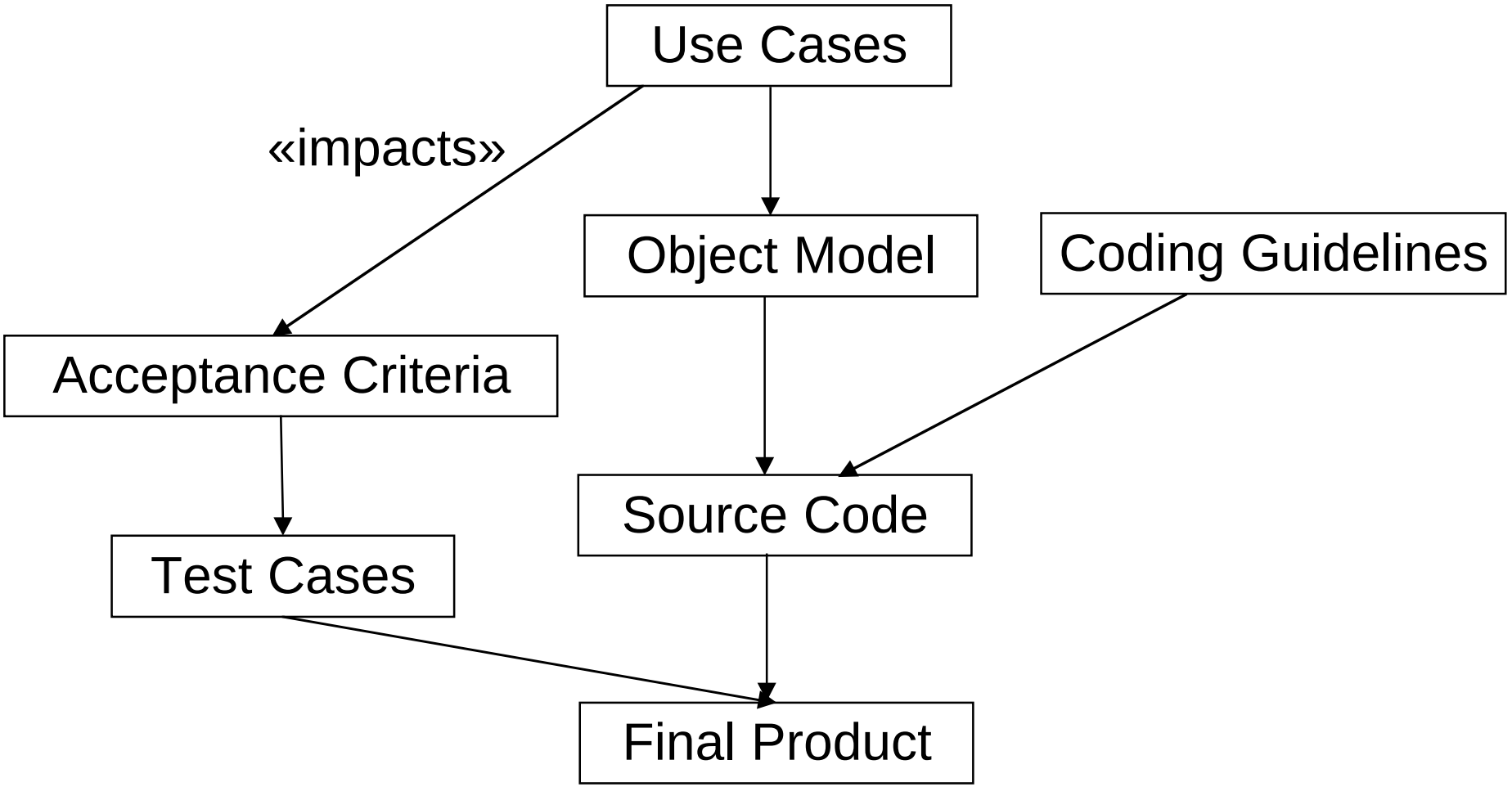
# Additional Work-Products

- **Management "products"**
  - Design and development processes
  - Resource plan
  - Schedule, milestones
  - Release plan
  - Quality assurance plan
  - Risk management plan

# Work-product Traceability Diagram

- Shows impacts of one work product on another

# Work-Product Traceability Diagram (partial)



# Project Management Terms

(not universal)

- **Tasks**: Small units of work
- **Activities**: Major work units
- **Milestones**: When activities are completed
- **Functions**: Pervasive, on-going management functions

# Tasks: Small units of work

- **Smallest** unit of management accountability
  - Atomic unit of planning and tracking
  - Finite duration, need resources, produce tangible result (documents, code)
- Specification of a task: Work package
  - Name, description of work to be done
  - **Preconditions** for starting, **duration**, required **resources**
  - Work product to be produced, acceptance criteria for it
  - Risk involved
- Completion criteria
  - Includes the acceptance criteria for the work products (deliverables) produced by the task.

# Activities: Major Units of Work

- Culminates in major project ***milestone***:
  - Internal checkpoint should not be externally visible
  - Scheduled event used to measure progress
- Milestone often produces baseline:
  - formally reviewed work product
  - under change control (change requires formal procedures)
- Activities may be grouped into larger activities:
  - Establishes hierarchical structure for project (phase, step, ...)
  - Allows separation of concerns
  - Precedence relations often exist among activities (PERT Chart)

# Examples of Activities

- Major Activities:
  - Planning
  - Requirements Elicitation
  - Requirements Analysis
  - System Design
  - Object Design
  - Implementation
  - System Testing
  - Delivery
- Activities during requirements analysis:
  - Refine scenarios
  - Define Use Case model
  - Define object model
  - Define dynamic model
  - Design User Interface

# Project “Functions”

- Examples:
  - Project management
  - Configuration Management
  - Documentation
  - Quality Control (Verification and validation)
  - Training

- Project Functions in the IEEE 1058 standard are called ***Integral processes*** in the IEEE 1074 standard.

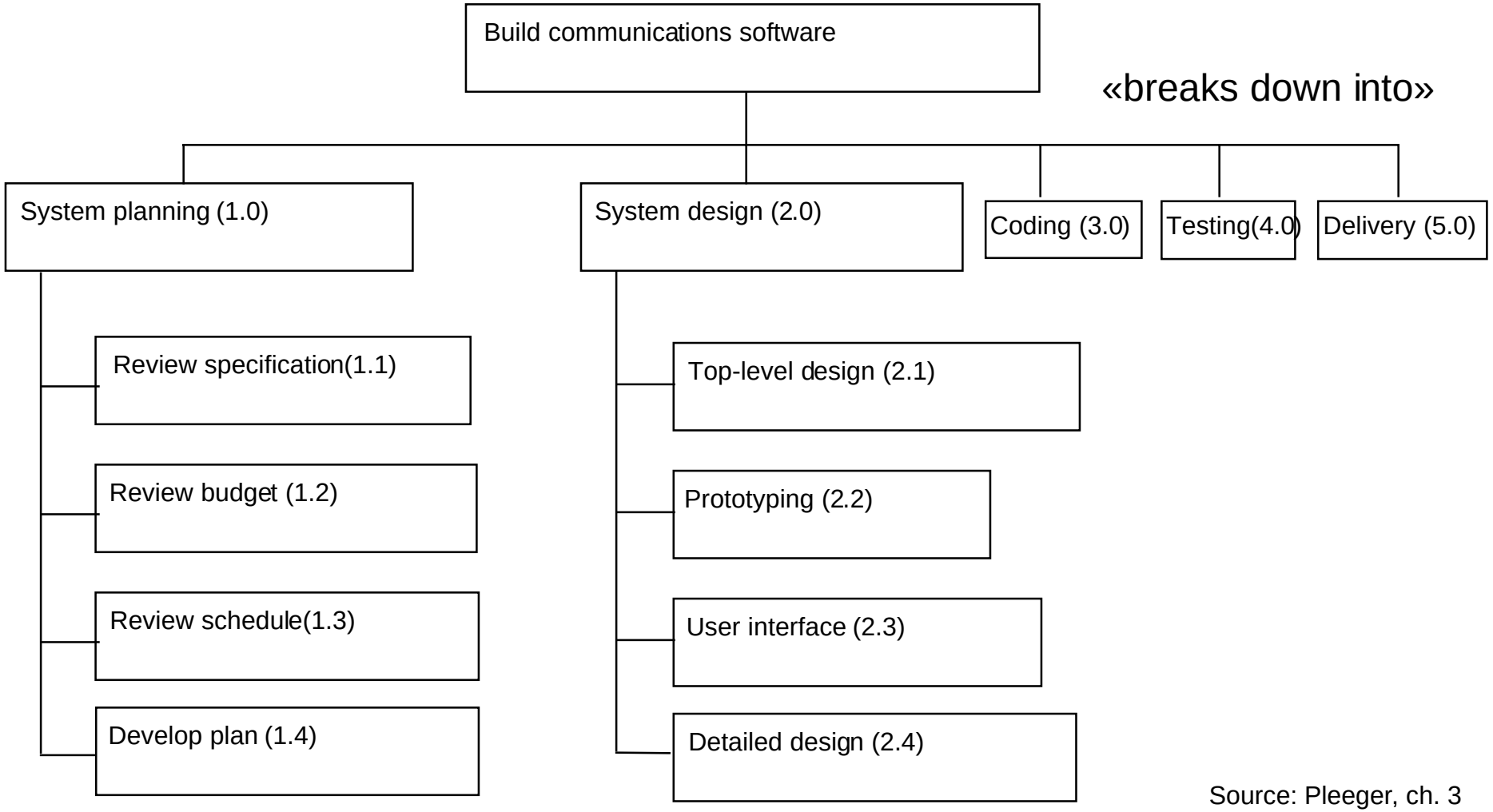
# Organizational Diagrams

- **Work breakdown structure (WBS)**: Shows **hierarchy** of work products
- **PERT chart**: Shows the **order** in which activities must be done (a partial order)
- **Gantt Chart or Schedule**: Shows scheduling of work products as a function of time

# Creating Work Packages

- Work Breakdown Structure (WBS)
  - Break up project into **activities** (phases, steps) and **tasks**.
  - *The work breakdown structure does **not** show the interdependence of the tasks*

# Work Breakdown Structure (WBS) Diagram




Source: Pleeger, ch. 3

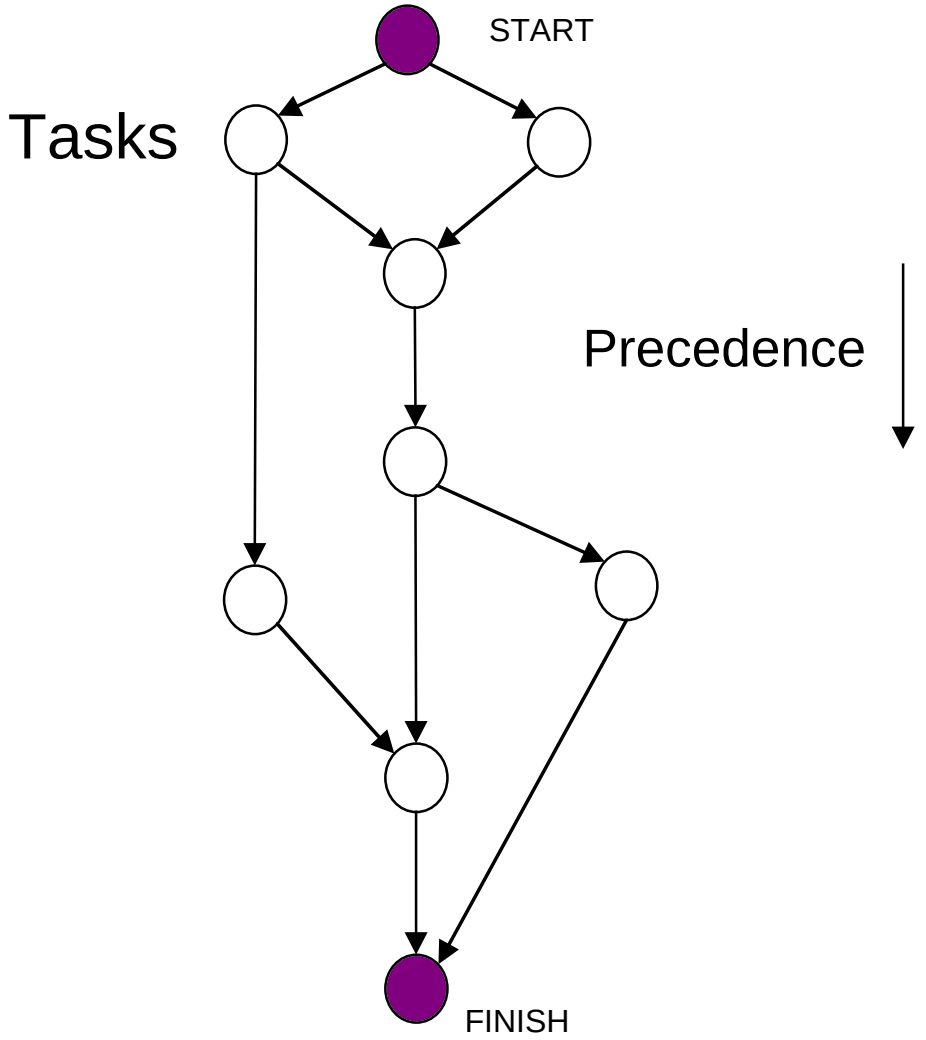
# WBS Tradeoffs

- Work breakdown structure influences cost and schedule
- Determination of work breakdown structure may be **incremental** and iterative

