Under Joe Hopper’s leadership, Hopper Specialty Company had grown into the biggest distributor of industrial hardware in northwest New Mexico. To provide the systems infrastructure needed for future growth, Joe acquired an inventory management system from NCR. In addition to the NCR hardware and systems software, the system included the Warehouse Manager software from Taylor Management Systems. The system was supposed to track the thousands of items in the Hopper inventory, including prices and balances. However, had Joe thoroughly investigated and tested this system, he would have discovered that:

- Warehouse Manager had not worked anywhere on an NCR computer.
- When two terminals accessed the system at once, both terminals locked up.
- When the locked terminals went back online, information—including prices, item balances, and general ledger data—was altered.
- Sales at the counter were supposed to take fractions of a second. Actual response times were as high as several minutes.

How did Joe let this happen to his company? Joe needed to conduct a more thorough investigation of the vendors and of existing installations of the proposed system. And, prior to implementation, the system should have been tested in an environment that resembled the one in which the system was to be asked to operate.

**Synopsis**

In this chapter, we complete our systems development study by discussing and illustrating the remaining phases of systems development, the systems design, implementation, and operation phases. As with Chapter 6, one goal is to highlight the technology that will help us achieve systems development objectives as we conduct systems selection and systems design. For example, we consider the software and hardware testing that may be necessary during the systems selection process. We need always to keep in mind that a major purpose for undertaking systems development is to leverage existing technology to provide competitive advantage to the organization. We also see the aspects of these systems development steps that are most affected by the implementation of enterprise systems and e-business systems.
LEARNING OBJECTIVES

• To name the goals, plans, tasks, and results of systems design, implementation, and operation
• To be able to evaluate the advantages and disadvantages of alternative sources for computer software and computer hardware
• To describe the process of choosing computer software and hardware
• To explain the importance of implementation planning
• To name the interdependent tasks that must be accomplished during systems implementation
• To explain the importance of thoroughly testing the new or revised system prior to putting the system into operation
• To describe the dual functions of post-implementation review
• To explain the difficulties associated with systems maintenance

Introduction

Figure 7.1 (page 206) depicts the systems development life cycle, including the phases, steps, and inputs and outputs for each step. In Chapter 6, we began our study of the systems development process and described the first phase of systems development, the systems analysis phase. In this chapter we consider the remaining phases.

The systems design phase includes two steps: systems selection (choosing software and hardware resources to implement new or revised systems) and structured systems design (detailed specification of new or revised software and the preparation of implementation plans). The systems implementation phase includes completing the design of the new or revised system and converting to that system. The systems operation phase includes two steps: the post-implementation review, during which we assess the adequacy of the new system to meet the users’ requirements and the quality of the development process, and systems maintenance, which involves making minor system repairs and modifications.

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

Systems implementation (in which the computer programs are written, the design of databases, files, and documents is finalized, and the system is put into operation) is analogous to the process of actually constructing the industrial park. Post-implementation review, in which the organization checks to see that the system does what it was supposed to do, is similar to the building inspection that occurs soon after the industrial park is completed. Systems maintenance, in which system errors are corrected and enhancements are added to the system, is similar to undertaking plumbing or electrical repairs or minor building modifications, such as moving interior walls and relocating doors.
Figure 7.1 Systems Development Life Cycle

- 1.0 Systems survey
- 2.0 Structured systems analysis
- 3.0 Systems selection
- 4.0 Structured systems design
- 5.0 Systems implementation
- 6.0 Post-implementation review
- 7.0 Systems maintenance

Key Points:
- Request for systems development
- Approved feasibility document
- External sources
- System analysis
- System design
- System implementation
- System operation

- Request for proposal
- Approved configuration plan
- Vendor proposals
- Approved systems design document
- Hardware and software vendors
- Project completion report
- Maintenance completion report
- Post-implementation review report

- Physical requirements
- Logical specification
- Logical requirements
- Physical specification

- Available hardware and software
- Miscellaneous environmental information
- Available hardware and software
- Miscellaneous environmental information
As you can see in Figure 7.1, systems selection lies between structured systems analysis (bubble 2.0) and structured systems design (bubble number 4.0). 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 resources will be used to implement the new system. Only after the resources are chosen does detailed design begin.

**Systems selection** is a set of procedures performed to choose the computer software and hardware for an Information System. The *systems selection goals* are to:

- **Determine what computer software will implement the logical specification developed in structured systems analysis.** We must decide between in-house software development by Information Systems personnel, end-users, or contract programmers versus off-the-shelf rental or purchase. For instance, we should consider whether home grown systems can provide the level of integration and functionality that could be achieved through use of an enterprise system.

- **Determine what computer hardware will satisfy the physical requirements established in structured systems analysis.** We must evaluate and choose the architecture (e.g., client/server, LAN) and the type, manufacturer, and model of each piece of computer equipment.\(^1\) In making our choice, we should also be aware of the implications for security and control of Information Systems. Additionally, to understand cost implications, consideration should be given in the decision 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 an applications 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.

Before we proceed, let’s look at the sequence of activities presented in Figure 7.1 and in these goals. Historically, the logical specification and the physical requirements were developed in the systems analysis step after the business processes had been documented and accepted, or remodeled (e.g., business process reengineering). Then, a software package would be chosen (and modified, if necessary) or developed in house. This is the sequence depicted in Figure 7.1 and used in this text. As we said in Chapter 6, however, when we implement an enterprise system we change the sequence of activities. With enterprise system implementations we start by choosing the package and then retrofit business processes to match those required by the enterprise system. So, while we present the sequence of the *SDLC* as “typical,” we ask you to be aware of the existence of practical variations in these activities.

---

\(^1\) 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.
Before proceeding to the intermediate steps in systems selection, let’s spend time examining various software and hardware procurement options that an organization must consider. Computer software can be purchased, rented, leased, or developed in-house. 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 7.1 compares these external and internal sources for computer software and hardware.

A review of Table 7.1 should lead you to conclude that external sources usually provide more capacity and affect the organization’s resources less, while internal sources are better for software that may provide a competitive advantage.

