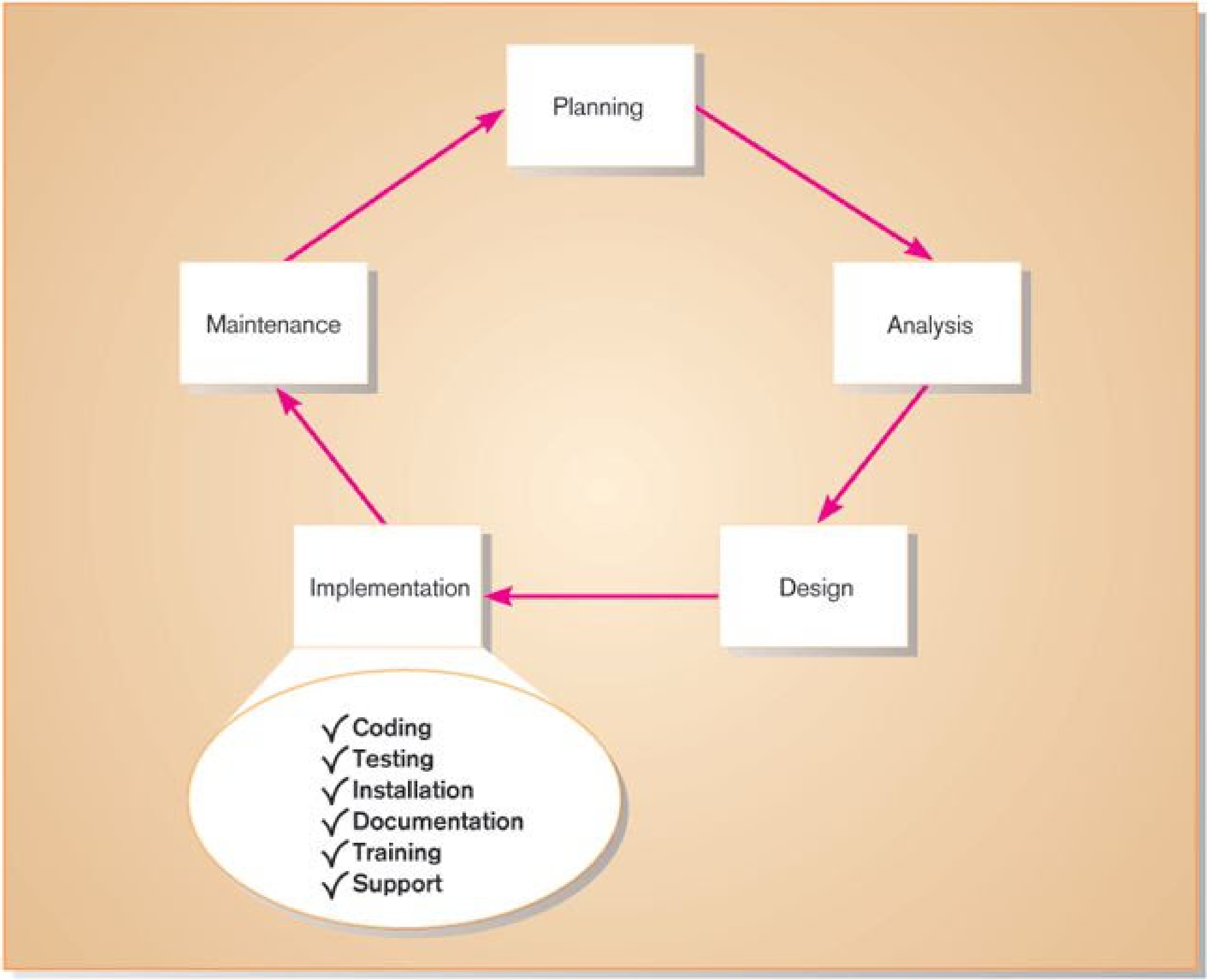
Figure 15-1

The systems development life cycle with the implementation phase highlighted



Purpose of System Implementation

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

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 can be performed in parallel with coding.
- Installation
  - The current system is replaced by the new system.

