Systems Analysis and Design

Chapter 8

System Implementation & Maintenance

The slide is mainly adopted from:

- J. S. Valacich, J. George, Modern Systems Analysis and Design. 8th Edition, Pearson 2017.
- I. Sommerville. Software Engineering. 10th Edition, Pearson, 2016.

Last update: Dec. 15, 2018

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
- Discuss the issues of providing support for end-users.
- Explain why system implementation sometimes fails.
- Describe maintenance management issues, including alternative organizational structures, quality measurement, processes for handling change requests, and configuration management.

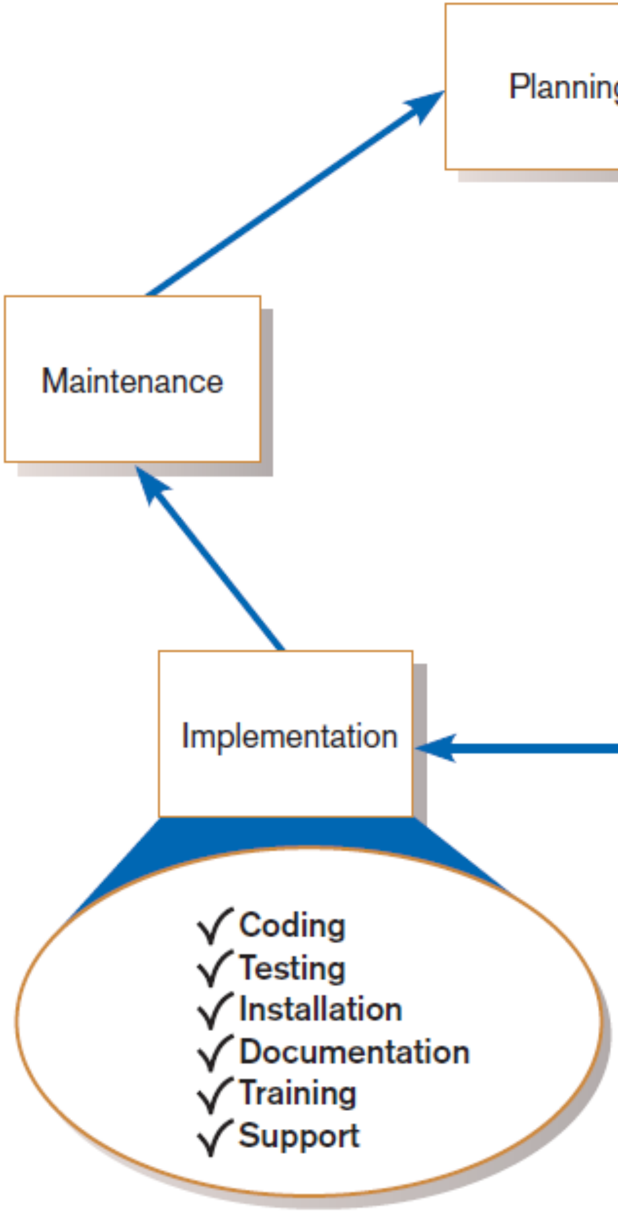


FIGURE 13-1
Systems development life cycle with the implementation phase highlighted

System Implementation

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

- 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. (e.g., testing is performed in parallel with coding)
- **Installation**
  - The current system is replaced by a new system.

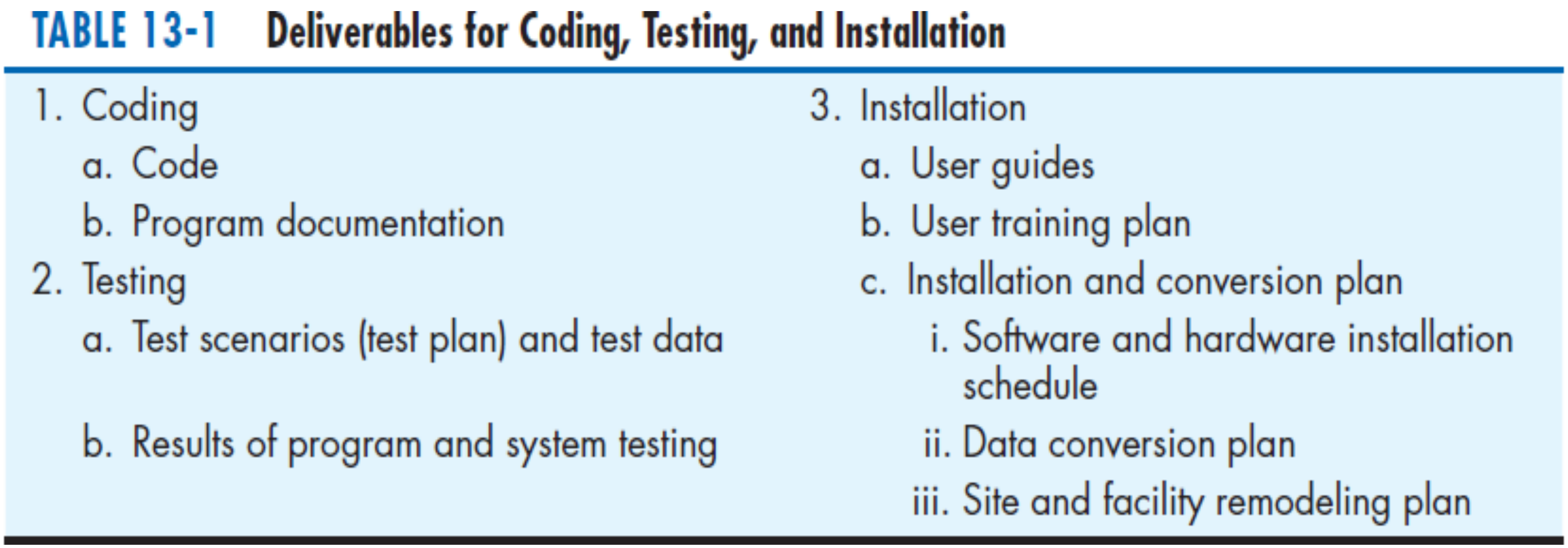


TABLE 13-1 Deliverables for Coding, Testing, and Installation

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

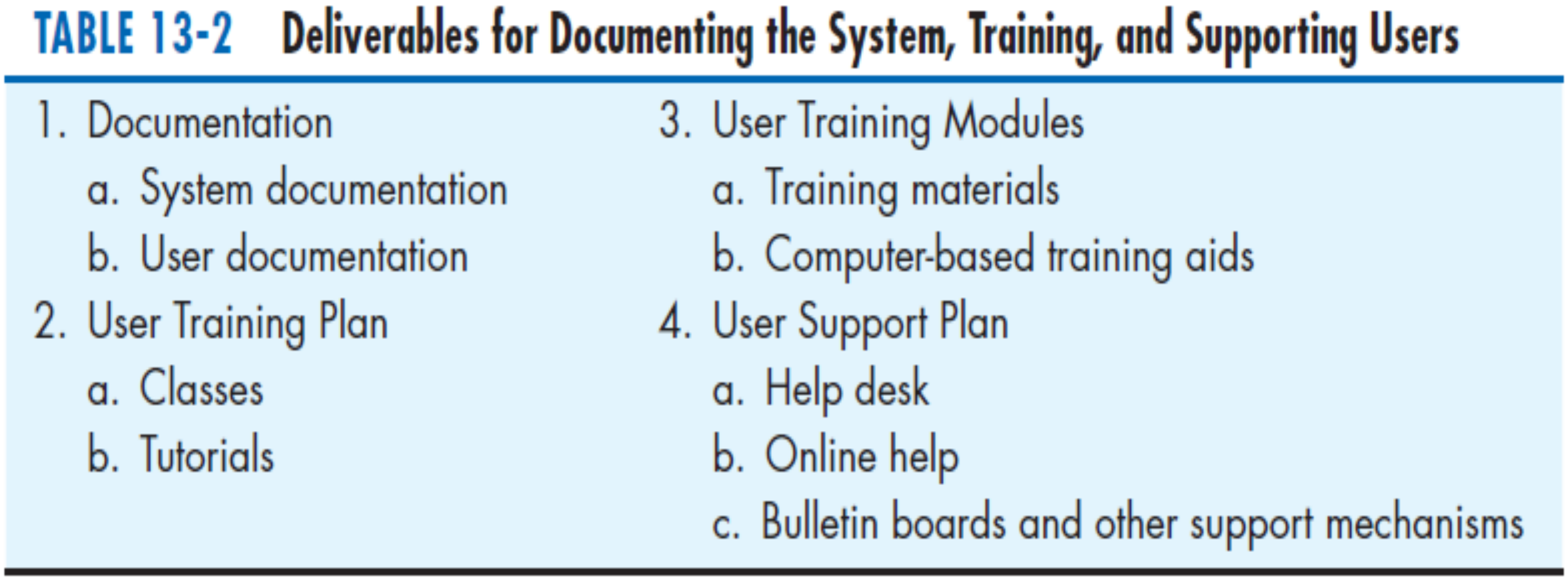


TABLE 13-2 Deliverables for Documenting the System, Training, and Supporting Users

1. Documentation	3. User Training Modules
a. System documentation	a. Training materials
b. User documentation	b. Computer-based training aids
2. User Training Plan	4. User Support Plan
a. Classes	a. Help desk
b. Tutorials	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.

