



# 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.

# 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.



# 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.



# 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 (manual)

- **Inspection:** a testing technique in which participants examine program code for predictable language-specific errors.
- **Desk checking:** a testing technique in which the program code is sequentially executed manually by the reviewer.



## Seven Different Types of Tests (Cont.) (Automated)

- ***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.

# Test case description form

Pine Valley Furniture Company  
*Test Case Description*

Test Case Number:

Date:

Test Case Description:

Program Name:

Testing State:

Test Case Prepared By:

Test Administrator:

Description of Test Data:

Expected Results:

Actual Results:



# The Testing Process (Cont.)

- Test cases 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.



# Combining Coding and Testing

- 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.



# 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

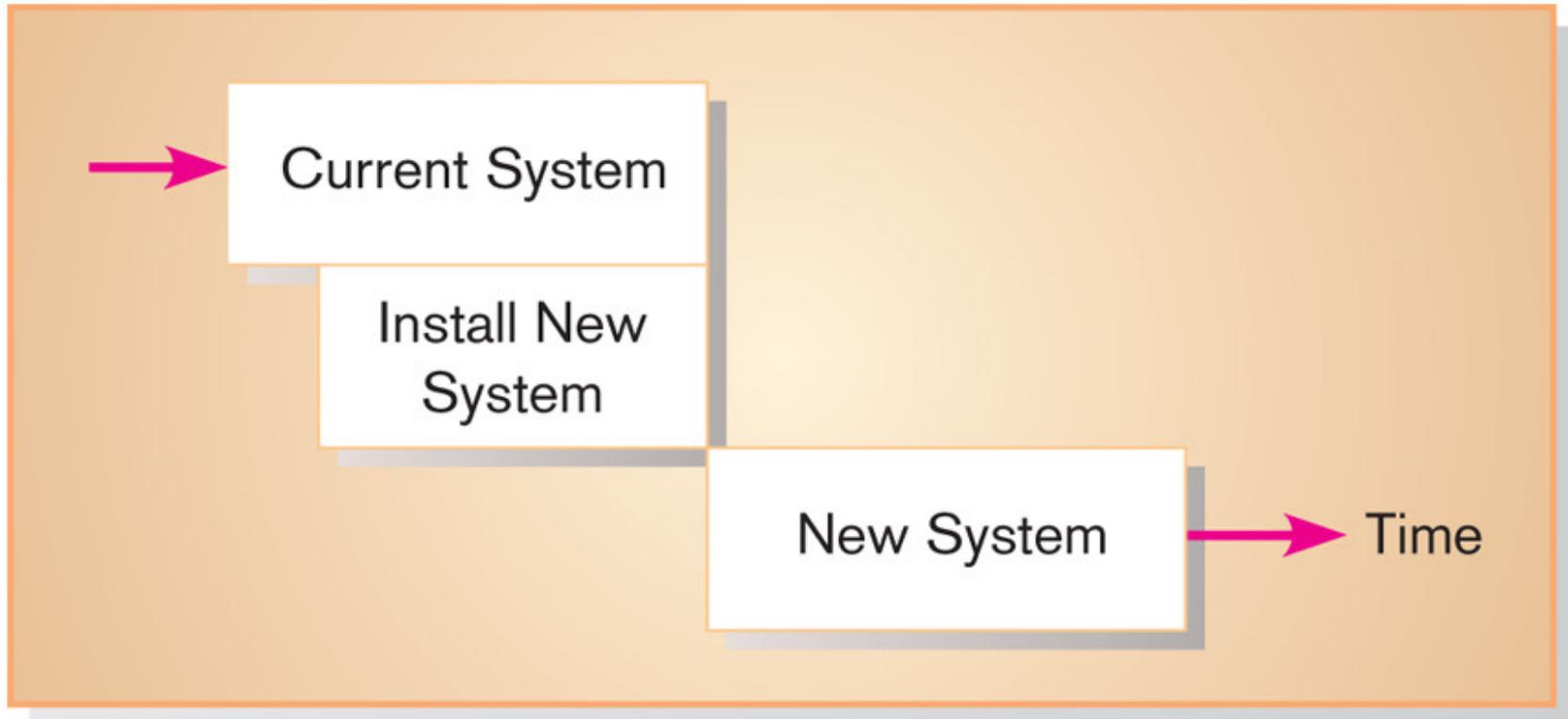
- Installation: the organizational process of changing over from the current information system to a new one.
- Four installation strategies:
  - Direct Installation (direct cut over).
  - Parallel Installation.
  - Single-location installation (pilot).
  - Phased Installation.