<table>
<thead>
<tr>
<th><strong>Table 7.1 Internal versus External Software and Hardware Sources</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Internal</strong></td>
</tr>
<tr>
<td><strong>Software</strong></td>
</tr>
<tr>
<td>Can be developed to meet all user needs, but may take longer than would a purchase.</td>
</tr>
<tr>
<td>Two company assets are built: the software and the experience of the development team.</td>
</tr>
<tr>
<td>Initial costs are higher, but cost is difficult to assess because 80% of the cost of a system over its life is for maintenance.</td>
</tr>
<tr>
<td>Outcome of the development process is uncertain.</td>
</tr>
<tr>
<td>We don’t know if the software will be delivered on time, within budget, and with required functionality.</td>
</tr>
<tr>
<td>Ongoing support and maintenance must be provided in-house and staffing may not be adequate.</td>
</tr>
<tr>
<td>Better option for software that may provide a competitive advantage.</td>
</tr>
<tr>
<td>Can control the development process.</td>
</tr>
<tr>
<td>Can ensure compatibility with existing and future applications.</td>
</tr>
<tr>
<td>Can adapt software to changing needs.</td>
</tr>
<tr>
<td><strong>Hardware</strong></td>
</tr>
<tr>
<td>Can determine level of control, security, and privacy.</td>
</tr>
<tr>
<td>Management and staff must be in-house.</td>
</tr>
<tr>
<td>Capacity limited.</td>
</tr>
<tr>
<td>Costs are mostly fixed.</td>
</tr>
<tr>
<td>Tailored to our needs.</td>
</tr>
</tbody>
</table>
sources can be matched more easily with the organization’s needs. Organizations implementing e-business Web sites must balance the imperative to launch a site rapidly (i.e., purchase an off-the-shelf e-commerce suite) with their desire to tailor a site to their needs and present unique content and performance characteristics to distinguish their site from those operated by their competitors (i.e., develop their own proprietary site).

Software Acquisition Alternatives

Internal Development Software can be developed internally, purchased or rented from a vendor, or, in some cases, leased from a third party. Table 7.1 lists criteria useful in making the develop-versus-buy decision. Within the organization, a systems development staff usually writes programs that are large and complex and involve a number of organizational units. Software development by users normally is appropriate when the program will be used by a small group or an individual, and must be tailored to that limited use. As more software becomes available, especially enterprise systems with many modules and add-on features, the option of in-house development has become increasingly rare.

External Acquisition Organizations not wishing to or unable to develop software in-house may purchase, rent, or lease a commercially available software package. Some organizations have rented software packages and used them to benchmark software being developed in-house. The rented software may also provide an interim solution while a system is developed in-house and might be retained on a long-term basis if it proves superior to the in-house solution.

Software can be acquired from computer manufacturers, software vendors, mail-order houses and retail stores (for personal computer software), as well as outsourcing firms, including service bureaus, systems integrators, and application service providers.

In general, outsourcing is the assignment of an internal function to an outside vendor. An organization can outsource its accounting, payroll, legal, data processing, strategic planning, or manufacturing functions, and so on. Since 1989, when the Eastman Kodak Company signed an agreement with IBM whereby IBM would own and operate Kodak’s data centers, outsourcing has increasingly been an option for organizations willing to have an outsider provide some or all of their Information Systems services. The Gartner Group asserts that “outsourcing has become a noncontroversial, mainstream approach to managing IT . . . and senior executives (IT and non-IT) endorse and promote its judicious use within multiple processes across their enterprise.” These projections show that, in addition to software, outsourcing is also a major alternative for the external acquisition of hardware and networks.

A service bureau is a firm providing information processing services, including software and hardware, for a fee. The service bureau owns and manages the software and hardware, which is installed on the service bureau’s property. Most of the services are on a fee-for-service basis, thus minimizing cost to the contracting organization. Many companies contract with service bureaus for payroll processing.

Review Question

What are reasons for developing software internally versus acquiring it from external sources?

Review Question

What are the external sources of software?

---

Systems integrators are consulting/systems development firms that develop and install systems under contract. EDS, Accenture, CAP Gemini, and other professional services firms are major players in systems integration. Advantages of systems integrators and consultants to develop systems include:

- Consulting firms have broad experience and knowledge of specialty and leading edge technology, helpful if a project involves a substantial upgrade in technology beyond available in-house expertise.
- Consulting firms have experience and may specialize in organization change, helpful to organizations not accustomed to change.
- Quick action must be taken to catch up with aggressive competition. Consultant expertise is flexible and rapid, and usually readily available for required services.

There is evidence, however, that the use of systems integrators doesn’t always work out. For example, one study looked at 16,000 IT projects and found that none of the projects that had heavy participation by big systems integrators was completed on time and within budget. Technology Excerpt 7.1 proposes seven steps to preventing these disasters.

### Technology Excerpt 7.1

**Guidelines for Effective Use of Systems Integrators**

Here are seven steps that may help prevent the disastrous situations associated with IT projects led by systems integrators (SI).

1. Before submitting a request for proposal, define the key project objectives and measures of success. If these can’t be defined, don’t proceed.
2. Review the SI’s proposal to determine that they have conducted due diligence in developing their proposal and that the project sponsors—IT and business process owners—understand the proposed solution.
3. Break the project into chunks of six months and tie contract payments to specific milestones.
4. Ensure that the roles and responsibilities of the SI and sponsoring team members have been defined and that the qualifications of these team members have been determined.
5. At project kickoff, determine high-risk factors and contingency plans to deal with those factors.
6. The project sponsor must meet regularly with the project manager to determine that the project is progressing as planned.
7. Before completing the project, ensure the achievement of project objectives and milestones.


---

A rapidly developing segment of the outsourcing market is the application service provider (ASP). Technology Insight 7.1 discusses ASPs. ASPs are similar to service bureaus and other outsourcing options. The ASP, however, provides its services via an easy-to-use Web browser over public networks, rather than more expensive private lines. Several ASPs exist that specialize in providing enterprise system services to organizations. As the current generation of enterprise systems moves to Web-enabled clients that function through use of browser software, the usability of enterprise systems in an ASP environment should become less complex and make this model of delivery even more cost effective.

The Gartner Group predicted that only 20 percent of the ASPs in existence at the end of 2000 would survive through the end of 2003. Therefore, an organization should consider carefully the type of applications that it outsources. The outsourcing of critical applications should probably be kept in-house. They may be too important to hand over to another organization. Support applications such as human resources and accounting might be better outsourcing candidates.

