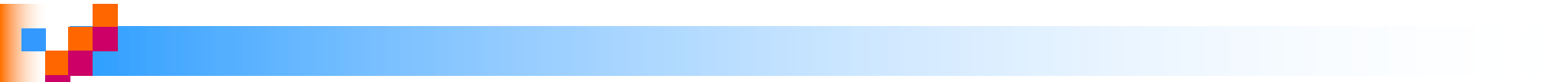# Modern Systems Analysis and Design
Seventh Edition

Jeffrey A. Hoffer

Joey F. George

Joseph S. Valacich

Chapter 13

System Implementation

Learning Objectives

✓ Describe the process of coding, testing, and converting an organizational information system and outline the deliverables and outcomes of the process.

✓ Prepare a test plan for an information system.

✓ Apply four installation strategies: direct, parallel, single-location, and phased installation.

✓ List the deliverables for documenting the system and for training and supporting users.

✓ Compare the many modes available for organizational information system training, including self-training and electronic performance support systems.

Learning Objectives (Cont.)

- ✓ Discuss the issues of providing support for end-users.
- ✓ Explain why system implementation sometimes fails.
- ✓ Describe the threats to system security and remedies that can be applied.
- ✓ Show how traditional implementation issues apply to electronic commerce applications.
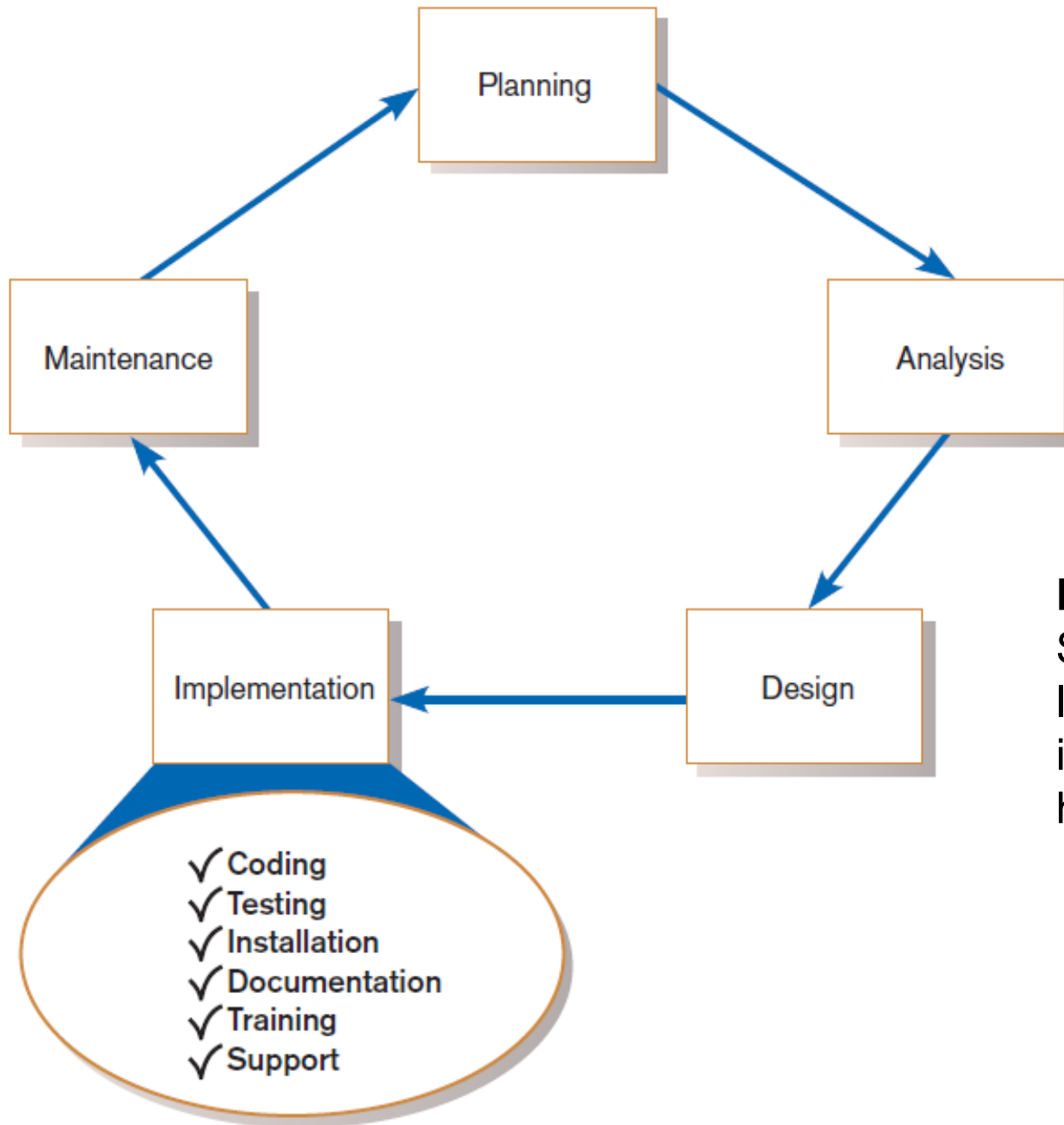
© 2014 Pearson Education, Inc. Publishing as Prentice Hall

