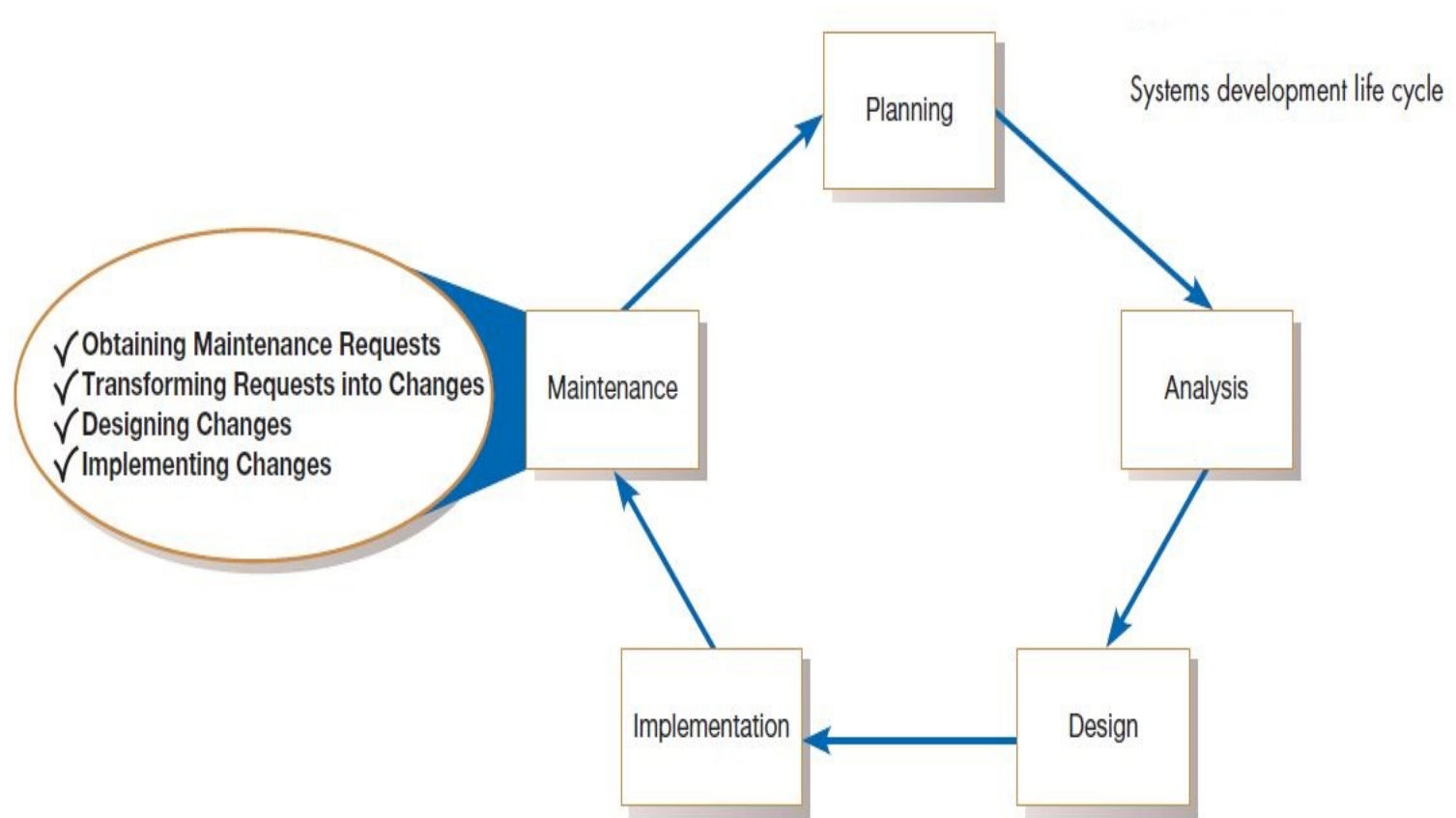


# Unit 5.2 Maintaining Information Systems


Introduction, Maintaining Information Systems(Process, Deliverables and Outcomes), Conducting Systems Maintenance (Types and Cost of Maintenance, Managing Maintenance, Using Automated Tools)

