# Project Management

Lecture 2a

**Network Planning Techniques: CPM-PERT**

Instructor

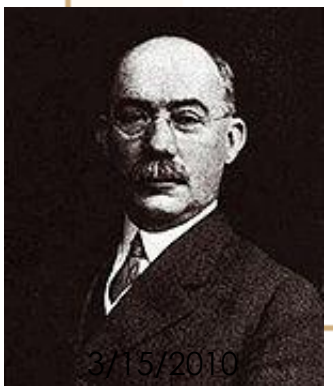
Carmi Bogot

# Agenda

- PM methods and tools
- CPM
- Critical Paths, Slack
- Probabilistic Task Times
- Task "Crashing" and Cost
- Conclusions and Discussion
- HW 1

# History of project management

- Big Projects
  - Pyramids (we built them)
  - Great Wall
  - Large workforce but no documented project management
- Formal Project Management
  - Henry Gantt (1861-1919) bar chart
  - 1957 Sputnik Crisis "Scientific Management"
  - Project Evaluation and Review Technique (PERT)
  - DuPont 1960 : Critical Path Method (CPM)
  - 1960 NASA: Mercury, Gemini, Apollo
    - Work breakdown Schedules (WBS)
    - Cost and Schedule tracking , configuration management

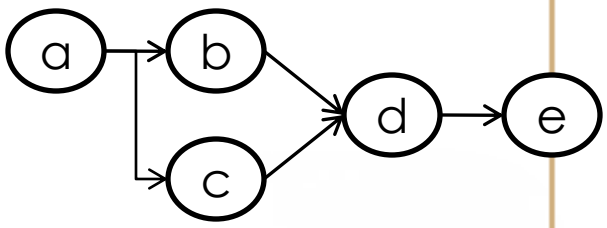


# Early Project Management

- Project Decomposition due to complexity
- Resource allocation and workload smoothing
- Schedule urgency *("within next 10 years JFK")*
- Causes
  - Complex relations between Government and Contractors
  - Cold war pressures for Nuclear Power, Space Race
- Other innovations
  - Project Manager as a central figure
  - Beginnings of Matrix Organization
- Professionalization since 1969
  - Other industries: computer, automotive
  - Project Management Institute (PMI)
  - ISO 10006: 1997 Quality in Project Management

