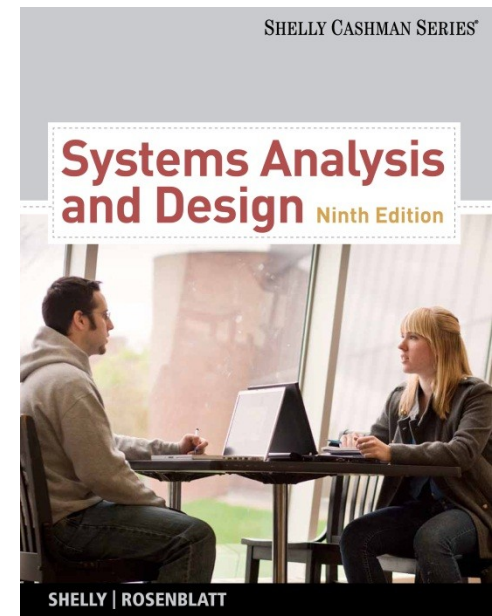


Systems Analysis and Design 9th Edition

Chapter 3

Managing Systems Projects



Chapter Objectives

- Explain project planning, scheduling, monitoring, and reporting
- Describe work breakdown structures, task patterns, and critical path analysis
- Explain techniques for estimating task completion times and costs

Chapter Objectives

- Describe various scheduling tools, including Gantt charts and PERT/CPM charts
- Analyze task dependencies, durations, start dates, and end dates
- Describe project management software and how it can assist you in project planning, estimating, scheduling, monitoring, and reporting

Chapter Objectives

- Discuss the importance of project risk management
- Understand why projects sometimes fail

Introduction

- You will learn about project planning, estimating, scheduling, monitoring, reporting, and the use of project management software
- You also will learn how to control and manage project changes as they occur

Overview of Project Management

- Project Management
- A successful project must be completed on time, within budget, and deliver a quality product that satisfies users and meets requirements
- Project manager or project leader
- Project coordinator

Overview of Project Management

- What Does a Project Manager Do?
 - Project manager, project leader
 - Project planning
 - Project scheduling
 - Project monitoring and controlling
 - Project reporting

Overview of Project Management

- Project Activities and Planning Steps

	Planning	Scheduling	Monitoring	Reporting
STEP1: Create a work breakdown structure	✓			
STEP 2: Identify task patterns	✓	✓		
STEP 3: Calculate the critical path	✓	✓		
Manage the operational project			✓	✓

Step 1: Create a Work Breakdown Structure

- Work breakdown structure (WBS)
- What is a Gantt Chart?
 - Task group
 - Can present an overview of the project's status, but does not provide detailed information that is necessary when managing a complex project

Step 1 – Cont.

- **Project Planning and Requirements Gathering:** This includes defining the scope, objectives, resources, and timelines of the project.
- **Software Development:**
 - **Module or Component Development:** Breaking down the software into smaller modules or components.
 - **Coding:** Writing the actual code for the various components.
 - **Unit Testing:** Testing individual modules or components.
- **Integration and Testing:**
 - **System Integration:** Combining different modules or components into a fully functioning system.
 - **System Testing:** Running tests to ensure the system meets all functional and non-functional requirements.
 - **Bug Fixes and Optimizations:** Addressing issues found during testing.
- **Deployment and Maintenance:**
 - **Deployment:** Installing the software into a production environment.
 - **User Training:** Providing documentation and training to users or clients.
- **Project Management Activities:**
 - **Risk Management:** Identifying and mitigating risks that could affect the project.
 - **Quality Assurance:** Ensuring that the software meets quality standards.

Step 1: Create a Work Breakdown Structure

- What is a PERT/CPM Chart?
 - The Program Evaluation Review Technique (PERT)
 - Critical Path Method (CPM)
 - The distinction between the two methods has disappeared over time

Step 1: Create a Work Breakdown Structure

- What is a PERT/CPM Chart ?

A **PERT (Program Evaluation Review Technique)** chart is used for **time estimation**, focusing on uncertain projects with variable task durations, while a **CPM (Critical Path Method)** chart identifies the **longest path of dependent tasks** critical to project completion. Both break down projects into tasks, showing dependencies and determining the fastest timeline.

