System Implementation

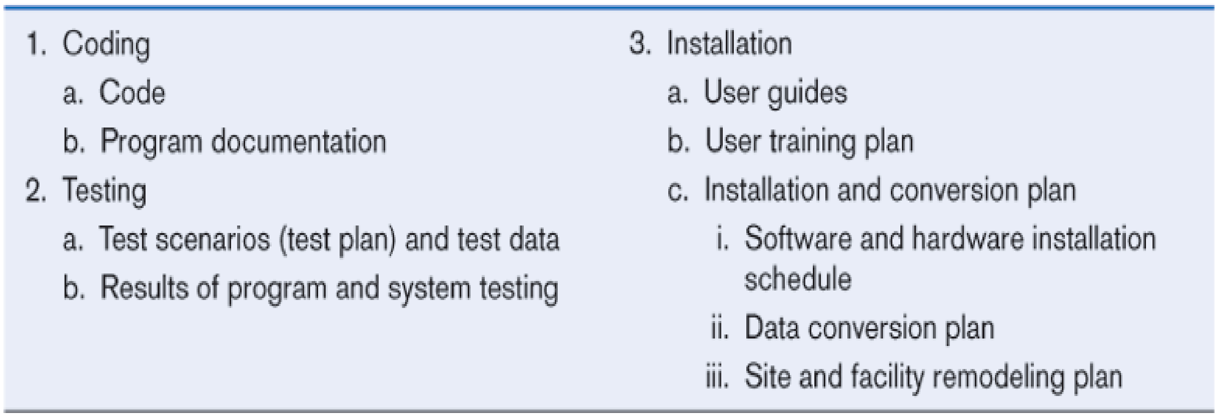
6.1 Purpose of System Implementation

- 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

6.2 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.

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

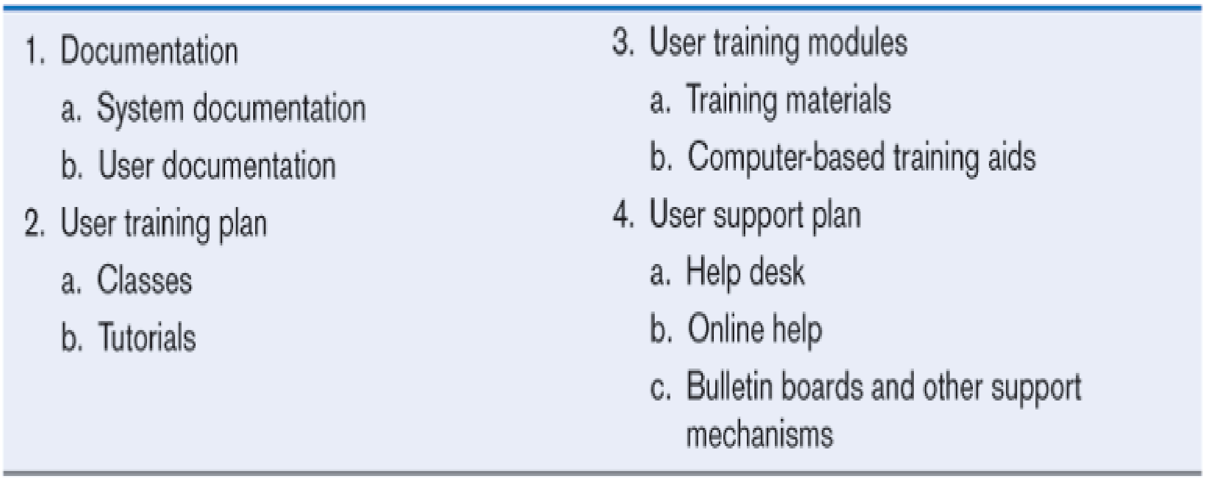


1. Coding
    a. Code
    b. Program documentation
2. Testing
    a. Test scenarios (test plan) and test data
    b. Results of program and system testing
3. Installation
    a. User guides
    b. User training plan
    c. Installation and conversion plan
        i. Software and hardware installation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