# Fundamental Approaches

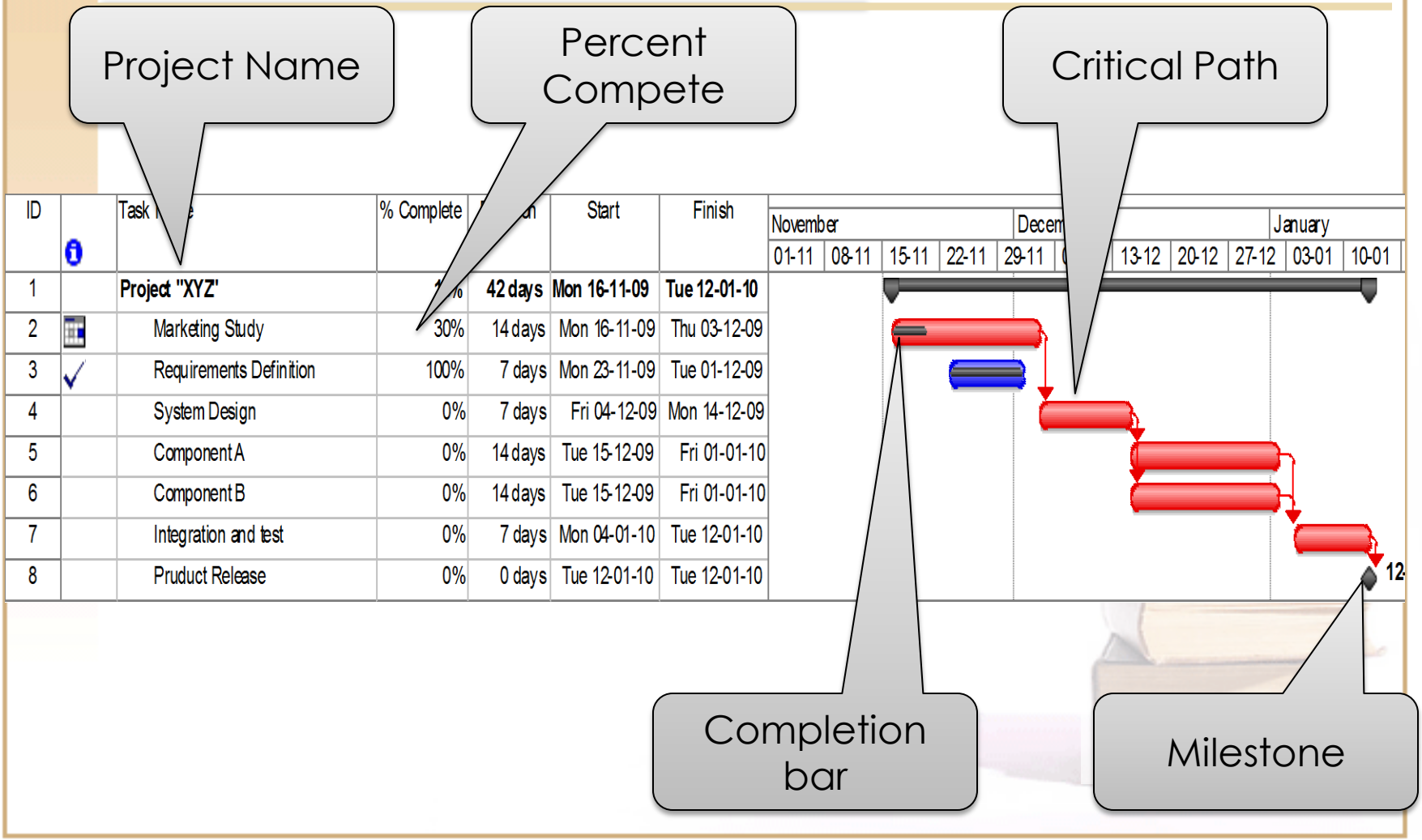
- How to represent task relationships
- Network Based methods
  - CPM, PERT
  - Task is a node or an arc
- Matrix – Based
  - DSM (design Structure Matrix)
- System Dynamics
  - Feedback loops, causal relationships
  - Stock and flows simulation
  - Tasks that are done or waiting are "stocks"
  - Doing project work causes a "flow"




# Gantt Charts

- Attributed to Henry Gantt
  - most popular tool (80%)
- Used to plan big shipbuilding projects (cargo ships WWI)
- Graphical way of showing task durations, project schedule
- Graphical way of showing task durations, and schedule
- Does not explicitly show relationships between tasks
- Limited use for project tracking
- Easy to understand