FIGURE 13-1
Systems development life cycle with the implementation phase highlighted

Planning

Analysis

Design

Implementation
- ✓ Coding
- ✓ Testing
- ✓ Installation
- ✓ Documentation
- ✓ Training
- ✓ Support

Maintenance

System Implementation

- Six major activities:
  - ☐ Coding
  - ☐ Testing
  - ☐ Installation
  - ☐ Documentation
  - ☐ Training
  - ☐ Support

System Implementation (Cont.)

- Purpose:
  - ☐ To convert final physical system specifications into working and reliable software
  - ☐ To document work that has been done
  - ☐ To provide help for current and future users

The Process of Coding, Testing, and Installation

- *Coding*
  - Physical design specifications are turned into working computer code.

- *Testing*
  - Tests are performed using various strategies.
  - Testing is performed in parallel with coding.

- *Installation*
  - The current system is replaced by a new system.

TABLE 13-1 Deliverables for Coding, Testing, and Installation

1. Coding
   a. Code
   b. Program documentation
2. Testing
   a. Test scenarios (test plan) and test data
   b. Results of program and system testing
3. Installation
   a. User guides
   b. User training plan
   c. Installation and conversion plan
      i. Software and hardware installation schedule
      ii. Data conversion plan
      iii. Site and facility remodeling plan

Documenting the System, Training Users, and Supporting Users

- Two audiences for final documentation:
  - Information systems personnel who will maintain the system throughout its productive life
  - People who will use the system as part of their daily lives

- User Training
  - Application-specific
  - General for operating system and off-the-shelf software

© 2014 Pearson Education, Inc. Publishing as Prentice Hall

TABLE 13-2   Deliverables for Documenting the System, Training, and Supporting Users

1. Documentation
   a. System documentation
   b. User documentation
2. User Training Plan
   a. Classes
   b. Tutorials
3. User Training Modules
   a. Training materials
   b. Computer-based training aids
4. User Support Plan
   a. Help desk
   b. Online help
   c. Bulletin boards and other support mechanisms

Software Application Testing

- A master test plan is developed during the analysis phase.

- During the design phase, unit, system and integration test plans are developed.

- The actual testing is done during implementation.

- Written test plans provide improved communication among all parties involved in testing.

TABLE 13-3   Table of Contents of a Master Test Plan

1. Introduction
   a. Description of system to be tested
   b. Objectives of the test plan
   c. Method of testing
   d. Supporting documents
2. Overall Plan
   a. Milestones, schedules, and locations
   b. Test materials
      i. Test plans
      ii. Test cases
      iii. Test scenarios
      iv. Test log
   c. Criteria for passing tests
3. Testing Requirements
   a. Hardware
   b. Software
   c. Personnel

4. Procedure Control
   a. Test initiation
   b. Test execution
   c. Test failure
   d. Access/change control
   e. Document control
5. Test-Specific or Component-Specific Test Plans
   a. Objectives
   b. Software description
   c. Method
   d. Milestones, schedule, progression, and locations
   e. Requirements
   f. Criteria for passing tests
   g. Resulting test materials
   h. Execution control
   i. Attachments

(*Source:* Adapted from Mosley, 1993.)

Seven Different Types of Tests

- Static or dynamic techniques
  - ☐ Static testing means that the code being tested is not executed.
  - ☐ Dynamic testing involves execution of the code.
- Test is automated or manual
  - ☐ Automated means computer conducts the test.
  - ☐ Manual means that people complete the test.

Seven Different Types of Tests

- **Inspection**: a testing technique in which participants examine program code for predictable language-specific errors

- **Walkthrough**: a peer group review of any product created during the systems development process, including code

- **Desk checking**: a testing technique in which the program code is sequentially executed manually by the reviewer

© 2014 Pearson Education, Inc. Publishing as Prentice Hall

Seven Different Types of Tests (Cont.)

- **Unit testing**: each module is tested alone in an attempt to discover any errors in its code

- **Integration testing**: the process of bringing together all of the modules that a program comprises for testing purposes
  - Modules are typically integrated in a top-down incremental fashion.

© 2014 Pearson Education, Inc. Publishing as Prentice Hall

Seven Different Types of Tests (Cont.)

- **System testing**: the bringing together of all of the programs that a system comprises for testing purposes
  - Programs are typically integrated in a top-down, incremental fashion.

Seven Different Types of Tests (Cont.)

- **Stub testing**: a technique used in testing modules, especially where modules are written and tested in a top-down fashion, where a few lines of code are used to substitute for subordinate modules

The Testing Process

- The purpose of testing is to confirm that the system satisfies the requirements.

