CHAPTER

23

AIS Development Strategies

LEARNING OBJECTIVES

After studying this chapter, you should be able to:

1. Describe how organizations purchase application software, vendor services, and hardware.
2. Explain how information system departments develop custom software.
3. Explain why organizations outsource their information systems, and evaluate the benefits and risks of this strategy.
4. Explain how business process management, prototyping, agile development, and computer-aided software engineering can help improve system development.

INTEGRATIVE CASE Shoppers Mart

Ann Christy is elated that the system Shoppers Mart (SM) so badly needed was approved and that she and her team have accurately assessed company needs. Now Ann needs to determine whether to purchase the software, develop it in-house, or outsource system development and operation. More specifically, she needs answers to these questions:

1. Can Ann buy the software she needs? If so, how should she buy hardware and software and select a vendor?
2. How do companies develop software in-house? Is this the best approach for SM?
3. How extensively should SM use end-user-developed software?
4. Should SM improve its existing system or redesign its business processes and develop a system to support them?
5. Is outsourcing the information system a viable alternative to obtaining a new system? Do the benefits of outsourcing outweigh its risks?
6. If SM decides to develop the system in-house, should it use technologies such as business process management, agile development, prototyping, or computer-assisted software engineering?

Ann decided to investigate design alternatives to determine the best course of action for Shoppers Mart.

Michail Petrov/123RF

Introduction

Companies have experienced the following difficulties when developing an accounting information system (AIS):

- Development requests are so numerous that projects are backlogged for years.
- Users discover that the new AIS does not meet their needs. This occurs because users find it hard to visualize how the AIS will operate by reviewing design documentation and because developers who do not understand business or user needs find it hard to make meaningful suggestions for improvement.
- Development takes so long the system no longer meets company needs. Fannie Mae spent eight years and $100 million developing the world's largest loan accounting system. Unfortunately, it no longer met many of Fannie Mae's needs.
- Users do not adequately specify their needs because they do not know what they need or they cannot communicate the needs to systems developers.
- Changes are difficult to make after requirements are frozen. If users keep changing requirements, the AIS may take forever to finish.

In this chapter, you learn three ways to obtain an information system: purchasing software, developing software in-house, and hiring a company to develop and operate the system. You also learn four ways to improve the development process: business process redesign, prototyping, agile development technologies, and computer-aided software engineering tools.

Purchasing Software

In the early days of computers, it was difficult to buy software that met user needs. That is no longer the case. A Deloitte & Touche survey found that most chief information officers expect to replace their current systems with commercially available software packages. Many organizations, especially larger ones, purchase Enterprise Resource Planning (ERP) packages that integrate all aspects of a company's organizations. Chapter 2 discusses ERP systems in more depth.

Consider the following examples:

- Hard Rock Cafe purchased customer relationship software and mailed promotional offers to 225,000 customers. A year later, profits from the increased traffic paid for the new system.
- WellPoint Health Networks installed payroll, benefits, and human resources software so employees could manage their benefits, saving $400,000 a year.
- Pacific Gas & Electric responded to California's power deregulation by spending three years and $204 million installing the largest customer information system in the utility industry.

commercial software - Programs for sale on the open market to a broad range of users with similar needs.

turnkey system - Software and hardware sold as a package such that the vendor installs the system and the user "turns on the key"; often written by vendors who specialize in a particular industry.

Software-as-a-Service (SaaS) providers - Companies that rent cloud-based software applications that customers can access using the Internet.

Commercial software is sold to users with similar requirements.A **turnkey system** is software and hardware sold as a package. The vendor installs the system and the user "turns on the key." Many turnkey systems are written by vendors who specialize in a particular industry, such as doctors, auto repair shops, restaurants, and retail stores.

A major problem with commercial, or off-the-shelf, software is that it may not meet all of a company's information needs. This is overcome by modifying the software. About 90% of Dow Chemical's software has been modified to match its business processes. The rest was written in-house. It is best when the vendor modifies the software, as unauthorized modifications may not be supported by the vendor and may make the program unreliable.

