CMSC 128

Introduction to Software Engineering
Second Semester AY 2009-2010

jachermocilla@uplb.edu.ph

Scheduling and Tracking

- Why is software delivered late?
  - Unrealistic deadline established by someone outside the SE group
  - Changes in requirements not reflected in schedule
  - Honest underestimate of effort and resources
  - Risks not considered when project started
  - Technical and human difficulties
  - Miscommunication
  - Failure to recognize and act on possible

Basic Principles

- How do projects fall behind schedule?
  - 'One day at a time.' --F Brooks
- In a project, a set of small tasks must be performed to accomplish a goal
  - Some of these tasks are 'critical' and may affect schedule
- Define all tasks, identify the critical tasks, track the progress of the critical tasks

Basic Principles

- Scheduling
  - Distributes estimated effort across the planned project duration by allocating the effort to specific software engineering tasks
  - Schedules evolves over time
  - From macroscopic (framework activities) to microscopic (tasks)
  - First view: End date for release is set
  - Second view: End date is set by SE organization

Basic Principles

- Compartmentalization
  - Product and process decomposition
- Interdependency
  - Some tasks must occur in sequence, others in parallel
- Time allocation
  - Each task must be allocated some work units
  - Start and completion date are functions of interdependency

Basic Principles

- Effort Validation
  - With a defined number of staff, no more than the allocated staff should have been allocated at a given time
- Defined responsibilities
  - Every task scheduled must be assigned to a specific team member
- Defined outcomes
  - Deliverables(work product) are produced

Basic Principles

- Defined milestones
  - Task or group of tasks is associated with a project milestone
  - A milestone is accomplished when one or more work products have been reviewed for quality and have been approved

People and Effort

- In a small project, a single person can do everything: analysis, design, code, test, etc.
- Big projects require more people
- So, if project is behind schedule, perhaps the project is big! Add more people??
  - Consider communication and training

People and Effort

- Consider four developers individually capable of producing 5K LOC/yr
- Produce a team from these developers, there will be six communication paths, assume that productivity will be reduced by 250 LOC/yr.
- What will be the productivity of each developer now?
- Imagine if we add two more developers because project is behind schedule!

People and Effort

- Relationship between people and productivity is not linear
- Can we conclude then that forming teams is counterproductive?
  - NO!
  - If communication serves to improve quality: formal technical reviews
- Benefit can be gained by using fewer people over a somewhat longer time span

Effort Distribution

- Estimation produces estimate of effort
- 40-20-40 rule
- Characteristics of project dictate the distribution of effort
  - Project planning – 2-3 percent
  - Requirements Analysis – 10–25 percent
  - Software Design – 20-25 percent
  - Coding – 15-20 percent
  - Testing – 30-40 percent

Defining Task Sets

- At the start of the project, a software process model is selected
- Process model selected consists of task sets (applies to any software process model)
- Task sets vary depending on the project
- A task set is a collection of software engineering work taks, milestones, and deliverables that must be accomplished to complete a project

Defining Task Sets

- Task set should provide enough discipline to achieve quality and not burden the team with unnecessary work
- Task sets are designed to accommodate different types of projects and different degrees of rigor

Types of Projects

- Concept Development
  - Explore some new concept, apply new tech
- New Application Development
  - From customer request
- Application Enhancement
  - Modifications obervable by the end user
- Application Maintenance
  - Modifications not obvious to end user
- Reengineering - rebuilding

Degree of Rigor

- Casual
  - All framework activites are applied but with only a minimum task set: umbrella activities are minimized documentation requirements are reduced
- Structured
  - All framework activities are applied, including all umbrella activities
- Strict
  - Same as structured but robust documentation

Degree of Rigor

- Quick Reaction
  - Framework activities are applied, but because of emergency situations, only essential activities will be applied
  - 'Back-filling','to follow'

Adaptation Criteria

- Selecting the degree of rigor must be systematic, based on some criteria
- Adaptation Criteria: size of project, number of potential users, mission criticality, application longevity, stability of rquirements, ease of developer/customer communication,maturity of applicable technology,performance constraints, embedded characteristics, staffing, reengineering

Computing a TSS Value

- Review the adaptation criteria
- Review the weighting factors assigned to each of the criteria.
  - 0.8-0.12
- Product=grade*weighting_factor*entry_point_multiplier
- Compute the average of all entries in the Product column