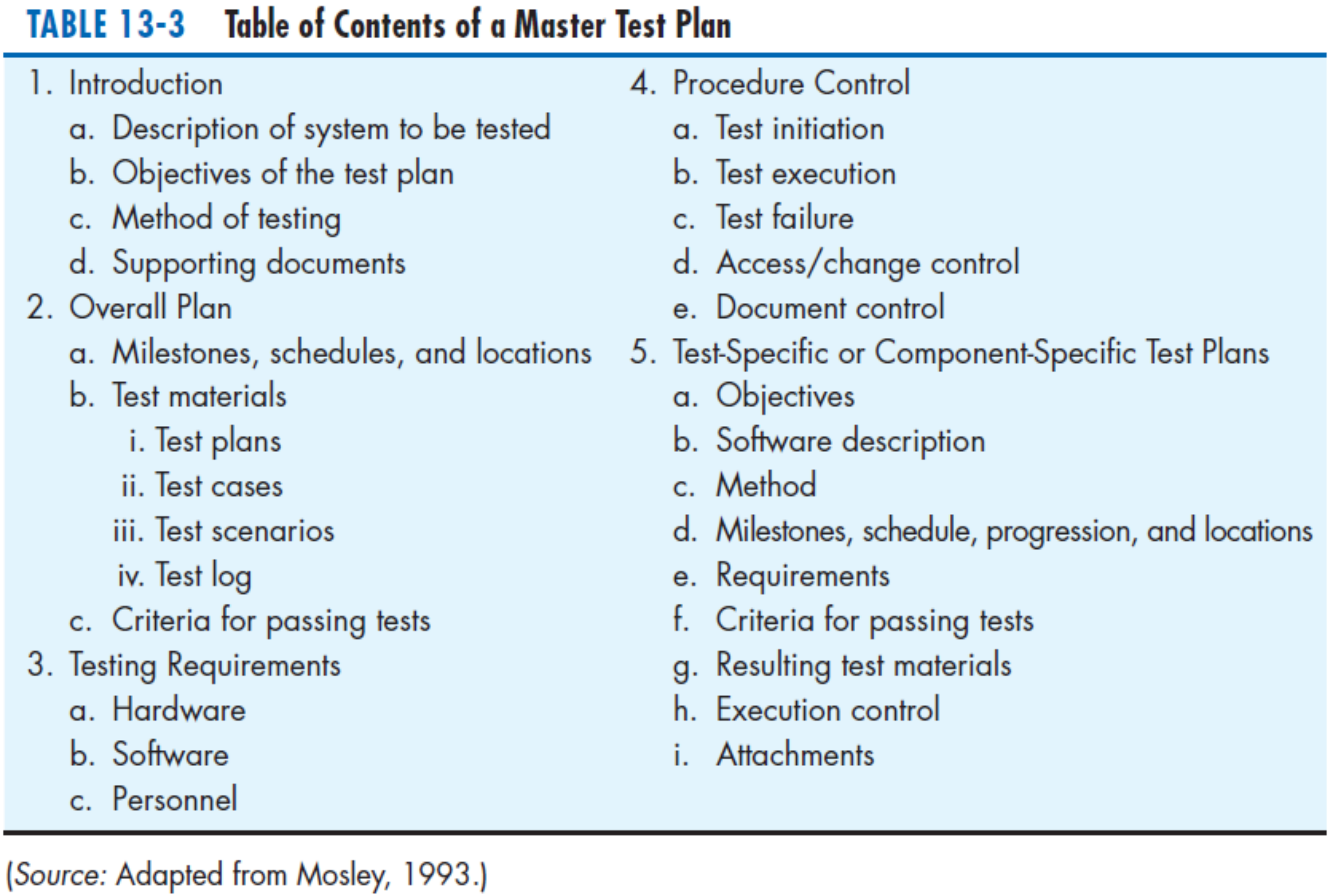


TABLE 13-3 Table of Contents of a Master Test Plan

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. (**Verification**)
  - **Dynamic** testing involves execution of the code. (**Validation**)
- Test is automated or manual
  - Automated means computer conducts the test.
  - Manual means that people complete the test.

Seven Different Types of Tests

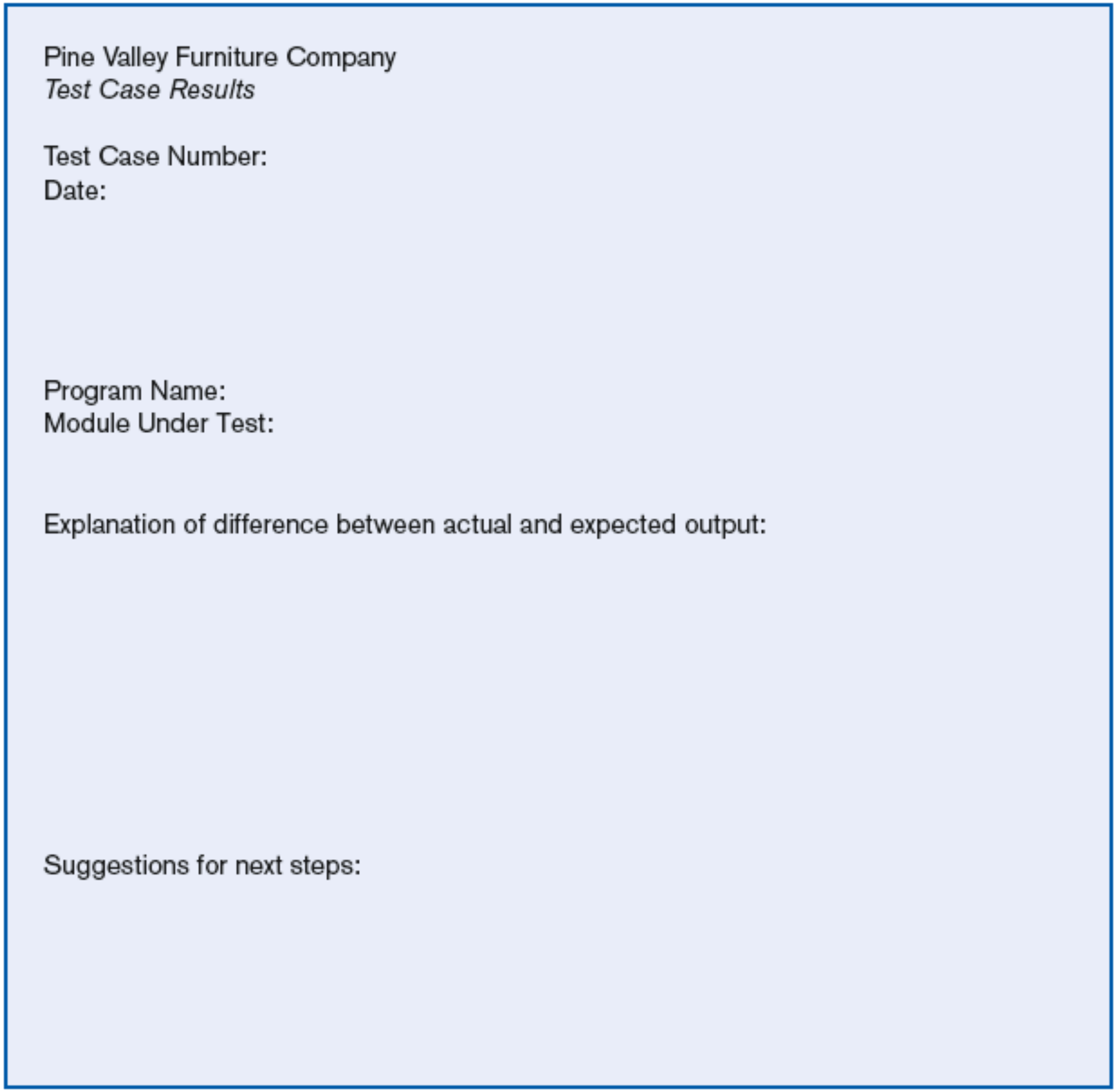
- **Inspection**: a testing technique in which participants examine program code for predictable language-specific errors
- **Walkthrough**: a peer group review of any product created during the systems development process, including code
- **Desk checking**: a testing technique in which the program code is sequentially executed manually by the reviewer
- **Unit testing**: each module is tested alone in an attempt to discover any errors in its code

Seven Different Types of Tests (Cont.)

- **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.
- **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.
- **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.
- 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.



Pine Valley Furniture Company
Test Case Results

Test Case Number:
Date:

Program Name:
Module Under Test:

Explanation of difference between actual and expected output:

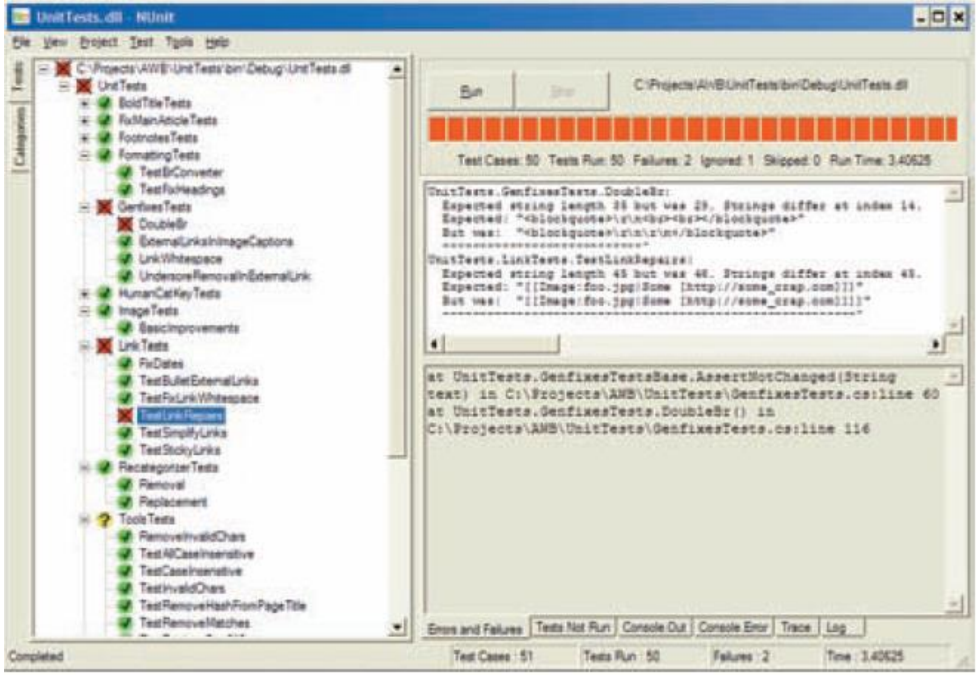
Suggestions for next steps:

FIGURE 13-4
Test case results form

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

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



```
C:\Projects\AWB\UnitTests\bin\Debug\UnitTests.dll
  UnitTests
    BoldTitleTests
    FixMainArticleTests
    FootnotesTests
    FormattingTests
      TestBrConverter
      TestFixHeadings
    GenfixesTests
      DoubleBr
      ExternalLinksInImageCaptions
      LinkWhitespace
      UnderscoreRemovalInExternalLink
    HumanCatKeyTests
    ImageTests
      BasicImprovements
    LinkTests
      FixDates
      TestBulletExternalLinks
      TestFixLinkWhitespace
      TestLinkRepairs
      TestSimplifyLinks
      TestStickyLinks
    RecategorizerTests
      Removal
      Replacement
    ToolsTests
      RemoveInvalidChars
      TestAllCaseInsensitive
      TestCaseInsensitive
      TestInvalidChars
      TestRemoveHashFromPageTitle
      TestRemoveMatches
```

C:\Projects\AWB\UnitTests\bin\Debug\UnitTests.dll

Test Cases: 50 Tests Run: 50 Failures: 2 Ignored: 1 Skipped: 0 Run Time: 3.40625

```
UnitTests.GenfixesTests.DoubleBr:
  Expected string length 35 but was 29. Strings differ at index 14.
  Expected: "<blockquote>\r\n<br><br></blockquote>"
  But was:  "<blockquote>\r\n\r\n</blockquote>"
  -------------------------^
UnitTests.LinkTests.TestLinkRepairs:
  Expected string length 45 but was 46. Strings differ at index 45.
  Expected: "[[Image:foo.jpg|Some [http://some_crap.com]]]"
  But was:  "[[Image:foo.jpg|Some [http://some_crap.com]]]]"
  ----------------------------------------------------------------^
```

```
at UnitTests.GenfixesTestsBase.AssertNotChanged(String
text) in C:\Projects\AWB\UnitTests\GenfixesTests.cs:line 60
at UnitTests.GenfixesTests.DoubleBr() in
C:\Projects\AWB\UnitTests\GenfixesTests.cs:line 116
```