6.3 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 other software

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

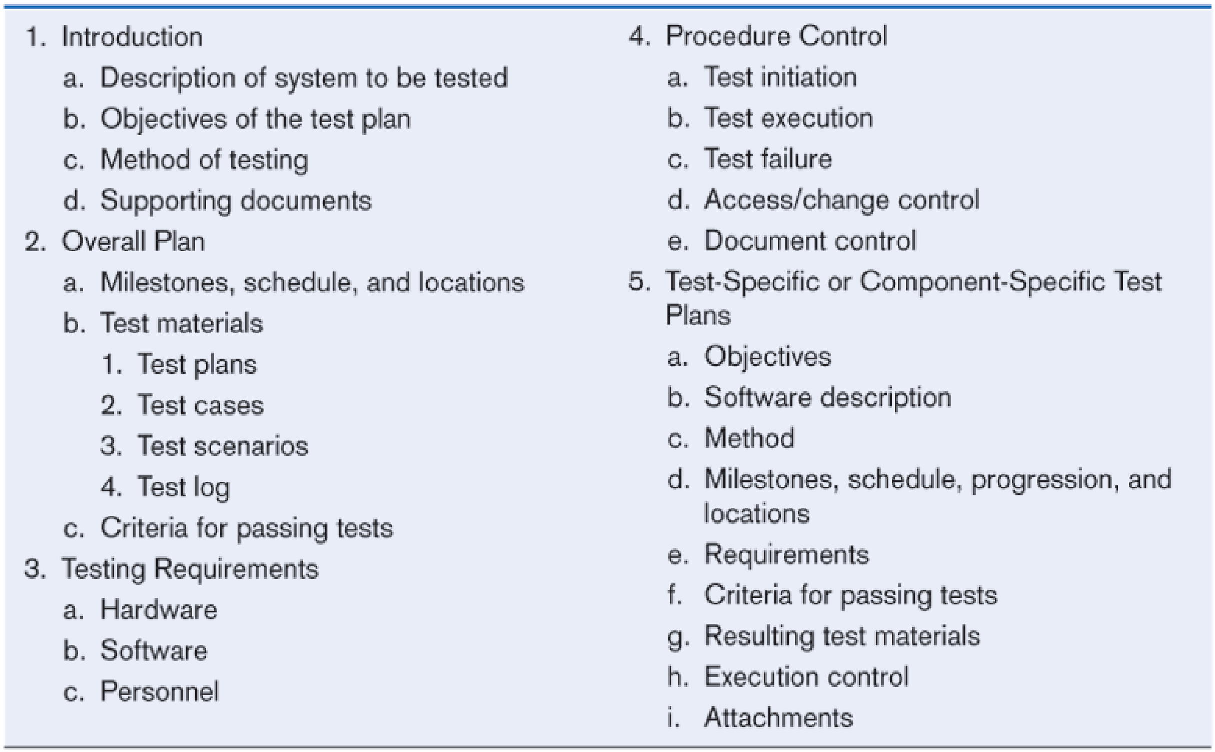
6.4 Testing

- **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).

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.)

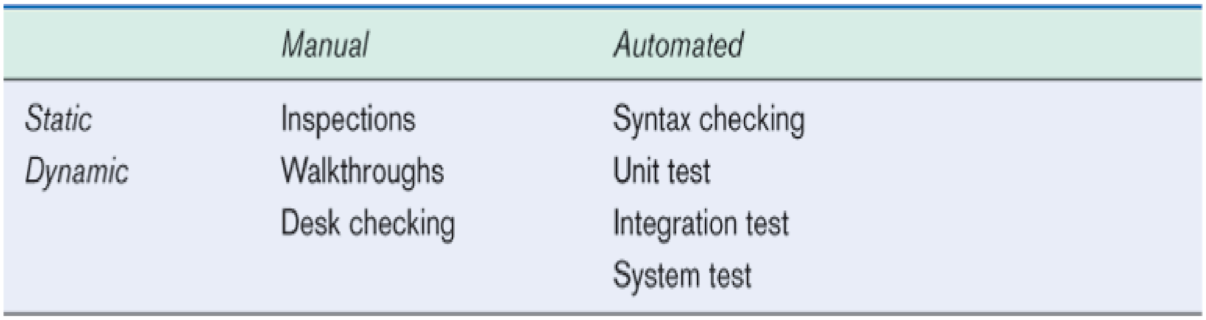
6.4.1 Software Application Testing

- 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.

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

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.

