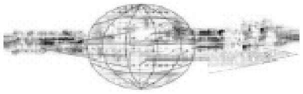
CHAPTER 7
SYSTEMS DEVELOPMENT



Management Information Systems, 10th edition,
By Raymond McLeod, Jr. and George P. Schell

Learning Objectives

- Recognize the systems approach as the basic framework for solving problems of all kinds.
- Know how to apply the systems approach in solving systems problems.
- Understand that the systems development life cycle (SDLC) is a methodology—a recommended way to develop systems.
- Be familiar with the main SDLC approaches—the traditional waterfall cycle, prototyping, rapid application development, phased development, and business process redesign.
- Know the basics of modeling processes with data flow diagrams and use cases.
- Understand how systems development projects are managed in a top-down fashion.

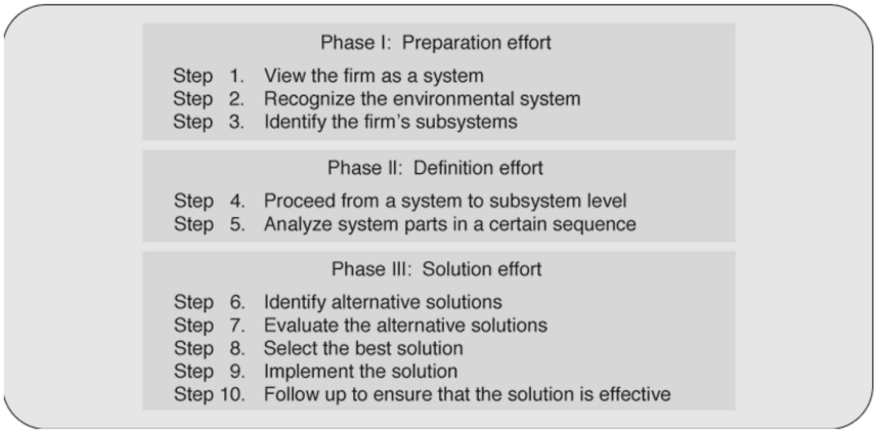
Introduction

- Both managers and systems developers can apply the systems approach when solving problems
- The approach consists of three phases of effort:
  - **Preparation** consists of viewing the firm as a system, recognizing the environmental system, and identifying the firm's subsystems
  - **Definition** involves proceeding from a system to a subsystem level and analyzing system parts in a certain sequence
  - **Solution** involves identifying the alternative solutions, evaluating them, and selecting the best one

THE SYSTEMS APPROACH

- John Dewey identified three series of judgments involved in adequately resolving a controversy
  1. Recognize the controversy
  2. Weigh alternative claims
  3. Form a judgment
- During the late 1960s/early 1970s, interest in systematic problem solving strengthened
- Management scientists and information specialists produced a recommended framework that became known as the **systems approach**—a series of problem-solving steps that ensure the problem is first understood, alternative solutions are considered, and the selected solution works (Figure 7.1)

Figure 7.1 Phases and Steps of the Systems Approach



Phase I: Preparation effort

Step 1. View the firm as a system
Step 2. Recognize the environmental system
Step 3. Identify the firm's subsystems

Phase II: Definition effort

Step 4. Proceed from a system to subsystem level
Step 5. Analyze system parts in a certain sequence

Phase III: Solution effort

Step 6. Identify alternative solutions
Step 7. Evaluate the alternative solutions
Step 8. Select the best solution
Step 9. Implement the solution
Step 10. Follow up to ensure that the solution is effective

The System's Approach (cont.)

1. If a manager can also regard the levels of management as subsystems, the importance of information flows becomes clear
2. A problem trigger – a signal that things are going better/worse than planned – usually stimulates a definition effort
3. A top-down analysis then begins of the system for which the manager is responsible
4. As the manager studies each system level, the system elements are analyzed in sequence (Figure 7.3)