# PERT Chart

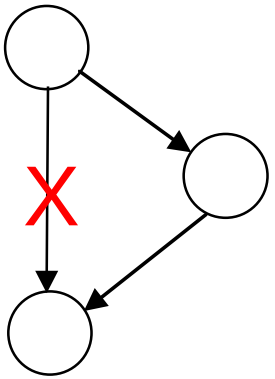
- “Program Evaluation and Review Technique”
- U.S. Navy, 1957

# PERT chart
# (Similar to UML Activity Diagram)



# To Build a PERT chart

- List all the tasks.
- For each tasks:
  - List the tasks that must (immediately) precede that task
  - Remove *implied* dependencies (called the "**transitive reduction**" of the graph)



# Tasks in Building a House: What are likely required orders?

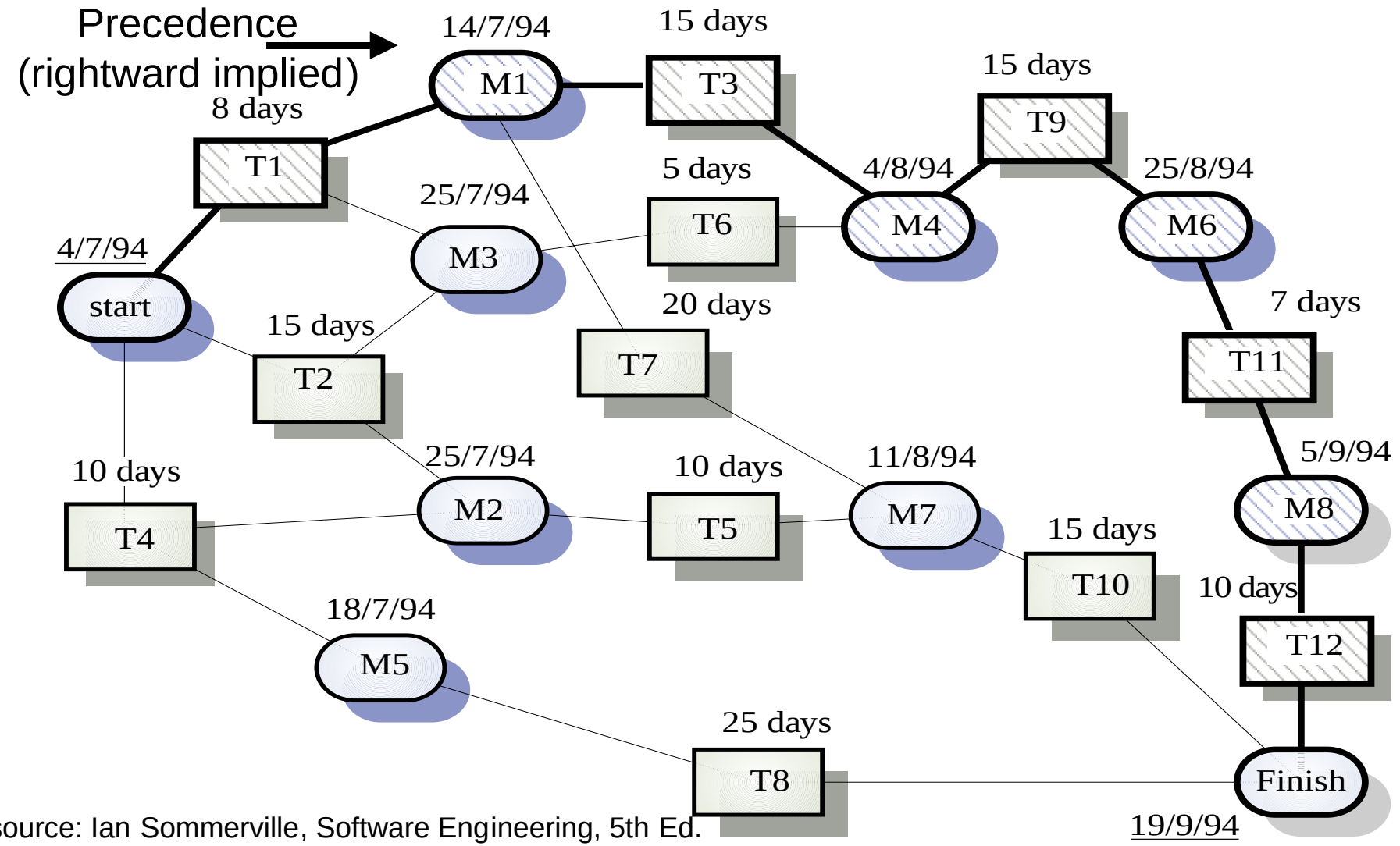
- Install roofing
- Install interior electrical
- Buy materials
- Excavate
- Build outside wall
- Install interior plumbing
- Install exterior siding
- Paint interior
- Install flooring
- Lay foundation
- Obtain permits
- Install exterior electrical
- Install exterior doors and fixtures
- Paint exterior
- Survey land
- Install interior doors and fixtures
- Install wallboard
- Install exterior plumbing

# Task durations and dependencies

| Task | Duration (days) | Dependencies |
|---|---|---|
| T1 | 8 | |
| T2 | 15 | |
| T3 | 15 | T1 |
| T4 | 10 | |
| T5 | 10 | T2, T4 |
| T6 | 5 | T1, T2 |
| T7 | 20 | T1 |
| T8 | 25 | T4 |
| T9 | 15 | T3, T6 |
| T10 | 15 | T5, T7 |
| T11 | 7 | T9 |
| T12 | 10 | T11 |

source: Ian Sommerville, Software Engineering, 5th Ed.

# Task network with Milestones (ovals)




source: Ian Sommerville, Software Engineering, 5th Ed.

# PERT -> Gantt

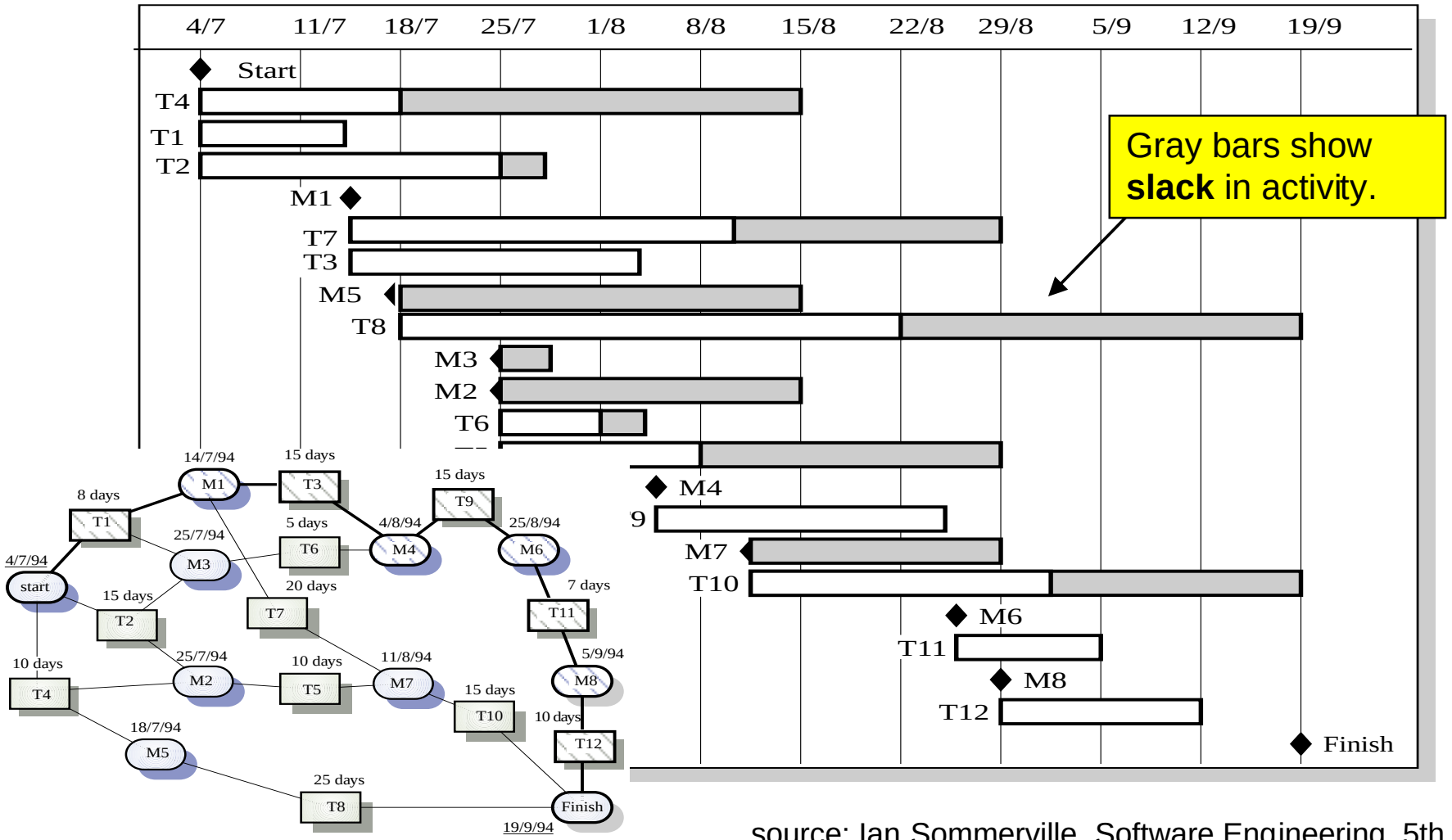
- As we saw, PERT enforces certain **precedence constraints.**

- Other kinds of constraints:
  - **Staff assignment** (task -> person) and loading constrains number of activities that can go on concurrently.
  - **Resource availability** further constrains time at which various activities can happen.

- The **Gantt chart** (or schedule) shows a **particular** scheduling of the tasks, subject to all of the constraints.