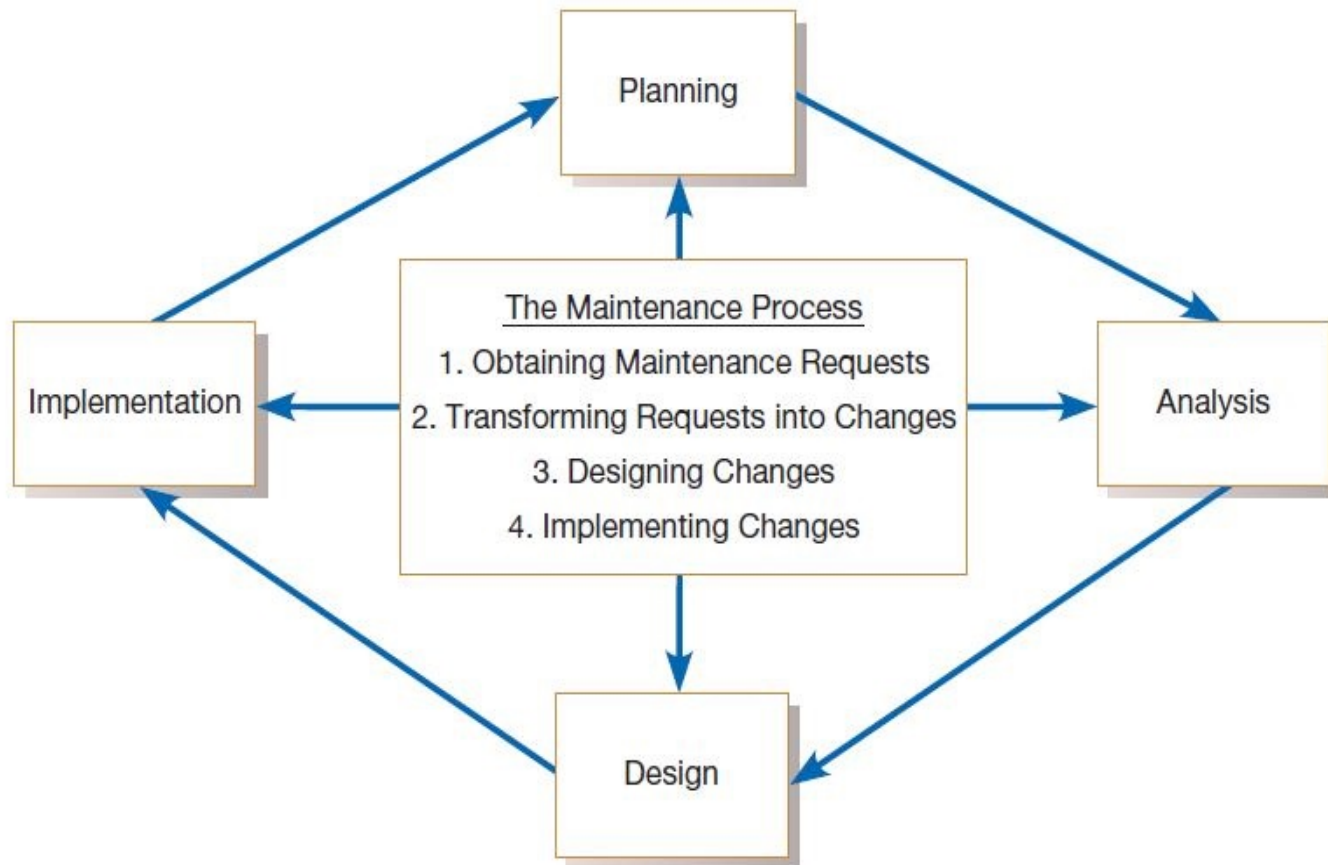


# System Maintenance

- In this chapter, we discuss systems maintenance, the largest systems development expenditure for many organizations. In fact, more programmers today work on maintenance activities than work on new development. Your first job after graduation may very well be as a maintenance programmer/analyst. This disproportionate (too large or too small in comparison to something else ) distribution of maintenance programmers is interesting because software does not wear out in a physical manner as do buildings and machines. There is no single reason why software is maintained; however, most reasons relate to a desire to evolve system functionality in order to overcome internal processing errors or to better support changing business needs. Thus, maintenance is a fact of life for most systems. This means that maintenance can begin soon after the system is installed.