Deliverables

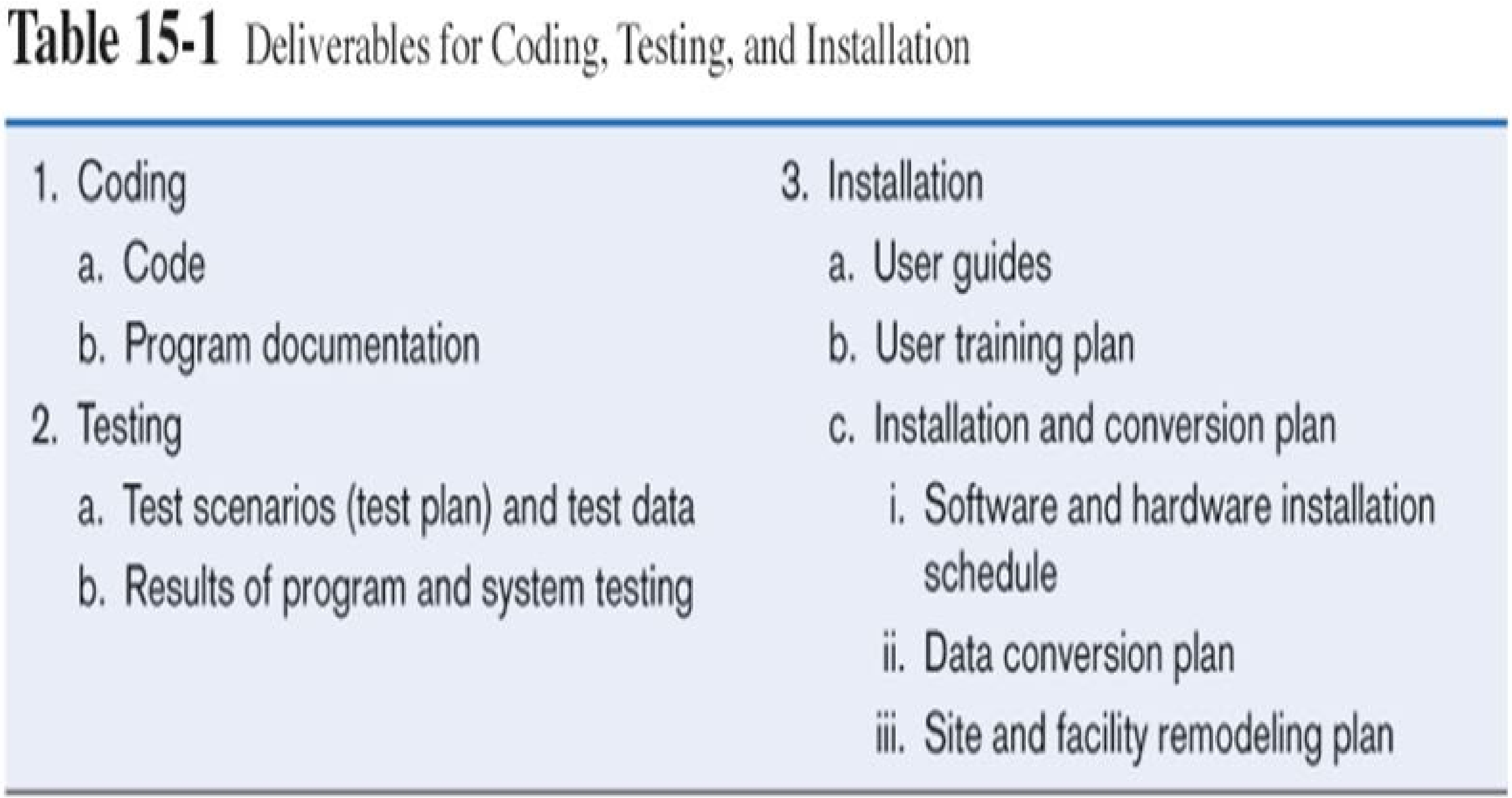


Table 15-1 Deliverables for Coding, Testing, and Installation

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

The Process of 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

Deliverables

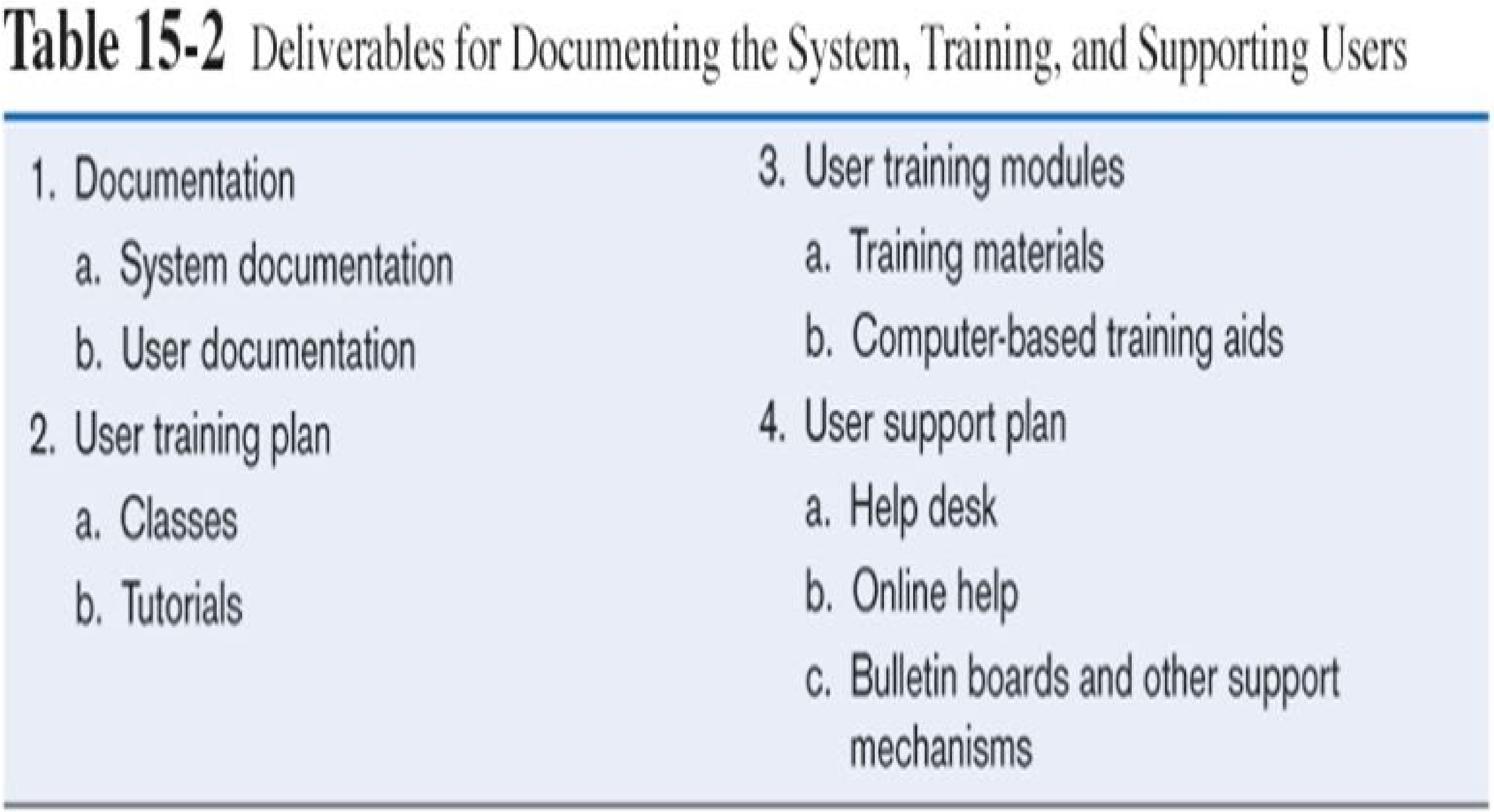


Table 15-2 Deliverables for Documenting the System, Training, and Supporting Users

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

Testing

- **Testing** involves the entire information systems. It requires testing each of:
  - – the individual programs (unit testing).
  - – testing the entire system of programs (system testing).
  - – testing applications with a large amount of data (volume testing).
  - – and testing all related systems together (integration testing).
  - – conducting tests required by the user (acceptance testing).

Significant Quote

Nixons Law

The man who can smile when things go wrong has thought of someone to blame.

Flon's axiom

There does not now, nor will there ever, exist a programming language in which it is the least bit hard to write bad programs.