- Testing must be planned.

- **Test case** is a specific scenario of transactions, queries or navigation paths.

The Testing Process (Cont.)

- Test cases represent either:
  - □ Typical system use
  - □ Critical system use, or
  - □ Abnormal system use.
- Test cases and results should be thoroughly documented so they can be repeated for each revision of an application.

© 2014 Pearson Education, Inc. Publishing as Prentice Hall

FIGURE 13-4

Test case results form

(*Source: Adapted from Mosley, 1993.*)

Pine Valley Furniture Company
Test Case Results

Test Case Number:
Date:

Program Name:
Module Under Test:

Explanation of difference between actual and expected output:

Suggestions for next steps:

Testing Harness

- Automated testing environment
- Reviews code for:
  - Errors
  - Standards violations
  - Other design flaws
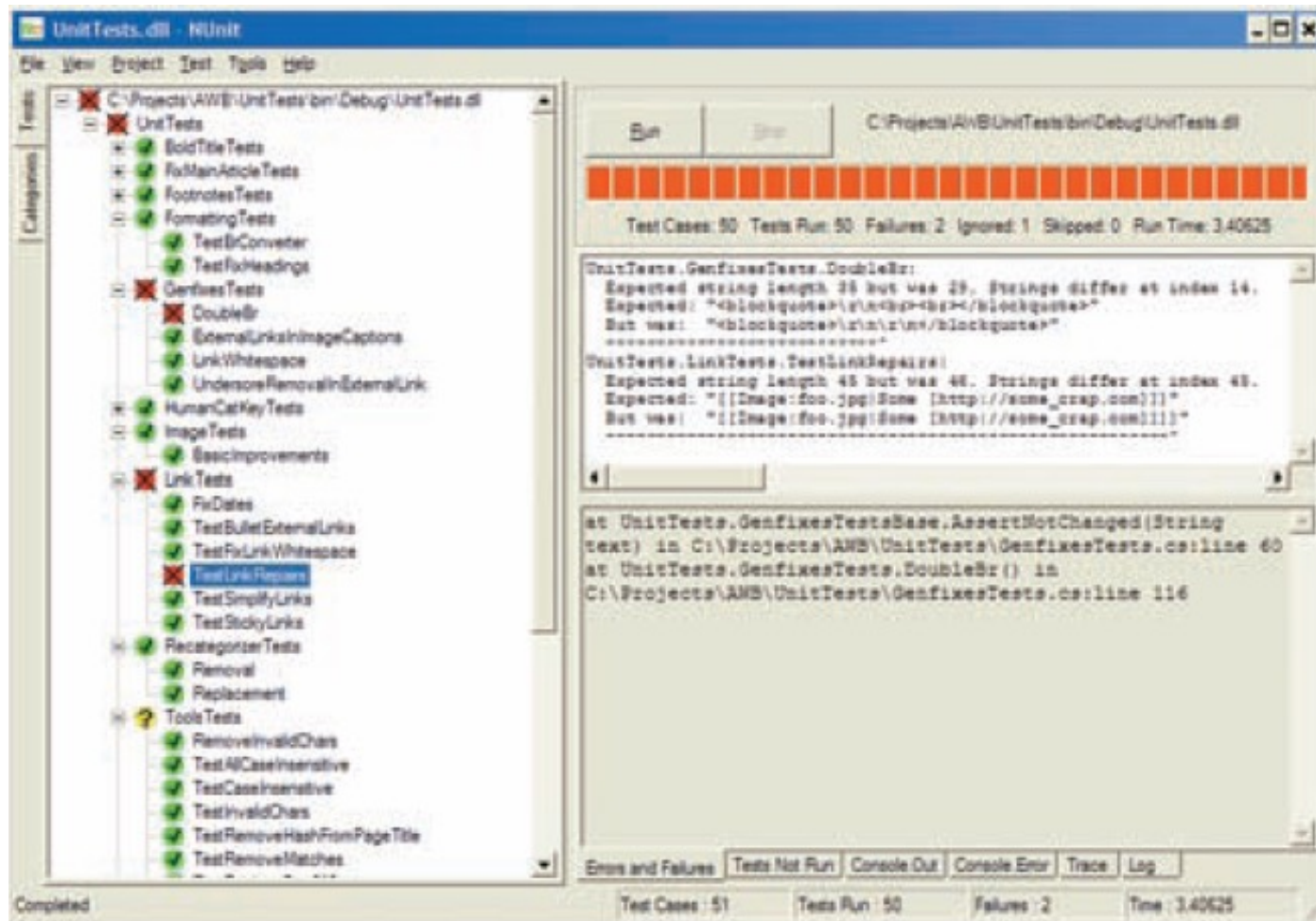- Expand the scope of the tests beyond the current development platform

Testing Harness (Cont.)

