

Chapter 13 – Software
Project Management

Software Engineering

Software Project Management process

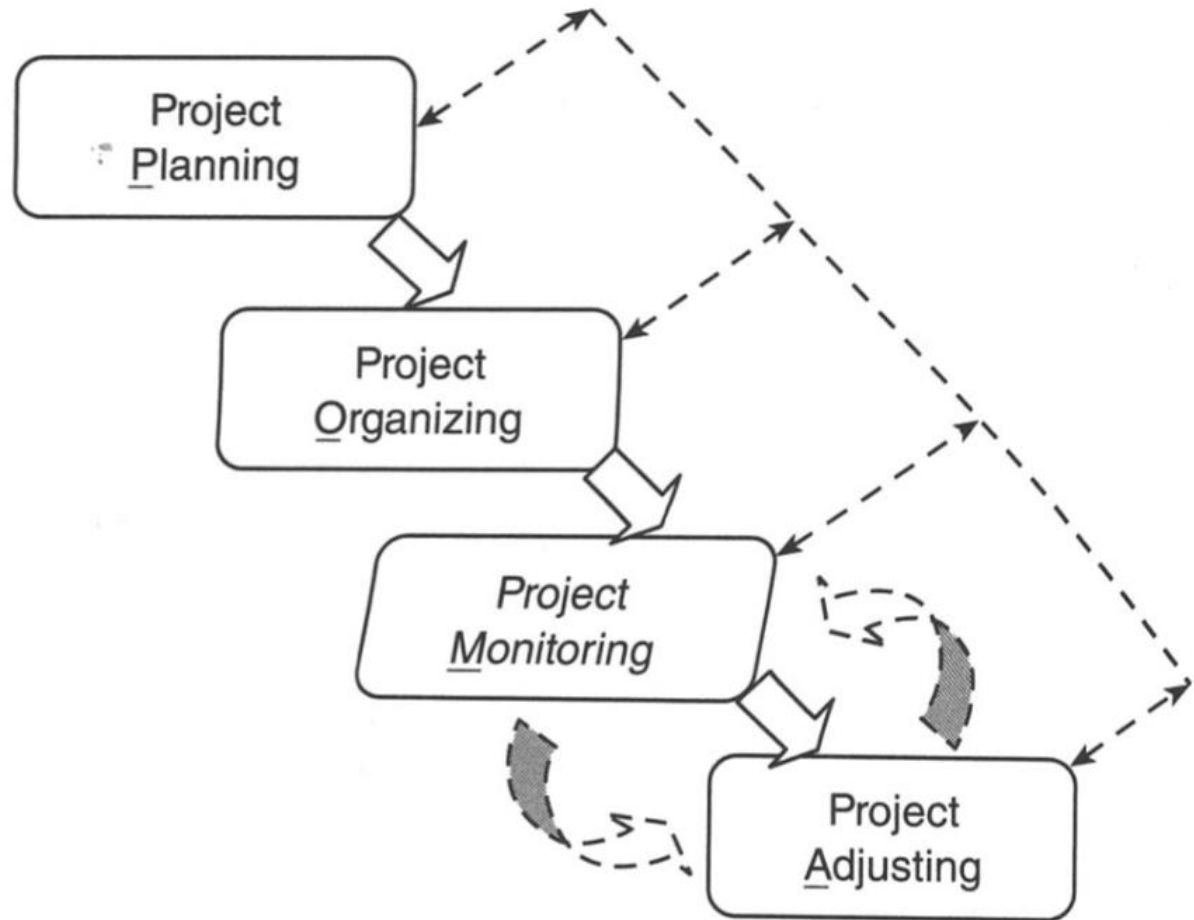


Figure 13.1 Software project management process.