**Summary of Software Acquisition Alternatives** An organization should consider the financial implications of the decision to develop (make) versus buy. Because software vendors can allocate software development costs across many products and across multiple copies of each product, the prices they charge to recover development costs are usually less than the organization would pay to develop the package in-house. Generally, software developed in-house for a mainframe computer can cost up to 10 times more than purchased software. And, annual maintenance of in-house software is typically 50 percent of the development cost, while

---

**TECHNOLOGY INSIGHT 7.1**

**Application Service Provider (ASP)**

An application service provider (ASP) is an external organization that hosts, manages, and provides access to application software and hardware over the Internet to multiple customers. The fee is typically a rental based on usage, similar to the rental pricing model used by service bureaus. ASPs, like most external sources of software and hardware, relieve the organization of the burden of developing (or even buying) and installing software and hardware. Because ASPs are accessed over the Internet, a user needs only a Web browser and an inexpensive PC or Internet appliance to obtain the ASP service.

When using an ASP, a user obtains consistently updated software. The user does not need to install or update software on a client or a server and does not need to hire technical staff to operate the application. ASPs are a good choice for noncritical, niche applications such as human resources, employee travel and expense reporting, and disbursement, although some companies use them for their complete enterprise system solutions.

---

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.

What is the bottom line from Table 7.1? When a suitable standard package exists, buy it rather than try to reinvent it in-house. Notice the emphasis on suitable. Other considerations must include the organization’s internal resources (personnel, capital) and available vendor support. Enterprise systems, for example, are off-the-shelf packages that are highly configurable, but still require compromises to benefit from the off-the-shelf nature and cost savings.

Many external sources of software, such as ASPs, require little up-front implementation expense and provide the benefits of some of the best software solutions available. Even when providing enterprise systems services, ASPs tend to shorten implementation time for an organization drastically, albeit limiting the amount of tailoring that can be done to match a system with a business process. On the other hand, ASPs can provide business process reengineering specialists to help organizations implement best practices into their business processes and to match the processes with the enterprise system provided. In general with ASPs, implementation, operations, and maintenance requirements are minimal, freeing organization personnel time to focus on their mission.

Contracted, in-house development is a software development option that combines some advantages of both in-house development and software purchase. An organization hires contract (nonemployee) analysts and programmers to develop a system. Because the services of these persons end at the completion of the project, the contracting option provides short-term labor and expertise that the organization requires. Some organizations use contractors to train their personnel to work with new technologies. When benefits are figured in, the contractors cost about the same as employees. These contractors can be on-site or they can be outsourced from anywhere in the world. For example, outsourcing of systems development services is a $6.2 billion industry in India. Services provided include maintenance of existing (i.e., legacy) systems, e-business development, and integration of applications.5

**Hardware Acquisition Alternatives**

Computer hardware can be purchased, rented, or leased from the manufacturer (vendor) or from a leasing company. Under these arrangements, hardware is acquired, installed in the organization’s facilities, and operated by the organization’s personnel. As noted in Table 7.1, possession and management by the organization is less flexible (because of fixed cost and limited capacity, for example) than is use of external sources, but it does permit the organization to control and tailor the system. An organization preferring not to own or manage its own computer facilities can use one of the outsourcing options described above, such as a service bureau or an ASP, to fulfill its hardware needs.

---

The Intermediate Steps in Systems Selection

Review Question
What is a request for proposal (RFP)?

Review Question
What are the factors an organization must consider in structuring the RFP and deciding to whom the RFP will be sent?

There are three major tasks in the systems selection process: prepare requests for proposals, evaluate vendor proposals, and complete configuration plan. We will now discuss each of those tasks.

Prepare Requests for Proposal

A request for proposal (RFP) is a document sent to vendors that invites submission of plans for providing hardware, software, and services. The organization may send an RFP to vendors from whom it has previously received proposals or with whom it has previously done business. 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, an example of external literature review, is described in Technology Insight 7.2. 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 7.1 (page 214) lists typical contents of an RFP.

Sources of Vendor Information

Analysts use 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 Market Intelligence 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 also provides a wealth of information directly from vendors. For example, a quick tour of the Web found sites for Symantec (network security, virus protection, etc. at smallbiz.symantec.com), IBM (http://www.ibm.com), Microsoft (http://www.microsoft.com), SAP (http://www.sap.com), J. D. Edwards (http://www.jdedwards.com), and Gateway (http://www.gateway.com). Through such sites, one could obtain news about upcoming products, lists of existing products, customer support, technical support, software purchases, and software fixes and upgrades.

Technology Insight 7.2
Review Question
What might be included in an RFP for software? In one for hardware?

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 software and hardware that can satisfy that long-term demand, the longer it will be before the system needs to be revised and new software and hardware obtained. This principle may be less relevant to industries in which organizations need to apply rapidly evolving technology in order to remain competitive.

Evaluate Vendor Proposals
Using vendor responses to the RFP, the logical specification, and the physical requirements, analysts must decide which, if any, proposal best meets the organization’s needs. The process of evaluating the vendor proposals includes three steps:

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 system, including, if applicable, the logical specification and physical requirements, which in turn include specification 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
   - Installation schedule
   - Cost
   - Ease of use
   - Compatibility with existing software
   - Expandability
10. Miscellaneous information, such as:
    - Conversion plans
    - Staff requirements
    - Financing alternatives

Exhibit 7.1 Typical Contents of a Request for Proposal
Review Question
What is the difference between a specification and a performance measure?

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

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 software, hardware, and services that best meet the organization’s requirements.

**Validate Vendor Proposals** The first evaluation step is to validate the vendor proposal to assess whether the system (software or hardware) does what the organization requires by studying a proposed system’s specifications and performance. Specifications are straightforward descriptions of the hardware and software—such as a software package’s maximum table sizes or a printer’s speed—that can be examined to determine whether the system has the ability to perform as required. Performance features can be determined only through testing, measurement, or evaluation and often include items such as user friendliness, vendor support, quality of documentation, reliability, and ability of system to produce complete, accurate, and timely output.

One commonly used method for measuring system performance involves measuring the system’s throughput, which is 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. Technology Excerpt 7.2 (page 216) describes some performance factors to be considered when choosing an ASP.

After eliminating those proposals that do not meet minimum requirements, the evaluation team tests the remaining systems to determine the accuracy of the vendors’ specifications and how well the equipment will work for the organization. Having determined what a system is, we test to determine what that system can do.

An evaluation team can test a system by:

- Varying input (workload) parameters, such as quantity, timing, and type of input.
- Varying system characteristics (parameters), such as quantity and size of data storage devices.
- Varying the factors being measured, such as CPU cycle time (a system parameter) or execution time (a performance measure).
- Testing an actual workload, such as a weekly payroll, or testing a workload model that is representative of the workload.
- Testing the actual system or a model of the system.

---

6 In a 1997 survey, financial and information technology executives rated superior vendor service as more important than the latest technology as a criterion for software selection. *Financial and Accounting Survey Results* (Deloitte & Touche LLP and Hyperion Software, 1997): 8.