- System Testing
  - The bringing together of all the programs that a system comprises for testing purposes.
- Stub Testing
  - Stubs are computer programs that act as temporary replacement for a called module and give the same output as the actual product or software. Test stubs are mainly used in incremental testing's top-down approach.

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 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

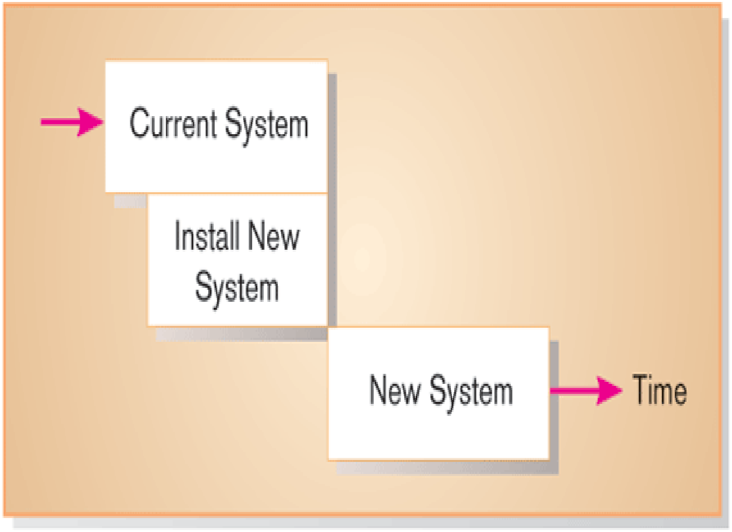
- Recovery testing
  - Forces software (or environment) to fail in order to verify that recovery is properly performed
- Security testing

  - 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

6.5. Installation or conversion methods

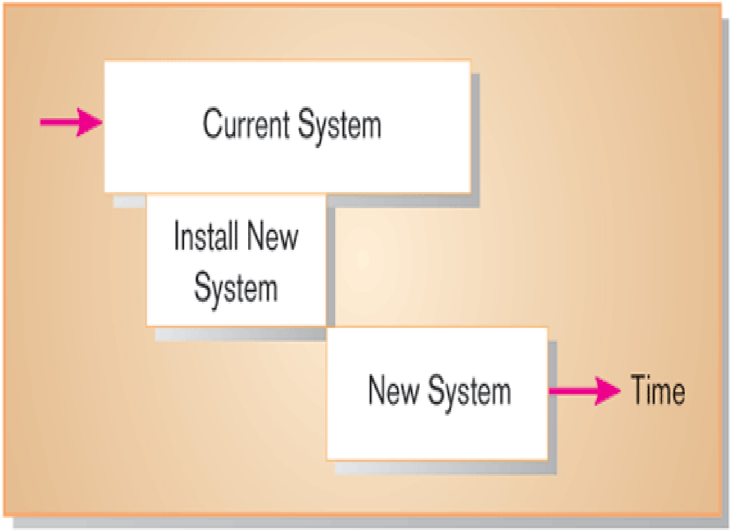
- The organizational process of changing over from the current information system to a new one
- Four installation strategies:
  - Direct Installation
  - Parallel Installation
  - Single-location installation
  - Phased Installation