TABLE 7.1 COMPUTING THE TASK SET SELECTOR

Adaptation Criteria	Grade	Weight	Entry Point Multiplier					Product
			Conc.	NDev.	Enhan.	Maint.	Reeng.	
Size of project	_____	1.20	0	1	1	1	1	_____
Number of users	_____	1.10	0	1	1	1	1	_____
Business criticality	_____	1.10	0	1	1	1	1	_____
Longevity	_____	0.90	0	1	1	0	0	_____
Stability of requirements	_____	1.20	0	1	1	1	1	_____
Ease of communication	_____	0.90	1	1	1	1	1	_____
Maturity of technology	_____	0.90	1	1	0	0	1	_____
Performance constraints	_____	0.80	0	1	1	0	1	_____
Embedded/nonembedded	_____	1.20	1	1	1	0	1	_____
Project staffing	_____	1.00	1	1	1	1	1	_____
Interoperability	_____	1.10	0	1	1	1	1	_____
Reengineering factors	_____	1.20	0	0	0	0	1	_____
Task set selector (TSS)								_____

TABLE 7.2 COMPUTING THE TASK SET SELECTOR—AN EXAMPLE

Adaptation Criteria	Grade	Weight	Entry Point Multiplier					Product
			Conc.	NDev.	Enhan.	Maint.	Reeng.	
Size of project	2	1.2	____	1	____	____	____	2.4
Number of users	3	1.1	____	1	____	____	____	3.3
Business criticality	4	1.1	____	1	____	____	____	4.4
Longevity	3	0.9	____	1	____	____	____	2.7
Stability of requirements	2	1.2	____	1	____	____	____	2.4
Ease of communication	2	0.9	____	1	____	____	____	1.8
Maturity of technology	2	0.9	____	1	____	____	____	1.8
Performance constraints	3	0.8	____	1	____	____	____	2.4
Embedded/nonembedded	3	1.2	____	1	____	____	____	3.6
Project staffing	2	1.0	____	1	____	____	____	2.0
Interoperability	4	1.1	____	1	____	____	____	4.4
Reengineering factors	0	1.2	____	0	____	____	____	0.0
Task set selector (TSS)								**2.8**

Interpreting TSS

- TSS < 1.2 , casual
- 1.0 < TSS < 3.0, structured
- TSS > 2.4, strict

Selecting SE Tasks

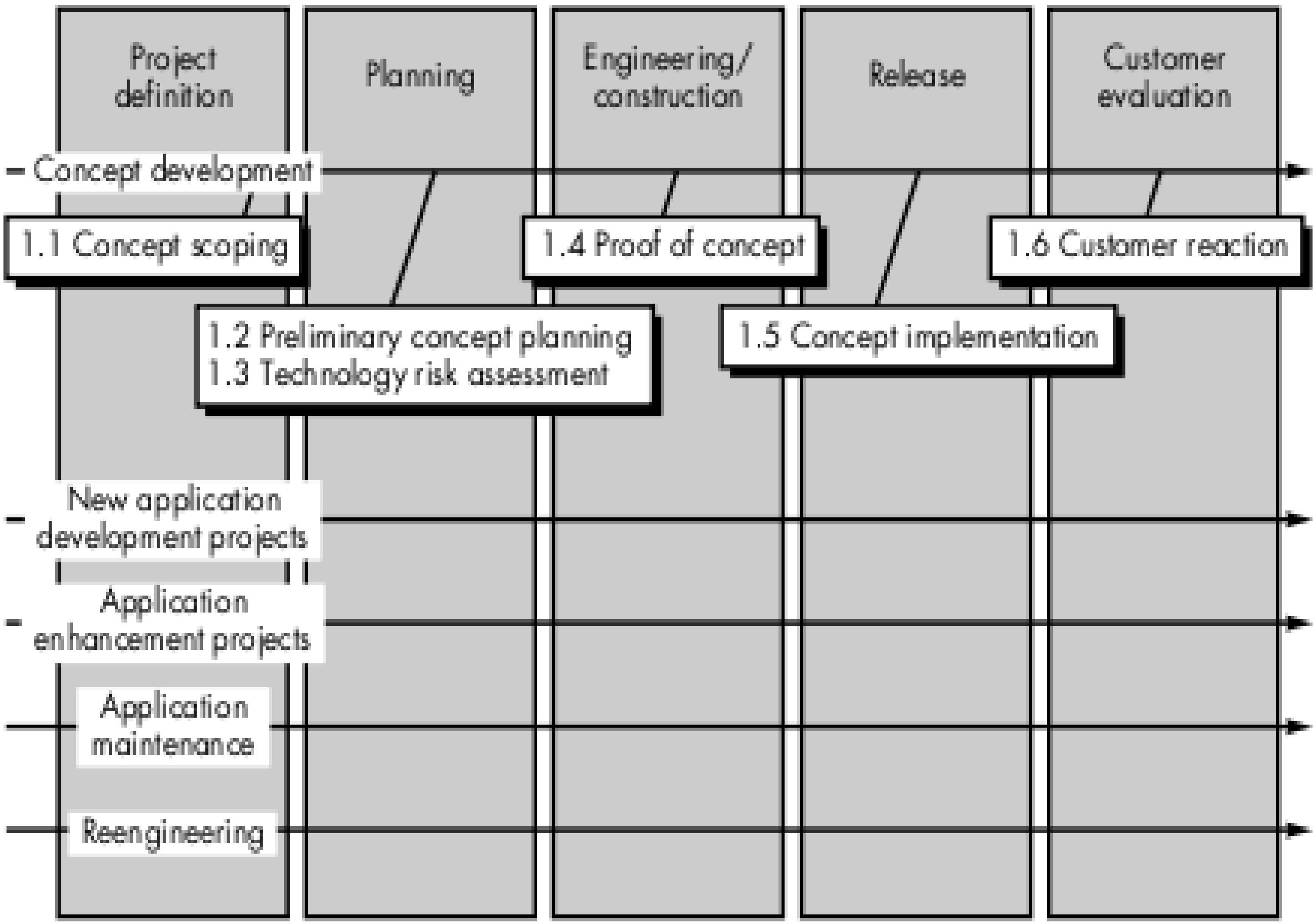
- Developing project schedule requires the task set to be distributed on the project timeline
- Task set vary depending on the project type and degree of rigor
- One type of project may flow smoothly to the next: from concept to new application