# Gantt Chart from PERT (inset)

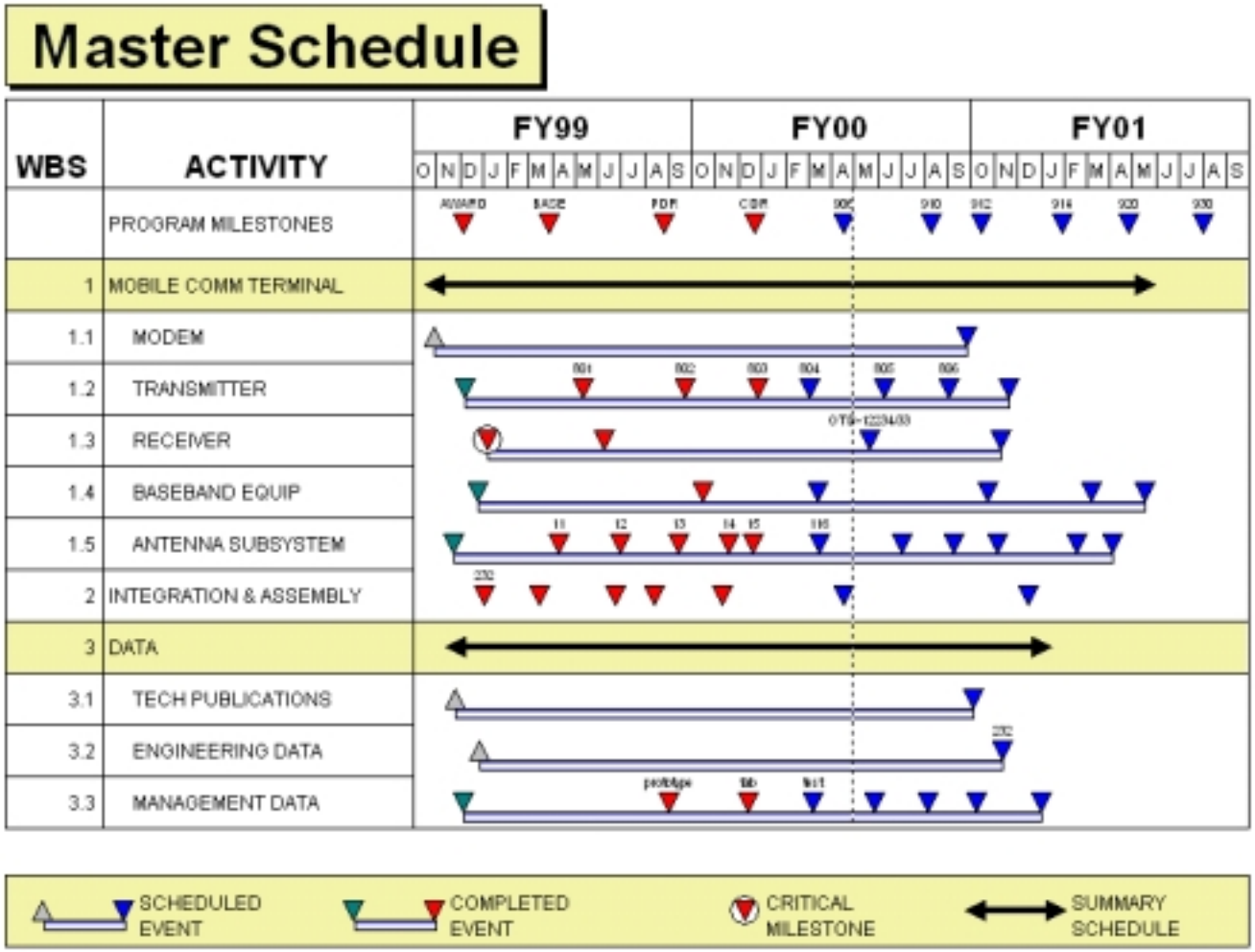
## (Timeline chart, Timing chart, Schedule)




source: Ian Sommerville, Software Engineering, 5th Ed.

# Real-life Gantt Chart

(source: http://www.kidasa.com/information/examples/aerospace/aerospace2.html)



# Henry L. Gantt (1861-1919)

(http://accel-team.com/scientific/scientific_04.html)

- well-known pioneer in the early days of scientific management
- remembered for his humanizing influence on management, emphasizing the conditions that have favorable psychological effects on the worker
- “The Gantt chart for which he will be remembered, is a visual display chart used for scheduling, which is based on time, rather than quantity, volume or weight.”

# Critical Path

- A “critical path” on a PERT chart is one in which the **sum of durations** of the tasks on the path equals the **shortest** overall project-completion time.
- The critical path is shaded on the preceding diagram.
- PERT is also called CPM (Critical Path Method)

# What is the Critical Path in Building your House, given the number of days required as shown?

- Install roofing (4)
- Install interior electrical (2)
- Buy materials (5)
- Excavate (7)
- Build outside wall (5)
- Install interior plumbing(4)
- Install exterior siding (3)
- Paint interior (2)
- Install flooring (2)
- Lay foundation (2)
- Obtain permits (24)
- Install exterior electrical (3)
- Install exterior doors and fixtures (2)
- Paint exterior (1)
- Survey land (1)
- Install interior doors and fixtures (2)
- Install wallboard (2)
- Install exterior plumbing (1)

# Exercise

- List as many ways you can for **shortening** the critical path in a set of tasks.

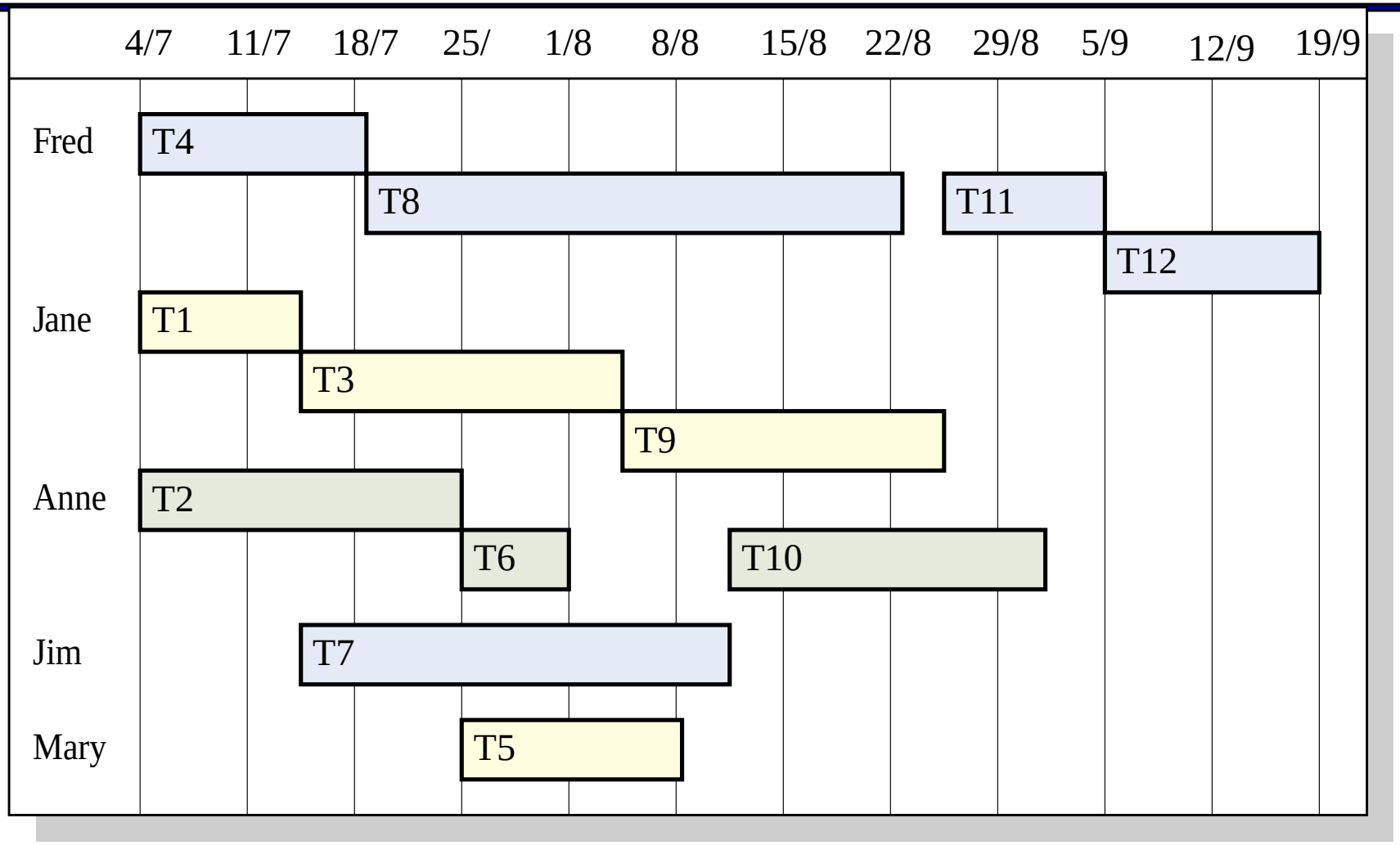
# Staff Loading & Resources

- Staff loading and resource constraints are two aspects of scheduling not represented directly on PERT charts
- Generally they have the effect of providing added sequencing, and therefore lengthening overall project time

# Staff Loading & Resources

- Staff loading refers to:
  - Some tasks can be assigned only to certain staff members, based on specialized skills.
  - A given staff member can only do so much at a time.
  - Staff members differ in their productivity on a given task.

# Staff loading




source: Ian Sommerville, Software Engineering, 5th Ed.

# Staff Loading & Resource Constraints

- Resource constraints refers to certain resources being needed for certain tasks, but being limited in the number of tasks they can support concurrently.
- Resources include:
  - Machines, workstations, storage units
  - Rooms, offices

# How Do Constraints Differ from Precedence Relationships?

# How Do Constraints Differ from Precedence Relationships?

- Precedence relationships specify a particular sequencing between two tasks.

# How Do Constraints Differ from Precedence Relationships?

- Precedence relationships specify a particular sequencing between two tasks.

- Constraints don't specify a particular sequence, but only that two tasks can't be done simultaneously.

# Tools

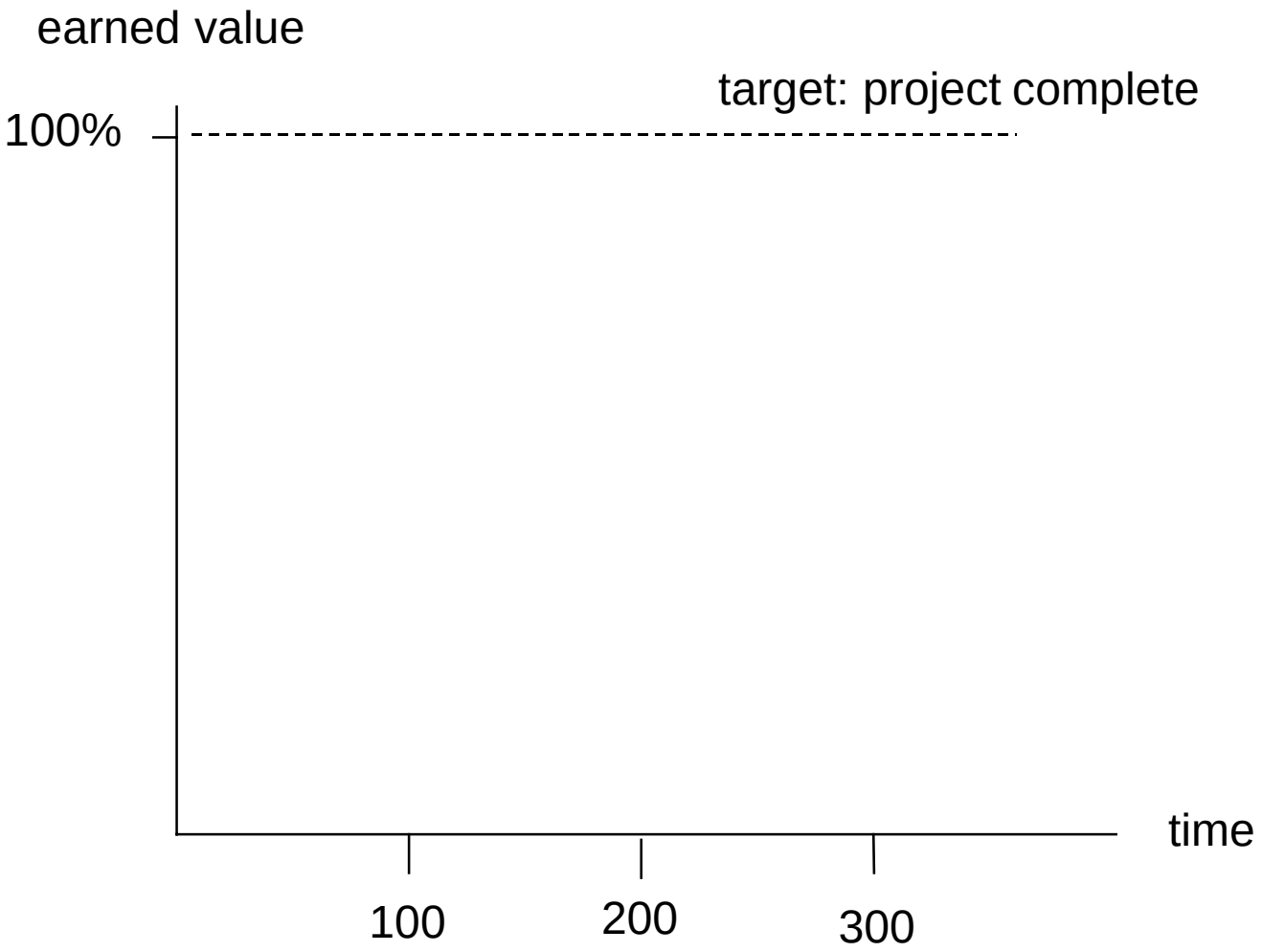
- Microsoft Project
  - Handles WBS
  - Handles PERT to Gantt
  - Does not handle resources
  - Sort of handles staff loading, except that a single staff member can be overloaded.