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



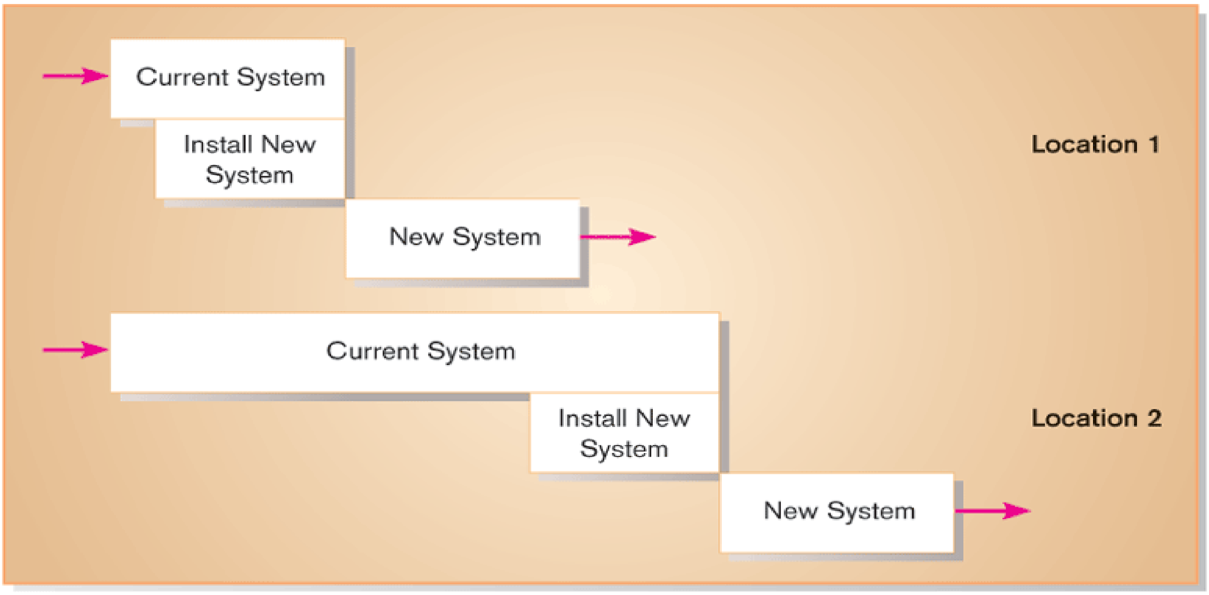
Turning off the old system when the new one is turned on

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



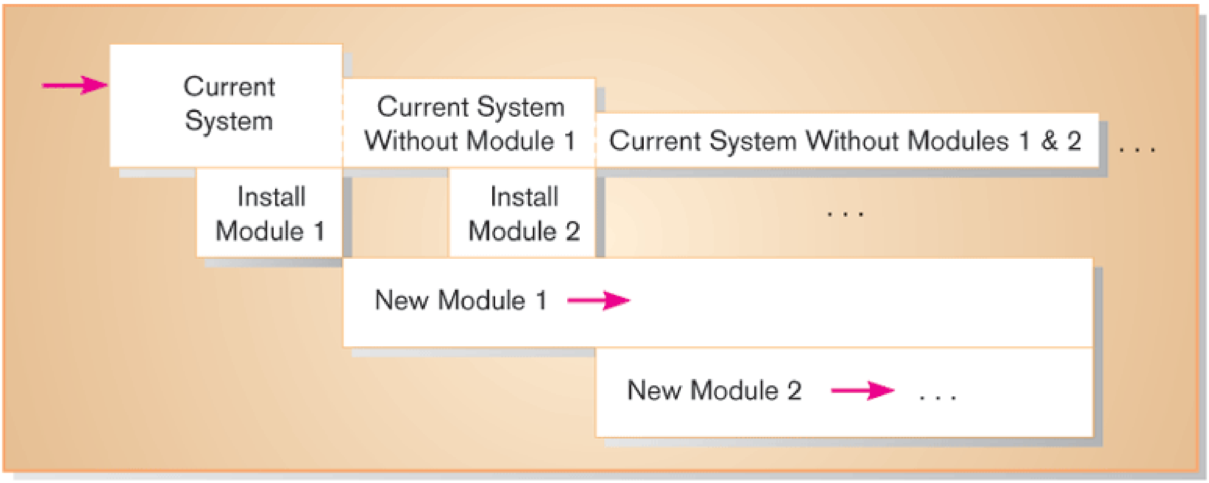
Running the old and new information systems at the same time until management decides the old system can be turned off

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
- 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

Training Information Systems Users

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

Supporting Information Systems Users

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

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

Implementation Success Factors

- Biggest measure of success: will it be used?
- Major factors are:
  - User age, education etc
  - Organizational support
  - Performance
  - System characteristics
  - 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

6.6 What Is Quality?

- Degree to which a system, component, or process meets

  (1) specified requirements, and

  (2) customer or user needs or expectations

- The totality of features and characteristics of a product or service that bear on its ability to satisfy specified or implied needs

Customer Satisfaction

Whole Organization Involvement

Quality Service

36

Quality Assurance

- A goal of quality assurance is continuous quality improvement.
- "Set of systematic activities providing evidence of the ability of the software process to produce a software product that is fit to use"
- **Software quality assurance** (**SQA**) consists of a means of monitoring the software engineering processes and methods used to ensure quality.