Ex. Concept Development

- Major Tasks (macroscopic)
  - Concept scoping
  - Concept planning
  - Technology risk assessment
  - Proof of concept
  - Concept implementation
  - Customer reaction to concept
- Major tasks refined through decomposition to create final schedule

Ex. Concept Development

FIGURE 7.1
Concept development tasks in a linear sequential model



Ex. Concept Development

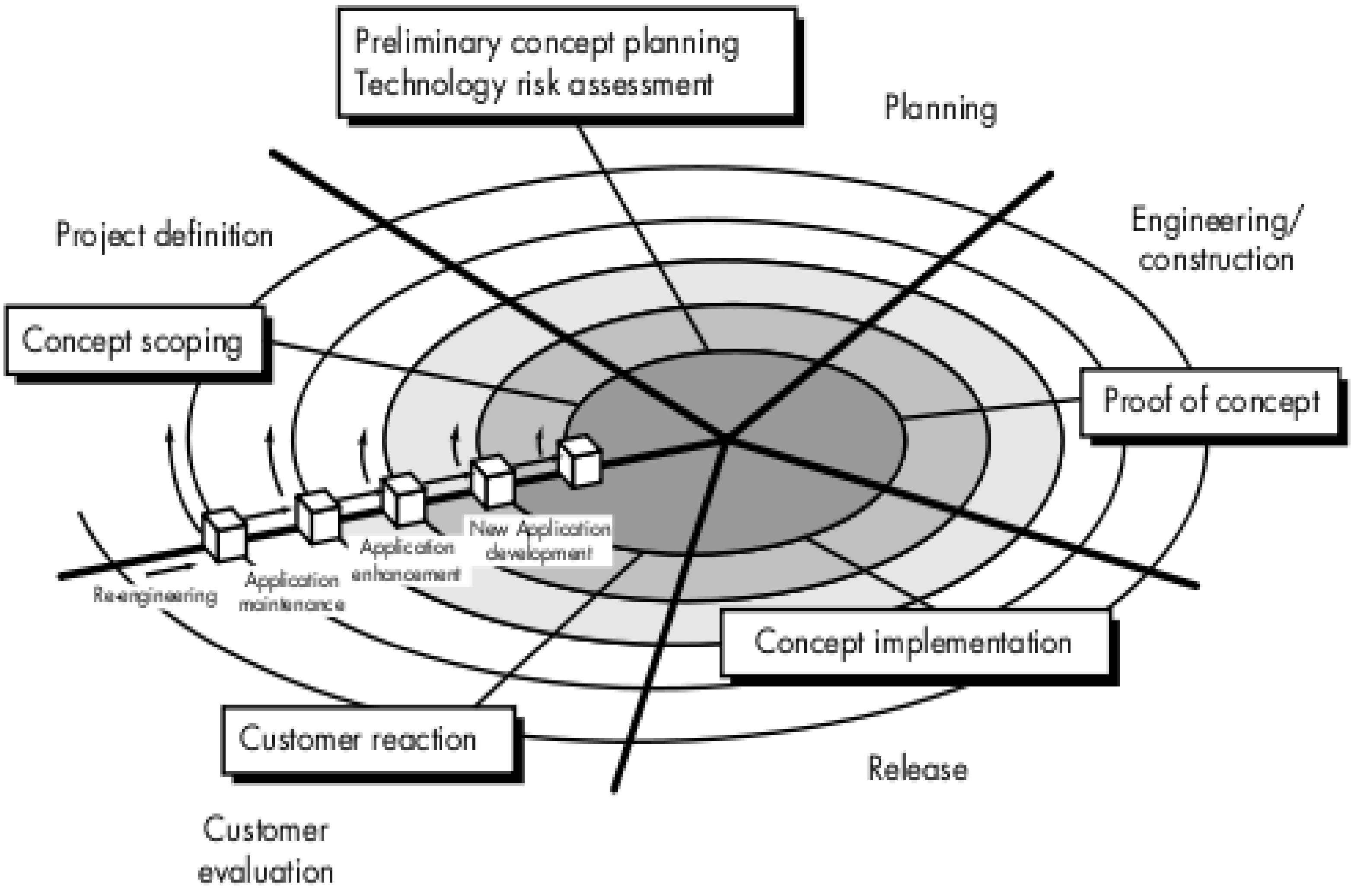


FIGURE 7.2
Concept development tasks using an evolutionary model

Task Networks

- Tasks and subtasks have interdependencies based on their sequence
- Some tasks can be performed in parallel
- Task network is a graphical representation of task flow
- Determine tasks that lie on the critical path
  - Tasks that determine overall schedule