Project Management Goals

- End results satisfy customer's needs
- Product attributes (quality, security, productivity, cost, etc.) are met
- Intermediate milestones are met
- Team members are effective with high morale
- Tools and other resources available and effectively utilized

Stage 1: Planning - Questions

The following needs to be determined during the planning stage:

- Nature of project, sponsors and users
- Needed and desired requirements
- Deliverables
- Constraints (schedule, cost, etc.)
- Risks

Planning

The following need to be determined:

- Requirements are accurately understood and specified
- Work effort, schedule and needed resources are estimated
- Measurable goals are defined and established
- Allocations of people, process, tools and facilities are determined
- Risks are identified and analyzed

Risk Analysis

3 major components to risk analysis are risk:

- identification
- prioritization
- mitigation plans

Included in Plan

Plan includes:

- Brief description of requirements and deliverables
- Work effort estimation – explained and shown in detail
- Needed resources – people, tools, methodologies & facilities
- Schedule - major and minor resources
- Project goals - including multiple project attributes
- Assumptions and risks – major and minor

SMART Project Goals

Project goals need to be SMART

Specific

Measurable

Attainable

Relevant

Time-bound

Stage 2: Organizing

Table 13.1 Pairing Planning and Organizing Activities

Planning	Organizing
Project content and deliverables	
Project tasks and schedule	Set up tracking mechanisms of tasks and schedules.
Project resources	Acquire, hire, and prepare resources such as people, tools, and processes.
Project goals and measurement	Establish mechanism to measure and track the goals.
Project risks	Establish mechanism to list, track, and assign risk mitigation tasks.

Stage 3: *Monitoring*

3 components of monitoring:

- Collecting project information
- Analysis and evaluation of data
- Presentation and communication

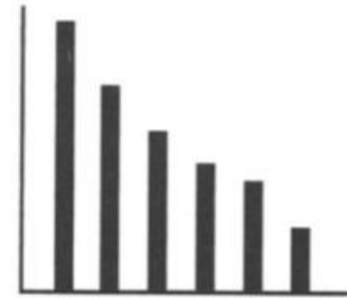
Visualizing and Reporting Data



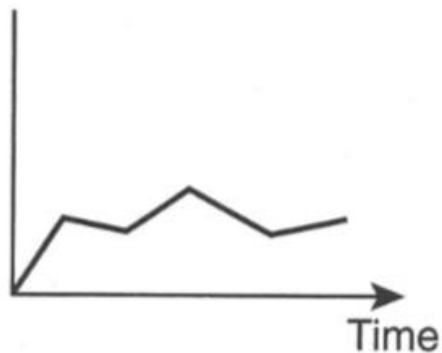
Pie chart



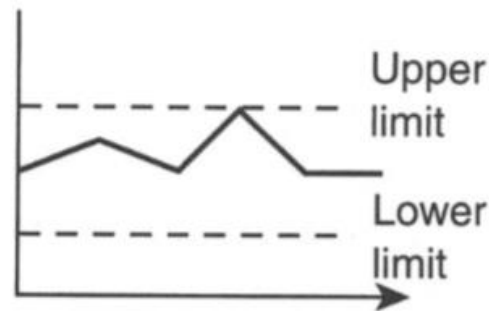
Histogram



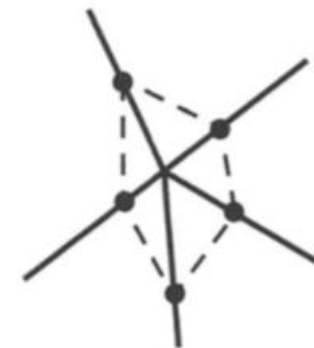
Pareto diagram



Time chart



Control chart



Kiviat chart

Figure 13.2 Visualization and reporting information.

Stage 4: Adjusting

Items that can be adjusted:

- Resources
- Schedule
- Project content

(Notice that quality is NOT on the list)

Coordination Efforts

Recall back in Chapter 2 engineering projects involves coordination of :

Process

People

Product

Kind of relates to items that can be adjusted:

- Resources
- Schedule
- Project content



Good
Cheap
Quick



Good
Cheap
Quick

Pick any 2...

Project Management Techniques

Techniques discussed:

1. Project effort estimation
2. Work breakdown structure
3. Project status tracking with earned value
4. Developing measurements and metrics

Project Effort Estimation

Project effort estimation:

$$\text{Units of effort} = a + b(\text{size})^c + \text{ACCUM}(\text{factors})$$

Unit can be person month, person days, person hours

a – base cost

b – scale

size – KLOC, function points, object points

c – allows amount increase to be non-linear

ACCUM – sum

Factors – technical, personnel, tools, process

AbOut Refactor Effort Estimation

Units of effort = $a + b(\text{size})^c + \text{ACCUM}(\text{factors})$

Effort AbOut refactor =
 $45 \text{ days} * n \text{ students} + 300 (3)^1 + (2 * n + 10 + 10)$

Unit – person hours

Base cost - 45 class days * number of students

KLOC - Current AbOut 2,000 LOC

Factors :

Test driven development – $2 * n$

GitLab – n

Documentation – n

·
·
·

Constructive Cost Model (COCOMO)

COCOMO has 3 models:
macro, intermediate and micro

Discussion is only on intermediate models.

