CHAPTER 17

ACQUIRING AND IMPLEMENTING ACCOUNTING INFORMATION SYSTEMS

LEARNING OBJECTIVES

AFTER READING THIS CHAPTER, YOU SHOULD BE ABLE TO:

- DESCRIBE THE SYSTEMS ACQUISITION/DEVELOPMENT PROCESS AND ITS MAJOR PHASES AND STEPS.
- UNDERSTAND THE DIFFERENCES IN THE PROCESS FOR PURCHASED VERSUS INTERNALLY DEVELOPED SYSTEMS.
- UNDERSTAND THE NATURE AND IMPORTANCE OF THE ACCOUNTANT'S INVOLVEMENT IN SYSTEMS DEVELOPMENT AND ACQUISITION PROJECTS.

The Standish Group, which has been reporting on systems success and failure rates since 1994, recently estimated that only 28 percent of attempted IT projects are successful—meaning that 72 percent suffer from either being stopped before completion, or if completed, are materially late, over budget, and/or missing required features. This means that if you work on a significant IT project in any capacity, almost three-fourths of the time you will end up failing! With such a serious overall failure rate, you are probably wondering what you can do to ensure the success of your projects. Standish's chairman, Jim Johnson, says the number one reason for project failure is lack of user involvement in the new system. This means that failure is not usually a technology problem—it is a people problem.[1]

The International Federation of Accountants (IFAC) in its International Education Guideline No. 11 states that in addition to being a systems user, as a professional accountant, you should expect to serve as a designer, evaluator, and/or manager of information systems. As you study this chapter, try to imagine yourself in each of these roles. As a designer, you will need to know what systems development tasks need to be completed—and when and how—for the IT project to be successful. For example, you will need to work with users to obtain systems specifications and for help in testing the system before implementation. As an evaluator and/or manager, you will need to determine if the systems outputs are correct and if the system consumes resources efficiently. Knowing how the system was developed and installed will help you zero in on problem areas. From each perspective, remember knowledge of the systems

[1] Frank Hayes, "Chaos is Back," Computerworld.com, (November 4, 2004).

development process can help ensure your IT project falls within that successful 28 percent. Otherwise, you may be trying to explain why your project is in the other 72 percent.

Synopsis

Driven by a host of sometimes complementary but often conflicting demands, the selection of accounting information systems (AIS) and other information systems throughout organizations requires considerable forethought. There are constraints, tradeoffs, and tough decisions at every turn in the road. For instance, users are demanding more robust services from their information systems, customers are insisting on quicker response times and more flexible interfaces, organizations are becoming more internally interconnected, trading partners are moving toward fully integrated supply chains, competitive business pressures are bearing down on today's organizations, and new opportunities provided by advanced information technologies are cropping up everywhere. As a result, a modern organization must adapt its information systems quickly and continuously. And because the tentacles of AIS reach deep into information systems throughout the organization, AIS must also continuously adjust to rapidly changing conditions.

Systems development comprises the steps undertaken to create, modify, or maintain an organization's information system. The systems development process is made up of four primary phases: systems analysis, design, implementation, and operation. The **systems development life cycle** is the term often used to describe the progression through the phases of the systems development process, from birth through implementation to ongoing use.

During the analysis phase, an organization decides whether it will develop the IT system in-house (i.e., make) or acquire the system from a third party (i.e., buy). In either case, the organization must perform a structured systems analysis, or needs analysis, to determine the features that the new system must have. The procedures in these steps must be performed exceptionally well to have any chance of developing systems that meet user needs because such needs are determined during systems analysis. Without a well-understood and documented target (i.e., user requirements), there is no chance of achieving a successful development process.

The second phase, systems design, takes the needs determined during the analysis process and defines the details of the system—to be developed in-house or acquired—that will meet those needs. It is also in this phase that the hardware and infrastructure to support the system are selected.

After the design is complete, the next step is the implementation of the new system. With a purchased system, the implementation includes configuring the software, as well as converting data, training users, and performing other practical tasks. The implementation process for an in-house developed system additionally includes the programming and testing of code.

The final phase of the SDLC is the systems operation. This phase includes the post-implementation review, where the project is evaluated. This phase is important because successes and failures in the current project can provide valuable lessons for future projects.

At different points in your professional career, you will probably have many roles within the systems development process. You will be a system user or business process owner articulating your needs, or you will be a member of the development team that must determine and document such needs. You may be in a position to decide what system to buy. You may manage an implementation or find problems that must be resolved in a new system. This chapter exposes you to the development process to give you an understanding of that process. That understanding should help ensure the success of your projects.

Introduction

The acquisition or in-house development of an information system is a major event in the life of an organization, and both require many of the same techniques of analysis, design, implementation, and systems operations. The major difference is how the system is implemented: with a prepackaged solution, internally developed system, or some combination of the two. On the surface, it seems intuitively appealing to build your own IS that is specifically tailored to your organization's needs. But in reality, this is a difficult undertaking. However, in some circumstances, in-house development is the only reasonable solution. In such cases, organizations should take precautionary measures aimed at reducing the risk of failure to an acceptably low level by following a structured approach to managing the project at every step of the process.

In this chapter, you will study a structured approach to developing information systems called the **systems development life cycle (SDLC) methodology**, which is a formalized, standardized, documented set of activities used to manage a systems development project.

Acquiring an AIS from External Parties

Organizations not wanting to or unable to develop software in-house may purchase, rent, or lease a commercially available software package. Such software can be acquired from computer manufacturers, software vendors, mail-order houses and retail stores (for microcomputer software), turnkey system suppliers, service bureaus, systems integrators, outsourcing firms, and application service providers (ASP). ASPs are an important segment of the outsourcing market. Technology Summary 17.1 (pg. 600) describes ASPs and summarizes some ASP services.

An organization should consider the financial implications of the decision to develop (make) versus buy software. Because software vendors can allocate development costs across many products and across multiple copies of each product, the prices they charge to recover these costs are usually substantially less than the organization would pay to develop the package in-house. Generally, software developed in-house can cost up to 10 times more than purchased software. Additionally, the annual maintenance of in-house software is typically 50 percent of the development cost, whereas the annual maintenance for purchased software normally costs only 25 percent of the purchase price.

To increase the potential market for a software package, vendors develop packages for a wide audience. This strategy leads to products that seldom possess characteristics exactly matching any particular organization's requirements. Organizations not satisfied with these generic packages can contract with a vendor to modify one of the vendor's existing software packages or develop a custom-tailored software package written specifically to meet the organization's unique requirements.

The bottom line is that when a *suitable* standard package exists, buy it rather than try to reinvent it in-house. Notice the emphasis on *suitable*. By using a standard package, you are gaining the benefit of a system that incorporates the best practices of other organizations—potentially providing you a way to improve your operations. Other

TECHNOLOGY SUMMARY 17.1

APPLICATION SERVICE PROVIDER (ASP)

An **application service provider (ASP)** offers an outsourcing mechanism whereby it hosts, manages, and provides access to application software and hardware over the Internet to multiple customers. The fee is really a rental based on usage. Some ASPs provide service for free, obtaining revenue from advertising and the sale of other services. ASPs relieve the organization of the burden of developing or buying and installing software and hardware. Because ASPs are accessed over the Internet, a user needs only a Web browser and a basic PC or other method of accessing the Internet (such as a browser-enabled mobile phone) to obtain the ASP service.

When using an ASP, a user obtains constantly updated software. The user does not need the technical resources to install or support the application. ASPs are a good choice for applications where the user does not want to purchase and maintain the application or simply would like access from any Internet access point. *QuickBooks Online Edition* and *TurboTax*, both from Intuit, Inc., are two consumer products that traditionally have been sold as ''purchase and install'' but are now available as an ASP service.

considerations when choosing to develop a system in-house include the organization's internal resources (personnel, capital) and available vendor support.

Managing the Systems Development Process

The systems development objectives are the following:

- To ensure the information system satisfies an organization's informational and operational needs (product-oriented objective)
- To develop/acquire an information system in an efficient and effective manner (process-oriented objective)

CONTROLS

The key to achieving the first objective is to control the process. Apparently, that is not an easy chore or more organizations would be successful at doing it. We can understand the complexity of the systems development process by comparing it to a construction project. Assume you are in charge of the construction of an industrial park. What problems and questions might you encounter? For instance, you might want to know the following: "How much of the project is whose responsibility?" "Who should handle legal and financial matters?" "Who obtains the building permits?" "Who is responsible for contacting the tenants/buyers to determine special needs?" The project's size and duration cause another set of problems. How will you coordinate the work of the carpenters, masons, electricians, and plumbers? How will you see that a tenant's special needs are incorporated into the specifications and then into the actual construction?

Information systems developers encounter similar problems. Given such problems, they have concluded that systems development must be carefully controlled and managed by following good project management principles and the organization's quality assurance framework, including its *systems development life cycle (SDLC) methodology*.

Systems Development Methodology

To effectively manage a development project, systems standardized methodology should be used. In this chapter, we use the *systems development life cycle (SDLC) methodology*. It is appropriate when information systems are developed, acquired, or maintained. Exhibit 17.1 describes characteristics of a systems development methodology.

EXHIBIT 17.1 Characteristics of a Systems Development Methodology

- The project is divided into a number of identifiable processes, each having a starting and ending point. Each process comprises several activities, one or more *deliverables*, and several *management control points*. The division of the project into these small, manageable steps facilitates both project planning and project control.
- Specific reports and other documentation, called **deliverables**, must be produced periodically during systems development to make development personnel accountable for faithful execution of systems development tasks. An organization monitors the development process by reviewing the deliverables that are prepared at the end of each key step. Many organizations rely on this documentation for training new employees; it also provides users with a reference while they are operating the system.
- Users, managers, and auditors are required to participate in the project. These people generally provide approvals, often called *signoffs*, at preestablished management control points. **Signoffs** signify approval of the development process and the system being developed.
- The system must be tested thoroughly prior to implementation to ensure that it meets users' needs.
- A training plan is developed for those who will operate and use the new system.
- Formal program change controls (see Chapter 8) are established to preclude unauthorized changes to computer programs.
- A post-implementation review of all developed systems must be performed to assess the effectiveness and efficiency of the new system and of the development process.

Following such a methodology should ensure that development efforts are efficient and consistently lead to information systems that meet organizational needs. These guidelines should be followed whether an organization is going to acquire an AIS or develop it in-house.

Figure 17.1 (pg. 602) presents the systems development life cycle. The right side of the figure depicts the four development phases: systems analysis, systems design, systems implementation, and systems operation. The bubbles in Figure 17.1 identify the seven development steps undertaken to complete the four phases of development. Arrows flowing into each bubble represent the inputs needed to perform that step, whereas outward-flowing arrows represent the product of a step. A development process may not necessarily proceed sequentially through these steps. Rather, steps may be performed iteratively, out of sequence, or not at all. Table 17.1 (pg. 603) lists the key purposes and tasks associated with the seven development steps (bubbles) shown in Figure 17.1. You should take some time now to review both the table and the figure. First, we will focus on the initial phase of the SDLC—systems analysis.