Step 1: Create a Work Breakdown Structure

- Which Type of Chart is Better?
 - Although a Gantt chart offers a **valuable snapshot view of the project**, PERT charts are more useful for **scheduling, monitoring, and controlling the actual work**
 - PERT and Gantt charts are not mutually exclusive techniques, and project managers often use both methods

Step 1: Create a Work Breakdown Structure

- Identifying Tasks in a Work Breakdown Structure
 - Task or activity
 - Event or milestone
 - Break the project down into smaller tasks, creating a work breakdown structure

Step 1: Create a Work Breakdown

- Identifying Tasks in a Work Breakdown Structure

- Listing the tasks

- Can be challenging, because the tasks **might be embedded in a document**
 - Create a table with columns for task number, description, duration, and predecessor tasks

Task No.	Description	Duration (Days)	Predecessor Tasks
1	Reserve the meeting room		
2	Order the marketing materials		
3	Brief the managers		
4	Send out customer e-mails		
5	Burn sample DVDs		
6	Load the new software		
7	Do a dress rehearsal		

Task Number	Description	Duration (Days)	Predecessor Tasks
1	Project Planning	5	-
2	System Design	10	1
3	Module Development	15	2
4	Integration & Testing	7	3
5	Deployment & Maintenance	3	4

Step 1: Create a Work Breakdown

- Identifying Tasks in a Work Breakdown Structure
 - Estimating Task Duration
 - Person-days (**This measures the amount of work one person can complete in a day.**)
 - Best-case estimate (**B**) (**This is the quickest time a task might take under ideal conditions, without any delays**)
 - Probable-case estimate (**P**) (**A more realistic estimate that considers normal delays and obstacles, often used for reliable planning.**)

Step 1: Create a Work Breakdown Structure

- Identifying Tasks in a Work Breakdown Structure
 - Estimating Task Duration
 - Worst-case estimate (**W**)
 - Weight

$$\frac{(B+4P+W)}{6}$$

6

Step 1: Create a Work Breakdown Structure

- Identifying Tasks in a Work Breakdown Structure
 - Factors Affecting Duration
 - Project size and scope
 - Human resources
 - Experience with similar project
 - constraints

Step 1: Create a Work Breakdown

- Displaying the Work Breakdown Structure
 - If you are managing a complex project with many tasks, you can use task groups, just as you would in a Gantt chart, to simplify the list. This structure allows for easier tracking and delegation, with a clear view of dependencies and the critical path. Would you like assistance in creating a specific WBS structure for your project?
- For example:
 - **Level 1:** Main project phases (e.g., Design, Development, Testing)
 - **Level 2:** Subtasks or deliverables within each phase (e.g., Module 1 Development, Integration Testing)
 - **Level 3:** Further breakdown of tasks if necessary (e.g., Coding, Debugging)

- **Any Questions
regarding the
project?**

Step 2: Identify Task Patterns

- What are Task Patterns?

Software Task Patterns refer to **reusable solutions** or best practices for commonly occurring tasks or problems in software development. **Some common task patterns include:**

- **Design Patterns:** Solutions to recurring design problems in software architecture.
- **Development Patterns:** Repeated approaches to handling tasks like error handling, logging, or resource management.
- **Testing Patterns:** Structured methods for writing and organizing tests.
- **Project Management Patterns:** Approaches to managing tasks in software projects, such as Agile task boards or Scrum sprints
- Task patterns can involve **dependent tasks**, **multiple successor** tasks, and **multiple predecessor tasks**

Step 2: Identify Task Patterns

- How do I Use Task Boxes to Create a Model?

TASK BOX FORMAT

Task Name	
Start Day/Date	Task ID
Finish Day/Date	Task Duration

Step 2: Identify Task Patterns

- What are the Main Types of Task Patterns?
 - Dependent Tasks
 - Multiple successor tasks
 - Concurrent task
 - Predecessor task
 - Successor task
 - Multiple Predecessor Tasks