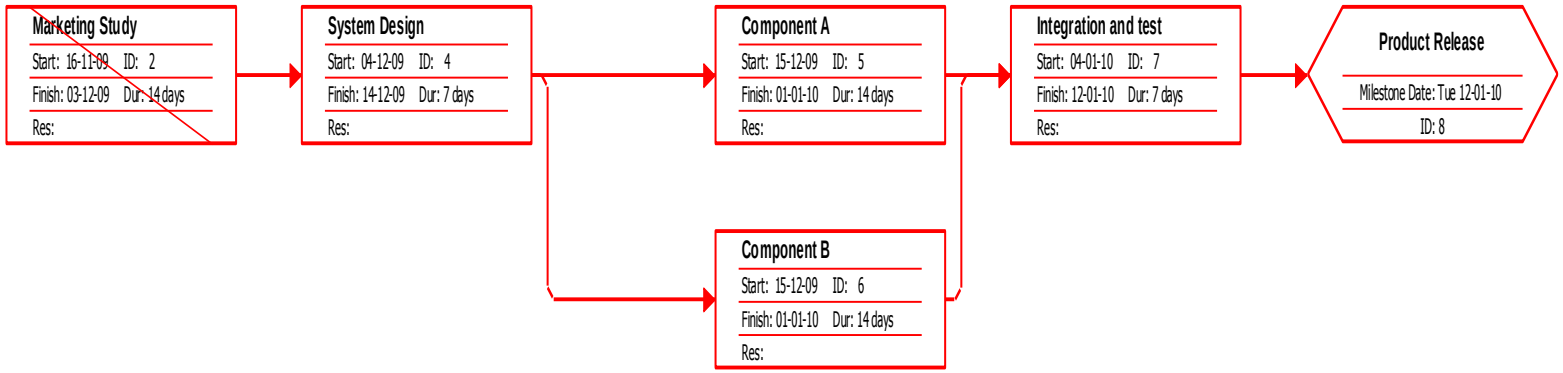
# Example Gantt




| ID | | Task Name | % Complete | Duration | Start | Finish |
|---|---|---|---|---|---|---|
| 1 | | **Project "XYZ"** | **1%** | **42 days** | **Mon 16-11-09** | **Tue 12-01-10** |
| 2 | | Marketing Study | 30% | 14 days | Mon 16-11-09 | Thu 03-12-09 |
| 3 | ✓ | Requirements Definition | 100% | 7 days | Mon 23-11-09 | Tue 01-12-09 |
| 4 | | System Design | 0% | 7 days | Fri 04-12-09 | Mon 14-12-09 |
| 5 | | Component A | 0% | 14 days | Tue 15-12-09 | Fri 01-01-10 |
| 6 | | Component B | 0% | 14 days | Tue 15-12-09 | Fri 01-01-10 |
| 7 | | Integration and test | 0% | 7 days | Mon 04-01-10 | Tue 12-01-10 |
| 8 | | Pruduct Release | 0% | 0 days | Tue 12-01-10 | Tue 12-01-10 |

# Critical Path Mapping

- Represent a project (set of tasks) as a network using graph theory
  - Capture durations
  - Capture dependencies

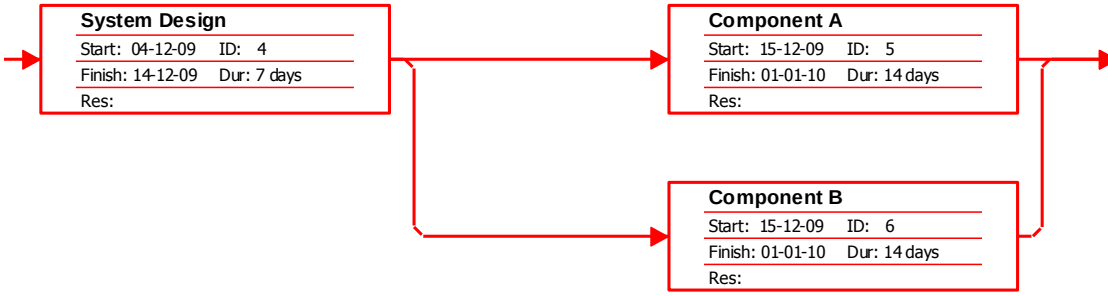


# CPM Assumptions

- Project consists of well defined tasks
- Project ends when all tasks are done
- Jobs may be started and stopped independently
- Jobs are ordered "technological sequence"