Intermediate models have 3 modes:
organic – simple
Semidetached – intermediate
Embedded – difficult

COCOMO

Steps for COCOMO:

1. Pick the mode that best reflects the project by considering 8 factors (see next slide)
2. Estimate size of project (KLOC- 1,000 lines of code)
3. Review 15 factors (see later slide), cost drivers, to determine impact
4. Determine effort via formula

COCOMO: Determine Mode of Project

8 factors to determine mode of project

1. Team's understanding
2. Team's experience
3. Need to conform with requirements
4. Need to conform with external interfaces
5. Need to develop concurrently with new systems
6. New and innovative technology, architecture
7. Need to meet or beat schedule
8. Project size

COCOMO: Determine Mode of Project - AbOut Refactor

8 factors to determine mode of project

1. Team's understanding - **medium**
2. Team's experience - **low**
3. Need to conform with requirements - **low**
4. Need to conform with external interfaces - **low**
5. Need to develop concurrently with new systems - **low**
6. New and innovative technology, architecture - **high**
7. Need to meet or beat schedule - **low**
8. Project size - **low**

Estimate that mode of AbOut Refactor is organic

Effort Estimations

Organic:	$\text{Effort} = [3.2 \times (\text{size})^{1.05}] \times \text{PROD } (f's)$
Semidetached:	$\text{Effort} = [3.0 \times (\text{size})^{1.12}] \times \text{PROD } (f's)$
Embedded:	$\text{Effort} = [2.0 \times (\text{size})^{1.20}] \times \text{PROD } (f's)$

15 Cost Drivers

- Product attributes
 1. Required software reliability
 2. Database size
 3. Product complexity
- Computer attributes
 4. Execution time constraint
 5. Main memory constraint
 6. Virtual machine complexity
 7. Computer turnaround time
- Personnel attributes
 8. Analyst capability
 9. Applications experience
 10. Programmer capability
 11. Virtual machine experience
 12. Programming language experience
- Project attributes
 13. Use of modern practice
 14. Use of software tools
 15. Required development schedule

Cost Drivers Values

Table 13.2 COCOMO Cost-Driver Values

Cost-Drivers	Very Low	Low	Nominal	High	Very High	Extra High
1	0.75	0.98	1.0	1.15	1.40	—
2	—	0.94	1.0	1.08	1.16	—
3	0.70	0.85	1.0	1.15	1.30	—
4	—	—	1.0	1.11	1.30	1.65
5	—	—	1.0	1.06	1.21	1.66
6	—	0.87	1.0	1.15	1.30	1.56
7	—	0.87	1.0	1.07	1.15	—
8	1.46	1.19	1.0	0.86	0.71	—
9	1.29	1.13	1.0	0.91	0.82	—
10	1.42	1.17	1.0	0.86	0.70	—
11	1.21	1.10	1.0	0.90	—	—
12	1.14	1.07	1.0	0.95	—	—
13	1.24	1.10	1.0	0.91	0.82	—
14	1.24	1.10	1.0	0.91	0.83	—
15	1.23	1.19	1.0	1.04	1.10	—

Function Points

Function points are a suggested improvement over LOC

Consider

1. External inputs
2. External outputs
3. External inquires
4. Internal logical files
5. External interface files

Function Point Weights

Determine if function is simple, average or complex and use weights

Table 13.3 Function Point Weights

Software Components	Simple	Average	Complex
External inputs	3	4	6
External outputs	4	5	7
External inquiries	3	4	6
Internal logical files	7	10	15
External interface files	5	7	10

International Function Point users Group (IFPUG)