Step 2: Identify Task Patterns

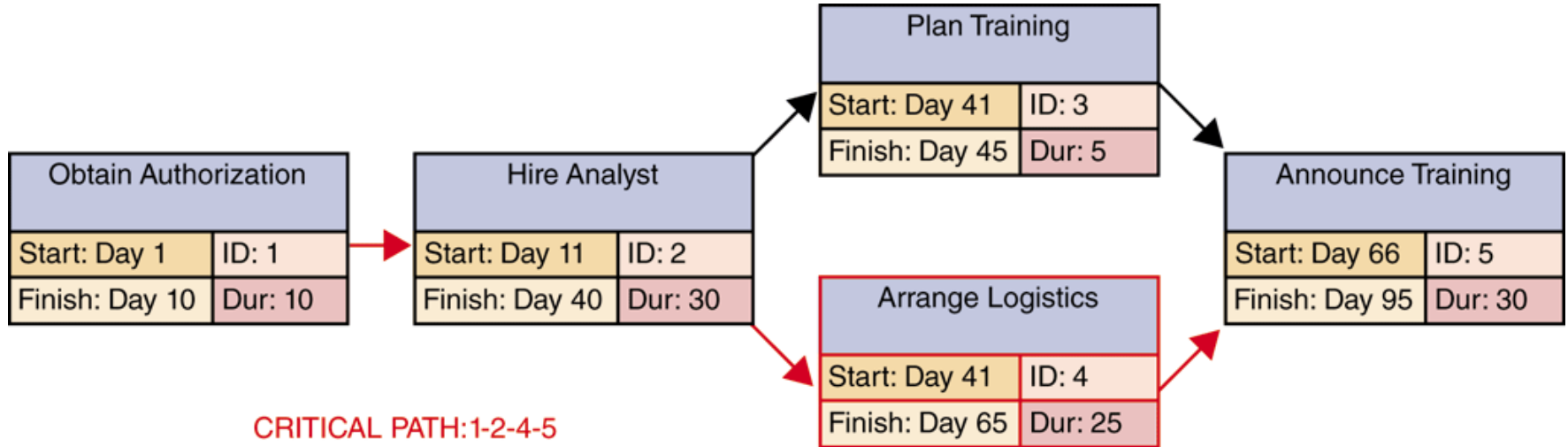
- How Do I Identify Task Patterns?
 - You can identify task patterns by looking carefully at the wording of the task statement
 - Words like *then*, *when*, *or* *and* are action words that signal a sequence of events
- How Do I Work With Complex Task Patterns?
 - When various task patterns combine, you must study the facts carefully in order to understand the logical sequence

Step 2: Identify Task Patterns

- How Do I Work With Complex Task Patterns?
 - Consider the following three fact statements and the task patterns they represent
 - Dependent tasks
 - Dependent tasks and multiple successor tasks
 - Dependent tasks, multiple successor tasks, and multiple predecessor tasks

Step 3: Calculate the Critical Path

- What Is a Critical Path?



Step 3: Calculate the Critical Path

- How Do I Calculate the Critical Path?
 - First, you should review the task patterns
 - The next step is to determine start and finish dates, which will determine the critical path for the project
 - Slack time