A Proven Successful Quality Assurance System : ISO 9000

ISO 9001:2000

- 0 – Introduction
- 1 – Scope
- 2 – Normative reference
- 3 – Terms and definitions

- 4 – Quality Management system
- 5 – Management responsibility
- 6 – Resource Management
- 7 – Product realization
- 8 – Measurement, analysis & improvement

ISO 9000:2005 – Vocabulary (examples)

- *Process* – set of interrelated or interacting activities which transform inputs into outputs
- *Product* – result of a process
- *Top management* – a person or group of people who direct and control an organization at the highest level
- *Requirement* – need or expectation that is stated, generally implied or obligatory
- *Customer satisfaction* – customer's perception of the degree to which the requirements have been fulfilled

6.7 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

The Flow of a Maintenance Request

Figure 16-9 How a maintenance request moves through an organization

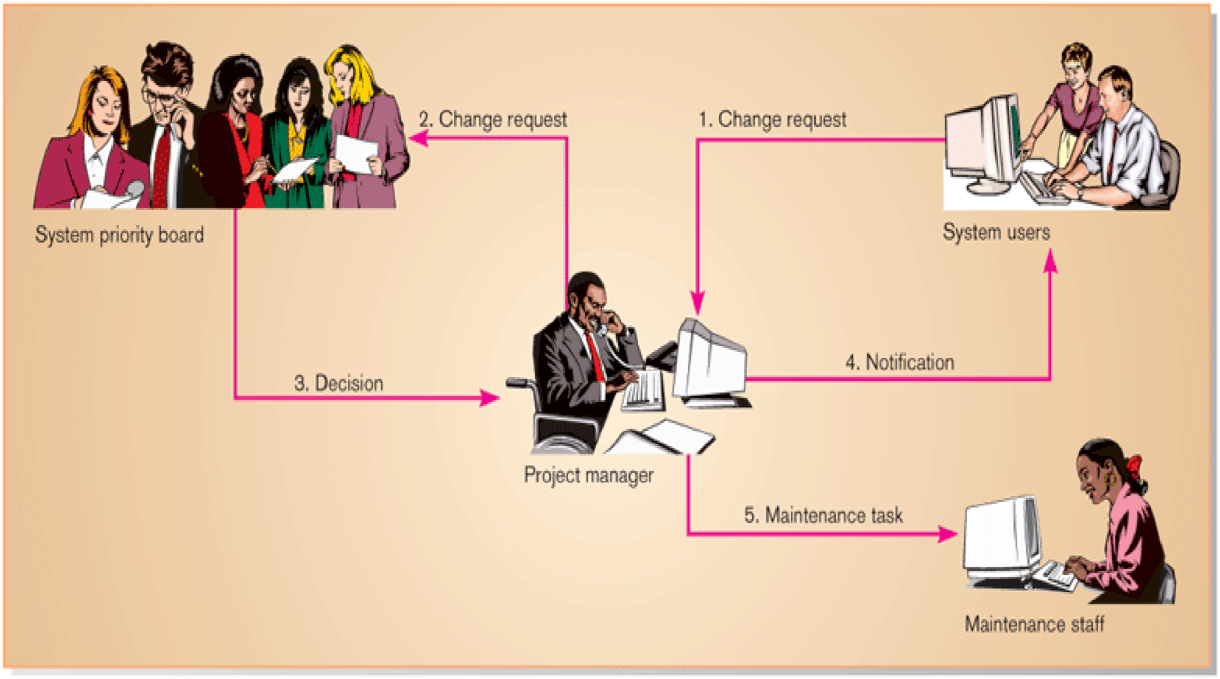
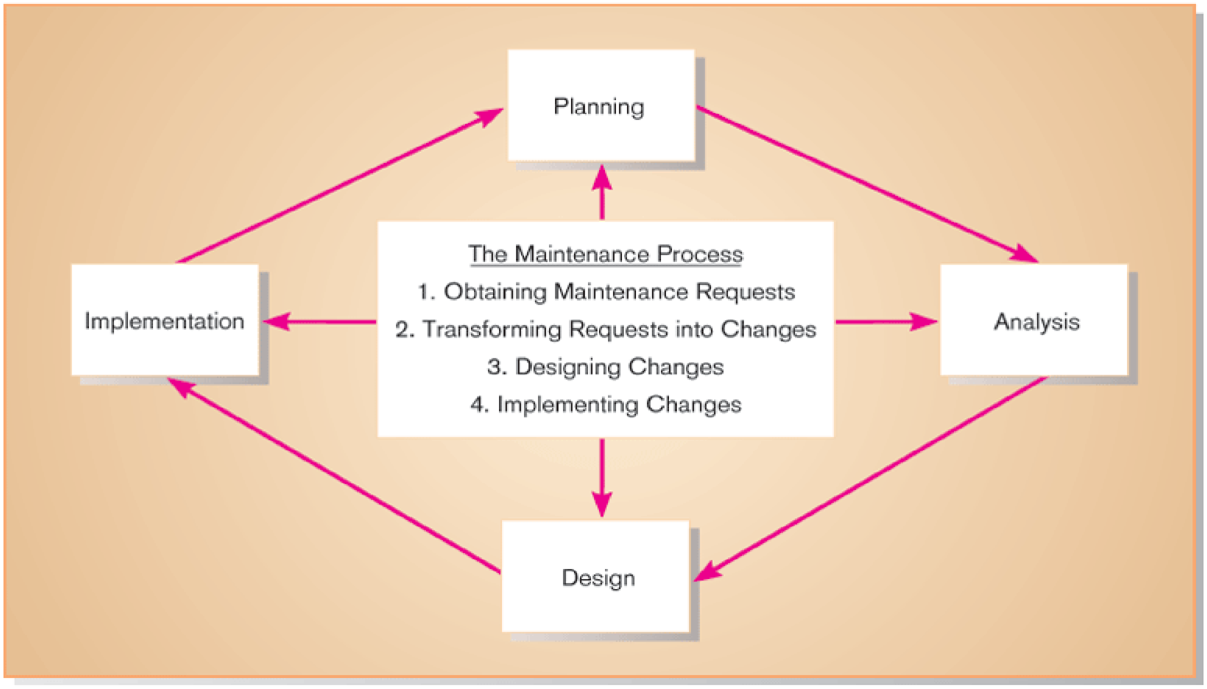