# 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.

# Direct installation

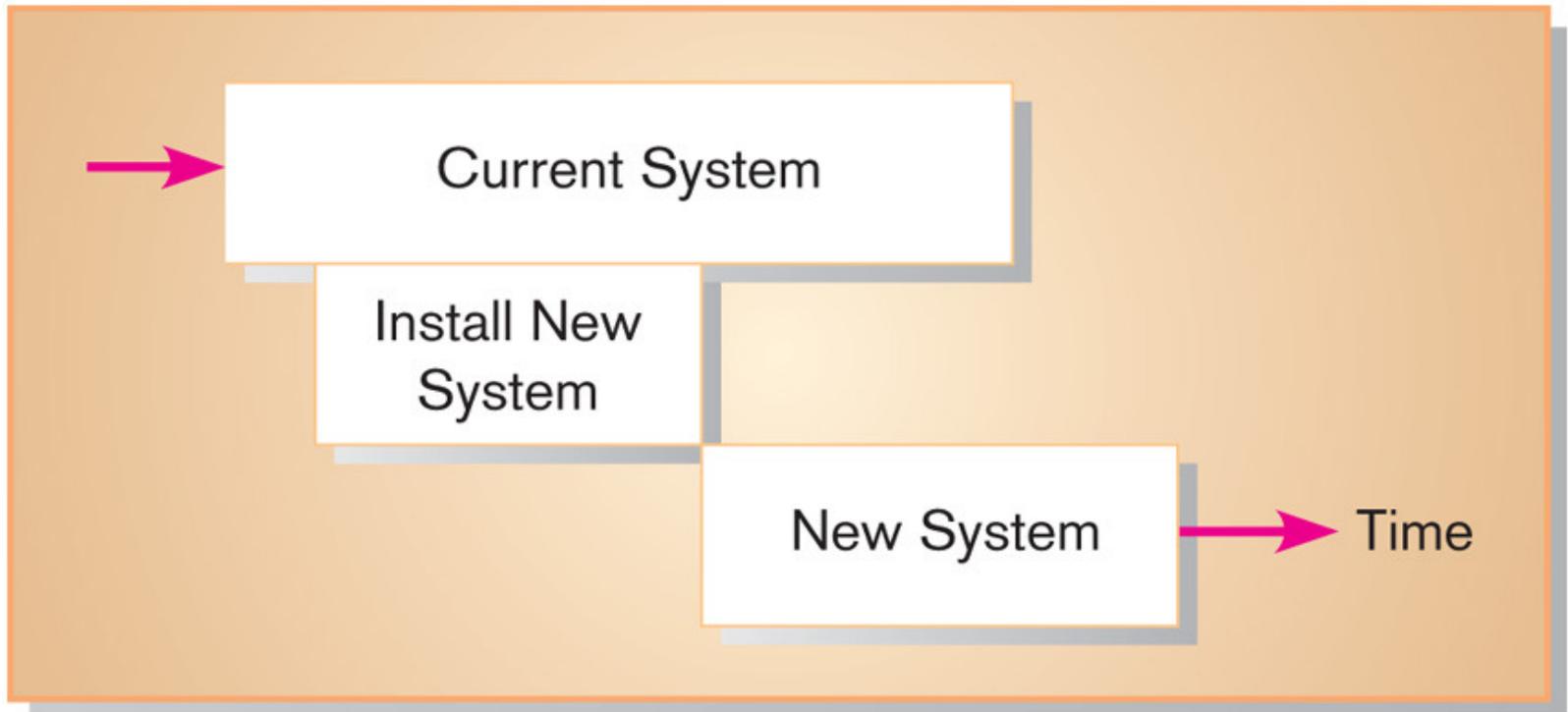




# Parallel Installation

- **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.