- **Maintaining Information Systems:** Once an information system is installed, the system is essentially in the maintenance phase of the systems development life cycle (SDLC). When a system is in the maintenance phase, some person within the systems development group is responsible for collecting maintenance requests from system users and other interested parties, such as system auditors, data center and network management staff, and data analysts. Once collected, each request is analyzed to better understand how it will alter the system and what business benefits and necessities will result from such a change. If the change request is approved, a system change is designed and then implemented. As with the initial development of the system, implemented changes are formally reviewed and tested before installation into operational systems.
- The maintenance phase is the last phase of the SDLC. It is here that the SDLC becomes a cycle, with the last activity leading back to the first. This means that the process of maintaining an information system is the process of returning to the beginning of the SDLC and repeating development steps until the change is implemented.

- 
- Four major activities occur within maintenance:
    - Obtaining maintenance requests
    - Transforming requests into changes
    - Designing changes
    - Implementing changes
  - Obtaining maintenance requests requires that a formal process be established whereby users can submit system change requests. personnel.




# Conducting Systems Maintenance

- A significant within organizations does not go to the development of new systems but to the maintenance of existing systems. We will describe various types of maintenance, factors influencing the complexity and cost of maintenance, and alternatives for managing maintenance.

# Types of Maintenance

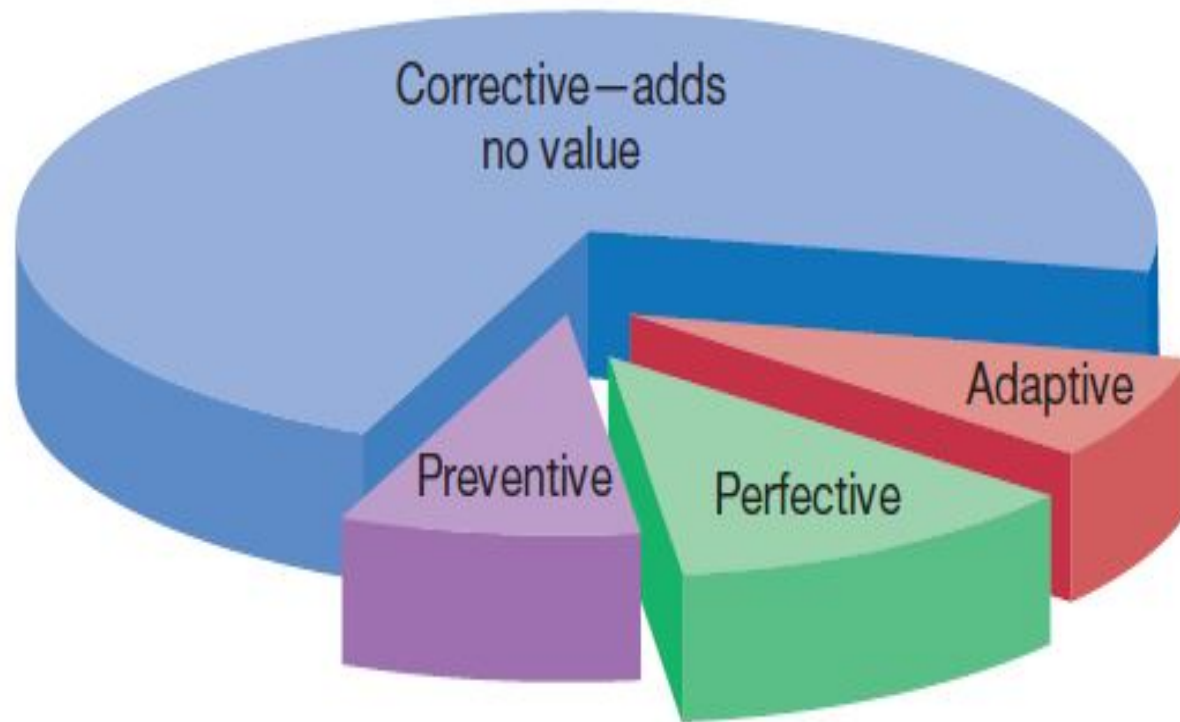
- By maintenance, we mean the fixing or enhancing of an information
- system. Corrective maintenance refers to changes made to repair defects in the design, coding, or implementation of the system. For example, if you had recently purchased a new home, corrective maintenance would involve repairs made to things that had never worked as designed, such as a faulty electrical outlet or a misaligned door. Most corrective maintenance problems surface soon after installation. When corrective maintenance problems surface, they are typically urgent and need to be resolved to curtail possible interruptions in normal business activities. Of all types of maintenance, corrective accounts for as much as 75 percent of all maintenance activity (Andrews and Leventhal, 1993; Pressman, 2005).