Figure 7.2 Each Business Area Is a System

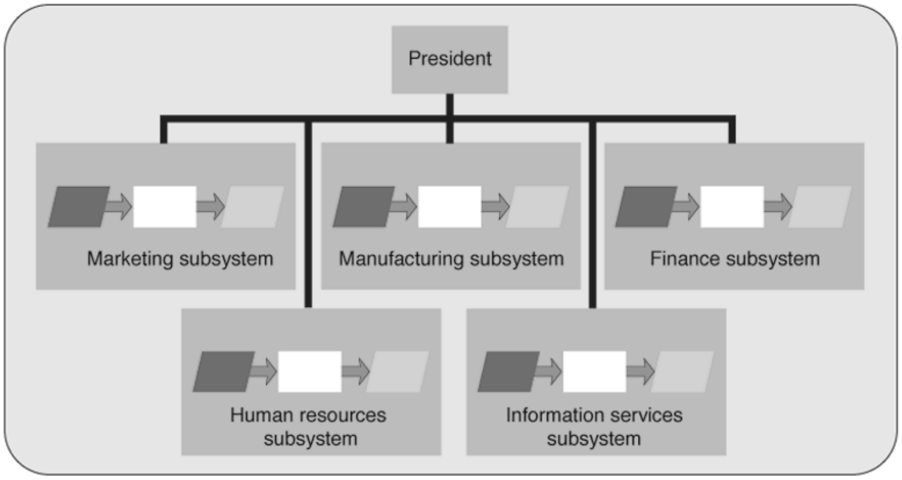
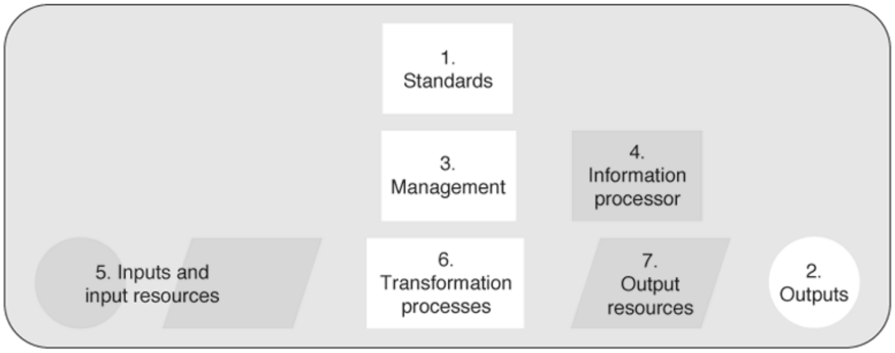
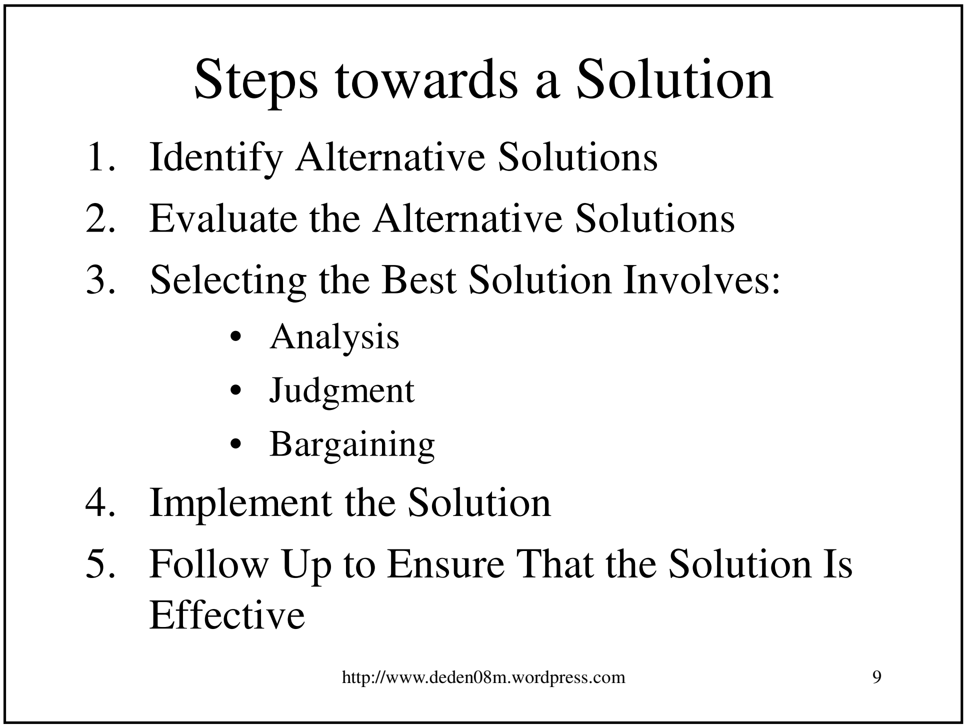


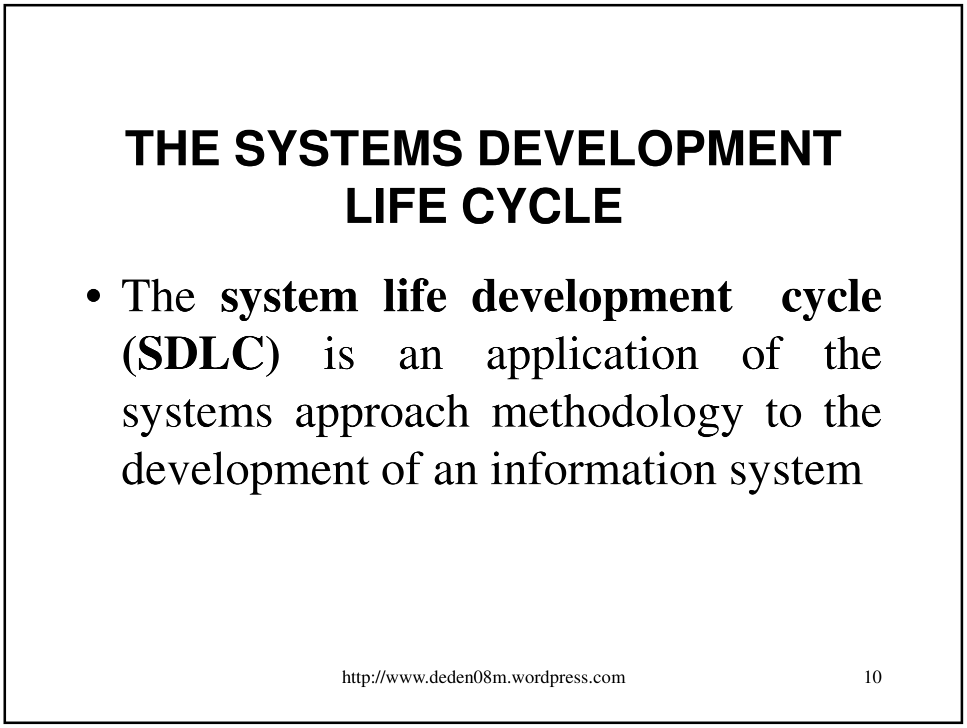
Figure 7.3 Each Part of the System Is Analyzed in Sequence





Steps towards a Solution

1. Identify Alternative Solutions
2. Evaluate the Alternative Solutions
3. Selecting the Best Solution Involves:
    - Analysis
    - Judgment
    - Bargaining
4. Implement the Solution
5. Follow Up to Ensure That the Solution Is Effective