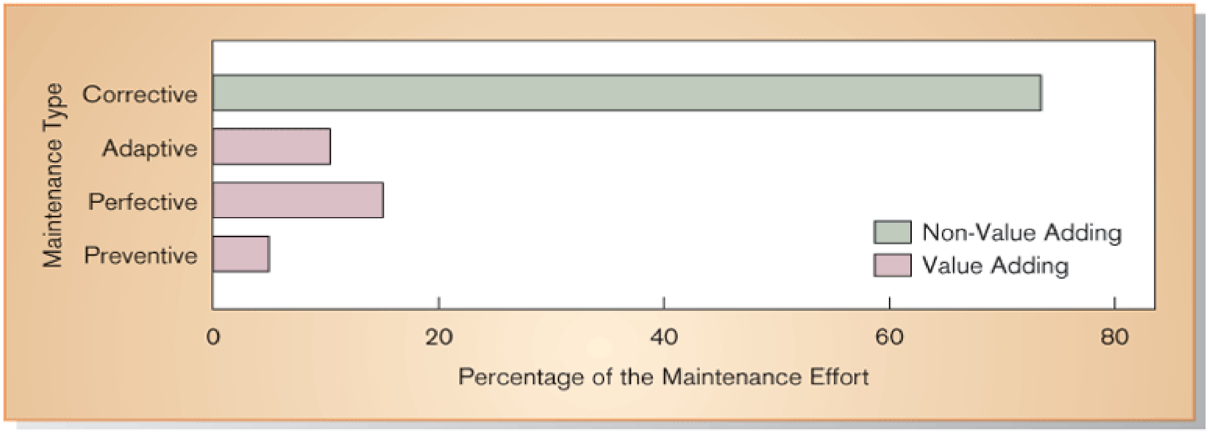
Figure 16-3 Maintenance activities parallel those of the SDLC.



6.8 Types of System Maintenance

- Corrective maintenance
  - Changes made to a system to **repair fault** 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**

Figure 16-4 Types of maintenance (Adapted from Andrews and Leventhal, 1993)



Source: Andrews D. C., and N. S. Leventhal. 1993. Fusion: Integrating IE, CASE, JAD: A Handbook for Reengineering the Systems Organization. Upper Saddle River, NJ: Prentice Hall.)