Figure 13-5
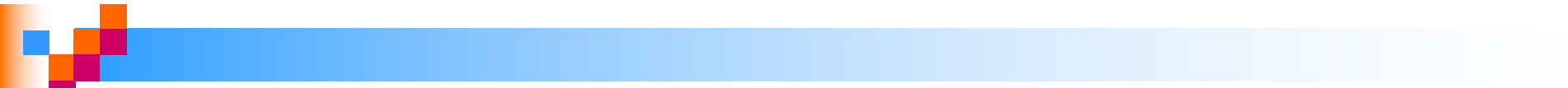NUnit, a unit testing framework for .NET

© 2014 Pearson Education, Inc. Publishing as Prentice Hall

Combining Coding and Testing

- Coding and testing often go together.

- Big companies have dedicated test staff.

- With eXtreme programming (XP) a common technique is *refactoring*.

- **Refactoring** = making a program simpler after adding a new feature

Acceptance Testing by Users

- **Acceptance testing**: the process whereby actual users test a completed information system, the end result of which is the users' acceptance of it

Acceptance Testing by Users (Cont.)

- **Alpha testing**: user testing of a completed information system using simulated data

- **Beta testing**: user testing of a completed information system using real data in the real user environment

Acceptance Testing by Users (Cont.)

- ## Types of Alpha Test:

  - ☐ *Recovery testing* — forces software (or environment) to fail in order to verify that recovery is properly performed

  - ☐ *Security testing* — verifies that protection mechanisms built into the system will protect it from improper penetration

  - ☐ *Stress testing* — tries to break the system

  - ☐ *Performance testing* — determines how the system performs on the range of possible environments in which it may be used
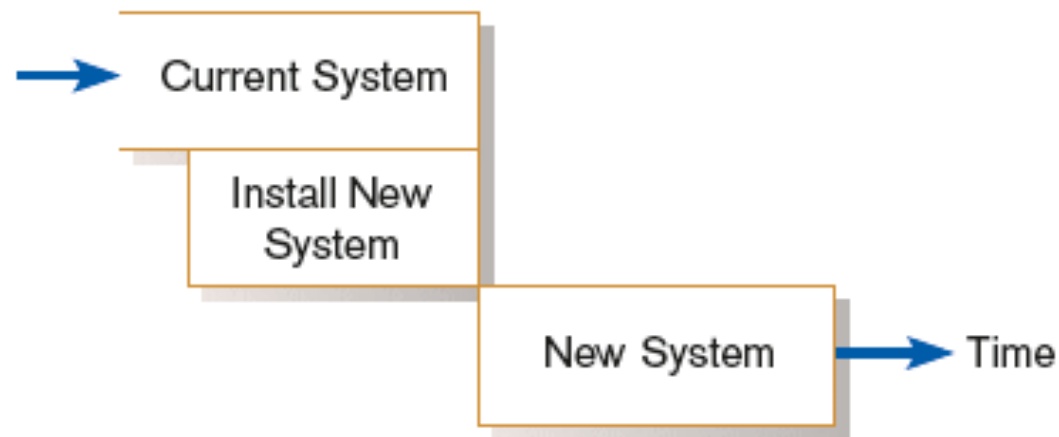
Installation

- **Installation**: the organizational process of changing over from the current information system to a new one

- Four installation strategies:
  - ☐ Direct Installation
  - ☐ Parallel Installation
  - ☐ Single-location installation
  - ☐ Phased Installation

© 2014 Pearson Education, Inc. Publishing as Prentice Hall

Direct Installation

- **Direct installation**: changing over from the old system to a new one by turning off the old system when the new system is turned on