Systems Survey

The **systems survey**, often called a **feasibility study** or **preliminary feasibility study**, is a set of procedures conducted to determine the practicability of a potential systems development project and to prepare a systems development plan for projects considered feasible. Refer to Figure 17.1 to see the systems survey's place in the SDLC (bubble 1.0), its inputs (request for systems development and miscellaneous environmental information), and its output (approved feasibility document). An organization conducts a systems survey to determine, as quickly and as inexpensively as possible, whether it is worthwhile to proceed with subsequent development steps.

So that we can understand the systems survey process, we will compare systems development to building an industrial park. The architect's role for the industrial park is

FIGURE 17.1 Systems Development Life Cycle

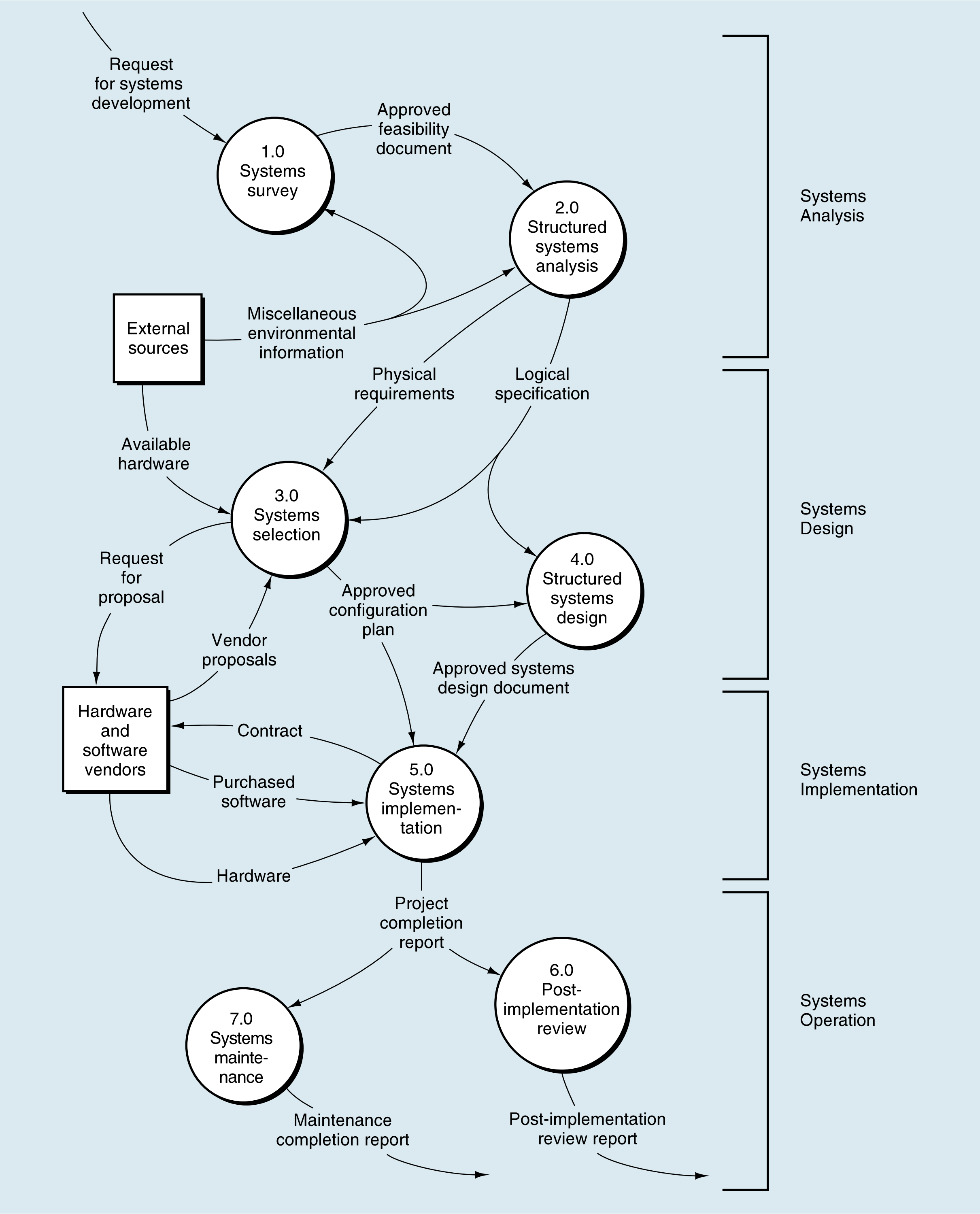

Request for systems development
1.0 Systems survey
Approved feasibility document
2.0 Structured systems analysis
External sources
Miscellaneous environmental information
Physical requirements
Logical specification
Available hardware
3.0 Systems selection
Request for proposal
Vendor proposals
Approved configuration plan
4.0 Structured systems design
Approved systems design document
Hardware and software vendors
Contract
Purchased software
Hardware
5.0 Systems implemen-tation
Project completion report
6.0 Post-implementation review
7.0 Systems mainte-nance
Maintenance completion report
Post-implementation review report
Systems Analysis
Systems Design
Systems Implementation
Systems Operation

TABLE 17.1 Information Systems Development Phases, Purposes, and Tasks*

Phase	Purpose	Tasks
Analysis (bubbles 1.0 and 2.0)	Develop specifications for a new or revised system.	Study the problem and the problem's environment. Propose alternative problem solutions.
Design (bubbles 3.0 and 4.0)	Develop an appropriate system manifestation.	Convert the logical solution into a physical design. Choose software and hardware. Write the detailed design specifications. Devise implementation plans, system tests, and training programs.
Implementation (bubble 5.0)	Begin using the new system.	Write, test, and debug the computer programs. Convert to new or revised system.
Operation (bubbles 6.0 and 7.0)	Use the new system.	Conduct post-implementation review. Perform systems maintenance.

*Refers to Figure 17.1

analogous to the analyst's role for systems development. In the preliminary stages of the industrial park project, the architect learns the general purpose of the industrial park (light manufacturing, warehousing, etc.). He or she also learns the approximate number of buildings and the size of each. From that information, the architect "sketches" a proposed park. From that sketch and the accompanying general specifications, estimated costs and estimated schedule, the developer decides whether or not to proceed with the proposed project. This process is similar to that followed in the systems survey, with the systems analyst assuming the architect's role and the organization's management (or the IT steering committee) replacing the developer.

The tasks required to complete the systems survey are as follows:

- *Determine the nature and the extent of each reported problem:* For instance, if the sales department reports a problem of late deliveries, we want to know whether deliveries really are late, whether there are delays within the organization, or whether our sales personnel are promising unrealistic delivery dates.
- *Determine the scope of the problem:* For example, is a reported purchasing problem confined to the purchasing process, or is there a broader problem requiring more extensive analysis?
- *Propose a course of action that might solve the problem:* For example, propose systems development of the purchasing process to correct a purchasing problem.
- *Determine the feasibility of any proposed development:* Is there a technically, operationally, and economically feasible solution to the problem? For example, does computer technology exist to solve the problem (technical feasibility), is the organization ready to accept the new solution (operational feasibility), and are the payback and internal rate of return estimates sufficient to allow the development project to proceed (economic feasibility)?

- *Devise a detailed plan for conducting the analysis step:* Determine who will conduct the analysis, who will head the project team, what tasks are required, and what the development timetable is.
- *Develop a summary plan for the entire development project:* Including, for example, when the system will be implemented, what divisions will participate, and so forth.

The systems survey tasks are important to help us determine the problem, the feasibility of solutions, and a plan for the project, but it is a *preliminary* analysis. After it is complete, we move to the *detailed* study of the proposed solution, the structured systems analysis.

Structured Systems Analysis

To help understand systems analysis, we will continue our industrial park analogy. If the developer approves the continuation of the project, the architect must conduct a *detailed* study to determine each building's specific use, required room sizes, electrical and plumbing requirements, floor load weights, private versus public areas, number of personnel who will occupy the completed buildings, technical requirements, and so on. During this detailed study, the architect develops a *functional* model of the proposed project. The detailed study by the architect is similar to systems analysis, and the logical specification (one of the outputs of the analysis step) is the model for the new system.

Systems Analysis Definition and Tasks

Structured systems analysis reflects a set of procedures conducted to generate the *specifications* for a new (or modified) information system or subsystem. Systems analysis is often called *structured systems analysis* when certain "structured" tools and techniques, such as DFDs, are used in conducting the analysis. To simplify our discussions, we will refer to structured systems analysis as simply systems analysis.

The tasks required to complete the *systems analysis* are as follows:

- *Define the problem precisely:* In the systems survey, we verified that there was a problem and determined the problem's sources. In systems analysis, we want to know and understand the problem in enough detail to solve it.
- *Devise alternative designs (solutions):* There is always more than one way to solve a problem or to design a system, and we want to develop several solutions from which to choose.
- *Choose and justify one of these alternative design solutions:* One solution must be chosen, and the choice should be justified using cost/effectiveness analysis or some other criterion, such as political or legal considerations (e.g., government reporting requirements).
- *Develop logical specifications for the selected design:* These detailed specifications are used to design the new system.
- *Develop the physical requirements for the selected design:* For example, we want to define requirements such as the amount of data to be stored, functional layouts for computer inquiry screens and reports, and processing response times, which lead to equipment requirements. There can be several alternative physical designs for each logical requirement.
- *Develop the budget for the next two systems development phases, systems design and systems implementation*: These budgets are critical in determining development costs and controlling later development activities.

The logical specifications and physical requirements become the criteria by which the user will accept the new or modified system. The better we perform systems analysis, the more likely that the system will meet user requirements and become accepted, implemented, and used effectively.

E-BUSINESS

Determination of user requirements in the analysis step can be more difficult in an e-business implementation. In such an implementation, we must determine user requirements inside *and outside* the organization. We must consider the functional needs of customers and business partners, as well any requirements for infrastructure to connect our internal systems to the outside users (e.g., customer, business partners).

The Analysis Deliverable: The Approved Systems Analysis Document

Systems analysis has three outputs: the logical specification and the physical requirements, and the budget, and schedule. All three outputs are generally part of a single **approved systems analysis document**—the final output of systems analysis. Exhibit 17.2 (pg. 606) outlines the contents of this document. The logical specification consists of all items except those designated as the budget and schedule (part 7) and the physical requirements (part 8). Study Exhibit 17.2 before proceeding. An understanding of the information collected and included in the approved systems analysis document will help you understand the steps that an analyst must perform *during* the systems analysis.

The main objective of the systems analysis phase is to define a future logical system and a future physical system. Completing the following steps will lead to the most successful development efforts:

1. Study the current physical system.
2. Document the current logical system.
3. Define the future logical system.
4. Design the future physical system.