Completed Test Cases : 51 Tests Run : 50 Failures : 2 Time : 3.40625

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

Combining Coding and Testing

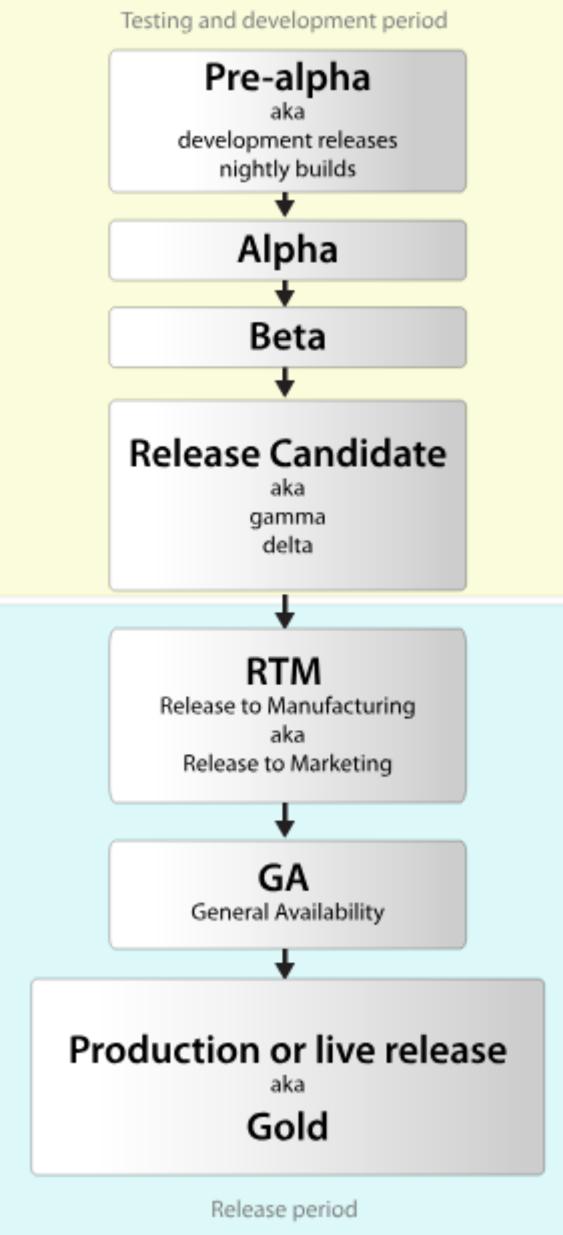
- Coding and testing often go together.
- Big companies have dedicated test staff.
- With eXtreme programming (XP) a common technique is *refactoring*.
- **Refactoring** = making a program simpler after adding a new feature

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

Software release life cycle

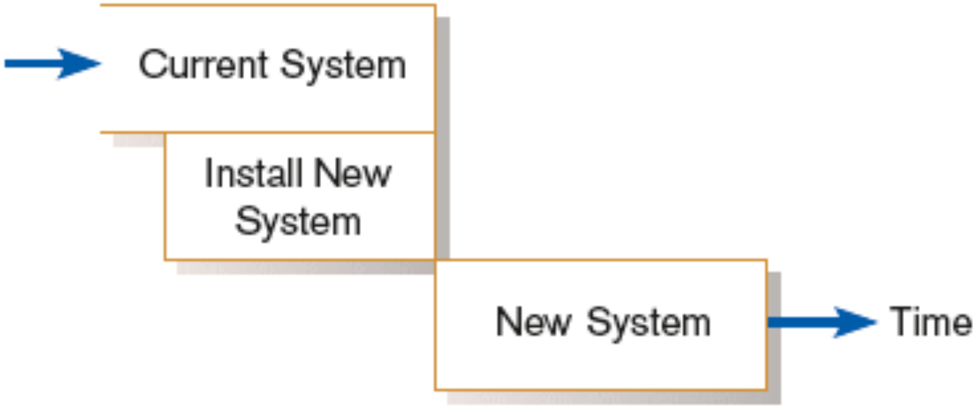
- A software release life cycle is the sum of the stages of development and maturity for a piece of computer software: ranging from its initial development to its eventual release, and including updated versions of the released version to help improve software or fix software bugs still present in the software.



Installation

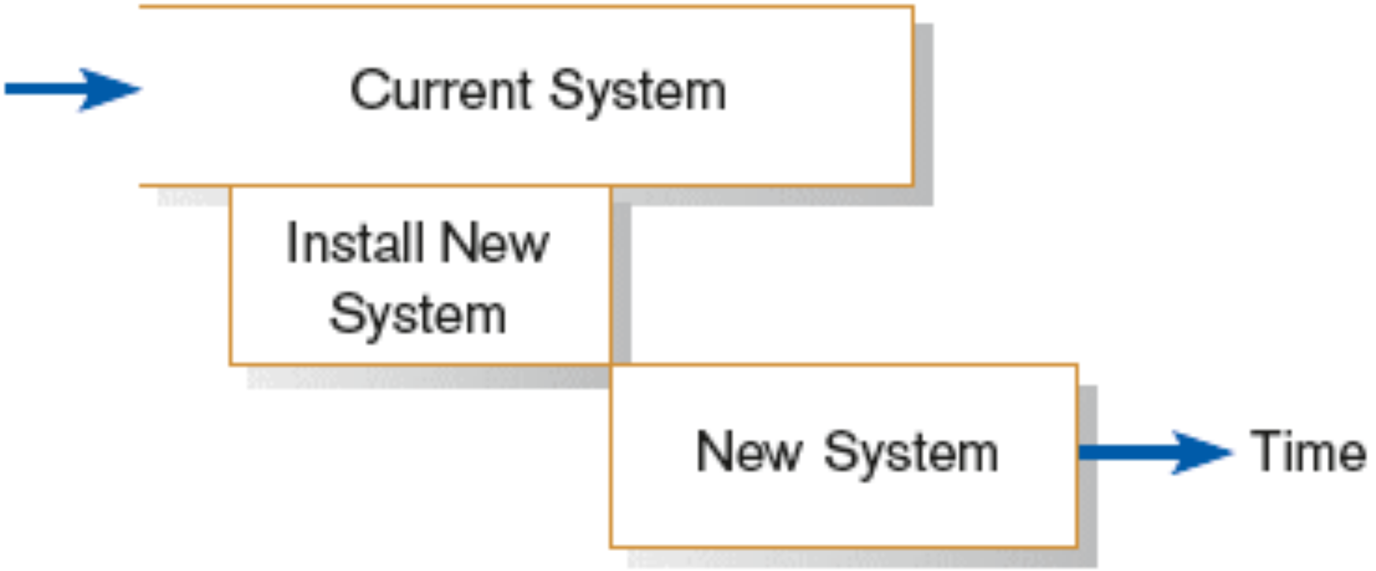
- **Installation**: the organizational process of changing over from the current information system to a new one
- Four installation strategies:
  - **Direct Installation**. changing over from the old system to a new one by turning off the old system when the new system is turned on
  - **Parallel Installation**
  - **Single-location installation**
  - **Phased Installation**

Direct Installation.



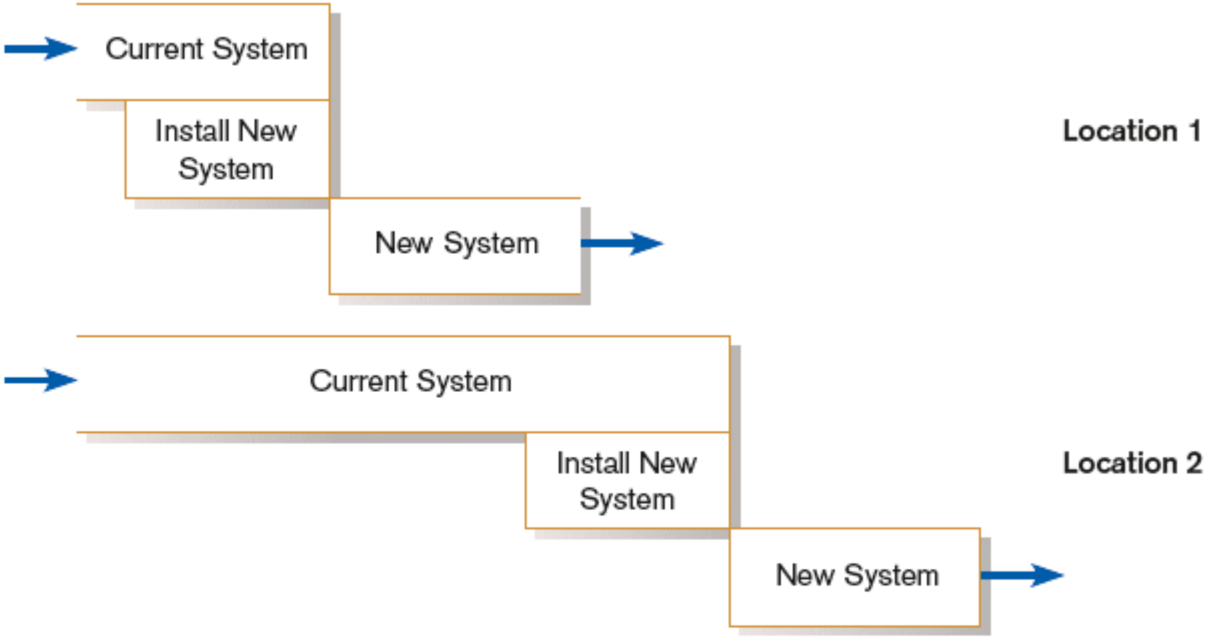
Parallel Installation

- **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



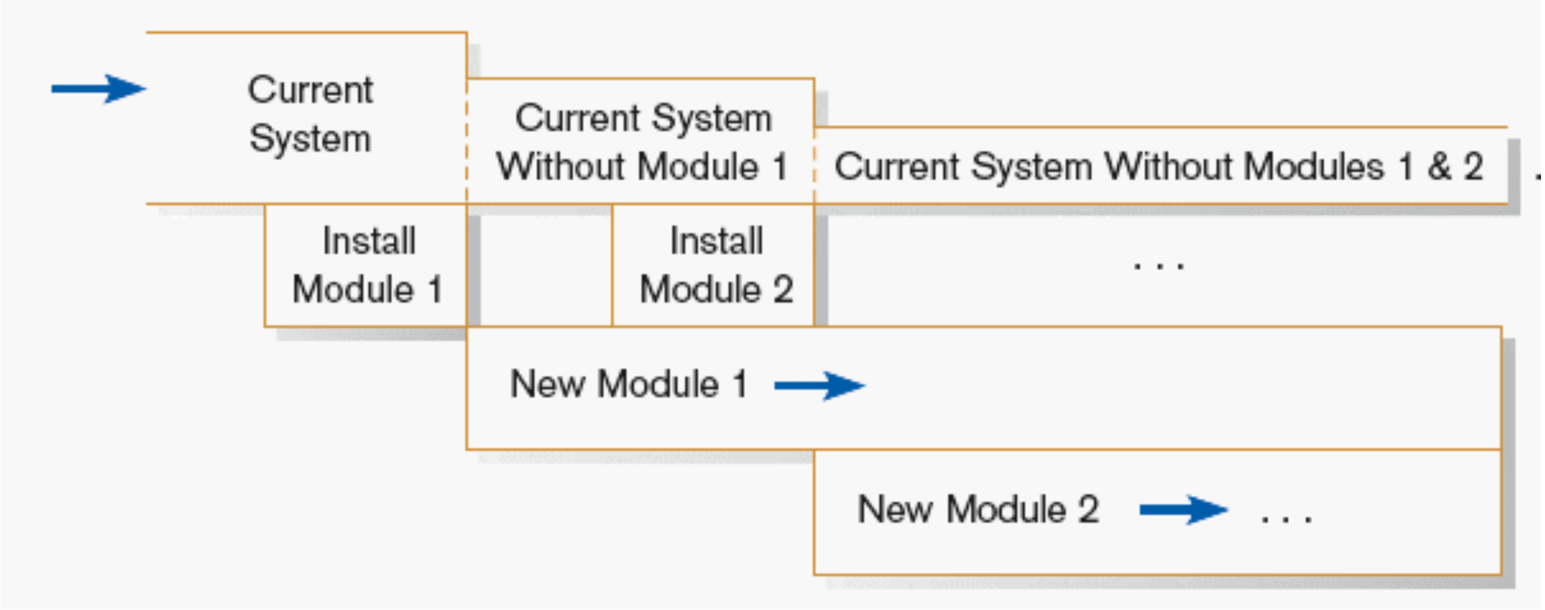
Single-Location Installation

- **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**



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



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
- **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