(Lawrence Flon)

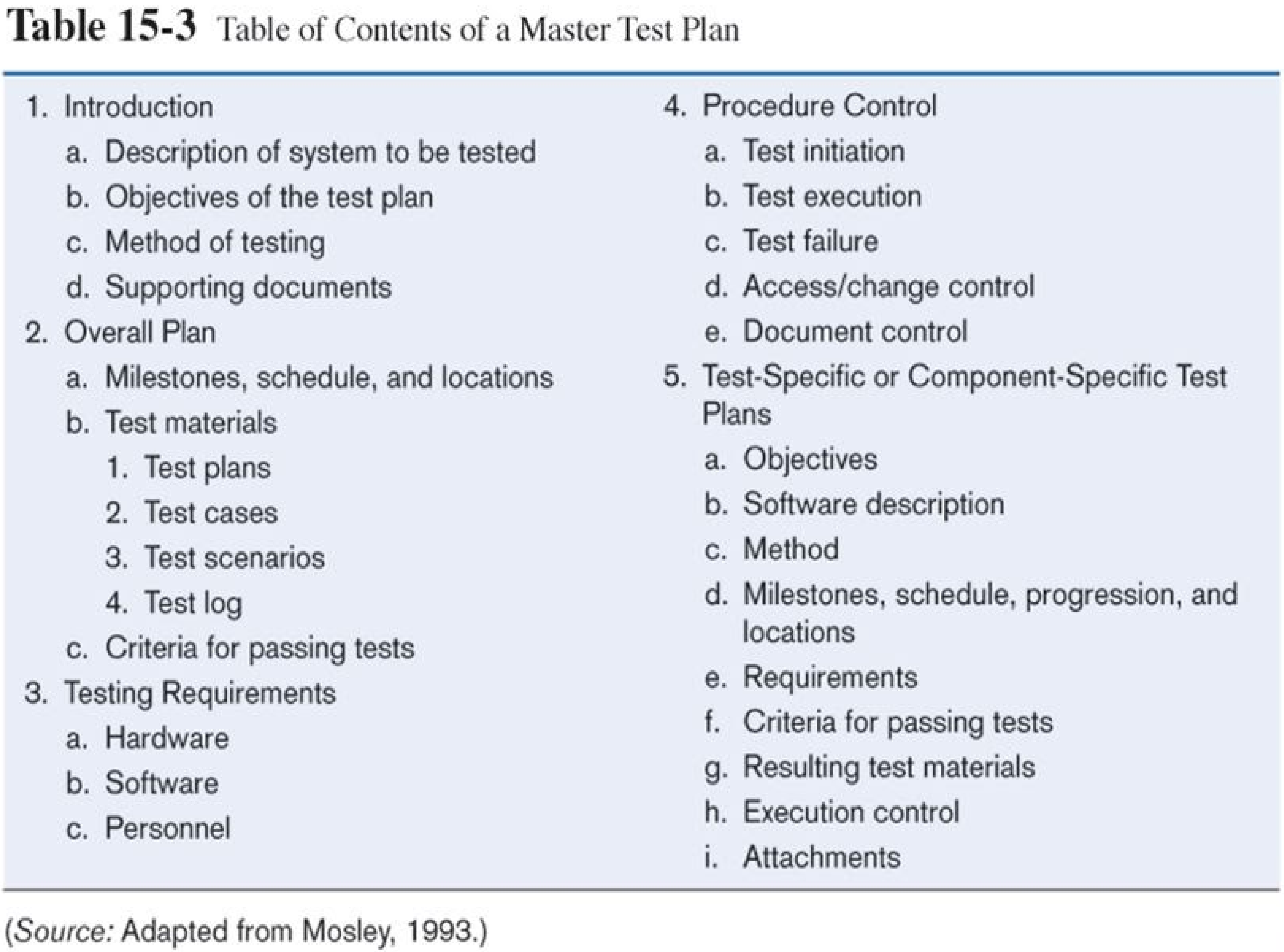


Table 15-3 Table of Contents of a Master Test Plan

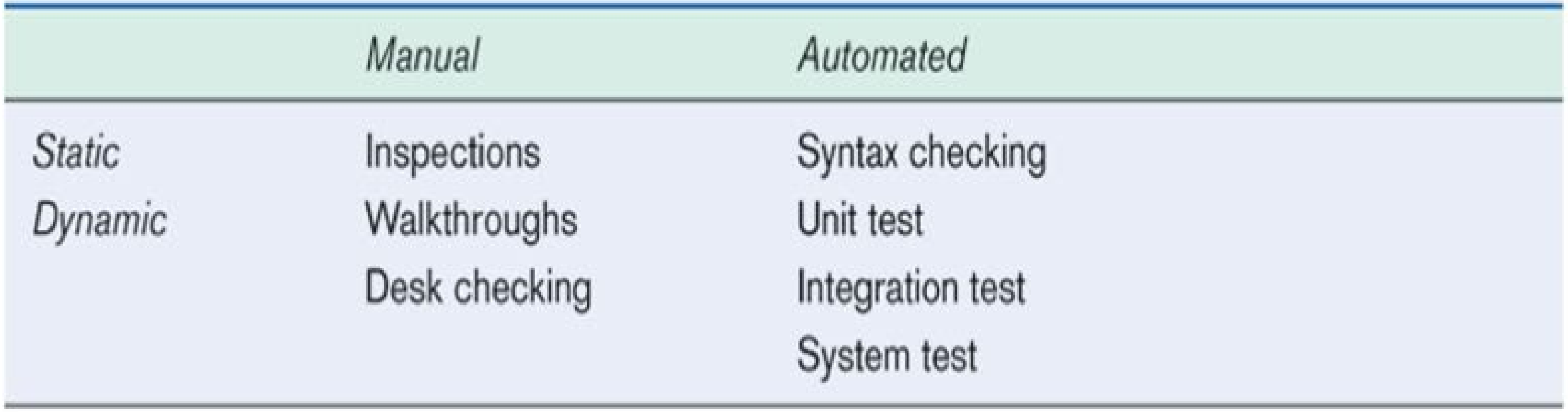
1. Introduction
   - a. Description of system to be tested
   - b. Objectives of the test plan
   - c. Method of testing
   - d. Supporting documents
2. Overall Plan
   - a. Milestones, schedule, and locations
   - b. Test materials
     1. Test plans
     2. Test cases
     3. Test scenarios
     4. 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.)