The first step the analysis team performs is to *study and document the current physical system.*[2] The team would read and interpret existing documentation, such as DFDs and systems flowcharts, and may correct that documentation or prepare new documentation as required. The team wants to build on the information available in the *approved feasibility document* and understand completely the current system operations. Given the system's goals, what should the system be doing? Should the order entry system be supporting customer inquiries? For what reason is the system operating as it is? Why are there errors?

To document the current logical system, the analyst removes all the physical elements from the *current physical DFD* and produces a *current logical DFD*, which reflects a description of the current logical system.

The next task, to define user requirements for the new/modified system, is one of the most important and most problematic tasks in systems analysis. This task will usually also include ranking the requirements in order of importance, in the event that resources or other constraints make it impossible to satisfy all user requirements. User requirements are usually placed in three categories. The first includes critical features without which the system would be unusable. The second category includes those features that make the system easier to use but are not critical in nature. The third category includes those features that are considered "nice to have" but do not materially add to the usefulness of the system.

[2] Actually, many development projects have proceeded without documenting the current physical environment. Either there was *no existing* system, or the analysts considered documenting the current physical system too costly.

EXHIBIT 17.2 Typical Contents of an Approved Systems Analysis Document

1. **Executive summary**
   a. Project summary
   b. Summary of proposed system
   c. Summary of impact of new system
   d. Summary of cost/effectiveness and recommendations
2. **Systems analysis summary**
   Summary of the facts gathered and analysis performed
3. **User requirements for the new system**
   a. Operating requirements
   b. Information requirements
   c. Control requirements
4. **Logical specifications for the new system**
   a. Data flow diagrams and narrative describing the new logical system
   b. Summary of improvements brought about by new logical design (effectiveness)
5. **Description of future physical system**
   a. Data flow diagrams, flowcharts, and narrative describing the new physical system
   b. Summary of cost/benefit improvements brought about by the new physical system
6. **New system constraints**
   a. Hardware and software constraints
   b. Interface constraints
   c. Contractual and legal requirements
7. **Design phase budget and schedule**
   a. Design phase personnel and computer requirements
   b. Development schedule
8. **Physical requirements**
   a. Workload and volume
   b. Response times
   c. Functional layouts of computer inquiry screens and reports
   d. System growth
9. **Recommendations**
   Project leader's recommendations
10. **Approvals**
11. **Attachments**
   a. Approved feasibility document
   b. Analysis memos, summaries, tables, graphs, charts
   c. Cost/effectiveness schedules

After defining the requirements for the new system, the current logical DFD should be modified to reflect the components of the future system. The development team may propose logical design alternatives for the future logical system that may involve derivatives of the following:

- Activities
- Data stores
- Control activities

After describing the logical system, we are now at the point of describing how the new system will operate. Working with a future logical system, an analysis team could devise several physical alternatives. An example of the alternatives that must be considered is the decision concerning which processes will be manual and which will be automated. These physical requirements are the core of the new system. Any new system selected will be based on the outcome of this process.

Cost/Effectiveness Study

To facilitate selecting a future physical system, the systems analysis team conducts a **cost/effectiveness study**, which provides quantitative and certain qualitative information concerning each of the alternatives. This information is used to decide which alternative best meets a user's needs. In making this determination, the team asks two questions: (1) Which alternative accomplishes the user's goals for the least cost (or greatest benefit)? and (2) Which alternative best accomplishes the user's goals for the system being developed?

Perform the Cost/Benefit Analysis. The **cost/benefit analysis** or study is performed first because the data are *relatively* easy to obtain and are more objective than the data on effectiveness. Also, for many decision makers and for many decisions, the cost/benefit criterion is the most important.

In conducting the cost/benefit study, the analyst *estimates* the costs and the benefits of the new system. **Direct costs (benefits)** are those directly attributable to the system or the system change. Examples of direct costs include equipment purchased, personnel salaries, site preparation, and materials and supplies. Direct benefits attributable to the system include items such as reduced personnel and hardware costs, and improved data reliability. **Indirect costs (benefits)** are not *directly* attributable to the system or the system change. Costs that we would normally associate with overhead expenses, such as personnel fringe benefits and utilities, are indirect costs. Indirect benefits are perhaps the most difficult to quantify. An example of an indirect benefit is increased revenue resulting from improved customer support; such benefits are extremely difficult to measure.

Tangible costs (benefits) *can* be *reasonably* quantified. Examples of costs include software purchases and insurance, whereas reduced equipment costs and increased revenue are examples of benefits. An **intangible cost (benefit)** is one that *cannot* be *reasonably* quantified. A productivity loss caused by low employee morale is an example of such a cost, whereas intangible benefits may accrue from improved information.

We incur **nonrecurring costs**, such as those for systems development, only once to get the system operational. **Recurring costs**, such as those for equipment rental, occur throughout all or most of the system's life.

Early in the development process, it may be difficult to estimate all costs. For example, hardware costs depend on decisions made in the next development phase, systems selection. The amount of intangible costs may never be known. What is important is to *identify* all costs. One such cost is that incurred by users participating in the development. Such costs should be, but seldom are, charged to the development project.

Intangible benefits take on increased importance as organizations develop a larger percentage of systems aimed at solving management and decision-making problems. Determining the benefits for the traditional "bread and butter" systems, such as payroll and accounts receivable, is relatively easy. The benefits for such systems usually include items such as reduced clerical costs. However, determining the benefits for a market analysis system presents problems. For nontraditional systems, the benefits may include "increased sales if we provide management with better information" or "improved morale with a modern user interface"—these benefits are difficult to quantify. In the cost/benefit analysis, we do our best to estimate costs and benefits. In the effectiveness analysis, we handle costs and benefits for which estimates were not possible.

Perform the Effectiveness Analysis. After conducting the cost/benefit analysis, the analysis team should determine each alternative's effectiveness. The ranking of each alternative on its relative capability to satisfy the user's requirements (goals) for the system either verifies the team's cost/benefit results or produces conflicting results. The effectiveness analysis might proceed as follows. The analysts and the user list all relevant criteria, including costs and benefits used in the cost/benefit analysis. At this time, we include intangible items for which monetary costs and benefits could not be estimated during the cost/benefit study. The users and the analysts jointly assign subjective rankings to each criterion for each alternative. The team then ranks the alternatives by summarizing the ratings.

The final step in selecting the alternative physical system is to recommend an alternative to the user, to management, and to the *IT steering committee*. Normally, making the recommendation is a straightforward process because the team recommends the highest-ranking alternative. However, there may be conflicting information, such as when one alternative ranks best in the cost/benefit analysis, but another ranks best in the effectiveness analysis. In cases where one alternative is not clearly superior, the user and analysts must confer about and agree on an alternative to be proposed to the IT steering committee.[3] Usually, information about all alternatives, along with the development team's recommendation, is presented to the committee.

Complete and Package the Approved Systems Analysis Document

To complete the systems analysis phase, the project team must collect the products of the analysis and organize these products into the *approved systems analysis document* (see Exhibit 17.2, pg. 606).

The first analysis deliverable is the logical specification. This is used in *systems selection* to choose the appropriate software to be acquired from external sources. Alternatively, if the software is developed in-house, it is used in *structured systems design* to design the software and to develop manual procedures, training programs, and so on.

The second analysis deliverable is the *physical requirements*. These requirements are used in *systems selection* to acquire computer equipment for the new system. In addition to the physical requirements related to hardware, the physical requirements should include *functional* layouts of inquiry screens and reports. These are important for at least two reasons. First, users are as much concerned with how they will interface with the system as they are with the system's logic. Their information needs must be clearly and completely identified at this point in the development. Second, it is virtually impossible to perform a high-quality software study unless we can compare the outputs of proposed vendor systems to our specific requirements. At this point, the sample reports and screens are called *functional* layouts because they show the information elements that are needed without getting into all the *details* of the screen or report design, a topic that is beyond the scope of this chapter.

Another deliverable, implicit at the conclusion of each systems development step, is the *budget and schedule*, which contains two major parts:

- The *budget*, obtained during the cost/benefit analysis, specifies the expected costs to complete the systems development.
- *Schedules* control systems development efforts by setting limits on the time to be spent on development activities and by coordinating those activities.

The final step in completing and packaging the approved systems analysis document is to obtain approvals. As discussed earlier, *signoffs* may be obtained from users, information services, management, and internal audit. In addition, the controller may sign off

[3] Be aware that reaching such an agreement can present practical problems that are beyond the text's scope. Suffice it to say that the "agreement" does not just magically happen.

to indicate that the cost/benefit analysis is reasonable. When the sign off step is complete, we are ready to move to the selection and design phases of the new system.

Systems Selection

The outcome of the system selection process will support either the purchase or development of an AIS. Many of the concepts are the same regardless of the source of the system. In this section, we discuss the process and identify the concepts that are different for purchased or developed systems. As you can see in Figure 17.1 (pg. 602), systems selection lies between structured systems analysis (bubble 2.0) and structured systems design (bubble 4.0). **Systems selection** is a set of procedures performed to choose the software specifications and hardware resources for an information system. Systems selection uses the new system's functional requirements (the logical specification) and physical requirements that were developed in the analysis phase to decide what software design and hardware resources will be used to implement the new system. Only after preliminary software design elements and hardware resources are chosen does the detailed design begin.

The systems selection tasks are as follows:

CONTROLS

- Determine what computer software design will implement the logical specification developed in structured systems analysis. When developing an information system, careful attention must be paid to integrating sound internal controls into the software rather than treating controls as a bolt-on afterthought.
- Determine what computer hardware will satisfy the physical requirements established in structured systems analysis (e.g., wireless versus wired access for warehouse picking operations, local versus off-site for file storage, and so on).[4] In making our choice, we should also be cognizant of the implications for the security and control of our information systems. Additionally, to fully understand the cost implications, consideration should be given to environmental controls (i.e., temperature, electrical, etc.).
- Choose acquisition financing methods that are in the best interest of the organization. We must decide whether it is better to purchase, rent, or lease the computer equipment. In addition, we must decide if our data center will be completely within our control or if we will use a service bureau, *application service provider*, or other outsourcing option.
- Determine appropriate acquisition ancillaries. We must establish contract terms, software site-licensing arrangements, computer maintenance terms, and software revision procedures and responsibilities.

The Systems Selection Deliverable: The Approved Configuration Plan

The **approved configuration plan**, the final output of systems selection (see Figure 17.1, pg. 602), summarizes the choices made in the study. The information in the configuration plan is used in the next phase of systems development and acquisition, structured systems

[4] An organization's existing hardware might be used to implement a new information system. In this case, the hardware phase of the study would verify that the existing hardware is adequate, given the physical requirements.

design, to build the software (for a system to be developed), acquire the hardware, acquire the software (for a system to be purchased), and develop the implementation plan. The approved configuration plan usually specifies the following items:

- Chosen software configuration and expected performance specifications
- Chosen hardware type, manufacturer, and model, including expected performance specifications
- Items to be included in the hardware contracts, such as proposed computer maintenance procedures and proposed procedures by which vendors will provide hardware revisions
- Results of testing alternative software design and hardware resources
- Assessment of financing and outsourcing alternatives

As indicated earlier, a company can decide to purchase an AIS solution from an external vendor, rather than developing an in-house solution. If acquisition is chosen over development, the configuration plan will include an analysis of the purchase alternatives and the recommended system based on the system selection process. Similarly, there are alternative ways to acquire needed computer hardware, as discussed next. The hardware issues raised next apply to situations where an AIS is acquired from an external vendor or developed internally.