- 
- This is unfortunate because corrective maintenance adds little or no value to the organization; it simply focuses on removing defects from an existing system without adding new functionality.
  - Adaptive maintenance involves making changes to an information system to evolve its functionality to changing business needs or to migrate it to a different operating environment. Within a home, adaptive maintenance might be adding storm windows to improve the cooling performance of an air conditioner. Adaptive maintenance is usually less urgent than corrective maintenance because business and technical changes typically occur over some period of time. Contrary to corrective maintenance, adaptive maintenance is generally a small part of an organization's maintenance effort, but it adds value to the organization.

- Perfective maintenance involves making enhancements to improve processing performance or interface usability or to add desired, but not necessarily required, system features (bells and whistles). In our home example, perfective maintenance would be adding a new room. Many systems professionals feel that perfective maintenance is not really maintenance but rather new development.
- Preventive maintenance involves changes made to a system to reduce the chance of future system failure. An example of preventive maintenance might be to increase the number of records that a system can process far beyond what is currently needed or to generalize how a system sends report information to a printer so that the system can easily adapt to changes in printer technology. In our home example, preventive maintenance could be painting the exterior to better protect the home from severe weather conditions. As with adaptive maintenance, both perfective and preventive maintenance are typically a much lower priority than corrective maintenance. Over the life of a system, corrective maintenance is most likely to occur after initial system installation or after major system changes. This means that adaptive, perfective, and preventive maintenance activities can lead to corrective maintenance activities if not carefully designed and implemented.