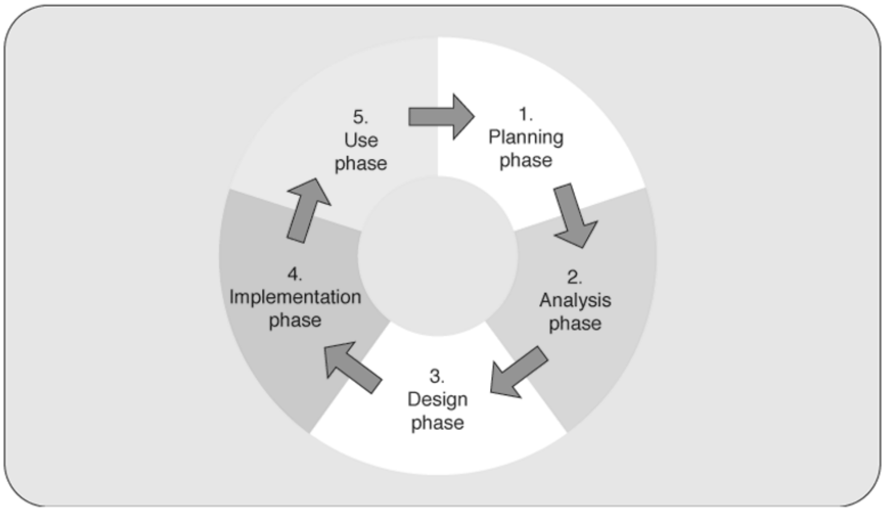
THE SYSTEMS DEVELOPMENT LIFE CYCLE

- The **system life development cycle (SDLC)** is an application of the systems approach methodology to the development of an information system

THE TRADITIONAL SDLC

- It didn't take the first system developers long to recognize a sequence if the project was to have the best chance of success:
  - Planning
  - Analysis
  - Design
  - Implementation
  - Use
- Figure 7.4 illustrates how the life cycle phases can fit into a circular pattern over time

Figure 7.4 The Circular Pattern of the System Life Cycle



PROTOTYPING

- A **prototype** is a version of a potential system that provides the developers and potential users with an idea of how the system will function when completed
- In prototyping, a prototype is produced as quickly as possible, perhaps overnight, to obtain user feedback that will enable the prototype to be improved
- Figure 7.5 shows the four steps involved in developing an evolutionary prototype
- Figure 7.6 shows the steps involved in developing a requirements prototype
- As prototyping has proven to be one of the most successful methodologies, it would be difficult to find a development project that didn't use it to some degree

Figure 7.5 Development of an Evolutionary Prototype

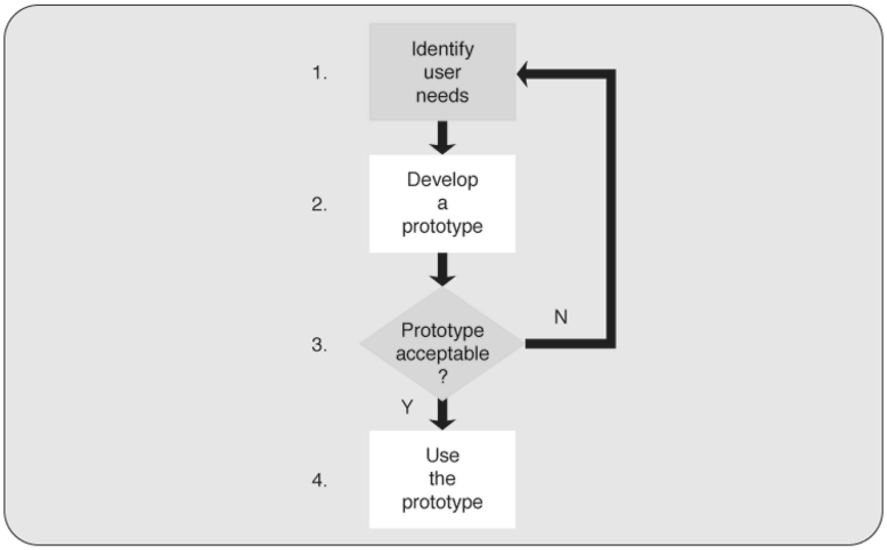
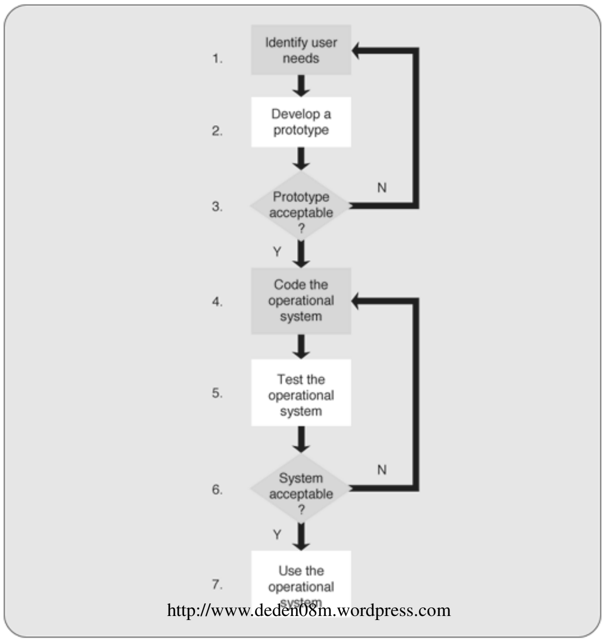


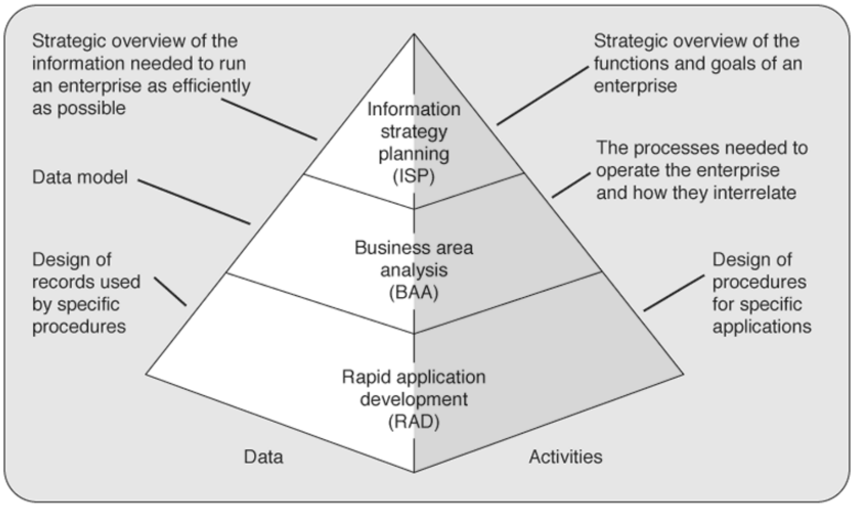
Figure 7.6 Development of a Requirements Prototype