© 2014 Pearson Education, Inc. Publishing as Prentice Hall

Parallel Installation

- **Parallel installation**: running the old information system and the new one at the same time until management decides the old system can be turned off
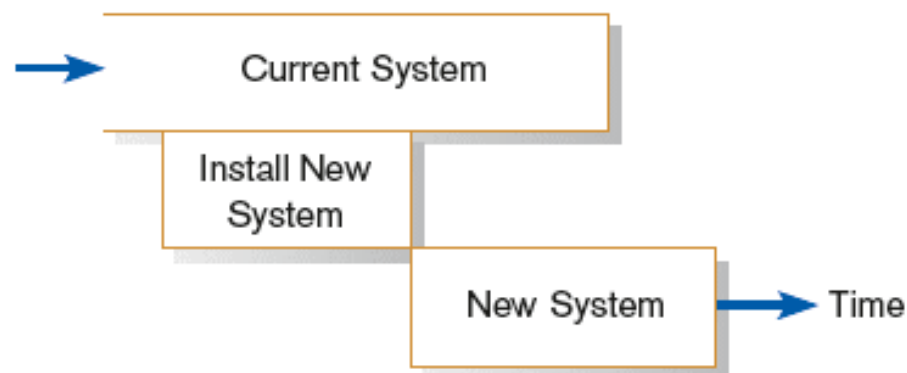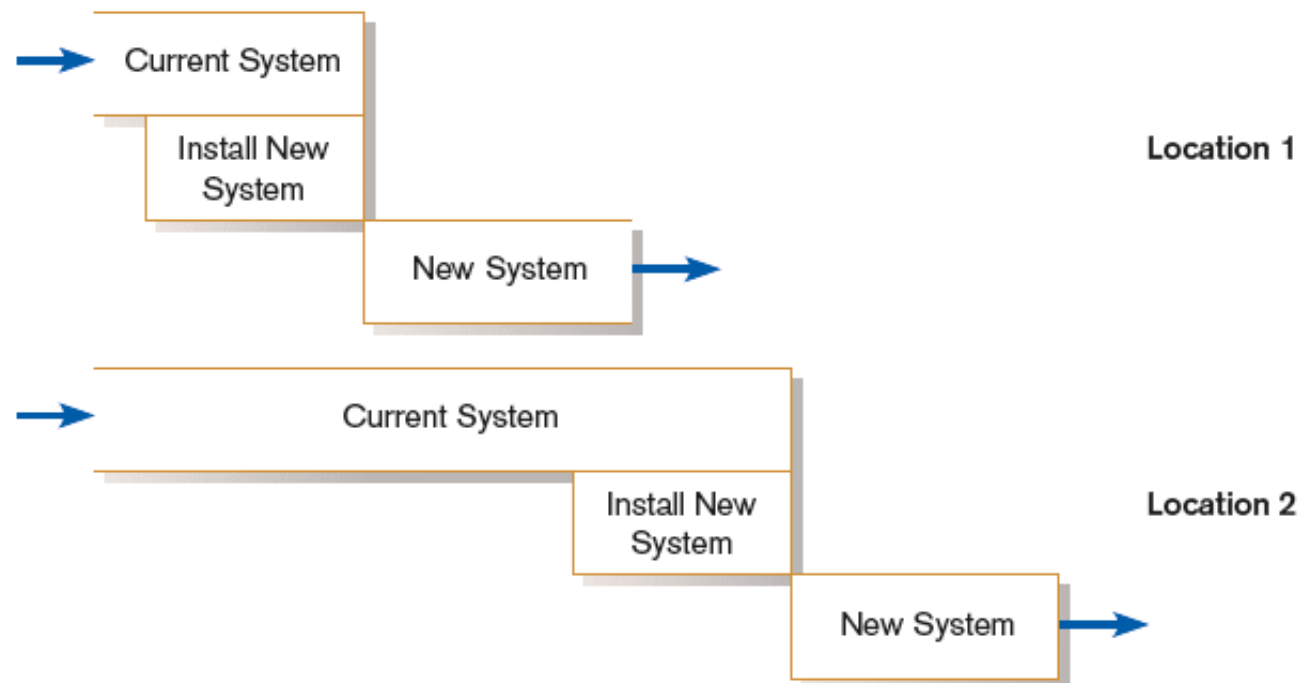
Single-Location Installation

- **Single-location installation**: trying out an information system at one site and using the experience to decide if and how the new system should be deployed throughout the organization

- Also known as location or pilot installation
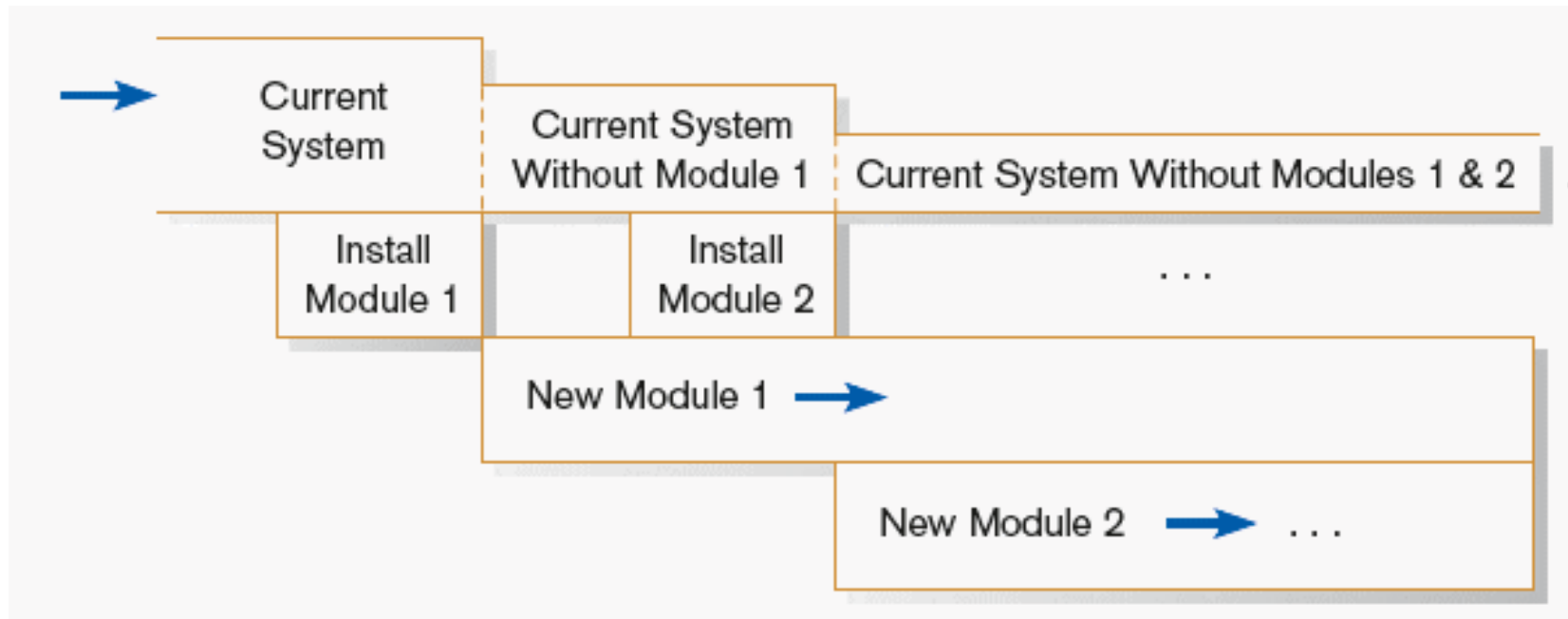
Single-Location Installation (cont.)