Software as a Service (SaaS) providers, provide cloud-based software applications that customers can access using the Internet. This provides scalability as the business grows and global access to information. It automates software upgrades, allows companies to focus on core financial competencies rather than information technology (IT) issues, and can reduce software costs and administrative overhead.This is one example of cloud computing services described in Chapters 11 and 13.

Companies that buy AIS software follow the normal systems development life cycle (SDLC) except for the following:

- During conceptual systems design, companies determine whether software that meets AIS requirements is available and, if so, whether to buy it or create their own.
- Some physical design and implementation and conversion steps can be omitted. For example, the company usually does not need to design, code, and test program modules or document the computer program.

SELECTING A VENDOR

Hardware, service, maintenance, and other AIS resource decisions can be made independently of the decision to make or purchase software, although they may depend on the software decision.

Vendors are found by referrals, at conferences, in industry magazines, and on the Internet. Choosing must be done carefully because vendors with little experience, insufficient capital, or a poor product go out of business and leave their customers and products with no support or recourse. Problems can occur even when established vendors are selected. For example, when Texas selected IBM to consolidate data centers across the state, service levels dropped dramatically, and routine tasks took far too long to perform. The problem was attributed to poor project requirements and selecting the vendor with the lowest bid. IBM almost lost the contract after it failed to back up critical systems.

ACQUIRING HARDWARE AND SOFTWARE

request for proposal (RFP) - A request for vendors to bid on a system to meet a company's specified needs.

Companies that buy large or complex systems send vendors a **request for proposal (RFP)**, asking them to propose a system that meets their needs. The best proposals are investigated to verify that company requirements can be met. Using an RFP is important because it:

1. ***Saves time.*** The same information is provided to all vendors, eliminating repetitive interviews and questions.
2. ***Simplifies the decision-making process.*** All responses are in the same format and based on the same information.
3. ***Reduces errors.*** The chances of overlooking important factors are reduced.
4. ***Avoids potential for disagreement.*** Both parties possess the same expectations, and pertinent information is captured in writing.

RFPs for exact hardware and software specifications have lower total costs and require less time to prepare and evaluate, but they do not permit the vendor to recommend alternative technology. Requesting a system that meets specific performance objectives and requirements leaves technical issues to the vendor but is harder to evaluate and often results in more costly bids.

The more information a company provides vendors, the better their chances of receiving a system that meets its requirements. Vendors need detailed specifications, including required applications, inputs and outputs, files and databases, frequency and methods of file updating and inquiry, and unique requirements. It is essential to distinguish mandatory requirements from desirable features.

EVALUATING PROPOSALS AND SELECTING A SYSTEM

Proposals that lack important information, fail to meet minimum requirements, or are ambiguous are eliminated. Proposals passing this preliminary screening are compared with system requirements to determine whether all mandatory requirements are met and how many desirable requirements are met. Top vendors are invited to demonstrate their system using company-supplied data to measure system performance and validate vendor's claims. Table 23-1 presents hardware, software, and vendor evaluation criteria.

System performance can be compared several ways. A **benchmark problem** is an input, processing, and output task typical of what the new AIS will perform. **Point scoring** assigns

benchmark problem - Comparing systems by executing an input, processing, and output task on different computer systems and evaluating the results.

point scoring - Evaluating the overall merits of vendor proposals by assigning a weight to each evaluation criterion based on its importance.