1. Identify user needs
2. Develop a prototype
3. Prototype acceptable ?
N
Y
4. Code the operational system
5. Test the operational system
6. System acceptable ?
N
Y
7. Use the operational system

RAPID APPLICATION DEVELOPMENT

- **Rapid Application Development (RAD)**, is a term coined by James Martin. It refers to a development life cycle intended to produce systems quickly without sacrificing quality
- **Information engineering** is the name that Martin gave to his overall approach to system development, which treats it as a firm-wide activity, while the term **enterprise** is used to describe the entire firm
- Figure 7.7 illustrates the top-down nature of information engineering, involving both data (the left face of the pyramid) and activities (the right face)

Figure 7.7 Rapid Application Development Is an Integral Part of Information Engineering


Strategic overview of the information needed to run an enterprise as efficiently as possible
Data model
Design of records used by specific procedures
Information strategy planning (ISP)
Business area analysis (BAA)
Rapid application development (RAD)
Strategic overview of the functions and goals of an enterprise
The processes needed to operate the enterprise and how they interrelate
Design of procedures for specific applications
Data
Activities

Source: James Martin, Rapid Application Development (New York: Macmillan, 1991), Figures 3.23 and 21.2 (combined). © 1991. Adapted by permission of Prentice Hall, Upper Saddle River, NJ.

RAD (cont.)

- RAD requires four essential ingredients:
  - Management
  - People
  - Methodologies
  - Tools
- Of all the components of information engineering, RAD has probably enjoyed the greatest support

PHASED DEVELOPMENT

- This is an approach for developing information systems that consists of six stages:
  1. Preliminary investigation
  2. Analysis
  3. Design
  4. Preliminary construction
  5. Final construction
  6. System test and installation
- The analysis, design, and preliminary construction stages are taken for each system module
- The six phased development stages are illustrated in Figure 7.8
- Figure 7.9 illustrates how the module phases are integrated into the system development

Figure 7.8 The Stages of the Phased Development Methodology

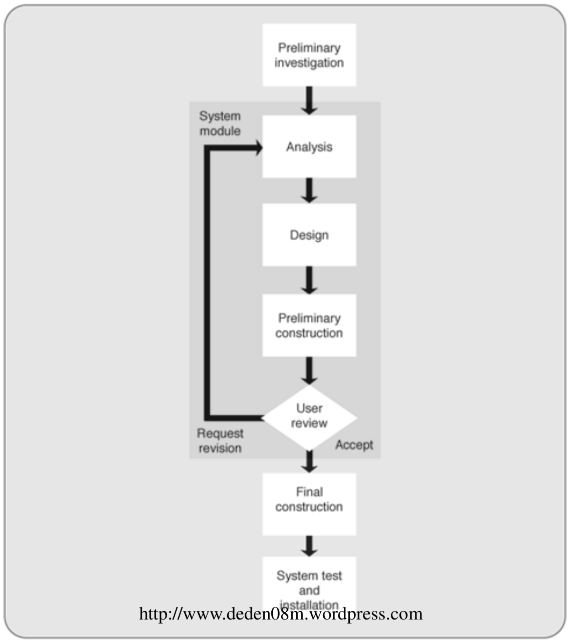
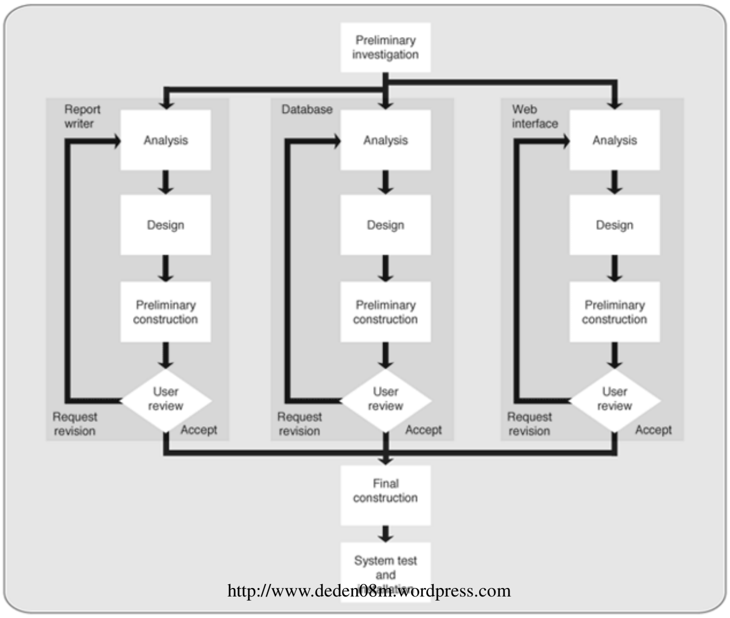


Figure 7.9 Analysis, Design, and Preliminary Construction Are Performed on Each System Module