# Tracking How Well are We Doing

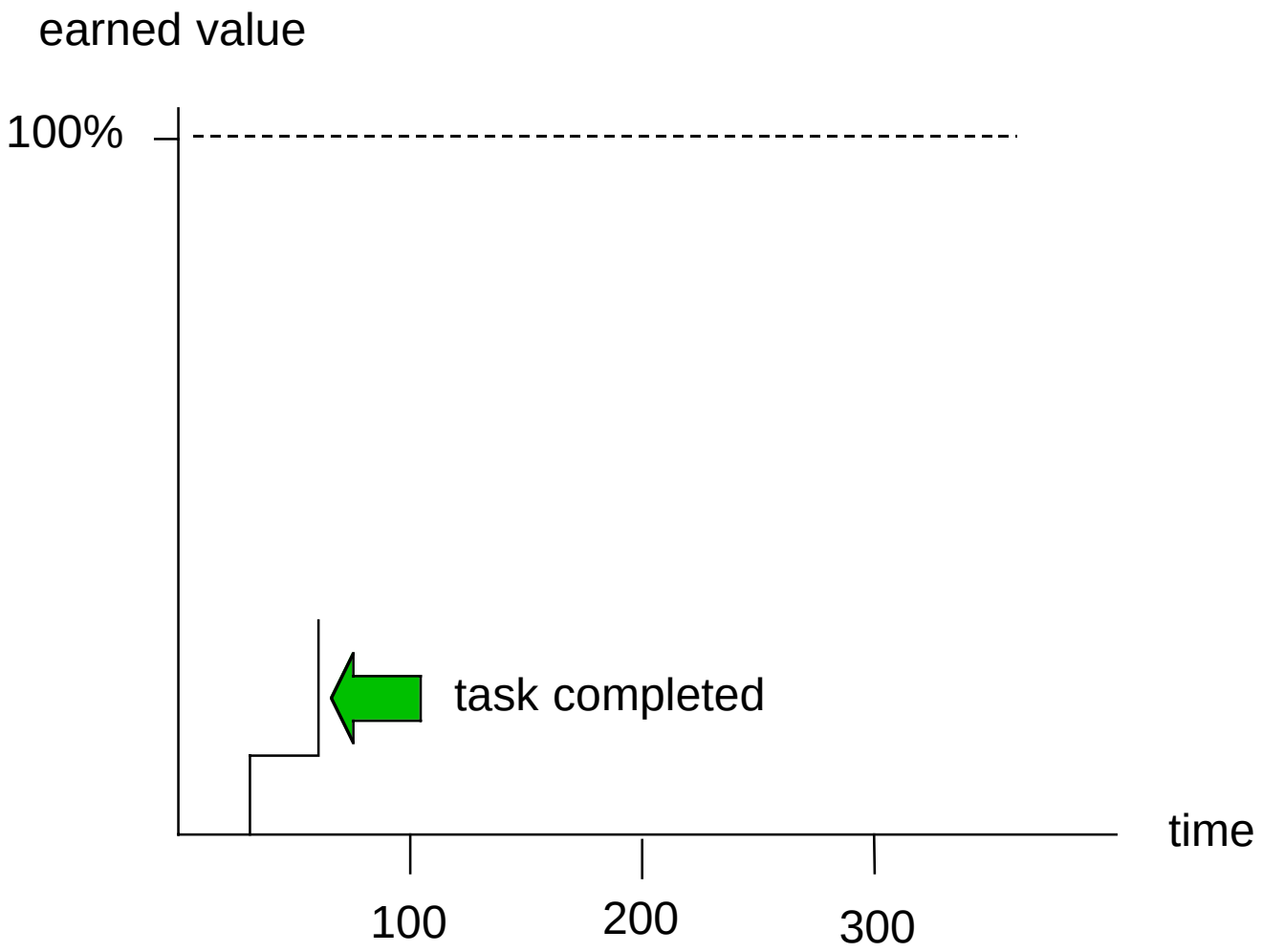
# Earned-Value Tracking Method/Chart

- One way to track how close to “done” the project is:
  - As key parts of a product are **completed**, the product “earns value”.
  - Express earned value in % of total value or $ (= % x budgeted amount for task)
  - Some variations allow **partial credit**, others don’t.

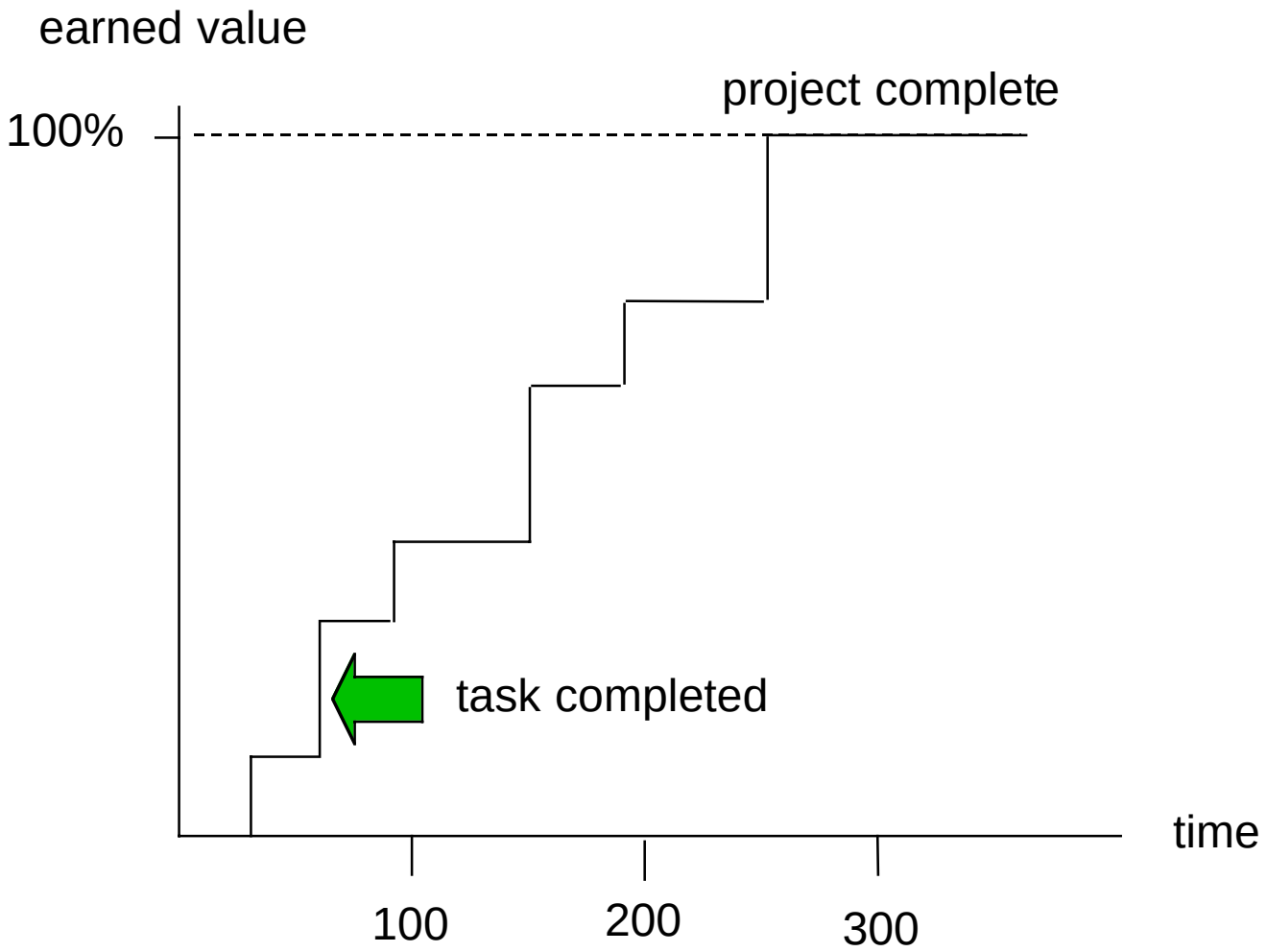
# Earned Value vs. Time



# Earned Value vs. Time



# Earned Value vs. Time



# Use of Earned Value Diagram

# Use of Earned Value Diagram

- Based on project Gantt chart, create a profile **plan** of earned value

# Use of Earned Value Diagram

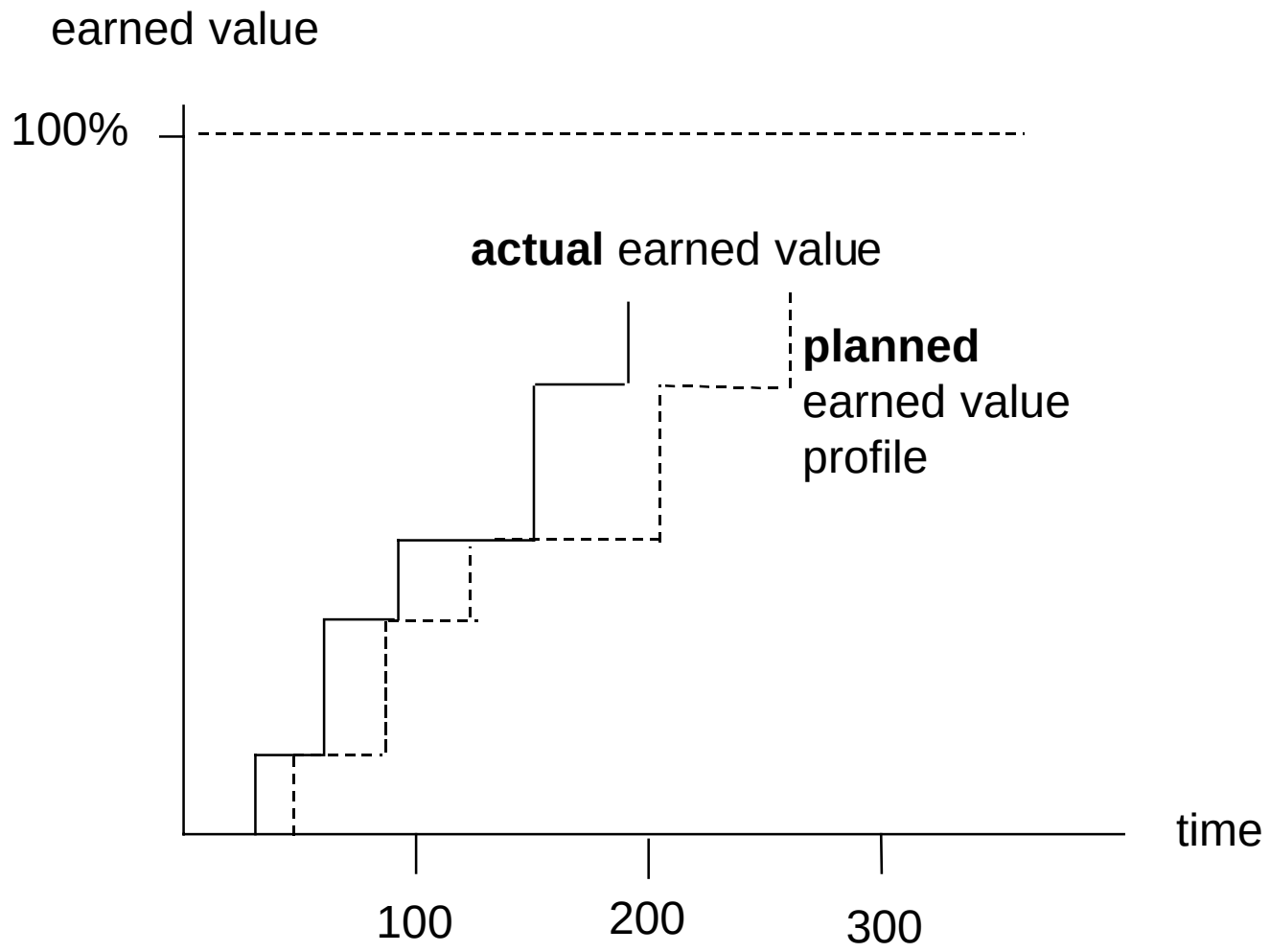
- Based on project Gantt chart, create a profile **plan** of earned value