7 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.
The Internet has made it possible for vendors to demonstrate their software on their Web sites. For example, at mySAP.com (http://www.mySAP.com) you can see demonstrations of and “test drive” SAP’s R/3 enterprise system. This site allows prospective buyers to evaluate the system’s capabilities and to identify the modules that will best meet their needs. Other vendors providing similar services include Microsoft’s Great Plains (http://www.greatplains.com), where you can test their enterprise system, and SBT (http://www.sbtcorp.com), where you can test their ACCPAC eTransact system.

Consider Other Data and Criteria  Rather than estimate vendor and system performance internally, the evaluation team can interview users of the vendors’ products and visit those sites to witness the system in action. Quite often, vendor presentations are made at the site of an existing user. **External interviews**—interviews conducted with personnel outside the organization—can provide valuable insights into vendor performance. Where appropriate, questionnaires can also be used to gather information from users. The following information might be collected from users:

- Were there delays in obtaining the software or hardware?
- Did the system have bugs?
- How responsive is the vendor to requests for service?
- Was the training the vendor provided adequate?


---

As mentioned in Technology Insight 7.2, there are several services that publish technical reviews, user surveys, and expert commentary on computer equipment, software, and a variety of related topics. The reviews and user surveys can be helpful when evaluating proposals.

A cost/benefit analysis is often used to determine the economic viability of the remaining vendor proposals. Quantifiable costs and benefits are summarized to determine whether vendor proposals can be justified economically. Ranking vendor proposals using the economic criteria is useful in the next step in systems selection, in which the evaluation team suggests which vendor proposal should be chosen. The identification and quantification of Information Systems costs and benefits, however, is a difficult process requiring as much art as science. Still, we must have some data with which to make a decision.

**Suggest Resources** At this point, the study team recommends one vendor proposal. Management then chooses the software and hardware resources. To recommend one vendor, the evaluation team compares the proposals that have not been eliminated. The evaluation team might list the relevant criteria and indicate the performance of each vendor on each criterion.

**Complete Configuration Plan**

To complete the configuration plan—the major output from the systems selection step in SDLC—the evaluation team must complete the software plan, complete the hardware plan, and obtain approvals. As with many of the steps in systems development, these processes are not necessarily sequential, but should be iterative.

The software plan documents how the logical specification will be implemented, using in-house development, vendor purchase or lease, ASP, or a combination of these. The hardware plan summarizes how the recommended vendor proposal will fulfill the physical requirements specified in structured systems analysis.

Once the configuration plan (i.e., the combined software and hardware plans) is completed, it must be approved by the Information Systems steering committee, IT management, internal auditor, the controller, legal counsel, and other appropriate management personnel. Once approved, the configuration plan is used in the next step in systems development: structured systems design.

**Introduction to Structured Systems Design**

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. Structured systems design also 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.

Refer to Figure 7.1 (page 206) to see that structured systems design is the fourth major step in the development of an Information System. **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. Structured systems
design is often called *detailed design* or *internal design*. 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.

The *structured systems design* goals are as follows:

- *Convert the structured specification into a reliable, maintainable design.* This is similar to the process of converting a building model into a 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.* Personnel must know how to use the new system effectively, and the information processing staff must know how to operate the system.
- *Develop a program that ensures that users and support personnel are adequately trained.*

### The Intermediate Steps in Structured Systems Design

The sequence of activities and the amount of effort expended for each activity in structured systems design differs depending on some of the decisions made earlier in the systems development process. For example, if the organization has chosen to install an enterprise system, the design steps will include reengineering of the business processes and specification of how the enterprise system will be configured to match those processes. If the organization has chosen to develop the software in-house, the structured systems design step includes design of the software that will be written during the implementation step.

### Specify Modules

If the software is to be developed in-house, it is at this point in the development process that we must specify the software design (i.e., detailed, internal design). The modular design of the software is one of the features unique to *structured systems development*. The main tool of the structured design process is the *structure chart*, a graphic tool for depicting the partitioning of a system into modules, the hierarchy and organization of these modules, and the communication interfaces between the modules.9

---

The structure chart’s overall appearance is similar to that of an organization chart. Each box on a structure chart is a module. These structure chart modules become computer program modules of 30 to 60 lines of computer program code (one-half to one page of code). During structured systems design, related activities are grouped together within a module. This grouping of activities leads to a more maintainable system because changes to one function of a system lead to changes in a minimum of modules.

Develop Implementation Plan and Budget

System designers possess valuable insights into how a system should be implemented. During the design phase, the project team documents these insights in an implementation plan. As the implementation plan evolves, the project team summarizes resources required to implement the new system into an implementation budget so that resources can be allocated and implementation tasks scheduled.

If the software is to be developed in-house, the structure chart, developed by the system designer, dictates which modules should be programmed and installed first, and this sequence becomes part of the plan. The systems designer uses the expected size and complexity of the computer programs to prepare a schedule and budget for the programmers required to write the program code.

Develop Implementation Test Plan

Each system module, and any interactions between modules, must be tested prior to implementation. Again, systems designers have valuable insights into how a system should be tested. As we saw in the discussion of structure charts, the inputs and outputs for each module (and module combination) are specified in the design phase so that the designers can specify test inputs and expected outputs and provide recommendations for the order in which the system’s pieces should be tested. Examples of test control strategies are found in Technology Application 7.1 (page 220).

Develop User Manual

Because the designer knows how the system and each program will operate, how each input should be prepared, and how each output is used, preparation of the user manual can begin in the design phase. At this point in the SDLC, the manual is used to begin briefing and training users. The development of the user manual proceeds concurrently and interactively with many other design activities. For example, user procedures usually depend on computer system procedures, but some system functions may depend on user procedures. Development team members must also know about user procedures so that they can design tests of those procedures.

Develop Training Program

User training should begin before the system is implemented and therefore must be planned during the design phase. Deciding when to conduct training is tricky. While training must be conducted before implementation, it cannot be too much before or trainees may forget what they learned. Training materials, user manuals, and the system used for training must also be consistent with the system actually implemented.
The approved systems design document has three main components: (1) the system design (structure charts and descriptions of logical processes); (2) the implementation, testing, and training plans; and (3) the user manual. The design project leader must assemble these components 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, IS management furnishes a supervisory/technical approval of the adequacy of the software specifications. Auditors ensure adequacy of the controls and the design process (including implementation planning).
Introduction to Systems Implementation

At this point in the SDLC, we have completed the systems analysis phase (the *systems survey* and *structured systems analysis*). We have also completed the systems design phase by selecting hardware and software (*systems selection*) and by preparing the systems design and the implementation plan (*structured systems design*). It is time to install and begin to use our new or modified system.