Function Points

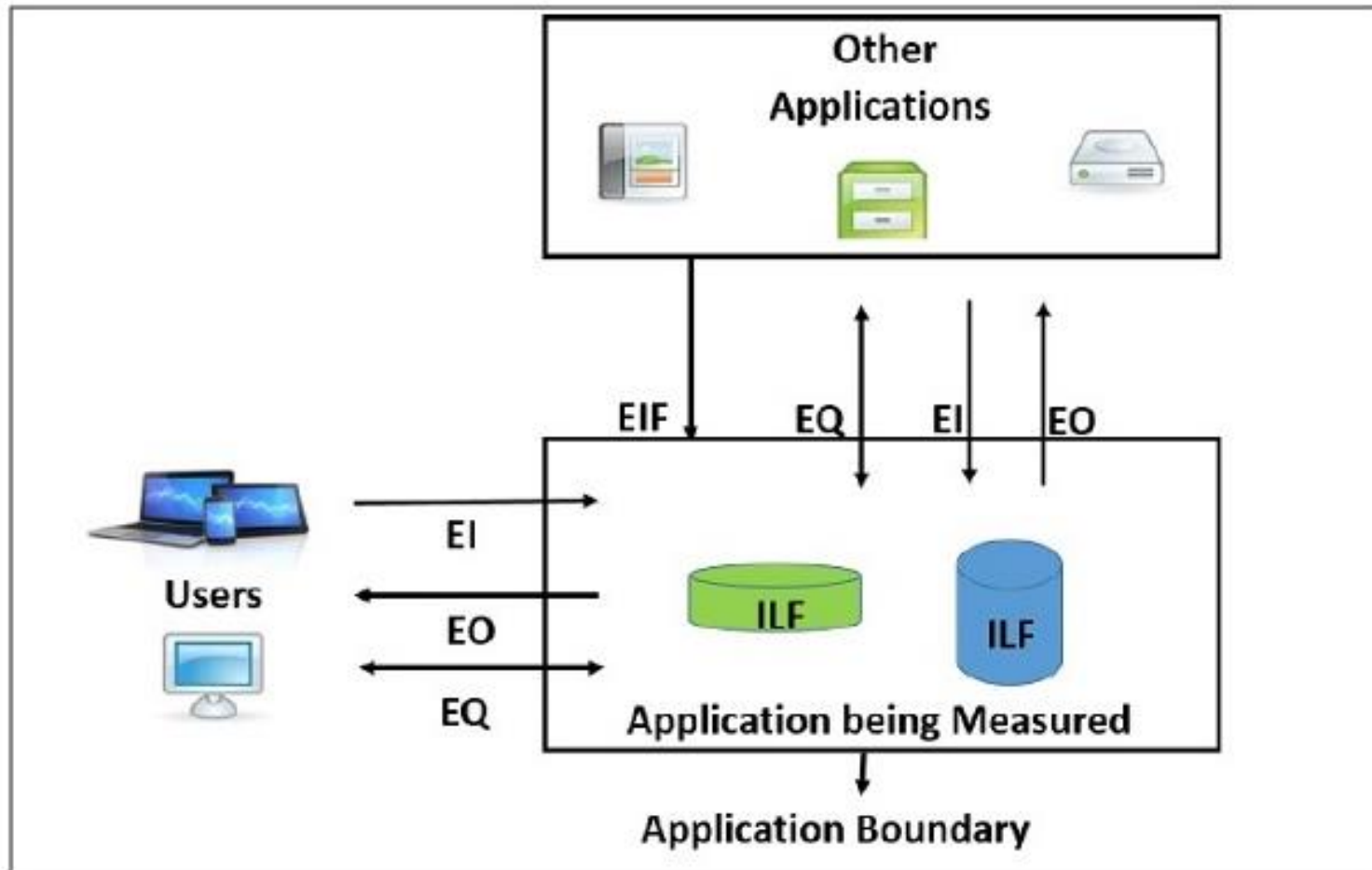


Figure 1: Application Boundary, Data Functions, Transaction Functions

Function Point Counting

Counting function points:

- Classify software items into **transactions** and **data entities** (get transactions from functional decomposition diagrams or data flow diagrams, data entities from ERD)
- Classify transactions into external inputs, outputs and queries
- Classify data entities into external and internal entities
- Rate complexity of components (EI, EO, ...) into low, medium and high and use a table of weights to find the unadjusted count
- Adjust the count based on 14 factors, each rated on a scale from 0 to 5

Weights to Calculate the Unadjusted Function Point

FUNCTION UNITS	LOW	AVG	HIGH
EI	3	4	6
EO	4	5	7
EQ	3	4	6
ILF	7	10	15
EIF	5	7	10

Multiply each individual function point to corresponding values in TABLE.