Project Monitoring and Control

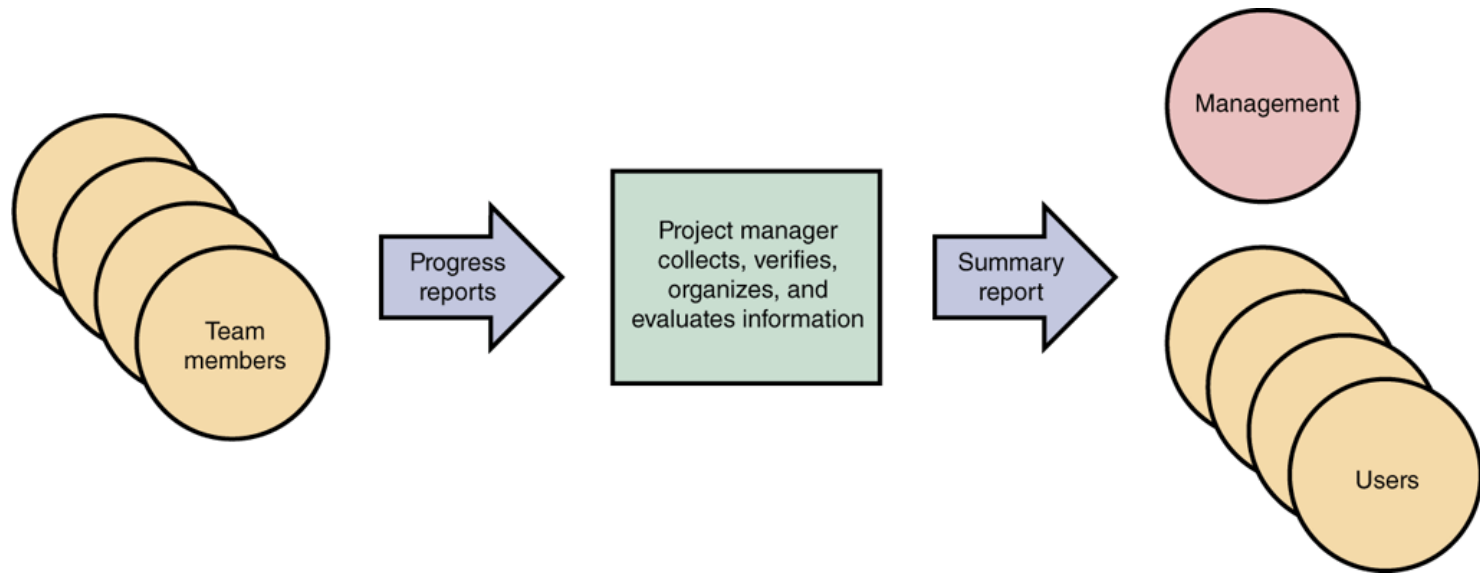
- Monitoring and Control Techniques
 - The project manager must keep track of tasks and progress of team members, compare actual progress with the project plan, verify the completion of project milestones, and set standards and ensure that they are followed
 - Structured walkthrough (it is a formal review process in software development, where developers, testers, and stakeholders collaboratively review a software product or its components (e.g., requirements, design, code) to identify defects, verify compliance with standards, and ensure that it meets predefined criteria. The structured walkthrough is aimed at improving quality by uncovering potential issues early in the development process)

Project Monitoring and Control

- Maintaining a Schedule
 - Maintaining a project schedule can be a challenging task
 - The better the **original plan**, the easier it will be to control the project
 - If **enough milestones** and frequent checkpoints exist, problems will be detected rapidly
 - Project managers often spend most of their time tracking the **tasks along the critical path**

Reporting

- Members of the project team regularly report their progress
- Project Status Meetings



Reporting

- Project Status Reports
 - A project manager must report regularly to his or her immediate supervisor, upper management, and users
 - Should explain what you are doing to handle and monitor the problem
 - Most managers recognize that problems do occur on most projects; it is better **to alert management sooner rather than later**

Project Management Examples

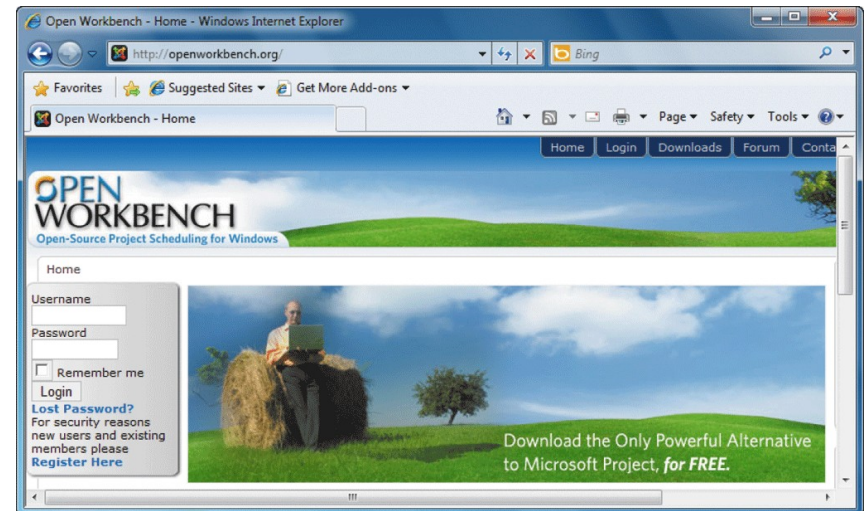
- PERT/CPM Example
 - You construct a PERT/CPM chart from this task list in a two-step process
 - Step 1: Create the work breakdown structure
 - Step 2: Enter start and finish times