BUSINESS PROCESS REDESIGN

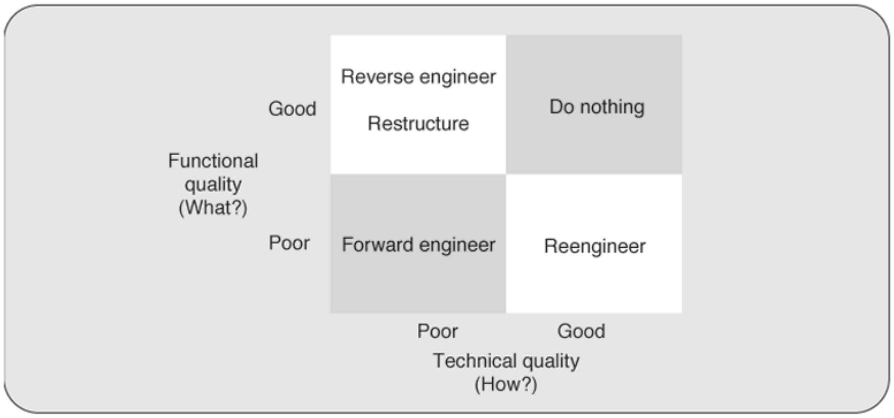
- The process of reworking the systems has been called **reengineering** or **business process redesign (BPR)**
- BPR affects the firm’s IT operation in two ways:
  1. IT can apply BPR to the redesign of **legacy systems** that can no longer be kept alive by ordinary maintenance
  2. When a firm applies BPR to its major operations, the effort invariably has a ripple effect that results in the redesign of information systems
- IT has devised **reverse engineering**, **restructuring**, and **reengineering** that can be applied separately or in combination for applying BPR

Three Techniques for Applying BPR

1. As used in computing, **reverse engineering** is the process of analyzing an existing system to identify its elements and their interrelationships, as well as to create documentation in a higher level of abstraction than currently exists.
2. **Restructuring** is the transformation of a system into another form without changing its functionality
3. **Reengineering** is the complete redesign of a system with the objective of changing its functionality

- The proper mix depends on the current state of the system in terms of its functional and technical quality. Figure 7.10 is a diagram that shows these two influences

Figure 7.10 BPR Component Selection Is Based on Both Functional and Technical Quality



		Technical quality (How?)	
		Poor	Good
Functional quality (What?)	Good	Reverse engineer Restructure	Do nothing
	Poor	Forward engineer	Reengineer

Sources: David Sharon, "The Psychology of Reengineering," IEEE Software 8 (November 1991), 74 © 1990 IEEE; and "Three R's: A White Paper on Application Re-Development," The Re-Development Investigation Team, Texaco Information Systems Enabling Center, Texaco, Inc. (January 30, 1992), 10.

PUTTING THE TRADITIONAL SDLC, PROTOTYPING, RAD, PHASED DEVELOPMENT, AND BPR IN PERSPECTIVE

- The traditional SDLC, prototyping, RAD, and BPR are methodologies that are recommended ways of developing an information system
- Currently, firms are revamping many systems that were implemented with computer technology that is now obsolete
- The name BPR is used for this. Prototyping, RAD, and phased development can be utilized in a BPR project to meet users' needs in a responsive way

Process Modeling

- As developers perform analysis and design, they model the system data, processes, and objects
- A **data flow diagram (DFD)** is a graphic representation of a system that uses four symbol shapes representing: (1) environmental elements with which the system interfaces, (2) processes, (3) data flows, and (4) storage of data – to illustrate how data flows through interconnected processes
- Figure 7.11 illustrates a DFD system that a firm might use to compute commissions for its sales representatives
- Figure 7.12 is a context diagram of the sales commission system
- Figure 7.13 shows a **Figure 4 diagram**

Figure 7.11 A Data Flow Diagram of a Sales Commission System

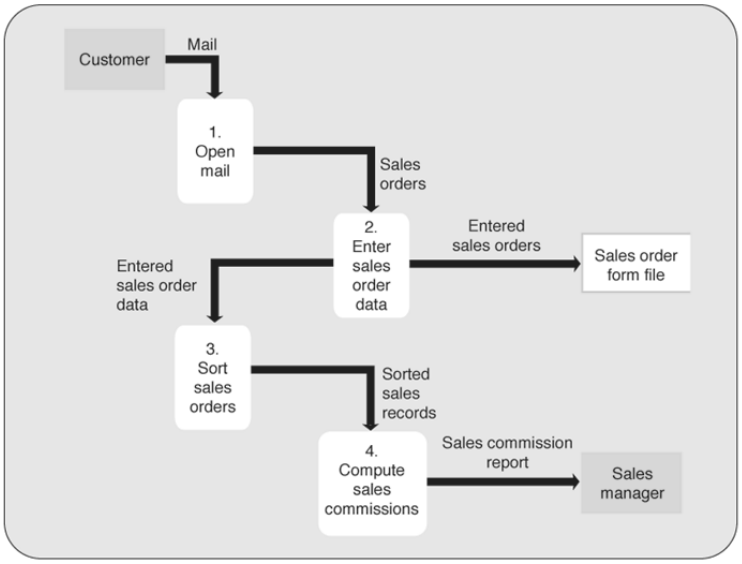


Figure 7.12 A Context Diagram of a Sales Commission System

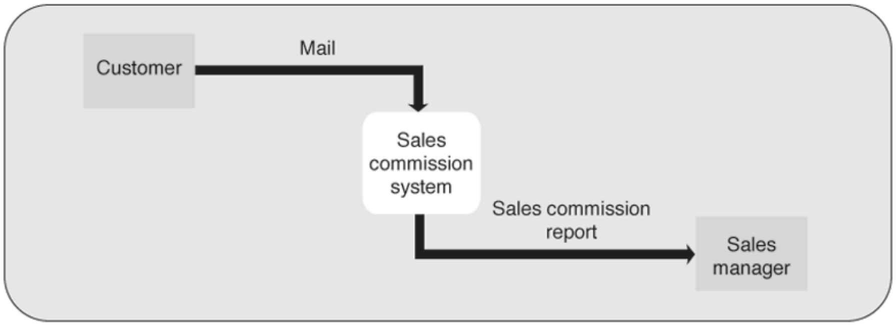
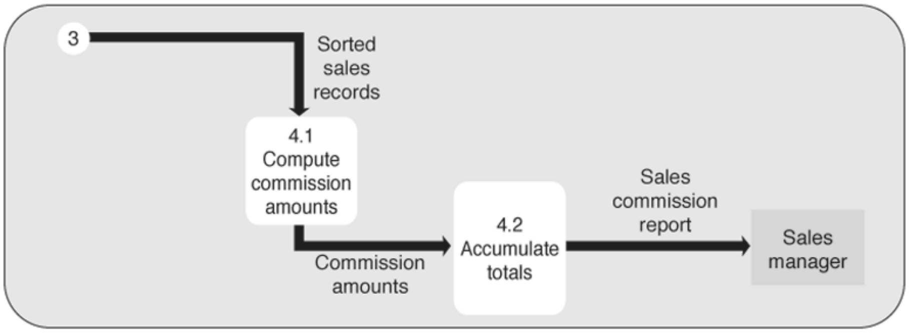