Task Networks

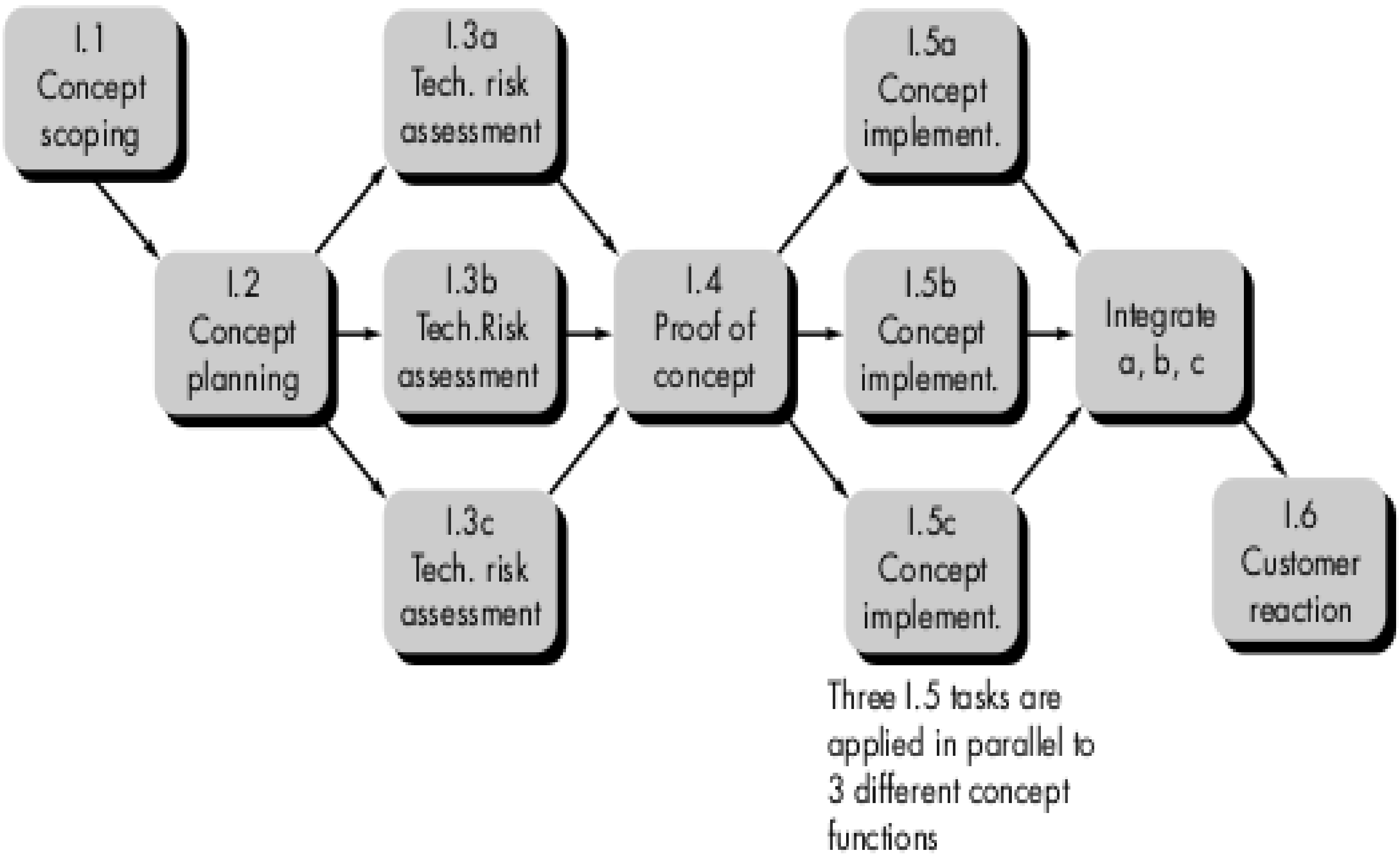


FIGURE 7.3 A task network for concept development

Scheduling

- Program Evaluation and Review Technique (PERT) and Critical Path Method (CPM)
  - Requires
    - Estimates of effort
    - Decomposition of product function
    - Selection of process model
    - Selection of project type and task set
- Work Breakdown Structure (WBS) – task decomposition as task network

Scheduling

- What can PERT/CPM do?
  - Determine critical path
    - Chain of tasks that determine the duration
  - Establish most likely time estimates for individual tasks
  - Calculate boundary times
    - Earliest time to start
    - Latest time to start
    - Earliest finish
    - Latest finish
    - Float time (leeway)

Gantt Charts

- WBS is created and entered in an automated scheduling tool
- For each task: effort, duration, start date, and human resource are specified
- Created for entire project or portions of it
- Example tools: MS Project, Imendio/Planner (linux)