**Systems implementation** is a set of procedures performed to complete the design (as necessary) contained in the *approved systems design document* and to test, install, and begin to use the new or revised Information System. Figure 7.1 (page 206) depicts systems implementation as the fifth major step in the development of an Information System.

The *systems implementation goals* are as follows:

- **Complete as necessary the design contained in the approved systems design document.** For example, the detailed contents of new or revised documents, computer screens, and database must be laid out and created.

- **Write, test, and document the programs and procedures required by the approved systems design document.**

- **Ensure, by completing the preparation of user manuals and other documentation and by 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.**

In this section we describe implementation approaches that can be taken to install the new or modified system. Figure 7.2 (page 222) depicts the three most common implementation approaches.

Figure 7.2(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, whether to terminate the operation of the old system. The parallel approach provides more control because the old system is not abandoned until users are satisfied that the new system adequately replaces the old. Although this approach makes good intuitive sense, in practice it frequently alienates users who perceive parallel operations as doubling their workload.

Figure 7.2(b), the **direct approach**, is often called the “Big Bang” approach and is the riskiest of the three approaches. At time $x$ the old system is stopped and the new system cuts in with no validation that the new system operates comparably to the old. While we will see a little later that it need not be so, enterprise systems are often implemented using this approach. Sometimes, as you’ll see in the Hershey story in Chapter 11, direct implementations can lead to disaster.
Figure 7.2(c), the modular approach, can combine parallel or direct approaches to tailor the implementation to the circumstances. With the modular approach, the new system is either implemented one subsystem at a time or is introduced into one organizational unit at a time. For example, a new Order-to-Cash 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 7.2(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.
Figure 7.3 depicts the modular schedule used at the Boston Scientific Corporation to implement SAP at all of its worldwide divisions and locations. As shown in this example, several installations were complete while two more were scheduled for the end of March. At Boston Scientific, several members of the project team were on location on each worldwide “go-live” date to provide assistance, ensure consistency of all implementations, and to learn and provide improvements for subsequent implementations.

The Intermediate Steps in Systems Implementation

As with structured systems design, the sequence of activities and the amount of effort expended on each depends on some of the decisions made earlier in the development process. For example, if the organization has decided to install an enterprise system, software and hardware may have been purchased and installed at this point in the development process. Also, if the software has been acquired (vs. developed in-house), as with an enterprise system, the only programming likely required would be to connect the enterprise software to any remaining legacy systems (i.e., old systems being retained).

Complete the Design

During systems implementation, we need to complete the detailed design of the new or revised systems. This may sound a little confusing. Didn’t we perform structured systems design in the previous development step? Yes, we did. But that design was related to the design of software that was to be developed. Now we must design input and output reports, documents, computer screens, database, manual processes, and certain computer processes, such as those needed to link new software to legacy systems.
Acquire Hardware and Software

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 site preparation are important parts of the computer acquisition process. Technical, legal, and financial expertise must be combined to negotiate and execute the contracts. Business process owners should review contracts to ensure that important user requirements, such as system availability and response times, are reflected. Detailed specifications protect both buyer and vendor. Technology Excerpt 7.3 provides some contract preparation guidelines.

The site to receive the computer equipment must be prepared carefully. 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 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.

Technology Excerpt 7.3

Guidelines for Preparing Contracts for Computing Resources

The following guidance regarding contracts for computer hardware, software, and computing services comes from experienced users of IT.

- Be cautious of a vendor contract that goes to great lengths to tell you what the vendor won’t do.
- Be clear on what is being provided by the vendor, including measurable service levels, such as availability.
- Obtain vendor warranties for intellectual property infringements, third-party indemnification, and nonconforming services. Determine the remedies for failure to meet contracted obligations.
- For a consulting engagement, include the names of the people who will work on the project and set a maximum turnover rate.
- Include a detailed project plan that lists what will be delivered, when it will be delivered, and how it will perform.
- Tie payments to completion of project phases and acceptance of deliverables, such as software, hardware, documentation, and training.
- Obtain the services of a procurement professional to ensure consistency across contracts and to provide an independent viewpoint in contract negotiations.
- If you want to make changes to source code, or have a third-party make changes, include this right in the contract.

Write, Configure, Test, Debug, and Document Computer Software

The next task in systems implementation is to produce or configure the software, test and debug the software, and complete the software documentation. For internally developed systems, the programming step is important because the programming task in systems development consumes more resources and time than any other development task.

If we have purchased a software package, 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 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. This is the point at which we ensure matching business processes that we have reengineered within the organization to elements of the enterprise system. For example, we can configure the SAP system to create and record a customer invoice automatically upon shipment of the goods, or we can require that the billing process be triggered later.

Some programming may remain, however. To tie the new enterprise system modules to legacy systems, program code must be written—in ABAP for SAP and C++ for J. D. Edwards, for example.

Select, Train, and Educate Personnel

The organization must choose personnel to use the new system and train them to perform their system-related duties. The system’s users must be educated about the new system’s purpose, function, and capabilities. Such training becomes even more important if jobs have been redesigned during business process reengineering. Training may come from 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. 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.

Computer-based training (CBT) provides learning via computer directly to the trainee’s computer screen. Training may be delivered over the Internet by vendors whose business it is to design, produce, and deliver such training. Enterprise system vendors, such as SAP and J. D. Edwards, have created extensive CBTs to help users learn the features of their systems. CBT can be much less expensive than lectures, and it also permits individualized instruction, which can take place when and where needed. The interactive nature of CBT can get and keep a trainee’s attention. However, some employees, particularly middle and senior management, prefer more personal, traditional delivery methods.

Complete User Manual

A user manual should describe operating procedures for both manual and automated systems functions. The manual should cover user responsibilities, system inputs,
computer system interfaces, manual files and databases, controls (including error
detection and correction), distribution and use of system outputs, and manual and
automated processing instructions. Good user manuals can improve system effi-
ciency and effectiveness. If users know how to use a system properly and they em-
ploy it willingly, the system will be used more frequently, more correctly, and more
productively.

The systems designer, the user, and the organization’s technical writing and train-
ing staff should cooperate in preparing the user manual. Because the systems designer
knows intimately what the system will do, he or she is well qualified to describe how
to use the system. The user, who must study the manual to learn the system and then
keep the manual as a reference for continued operation of the system, must make
sure that the manual is relevant for the tasks to be performed and that it is complete,
accurate, and clear.