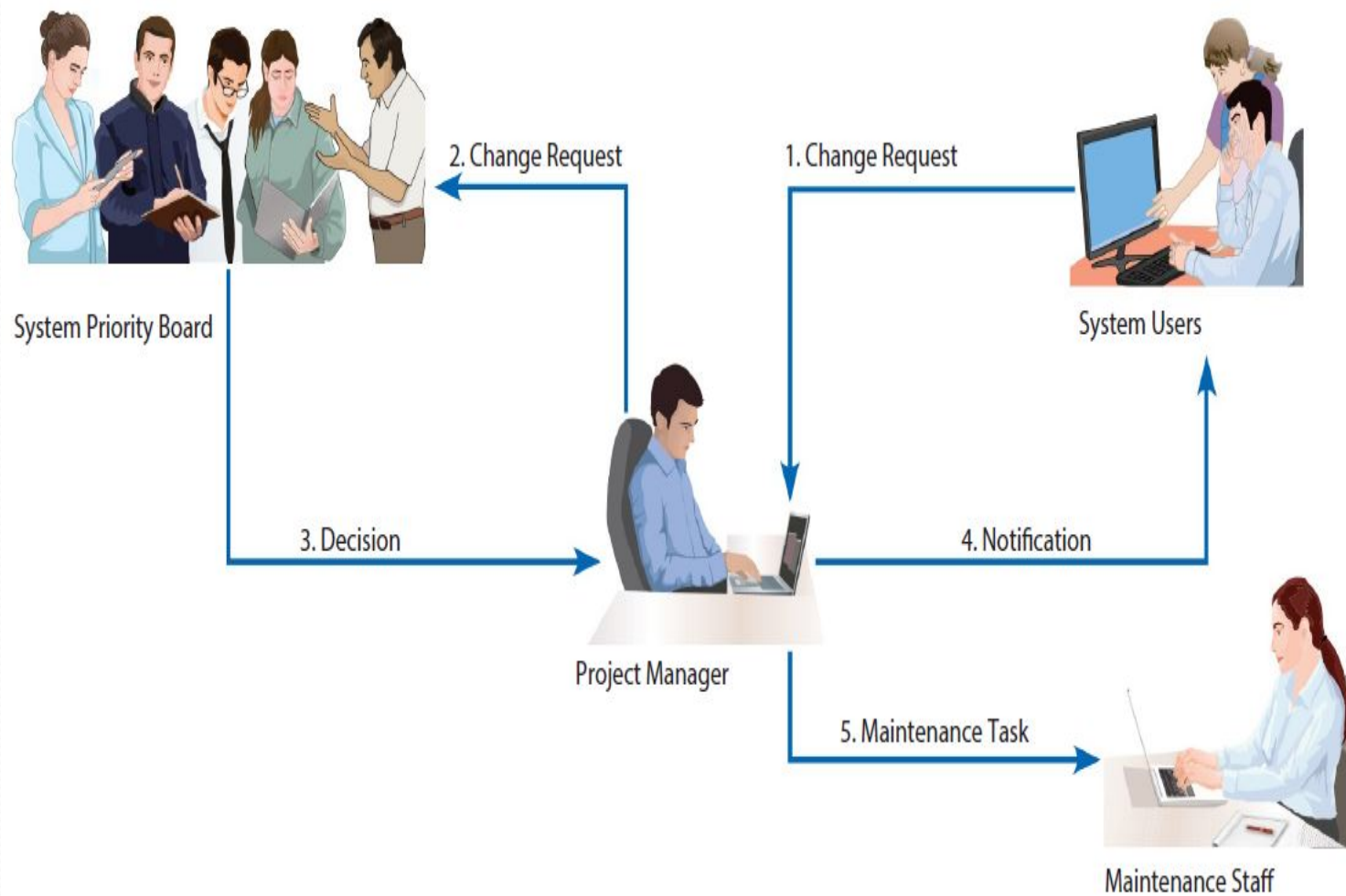
## Maintenance Type



- Information systems maintenance costs are a significant expenditure. For some organizations, as much as 60 to 80 percent of their information systems budget is allocated to maintenance activities (Kaplan, 2002). These huge maintenance costs are due to the fact that many organizations have accumulated more and more older so-called legacy systems that require more and more maintenance. More maintenance means more maintenance work for programmers.
- For systems developed in-house, on average, 52 percent of a company's
- programmers are assigned to maintain existing software (Lytton, 2001). In situations where a company has not developed its systems in-house but instead has licensed software, as in the case of ERP systems, maintenance costs remain high. The standard cost of maintenance for most ERP vendors is 22 percent annually (Nash, 2010).

# Managing Maintenance

- As maintenance activities consume more and more of the systems development budget, maintenance management has become increasingly important. Today, far more programmers worldwide are working on maintenance than on new development. In other words, maintenance is the largest segment of programming personnel, and this implies the need for careful management.
- Managing Maintenance Personnel: One concern with managing maintenance relates to personnel management. Historically, many organizations had a “maintenance group” that was separate from the “development group.” With the increased number of maintenance personnel, the development of formal methodologies and tools, changing organizational forms, end-user computing, and the widespread use of very high-level languages for the development of some systems, organizations have rethought the organization of maintenance and development personnel.





# Configuration Management

- A final aspect of managing maintenance is configuration management, which is the process of ensuring that only authorized changes are made to a system. Once a system has been implemented and installed, the programming code used to construct the system represents the baseline modules of the system. The baseline modules are the software modules for the most recent version of a system whereby each module has passed the organization's quality assurance process and documentation standards. A system librarian controls the checking out and checking in of the baseline source code modules. If maintenance personnel are assigned to make changes to a system, they must first check out a copy of the baseline system modules—no one is allowed to directly modify the baseline modules. Only those modules that have been tested and have gone through a formal check-in process can reside in the library. Before any code can be checked back in to the librarian, the code must pass the quality control procedures, testing, and documentation standards established by the organization.