Gantt Chart

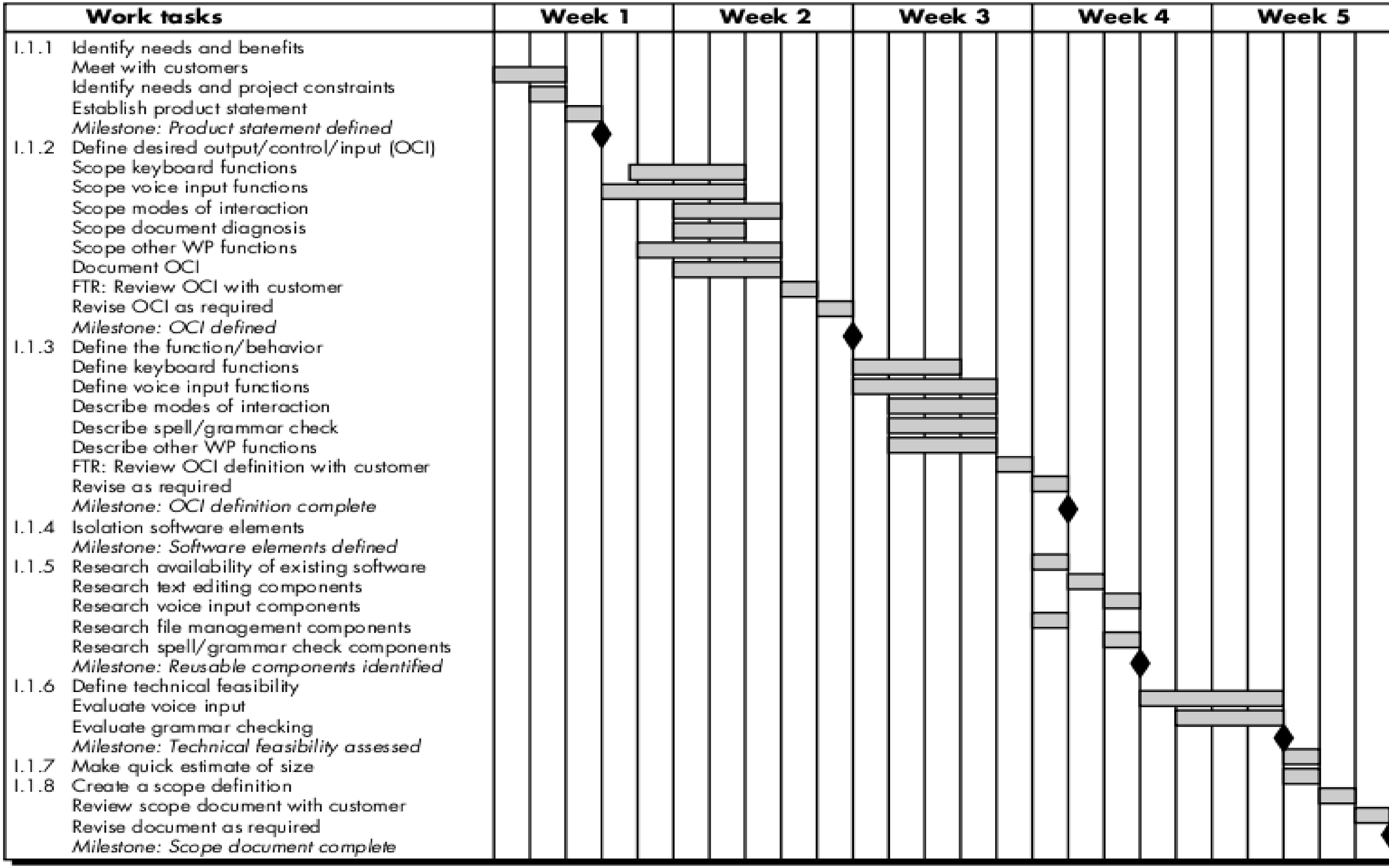


FIGURE 7.4 An example timeline chart

Project Table

Work tasks	Planned start	Actual start	Planned complete	Actual complete	Assigned person	Effort allocated	Notes
I.1.1 Identify needs and benefits							Scoping will require more effort/time
Meet with customers	wk1, d1	wk1, d1	wk1, d2	wk1, d2	BLS	2 p-d	
Identify needs and project constraints	wk1, d2	wk1, d2	wk1, d2	wk1, d2	JPP	1 p-d	
Establish product statement	wk1, d3	wk1, d3	wk1, d3	wk1, d3	BLS/JPP	1 p-d	
Milestone: Product statement defined	wk1, d3	wk1, d3	wk1, d3	wk1, d3			
I.1.2 Define desired output/control/input (OCI)							
Scope keyboard functions	wk1, d4	wk1, d4	wk2, d2		BLS	1.5 p-d	
Scope voice input functions	wk1, d3	wk1, d3	wk2, d2		JPP	2 p-d	
Scope modes of interaction	wk2, d1		wk2, d3		MLL	1 p-d	
Scope document diagnostics	wk2, d1		wk2, d2		BLS	1.5 p-d	
Scope other WP functions	wk1, d4	wk1, d4	wk2, d3		JPP	2 p-d	
Document OCI	wk2, d1		wk2, d3		MLL	3 p-d	
FTR: Review OCI with customer	wk2, d3		wk2, d3		all	3 p-d	
Revise OCI as required	wk2, d4		wk2, d4		all	3 p-d	
Milestone: OCI defined	wk2, d5		wk2, d5				
I.1.3 Define the function/behavior							

FIGURE 7.5 An example project table

Project Tracking

- Project schedule provides roadmap
- How to track?
  - Periodic project status meeting
  - Evaluate results of all reviews
  - Determine if milestones have been achieved
  - Compare actual start date to planned start date
  - Informal meeting with practitioners to obtain subjective assessment of progress

Control

- Employed to administer project resources, cope with problems, and direct project staff
- Project is going well, light control
- When problems occur
  - Allocate resource
  - Redeploy staff
  - Redefine Schedule

Control

- Deadline pressures??!!!
- Time-boxing
  - Complete product may not be delivered on time
  - Incremental approach is selected
  - Schedule of tasks associated with increment is adjusted backwards from deadline of increment
  - When task hits boundary, work stops and next task begins

Summary

- Scheduling is the culmination of planning activity
- Scheduling starts with process decomposition
- Task set is adapted based on type and degree of rigor
- Scheduling tools include PERT, CPM, Gantt Chart, Project Table
- Schedule used to track and control project

Reference

- Roger S. Pressman.Software Engineering: A Practitioner's Approach, 4th Ed.McGraw-Hill,1997. Chapter 5