Phased Installation

- **Phased Installation**: changing from the old information system to the new one incrementally, starting with one or a few functional components and then gradually extending the installation to cover the whole new system

© 2014 Pearson Education, Inc. Publishing as Prentice Hall

Phased Installation (cont.)

© 2014 Pearson Education, Inc. Publishing as Prentice Hall

Planning Installation

- **Considerations**
  - ☐ Data conversion
    - ■ Error correction
    - ■ Loading from current system
  - ☐ Planned system shutdown
  - ☐ Business cycle of organization

Documenting the System

- **System documentation**: detailed information about a system's design specifications, its internal workings, and its functionality

- **User documentation**: written or other visual information about an application system, how it works, and how to use it

Documenting the System (Cont.)

- **Internal documentation**: system documentation that is part of the program source code or is generated at compile time

- **External documentation**: system documentation that includes the outcome of structured diagramming techniques such as data flow and E-R diagrams
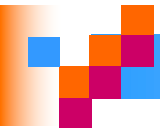
TABLE 13-5   SDLC and Generic Documentation Corresponding to Each Phase

Generic Life-Cycle Phase	Generic Document
Requirements Specification	System Requirements Specification
	Resource Requirements Specification
Project Control Structuring	Management Plan
	Engineering Change Proposal
System Development	
Architectural design	Architecture Design Document
Prototype design	Prototype Design Document
Detailed design and implementation	Detailed Design Document
Test specification	Test Specifications
Test implementation	Test Reports
System Delivery	User's Guide
	Release Description
	System Administrator's Guide
	Reference Guide
	Acceptance Sign-Off

(*Source:* Adapted from Bell and Evans, 1989.)

Figure 13-7
Example of online user documentation
(*Source:* Microsoft Corporation.)

Generic User's Guide Outline

Preface
1. Introduction
    1.1. Configurations
    1.2 Function flow
2. User interface
    2.1 Display screens
    2.2 Command types
3. Getting started
    3.1 Login
    3.2 Logout
    3.3 Save
    3.4 Error recovery
    3.n [Basic procedure name]

n. [Task name]
Appendix A—Error Messages
    ([Appendix])
Glossary
    Terms
    Acronyms
Index

Source: Adapted from Bell and Evans, 1989.)

Training and Supporting Users

- **Support**: providing ongoing educational and problem-solving assistance to information system users

- For in-house developed systems, support materials and jobs will have to be prepared or designed as part of the implementation process.
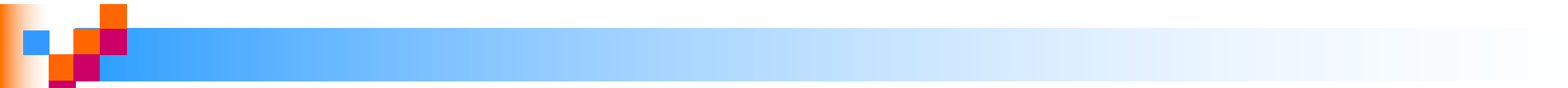
Training Information Systems Users

- **Potential training topics**
  - Use of the system
  - General computer concepts
  - Information system concepts
  - Organizational concepts
  - System management
  - System installation

Types of Training Methods

- Resident expert

- Traditional instructor-led classroom training

- E-learning, distance learning

- Blended learning (instructor plus e-learning)

- Software help components

- External sources (e.g. vendors)

Training Information Systems Users (Cont.)