Factors to Adjust Function Point Count

1. Data Communication
2. Distributed data processing
3. Performance
4. Heavily used configuration
5. Transaction rate
6. Online data entry
7. End user efficiency
8. Online update
9. Complex processing
10. Reusability
11. Installation ease
12. Operational ease
13. Multiple sites
14. Facilitate change

Function Points Versus Story Points

Function points are standardized and good for estimating effort from a user's perspective

Story points are a team's agreed amount of effort to do some work, it is local to the project

Beneficial to use both

Work Breakdown Structure

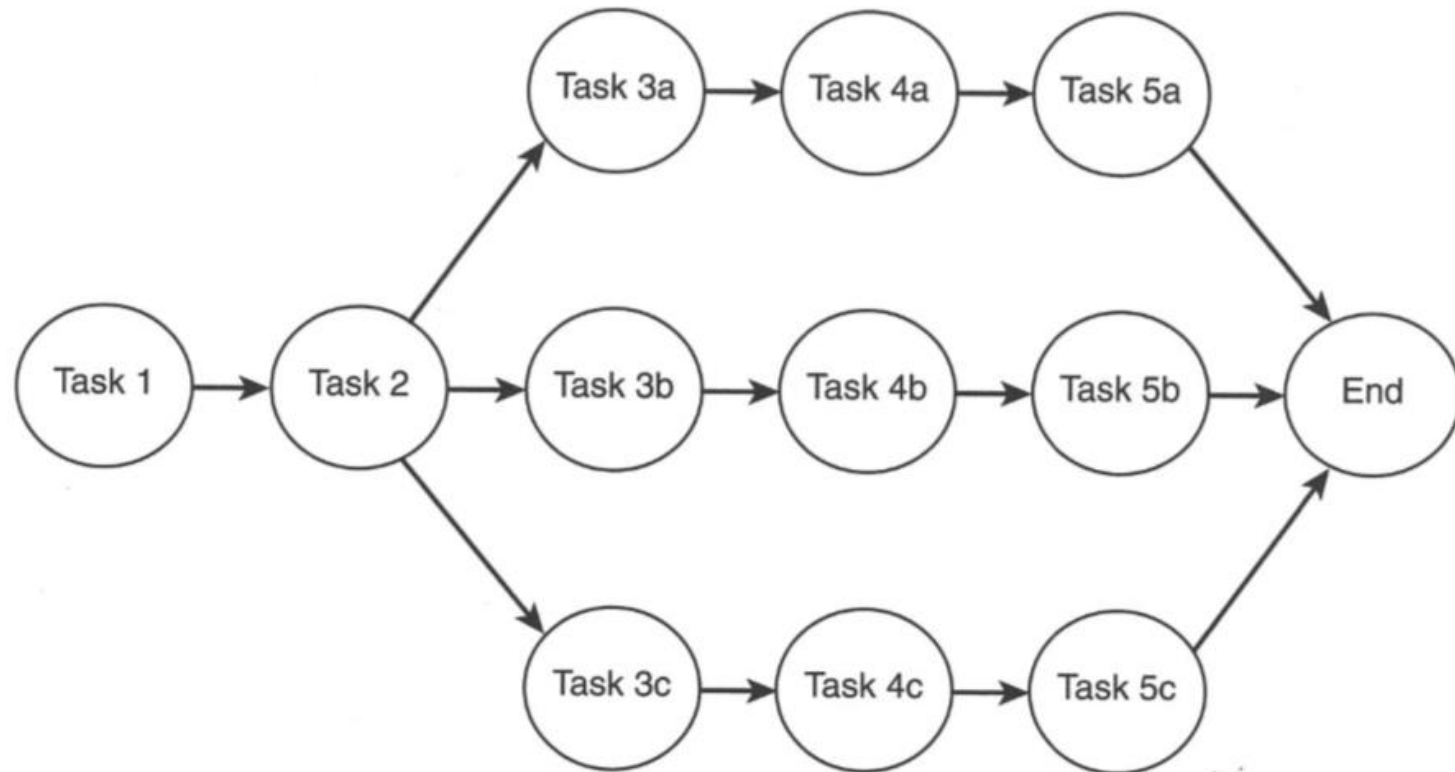


Figure 13.3 A WBS network of tasks.