# Graph representations

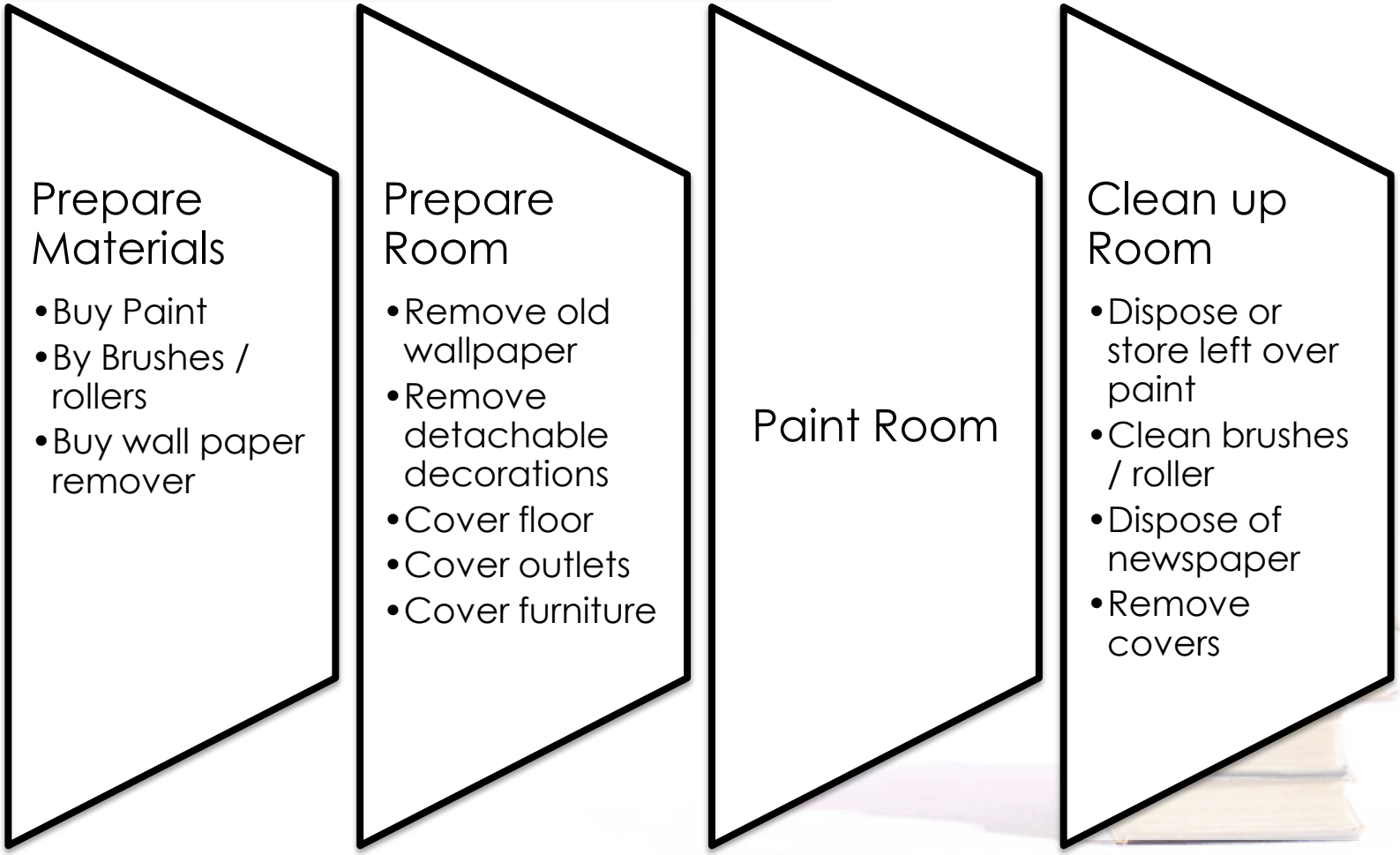
- Tasks as Nodes
  - Contain Resources and Duration
- Tasks as Arcs
  - Tasks are uni-directional arrows
  - Nodes are the "states"
  - Kelly-Walker form



# Work Breakdown Structure

- Used to create the task list
- Tree-decomposition of project tasks
- Identifies "terminal elements"
- Required by Government as part of SOW
- Use "stick-notes" method early on
- Carl L Pritchard "*Nuts and Bolts Series 1: how to Build a Work Breakdown Structure*"