TASK BOX FORMAT

Task Name	
Start Day/Date	Task ID
Finish Day/Date	Task Duration

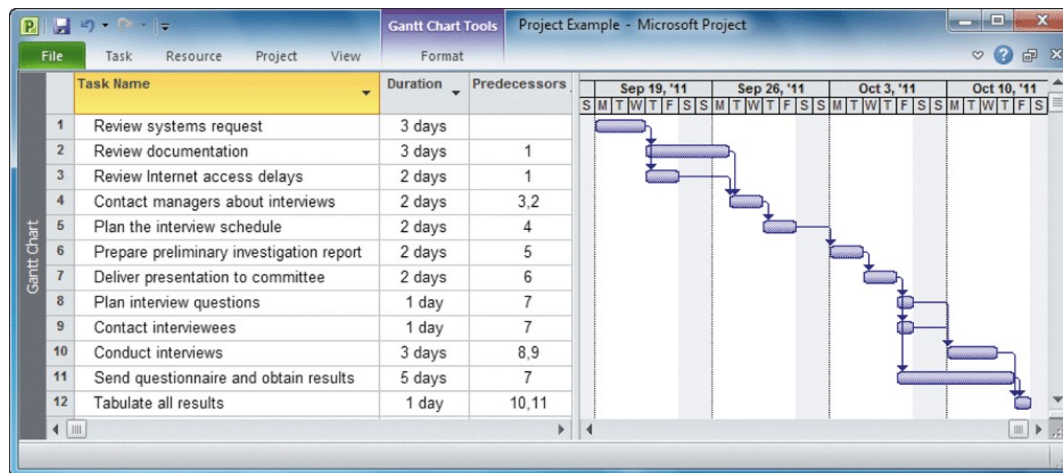
Project Management Examples

- Software-Driven Example
 - Open Workbench
 - Open-source software
 - When you use project management software, you follow the same step-by-step process to develop a WBS and create various types of charts



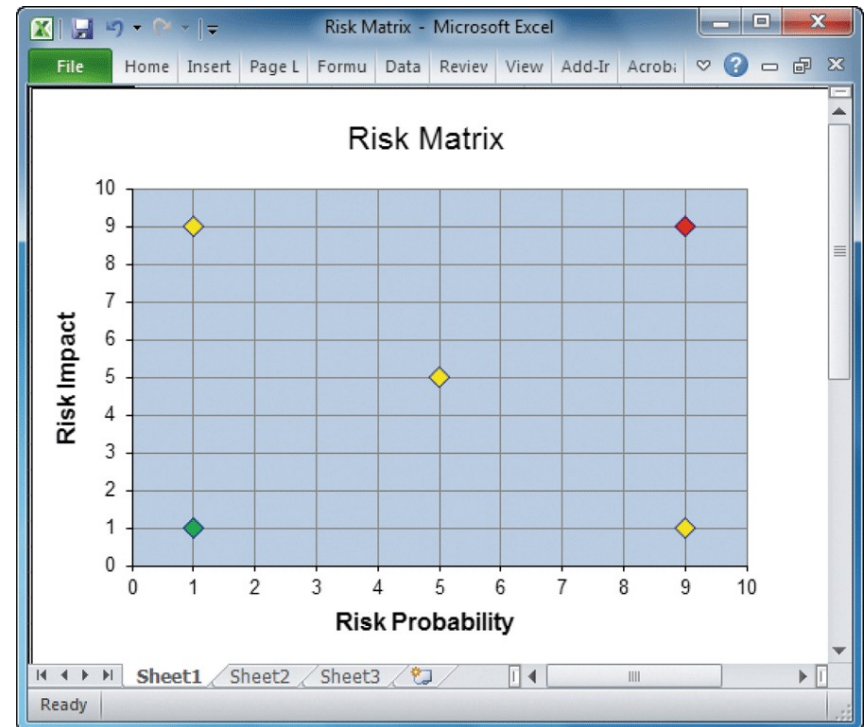
Project Management Examples

- Software-Driven Example
 - Work breakdown structure
 - Gantt chart
 - Network diagram
 - Project planning is a dynamic task and involves constant change



Risk Management

- Every IT project involves risks that systems analysts and project managers must address
- Risk management
- Steps in Risk Management
 - Develop risk management plan
 - Identify the risks
 - Risk identification