TABLE 23-1 Hardware, Software, and Vendor Evaluation Criteria

Hardware Evaluation	Are hardware costs reasonable, based on capabilities and features? Are processing speed and capabilities adequate for the intended use? Are secondary storage capabilities adequate? Are the input and output speeds and capabilities adequate? Is the system expandable? Is the hardware based on old technology that will soon to be out of date? Is the hardware available now? If not, when? Is the hardware compatible with existing hardware, software, and peripherals? How do performance evaluations compare with competitors? What are the availability and cost of support and maintenance? What warranties come with the system? Is financing available (if applicable)?
Software Evaluation	Does the software meet all mandatory specifications? How well does the software meet desirable specifications? Will program modifications be required to meet company needs? Does the software have adequate control capabilities? Is the performance (speed, accuracy, reliability) adequate? How many companies use the software? Are they satisfied? Is documentation adequate? Is the software compatible with existing software? Was the software demonstration/test-drive adequate? Does the software have an adequate warranty? Is the software flexible, easily maintained, and user-friendly? Is online inquiry of files and records possible? Will the vendor keep the software up to date?
Vendor Evaluation	How long has the vendor been in business? Is the vendor financially stable and secure? How experienced is the vendor with the hardware and software? Does the vendor stand behind its products? How good is its warranty? Does the vendor regularly update its products? Does the vendor provide financing? Will the vendor put promises in a contract? Will the vendor supply a list of customer references? Does the vendor have a reputation for reliability and dependability? Does the vendor provide timely support and maintenance? Does the vendor provide implementation and installation support? Does the vendor have high-quality, responsive, and experienced personnel? Does the vendor provide training?

a weight to each evaluation criterion based on its importance. For each criterion, vendors are scored based on how well their proposals meet the requirement, and the weighted score totals are compared. In Table 23-2, vendor 3 offers the best system because its system scored 190 points more than vendor 2 did.

requirement costing - Comparing systems based on the cost of all required features; when software does not meet all requirements, the cost of developing unavailable features is estimated and added to its cost.

A **requirement costing** estimates the cost of purchasing or developing unavailable features. Total AIS costs, which is the cost of acquiring the system and the cost of developing the unavailable features, provides an equitable basis for comparing systems.

Because neither point scoring nor requirements costing is totally objective, the final choice among vendor proposals is not clear-cut. Point-scoring weights and scores are assigned subjectively, and dollar estimates of costs and benefits are not included. Requirement costing overlooks intangible factors such as reliability and vendor support.

Once the best AIS is identified, the software is thoroughly test-driven, other users are contacted to determine their satisfaction with the choice, vendor personnel are evaluated, and proposal details are confirmed to verify that the best AIS on paper is the best in practice. The lessons Geophysical Systems learned from its vendor selection highlight the importance of a thorough vendor evaluation (see Focus 23-1).

TABLE 23-2 Point-Scoring Evaluation of Vendor Proposals

		Vendor 1		Vendor 2		Vendor 3	
Criterion	Weight	Score	Weighted Score	Score	Weighted Score	Score	Weighted Score
Hardware compatibility	60	6	360	7	420	8	480
Hardware speed	30	6	180	10	300	5	150
Memory expansion	60	5	300	7	420	8	480
Hardware current	30	9	270	9	270	6	180
Software compatibility	90	7	630	7	630	9	810
Online inquiry capabilities	40	9	360	10	400	8	320
Controls	50	7	350	6	300	9	450
Positive references	40	10	400	8	320	6	240
Documentation	30	9	270	8	240	7	210
Easily maintained; updated regularly	50	7	350	8	400	9	450
Network capabilities	50	8	400	7	350	8	400
Vendor support	70	6	420	9	630	10	700
Totals			**4,290**		**4,680**		**4,870**

FOCUS 23-1 A Software Purchase That Went Awry

Geophysical Systems Corporation (GSC) developed a sonar device to analyze the production potential of oil and gas discoveries. GSC needed software to analyze the data generated by the sonar device and paid Seismograph Service $20 million to create it. When the Seismograph system could not accurately process the massive volume of data and perform the complex computations needed, GSC clients canceled their contracts. GSC went from yearly sales of $40 million and profits of $6 million to filing for bankruptcy two years later.

GSC sued, claiming Seismograph's system did not perform as promised and that Seismograph knew that before it began development. The jury awarded GSC $48 million for lost profits and the cost of the computer system. Seismograph appealed, claiming its system did work and that GSC's decline resulted from a slump in oil prices.

GSC's experience is common; many systems development projects do not produce the intended results.

Development by In-House Information Systems Departments

Organizations develop **custom software** when doing so provides a significant competitive advantage. There is little benefit to a custom-written payroll or accounts receivable system, whereas there may be significant benefits to sophisticated, just-in-time inventory management or product manufacturing software.

custom software - Software developed and written in-house to meet the unique needs of a particular company.