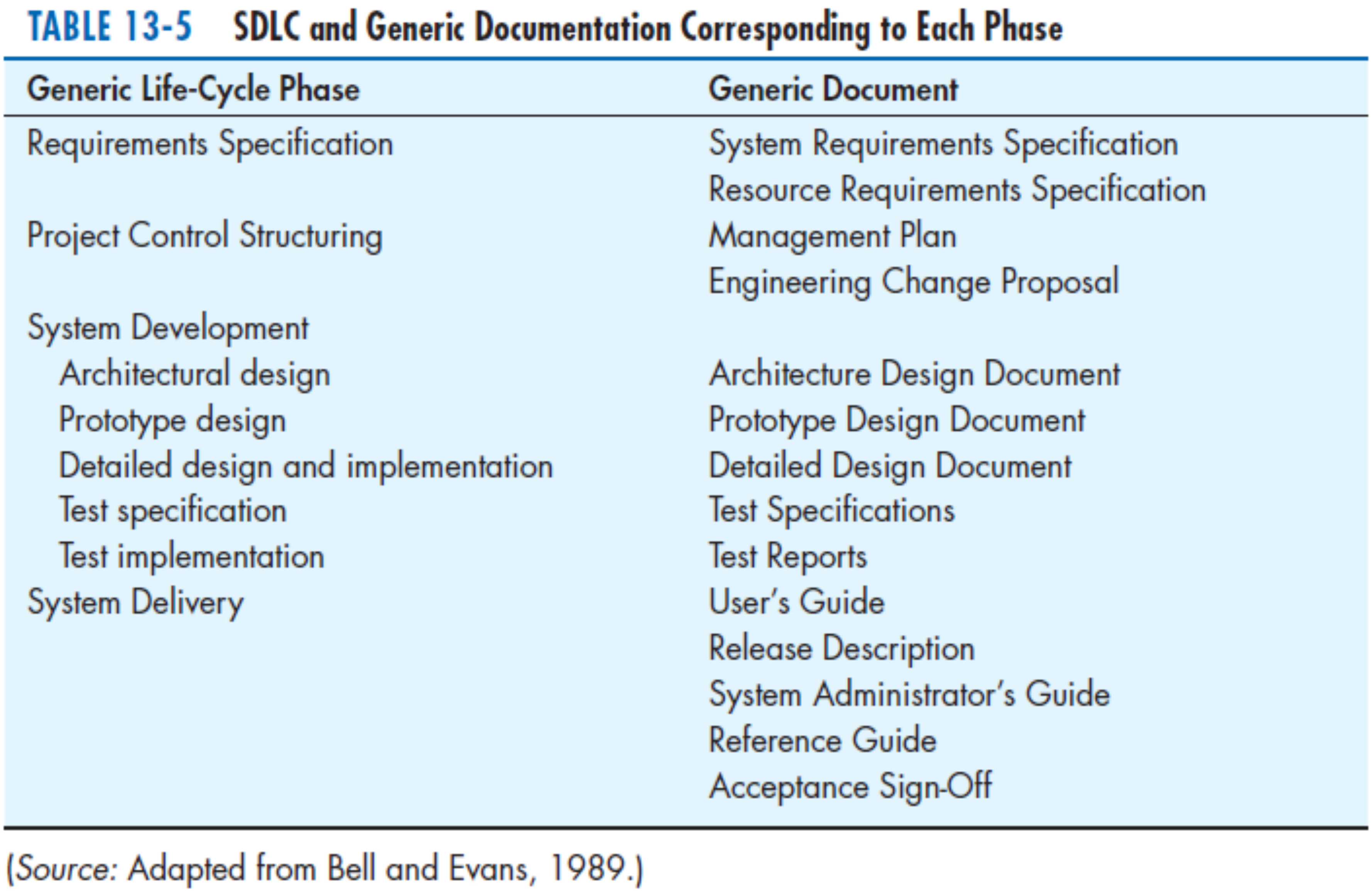
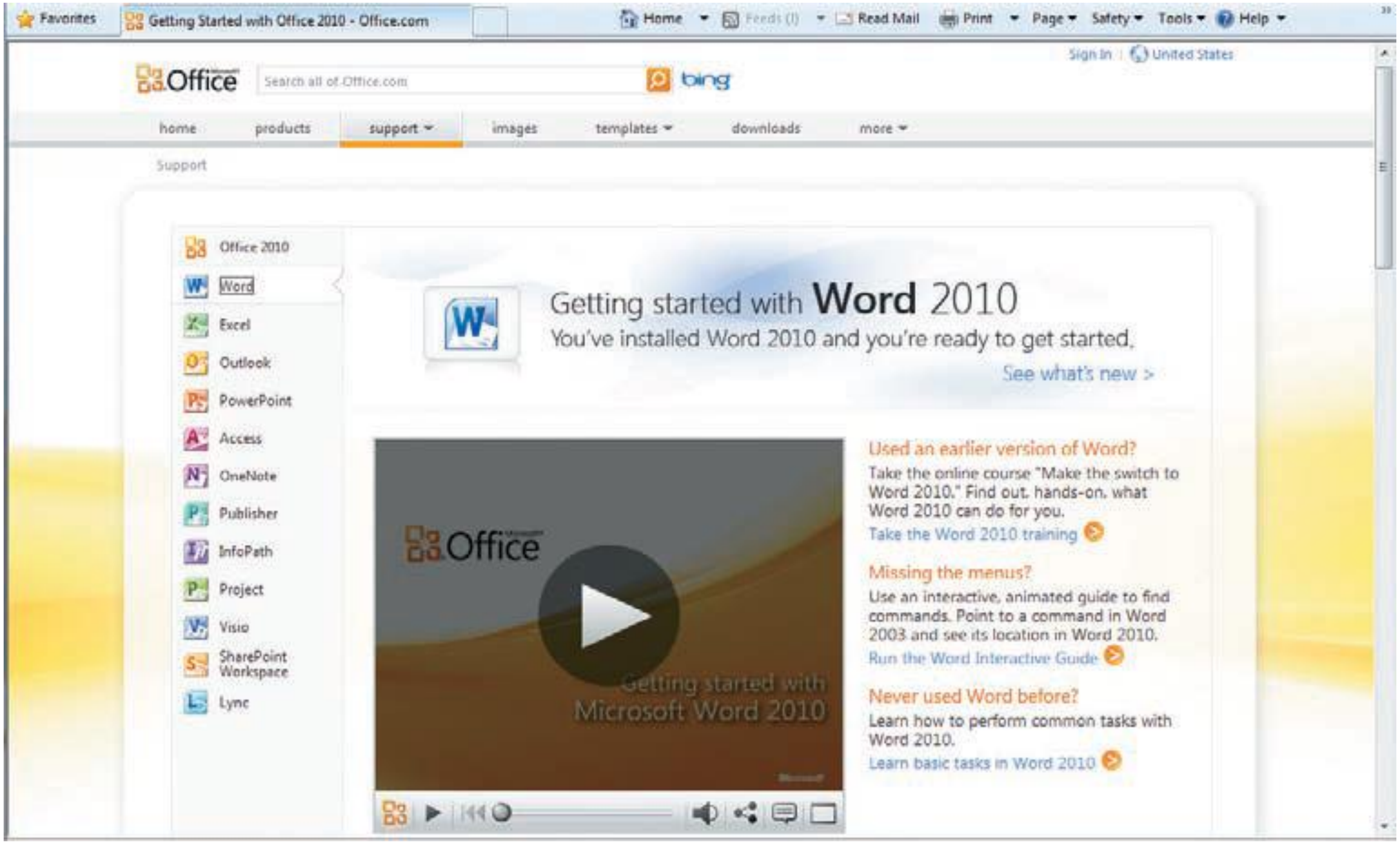


TABLE 13-5 SDLC and Generic Documentation Corresponding to Each Phase

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.)



Getting started with **Word** 2010

You've installed Word 2010 and you're ready to get started.

See what's new >

Used an earlier version of Word?

Take the online course "Make the switch to Word 2010." Find out, hands-on, what Word 2010 can do for you.

Take the Word 2010 training

Missing the menus?

Use an interactive, animated guide to find commands. Point to a command in Word 2003 and see its location in Word 2010.

Run the Word Interactive Guide

Never used Word before?

Learn how to perform common tasks with Word 2010.

Learn basic tasks in Word 2010

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)

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
  - Internet-based online support forums
  - On-demand fax
  - Voice response systems
  - Knowledge bases

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
- 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