# Parallel installation

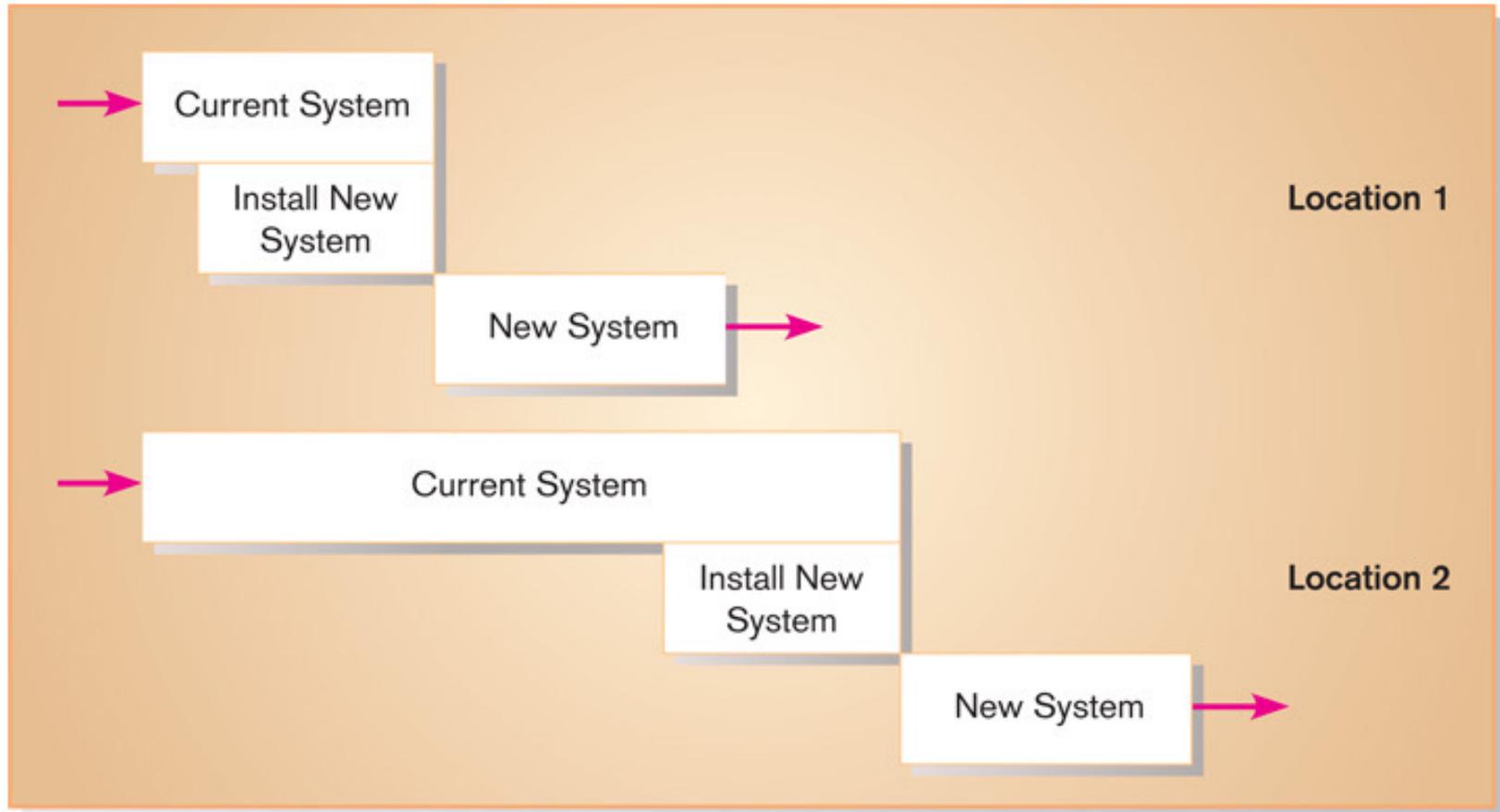




# Single-Location Installation

- **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*.