- **Electronic performance support system (EPSS)**: component of a software package or an application in which training and educational information is embedded

- An EPSS can take several forms, including a tutorial, an expert system shell, and hypertext jumps to reference materials.

Supporting Information Systems Users

- Support is important to users, but has often been inadequate.

- Providing support can be expensive and time-consuming.

- Vendors usually charge for their support, using 900- numbers, or charge a fee for unlimited or monthly support.

Automating Support

- One approach is through automation.
  - ☐ Internet-based online support forums
  - ☐ On-demand fax
  - ☐ Voice response systems
  - ☐ Knowledge bases

© 2014 Pearson Education, Inc. Publishing as Prentice Hall

Providing Support Through a Help Desk

- **Help desk**: a single point of contact for all user inquiries and problems about a particular information system or for all users in a particular department

Providing Support Through a Help Desk (Cont.)

- Requires
  - □ *Technical skills*: extensive knowledge about how to use the system and typical problems that can be encountered
  - □ *People skills*: good listening and communication, dealing with complaints and frustrations

Support Issues for the Analyst to Consider

- User questions and problems

- Recovery and backup

- Disaster recovery

- PC maintenance

- Writing newsletters

- Setting up user groups

Organizational Issues in Systems Implementation

- Why does implementation sometimes fail?
- Traditional wisdom of primary success factors:
  - Management support
  - User involvement
  - But these are not enough
- Other important factors
  - Commitment to project
  - Commitment to change
  - Extent of project definition and planning