The hurdles that must be overcome to develop quality software are the significant amounts of time required, the complexity of the system, poor requirements, insufficient planning, inadequate communication and cooperation, lack of qualified staff, and poor top-management support.

Custom software is created in-house or by an outside company hired to write the software or assemble it from its inventory of program modules. When using an outside developer, a company maintains control over the development process as follows:

- Carefully select a developer that has experience in the company's industry and an in-depth understanding of how the company conducts its business.
- Sign a contract that rigorously defines the relationship between the company and the developer, places responsibility for meeting system requirements on the developer, and allows the project to be discontinued if key conditions are not met.
- Plan the project in detail and frequently monitor each step in the development.
- Communicate frequently and effectively.
- Control all costs and minimize cash outflows until the project is accepted.

There is no single right answer to the build-or-buy decision. Different companies come to different conclusions. After developing its own software, Gillette decided to purchase commercial software when possible to gain a greater competitive advantage from deciding *how* software should be used rather than determining *what* software should be used and then creating it. If commercial software does not meet all of Gillette's needs, it is modified using high-level development tools.

Pepsi moved in the opposite direction. It bought most of its mainframe software but, after moving to a client/server architecture, it could not find software sophisticated enough to meet its needs. Although Pepsi still buys software when it can find it, it has created most of its software.

Chapter 22 discusses in more depth the process used to develop custom software.

END-USER-DEVELOPED SOFTWARE

After the automobile was introduced, a famous sociologist predicted that the automobile market would not exceed 2 million cars because only that many people would be willing to serve as chauffeurs. It was once predicted that the telephone system would collapse because the geometric growth in calls would require everyone to be telephone operators. Instead, equipment was developed that automated operator functions.

After the introduction of computers, an expert claimed that the demand for information systems would grow so astronomically that almost everyone would have to become a programmer. Does this sound familiar? The solution is to help end users meet their own information needs. As with telephones, technology is being developed to automate much of the process for us. Just as most people have learned to drive automobiles, so will inexpensive PCs, a wide variety of powerful and inexpensive software, increased computer literacy, easier-to-use programming tools, and the Internet allow most organizations and people to meet their information needs.

End-user computing (EUC) is the hands-on development, use, and control of computer-based information systems by users. EUC is people using IT to meet their information needs rather than relying on systems professionals. For example, a savings and loan in California wanted a system to track loan reserve requirements. When the information systems (IS) department said the system would take 18 months to develop, the loan department used a PC and a database program to develop a functional program in one day. Enhancing the program took several more days. The loan department not only cut the development time from 18 months to a few days, but also got the exact information it needed because users developed the system themselves.

end-user computing (EUC) - The hands-on development, use, and control of computer-based information systems by users.

The following are examples of appropriate end-user development:

- Retrieving information from company databases to produce simple reports or to answer one-time queries.
- Performing "what-if," sensitivity, or statistical analyses.
- Developing applications using software such as a spreadsheet or a database system.
- Preparing schedules, such as depreciation schedules and loan amortizations.

End-user development is inappropriate for complex systems, such as those that process a large number of transactions or update database records. Therefore, it is not used for processing payroll, accounts receivables and payables, general ledger, or inventory.

As end users meet their information needs, they realize they can use computers to meet more and more information needs. Increased access to data also creates many new uses and information needs. The result is a tremendous ongoing growth in EUC.

The growth in EUC has altered the information system staff's role. They continue to develop and maintain transaction processing systems and companywide databases. In addition, they provide users with technical advice and operational support and make as much information available to end users as possible. Although this has created more work for the IS staff, it is counterbalanced by a decreased demand for traditional services. If the trend in EUC continues, it will represent 75% to 95% of all information processing by the end of the next decade.