Test Classification

- Manual vs. Automated
- Static (syntax only) vs. Dynamic (execution)

Table 15-4 A Categorization of Test Types



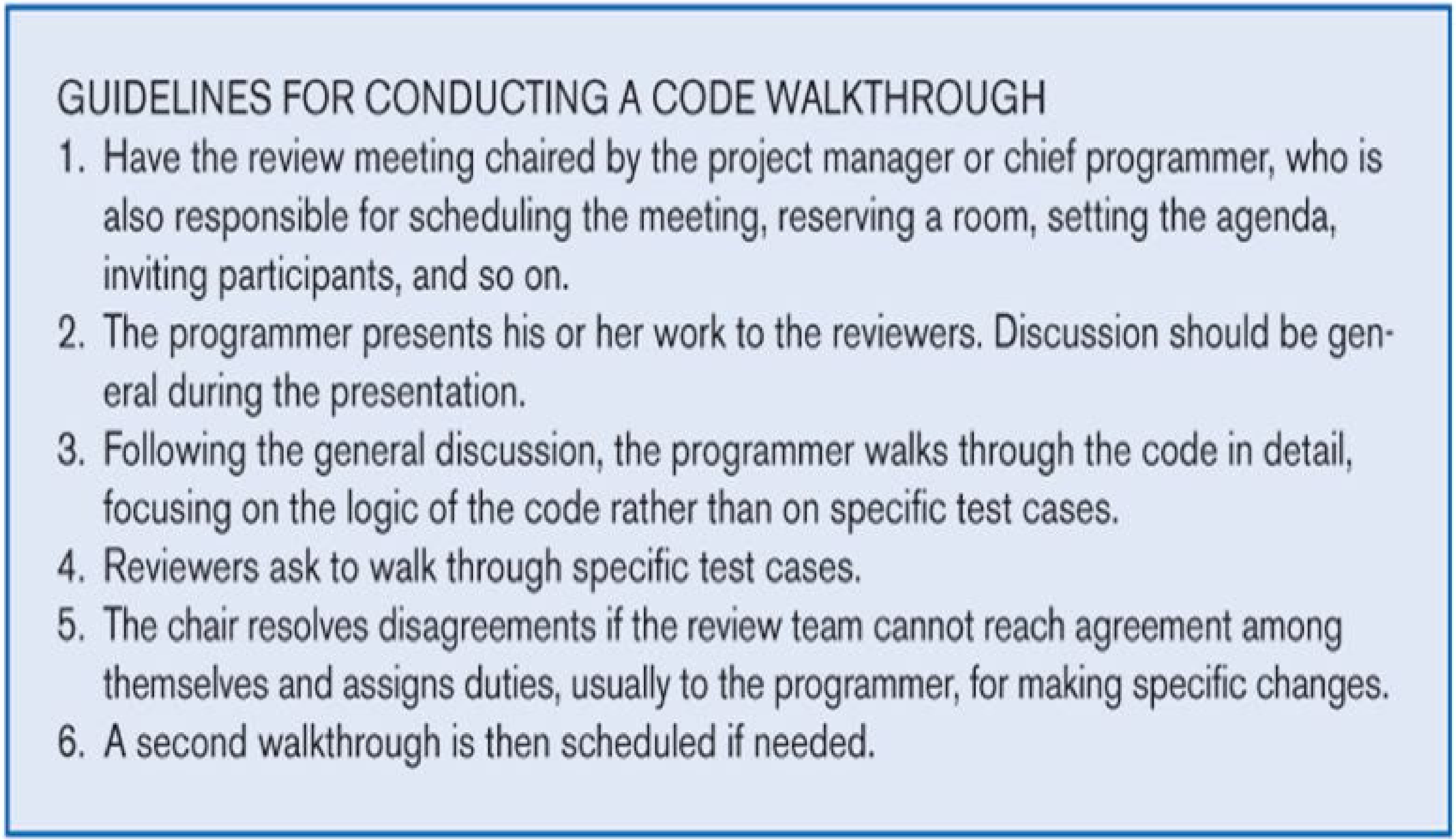
	Manual	*Automated*
Static	Inspections	Syntax checking
Dynamic	Walkthroughs	Unit test
	Desk checking	Integration test
		System test

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

Manual Testing Techniques

- 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; also called a structured walkthrough
- Desk Checking
  - A testing technique in which the program code is sequentially executed manually by the reviewer

Figure 15-2 Steps in a typical walkthrough



GUIDELINES FOR CONDUCTING A CODE WALKTHROUGH

1. Have the review meeting chaired by the project manager or chief programmer, who is also responsible for scheduling the meeting, reserving a room, setting the agenda, inviting participants, and so on.
2. The programmer presents his or her work to the reviewers. Discussion should be general during the presentation.
3. Following the general discussion, the programmer walks through the code in detail, focusing on the logic of the code rather than on specific test cases.
4. Reviewers ask to walk through specific test cases.
5. The chair resolves disagreements if the review team cannot reach agreement among themselves and assigns duties, usually to the programmer, for making specific changes.
6. A second walkthrough is then scheduled if needed.

Source: Adapted from Yourdon, 1989.

Automated Testing Techniques

- Syntax Checking
  - The compiler is run against the source code to identify syntax errors.
- Unit Testing
  - Each module is tested alone in an attempt to discover any errors in its code, also called module testing.
- 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.

Automated Testing Techniques (cont.)

- System Testing
  - The bringing together of all 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, 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.

Test Cases

- Test case: a scenario of transactions, queries or navigation paths
- Can represent either:
  - Typical system use
  - Critical system use
  - Abnormal system use
- Test cases and results should be thoroughly documented so they can be repeated for each revision of an application.

Test Cases (cont.)

- Test cases are usually developed by analysts.
- Test cases should not be created by the programmers.
- Separate people should program and test in order to ensure objectivity.
- Programmers use symbolic debuggers to isolate causes for errors.

User Acceptance Testing

- Actual users test a completed information system.
- End result is the users' final acceptance of the system.
- Alpha testing: use simulated data
- Beta testing: use real data in real user environment

