Chapter 1

Systems Development Environment
Basic Notions

✓ Information Systems Analysis and Design
  ■ Complex organizational process whereby computer-based information systems are developed and maintained

✓ Application Software
  ■ Computer software designed to support organizational functions or processes

✓ Systems Analyst
  ■ Organizational role most responsible for analysis and design of information systems
A Modern Approach to Systems Analysis and Design

- 1950s: focus on efficient automation of existing processes
- 1960s: advent of 3GL, faster and more reliable computers
- 1970s: system development becomes more like an engineering discipline
- 1980s: major breakthrough with 4GL, CASE tools, object oriented methods
- 1990s: focus on system integration, GUI applications, client/server platforms, Internet
- The new century: Web application development, wireless PDAs, component-based systems
Developing Information Systems and the SDLC

✓ System Development Methodology
  ■ Standard process followed in an organization
  ■ Consists of:
    ✷ Analysis
    ✷ Design
    ✷ Implementation
    ✷ Maintenance
Systems Development Life Cycle (SDLC)

- Traditional methodology for developing, maintaining, and replacing information systems
- Phases in SDLC:
  - Planning
  - Analysis
  - Design
  - Implementation
  - Maintenance
Standard and Evolutionary Views of SDLC
SDLC Planning Phase

Identify, analyze, prioritize, and arrange IS needs

Planning

Maintenance

Analysis

Implementation

Design
SDLC Analysis Phase

Study and structure system requirements

- Planning
- Analysis
- Design
- Implementation
- Maintenance
SDLC Design Phase

- **Logical design**: functional features described independently of computer platform
- **Physical design**: logical specifications transformed to technology-specific details

Convert recommended solution to system specifications
SDLC Implementation Phase

Code, test, install
SDLC Maintenance Phase

Systematically repair and improve the information system
Analysis-Design-Code-Test is in the Heart of System Development

Current practice combines analysis, design, and implementation into a single iterative and parallel process of activities.
Traditional Waterfall SDLC

One phase begins when another completes, little backtracking and looping.
Problems with Waterfall Approach

✓ System requirements “locked in” after being determined (can't change)
✓ Limited user involvement (only in requirements phase)
✓ Too much focus on milestone deadlines of SDLC phases to the detriment (harm) of sound development practices
Alternatives to Traditional Waterfall SDLC

✓ Prototyping
✓ CASE tools
✓ Joint Application Design (JAD)
✓ Rapid Application Development (RAD)
✓ Agile Methodologies
✓ eXtreme Programming
Prototyping

Iterative development process:
- Requirements quickly converted to a working system
- System is continually revised
- Close collaboration between users and analysts
CASE Tools

✓ Computer-Aided Software Engineering
✓ Software tools providing automated support for systems development
✓ Project dictionary/workbook: system description and specifications
✓ Diagramming tools
✓ Example products: Oracle Designer, Rational Rose
Joint Application Design (JAD)

- Structured process involving users, analysts, and managers
- Several-day intensive workgroup sessions
- Purpose: to specify or review system requirements
Rapid Application Development (RAD)

- Methodology to decrease design and implementation time
- Involves: prototyping, JAD, CASE tools, and code generators
Agile Methodologies

✓ Motivated by recognition of software development as fluid, unpredictable, and dynamic

✓ Three key principles
  ■ Adaptive rather than predictive
  ■ Emphasize people rather than roles
  ■ Self-adaptive processes
eXtreme Programming

✓ Short, incremental development cycles
✓ Automated tests
✓ Two-person programming teams
✓ Coding and testing operate together
✓ Advantages:
  ■ Communication between developers
  ■ High level of productivity
  ■ High-quality code
Object-Oriented Analysis and Design

✓ Based on objects rather than data or processes
✓ Object: a structure encapsulating attributes and behaviors of a real-world entity
✓ Object class: a logical grouping of objects sharing the same attributes and behaviors
✓ Inheritance: hierarchical arrangement of classes enable subclasses to inherit properties of superclasses
Rational Unified Process (RUP) involves an iterative, incremental approach to system development.
In this chapter you learned how to:

- Define information systems analysis and design.
- Describe the different types of information systems.
- Describe the information Systems Development Life Cycle (SDLC).
- Explain Rapid Application Development (RAD), prototyping, Joint Application Development (JAD), and Computer Aided Software Engineering (CASE).
- Describe agile methodologies and eXtreme programming.