Figure 7.13 A Figure 4 Diagram of a Sales Commission System



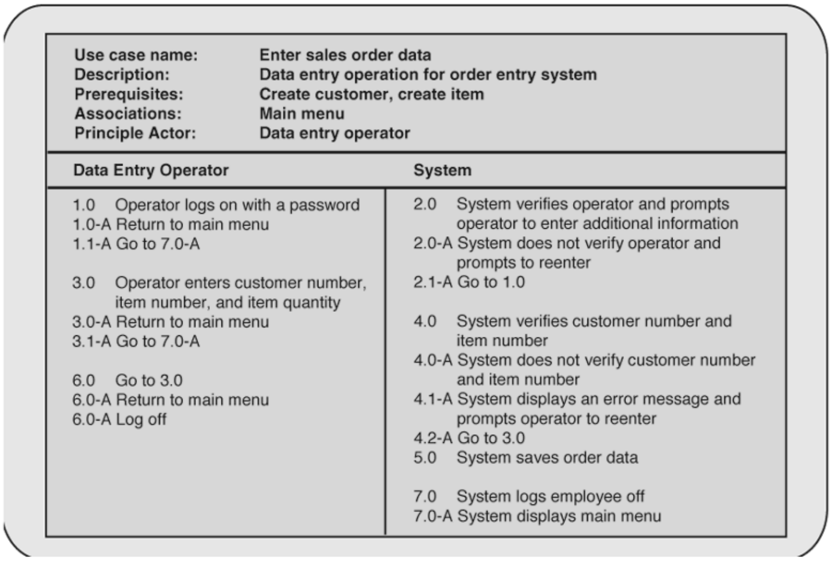
Use Cases

- A **use case** is a narrative description in an outline form of the dialog that occurs between a primary (usually a computer program) and a secondary system (a person interacting with the computer program)

There are two use case formats:

- A continuous narrative with each action numbered sequentially;and
- The other is called the **ping pong format** because it consists of two narratives and the numbering indicates how the tasks alternate between the primary and secondary systems (Figure 7.14)
- A set of guidelines for preparing a use case in the ping pong format is shown in Figure 7.15

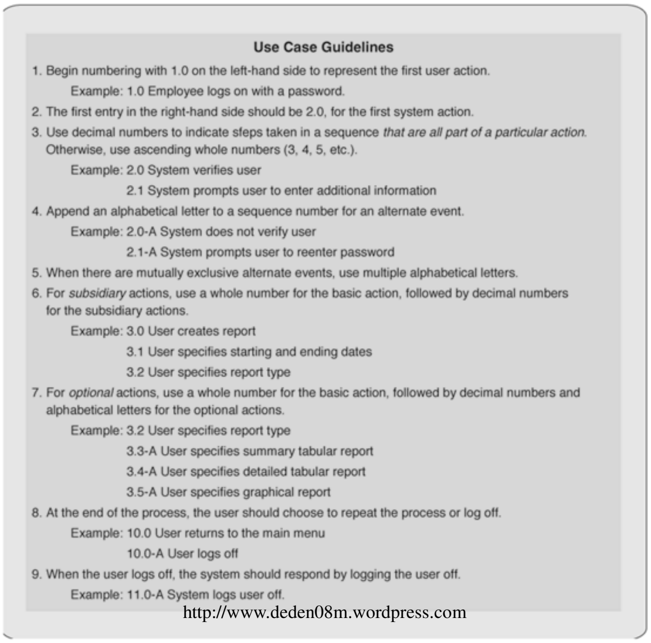
Figure 7.14 A Use Case



Use case name:	**Enter sales order data**
Description:	**Data entry operation for order entry system**
Prerequisites:	**Create customer, create item**
Associations:	**Main menu**
Principle Actor:	**Data entry operator**

Data Entry Operator	**System**
1.0 Operator logs on with a password 1.0-A Return to main menu 1.1-A Go to 7.0-A	2.0 System verifies operator and prompts operator to enter additional information 2.0-A System does not verify operator and prompts to reenter 2.1-A Go to 1.0
3.0 Operator enters customer number, item number, and item quantity 3.0-A Return to main menu 3.1-A Go to 7.0-A	4.0 System verifies customer number and item number 4.0-A System does not verify customer number and item number 4.1-A System displays an error message and prompts operator to reenter 4.2-A Go to 3.0 5.0 System saves order data
6.0 Go to 3.0 6.0-A Return to main menu 6.0-A Log off	7.0 System logs employee off 7.0-A System displays main menu