# WBS Painting a Room

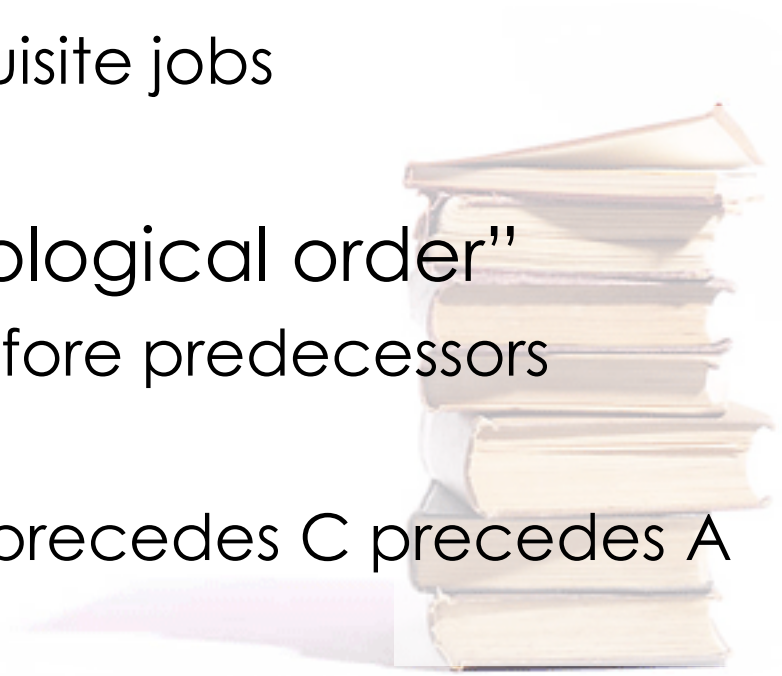


# WBS Guidelines

- No more than 100 – 200 elements
  - If needed make separate sub-project
- Up to 3 or 4 levels deep
- Not more than 5 – 9 jobs at one level
  - Human cognition
  - Dilution of attention
  - Too many dependencies
- Jobs should be a similar size
- Level of detail
  - Unclear?

# Task List

- List all tasks in a table with
  - Identifying symbol
  - Task description
  - Immediate prerequisite jobs
  - Expected duration
- Arrange in "technological order"
  - No job appears before predecessors
  - No Iterations
  - Job A precedes B precedes C precedes A