ADVANTAGES AND DISADVANTAGES OF END-USER COMPUTING

EUC offers the following advantages:

- ***User creation, control, and implementation.*** Users, rather than the IS department, control the development process. Users decide whether a system should be developed and what information is important. This ownership helps users develop better systems.
- ***Systems that meet user needs.*** Systems developed by end users are more likely to meet user needs. Users discover flaws that IS people do not catch. Many of the user-analyst-programmer communication problems in traditional program development are avoided.
- ***Timeliness.*** Much of the lengthy delay inherent in traditional systems development is avoided, such as time-consuming cost–benefit analyses, detailed requirements definitions, and the delays and red tape of the approval process.
- ***Freeing up of systems resources.*** The more information needs users meet, the more time the IS department can spend on other development and maintenance activities. This reduces both the visible and the invisible backlog of systems development projects.
- ***Versatility and ease of use.*** Most EUC software is easy to understand and use. Users can change the information they produce or modify their application any time their requirements change. With a laptop computer, employees can complete work at home, on a plane—almost anywhere.

However, there are significant drawbacks to EUC and to eliminating the involvement of analysts or programmers in the development process.

- ***Logic and development errors.*** With little experience in systems development, end users are more likely to make errors and less likely to recognize when errors have occurred. They may solve the wrong problem, poorly define system requirements, apply an inappropriate analytical method, use the wrong software, use incomplete or outdated information, use faulty logic, or incorrectly use formulas or software commands. An oil and gas company developed a complex spreadsheet that showed that a proposed acquisition was profitable. When their CPA firm tested the model and agreed with it, a board of directors meeting was scheduled to propose the acquisition. Shortly before the meeting, a presenter tested the model so that he could understand how it worked and answer tough questions. He discovered formulas that distorted the projections, so he called in the creator and the CPA firm. The corrected formulas showed a significant loss on the acquisition. The board presentation was canceled, and the spreadsheet creator and CPA firm were fired.

- ***Inadequately tested applications.*** Users are less likely to test their applications rigorously, either because they do not recognize the need to do so or because of the difficulty or time involved. One Big Four CPA firm found that 90% of the spreadsheet models it tested had at least one calculation error.
- ***Inefficient systems.*** Most end users are not programmers nor are they trained in systems development. As a result, their systems are not always efficient. One bank clerk spent three weeks developing a program that examined each cell in a spreadsheet and changed its value to zero if it was a negative amount. When the 60-page program began returning a "too many nested ifs" error message, the clerk called in a consultant. Within five minutes, the consultant developed a finished application using a built-in spreadsheet function.
- ***Poorly controlled and documented systems.*** Many end users do not implement controls to protect their systems. User-created systems are often poorly documented because the user considers the task boring or unimportant. Users fail to realize that without documentation, others cannot understand how their system works.
- ***System incompatibilities.*** Companies that add end-user equipment without considering the technological implications have a diversity of hardware and software that is difficult to support or network. Aetna Life & Casualty spent more than $1 billion a year on IT to gain a competitive advantage. The result was 50,000 PCs from a few dozen manufacturers, 2,000 servers, 19 incompatible e-mail systems, and 36 different communications networks. Aetna finally realized it needed to shift its emphasis from owning the latest technology to the effective use of technology. Aetna standardized its systems and now uses only a few different PCs, Microsoft software, two e-mail systems, and one network. The result is compatibility across all systems and significantly less cost.
- ***Duplication of systems and data; wasted resources.*** End users are typically unaware that other users have similar information needs, resulting in duplicate systems. Inexperienced users may take on more development than they are able to accomplish. Both of these problems end up wasting time and resources.
- ***Increased costs.*** A single PC purchase is inexpensive; buying hundreds or thousands is costly. So is updating the hardware and software every few years. EUC has a high opportunity cost if it diverts users' attention from their primary jobs. It also increases time and data demands on corporate information systems.

It is possible to achieve the proper balance between the benefits and risks of end-user systems by training users, using systems analysts as advisers, and requiring user-created systems to be reviewed and documented prior to use.

MANAGING AND CONTROLLING END-USER COMPUTING

Organizations must manage and control EUC. Giving the IS department control discourages EUC and eliminates its benefits. However, if the organization maintains no controls over end users, such as what EUC tools are purchased or how they are used, it is likely to lead to significant problems. It is best to provide enough guidance and standards to control the system yet allow users the flexibility they need.