Figure 7.15 Use Case Guidelines



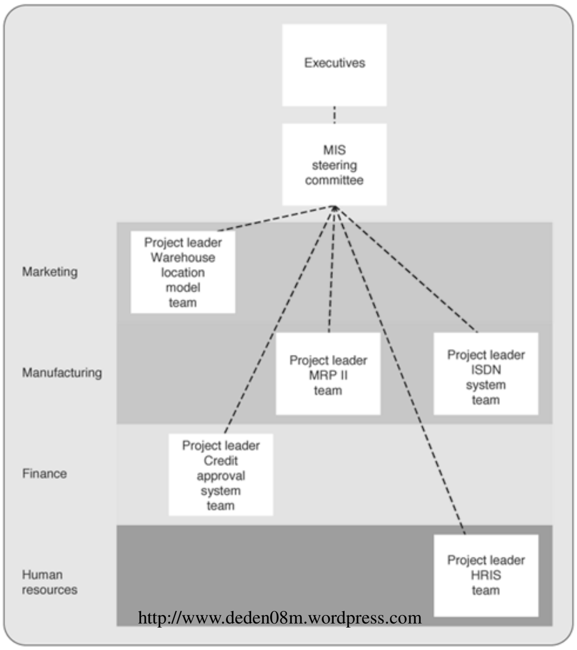
Use Case Guidelines

1. Begin numbering with 1.0 on the left-hand side to represent the first user action.
   Example: 1.0 Employee logs on with a password.
2. The first entry in the right-hand side should be 2.0, for the first system action.
3. Use decimal numbers to indicate sfeps taken in a sequence *that are all part of a particular action.* Otherwise, use ascending whole numbers (3, 4, 5, etc.).
   Example: 2.0 System verifies user
   2.1 System prompts user to enter additional information
4. Append an alphabetical letter to a sequence number for an alternate event.
   Example: 2.0-A System does not verify user
   2.1-A System prompts user to reenter password
5. When there are mutually exclusive alternate events, use multiple alphabetical letters.
6. For *subsidiary* actions, use a whole number for the basic action, followed by decimal numbers for the subsidiary actions.
   Example: 3.0 User creates report
   3.1 User specifies starting and ending dates
   3.2 User specifies report type
7. For *optional* actions, use a whole number for the basic action, followed by decimal numbers and alphabetical letters for the optional actions.
   Example: 3.2 User specifies report type
   3.3-A User specifies summary tabular report
   3.4-A User specifies detailed tabular report
   3.5-A User specifies graphical report
8. At the end of the process, the user should choose to repeat the process or log off.
   Example: 10.0 User returns to the main menu
   10.0-A User logs off
9. When the user logs off, the system should respond by logging the user off.
   Example: 11.0-A System logs user off.

PROJECT MANAGEMENT

- Today, it is possible for life cycle management to span several organizational levels and involve managers outside of IT
- Figure 7.16 shows the hierarchical nature of project management
- In this example, there are five development projects going at the same time, all managed by the MIS steering committee

Figure 7.16 Managers of a System Life Cycle Are Arranged in a Hierarchy


Executives
MIS steering committee
Marketing
Project leader Warehouse location model team
Manufacturing
Project leader MRP II team
Project leader ISDN system team
Finance
Project leader Credit approval system team
Human resources
Project leader HRIS team

The MIS Steering Committee

- The **MIS Steering Committee** performs three main functions:
  - It **establishes policies** that ensure computer support for achieving the strategic objectives of the firm
  - It **provides fiscal control** by serving as the approval authority for all requests for computer-related funds
  - It **resolves conflicts** that arise concerning priorities for computer use

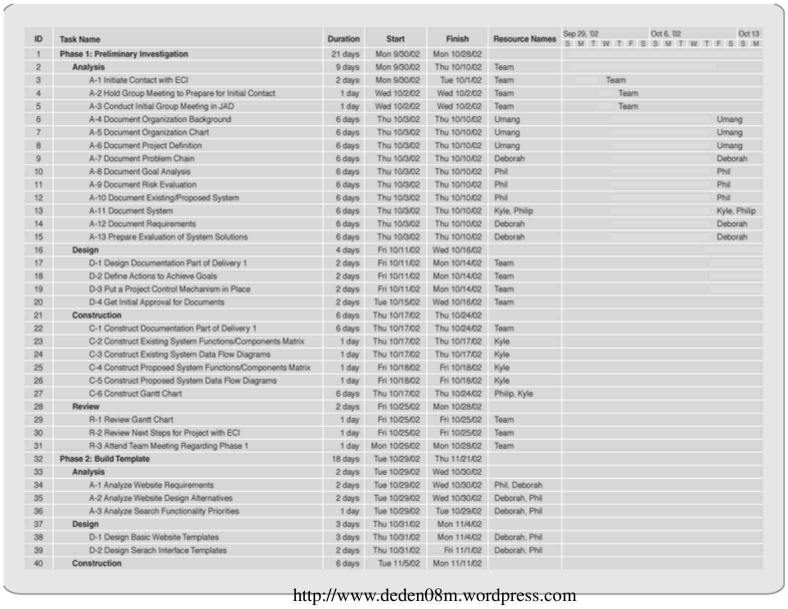
Project Leadership

- A **project team** includes all of the persons who participate in the development of an information system
- A team might have as many as a dozen members, consisting of some combination of users, information specialists, and may include an internal auditor
- A team or project leader, who provides direction throughout the life of the project, directs the team activity

The Project Management Mechanism

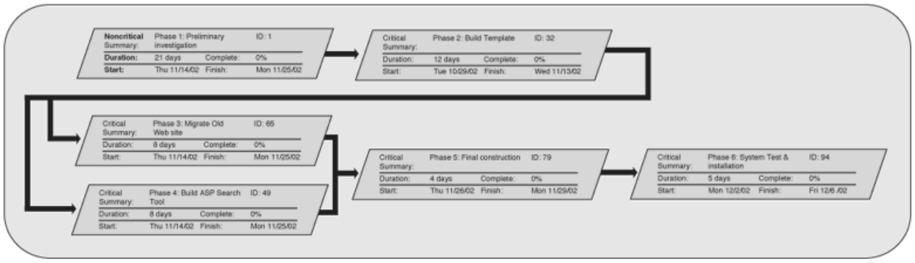
- The basis for project management is the project plan
- A popular format for a detailed plan is a Gantt chart, which identifies the tasks, who will perform them, and when they will be performed
- A **Gantt chart** is a horizontal bar chart that includes a bar for each task to be performed arranged in a time sequence
- Figure 7.17 is the first part of a Gantt chart, prepared using a Microsoft Excel spreadsheet
- A complement to the Gantt chart is the **network diagram.** Figure 7.18 is a high-level network diagram that identifies the phases of a project