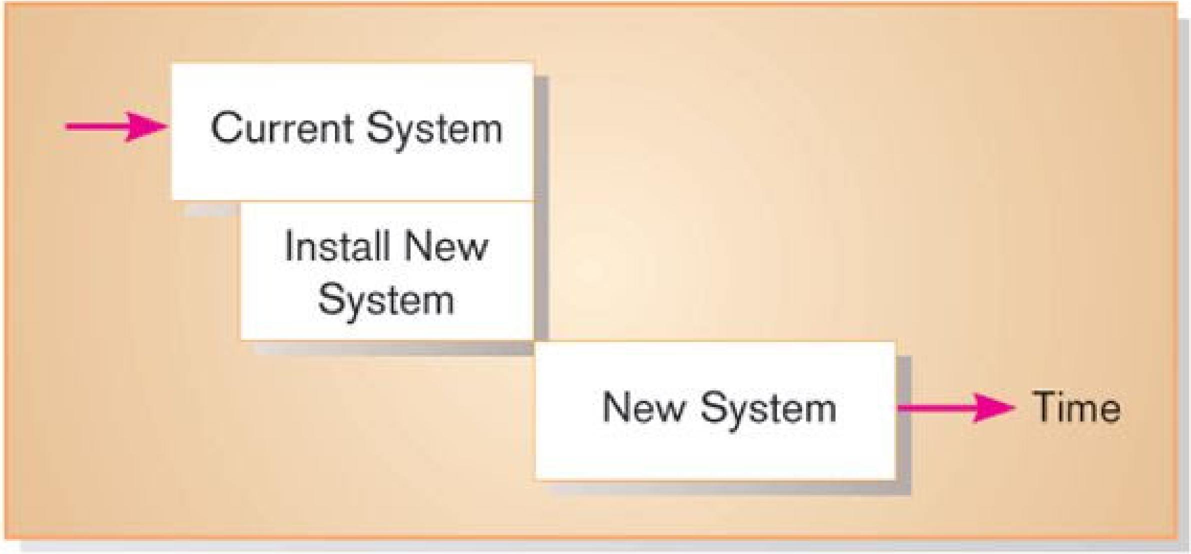
Types of Alpha Tests

- 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

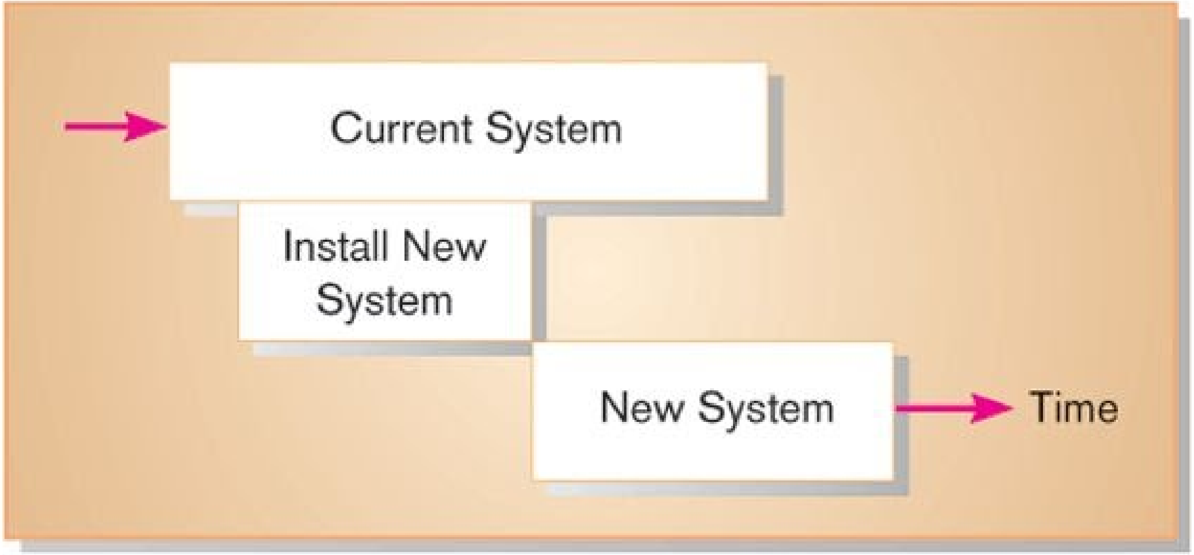
- The organizational process of changing over from the current information system to a new one
- Four installation strategies:
  - Direct Installation
  - Parallel Installation
  - Single-location installation
  - Phased Installation

Figure 15-6a Comparison of installation strategies - Direct installation



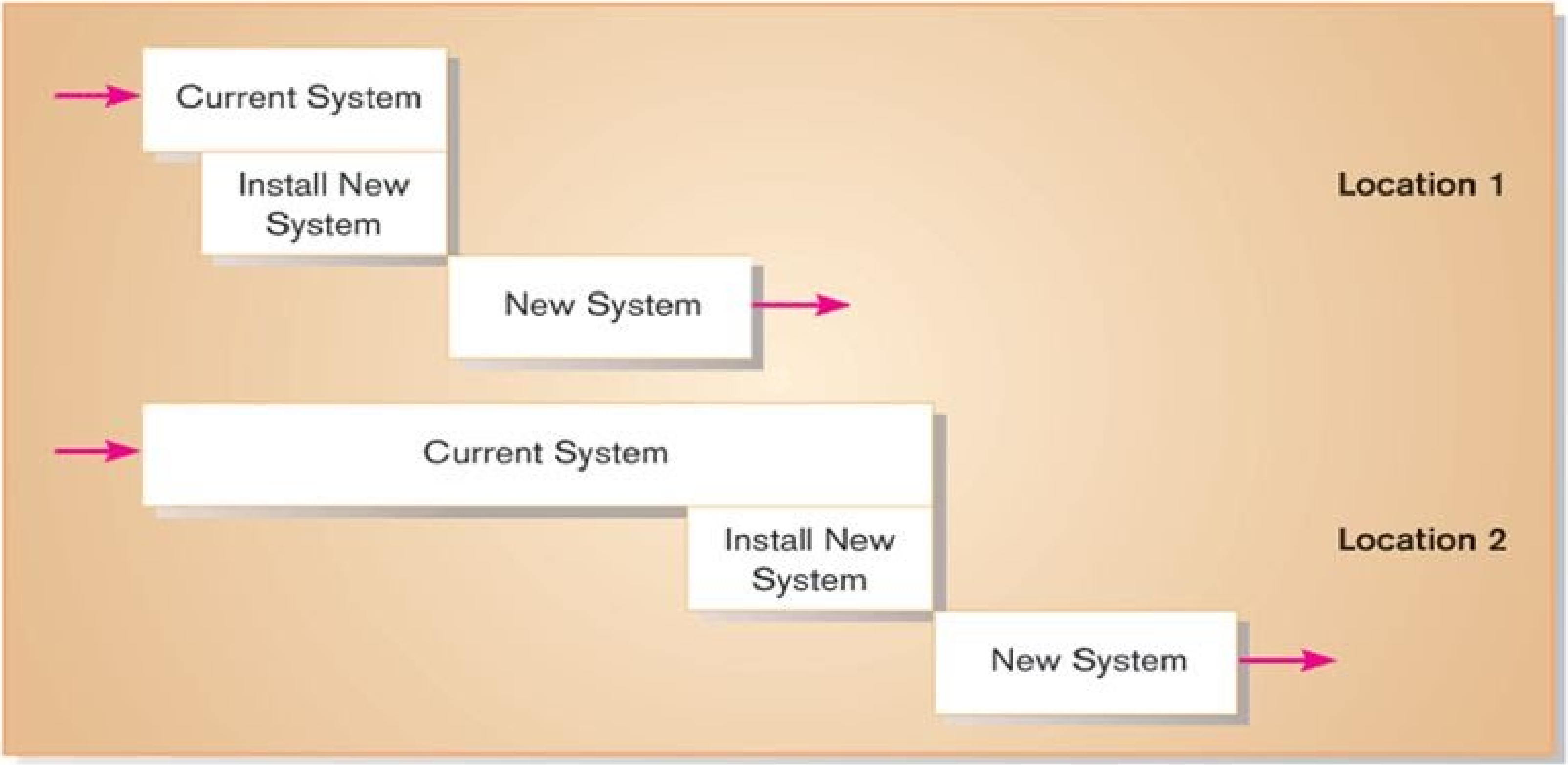
Turning off the old system when the new one is turned on