A **help desk** supports and controls end-user activities. The 60 help desk analysts and technicians at Schering-Plough handle 9,000 calls a month. Front-line analysts use expert system software to find scripted answers to user questions. Second-line technicians handle queries that are more complicated. Other companies use multimedia software with animation or videos to help staffers walk callers through a complicated process.

help desk - Analysts and technicians who answer employee questions with the purpose of encouraging, supporting, coordinating, and controlling end-user activity.

Help desk duties include resolving problems, disseminating information, evaluating new hardware and software products and training end users how to use them, assisting with application development, and providing technical maintenance and support. Help desks also develop and implement standards for hardware and software purchases, documentation, application testing, and security. Lastly, the help desk controls access to and sharing of corporate data among end users, while ensuring that the data are not duplicated and that access to confidential data remains restricted.

Outsourcing the System

outsourcing - Hiring an outside company to handle all or part of an organization's data processing activities.

Outsourcing is hiring an outside company to handle all or part of an organization's data processing activities. In mainframe outsourcing agreements, outsourcers buy client computers, hire the client's IS employees, operate and manage the system on the client's site, or migrate the system to the outsourcer's computers. Many outsourcing contracts are in effect for up to 10 years and cost millions of dollars a year. In a client/server or a PC outsourcing agreement, a service, function, or segment of business is outsourced. Most Fortune 500 companies outsource their PC support function. Royal Dutch Shell, the international oil company, has 80,000 PCs worldwide and outsources its installation, maintenance, training, help desk, and technical support.

Outsourcing was initially used for standardized applications such as payroll and accounting or by companies who wanted a cash infusion from selling their hardware. In 1989, Eastman Kodak surprised the business world by hiring IBM to run its data processing operations, DEC to run its telecommunications functions, and Businessland to run its PC operations. Kodak retained its IS strategic planning and development role, but outsourcers performed the implementation and operation responsibilities. The results were dramatic. Computer expenditures fell 90%. Operating expenses decreased 10% to 20%. Annual IS savings during the 10-year agreement were expected to be $130 million. Several years later, Xerox signed what was then the largest outsourcing deal in history: a $3.2 billion, 10-year contract with EDS to outsource its computing, telecommunications, and software management in 19 countries.

In one survey, 73% of companies outsourced some or all of their information systems, and most outsourced to several companies to increase flexibility, foster competition, and reduce costs. Most companies do not, however, outsource strategic IT management, business process management, or IT architecture.

Many smaller companies outsource. One company with annual revenues of $1 million outsources all accounting functions to a local CPA. Whenever they want, the owners can view all their transactions on the CPA's website and produce a myriad of reports. They also outsourced all IT processes, including website design and maintenance.

ADVANTAGES AND DISADVANTAGES OF OUTSOURCING

There are a number of significant advantages to outsourcing:

- ***A business solution.*** Outsourcing is a viable strategic and economic business solution that allows companies to concentrate on core competencies. Kodak focused on what it does best and left data processing to qualified computer companies. Kodak treats its outsourcers as partners and works closely with them to meet strategic and operational objectives.
- ***Asset utilization.*** Organizations improve their cash position and reduce expenses by selling assets to an outsourcer. Health Dimension outsourced data processing at its four hospitals so it could use its limited monetary resources to generate revenue.
- ***Access to greater expertise and better technology.*** Del Monte Foods turned to outsourcing because the cost and time involved in staying at the cutting edge of technology were rising significantly.
- ***Lower costs.*** IBM outsources programming to Chinese companies, whose labor costs are 30% of those in the United States. Outsourcers lower costs by standardizing user applications, buying hardware at bulk prices, splitting development and maintenance costs between projects, and operating at higher volumes. Continental Bank will save $100 million during its 10-year contract. However, Occidental Petroleum rejected outsourcing as costing more than internal AIS development and operation.
- ***Less development time.*** Experienced industry specialists develop and implement systems faster and more efficiently than in-house staff. Outsourcers also help cut through systems development politics.
- ***Elimination of peaks-and-valleys usage.*** Seasonal businesses require significant computer resources part of the year, and little the rest of the year. From January to March, W. Atlee Burpee's computers operated at 80% capacity processing seed and gardening

orders and at 20% the rest of the time. Outsourcing cut Burpee's processing costs in half by paying based on how much the system is used.