# Pilot installation

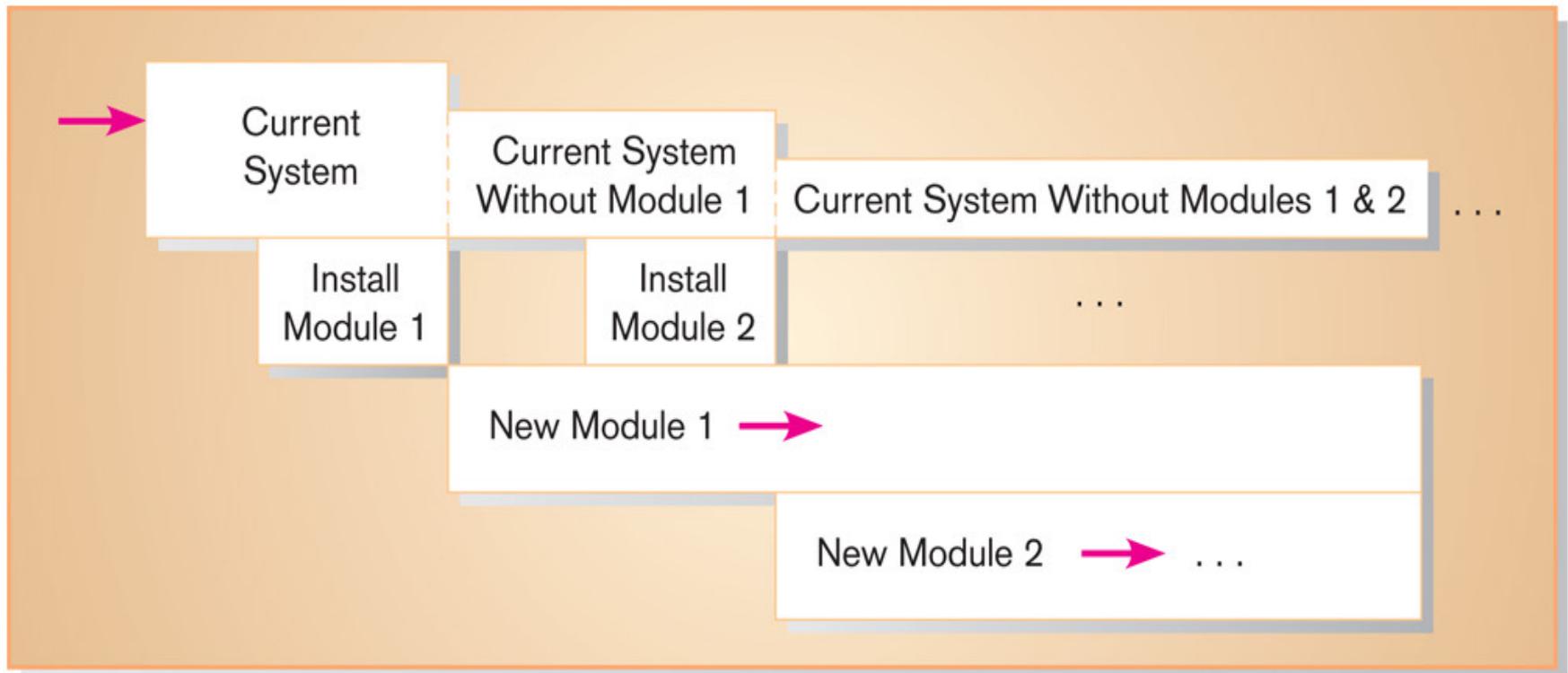




# Phased Installation

- **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.

# Phased installation





# 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.



# Preparing User Documentation

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



# 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 (how to enter a form).
  - General computer concepts (files, data bases).
  - Information system concepts (batch processing).
  - Organizational concepts (FIFO inv accounting).
  - System management (asking for changes in system).
  - System installation (how to install the system).



# Types of training methods

- Resident expert
- Traditional instructor led classroom training
- E-learning – distance learning
- Combination of the above (blended/hybrid)
- Software help components
- External sources, such as 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.