- Track **actual** earned value against plan

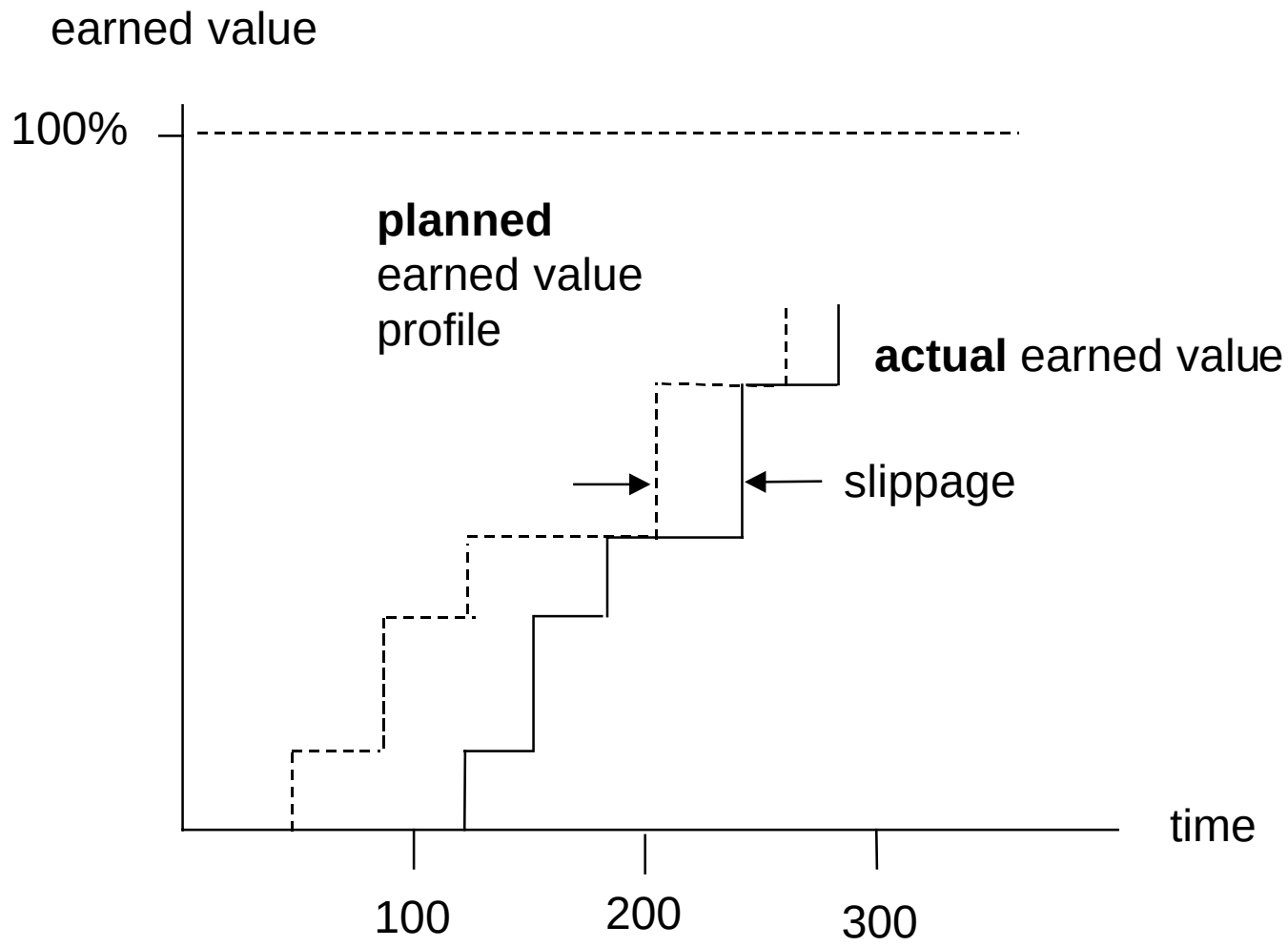
# Use of Earned Value Diagram

- Based on project Gantt chart, create a profile **plan** of earned value
- Track **actual** earned value against plan
- Use **discrepancies** to make projections about delivery dates, cost, etc.

# Project Ahead of Schedule



# Project Behind Schedule

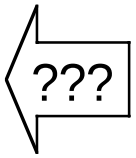


# Earned Value with **sub-tasks** in $ (% x budget)

**Earned Value Calculation Example**

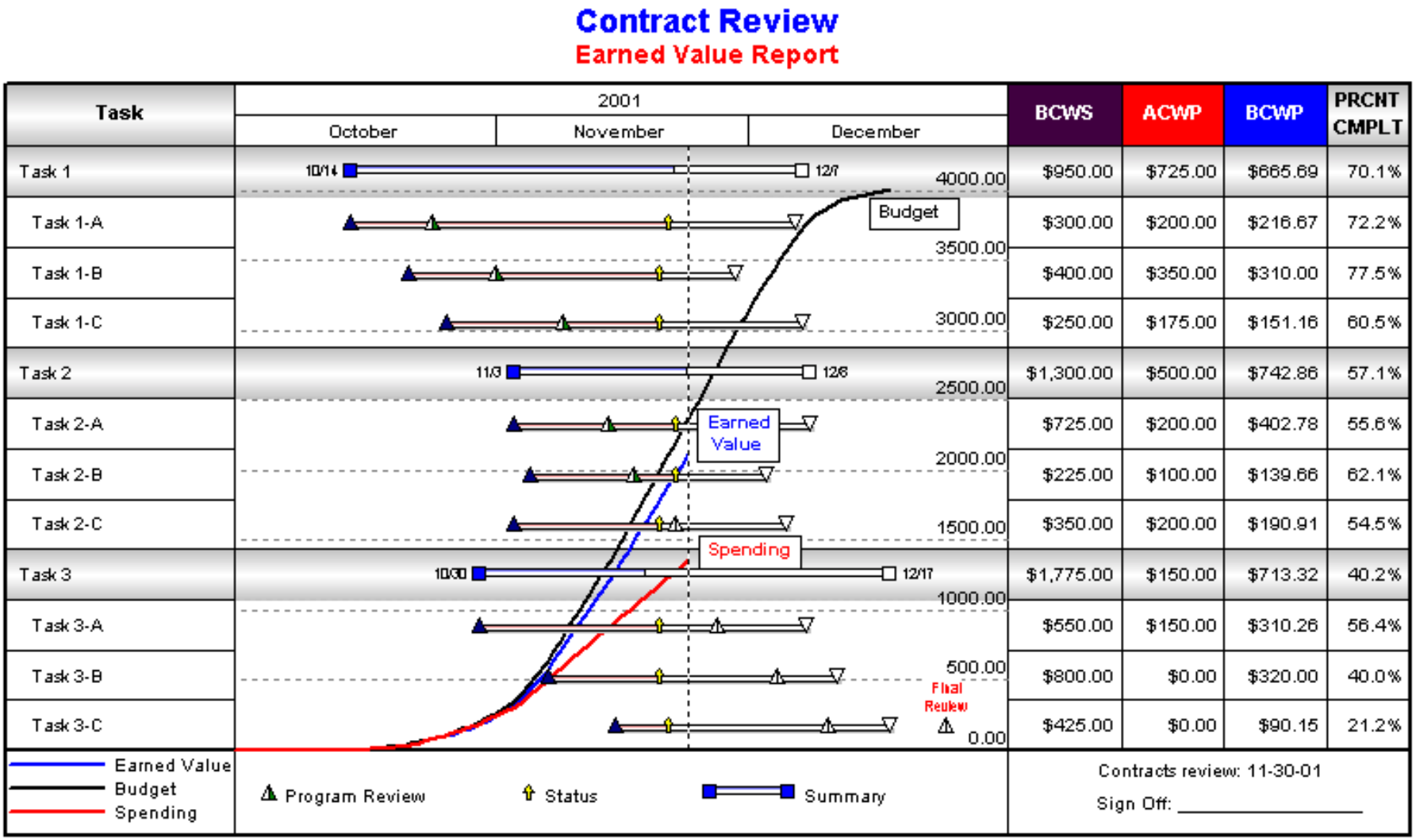
| Task | 2000 | Budget | Percent Complete | Earned Value |
|---|---|---|---|---|
| | Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec | | | |
| Task 1 | | | 100.0% | $0.00 |
| Task1 -1 | | $333.00 | 100.0% | $333.00 |
| Task 1-2 | | $444.00 | 100.0% | $444.00 |
| Task 1-3 | | $555.00 | 100.0% | $555.00 |
| Task 1-4 | | $55.00 | 100.0% | $55.00 |
| Task 2 | | | 80.1% | $0.00 |
| Task 2-1 | | $434.00 | 100.0% | $434.00 |
| Task 2-2 | | $333.00 | 65.0% | $216.45 |
| Task 2-3 | | $111.00 | 100.0% | $111.00 |
| Task 2-4 | | $333.00 | 50.0% | $166.50 |
| Task 3 | | | 56.7% | $0.00 |
| Task 3-1 | | $444.00 | 70.0% | $310.80 |
| Task 3-2 | | $5,555.00 | 67.5% | $3,747. |
| Task 3-3 | | $666.00 | 37.0% | $246.42 |





# Example Gantt Chart with Earned Value Overlay

(source: http://www.kidasa.com/information/solutions/evalue/index.html)