Hardware Acquisition Alternatives

Before we proceed to the intermediate steps in systems selection, let's spend time examining the various hardware procurement options that an organization must consider. Hardware can be acquired (rented, leased, or purchased) by an organization and managed by the organization's personnel. Alternatively, the hardware can be owned and managed by external entities. Table 17.2 compares these external and internal sources for computer hardware. A review of the table should lead you to conclude that external sources usually provide more capacity and affect the organization's resources less, whereas internal sources can be matched more easily with the organization's needs.

Using internal acquisition, computer hardware can be purchased, rented, or leased from the manufacturer (vendor) or from a leasing company. In such cases, the hardware is acquired, installed in the organization's facilities, and operated by the organization's personnel. As noted in Table 17.2, possession and management by the organization (internal hardware source) is less flexible (because of fixed cost and limited capacity, for example) than is the use of external sources, but it does permit the organization to control and tailor the system.

External acquisition is suited for an organization preferring not to own or manage its own computer facilities. The organization can use a service bureau, outsourcing, or

TABLE 17.2 Internal versus External Hardware Sources

Internal	External
Can determine level of control, security, and privacy.	Level of control may vary and be difficult to attain, especially if many companies use the same hardware.
Management and staff must be in-house.	Management and staff are provided.
Capacity limited.	Additional capacity may be available.
Costs are mostly fixed.	Costs are mostly variable.
Tailored to our needs.	Tailoring varies.

an *ASP*—to fulfill its hardware needs. A **service bureau**, a firm providing information processing services, including hardware and software for a fee, can provide the services less expensively and in a timelier manner than would be possible with an in-house computer. Generally, service bureau contracts include company involvement with the system. **Outsourcing** (often called co-sourcing) is the assignment of an internal function to an outside vendor.

Outsourcing has come to encompass many of the external hardware acquisition alternatives that have been available for years. The new twist to these alternatives is the ownership by the outsourcing firm of the user organization's computer facility. Organizations can retain management of their IT while obtaining some application functionality through ASPs in a more flexible and less costly way than through a service bureau.

The main advantage of using a service bureau, outsourcing, or using an ASP is that the user organization does not have to operate and maintain the computing resource. The downside to any of these external acquisition methods is that the organization must be willing to sacrifice its independence and must compromise scheduling and data security.

The Intermediate Steps in Systems Selection

There are two primary steps in the systems selections phase of the SDLC:

1. Prepare requests for proposal.
2. Evaluate vendor proposals.

Prepare Requests for Proposal. A **request for proposal (RFP)** is a document sent to vendors that invites submission of plans for providing hardware, software (for a purchased system), and related services. Before preparing any requests for proposal, a firm must decide what approach will be taken for soliciting the proposals. An organization can ask one vendor for a proposal, or it can ask many. An organization satisfied with its present vendor might send an RFP to only that vendor. An organization might choose to stay with its present vendor to minimize program conversion costs, to obtain attractive contract clauses (such as discounts or future benefits), and to reduce retraining costs. Also, if an organization has a specialized need that can be met by only one vendor, the organization might send an RFP to only that vendor. Unless an organization has a particular reason for limiting its RFPs to a single vendor, submission to several vendors is preferred because with multiple proposals, an organization has a variety of possibilities from which to choose. In addition, vendor concessions and discounts can be obtained if an organization retains a bargaining position by dealing with several vendors.

The organization can request bids on specific computer configurations (e.g., request a bid for Model "XYZ") or to meet general performance objectives (e.g., request a bid for a computer system capable of handling the entry of 5,000 business events, such as customer orders, each hour). The former approach leads to a simple evaluation of proposals, but it assumes that the systems analyst preparing the RFP knows what equipment will meet the organization's requirements. The latter approach allows vendors to propose solutions to an organization's requirements; the organization may not have anticipated some of these solutions. Choices made from solutions generated entirely within the organization may be suboptimal.

After deciding how to prepare the requests (i.e., the approach to be used), the organization must then decide to which specific vendors the RFPs will be sent. Vendors from whom the organization has previously received proposals or with whom it has previously done business are candidates for receiving proposals. The analysts assigned to conduct systems selection also might research vendor evaluations published in the computer press or in other computer-based or paper-based services. This research is described in Technology Summary 17.2.

Technology Summary 17.2

SOURCES OF VENDOR INFORMATION

Analysts can utilize a variety of paper-based, computer-based, and online services to identify and evaluate computer hardware, software, and vendors. The information contained in these services, especially that resulting from independent expert analysis of a vendor and its products or from user surveys, can provide valuable insight into the vendor's quality, financial condition, number of installed systems, and similar information. Some are reports such as those available from Gartner Group, Inc. (http://www.gartner.com). Gartner services include Dataquest with research and advice in a number of areas including benchmarks, performance measurement, software and hardware products, and vendor selection. Another Gartner service, Datapro, publishes reports in such categories as computer systems and software library, communications library, managing data networks, computers and peripherals, and e-business and Internet.

Magazines—both printed and online—also provide independent reviews of vendors, hardware, and software. For example ZDNet (http://www.zdnet.com) publishes reviews in their online magazine *eWEEK* and in magazines that are both printed and published online, such as *PC Magazine* and *Computer Shopper*.

In addition to these independent sources of information about software, hardware, and vendors, the Internet provides a wealth of information directly from the vendors. For example, in a quick tour of the Web (March 2006), sites were found for Symantec (network security, virus protection, etc.—http://www.smallbiz symantec.com), IBM (http://www.ibm.com), Microsoft (http://www.microsoft.com), SAP (http://www.sap.com), and Gateway (http://www.gateway.com). Through these sites, news was obtained about upcoming products, lists of existing products, customer support, technical support, software purchases, and software fixes and upgrades.

Finally, in addition to traditional software vendors, you can connect via virtual meeting places on the Internet such as those at http://www.SourceXchange.com and http://www.Cosource.com with open-source programmers. These programmers may be able to meet your software needs in less time and for less money than would be the case through other means. They build applications using existing open-source components rather than develop them from scratch or by using commercial software. Once developed, however, the application is made available to others via the open-source software network. Consequently, open-source programming should not be used for business-critical systems or those that provide a competitive advantage.

Source: Dominique Deckmyn, ''Open Source Projects Get Done Cheaply,'' *Computerworld* (April 24, 2000): 44.

Using the information contained in the logical specification and/or in the physical requirements, the analysts prepare the RFPs and send them to the chosen vendors. Exhibit 17.3 lists the typical contents of an RFP.

Note that Exhibit 17.3 assumes that the RFP asks for a bid for performance objectives, rather than for a particular computer configuration or product. The section on projected growth requirements is important relative to the RFP. The better an organization accurately projects the long-term requirements for a new system and obtains hardware that can satisfy that long-term demand, the longer it will be before the system needs to be revised and new hardware obtained.

Evaluate Vendor Proposals. The second task in the system selection phase is to evaluate vendor proposals. Using the vendor responses to the RFP and the physical requirements, the analysts must decide which, if any, proposal best meets the organization's needs. The process of evaluating the vendor proposals includes three steps:

1. Validate vendor proposals.
2. Consider other data and criteria.
3. Suggest resources.

EXHIBIT 17.3 Typical Contents of a Request for Proposal

1. A description of the scope of the request, including a brief description of the hardware, software, and services for which a proposal is requested
2. A description of the AIS including, if applicable, the *logical specification* and *physical requirements*, which in turn include specifications for the:
   - Inputs
   - Outputs
   - Data storage
   - Processes
   - Controls
3. Procedures for submitting proposals, including a timetable for proposal submission, evaluation of proposals, and final decision date
4. Price and budget constraints
5. Vendor information required, including:
   - Contract terms
   - Warranty
   - General company information
6. Hardware performance objectives, such as:
   - Data storage capacities and access requirements
   - Input/output speeds and volumes
   - Data communication requirements
   - Computational demands
7. Software performance objectives, such as:
   - Inputs
   - Required outputs
   - Data table sizes and access requirements
   - Operating system requirements
   - Utilities
   - Language compilers
8. Projected growth requirements, including expected changes in input/output volumes
9. Criteria to be used in evaluating vendor proposals, such as:
   - Ability to meet performance objectives
   - Benchmarks
   - Reliability
   - Documentation
   - Training
   - Backup
   - Maintenance assistance

Many organizations assign a team to evaluate the proposals. The team could consist of personnel with IT technical expertise, business process owners, system users, external consultants, lawyers, and accountants. The evaluation team completes these three steps to suggest the hardware and services that best meet the organization's requirements.

The first evaluation step is to validate the vendor proposal to assess whether the system hardware does what the organization requires. To determine whether a system meets the requirements of the RFP, the evaluation team can study a proposed system's specifications and performance.

Specifications are straightforward descriptions of the hardware. For example, a server's storage space or a printer's speed can be examined to determine whether the hardware has sufficient capacity and speed to perform as required. Other specifications include items such as compatibility with other current or future technologies, as well as potential for future expandability and upgrades.

Performance features can be determined only through testing, measurement, or evaluation and often include items such as effective vendor support, documentation quality, system reliability, and scalability. One commonly used method for measuring system performance involves measuring the system's throughput, which reflects the quantity of work performed in a period of time. For instance, the number of invoices that a system processes in one hour is a measure of throughput. Other performance measures, such as ease of use, are more subjective and may be more difficult to determine.

Notice that during validation we are not comparing vendor proposals; we are determining which proposals can meet our requirements. Those proposals that can meet our requirements are compared when we "suggest resources."

RFPs often distinguish mandatory and desirable system characteristics. As a first step in the validation process, many proposals can be rejected because they fail to meet mandatory effectiveness requirements. At this point in our analysis, we have not yet tested the system and can evaluate the system only on the basis of the system's parameters, not the system's performance. Still, even these specifications might not hold up in a test. For example, hardware compatibility may be specified but not work in certain configurations.

After completing the first-level effectiveness analysis, the evaluation team tests[5] the remaining systems (i.e., those that satisfy mandatory requirements) to determine the accuracy of the vendors' specifications and how well the equipment will work for the organization. One way to compare systems is to use benchmark data. A **benchmark** is a representative workload, processed on each vendor's proposed system configuration, to obtain comparative throughput measures. In the first step in our analysis, we determined what a system is; in testing, we determine what that system can do.