Figure 7.17 A Gantt Chart



ID	Task Name	Duration	Start	Finish	Resource Names
1	**Phase 1: Preliminary Investigation**	21 days	Mon 9/30/02	Mon 10/28/02	
2	**Analysis**	9 days	Mon 9/30/02	Thu 10/10/02	Team
3	A-1 Initiate Contact with ECI	2 days	Mon 9/30/02	Tue 10/1/02	Team
4	A-2 Hold Group Meeting to Prepare for Initial Contact	1 day	Wed 10/2/02	Wed 10/2/02	Team
5	A-3 Conduct Initial Group Meeting in JAD	1 day	Wed 10/2/02	Wed 10/2/02	Team
6	A-4 Document Organization Background	6 days	Thu 10/3/02	Thu 10/10/02	Umang
7	A-5 Document Organization Chart	6 days	Thu 10/3/02	Thu 10/10/02	Umang
8	A-6 Document Project Definition	6 days	Thu 10/3/02	Thu 10/10/02	Umang
9	A-7 Document Problem Chain	6 days	Thu 10/3/02	Thu 10/10/02	Deborah
10	A-8 Document Goal Analysis	6 days	Thu 10/3/02	Thu 10/10/02	Phil
11	A-9 Document Risk Evaluation	6 days	Thu 10/3/02	Thu 10/10/02	Phil
12	A-10 Document Existing/Proposed System	6 days	Thu 10/3/02	Thu 10/10/02	Phil
13	A-11 Document System	6 days	Thu 10/3/02	Thu 10/10/02	Kyle, Philip
14	A-12 Document Requirements	6 days	Thu 10/3/02	Thu 10/10/02	Deborah
15	A-13 Prepare Evaluation of System Solutions	6 days	Thu 10/3/02	Thu 10/10/02	Deborah
16	**Design**	4 days	Fri 10/11/02	Wed 10/16/02	
17	D-1 Design Documentation Part of Delivery 1	2 days	Fri 10/11/02	Mon 10/14/02	Team
18	D-2 Define Actions to Achieve Goals	2 days	Fri 10/11/02	Mon 10/14/02	Team
19	D-3 Put a Project Control Mechanism in Place	2 days	Fri 10/11/02	Mon 10/14/02	Team
20	D-4 Get Initial Approval for Documents	2 days	Tue 10/15/02	Wed 10/16/02	Team
21	**Construction**	6 days	Thu 10/17/02	Thu 10/24/02	
22	C-1 Construct Documentation Part of Delivery 1	6 days	Thu 10/17/02	Thu 10/24/02	Team
23	C-2 Construct Existing System Functions/Components Matrix	1 day	Thu 10/17/02	Thu 10/17/02	Kyle
24	C-3 Construct Existing System Data Flow Diagrams	1 day	Thu 10/17/02	Thu 10/17/02	Kyle
25	C-4 Construct Proposed System Functions/Components Matrix	1 day	Fri 10/18/02	Fri 10/18/02	Kyle
26	C-5 Construct Proposed System Data Flow Diagrams	1 day	Fri 10/18/02	Fri 10/18/02	Kyle
27	C-6 Construct Gantt Chart	6 days	Thu 10/17/02	Thu 10/24/02	Philip, Kyle
28	**Review**	2 days	Fri 10/25/02	Mon 10/28/02	
29	R-1 Review Gantt Chart	1 day	Fri 10/25/02	Fri 10/25/02	Team
30	R-2 Review Next Steps for Project with ECI	1 day	Fri 10/25/02	Fri 10/25/02	Team
31	R-3 Attend Team Meeting Regarding Phase 1	1 day	Mon 10/26/02	Mon 10/28/02	Team
32	**Phase 2: Build Template**	18 days	Tue 10/29/02	Thu 11/21/02	
33	**Analysis**	2 days	Tue 10/29/02	Wed 10/30/02	
34	A-1 Analyze Website Requirements	2 days	Tue 10/29/02	Wed 10/30/02	Phil, Deborah
35	A-2 Analyze Website Design Alternatives	2 days	Tue 10/29/02	Wed 10/30/02	Deborah, Phil
36	A-3 Analyze Search Functionality Priorities	1 day	Tue 10/29/02	Tue 10/29/02	Deborah, Phil
37	**Design**	3 days	Thu 10/31/02	Mon 11/4/02	
38	D-1 Design Basic Website Templates	3 days	Thu 10/31/02	Mon 11/4/02	Deborah, Phil
39	D-2 Design Serach Interface Templates	2 days	Thu 10/31/02	Fri 11/1/02	Deborah, Phil
40	**Construction**	6 days	Tue 11/5/02	Mon 11/11/02	

Figure 7.18 A Network Diagram



Noncritical Summary: Phase 1: Preliminary investigation ID: 1
Duration: 21 days Complete: 0%
Start: Thu 11/14/02 Finish: Mon 11/25/02

Critical Summary: Phase 2: Build Template ID: 32
Duration: 12 days Complete: 0%
Start: Tue 10/29/02 Finish: Wed 11/13/02

Critical Summary: Phase 3: Migrate Old Web site ID: 65
Duration: 8 days Complete: 0%
Start: Thu 11/14/02 Finish: Mon 11/25/02

Critical Summary: Phase 4: Build ASP Search Tool ID: 49
Duration: 8 days Complete: 0%
Start: Thu 11/14/02 Finish: Mon 11/25/02

Critical Summary: Phase 5: Final construction ID: 79
Duration: 4 days Complete: 0%
Start: Thu 11/26/02 Finish: Mon 11/29/02

Critical Summary: Phase 6: System Test & installation ID: 94
Duration: 5 days Complete: 0%
Start: Mon 12/2/02 Finish: Fri 12/6 /02

END OF CHAPTER 7