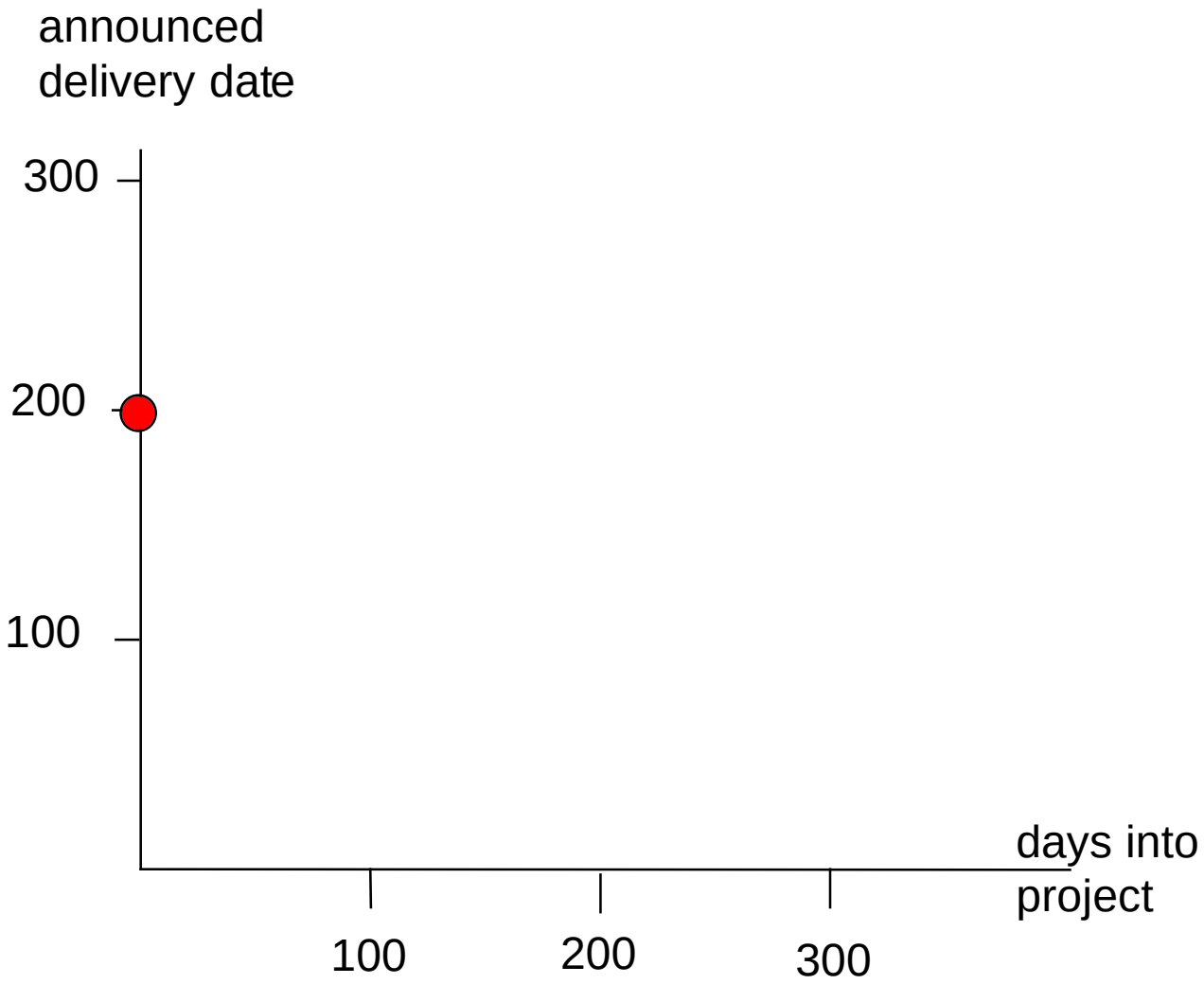


# Implication of Slippage

- Slippage in earned value may imply slippage in delivery date,

  especially if the slipped task is on the critical path.