Success Factors System Use

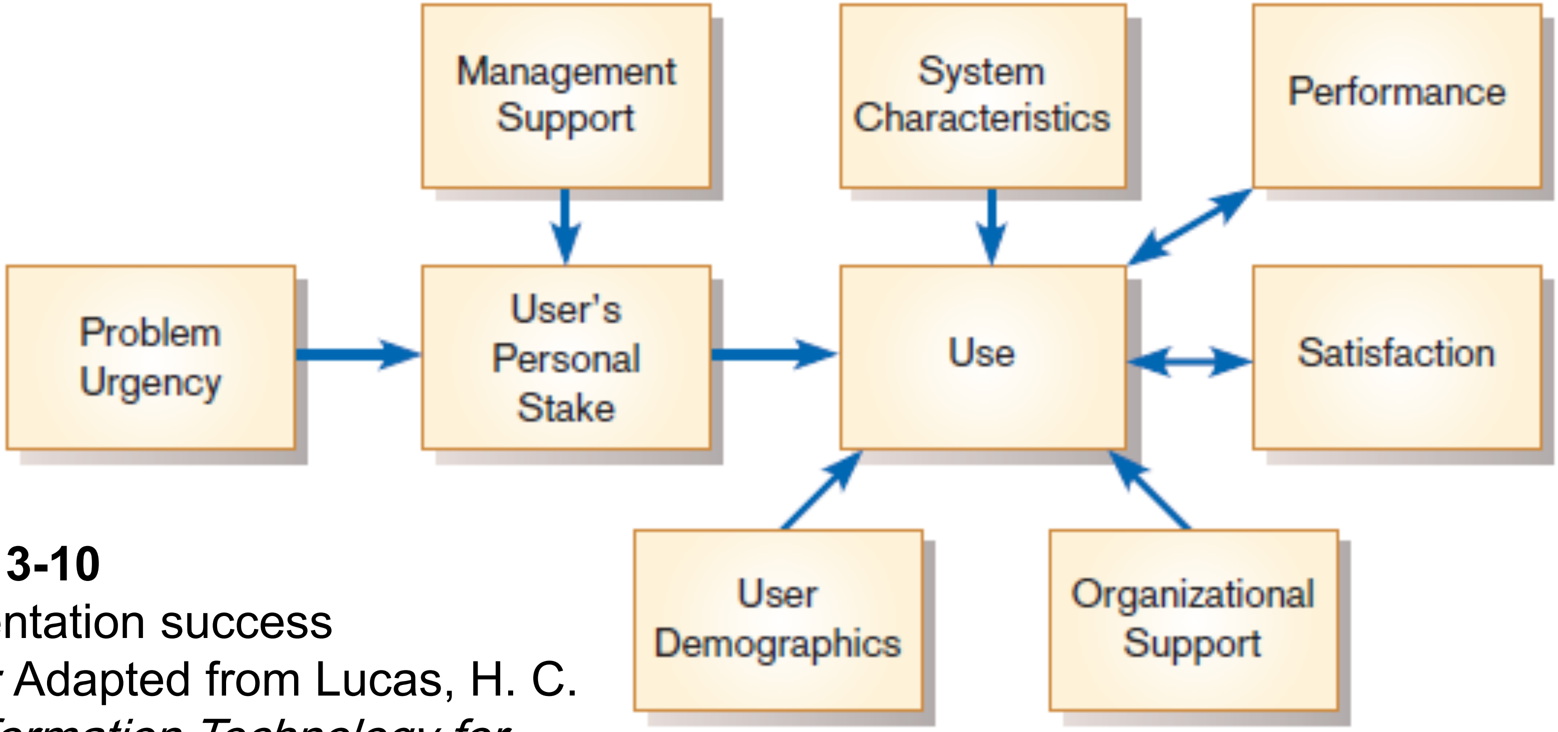
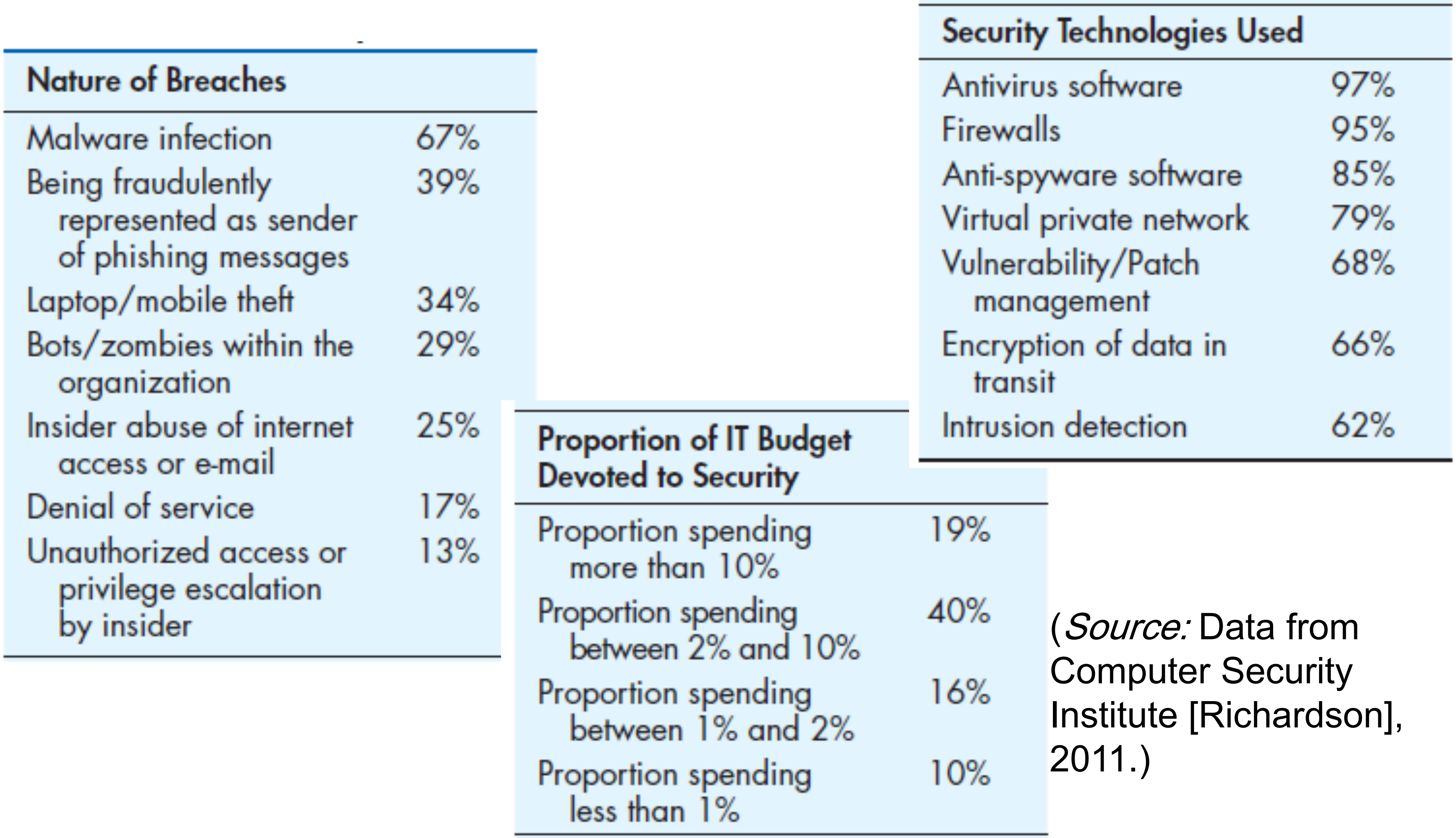


Figure 13-10
Implementation success
(*Source:* Adapted from Lucas, H. C. 1997. *Information Technology for Management*. New York: McGraw-Hill, with the permission of the McGraw-Hill Companies. All rights reserved.)

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.)

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.

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

Maintaining Information Systems

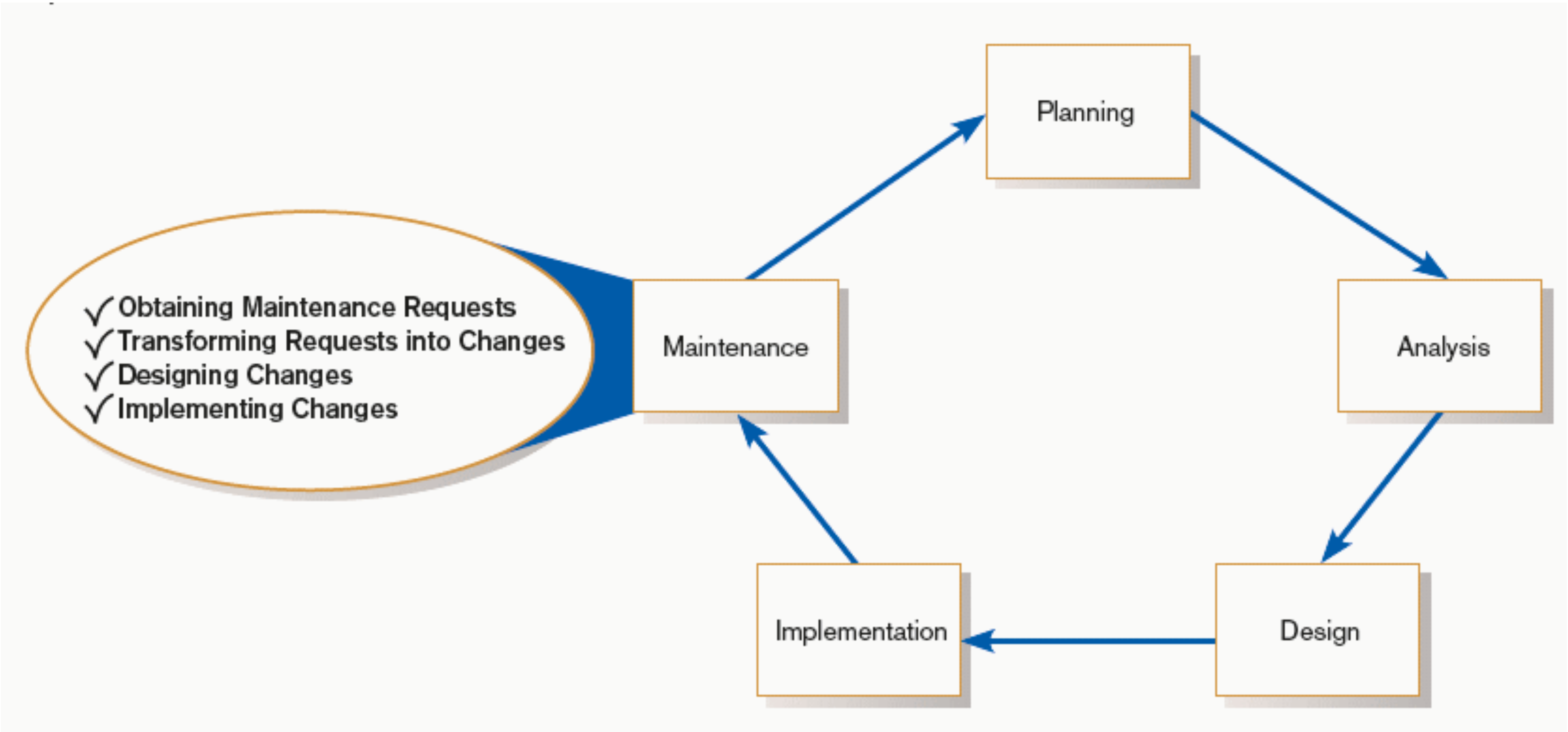
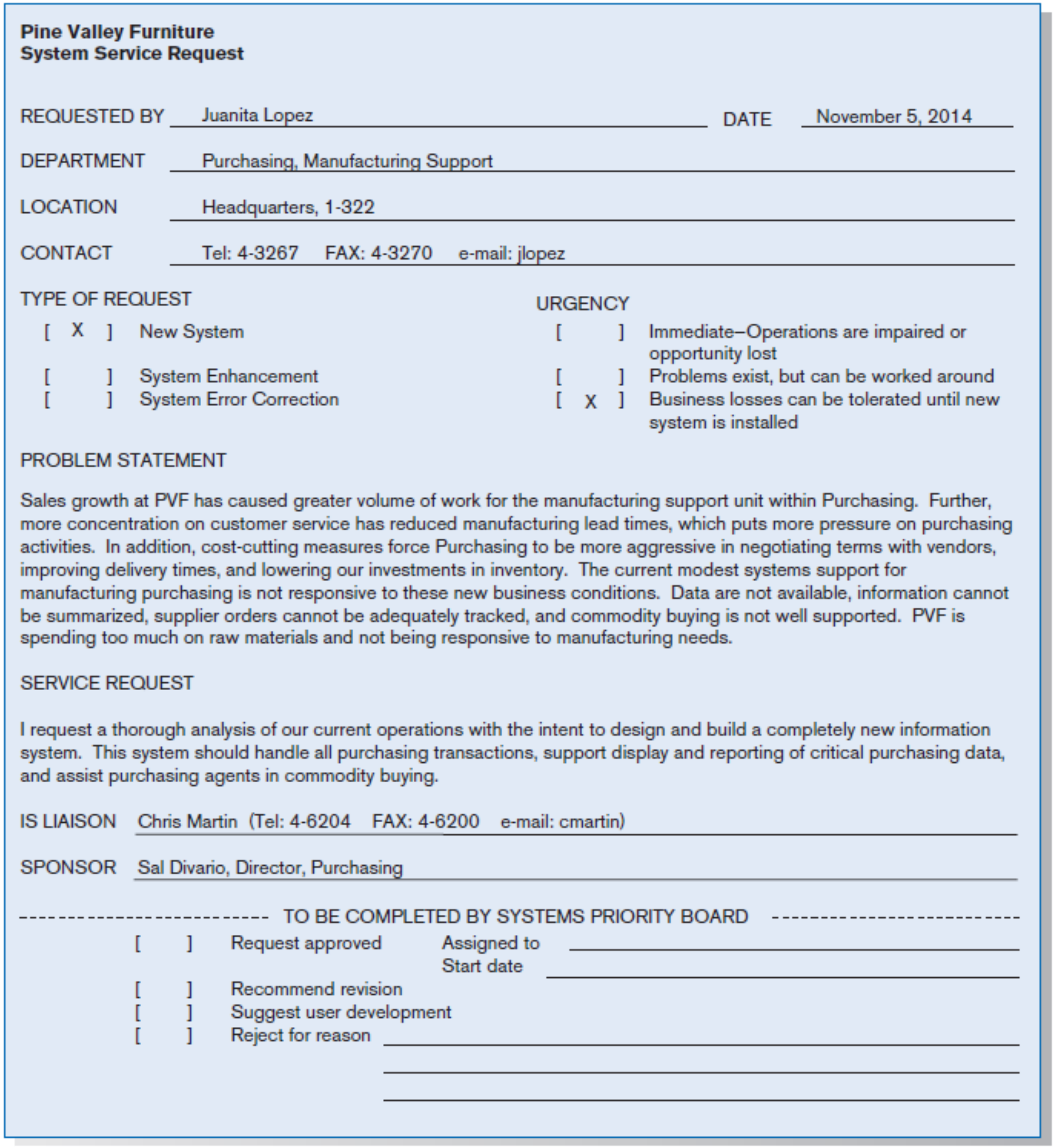