After we have validated vendor proposals, we might consider other data and criteria. Quite often, vendor presentations are made at the site of an existing user. This approach gives the evaluation team an opportunity to see the system in a working environment. External interviews—interviews conducted with personnel outside the organization—can provide valuable insights into vendor performance.

Finally, the team must summarize its findings by recommending one vendor proposal. To recommend one vendor, the evaluation team must compare the proposals that have not been eliminated. To support the recommendation, the evaluation team should list the relevant criteria and indicate the performance of each vendor on each criterion. Table 17.3 depicts such an analysis. The criteria—we have presented only a sample here—would be those indicated in the logical specification and physical requirements as well as any others the evaluation team considers important. The evaluation team uses the performance measures gathered during system testing and when other data and criteria were considered.

TABLE 17.3 Detailed Vendor Comparison

	Vendors		
Criteria	**A**	**B**	**C**
Documentation quality	good	good	poor
Cost of typical configuration	$53,630	$29,900	$59,300
Monthly maintenance of typical configuration	$422	$515	$448
Maximum number of workstations	8	32	32
Benchmark results:			
Number of invoices/minute	2	4	6
Query response time (seconds)	1	1.5	2.0

[5] Often, vendors will propose a system that does not actually exist *yet*. In such cases, we cannot test an actual system; our only option is to *simulate* the proposed system, as discussed later in this section.

An analysis such as that in Table 17.3 might not clearly indicate a superior vendor. It is normally advisable and necessary to perform a detailed analysis that assigns scores to each vendor on each relevant criterion. In addition to scoring each, the evaluation team ranks the importance of each criterion by assigning it a weight. Of course, the comparison is only as valid as the weights and scoring values used. Also, if the results of this analysis agree with the previous evaluations and the intuition of the evaluation team, then the team receives a certain amount of comfort from this analysis. Finally, using this type of weighted scoring system facilitates communication of the team's recommendations to the IT steering committee and the user, and thus helps gain support for that recommendation.

Structured Systems Design

In structured systems design, for systems under development, the software specifications are refined for new or revised business process applications, and implementation plans are prepared. If the AIS is being purchased, an organization will focus on the implementation process.

Recall from earlier discussions that certain systems development tasks are comparable to tasks undertaken in the construction of an industrial park. Systems selection, in which the software design and hardware resources are chosen, is similar to drafting blueprints and choosing contractors for a construction project. Structured systems design, in which the software is designed and the implementation is planned, is similar to finalizing blueprints and other construction-related plans.

Studies have shown that systems developed using structured systems design techniques are less costly over the life of the system because maintenance of the system is less expensive. Also, structured systems design avoids design errors that further increase the cost of the system. Implementation planning, conducted during structured systems design and introduced in this section, increases the probability of a smooth transition to the new information system.

Definition and Goals

Structured systems design is a set of procedures performed to convert the logical specification into a design that can be implemented on the organization's computer system. Concurrent with specification of the system's design, plans are developed for testing and implementing the new system and for training personnel. Portions of the user manual are also developed at this time.

Figure 17.1 (pg. 602) shows that structured systems design is the fourth major step in the development of an information system (bubble 4.0). Examine this figure to see the position that structured systems design holds in the SDLC.

Let's return to our earlier analogy between systems development and the construction of an industrial park. Converting the information system's logical specification into detailed design specifications is similar to finalizing the construction blueprints. The models developed earlier in the construction project are not detailed enough to allow the actual construction to begin; final blueprints provide that detail. Also, planning must be undertaken to determine the construction schedule for the buildings. Preconstruction planning is analogous to the computer system implementation planning done at this juncture in the SDLC. For example, the development team must plan how much of the system to implement and when.

The structured systems design tasks are as follows:

- *Convert the structured specification into a reliable, maintainable design:* This is similar to the process of converting the building model into a final blueprint.

- *Develop a plan and budget that will ensure an orderly and controlled implementation of the new system:* Procedures must be devised to get the hardware in place, the programming completed, the training conducted, and the new system operating.
- *Develop an implementation test plan that ensures that the system is reliable, complete, and accurate:* A plan must be developed to test the system to ensure that it does what the user wants it to do.
- *Develop a user manual that facilitates efficient and effective use of the new system by operations and management personnel:* These personnel must know how to use the new system effectively, and the information processing staff must know how to operate the system.

The Systems Design Deliverable: The Approved Systems Design Document

The approved systems design document, the final deliverable of structured systems design (see Figure 17.1 on pg. 602), documents the system design and summarizes the implementation, training, and test plans. The design document is used by the following:

- *Programmers:* To write the computer programs and program interfaces.
- *Personnel department:* To develop and conduct training and education programs.
- *Information systems personnel:* To test and implement the system.

The design project leader must assemble the components of the systems design document and obtain the required user approvals (to ensure the adequacy of the design and plans) and management approvals (to signify concurrence with the design, training, and implementation process). In addition, for systems to be developed, IT management furnishes a supervisory/technical approval of the adequacy of the software specifications, and auditors ensure adequacy of the controls and the design process (including implementation planning).

Systems Implementation

Systems implementation reflects a set of procedures performed to complete the design contained in the approved systems design document and to test, install, and begin to use the new or revised information system. Figure 17.1 (pg. 602) depicts systems implementation as the fifth major step in the development of an information system. Examine this figure to see the position that systems implementation holds in the SDLC. You can see that systems implementation follows structured systems design and has as its two major inputs, the approved configuration plan (developed during the systems selection step) and the approved systems design document (produced in the structured systems design step of systems development).

In the implementation stage of the SDLC, the most significant difference between in-house development of a system and the purchase of a system is the application programming. For purchased systems, the software manufacturer completes the programming step. However, even purchased systems frequently require programming tasks, such as writing interfaces between the new system and existing software.

Recall our earlier analogy that certain systems development tasks are comparable to those undertaken in the planning and construction of an industrial park. Systems implementation (in which the computer programs are written and the system is put into operation) is analogous to the process of actually constructing the industrial park.

During systems implementation, the organization acquires the specified computer resources, prepares the site to receive them, and installs the new software and hardware. The approved systems design document is used to complete the design, write the computer programs, conduct the training, test the system, and install the new or revised system.

The systems implementation tasks are as follows:

- Complete, as necessary, the design contained in the approved systems design document: New information may have become available since the design was approved. It is easier to address such issues before the programs are written. This applies to in-house systems only.
- Write, test, and document the programs and procedures required by the approved systems design document: This applies to in-house systems only.
- Ensure, by completing the preparation of user manuals and other documentation and by educating and training personnel, that the organization's personnel can operate the new system.
- Determine, by thoroughly testing the system with users, that the system satisfies the users' requirements.
- Ensure a correct conversion by planning, controlling, and conducting an orderly installation of the new system.

The Systems Implementation Deliverable: The Project Completion Report

To understand the implementation process, you need to see where you are going. Systems implementation ends with the operation of the newly acquired or revised system and with the submission of a project completion report. The project completion report summarizes the implementation activities and provides documentation for operating the new system and for conducting the post-implementation review and systems maintenance. The project completion report usually includes the following items:

- Summary of requirements satisfied by the new system
- Estimated and actual duration of each development stage
- Estimated and actual systems performance (e.g., response time, costs, benefits)
- System documentation, which provides an overview of the new system
- Program documentation, which includes source code and other related items
- User manual, which describes operating procedures for both manual and automated procedures
- Operations run manual, which contains operating instructions for computer operations
- System test report
- User training programs and manuals
- Operator training programs and manuals

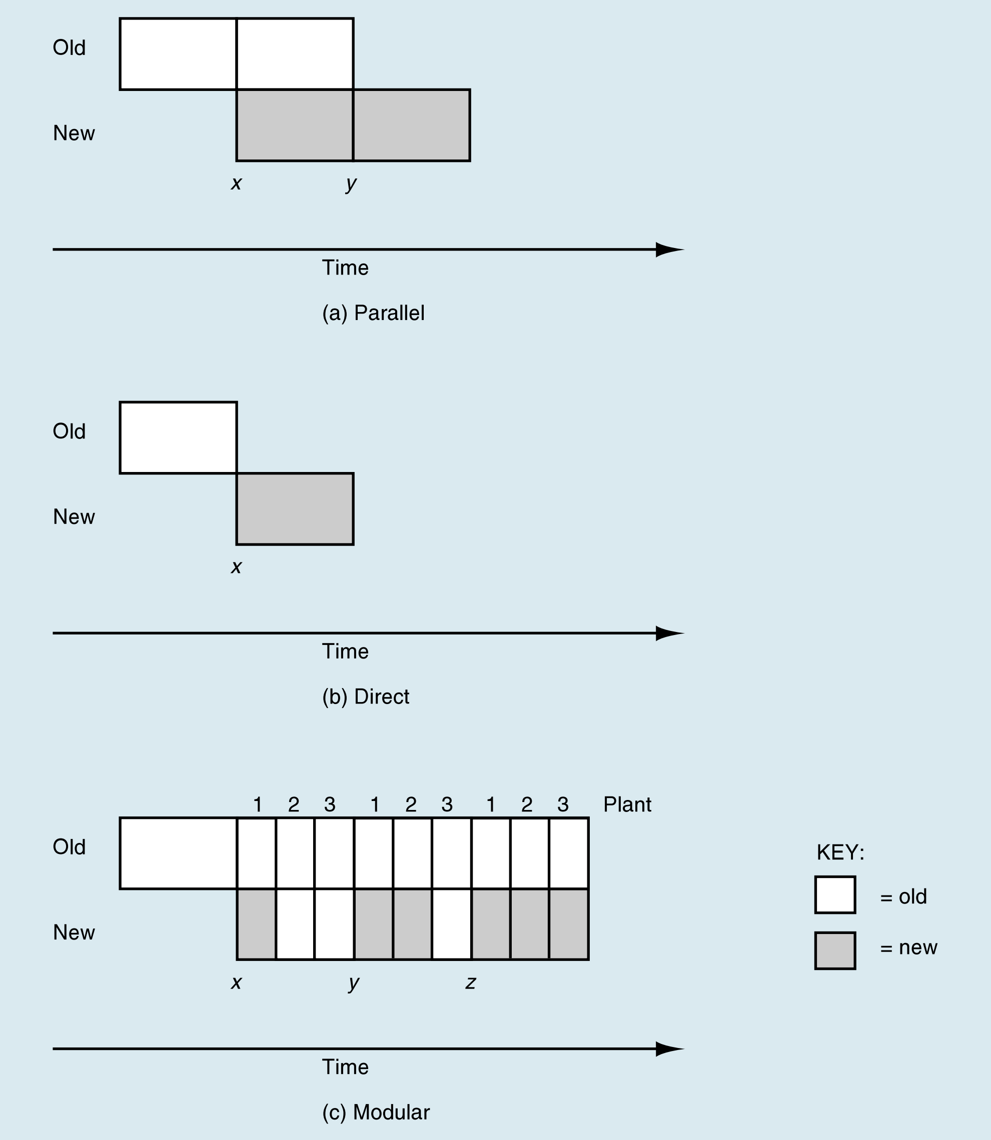
Approaches to Implementation

There are alternative implementation approaches that can be taken to install whatever portion of the system that has been developed. Selecting an appropriate approach can greatly facilitate conversion to the new system. Depending on the approach and circumstances, greater control and user satisfaction can be ensured. Figure 17.2 (pg. 618) depicts the three most common implementation approaches.