Figure 15-6b Comparison of installation strategies - Parallel installation



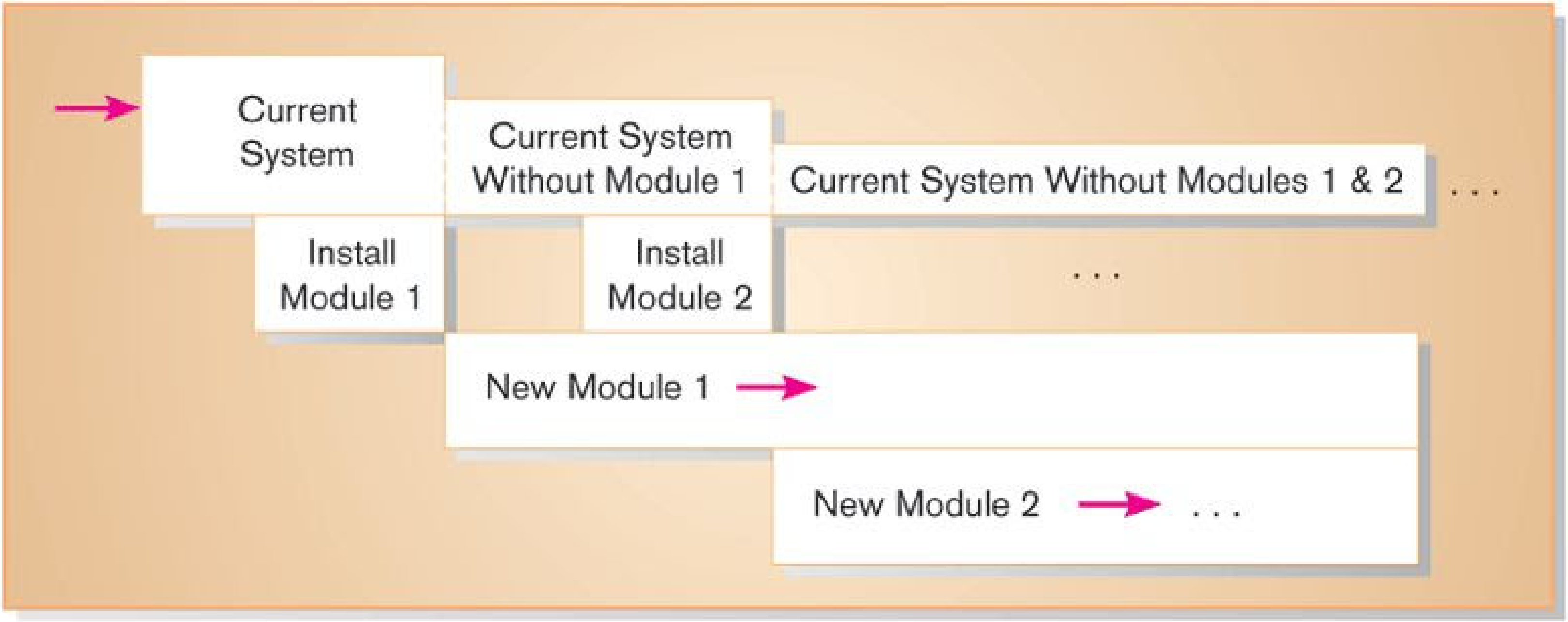
Running the old and new information systems at the same time until management decides the old system can be turned off

Figure 15-6c Comparison of installation strategies - Single-location installation (with direct installation at each location)



Trying out an information system at one site, then deciding if and how the new system should be deployed throughout the organization

Figure 15-6d Comparison of installation strategies - 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
  - Intended audience: maintenance programmers
  - Internal documentation: embedded in the program source code or generated at compile time
  - External documentation: includes data flow and entity-relationship diagrams

Documenting the System (cont.)

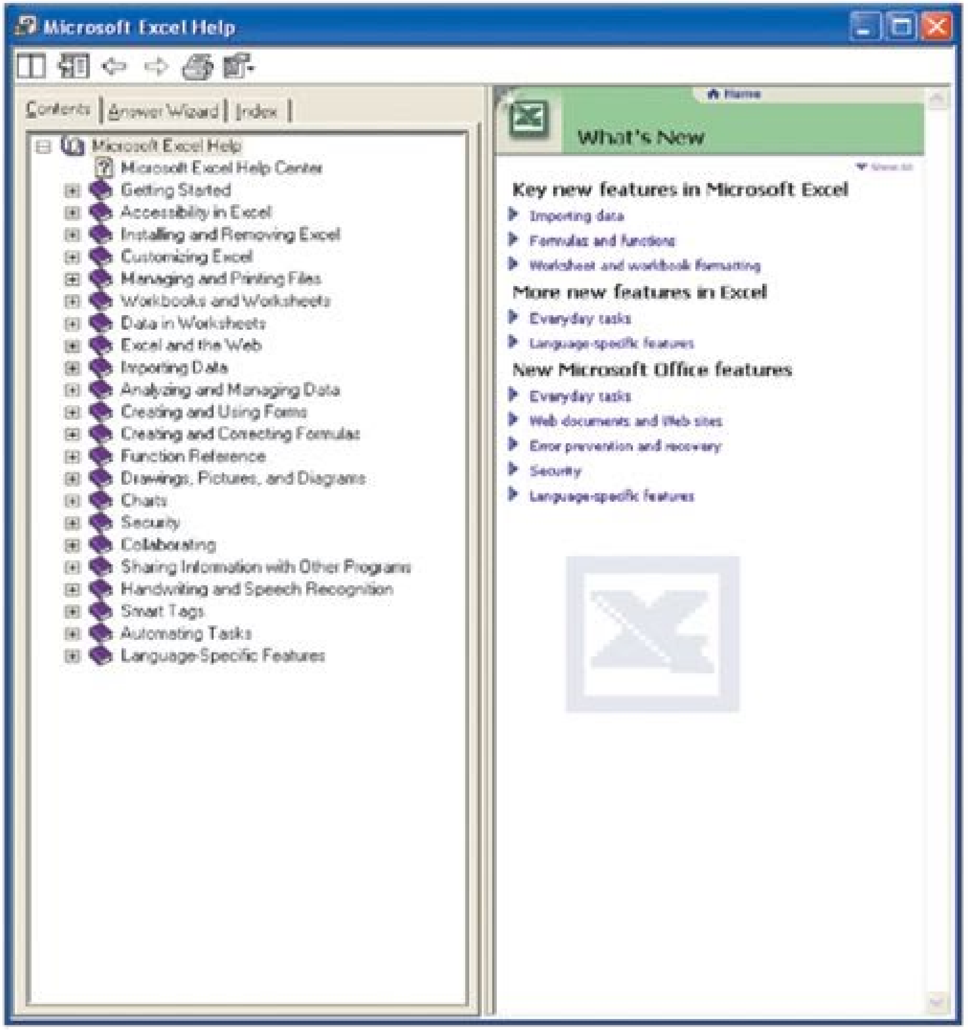
- User Documentation
  - Written or other visual information about an application system, how it works, and how to use it
- Preparing user documentation
  - Traditional source has been information systems department
  - Application-oriented documentation is now often supplied by vendors and users themselves