The organization’s training staff should be involved in preparing the manual be-
cause they must train users to operate the new system. The staff must learn the sys-
tem themselves and develop separate training materials and/or use the user manual
as the training vehicle. Therefore they are very interested in the user manual and
should have input to its development.

**Test System**

Beyond testing program modules, the entire system is tested to determine that it meets
requirements established by business process owners and users, and that it can be
used and operated to the satisfaction of both users and system operators. Testing is
carried out by systems developers, by developers and the users together, and finally
by users. The more closely the test can simulate a production environment (e.g., peo-
ple, machines, data, inputs), the more representative the test will be and the more
conclusive the results.

Several types or levels of tests are usually completed before a system can be im-
plemented. From the users’ point of view, three of these tests are the most impor-
tant. 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 assis-
tance 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 ade-
quacy 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 en-
vironment. This final test determines whether new equipment and other factors in
the environment—such as data entry areas, document and report deliveries, tele-
phones, and electricity—are satisfactory.

As noted earlier, enterprise systems are often implemented using a direct (Big
Bang) approach. Successful implementations often involve extensive testing. For ex-
ample, before implementing SAP at Lucent Technologies, Inc., more than 70 busi-
ness users tested the system for six months. At the Gillette Company, 150 workers
ran test transactions for four months.10

---

Obtain Approvals

The project completion report—the systems implementation deliverable—is approved as follows:

- Users verify that the system, including the user manual, meets their requirements. Users also approve conversion and training plans to confirm that these plans are adequate.

- IT confirms that the system has been completed and that it works. IT also approves the training and conversion plans. Finally, IT performs a technical review of the system to determine that acceptable design and programming standards have been applied.

- Management reviews the systems performance objectives, cost, and projected benefits to ensure that implementation is consistent with the best interests of the organization.

- IT audit compares test results with the original system requirements and specifications to determine that the system has been tested and will operate satisfactorily. IT audit is also interested in the adequacy of controls within the system and the controls identified for the conversion process.

Conduct Conversion

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

As existing data are mapped into the new system, exception-reporting situations must be devised to ensure that data are converted accurately. Users must suggest control totals that can be used to test completeness and accuracy of data conversion. For example, the total number of inventory items, the total on-hand quantity for all inventory items, or a hash total (described in Chapter 9) of inventory item numbers might be used as totals.

Boston Scientific (see Figure 7.3, page 223) implemented SAP over the course of two years at each of its worldwide divisions and locations (i.e., a modular approach). But since SAP was implemented using the direct approach at each of those locations, the data conversion was tested at least seven times, until there were no errors! The company believes that this testing was the key to the successful implementations.11

Both manual and computer-based processes must be converted. Conversion to new computer programs must be undertaken using program change controls (described in Chapter 8) to ensure that only authorized, tested, and approved versions of programs are promoted to production status.

The systems development project team now writes the project completion report, the final step in the implementation process. This report includes a summary of conversion activities and information with which to operate and maintain the new system.

An organization should periodically examine the system in its production environment to determine whether the system is continuing to satisfy users’ needs. If it is possible to make the system work better, its value to users will increase. There are three different types of periodic examination:

1. The post-implementation review is conducted to follow up a system’s recent implementation. This review is analogous to a follow-up examination that a doctor might perform after an operation.
2. Systems maintenance, performed in response to a specific request, is conducted if the system has a relatively minor deficiency. This examination is similar to one a doctor performs on sick people.
3. The periodic systems survey is undertaken whenever it is likely that the costs of the review will be less than the value of the improvements that the review will suggest. This reevaluation is like a periodic physical examination.

The Post-Implementation Review

The post-implementation review is an examination of a working information system, conducted soon after that system’s implementation. The post-implementation review determines 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 post-implementation review should be brief and inexpensive. Examinations conducted in response to a specific deficiency, systems maintenance, are discussed in the next section.

Post-implementation review goals are as follows:

- Determine whether users are satisfied with the new system.
- Identify the degree of correspondence between system performance requirements and the system’s achieved performance.
- Evaluate the quality of the new system’s documentation, training programs, and data conversions.
- Review the performance of the new system and, if necessary, recommend improvements.
- Ascertaining that the organization’s project management framework and SDLC were followed during development.
- Perfect the cost/effectiveness analysis process by reviewing cost projections and benefit estimations and determining the degree to which these were achieved.
- Perfect 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.

Consultants, IT auditors, or systems analysts (other than those who developed the system) may conduct the post-implementation review. The post-implementation review...
view is performed as soon as the system is operating at full capacity, which could be a month or a 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.

**Systems Maintenance**

**Systems maintenance** is the modification (e.g., repair, correction, enhancement) of existing applications. Systems maintenance expenditures can account for 50 to 70 percent of the total cost of a system over its total life cycle. For example, 80 percent of the total cost of software is in maintenance. 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.

Organizations often adopt the following procedures and controls for their systems maintenance process:

- Because systems maintenance is like 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, users 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 maintenance.
- 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 (see Chapter 8) 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 existing programs, and therefore cannot effectively or efficiently modify them.

Conclusions

Systems selection is a process that is central to the success of systems development. Recall that the first objective of systems development is “to develop information systems that satisfy an organization’s informational and operational needs.” For this reason, one key to the success of systems development is to ensure that systems selection criteria are based on user requirements (i.e., the logical specifications and physical requirements) developed during the systems analysis phase of systems development.

Another key to systems development success is the full evaluation of available software and hardware resources. As the quantity of resources has grown, it has become more difficult to identify and evaluate all available resources. On the other hand, the Internet has made available to us large quantities of up-to-date, independent information to assist in the selection process. Indeed, as noted earlier in the chapter, we can see product demos or actually conduct tests at many vendor Web sites.

Finally, the success of systems development projects is found in the details. It may be such things as user manuals, training, and implementation schedules and plans that determine the success of the new or modified system.

There may be, however, another twist on the cause-and-effect relationship between successful completion of systems development steps and the achievement of the systems development objectives. At the time we have implemented a system and conducted the post-implementation review, we might measure the development process as successful. That is, we have delivered a system that meets most of the user requirements, we have implemented the system on time and within budget, and there don’t seem to be any bugs. These are all short-term measures.