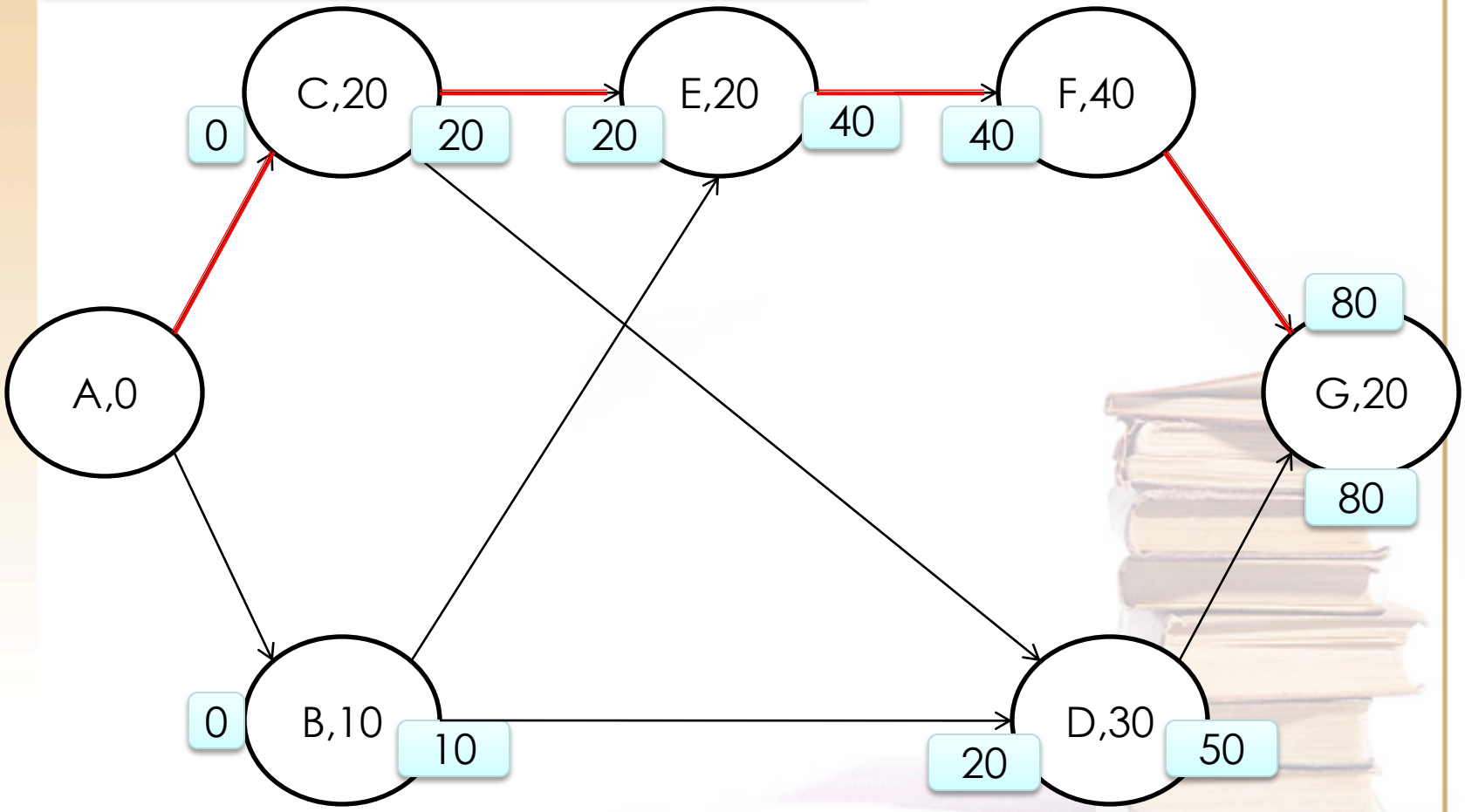
# Example Task list

| ID | | Task Name | % Complete | Duration | Start | Finish | Predecessors |
|---|---|---|---|---|---|---|---|
| 1 | | **Project "XYZ"** | **7%** | **42 days** | **Mon 16-11-09** | **Tue 12-01-10** | |
| 2 | | Marketing Study | 30% | 14 days | Mon 16-11-09 | Thu 03-12-09 | |
| 3 | | Requirements Definition | 0% | 7 days | Mon 23-11-09 | Tue 01-12-09 | |
| 4 | | System Design | 0% | 7 days | Fri 04-12-09 | Mon 14-12-09 | 2 |
| 5 | | Component A | 0% | 14 days | Tue 15-12-09 | Fri 01-01-10 | 4 |
| 6 | | Component B | 0% | 14 days | Tue 15-12-09 | Fri 01-01-10 | 4 |
| 7 | | Integration and test | 0% | 7 days | Mon 04-01-10 | Tue 12-01-10 | 6,5 |
| 8 | | Product Release | 0% | 0 days | Tue 12-01-10 | Tue 12-01-10 | 7 |

# Critical Path Algorithm

- For large projects there are many paths
- An algorithm is needed to identify the CP efficiently
- Develop information about each task in the context of the project
- Times
  - Start time (S)
  - For each job: earliest start (ES)
    - When all predecessors are completed
  - Job duration t
  - Earliest Finish (EF) = (ES)+t
- Show algorithm using project graph

# CP Algorithm - Graphical



# HW1 Introduction

- You are Project Manager for a project to develop a corporate web site
- Plan the project
  - Task list
  - Project graph
  - Critical path
  - Slack times
  - "managerial" issues