WBS with Time Units

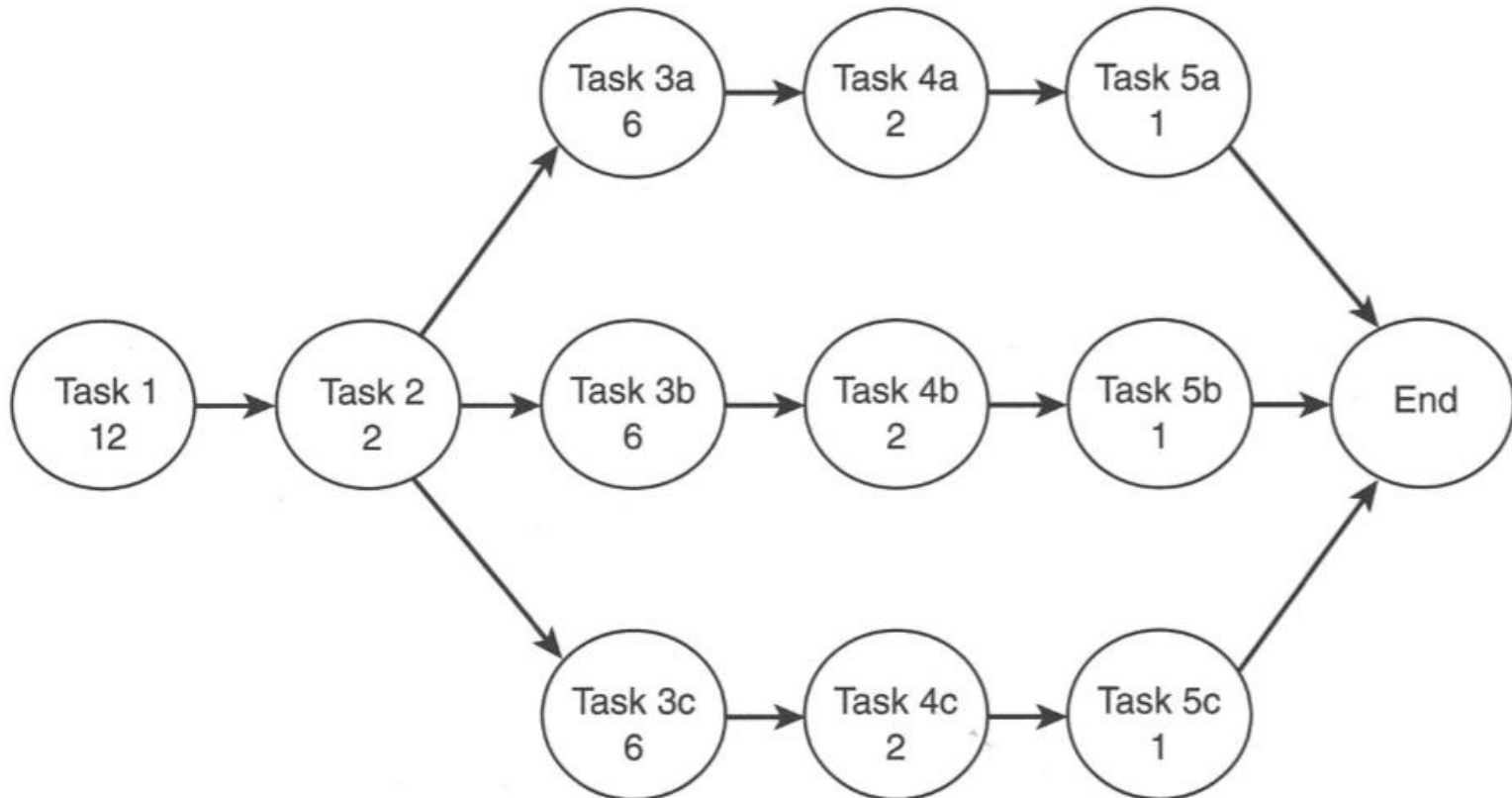


Figure 13.4 Task network with estimated time units.