Factors Influencing System Use

- Personal stake of users

- System characteristics

- User demographics

- Organizational support
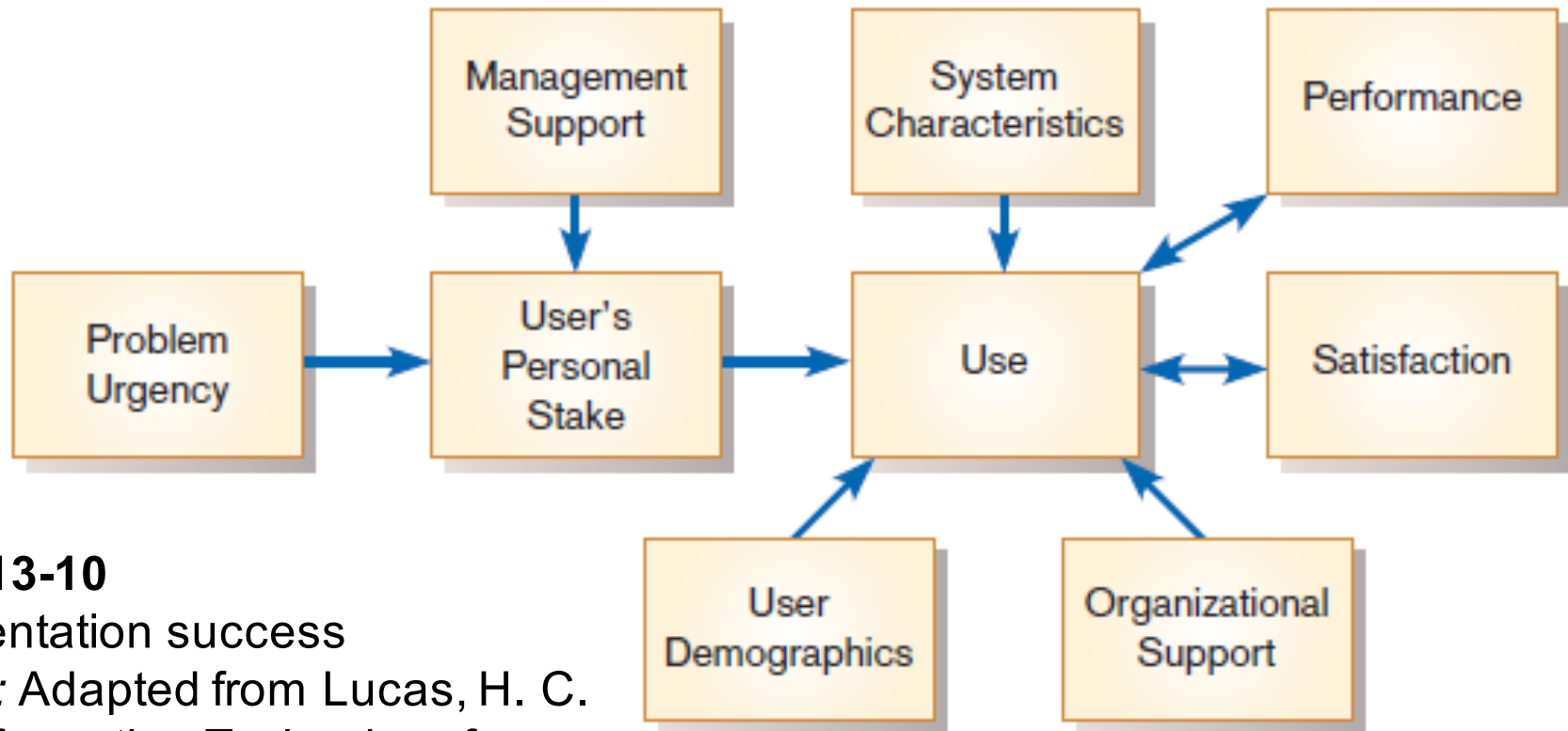
- Performance

- Satisfaction

Success Factors

Figure 13-10
Implementation success
(*Source:* Adapted from Lucas, H. C. 1997. *Information Technology for Management.* New York: McGraw-Hill, with the permission of the McGraw-Hill Companies. All rights reserved.)
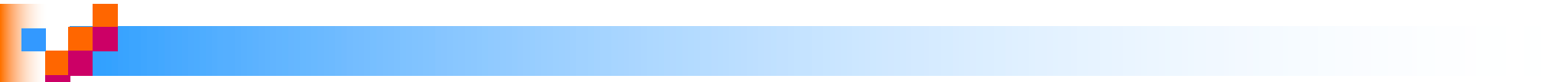
Security Issues

- Increasingly important issue for organizations and their management
- **Mal**icious soft**ware** (*malware*): includes Trojan horses, worms, viruses, and other kinds
- External sources of threats include laptop theft, system penetration, and denial of service.

Table 13-8 Selected Statistics on IT Security

Nature of Breaches	
Malware infection	67%
Being fraudulently represented as sender of phishing messages	39%
Laptop/mobile theft	34%
Bots/zombies within the organization	29%
Insider abuse of internet access or e-mail	25%
Denial of service	17%
Unauthorized access or privilege escalation by insider	13%

Security Technologies Used	
Antivirus software	97%
Firewalls	95%
Anti-spyware software	85%
Virtual private network	79%
Vulnerability/Patch management	68%
Encryption of data in transit	66%
Intrusion detection	62%

Proportion of IT Budget Devoted to Security	
Proportion spending more than 10%	19%
Proportion spending between 2% and 10%	40%
Proportion spending between 1% and 2%	16%
Proportion spending less than 1%	10%

(*Source:* Data from Computer Security Institute [Richardson], 2011.)

Electronic Commerce Application: System Implementation for Pine Valley Furniture's WebStore

- Developing test cases for the WebStore include testing categories as follows:
  - Simple functionality
  - Multiple functionality
  - Function chains
  - Elective functions
  - Emergency/crisis

© 2014 Pearson Education, Inc. Publishing as Prentice Hall

Developing Test Cases for WebStore

- Test case forms had the following sections:
  - ☐ Test Case ID
  - ☐ Category/Objective of Test
  - ☐ Description
  - ☐ System Version
  - ☐ Completion Date
  - ☐ Participants
  - ☐ Machine Characteristics (processor, operating system, memory, browser, etc.)
  - ☐ Test Result
  - ☐ Comments

Bug Tracking and System Evolution

- Bug-tracking form has the following categories:
  - ☐ Bug Number (simple incremental number)
  - ☐ Test Case ID that Generated the Bug
  - ☐ Is the Bug Replicable?
  - ☐ Effects
  - ☐ Description
  - ☐ Resolution
  - ☐ Resolution Date
  - ☐ Comments
- As batches of bugs are fixed, the version number of the software is incremented (below 1.0 during development and testing).

Alpha and Beta Testing the WebStore

■ Alpha Testing:

  □ PVF employees who actively participated received a t-shirt and $100 to shop.

  □ Development team conducted extensive recovery, security, stress, and performance testing.

■ Beta Testing

  □ PVF recruited several of their established customers to help in beta testing.

© 2014 Pearson Education, Inc. Publishing as Prentice Hall

WebStore Installation

- WebStore was ready to go online and development team recommended to top management that it was time to "flip the switch".

© 2014 Pearson Education, Inc. Publishing as Prentice Hall

Project Close-Down

- Evaluate team.
  - □ Reassign members to other projects.
- Notify all affected parties that the development project is ending and that you are switching to operation and maintenance mode.
- Conduct post project reviews.
- Close out customer contract.
  - □ Formal signoff

Summary

- In this chapter you learned how to:

  ✓ Describe the process of coding, testing, and converting an organizational information system and outline the deliverables and outcomes of the process.

  ✓ Prepare a test plan for an information system.

  ✓ Apply four installation strategies: direct, parallel, single-location, and phased installation.

  ✓ List the deliverables for documenting the system and for training and supporting users.

Summary (Cont.)

✓ Compare the many modes available for organizational information system training, including self-training and electronic performance support systems.

✓ Discuss the issues of providing support for end-users.

✓ Explain why system implementation sometimes fails.

✓ Describe the threats to system security and remedies that can be applied.

✓ Show how traditional implementation issues apply to electronic commerce applications.