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.

FIGURE 13-1
Systems development life cycle with the implementation phase highlighted
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 performed in parallel with coding.

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

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

<table>
<thead>
<tr>
<th>Coding</th>
<th>Testing</th>
<th>Installation</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Code</td>
<td>a. Test scenarios (test plan) and test data</td>
<td>a. User guides</td>
</tr>
<tr>
<td>b. Program documentation</td>
<td>b. Results of program and system testing</td>
<td>b. User training plan</td>
</tr>
<tr>
<td>2. Testing</td>
<td></td>
<td>c. Installation and conversion plan</td>
</tr>
<tr>
<td>a. Test scenarios (test plan) and test data</td>
<td></td>
<td>i. Software and hardware installation schedule</td>
</tr>
<tr>
<td>b. Results of program and system testing</td>
<td></td>
<td>ii. Data conversion plan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>iii. Site and facility remodeling plan</td>
</tr>
</tbody>
</table>
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

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

<table>
<thead>
<tr>
<th>1. Documentation</th>
<th>3. User Training Modules</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. System documentation</td>
<td>a. Training materials</td>
</tr>
<tr>
<td>b. User documentation</td>
<td>b. Computer-based training aids</td>
</tr>
<tr>
<td>a. Classes</td>
<td>a. Help desk</td>
</tr>
<tr>
<td>b. Tutorials</td>
<td>b. Online help</td>
</tr>
<tr>
<td></td>
<td>c. Bulletin boards and other support mechanisms</td>
</tr>
</tbody>
</table>
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.

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

<table>
<thead>
<tr>
<th>1. Introduction</th>
<th>2. Overall Plan</th>
<th>3. Testing Requirements</th>
<th>4. Procedure Control</th>
<th>5. Test-Specific or Component-Specific Test Plans</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Description of system to be tested</td>
<td>a. Milestones, schedule, and locations</td>
<td>a. Hardware</td>
<td>a. Test Initiation</td>
<td>a. Objectives</td>
</tr>
<tr>
<td>c. Method of testing</td>
<td>i. Test plans</td>
<td>c. Personnel</td>
<td>c. Test failure</td>
<td>c. Method</td>
</tr>
<tr>
<td>d. Supporting documents</td>
<td>ii. Test cases</td>
<td></td>
<td>d. Test failure</td>
<td>d. Milestones, schedule, progression, and locations</td>
</tr>
<tr>
<td></td>
<td>iii. Test scenarios</td>
<td></td>
<td>e. Document control</td>
<td>e. Requirements</td>
</tr>
<tr>
<td></td>
<td>iv. Test log</td>
<td></td>
<td></td>
<td>f. Criteria for passing tests</td>
</tr>
<tr>
<td></td>
<td>c. Criteria for passing tests</td>
<td></td>
<td></td>
<td>g. Resulting test materials</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>h. Execution control</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>i. Attachments</td>
</tr>
</tbody>
</table>

(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
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.

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.
Automated Testing

- Improves testing quality
- Reduce testing time up to 80%
- Functions:
  - Create recorded data entry and user action scripts
  - Compare test results between test cases
  - Simulate high-volume for stress-testing

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
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.

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.
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**

<table>
<thead>
<tr>
<th>Generic Life-Cycle Phase</th>
<th>Generic Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements Specification</td>
<td>System Requirements Specification</td>
</tr>
<tr>
<td>Project Control Structuring</td>
<td>Resource Requirements Specification</td>
</tr>
<tr>
<td>System Development</td>
<td>Management Plan</td>
</tr>
<tr>
<td>Architectural design</td>
<td>Engineering Change Proposal</td>
</tr>
<tr>
<td>Prototype design</td>
<td>Architecture Design Document</td>
</tr>
<tr>
<td>Detailed design and implementation</td>
<td>Prototype Design Document</td>
</tr>
<tr>
<td>Test specification</td>
<td>Detailed Design Document</td>
</tr>
<tr>
<td>Test implementation</td>
<td>Test Specifications</td>
</tr>
<tr>
<td>System Delivery</td>
<td>Test Reports</td>
</tr>
<tr>
<td></td>
<td>User’s Guide</td>
</tr>
<tr>
<td></td>
<td>Release Description</td>
</tr>
<tr>
<td></td>
<td>System Administrator’s Guide</td>
</tr>
<tr>
<td></td>
<td>Reference Guide</td>
</tr>
<tr>
<td></td>
<td>Acceptance Sign-Off</td>
</tr>
</tbody>
</table>

(Source: Adapted from Bell and Evans, 1989.)
Preparing User Documentation

- Traditional source has been information systems department.
- Application-oriented documentation is now often supplied by vendors and users themselves.

FIGURE 13-6
Example of online user documentation
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 and Supporting Users (Cont.)

- **Computing infrastructure**: all of the resources and practices required to help people and adequately use computer systems to do their primary work.

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)
- 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 extremely important to users.
- Providing support can be expensive and time-consuming.

Automating Support

- One approach is through automation.
  - 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

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

- Biggest measure of success: Will it be used?
- Major factors influencing implementation success:
  - Management support
  - User involvement
  - 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

Security Issues

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

- Antivirus software
- Firewalls
- Anti-spyware software
- Intrusion detection systems
- Biometrics

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
Developing Test Cases for WebStore

Test case forms had the following sections:
- Test Case ID
- Category/Objective of Test
- Description
- System Version

Developing Test Cases for WebStore (Cont.)
- 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 Generate 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 (i.e. 1.0, 2.0, 3.0, etc.).

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.
WebStore Installation

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

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.