Schedule Estimate

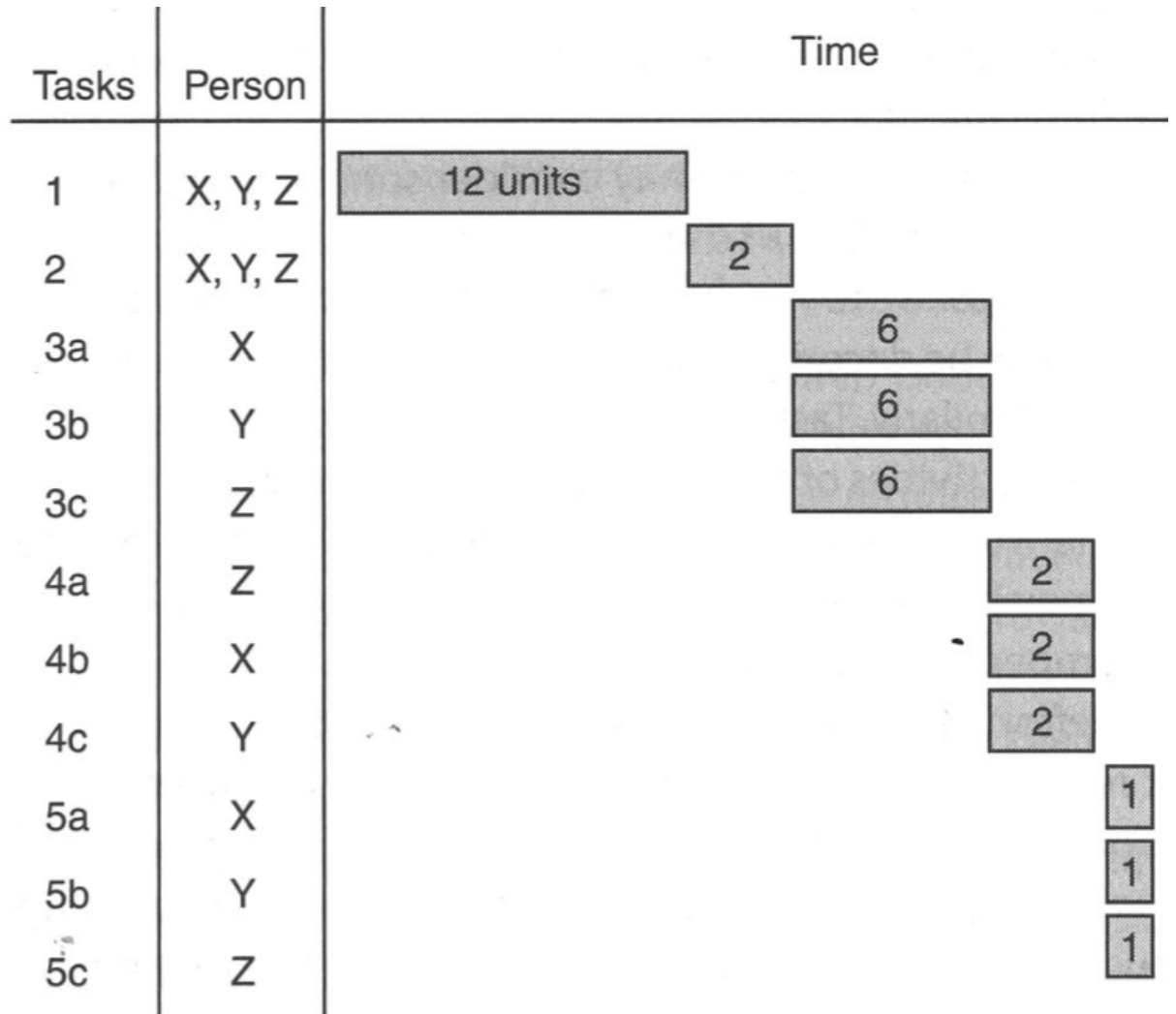


Figure 13.5 Initial schedule estimate.

Earned Value Example

Table 13.4 Earned Value Example Date: 4/5/2012

Work Tasks	Estimated Effort in Pers-days	Actual Effort Spent So Far in Pers-days	Estimated Completion Date	Actual Completion Date
1	10	10	2/5/2012	2/5/2012
2	15	25	3/15/2012	3/25/2012
3	30	15	4/25/2012	
4	25	20	5/5/2012	4/1/2012
5	15	5	5/25/2012	
6	20	15	6/10/2012	