Table 15-6
Outline of a Generic User's Guide

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.)

Figure 15-8 Contents of the help facility in Microsoft ExcelTM



Microsoft Excel Help
- Microsoft Excel Help Center
- Getting Started
- Accessibility in Excel
- Installing and Removing Excel
- Customizing Excel
- Managing and Printing Files
- Workbooks and Worksheets
- Data in Worksheets
- Excel and the Web
- Importing Data
- Analyzing and Managing Data
- Creating and Using Forms
- Creating and Correcting Formulas
- Function Reference
- Drawings, Pictures, and Diagrams
- Charts
- Security
- Collaborating
- Sharing Information with Other Programs
- Handwriting and Speech Recognition
- Smart Tags
- Automating Tasks
- Language-Specific Features

Key new features in Microsoft Excel
- Importing data
- Formulas and functions
- Worksheet and workbook formatting

More new features in Excel
- Everyday tasks
- Language-specific features

New Microsoft Office features
- Everyday tasks
- Web documents and Web sites
- Error prevention and recovery
- Security
- Language-specific features

User documentation is typically in the form of online help

Training Information Systems Users

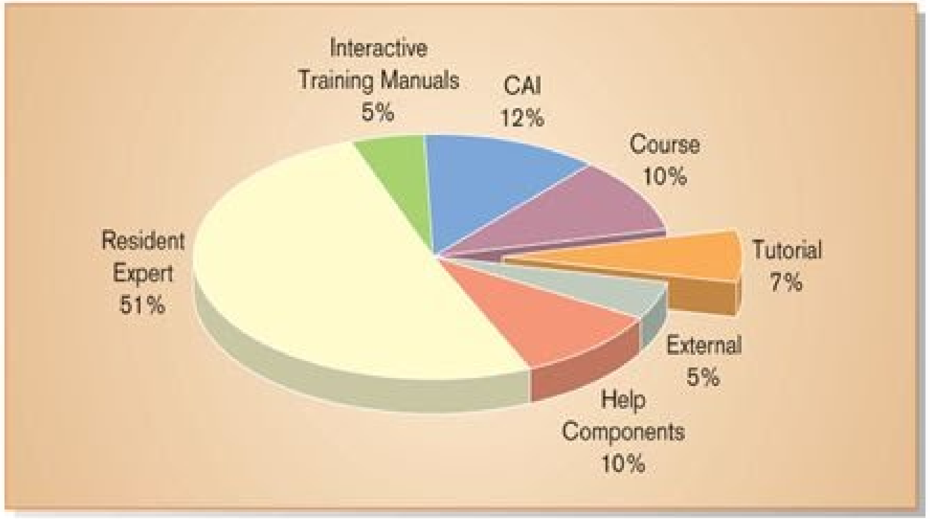
- Potential training topics
  - Use of the system
  - General computer concepts
  - Information system concepts
  - Organizational concepts
  - System management
  - System installation

Table 15-7 Seven Common Methods for Computer Training

1. Tutorial—one person taught at a time
2. Course—several people taught at a time
3. Computer-aided instruction
4. Interactive training manuals—combination of tutorials and computer-aided instruction
5. Resident expert
6. Software help components
7. External sources, such as vendors

(*Source:* Adapted from Nelson and Cheney, 1987.)