It is not until we conduct systems maintenance that we discover that the system has some long-term faults. It may not be, for example, flexible, scalable, reliable, or maintainable. These faults are what drive up the life cycle cost of the system, that cause maintenance costs to be 50 to 70 percent of the long-term costs. The solution is to incorporate the long-term requirements (e.g., flexibility, maintainability) into the initial user requirements and to measure the success of the implementation over the long run, rather than at the time of implementation.

REVIEW QUESTIONS

RQ7-1 What is systems selection?
RQ7-2 What are systems selection goals?

RQ7-3 What are reasons for developing software internally versus acquiring it from external sources?

RQ7-4 What are the external sources of software?

RQ7-5 What is a systems integrator?

RQ7-6 What is an application service provider (ASP)?

RQ7-7 What are reasons for using external and internal sources of hardware?

RQ7-8 What is a request for proposal (RFP)?

RQ7-9 What are the factors an organization must consider in structuring the RFP and deciding to whom the RFP will be sent?

RQ7-10 What might be included in an RFP for software? In one for hardware?

RQ7-11 What is the difference between a specification and a performance measure?

RQ7-12 What is structured systems design?

RQ7-13 What are the structured systems design goals?

RQ7-14 What is systems implementation?

RQ7-15 What are the systems implementation goals?

RQ7-16 What are the three major approaches to implementing an Information System?

RQ7-17 What is the post-implementation review?

RQ7-18 What are the post-implementation review goals?

RQ7-19 What is systems maintenance?

RQ7-20 Why is the management and control of systems maintenance so important?

DQ7-1 You are charged with recommending how to reengineer your company’s SDLC so that the typical 9-month development process could be cut to 6 weeks. Describe your new process. Which steps take on more or less importance in your new approach?

DQ7-2 As discussed in the chapter, there are companies that specialize in providing “contract” analysis and programming services to clients for a fee. Discuss the relative advantages to the client of using contract services versus other available alternatives.

DQ7-3 “There are enough software packages on the market today to preclude any organization needing to write another application program.” Do you agree? Why or why not?
DQ7-4  “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? Why or why not?

DQ7-5  “Nobody ever got fired for choosing IBM.” Comment on this statement in light of the information in this chapter.

DQ7-6  “A vendor would not propose a system that would not meet an organization’s needs. Therefore, ‘validation’ of vendor proposals is not really necessary.” Do you agree? Why or why not?

DQ7-7  “Surveys of existing users of software and hardware, such as those published by Dataquest Market Intelligence and Datapro, are biased. Only those users who are very happy or very displeased with their software and/or equipment respond to such surveys.” Do you agree? Why or why not?

DQ7-8  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).

DQ7-9  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.

DQ7-10 Assume that you are the manager of an accounts receivable department. How might you be involved in system testing?

DQ7-11 Refer to the story about Hopper Specialty Company at the beginning of the chapter. Describe, using several examples, how Joe Hopper would have avoided or minimized the problems that he encountered by following the systems development procedures described in this chapter.

PROBLEMS

P7-1  Read the following article and answer the questions that follow: Milo Geyelin, “Doomsday Device: How an NCR System For Inventory Turned Into a Virtual Saboteur,” Wall Street Journal (August 8, 1994): A1.

a. At the start of Hooper Specialty’s inventory system conversion project, how would you characterize it in terms of size, degree of definition, technology familiarity, and organizational readiness? Note: See Chapter 6 for a description of these concepts. Your answer should include several risks for Hooper.

b. Given the risks that you identify in answering part a, as well as the control principles discussed in this text, what measures could Joe Hopper have taken to prevent, detect, or correct problems with the NCR inventory system?

c. Next, consider the situation from the point of view of NCR. What risk did they face? What measures could they have put in place to prevent, detect, and correct problems with the inventory software they acquired from Taylor Management Systems?
P7-2 Using the Web sites listed in Technology Insight 7.2 (page 213) as a starting point, answer the following questions:

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.

P7-3 Obtain a computer operations “run manual” from an actual organization. Your college or university might be a source. Prepare a report that summarizes the contents of the run manual. Comment on the apparent reason for including each major item in the manual. If you are unsure of the reason for including certain material, interview the computer operations manager to determine the reason.

P7-4 The Boston Edison Company provides electric service to the residents and businesses of Boston and several other eastern Massachusetts communities. This problem concerns the following hypothetical billing procedures for Boston Edison.

Field personnel take electric meter readings 10 days prior to the end of each customer’s monthly billing cycle. These personnel key the meter readings into handheld units. In computer operations at the home office, the meter reading units are read by the computer, which also accesses the stored customer records and other necessary data. The quantity of kilowatts consumed and the amount due are computed, and the bill is printed (a sample bill is shown in Figure 7.4, page 234). Customers return the top half of their bills with their payments.

Because of the steady growth in number of customers and the increased need for managerial information, Boston Edison’s management has decided to upgrade its customer billing system. The new system will retain the present meter reading procedures, but the rest of the system will be modernized. The new system should also enable users to access customer records when desired and should provide improved information for decision making.

For each numbered item on the Boston Edison bill (Figure 7.4), indicate the immediate (versus ultimate) source of the item. For instance, the immediate source of the current meter reading would be the meter reading unit (i.e., event data), as opposed to the ultimate source, which is the meter itself. Some items may have more than one source. You have the following choices:

- C = customer records (a combination of customer and accounts receivable master data)
- CG = computer generated (such as a date or time supplied by the system)
- CC = computer calculated
- ED = event data
- CO = console operator (such as batch totals or a date to be used)
**Figure 7.4** Boston Edison Bill to Accompany Problem 7-4

**BOSTON EDISON**

Your consumer rights are explained on the back of this bill.

Pay at Customer Service Center or agency or mail to Box 488, Boston, MA 02199

---

<table>
<thead>
<tr>
<th>YOUR NAME</th>
<th>123 MAIN STREET</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANYTOWN, MA 00000-0000</td>
<td></td>
</tr>
</tbody>
</table>

---

**FOR SERVICE TO**

USE THIS BILL AS PROOF OF PAYMENT. PLEASE NOTIFY US 10 DAYS BEFORE MOVING.