FIGURE 14-1
Systems development life cycle

The Process of Maintaining Information Systems

- Process of returning to the beginning of the SDLC and repeating development steps focusing on system change until the change is implemented
- Maintenance is the longest phase in the SDLC.

- Four major activities:
  - Obtaining maintenance requests
  - Transforming requests into changes
  - Designing changes
  - Implementing changes



Pine Valley Furniture
System Service Request

REQUESTED BY Juanita Lopez DATE November 5, 2014

DEPARTMENT Purchasing, Manufacturing Support

LOCATION Headquarters, 1-322

CONTACT Tel: 4-3267 FAX: 4-3270 e-mail: jlopez

TYPE OF REQUEST

[X] New System

[ ] System Enhancement

[ ] System Error Correction

URGENCY

[ ] Immediate–Operations are impaired or opportunity lost

[ ] Problems exist, but can be worked around

[X] Business losses can be tolerated until new system is installed

PROBLEM STATEMENT

Sales growth at PVF has caused greater volume of work for the manufacturing support unit within Purchasing. Further, more concentration on customer service has reduced manufacturing lead times, which puts more pressure on purchasing activities. In addition, cost-cutting measures force Purchasing to be more aggressive in negotiating terms with vendors, improving delivery times, and lowering our investments in inventory. The current modest systems support for manufacturing purchasing is not responsive to these new business conditions. Data are not available, information cannot be summarized, supplier orders cannot be adequately tracked, and commodity buying is not well supported. PVF is spending too much on raw materials and not being responsive to manufacturing needs.

SERVICE REQUEST

I request a thorough analysis of our current operations with the intent to design and build a completely new information system. This system should handle all purchasing transactions, support display and reporting of critical purchasing data, and assist purchasing agents in commodity buying.

IS LIAISON Chris Martin (Tel: 4-6204 FAX: 4-6200 e-mail: cmartin)

SPONSOR Sal Divario, Director, Purchasing

-------------------------- TO BE COMPLETED BY SYSTEMS PRIORITY BOARD --------------------------

[ ] Request approved Assigned to ______
Start date ______

[ ] Recommend revision

[ ] Suggest user development

[ ] Reject for reason ______

FIGURE 14-2
System Service Request for purchasing fulfillment system (Pine Valley Furniture)

Deliverables and Outcome

- The maintenance phase of the SDLC is basically a subset of the activities of the entire development process.
- The deliverables and outcomes from the process are the development of a new version of the software and new versions of all design documents created or modified during the maintenance effort.

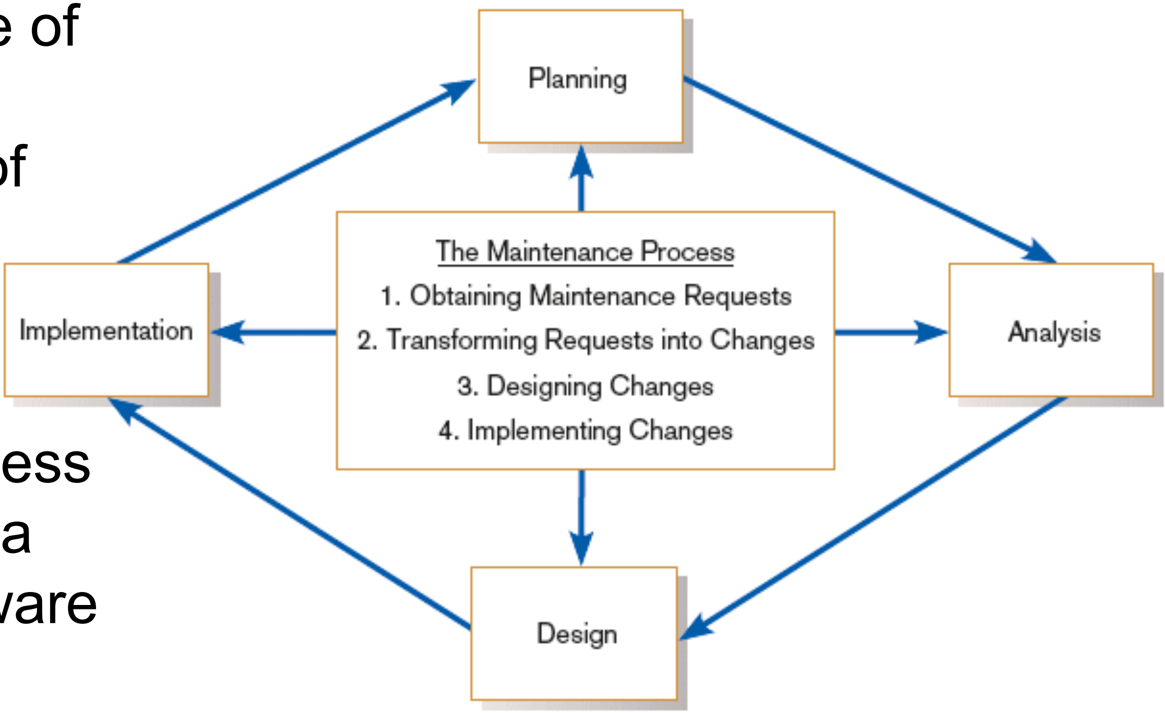


FIGURE 14-3
Maintenance activities parallel those of the SDLC

Types of System Maintenance

- **Maintenance**: changes made to a system to fix or enhance its functionality.
- **Corrective maintenance**: changes made to a system to repair flaws in its design, coding, or implementation.
- **Adaptive maintenance**: changes made to a system to evolve its functionality to changing business needs or technologies
- **Perfective maintenance**: changes made to a system to add new features or to improve performance
- **Preventive maintenance**: changes made to a system to avoid possible future problems

Types of System Maintenance (Cont.)

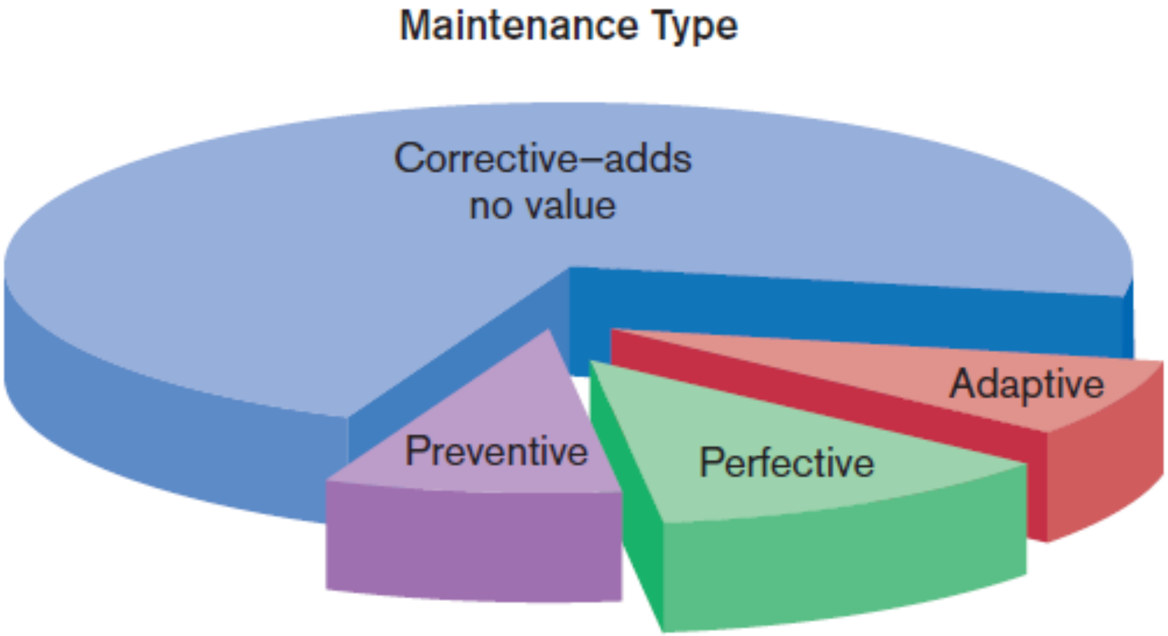


Figure 14-4
Value and non-value adding of different types of maintenance (*Sources:* Based on Andrews and Leventhal,1993; Pressman, 2005.)