# Slip Chart


announced
delivery date
300
200
100
days into
project
100
200
300

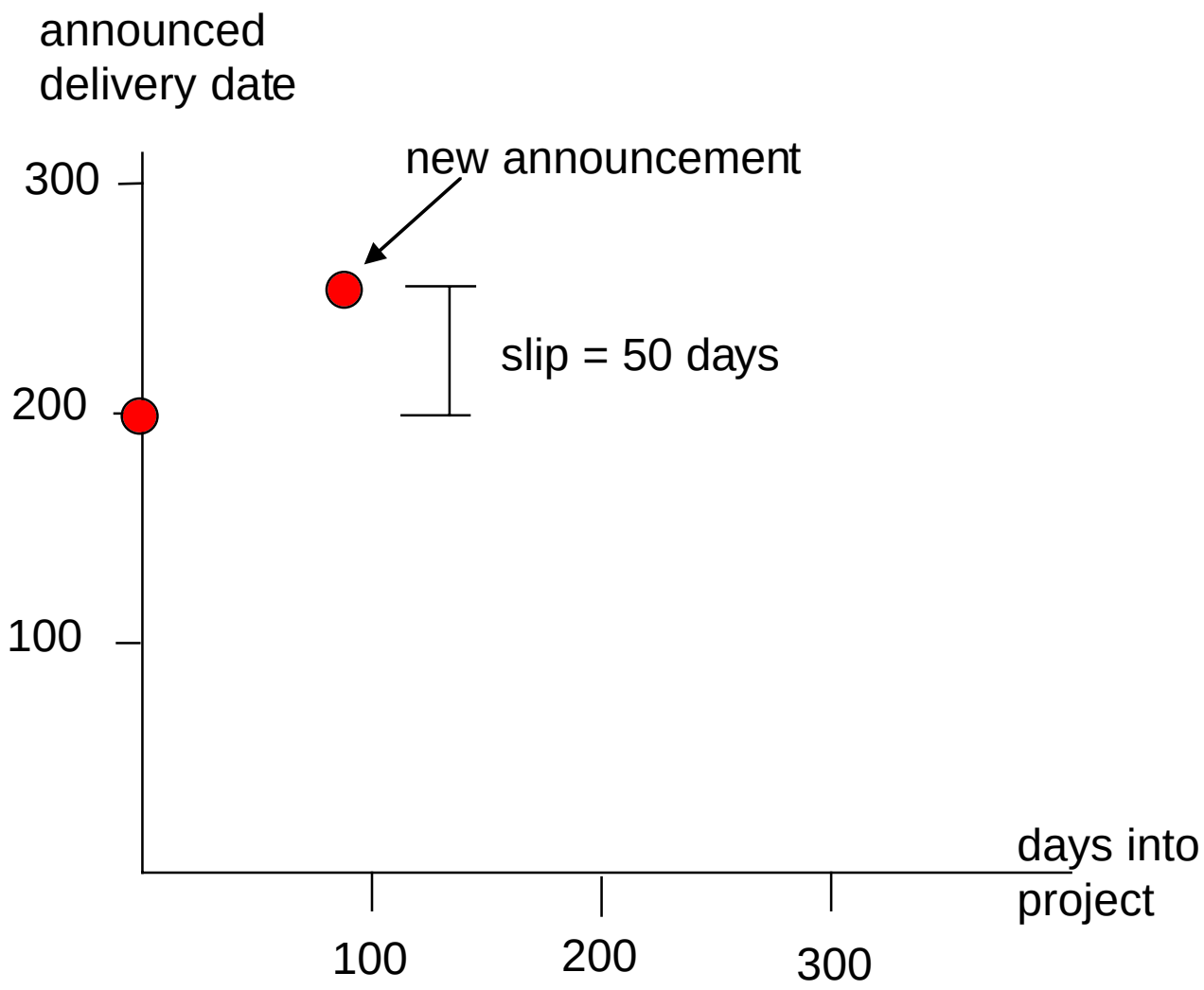
# Slip Chart

slip = delivery date

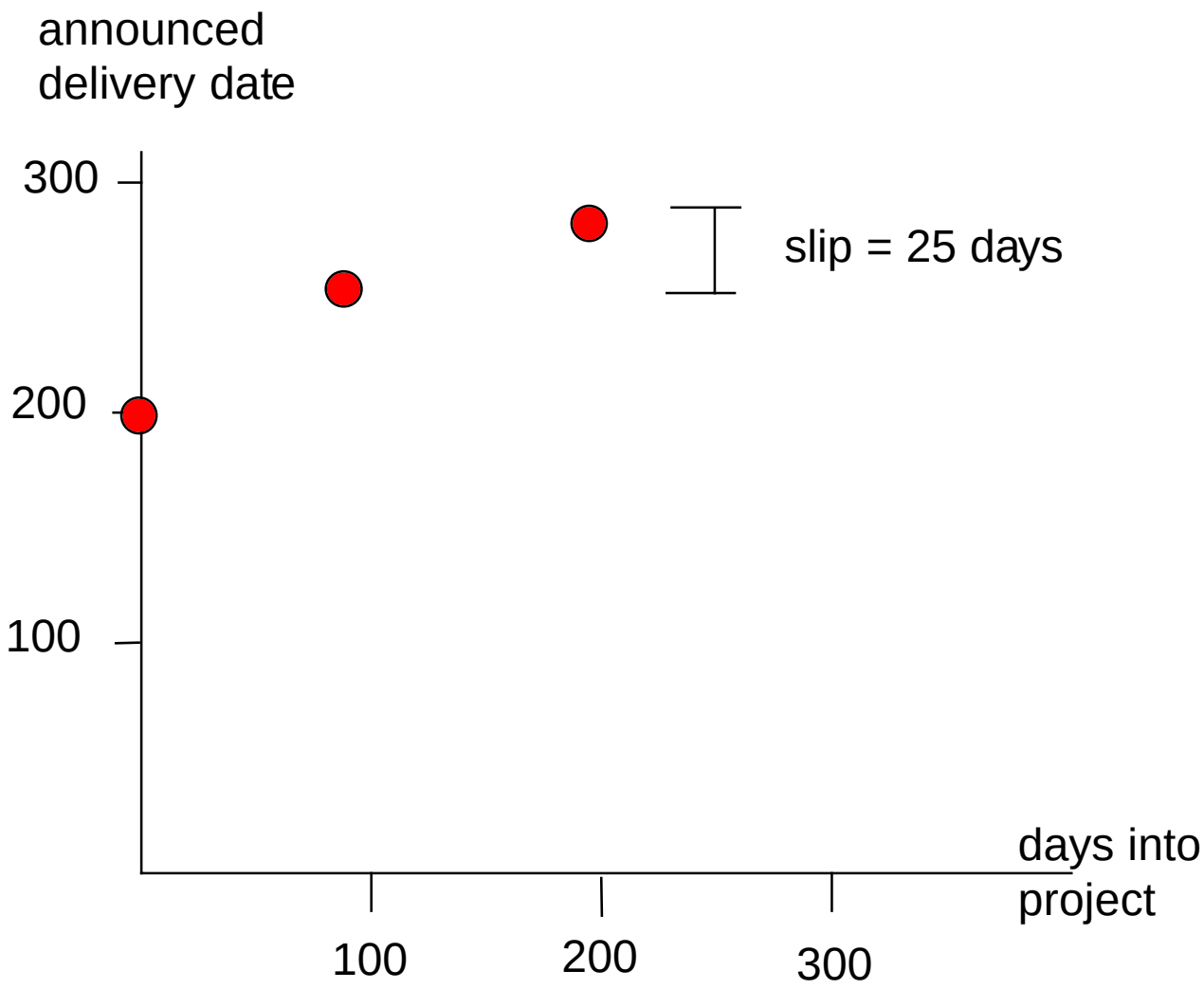
minus

previously announced

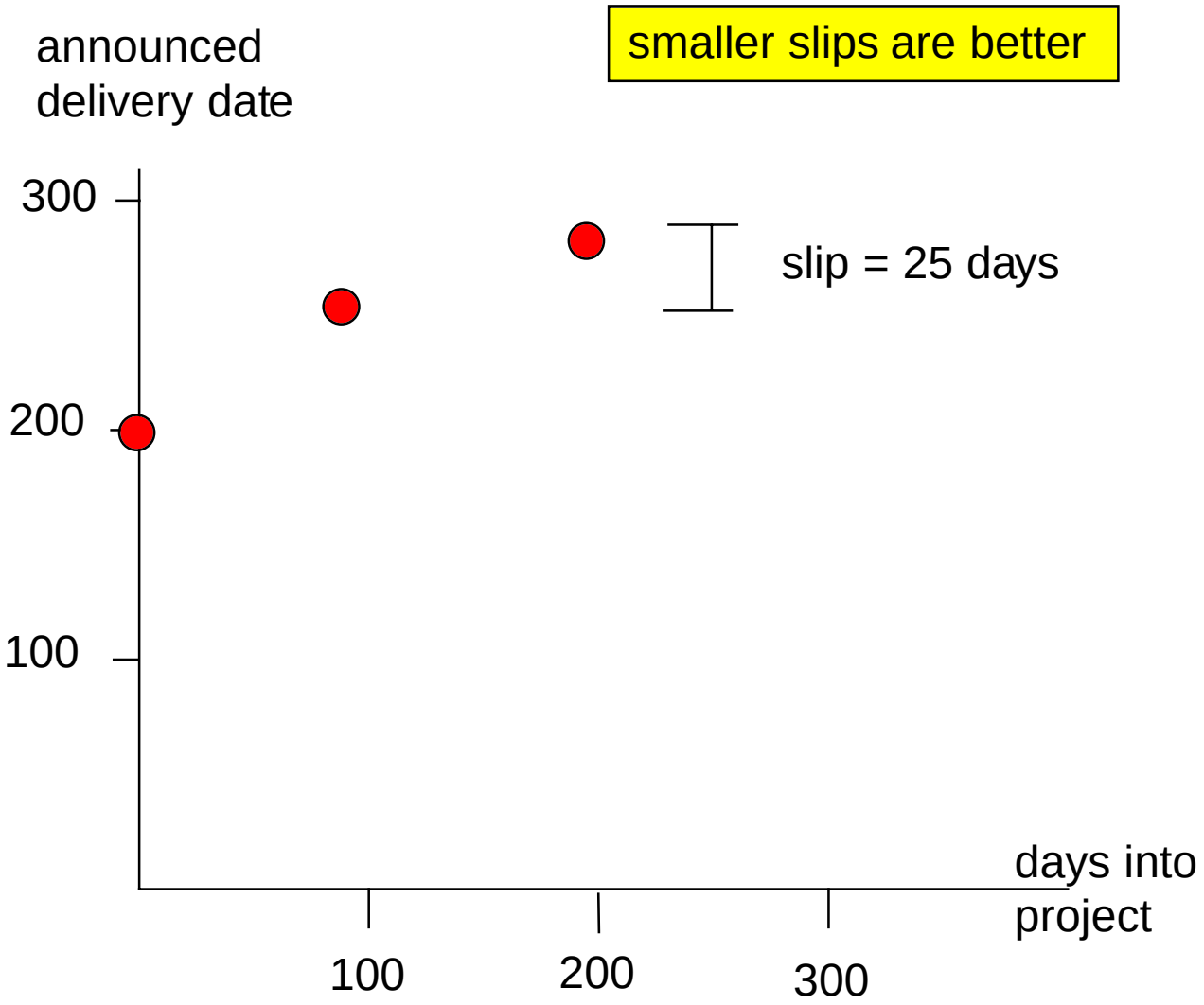
delivery date



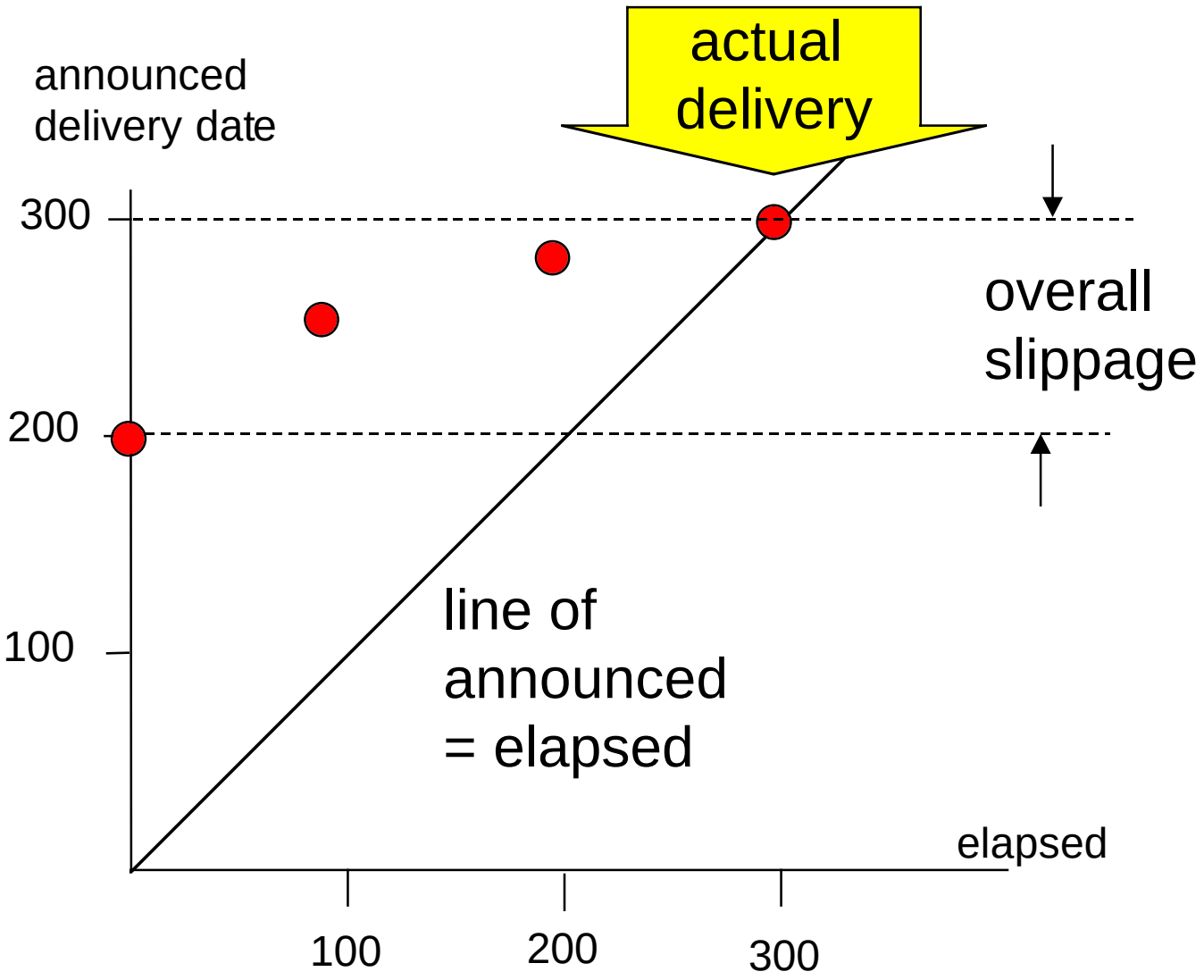
# Slip Chart


announced
delivery date
300
200
100
slip = 25 days
days into
project
100
200
300

# Slip Chart


announced
delivery date
smaller slips are better
300
200
100
slip = 25 days
days into
project
100
200
300

# Slip Chart

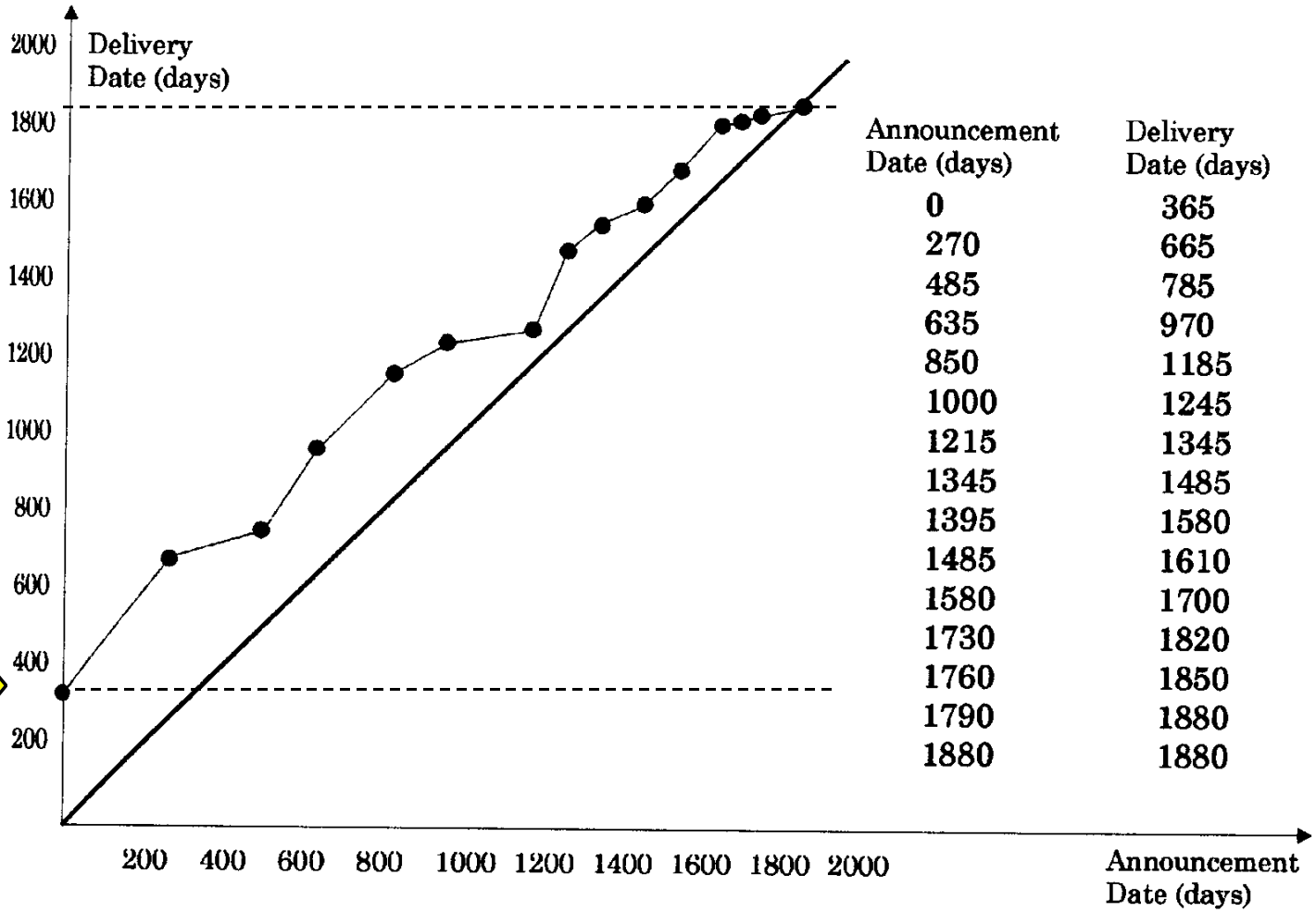

actual
delivery
announced
delivery date
300
200
100
overall
slippage
line of
announced
= elapsed
elapsed
100
200
300

# Word for Windows 1.0 Slip

actual

original

1880
365
= 515%
net slip




| Announcement Date (days) | Delivery Date (days) |
|---|---|
| 0 | 365 |
| 270 | 665 |
| 485 | 785 |
| 635 | 970 |
| 850 | 1185 |
| 1000 | 1245 |
| 1215 | 1345 |
| 1345 | 1485 |
| 1395 | 1580 |
| 1485 | 1610 |
| 1580 | 1700 |
| 1730 | 1820 |
| 1760 | 1850 |
| 1790 | 1880 |
| 1880 | 1880 |

source: Dwayne Phillips, The software project manager's handbook, IEEE, 1998.