CONTROLS

Figure 17.2(a), the parallel approach, provides the most control of the three. In the parallel approach, both the old and new systems operate together for a time. During this period, time x to time y (which is usually one operating cycle, such as one month or one quarter), the outputs of the two systems are compared to determine whether the new system is operating comparably to the old. At time y, management makes a decision, based on the comparison of the two systems' outputs, concerning whether to terminate

FIGURE 17.2 Implementation Approaches



the operation of the old system. The parallel approach provides a high level of control because the old system is not abandoned until users are satisfied that the new system adequately replaces the old. However, greater control comes at a cost; meaning, it is expensive to keep two systems running simultaneously. For instance, running two systems side-by-side can require duplicate or excess computer processing capacity, data storage space, and human labor.

ENTERPRISE SYSTEMS

CONTROLS

Figure 17.2(b), the direct approach, is the riskiest of the three approaches because at time x, the old system is stopped and the new system is begun. This implementation method is also referred to as the "big bang" or "cold turkey" approach. There can be no validation that the new system operates comparably to the old because the lights are "turned off" with the old system and simultaneously "turned on" with the new system. Enterprise systems are often implemented using this approach. Direct implementations can lead to disaster if not carefully tested, planned, and executed. So, why take this approach? Certainly, some level of comfort or control is lost with the direct approach. But, this implementation method can be less costly than the parallel approach—if all goes well. With very large implementations, such as enterprise systems, it is often capacity and/or cost prohibitive to take the parallel approach. On the bright side, the direct approach forces users to learn the new system because they do not have the old system to fall back on. Although this might have immediate negative effects on satisfaction because users do not want to let go of the old system or they fear the new system, the new system can get up and running very quickly. And, if the implementation is properly planned and the users are thoroughly trained, direct implementations can ultimately lead to increased user satisfaction as compared to the old system.

CONTROLS

Figure 17.2(c) (pg. 618), the modular approach, can be combined with the parallel or the direct approaches to tailor the implementation to the circumstances. With the modular approach, the new system is either implemented one subsystem or module at a time or is introduced one organizational unit at a time. The modular approach is also referred to as the phased approach. For example, a new order entry/sales system could be implemented by first changing the sales order preparation and customer inquiry portions, followed by implementing the link to the billing system, followed by the link to the inventory system. Figure 17.2(c) depicts the gradual implementation of a new system into three organizational units. A new payroll system is installed for the employees of plant 1 at time x, followed by plant 2 at time y, and finally by plant 3 at time z. Implementation at any plant could be direct or parallel. Modular implementation permits pilot testing of a system or system component and elimination of any problems discovered before full implementation. If properly planned and executed, modular implementations can combine the safety and control of a parallel implementation with the cost/time savings of a direct approach.

The Intermediate Steps in Systems Implementation

The first step of the implementation process is to complete or update the design of input and output reports, documents, computer screens, the database, manual processes, and certain computer processes. This completion is required to reflect changes since the design document was approved. No matter what the scope of a project and the diligence of the design team, there are always changes! By adjusting the design at this point, you will minimize the cost and impact on the overall system.

At any time after the computer resources are chosen and indicated in the approved configuration plan, the software and hardware may be acquired, the site prepared, and the computer system installed.

Contract negotiation and preparation is an important part of the computer acquisition process. Computer, legal, and financial expertise must be combined to negotiate and execute the contracts. Contracts are necessary for computer hardware and software lease, rental, or purchase as well as for hardware and software service. One important point to remember when contracts are negotiated is that nothing should be left out of the contract; nothing should be assumed. Detailed specifications protect the buyer and the seller and keep them both out of court, unless one fails to perform contract provisions.

Another acquisition issue involves the preparation of the site to receive the computer equipment. Sufficient electrical power and power protection, air conditioning, and security, as well as the computer room's physical structure and access to that room, must be planned for and provided. If the contracts are well written and the site well prepared, installation of the computer hardware, software, and related equipment should be relatively straightforward. Contingency plans to allow for delays in site preparation or equipment delivery should be considered.

The next task in systems implementation depends on whether the programs are to be custom programmed or purchased. If they are to be programmed, you also have to include time to test and debug the programs, and complete the program documentation. The programming process is important because the programming task in systems development consumes more resources and time than any other development task.

To test the system, the programmer must develop the test plan, which outlines how each program module will be tested. The test plan includes the test data that the program unit is expected to handle. The user, the programmer, and another member of the programming team do a "walkthrough" of the module specifications and the test plan to determine that the latter is adequate; then the programmer codes the program or creates the code using computer software (a code generator) that "writes" program code. After coding, a different programmer[6] does a "walkthrough" of the code to see that the code faithfully and without error implements the module specifications. The programmer tests the individual module and removes any errors found during testing. This removal of program errors is called *debugging*. Finally, the programmer must complete the program documentation. Maintenance programmers will use this documentation to make program changes, correct errors in the programs, and add enhancements to the programs.

ENTERPRISE SYSTEMS

If a software package has been purchased, much of the programming step is replaced with procedures to configure the system for this application. During the implementation of an enterprise system, this process can be quite extensive as we configure the system to select, for example, the steps to be completed for each business process; the design of the screens to be displayed at each step of the process; and the data to be captured, stored, and output during the processes. Even if software is purchased, frequently there will be programming required to link the new system with existing applications.

The organization must choose the personnel who will operate the new system and must train them to perform their system-related duties. The system's users must be educated about the new system's purpose, function, and capabilities. Training may be given through a combination of schools run by software vendors, hardware vendors, vendors specializing in training, and programs conducted by the organization itself. Computer-assisted learning, such as interactive tutorials, might also be used.

An organization must choose the training delivery system that matches its training needs. The two variables to be considered in choosing a delivery method are the person being trained and the training objective. For example, formal classroom presentations, delivered to any audience, are appropriate for general overviews of a computer system. On-the-job training can be an effective component of a plan that includes instruction, follow-up, and ongoing assistance. Tutorials introduce relatively computer-literate users to software and hardware utilization. Without assistance, tutorials can be difficult for novices to use.

Resident experts may be a cost-effective way to deliver knowledge of a system to a large audience. In such situations, the experts are trained first, and then they instruct the other users. Online Help and Explanation facilities, along with well-designed screens and reports, can reduce the amount of up-front training necessary and provide ongoing guidance to system users.

Test the System

CONTROLS

Beyond testing program modules, the entire system is tested to determine that it meets the requirements established by the business process owners and users and that it can be

[6] This is done by a different programmer because it is easier for someone that did not do the coding to find errors.

used and operated to the satisfaction of both users and system operators. Testing is carried out by the systems developers, by the developers and the users together, and finally by the users. The more closely the test can simulate a production environment (e.g., people, machines, data, and inputs), the more representative the test will be and the more conclusive the results will be. Each test consists of the following steps:

1. Specify the test conditions.
2. Review the test conditions (i.e., walkthrough).
3. Create test data.
4. Execute the tests.
5. Evaluate the results.

ENTERPRISE SYSTEMS

Several types or levels of tests are usually completed before a system can be implemented. From the users' point of view, three of these tests are the most important. The *system test* verifies the new system against the original specifications. This test is conducted first by the development team and then by the users with the assistance of the team. The *acceptance test* is a user-directed test of the complete system in a test environment. The purpose is to determine, from the user's perspective, whether all components of the new system are satisfactory. The user tests the adequacy of the system (both manual and automated components), of the manuals and other documentation, and of the training the users received. Finally, the *operations test* or environmental test runs a subset of the system in the actual production environment. This final test determines whether new equipment and other factors in the environment—such as data entry areas, document and report deliveries, telephones, and electricity—are satisfactory. After this final test, it is time for the conversion.

Conduct Conversion

CONTROLS

After all the previous implementation steps have been completed and signed off, the organization must carefully convert to the new system. Conversion includes converting the data and the processes (i.e., the programs). Controls must be in place to ensure the accurate, complete, and authorized conversion of the data and programs.

As the existing data are mapped into the new system, exception-reporting situations must be devised to ensure that the data are converted accurately. The user must suggest control totals that can be used to test the completeness and accuracy of the data conversion. For example, the total number of inventory items, the total on-hand quantity for all inventory items, or a hash total of inventory item numbers might be used as totals.

Both manual-based processes and computer-based processes must be converted. Conversion to new computer programs must be undertaken using program change controls to ensure that only authorized, tested, and approved versions of the programs are promoted to production status.

After the conversion is completed, the systems development project team now writes the project completion report, the final step in the implementation process.

Post-Implementation Review

The **post-implementation review** involves an examination of a working information system, conducted soon after that system's implementation to determine whether the user's requirements have been satisfied and whether the development effort was efficient and conducted in accordance with the organization's systems development standards. The review should be brief and inexpensive. Examinations conducted in response to a specific deficiency, systems maintenance, are discussed in the next section.

The post-implementation review tasks are as follows:

- Determine if the user is satisfied with the new system.
- Identify how well the system's achieved performance corresponds to the performance requirements, recommending improvements if necessary.
- Evaluate the quality of the new system's documentation, training programs, and data conversions.
- Ascertain that the organization's project management framework and SDLC were followed during development.
- Recommend improvements to the systems development/acquisition standards manual if necessary.
- Improve the cost/effectiveness analysis process by reviewing cost projections and benefit estimations and determining the degree to which these were achieved.
- Improve project-planning procedures by examining total project costs and the project team's ability to adhere to project cost estimates and schedules.
- Make any other recommendations that might improve the operation of the system or the development of other information systems.

CONTROLS

Internal auditors, IT auditors, or systems analysts (other than those who developed the system) may conduct the post-implementation review. If the organization has insufficient personnel with the required expertise who are independent of the system's development, consultants may be hired to conduct the review. The independence of the review is important because the review provides feedback (i.e., a control) on the development process.

The post-implementation review is performed as soon as the system is operating at full capacity, which could be one month or one year after implementation. The review should examine a fully functioning system so as not to draw erroneous conclusions about system performance. The review should be conducted soon enough after implementation to be able to take advantage of any improvements that can be made to the system or to the systems development methods used.

The following parties *sign off* on the post-implementation review report:

- Business process owners/users sign off to indicate that the system was performing as depicted in the report and to indicate concurrence with the report's conclusions and recommendations.
- Auditors participating in the review sign off to indicate that appropriate procedures were followed in performing the review and to concur with the report's conclusions and recommendations.