The Cost of Maintenance

- Many organizations allocate 60-80% of information systems budget to maintenance.
- **Maintainability**: the ease with which software can be understood, corrected, adapted, and enhanced

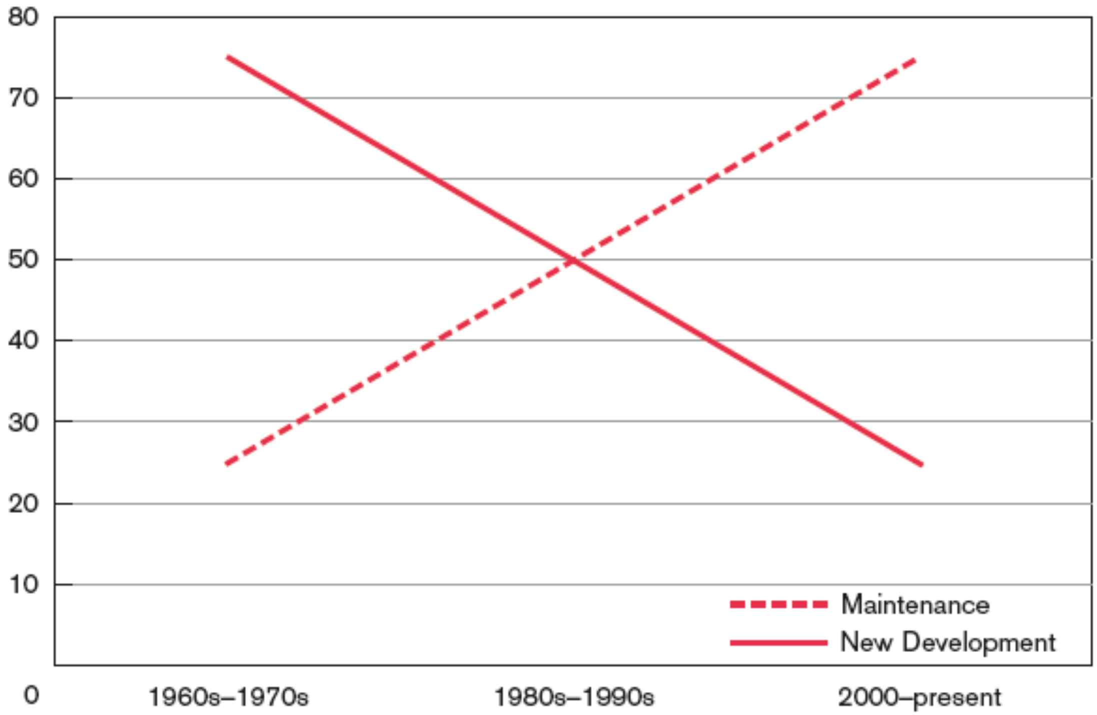


FIGURE 14-5
New development versus maintenance as
a percentage of the software budget over the years

(*Source:* Based on Pressman, 2005.)

The Cost of Maintenance (Cont.)

- Factors that influence system maintainability:
  - Latent defects
  - Number of customers for a given system
  - Quality of system documentation
  - Maintenance personnel
  - Tools
  - Well-structured programs

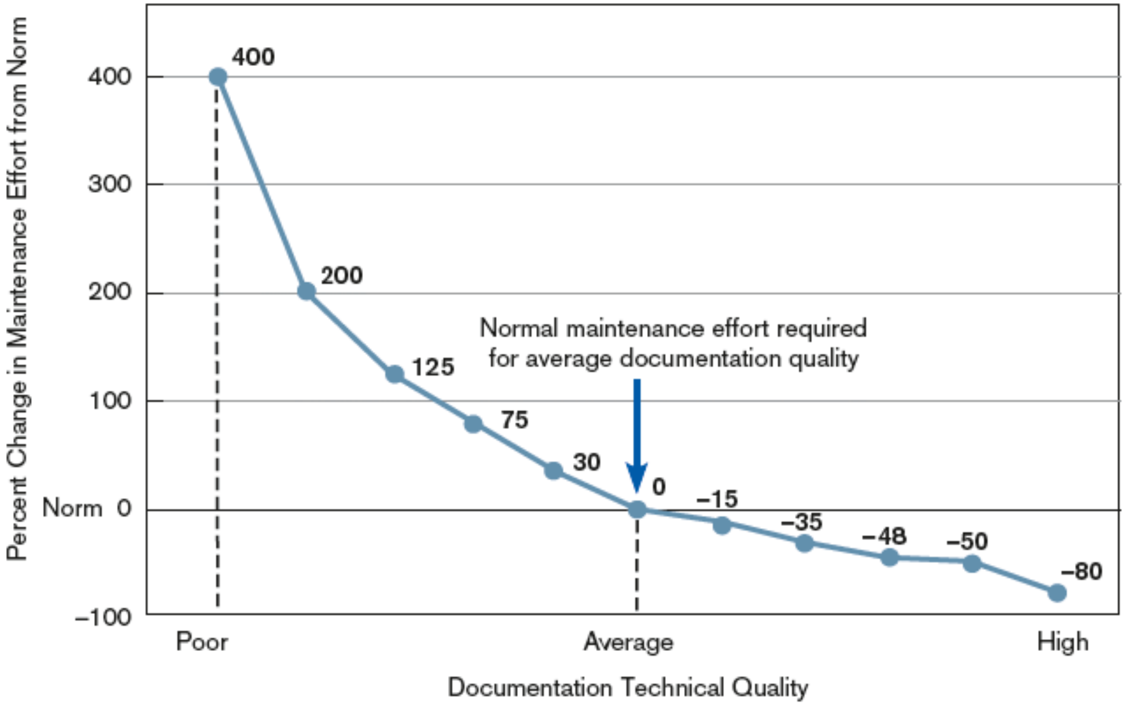


FIGURE 14-6
Quality documentation eases Maintenance

Managing Maintenance Personnel

- Number of people working in maintenance has surpassed number working in development.
- Maintenance work is often viewed negatively by IS personnel.
- Organizations often rotate personnel in and out of maintenance roles in order to lessen negative feelings about maintenance.
- Organizations have historically rewarded people involved in new development better than maintenance personnel.

Managing Maintenance Personnel (Cont.)

- Three possible organizational structures:
  - **Separate** — maintenance group consists of different personnel than development group
  - **Combined** — developers also maintain systems
  - **Functional** — maintenance personnel work within the functional business unit

Managing Maintenance Personnel (Cont.)

Maintenance Organization Type	Advantages	Disadvantages
Separate	Improved system and documentation quality	Ignorance of critical undocumented information
Combined	Maintenance group knows all about system	Less emphasis on good documentation
Functional	Personnel have vested interest	Limited job mobility and human or technical resources

Measuring Maintenance Effectiveness

- Must measure the following factors:
  - Number of failures
  - Time between each failure
  - Type of failure
- **Mean time between failures** (**MTBF**): a measurement of error occurrences that can be tracked over time to indicate the quality of a system

Measuring Maintenance Effectiveness (Cont.)

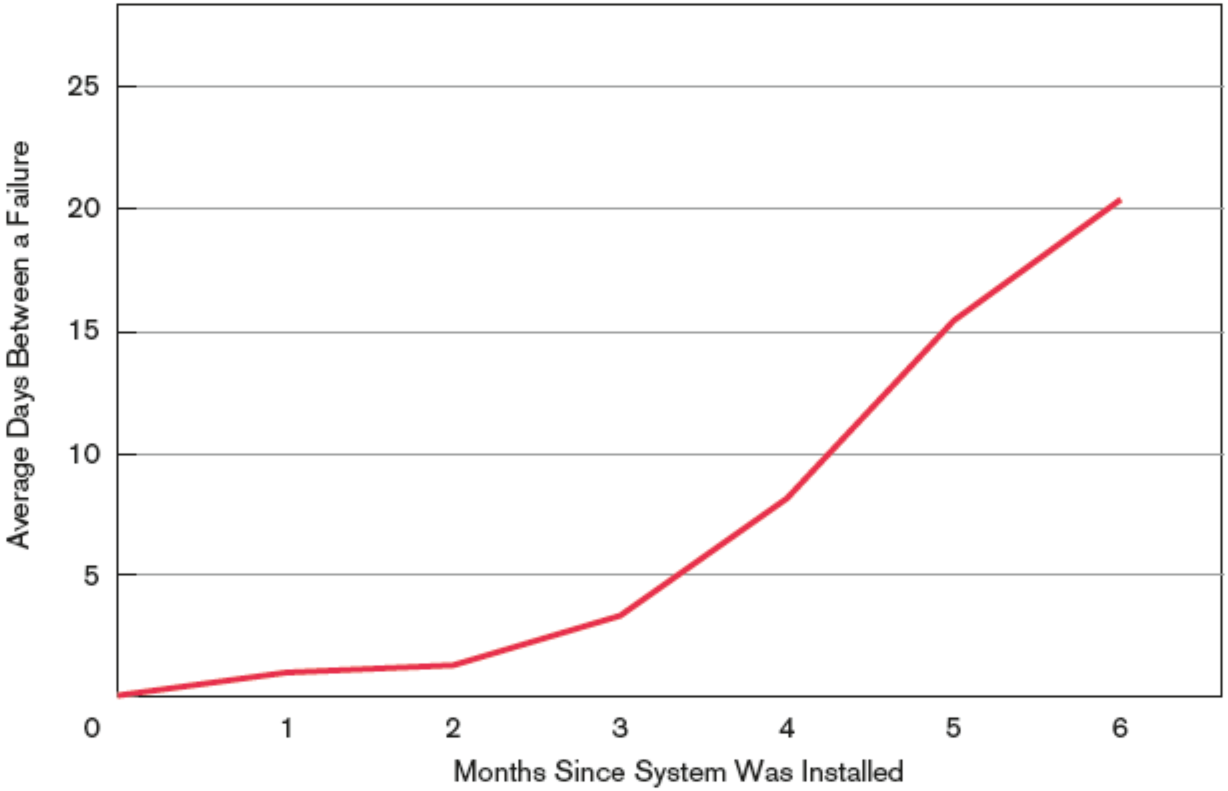


FIGURE 14-7
How the mean time between failures should change over time

Controlling Maintenance Requests

- Maintenance requests can be frequent.
- Prioritize based on type and urgency of request.
- Evaluations are

based on feasibility analysis.