- ***Facilitation of downsizing.*** Companies that downsize often have an unnecessarily large AIS function. General Dynamics downsized dramatically because of reductions in defense industry spending. It signed a $3 billion, 10-year outsourcing contract even though its IS function was rated number one in the aerospace industry. It sold its data centers to CSC for $200 million and transferred 2,600 employees to CSC.

However, not all outsourcing experiences have been successful. Between 25% and 50% of outsourcing agreements fail or are major disappointments. In one survey, company executives labeled 17% of them disasters and almost 50% were brought back in-house. There have been a number of significant outsourcing failures, including the problems EDS has had with its U.S. Navy contract (see Focus 22-2). Another is JPMorgan Chase's cancellation of its $5 billion, seven-year deal with IBM.

Outsourcing failures are caused by failure to prepare properly, lukewarm company buy-in, blind imitation of competitors, thinking that outsourcing will solve deeper problems, shifting responsibility for a bad process to someone else, and entering into ill-defined agreements that do not meet expectations. Finally, many companies do not realize that systems development is a more complex management challenge when performed by outsiders.

Companies that outsource often experience some of the following drawbacks:

- ***Inflexibility.*** Many contracts are for 10 years. If a company is dissatisfied or has structural changes, the contract is difficult or costly to break. Before they merged, Integra Financial and Equimark had contracts with different outsourcers. Canceling one of them cost $4.5 million.
- ***Loss of control.*** A company runs the risk of losing control of its system and its data. For that reason, Ford's outsourcing agreement prevents CSC from working with other automobile manufacturers.
- ***Reduced competitive advantage.*** Companies may lose sight of how their AIS produces competitive advantages. Outsourcers are not as motivated as their clients to meet competitive challenges. Companies can mitigate this problem by outsourcing standard business processes (payroll, cash disbursements, etc.) and customizing those that provide competitive advantages.
- ***Locked-in system.*** It is expensive and difficult to reverse outsourcing. A company may have to buy new equipment and hire a new data processing staff, often at prohibitive costs. When Blue Cross of California decided to end its agreement, it knew virtually nothing about its system and could not afford to discharge EDS. In contrast, LSI Logic brought its system back in-house at significant dollar and personnel savings when it installed an enterprise resource planning (ERP) system.
- ***Unfulfilled goals.*** Critics claim some outsourcing benefits, such as increased efficiency, are a myth. USF&G canceled its $100 million contract with Cigna Information Services after 18 months when Cigna could not make the system work properly.
- ***Poor service.*** Common complaints are that responsiveness to changing business conditions is slow or nonexistent and migration to new technologies is poorly planned.
- ***Increased risk.*** Outsourcing business processes can expose a company to significant operational, financial, technology, strategy, personnel, legal, and regulatory risks.

Methods for Improving Systems Development

The systems analysis and design process has evolved considerably since computer programming began in the 1950s with programs developed in machine language or assembly language. Third generation languages were introduced in the 1960s, database management systems in the 1970s, and fourth generation languages in the 1980s. The 1990s brought visual development technologies and integrated enterprise resource planning systems (ERP). By the 2000s, the development of Internet-based web systems and the use of mobile devices, cloud computing, and resource sharing systems were in full swing. The coming years will undoubtedly produce more improvements and changes.

Information system development is a complex and difficult process, fraught with many failures. As time has evolved, the software industry has developed a number of techniques to simplify, improve, and speed up the development process. This section of the chapter discusses some of the more important ones, including business process management, prototyping, agile technologies, and computer-aided software engineering.