# Slip vs. Lead

- slip = delivery date
  minus
  previously announced delivery date

- lead = previously announced delivery date
  minus
  date on which new delivery announced

# Example of Lead

- lead = previously announced delivery date
  minus
  date on which new delivery announced

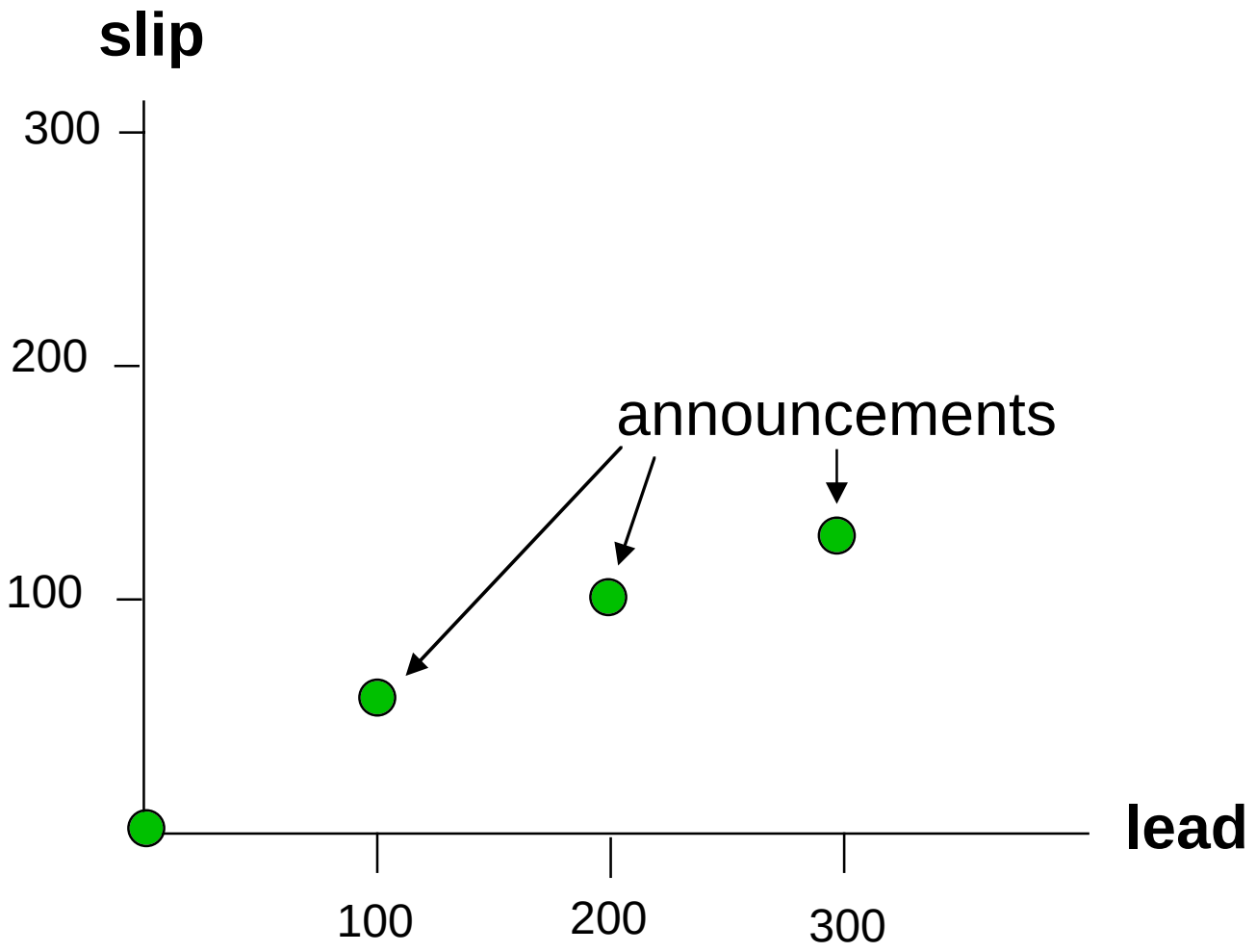
- Example:
  - Original delivery date = day **200**
  - On day **100** announce new delivery date:
    day 300: ***moderate* lead**

# Example of Lead

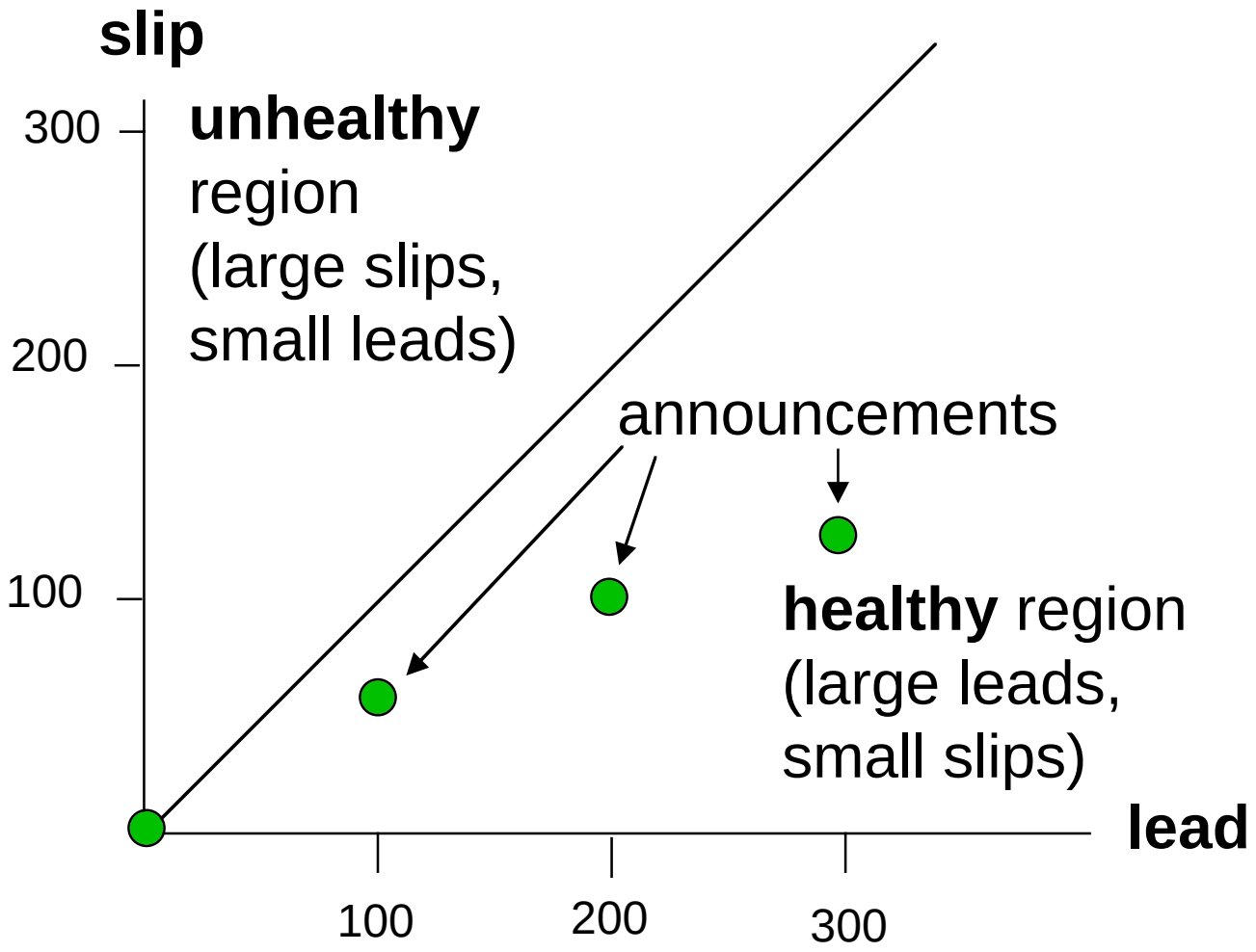
- lead = previously announced delivery date
  minus
  date on which new delivery announced

- Original delivery date = day **200**
- On day **195** announce new delivery date:
  day 300: very **small lead**
- **Small leads are bad** (but small slips are good)

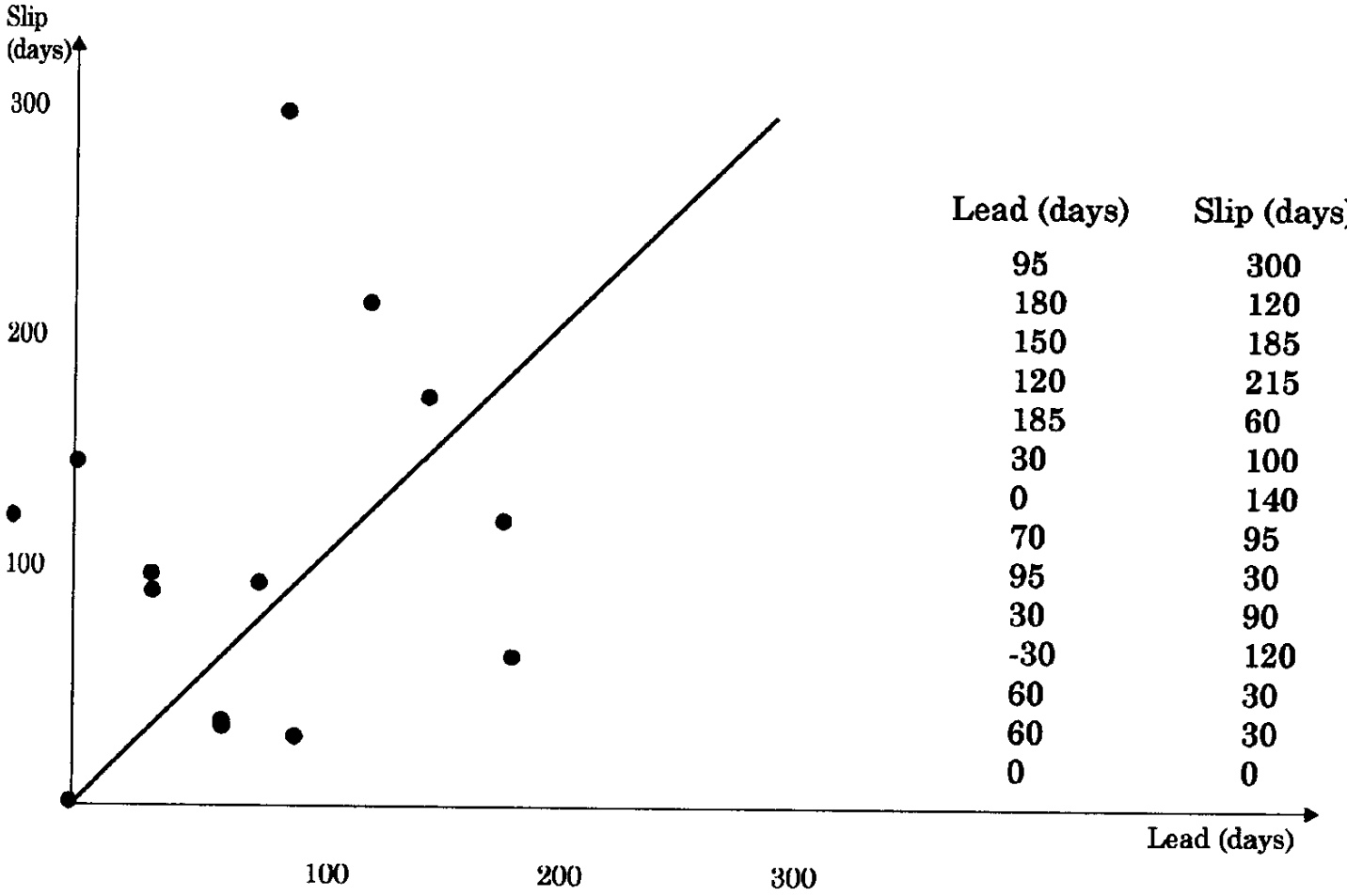
# Slip-Lead Chart
# (plots slips vs. leads)

# Slip-Lead Chart
# (plots slips vs. leads)



# Word for Windows 1.0 Slip/Lead




| Lead (days) | Slip (days) |
|---|---|
| 95 | 300 |
| 180 | 120 |
| 150 | 185 |
| 120 | 215 |
| 185 | 60 |
| 30 | 100 |
| 0 | 140 |
| 70 | 95 |
| 95 | 30 |
| 30 | 90 |
| -30 | 120 |
| 60 | 30 |
| 60 | 30 |
| 0 | 0 |

source: Dwayne Phillips, The software project manager's handbook, IEEE, 1998.