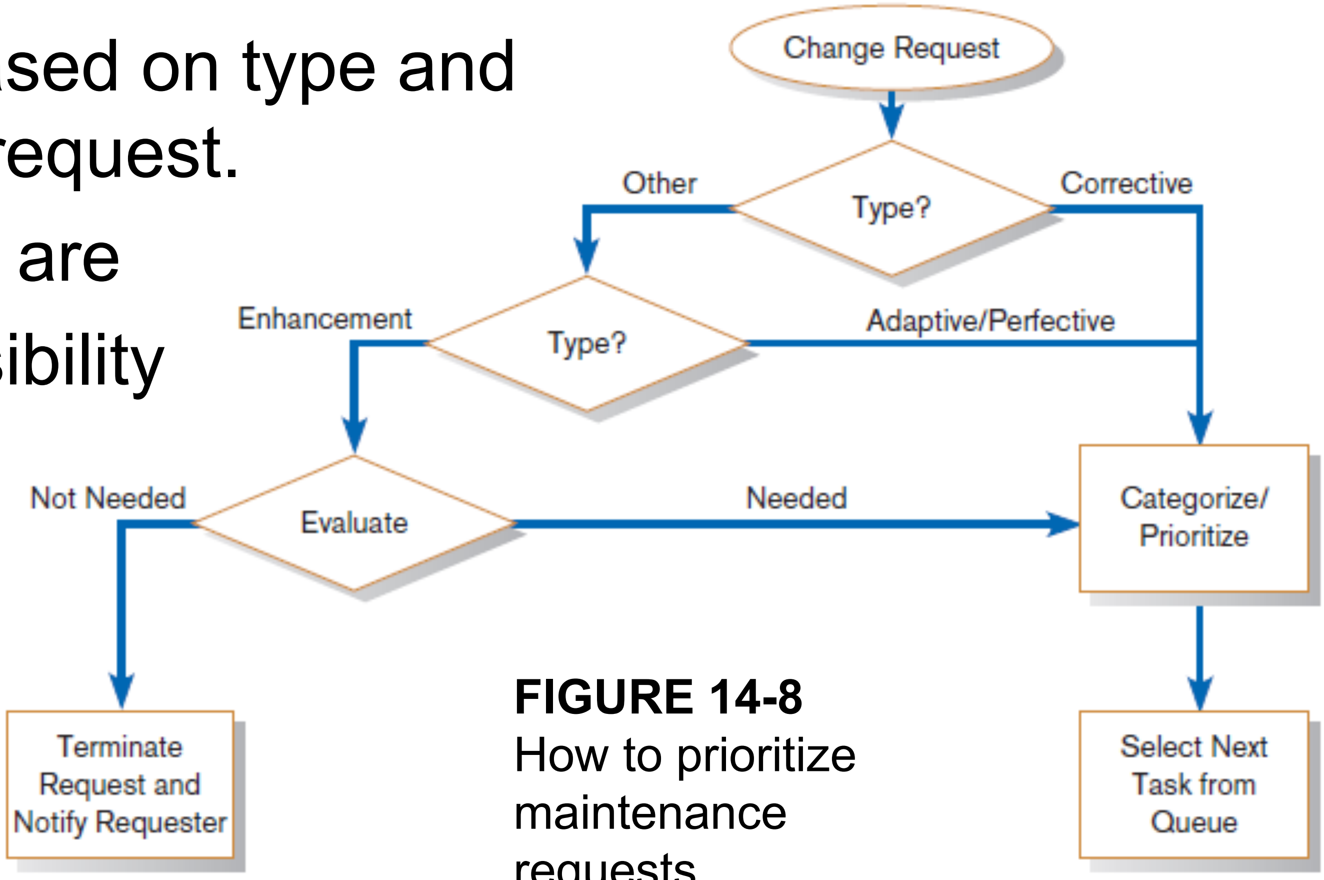


FIGURE 14-8
How to prioritize maintenance requests

Controlling Maintenance Requests (Cont.)

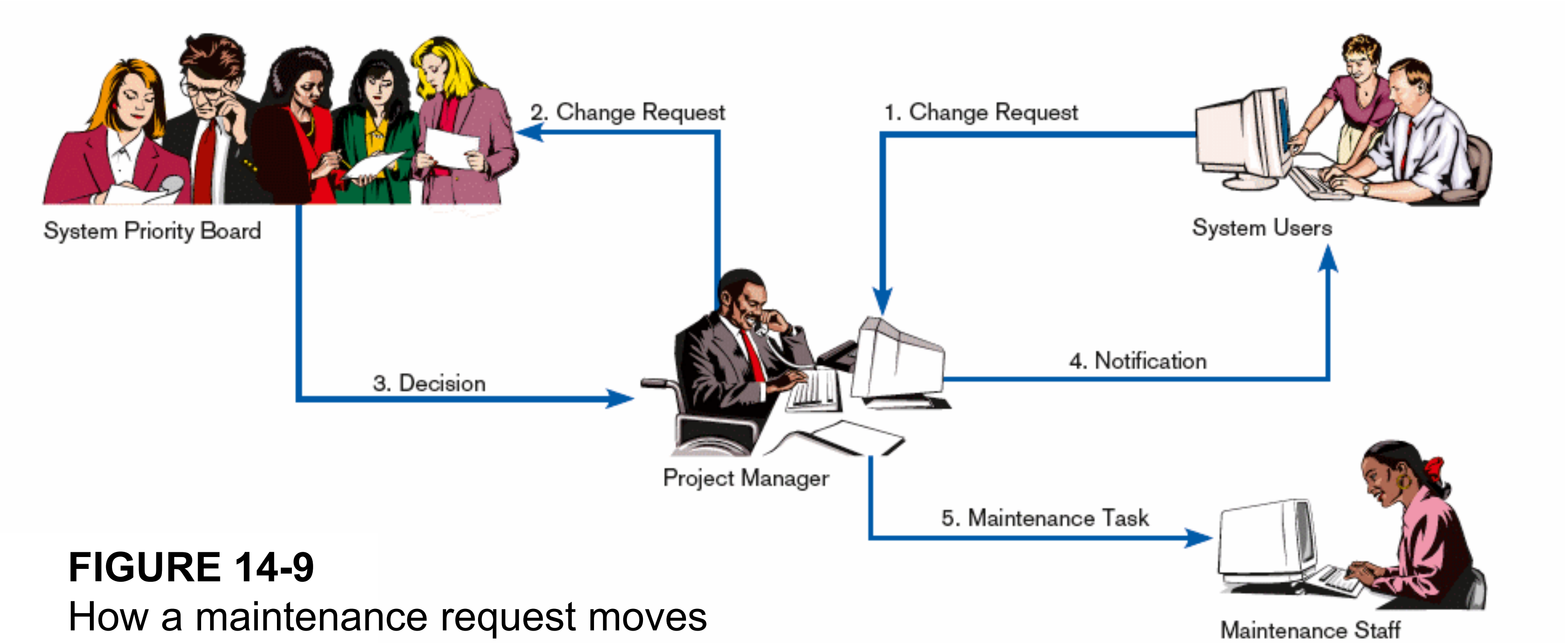


FIGURE 14-9
How a maintenance request moves through an organization

Configuration Management

- **Configuration management**: the process of ensuring that only authorized changes are made to the system
- **Baseline modules**: software modules that have been tested, documented, and approved to be included in the most recently created version of a system
- **System librarian**: a person responsible for controlling the checking out and checking in of baseline modules when a system is being developed or maintained
- **Build routines**: guidelines that list the instructions to construct an executable system from the baseline source code

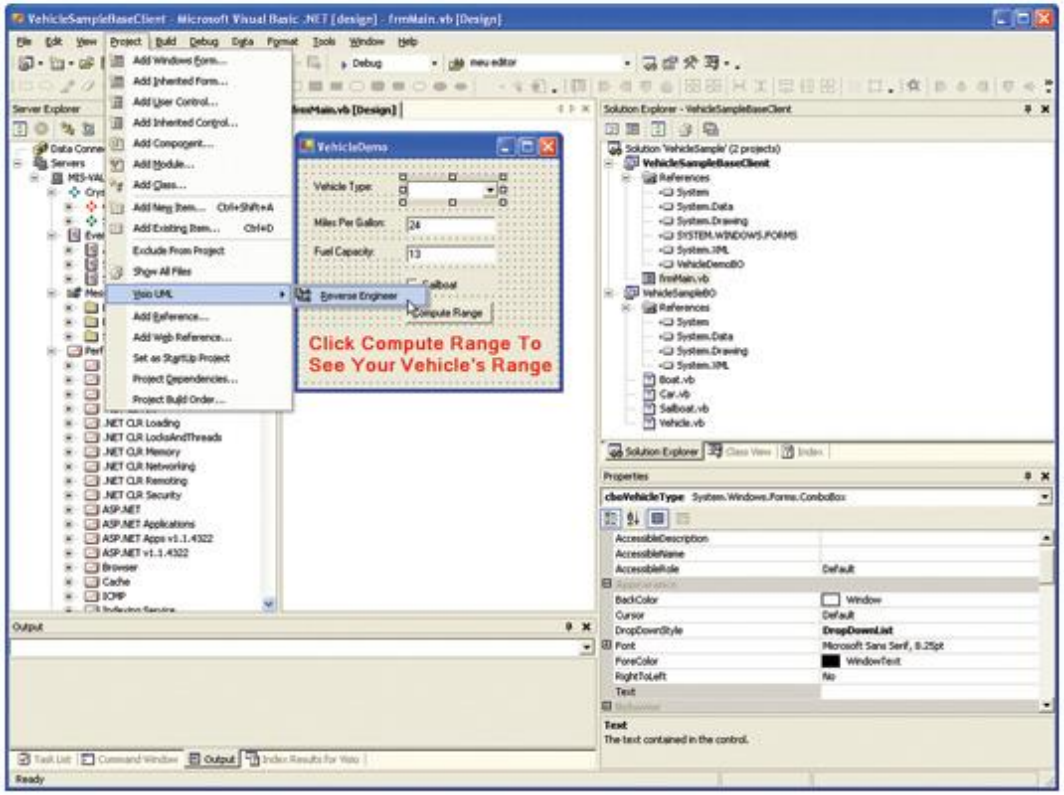
Role of CASE and Automated Development Tools in Maintenance

- Traditional systems development
  - Emphasis is on coding and testing.
  - Changes are implemented by coding and testing first.
  - Documentation is done after maintenance is performed.
  - Keeping documentation current is often neglected due to time-consuming nature of task.
- Development with CASE
  - Emphasis is on design documents.
  - Changes are implemented in design documents.
  - Code is regenerated using code generators.
  - Documentation is updated during maintenance.

Role of CASE and Automated Development Tools in Maintenance (Cont.)

- **Reverse engineering**: automated tools that read program source code as input and create graphical and textual representations of design-level information such as program control structures, data structures, logical flow, and data flow
- **Reengineering**: automated tools that read program source code as input; perform an analysis of the program’s data and logic; and then automatically, or interactively with a systems analyst, alter an existing system in an effort to improve its quality or performance

Role of CASE and Automated Development Tools in Maintenance (Cont.)



Vehicle Type:

Miles Per Gallon: 24

Fuel Capacity: 13

Click Compute Range To See Your Vehicle's Range

cboVehicleType System.Windows.Forms.ComboBox

AccessibleDescription	
AccessibleName	
AccessibleRole	Default
BackColor	Window
Cursor	Default
DropDownStyle	DropDownList
Font	Microsoft Sans Serif, 8.25pt
ForeColor	WindowText
RightToLeft	No
Text	

Text
The text contained in the control.

Figure 14-10 Visual Studio .NET engineer applications into Visio UML diagrams (*Source:* Microsoft Corporation.)

Web Site Maintenance

- Special considerations:
  - 24 × 7 × 365
    - Nature of continuous availability makes maintenance challenging.
    - Pages under maintenance can be locked.
      - Consider using date and time stamps to indicate when changes are made instead.
  - Check for broken links

Web Site Maintenance (Cont.)

- HTML Validation
  - Pages should be processed by a code validation routine before publication.
- Reregistration
  - When content significantly changes, site may need to be reregistered with search engines.
- Future Editions
  - Consistency is important to users.
  - Post indications of future changes to the site.
  - Batch changes.

Electronic Commerce Application: Maintaining an Information System for Pine Valley Furniture’s WebStore

- To maintain PVF’s WebStore, the following questions need to be addressed:
  - “How much is our Web site worth?”
  - “How much does it cost our company when our Web site goes down?”
  - “How reliable does our Web site need to be?”

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.
  - Describe several facts that influence the cost of maintaining an information system and apply these factors to the design of maintainable systems.
  - Describe maintenance management issues, including alternative organizational structures, quality measurement, processes for handling change requests, and configuration management.
  - Explain the role of CASE tools in maintaining information systems.

References

- Chapter 13-14, J. S. Valacich, J. George, **Modern Systems Analysis and Design**. 8th Edition, Pearson 2017.