---

**ELECTRICITY USED THIS PERIOD**

FEB 14 2003 METER READING: 16786
JAN 17 2003 ESTIMATED READ: 16006

KWH CONSUMED THIS PERIOD: 780

KWH/DAY - THIS PERIOD: 27.9

NEXT SCHEDULED READ APR 14, 2003

---

**SUMMARY**

PREVIOUS TOTAL BILL: $67.21
PAYMENTS AS OF FEB 17: $67.21
BALANCE: 0.00
CURRENT CHARGES: $73.55
AMOUNT NOW DUE: $73.55

---

**COST OF ELECTRICITY - RATE R1**

BASIC MONTHLY CHARGE: $5.83
ENERGY CHARGE: 780 KWH @ $0.056340: 43.95
FUEL CHARGE: 780 KWH @ $0.030480: 23.77
CURRENT CHARGES: $73.55
AVERAGE COST: $2.63 PER DAY

---

**KWH PER MONTH**

<table>
<thead>
<tr>
<th>FEB 03</th>
<th>780</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAN 03</td>
<td>778</td>
</tr>
<tr>
<td>DEC 02</td>
<td>681</td>
</tr>
<tr>
<td>NOV 02</td>
<td>561</td>
</tr>
<tr>
<td>OCT 02</td>
<td>508</td>
</tr>
<tr>
<td>SEP 02</td>
<td>714</td>
</tr>
</tbody>
</table>

---

**TWO NEW PAY STATIONS OPEN**

TWO NEW BOSTON EDISON PAY STATIONS ARE OPEN FOR YOUR CONVENIENCE! YOU MAY NOW PAY YOUR ELECTRIC BILL AT LOCAL OFFICES OF THE MEDWAY SAVINGS BANK DURING THE HOURS WHEN THE BANK IS OPEN. BANK OFFICES ARE LOCATED AT 1098 MAIN STREET, MILLIS, AND 81 MAIN STREET, MEDWAY. BOTH THE MEDWAY SAVINGS BANK AND BOSTON EDISON WELCOME YOU TO THESE PAY STATIONS!

---

**STATEMENT DATE**

FEB 17, 2003

**ACCOUNT NUMBER**

31 9048 961462

**METER NUMBER**

740270

**AMOUNT NOW DUE**

$73.55

---

**PLEASE RETAIN THIS PORTION FOR YOUR RECORDS**

---

**BOSTON EDISON**

800 Boylston Street, Boston, MA 02199
Arrange your answer as follows:

<table>
<thead>
<tr>
<th>Item No.</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>C</td>
</tr>
<tr>
<td>2.</td>
<td>?</td>
</tr>
<tr>
<td>3.</td>
<td>CG</td>
</tr>
<tr>
<td>etc.</td>
<td></td>
</tr>
</tbody>
</table>

**P7-5** Assume that you are working with a payroll application that produces weekly paychecks, including paystubs. Listed below are 20 data elements that would appear on the paycheck/paystub. 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:

<table>
<thead>
<tr>
<th>Item No.</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>T, E</td>
</tr>
<tr>
<td>2.</td>
<td>?</td>
</tr>
<tr>
<td>etc.</td>
<td></td>
</tr>
</tbody>
</table>

The items to be considered are as follows:

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Employee identification number</td>
</tr>
<tr>
<td>2.</td>
<td>Social security number</td>
</tr>
<tr>
<td>3.</td>
<td>Employee name</td>
</tr>
<tr>
<td>4.</td>
<td>Employee address</td>
</tr>
<tr>
<td>5.</td>
<td>Regular hours worked</td>
</tr>
<tr>
<td>6.</td>
<td>Overtime hours worked</td>
</tr>
<tr>
<td>7.</td>
<td>Pay rate classification</td>
</tr>
<tr>
<td>8.</td>
<td>Hourly pay rate</td>
</tr>
<tr>
<td>9.</td>
<td>Regular earnings</td>
</tr>
<tr>
<td>10.</td>
<td>Overtime earnings</td>
</tr>
<tr>
<td>11.</td>
<td>Total earnings</td>
</tr>
</tbody>
</table>
Number | Description
---|---
12. | Deduction for federal income tax
13. | Deduction for state income tax
14. | Deduction for FICA 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)

**P7-6**

Shown in Figure 7.5 is a flowchart that depicts the computer logic for updating *sequential* inventory master data for either of two types of events: goods received or goods issued.

Develop data to test the logic of the inventory update program. The test data should allow for all possible combinations of master data and event data records.

**Note:** There can be more than one event for a particular part number; be sure to provide for this possibility.

**P7-7**

Read the scenario below and answer the following questions:

Fleet Shoe Company is having problems with its automated distribution system. The main warehouse is almost at a standstill and retailers are getting few if any Fleet shoes. Fleet had received recognition for its state-of-the-art warehouse system. However, just prior to switching over to this new system, Fleet scrapped the system’s software and computer hardware and adopted a new architecture. During the development, there had been very high turnover of IT staff and Fleet had fired its lead systems integrator.

The new system was to automate the movement of goods in the warehouse and was to include tilting trays, conveyor belts, lifting equipment, and scanners. To operate properly, such systems require quite a bit of fine-tuning. The goal was to increase capacity, boost productivity, cut staff by 50 percent, and cut the time to get orders out the door to 24 hours. The software, not the hardware, seemed to be the problem. It was designed to run under UNIX, but Fleet decided to use fault-tolerant computers that run a proprietary (i.e., hardware specific) operating system. When the software vendor went out of business, they had not completed porting (i.e., transferring) their software to the proprietary operating system.

Fleet’s choice to replace the original platform was a computer system that itself ran warehouse management software. It is this option that brought Fleet to its knees. The new system was slower than expected. To get shoes to retailers, Fleet shipped directly from overseas factories and warehouses. Comments from industry specialists and consultants pointed out the chaos that often results from instantaneous changeovers. Another speculated that Fleet did not place much importance on warehousing and rather concentrated on other aspects of its operations.

a. How would you characterize this project in terms of size, degree of definition, technology familiarity, and organizational readiness?

b. Describe specific risks or concerns that you have for this project. Clearly explain why each is a risk or concern and the specific actions that you would recommend to mitigate the risk or concern.
Figure 7.5  Program Flowchart to Accompany Problem 7-6

Variables or Abbreviations:

\[ P_1 = \] Part # per master data (MD)
\[ P_2 = \] Part # per event data record
\[ T = \] Event code:
  1 = receipt
  2 = issue
\[ Q = \] Event quantity
\[ B = \] Balance on hand per MD
\[ EOF = \] End of file. Last record has dummy part #9999.

Start

Open database

Read first record from old MD

Read first event data record

Read next event data record

\[ P_2 > P_1 \] ?

Print new MD record

Read next record from old MD

\[ P_1 = 9999 \] ?

Print EOF dummy record on new MD

\[ B = B + Q \]

\[ B = B - Q \]

\[ T = 1 \] ?

Close database

Stop