Systems Maintenance

Systems maintenance is the modification (e.g., repair, correction, or enhancement) of existing applications. Systems maintenance expenditures can account for 50 to 80 percent of the total cost of a system over its total life cycle. These costs should be reason enough to reduce the need for maintenance and to carefully monitor maintenance that cannot be avoided.

You should realize that not all maintenance expense is necessarily bad; rather, the issue is the relative amount spent on systems maintenance. After all, applications must be adapted to a changing environment and improved over time. There are three types of maintenance activities:

- *Corrective* maintenance must be performed to fix errors.
- *Perfective* maintenance is conducted to improve the performance of an application.
- *Adaptive* maintenance adjusts applications to reflect changing business needs and environmental challenges.

The competitiveness of many industries and the need for organizations to remain flexible in such environments increase the importance of adaptive maintenance.

The following systems maintenance tasks are established in light of the systems maintenance costs and concerns cited, as well as other issues to be discussed in this section:

- Accomplish system changes (and reconfigurations) quickly and efficiently.
- Prevent system changes from causing other system problems.
- Make system changes that are in the organization's overall best interest.
- Perfect systems development and systems maintenance procedures by collecting and using information about system changes.
- Supplant systems maintenance with the systems survey if requested changes are significant or if they would destroy the system.
- Minimize control exposure and organizational disruption that can be caused by systems maintenance.

CONTROLS

To accomplish the systems maintenance tasks, organizations often adopt the following procedures and controls for their systems maintenance process:

- Because systems maintenance is like a miniature systems development, it should include analysis, cost/benefit study, design, implementation, and approvals for each development step. In systems maintenance, certain *SDLC* procedures deserve more attention than others. For example, changes must be tested prior to implementation to determine that a change corrects the problem and does not cause other problems. Participants and signoffs should be the same as those required for systems development. For example, the user should review system changes.
- By charging users for maintenance costs, an organization can reduce the submission of frivolous maintenance requests.
- By adopting a formal procedure for submitting change requests, batching these requests together for each application, and then prioritizing the batches, management can gain control of systems maintenance and reduce the expense and disruptions caused by the maintenance process.
- During systems maintenance, information should be gathered that provides feedback to improve the operation of the system and to improve the systems development process. For instance, poor quality application documentation and inadequate user training can cause numerous systems maintenance requests. Correcting these deficiencies can preclude the need for similar maintenance requests in the future. Likewise, improvements in the systems development process can prevent deficiencies from occurring in other systems when they are being developed.
- Management should see that *program change controls* are used to ensure that all modifications to computer programs are authorized, tested, and properly implemented.
- High-quality documentation must be created and maintained. Without current, accurate documentation, maintenance programmers cannot understand, and therefore cannot effectively or efficiently modify, existing programs.

CONTROLS

CONTROLS

Accountant Involvement in AIS Development/Acquisition

As an accountant, you are uniquely qualified to participate, and even lead, the development/acquisition process. As an information management and business measurement professional, you will bring to the table important knowledge and skills, including knowledge of the business processes and the business context for the new AIS, expertise in internal controls that will be so important during the development process and for the new system, and skills with metrics that will be needed to make important decisions and to monitor the acquisition process. Consider some of the specific roles you will eventually play in the process.

- *User:* Whether you are a staff accountant, controller, or chief financial officer, you will be using an AIS on a regular basis. In any of these roles, you might be a business process owner as well, which means that you will be responsible for (or own) certain data sets (such as payroll, general ledger, or customer billing data). As such, you very well could be the person who initiates the AIS acquisition cycle because you will be in prime position to recognize deficiencies and incompatibilities related to the existing AIS.
- *Analyst:* Because you will be intimately familiar with functionality and controllability aspects of accounting systems, you could be asked to participate on a systems analysis team to conduct a preliminary survey used to determine what is needed in a new AIS. Even in those situations where the accountant is not familiar with a particular business process, he or she can contribute their skills in acquiring and documenting descriptions of processes.
- *Purchaser:* You could also become a member of the AIS selection team. In this capacity, you will have to match the organization's requirements with the capability of various commercially available accounting systems, evaluate a set of potential AIS solutions, and help to make the final purchase selection.
- *Implementer:* After the AIS is purchased, it is time to switch the old with the new, which is much easier said than done. You could become involved with this delicate and vitally important phase of the acquisition cycle. If so, you will have to be equally adept at dealing with people, accounting, and technology. In particular, your intricate knowledge of accounting and information technology will help you to be successful in this role.
- *Consultant:* Although the Sarbanes-Oxley Act prohibits CPA firms who audit a public company to also serve as systems consultants for the same company,[7] AIS consulting is nevertheless alive and well. Even though all but one major CPA firm (Deloitte Touche Tohmatsu) has spun-off their consulting divisions into separate entities, these new consulting companies continue to need accountants. Also, CPA firms still consult (they often call themselves "business advisors"), just not for their audit clients. As a consultant, you can become involved with any or all of the AIS acquisition phases.
- CONTROLS *Internal Auditor:* Because internal auditors are knowledgeable about AIS function and control requirements, they often serve as advisors or consultants during the acquisition cycle. Also, as agents of management and the board of directors, internal auditors ensure that the acquisition team has followed the organization's standard procedures for systems acquisition and that the process has been efficient and effective.
- *External Auditor:* As an external auditor, you will conduct an internal control assessment. During your assessment, you will have to determine how changes in the AIS might have affected overall audit risk. As discussed in Chapter 7 of this textbook, professional guidance in this regard is found in Statement on Auditing

[7] The rationale for this change was to prevent auditors from assessing the quality (i.e., auditing) of the work performed by members of their own firm. To do so would violate their independence in appearance or in fact.

Standards (SAS) 94, entitled "The Effect of Information Technology on the Auditor's Consideration of Internal Control in a Financial Statement Audit." Thus, external auditors will review the acquisition cycle and assess if and how internal controls have been impacted.

You can clearly see how accountants can play many valuable roles in the AIS acquisition and development cycle. Most likely, you will eventually find yourself in one or more of these roles. Even if you do not enjoy technology, you cannot escape the natural marriage of accounting and information technology, as the lines between the two have been forever blurred.

SUMMARY

E-BUSINESS

In the past, it often took years for a new system to move through the initial steps in the SDLC (i.e., bubbles 1 through 5 in Figure 17.1 on pg. 602). Now we move at "Internet speed" and must develop business-to-business and other e-business infrastructures in 90 to 180 days. If we don't, we may be put out of business or absorbed by those organizations that can.[8] Oftentimes, new systems development projects involve changes in the way a company operates. Such changes, coupled with the "need for speed," raise the stakes regarding the odds of successfully accomplishing the systems development objectives.

The SDLC methodology is a tool that can help ensure successful system development projects. With successful projects, an organization can compete effectively in today's fast-paced environment. The knowledge of this tool can help you bring technological success to your organization.

KEY TERMS

systems development

systems development life cycle (SDLC)

systems development life cycle (SDLC) methodology

application service provider (ASP)

systems survey

feasibility study

preliminary feasibility study

structured systems analysis

approved systems analysis document

cost/effectiveness study

cost/benefit analysis

direct costs (benefits)

indirect costs (benefits)

tangible costs (benefits)

intangible costs (benefits)

nonrecurring costs

recurring costs

systems selection

approved configuration plan

service bureau

outsourcing

request for proposal (RFP)

benchmark

structured system design

systems implementation

post-implementation review

[8] Peter G. W. Keen, "Six Months—Or Else," *Computerworld* (April 10, 2000): 48.

Review Questions

RQ 17-1 What is systems development?

RQ 17-2 What is the systems development life cycle?

RQ 17-3 What is the systems development life cycle (SDLC) methodology?

RQ 17-4 What are the systems development objectives?

RQ 17-5 What are the tasks required to complete the systems survey?

RQ 17-6 What is structured systems analysis?

RQ 17-7 What are the tasks required to complete a structured systems analysis?

RQ 17-8 Why do we study and document the current physical environment?

RQ 17-9 Compare and contrast tangible and intangible benefits.

RQ 17-10 What is the purpose of conducting an effectiveness analysis?

RQ 17-11 What is systems selection?

RQ 17-12 What does the approved configuration plan specify?

RQ 17-13 What are the reasons for using external versus internal sources of hardware?

RQ 17-14 What is a request for proposal (RFP)?

RQ 17-15 What are the approaches to obtaining an RFP?

RQ 17-16 Why might a company issue an RFP for general performance objectives?

RQ 17-17 What is the difference between a specification and a performance measure?

RQ 17-18 Describe the process used to evaluate vendor proposals.

RQ 17-19 What is structured systems design?

RQ 17-20 What is systems implementation?

RQ 17-21 What are the three major approaches to implementing an information system?

RQ 17-22 What is the riskiest approach to systems implementation?

RQ 17-23 What two variables should be considered when choosing a method to train employees to operate a new system?

RQ 17-24 What is the purpose of testing the entire system as opposed to testing part of the system?

RQ 17-25 Describe the major steps in a post-implementation review.

RQ 17-26 What are the three types of maintenance activities?

RQ 17-27 What skills do accountants have to contribute to the development/acquisition process?

RQ 17-28 What roles can an accountant play in the development or acquisition of an AIS?

Discussion Questions

DQ 17-1 Discuss several factors affecting (negatively or positively) the achievement of systems acquisition and/or development objectives.

DQ 17-2 How might the absence of an organization's strategic plan for the information system affect the conduct of a preliminary survey? *Hint:* Discuss the potential difficulties of making preliminary survey decisions in the absence of each of the strategic plan components.

DQ 17-3 In doing a preliminary survey for the proposed automation of the cash disbursements system of Maui Company, the analyst in charge reached the tentative conclusion that

Clara Maxwell, the popular cashier with more than 30 years of company service, will be displaced and perhaps asked to consider early retirement. Discuss how this scenario relates to the concept of ''operational feasibility'' presented in this chapter. Discuss the potential impact on the success of the new disbursements system.

DQ 17-4 ''Choosing among renting, leasing, and purchasing an AIS is strictly a financial decision and should be done by the finance staff.'' Do you agree? Discuss fully.

DQ 17-5 ''A vendor would never propose a system that would not meet an organization's needs. Therefore, external validation of vendor proposals is not really needed.'' Do you agree? Discuss fully.

DQ 17-6 ''As long as we plan a systems development project and carry out the project in an orderly manner, we don't need a formal, documented systems development methodology.'' Do you agree? Discuss fully.

DQ 17-7 One of the tasks of systems analysis is to choose and justify one of the alternative design solutions. Would it not be more effective, efficient, and practical for the systems analyst to pass along *all* alternative design solutions to top management (perhaps to the IT steering committee), together with arguments for and against each alternative, and let top management choose one of them? After all, top management has a broad perspective that the systems analyst does not possess. Discuss fully.

DQ 17-8 Discuss the decisions that must be made prior to initiating structured systems analysis. Indicate how the systems survey contributed to the decisions.