Risk Management

- Steps in Risk Management (continued)
 - Analyze the risks
 - Qualitative risk analysis
 - Quantitative risk analysis
 - Create a risk response plan
 - Monitor risks

Risk Management

- Risk Management Software
 - Most project management software includes powerful features
 - The IT team can make a recommendation regarding the risks
 - Depending on the nature and magnitude of the risk, the final decision might be made by management

Managing for Success

- Business Issues
 - The major objective of every system is to provide a solution to a business problem or opportunity
 - A system that falls short of business needs also produces problems for users and reduces employee morale and productivity
 - Project creep

Managing for Success

- Budget Issues
 - Cost overruns typically result from one or more of the following:
 - Unrealistic estimates
 - Failure to develop an accurate forecast that considers all costs over the life of the project
 - Poor monitoring of progress and slow response to early warning signs of problems

Managing for Success

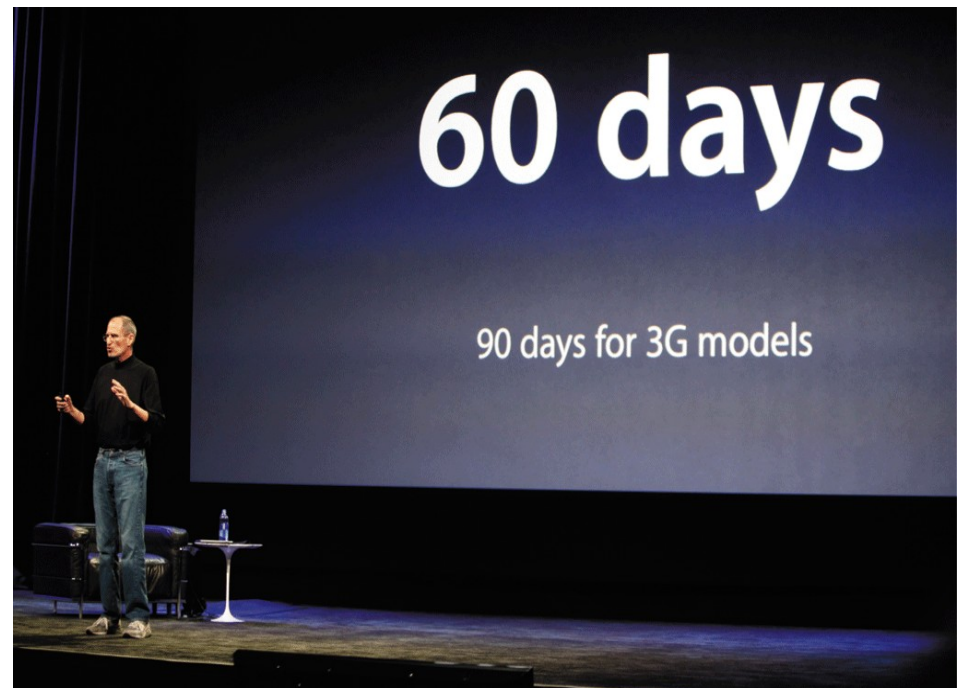
- Budget Issues
 - Cost overruns typically result from one or more of the following:
 - Schedule delays due to factors that were not foreseen
 - Human resource issues, including turnover, inadequate training, and motivation

Managing for Success

- Schedule Issues
 - Problems with timetables and project milestones can indicate a failure to recognize task dependencies, confusion between effort and progress, poor monitoring and control methods, personality conflicts among team members, or turnover of project personnel

The Bottom Line

- When problems occur, the project manager's ability to handle the situation becomes the critical factor



The Bottom Line

- Sometimes, when a project experiences delays or cost overruns, the system still can be delivered on time and within budget if several less critical requirements are trimmed
- Brooks' Law

Chapter Summary

- Project management is the process of planning, scheduling, monitoring and controlling, and reporting upon the development of an information system
- Project managers are responsible for project planning, scheduling, monitoring, and reporting
- Planning, scheduling, monitoring and reporting all take place within a larger project development framework

Chapter Summary

- In project scheduling, the project manager develops a specific time for each task, based on available resources and whether or not the task is dependent on other predecessor tasks
- Every successful information system must support business requirements, stay within budget, and be available on time
- Sound project management involves the same skills as any other management

Chapter Summary

- Chapter 3 complete