# 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.



# The Process of Maintaining Information Systems (Cont.)

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



# Deliverables and Outcome

- The maintenance phase of the SDLC is basically of subset of the activities of the entire development process.



# Deliverables and Outcome (Cont.)

- 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.



# Types of System Maintenance

- **Maintenance:** changes made to a system to fix or enhance its functionality.
- **Types:** corrective, adaptive, perfective, and preventive



# Types of System Maintenance (Cont.)

- **Corrective maintenance:** changes made to a system to repair flaws in its design, coding, or implementation.



# Types of System Maintenance (Cont.)

- **Adaptive maintenance:**  
changes made to a system to evolve its functionality to changing business needs or technologies.



# Types of System Maintenance (Cont.)

- **Perfective maintenance:**  
changes made to a system to add new features or to improve performance.



# Types of System Maintenance (Cont.)

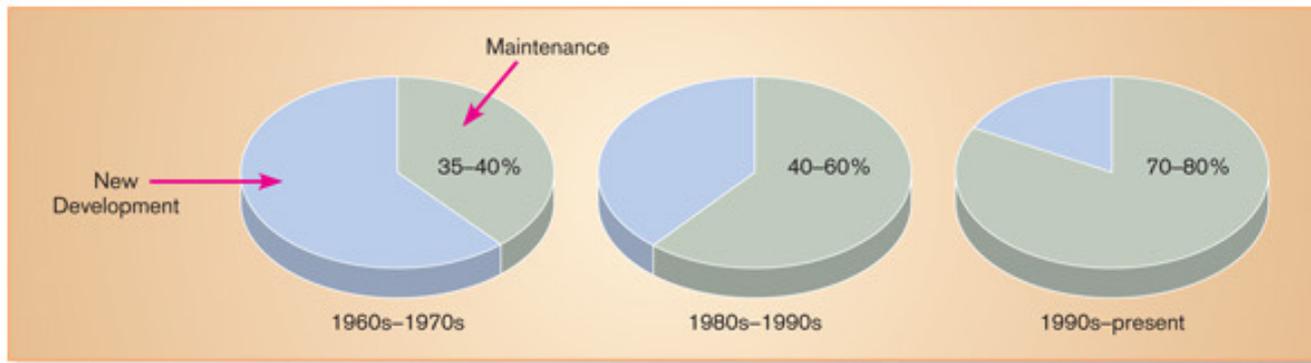
- **Preventive maintenance:**  
changes made to a system to  
avoid possible future problems.



# The Cost of Maintenance

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

# The Cost of Maintenance (Cont.)



New development versus maintenance as a percent of software budget

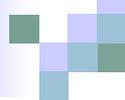
# The Cost of Maintenance (Cont.)

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



# Managing Maintenance Personnel

- Number of people working in maintenance has surpassed number working in development.
- Maintenance work is often viewed negatively by IS personnel.



# Managing Maintenance Personnel (Cont.)

- Organizations often rotate personnel in and out of maintenance roles in order to lessen negative feelings about maintenance.
- Organizations have historically have 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.)

| <b>Maintenance Organization Type</b> | <b>Advantages</b>                         | <b>Disadvantages</b>                                  |
|--------------------------------------|---|---|
| <b>Separate</b>                      | Improved system and documentation quality | Ignorance of critical undocumented information        |
| <b>Combined</b>                      | Maintenance group knows all about system  | Less emphasis on good documentation                   |
| <b>Functional</b>                    | 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.

# Measuring Maintenance Effectiveness (Cont.)

- **Mean time between failures (MTBF):** a measurement of error occurrences that can be tracked over time to indicate the quality of a system.



# Controlling Maintenance Requests

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



# Configuration Management

- **Configuration management:** the process of ensuring that only authorized changes are made to the system.

# Configuration Management (Cont.)

- **Baseline modules:** software modules that have been tested, documented, and approved to be included in the most recently created version of a system.



# Configuration Management

- **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.



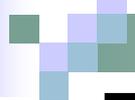
# Role of CASE and Automated Development Tools in Maintenance

- Traditional systems development
  - Emphasis 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.

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

## ■ 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.

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

- **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.