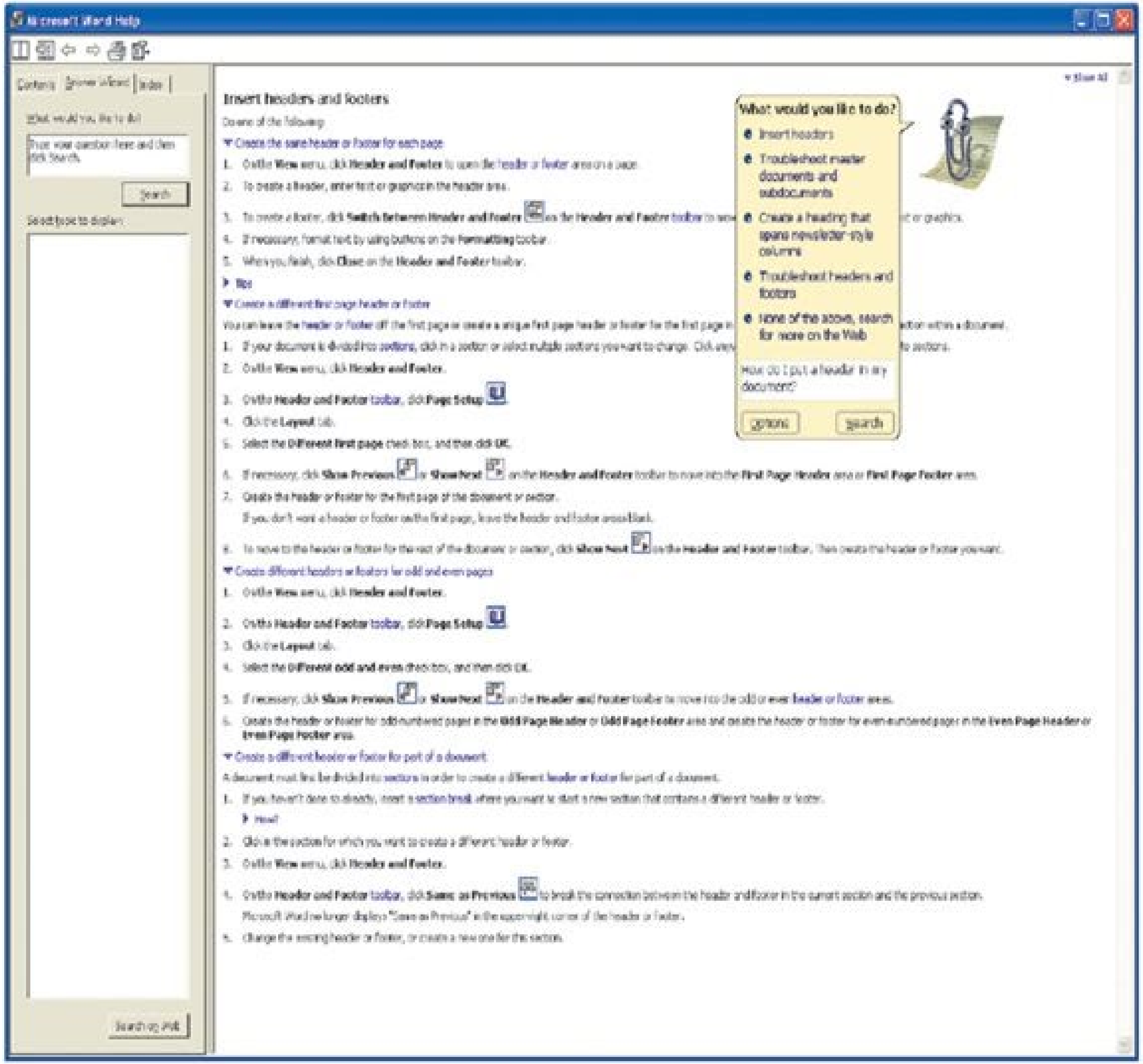
Figure 15-11 Frequency of use of computer training methods



Source: From Nelson and Cheney, "Training End Users: An Exploratory Study," MIS Quarterly, 1987, used with permission.

By far the most common training method is informal, via interaction with an in-house expert on the software

Figure 15-12 A Microsoft Office Assistant™ note



Insert headers and footers

Do one of the following:

▼ Create the same header or footer for each page

1. On the **View** menu, click **Header and Footer** to open the header or footer area on a page.
2. To create a header, enter text or graphics in the header area.
3. To create a footer, click **Switch Between Header and Footer** on the **Header and Footer** toolbar to mov … t or graphics.
4. If necessary, format text by using buttons on the **Formatting** toolbar.
5. When you finish, click **Close** on the **Header and Footer** toolbar.

▶ Tips

▼ Create a different first page header or footer

You can leave the header or footer off the first page or create a unique first page header or footer for the first page in … ection within a document.

1. If your document is divided into sections, click in a section or select multiple sections you want to change. Click any … le sections.
2. On the **View** menu, click **Header and Footer**.
3. On the **Header and Footer** toolbar, click **Page Setup**.
4. Click the **Layout** tab.
5. Select the **Different first page** check box, and then click **OK**.
6. If necessary, click **Show Previous** or **Show Next** on the **Header and Footer** toolbar to move into the **First Page Header** area or **First Page Footer** area.
7. Create the header or footer for the first page of the document or section.
   If you don't want a header or footer on the first page, leave the header and footer areas blank.
8. To move to the header or footer for the rest of the document or section, click **Show Next** on the **Header and Footer** toolbar. Then create the header or footer you want.

▼ Create different headers or footers for odd and even pages

1. On the **View** menu, click **Header and Footer**.
2. On the **Header and Footer** toolbar, click **Page Setup**.
3. Click the **Layout** tab.
4. Select the **Different odd and even** check box, and then click **OK**.
5. If necessary, click **Show Previous** or **Show Next** on the **Header and Footer** toolbar to move into the odd or even header or footer areas.
6. Create the header or footer for odd-numbered pages in the **Odd Page Header** or **Odd Page Footer** area and create the header or footer for even-numbered pages in the **Even Page Header** or **Even Page Footer** area.

▼ Create a different header or footer for part of a document

A document must first be divided into sections in order to create a different header or footer for part of a document.

1. If you haven't done so already, insert a section break where you want to start a new section that contains a different header or footer.
   ▶ How?
2. Click in the section for which you want to create a different header or footer.
3. On the **View** menu, click **Header and Footer**.
4. On the **Header and Footer** toolbar, click **Same as Previous** to break the connection between the header and footer in the current section and the previous section.
   Microsoft Word no longer displays "Same as Previous" in the upper-right corner of the header or footer.
5. Change the existing header or footer, or create a new one for this section.

What would you like to do?

- Insert headers
- Troubleshoot master documents and subdocuments
- Create a heading that spans newsletter-style columns
- Troubleshoot headers and footers
- None of the above, search for more on the Web

How do I put a header in my document?

Electronic Performance Support Systems (EPSS), like Microsoft Office Assistant, are components of software applications that embed training and information for the user, in the form of tutorials, expert systems, and hyperlink jumps to reference topics.

Supporting Information Systems Users

- Support is extremely important to users
- Providing support can be expensive and time-consuming
- One approach is through automation
  - Internet-based online support forums
  - On-demand fax
  - Voice response systems
  - Knowledge bases

Providing Support via 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

- User questions and problems
- Recovery and backup
- Disaster recovery
- PC maintenance
- Writing newsletters
- Setting up user groups

Implementation Success Factors

- Biggest measure of success: will it be used?
- Major factors influencing likelihood of use:
  - Personal stake of users
  - System characteristics
  - User demographics
  - Organizational support
  - Performance
  - Satisfaction

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 system conversion.
  - ✓ Prepare a test plan for an information system.
  - ✓ Apply four installation strategies.
  - ✓ List deliverables for documentation, training, and user support.
  - ✓ Distinguish between system and user documentation.
  - ✓ Compare different user training modes.
  - ✓ Discuss issues of end-user support.
  - ✓ Explain factors influencing implementation success.