By far, most maintenance is corrective, and therefore urgent and non-value adding.

6.9 The Cost of Maintenance

- Many organizations allocate eighty percent of information systems budget to maintenance
- Factors that influence system maintainability:
  - Latent defects
  - Number of customers for a given system

  - Quality of system documentation
  - Maintenance personnel
  - Tools
  - Well-structured programs

6.10 Conducting System Maintenance: Managing Maintenance

- Industry-wide, the number of people working in maintenance has surpassed number working in development.
- 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.

6.11 Conducting System Maintenance Measures of Effectiveness

- Important factors to consider
  - 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

6.12 Selecting hardware and software

- The main steps of the selection process are listed below:

1. Requirement Analysis:

System configuration requirements are clearly identified and a decision to acquire the system is taken in this step.

2. **Preparation of tender specifications:**

After studying the feasibility and deciding upon the configuration, tender documents are prepared for the benefit of vendors to clarify the details of various specifications

3. Inviting tenders:

After the preparation of tender specifications, tenders are invited.

4. Short listing:

This step involves the following activities.

i. All tendered bids are opened on a pre-defined date and time.

ii. Deviations from the specifications, if any, in each bid are noted.

iii. A comparative summery is prepared against the list of tendered technical features.

5. Detailed evaluation of short listed vendors:

This step primarily involves getting any finer technical clarifications.

6. Negotiation and procurement decision:

Because of the extensive competition, computer system vendors may offer significant concessions. Negotiations are held to maximize these concessions.

7. Delivery and installation:

In this step, the vendor delivers the hardware/ software to the buyer’s organization. If conforms to the specifications, the vendor installs the system in the premises of the organization.

8. Post-installation review:

After the system is installed, a system evaluation is made to determine how closely the new system conforms to the plan. system specifications and user requirements are audited.

6.13 Eight Reasons for Late Software Delivery

- An **unrealistic deadline** established by someone outside the software engineering group .
- **Changing customer requirements** that are not reflected in schedule changes
- An **honest underestimate** of the amount of **effort** and /or the number of **resources** that will be required to do the job

- Predictable and/or unpredictable **risks** that were not considered when the project started.
- **Technical difficulties** that could not have been foreseen in advance
- **Human difficulties** that could not have been foreseen in advance
- **Miscommunication** among project staff that results in delays
- A **failure by project management** to recognize that the project is falling behind schedule and a lack of action to correct the problem

6.14 Basic Principles for Project Scheduling

- **Compartmentalization**
  - The project must be compartmentalized into a number of manageable activities, actions, and tasks.
- **Interdependency**
  - The interdependency of each compartmentalized activity, action, or task must be determined
  - Some tasks must occur in sequence while others can occur in parallel
  - Some actions or activities cannot commence until the work product produced by another is available
- **Time allocation**
  - Each task must be assigned a start date and a completion date that are a function of the interdependencies
  - Start and stop dates are also established based on whether work will be conducted on a full-time or part-time basis
- **Effort validation**
  - Every project has a defined number of people on the team
  - As time allocation occurs, the project manager must ensure that no more than the allocated number of people have been scheduled at any given time
- **Defined responsibilities**
  - Every task that is scheduled should be assigned to a specific team member
- **Defined outcomes**

  - Every task that is scheduled should have a defined outcome for software projects such as a work product or part of a work product
  - Work products are often combined in deliverables

- **Defined milestones**
  - Every task or group of tasks should be associated with a project milestone

Assignment VI

1. What do you mean by implementation plan? Explain.
2. What is the role of software application testing?
3. What is testing? Explain in detail.
4. What do you mean by quality assurance.? Explain with example.
5. What are main deliverables from testing and installation.
6. What are two important things to remember about testing systems?
7. Differentiate between system documentation and user documentation?
8. Why we need documentation in system development process. Explain types of documentation.
9. What are the different types of maintenance?
10. Explain the process of system implementation and maintenance.
11. Comparison between corrective, adaptive, perfective and preventive maintenance.
12. Explain the steps in the maintenance process and contrast them with the phase of the systems development life cycle.
13. What managerial issues can be better understood by measuring maintenance effectiveness? Explain.
14. Compare the different conversion methods with each other.
15. Explain the factors that influence the cost of maintenance.