DQ 17-9 Indicate whom you would include on a systems analysis project team in each of the following situations, and discuss fully the reasons why you would include each member you suggest.

a. A college's system for tracking students from ''cradle to grave'' (i.e., from the time that prospective students apply for admission until the time that alumni die and are listed in the ''In Memoriam'' section of the alumni magazine).

b. A bank's system for *integrating* the various, previously separate affiliations that it has with its customers.

c. A *materials requirements planning (mrp)* system, as discussed in Chapter 15. (*Note:* Discuss only if you have studied Chapter 15.)

DQ 17-10 Propose ways that the following intangible and/or indirect costs and benefits *might* be measured. Discuss whether they are indirect and/or intangible.

a. Decreased worker productivity.

b. Increased customer support in the form of improved product service and maintenance.

c. Increased customer support in the form of more timely and accurate responses to customer inquiries.

d. Deteriorated vendor relations, as evidenced by more stringent credit terms offered by vendors.

e. Deteriorated vendor relations, as evidenced by longer lead times, poorer quality goods, and more frequent backorder situations.

f. Improved management decision making.

g. Improved competitive advantage.

DQ 17-11 ''If the results of the cost/benefit analysis do not agree with those of the effectiveness analysis, there is probably no difference among the alternatives.'' Discuss fully.

DQ 17-12 Discuss why the knowledge of documentation procedures (such as DFDs and systems flowcharts) is important to the acquisition/development process.

DQ 17-13 The Discovery Company requests bids from hardware vendors for specific configurations rather than bids for general performance objectives because it ‘‘knows what it needs.’’ Discuss fully.

DQ 17-14 ‘‘An organization puts itself at a disadvantage by asking only one vendor (versus asking several vendors) for a proposal for software or hardware.’’ Do you agree? Discuss fully.

DQ 17-15 Compare and contrast the efficiency and effectiveness of an in-house data center, an arrangement with an *outsourcing* vendor to own and operate a data center for us, a *service bureau*, and an *application service provider (ASP)*.

DQ 17-16 Assume that you are the manager of an accounts receivable department. How might you be involved in system testing? Discuss fully.

DQ 17-17 a. Which, if any, category of application systems *maintenance—corrective*, *perfective*, or *adaptive*—presents the greatest risk from a *control* standpoint? Explain.

b. Which, if any, pervasive control plans from Chapter 8 might be effective in controlling systems maintenance activities? What control plans other than those in Chapter 8 might be used? Discuss fully.

DQ 17-18 Give examples, other than those used in this chapter, of situations in which each of the three implementation approaches is most appropriate. Explain why that implementation approach is most appropriate.

DQ 17-19 Refer to the typical contents of a project completion report. Which parts of the report would be useful in performing a *post-implementation review*? Discuss fully.

DQ 17-20 Periodically, students comment that they are studying to be an accountant, not an IS worker. Discuss the flaw of such reasoning.

PROBLEMS

P 17-1 Conduct research of current literature and databases to find reports of systems development project failures. Prepare a report or presentation (subject to your instructor’s instructions) describing the failure. Include in your report, the elements of feasibility and project risk that may have been miscalculated or mismanaged and led to the project failure.

P 17-2 Conduct research of current literature and databases to find two application service providers (ASPs) that might be of interest to accountants. Prepare a report or presentation (subject to your instructor’s guidelines) that discusses the positives and negatives of each ASP. If you were the decision maker in a large, multinational manufacturing company, would you use either of these ASPs? Why or why not?

P 17-3 The OFFICE-ALL Company is a wholesale distributor of office supplies. It sells pencils and pens, paper goods (including computer paper and forms), staplers, calendars, and other items, excluding furniture and other major items such as copy machines that you would expect to find in an office. Sales have been growing at 5 percent per year during the past several years. Mr. Chair, the OFFICE-ALL president, recently attended a national office supplies convention. In conversations during that convention, he discovered that sales for OFFICE-ALL's competitors have been growing at 15 percent per year. Arriving back home, he did a quick investigation and discovered the following:

- OFFICE-ALL's customer turnover is significantly higher than the industry average.
- OFFICE-ALL's vendor turnover is significantly lower than the industry average.
- The new market analysis system was supposed to be ready two years ago but has been delayed for more than one year in systems development.
- A staff position, reporting to the president, for a person to prepare and analyze cash budgets was created two years ago but has never been filled.

Mr. Chair has called on you to conduct a systems survey of this situation. You are to assume that a request for systems development has been prepared and approved. The information system at OFFICE-ALL is much like that depicted in Chapters 10 through 16.

Make and describe all assumptions that you believe are necessary to solve any of the following:

a. What are the specific tasks of *this* systems survey?

b. Indicate specific *quantifiable* benefits and costs that should be examined in assessing the economic feasibility of any solutions that might be proposed. Explain how you would go about quantifying each benefit or cost.

c. Propose and explain three different scopes for the systems analysis. Use a context diagram to describe each scope alternative. *Hint:* What subsystems *might* be involved in an analysis?

P 17-4 For each problem described, list and explain the documentation you would recommend for gathering and analyzing related facts. *Note:* It is *not* necessary to simulate the documentation. Confine your answer to a listing and brief explanation (one to two sentences) for each type of documentation that you recommend.

a. The college admissions office is experiencing a decline in applications.

b. The college admissions office is experiencing a decline in the percentage of students coming to the college after being accepted by the college.

c. A company is experiencing an increase in the size of receivables.

d. A faculty member has noticed that fewer students are signing up for her classes.

P 17-5 Atlanta Insurance Agency is negotiating for the acquisition of computer equipment from York Corporation effective 1/1/XXXX. Atlanta has asked for your assistance in evaluating the available financing alternatives.

One alternative is to purchase the equipment outright for a unit purchase price (UPP) of $120,000 plus 5 percent sales tax, plus destination, unpacking, and installation charges estimated at $2,000. The estimated useful life of the equipment is five years, at the end of which, the salvage value is estimated at $8,000. If Atlanta purchases the equipment it will use straight-line depreciation over a five-year life for tax purposes (instead of the

EXHIBIT 17.4 Lease Arrangements for Problem 17.5

Initial lease term:	
Duration[a]	3 years
Monthly lease payment	$4,000
Payable	In arrears at end of each month
Renewal option terms (one-year renewal periods at election of lessee):	
Annual renewal rate as a percent of unit purchase price (UPP):	
Year 4	5 percent
Year 5	3.5 percent
Payable	Annually in advance
Option to purchase—at end of any lease anniversary date, starting with the second anniversary. Purchase option price is a sliding scale, based on UPP, as follows:	
Second lease anniversary	46 percent of UPP
Third anniversary	10 percent of UPP
Fourth and fifth anniversaries	Excluded from consideration[b]
Other charges borne by lessee:	
Destination, unpacking, and installation	$2,000 (estimated)

[a] The lease is written as a ''net'' lease, wherein the lessee pays for maintenance and casualty insurance. Because these annual expenses would also apply to the purchase alternative, they have been ignored.

[b] Although purchase options beyond the third-year anniversary are available. Whipper has excluded them from consideration.

MACRS method). Its marginal income tax rate is 40 percent. For simplicity, assume that the UPP and other out-of-pocket costs will come from existing working capital.

Leasing the equipment through the financing subsidiary of York Corporation is another possibility. The key provisions of the lease arrangements include those shown in Exhibit 17.4.

Required:

Use spreadsheet software to prepare a comparative analysis of the following financing alternatives, using an approach based on discounted cash flows. Show the *details of each* alternative in two columns: one column for nominal dollars and one for discounted amounts. Use a before-tax discount factor of 12 percent.

a. Outright purchase

b. Lease, with exercise of option to purchase at the end of year 2

c. Lease, with exercise of option to purchase at the end of year 3

d. Lease, with renewal at the end of year 3 and another renewal at the end of year 4

Whenever income tax calculations are required, assume that the cash savings from income taxes occurs at the *end of the year* in which the tax deduction occurs.

For alternatives b and c, as in the case of the purchase alternative, assume that the option purchase price will come from existing working capital. Also, for cost and personal property tax purposes, consider that the option purchase price is subject to a 5 percent sales tax.

Obtain hard copy printouts of both the results of the calculations and the spreadsheet formulas.

P 17-6 Using the Web sites listed in Technology Summary 17.2 (pg. 612) as a starting point, resolve the following issues:

a. Select sites (or parts of sites) that describe two similar software or hardware products. Write a summary that compares and contrasts the information provided about those products.

b. Select two sites that provide demos of a system. Write a report that compares and contrasts those demos in terms of the functionality and what you are able to learn about the system from the demo.

c. Select two sites that provide tests of a system. Write a report that compares and contrasts those tests in terms of the functionality and what you are able to learn about the system from the test.

P 17-7 Assume that you are working with a payroll application that produces weekly paychecks, including pay stubs. Listed on pg. 632 are 20 data elements that appear on the paycheck/pay stub. For each numbered item, indicate the immediate (versus ultimate) source of the item. For instance, the immediate source of the number of exemptions for an employee would be the employee master data, as opposed to the ultimate source, which is the W-4 form filed by the employee. Some items may have more than one source, as in the case of item 1. You have the following choices:

- E = employee master data
- T = time records (these are in machine-readable form and show, for each employee for each day, the time punched *in* in the morning, *out* at lunch, *in* after lunch, and *out* in the evening)
- H = table of hourly wage rates (i.e., wage rate ‘‘class’’ and hourly rate for each class)
- W = table of state and federal income tax withholding amounts plus FICA tax rate and annual ‘‘cutoff’’ amount for FICA wages
- CG = computer generated (such as a date or time of day supplied by the system)
- CC = computer calculated
- CO = console operator (such as batch totals or a date to be used)

Arrange your answer as follows:

Item No.	Source
1.	E, T
2.	?
etc.	

The items to be considered are as follows:

Number	Description
1.	Social security number
2.	Employee name
3.	Employee address
4.	Employee identification number
5.	Pay rate classification
6.	Regular hours worked
7.	Overtime hours worked
8.	Hourly pay rate
9.	Regular earnings
10.	Overtime earnings
11.	Total earnings
12.	Deduction for state income tax
13.	Deduction for FICA tax
14.	Deduction for federal income tax
15.	Union dues withheld (flat amount based on length of service)
16.	Net pay
17.	Check number (same number is also preprinted on each check form)
18.	Year-to-date amounts for items 11 through 14
19.	Pay period end date
20.	Date of check (employees are paid on Wednesday for the week ended the previous Friday)

P 17-8 Select an AIS system segment (such as purchasing, sales, or payroll), investigate, and write a report on the tangible and intangible costs and benefits associated with your selection.

P 17-9 The chapter discusses the role of accountants in the acquisition/development process. Select a role that you expect to play in the process by the time you are at mid-career. Determine the skills you currently have and those you need to play that role. Develop a plan to attain the missing skills. (In addition to the text, you should use the IFAC #11 document, as well as other resources.)