BUSINESS PROCESS MANAGEMENT

business process reengineering (BPR) - The thorough analysis and redesign of business processes and information systems to achieve dramatic performance improvements; often a drastic, one-time event.

business process management (BPM) - A systematic approach to continuously improving and optimizing business processes; a more gradual improvement facilitated by technology.

As organizations seek to improve their information systems and comply with legal and regulatory reforms, they are paying greater attention to their business processes. **Business process reengineering (BPR)** is a drastic, one-time-event approach to improving and automating business processes. However, it has had a low success rate. With further improvements, BPR has evolved into **business process management (BPM)**, a systematic approach to continuously improving and optimizing an organization's business processes. BPM is a more gradual and ongoing business process improvement supported and enabled by technology. As a result, BPM is a good way to introduce both a human and a technological change capability into an organization.

Some of the important principles underlying BPM are the following:

- ***Business processes can produce competitive advantages.*** Innovative processes that help business respond to changing consumer, market, and regulatory demands faster than competitors create competitive advantages. Good business process design is vital to an organization's success. For example, if a competitive bidding process is time sensitive and requires coordination between multiple functions, a poorly designed bid process can handicap the process so much that effective and profitable bids are not prepared.
- ***Business processes must be managed end to end.*** BPM views business processes as strategic organizational assets that should be understood, managed, and improved. Even if each part of a multifunctional business process functions well independently, the entire process may be suboptimal if there is inadequate communication and coordination among functional units (sales, production, etc.). Managing business processes from inception to completion can control such problems. A process owner is designated, performance standards are set, and control and monitoring processes are established.
- ***Business processes should be agile.*** Organizations must continuously improve and adapt their business processes to compete. This requires flexibility and business process automation technology that supports rapid modifications.
- ***Business processes must be aligned with organizational strategy and needs.*** To be effective and efficient, a company must align its business processes with its business strategy.

business process management systems (BPMS) - Systems that automate and facilitate business process improvements throughout the SDLC.

Business process management systems (BPMS) automate and facilitate business process improvements. A BPMS can improve communication and collaboration, automate activities, and integrate with other systems and with other partners in the value chain. Some people claim that BPMS is the bridge between IT and business. Many companies worldwide are successfully implementing BPMS-based processes.

Like enterprise resource planning (ERP) systems, BPMS are enterprise-wide systems that support corporate activities. However, ERP systems are data-centered and BPMS are process-centered. Most manufactures of ERP systems are now integrating BPM into their systems.

A BPMS has the following four major components:

- A process engine to model and execute applications, including business rules.
- Business analytics to help identify and react to business issues, trends, and opportunities.
- Collaboration tools to remove communication barriers.
- A content manager to store and secure electronic documents, images, and other files.

INTERNAL CONTROLS IN A BUSINESS PROCESS MANAGEMENT SYSTEM A BPMS can improve internal controls. In event-based (as opposed to process-based) systems, users are granted access only to certain activity types. When authorization is granted using other parameters, such as dollar amounts, system developers have to include complex and expensive

authorization restrictions. A BPMS uses the organization's business process rules to determine the correct person to perform a task and authorizes that person to perform it.

Segregation of duties can also be improved in a BPMS. In many event-based systems, the procedures for obtaining management approvals lengthen the business process and add additional costs. BPMS reduces the delays and costs by instantaneously transferring items needing approval to the manager. Within a few minutes, the manager can inspect and authorize the electronic form and transfer it to the next step in the process. BPMS have several other innovative authorization mechanisms, such as delegating authority to co-managers and creating a pool of authorizing managers to reduce bottlenecks when managers are overburdened or unavailable.

Application controls are also strengthened by a BPMS. In event-based systems, users identify what actions must be done, such as billing a customer when goods are shipped. If the action is not taken, an error occurs, such as doing something twice, not doing it at all, or doing the wrong thing. A BPMS uses a proactive process management approach that eliminates such problems. Users do not have to decide whether to take action and then decide which action is correct. The BPMS, using the company's business rules, decides what action must take place and forwards the task to the appropriate person's task list, where it remains until it is executed. The person gets an e-mail informing them that the task awaits their attention. This process prevents errors because it prevents procedures from being circumvented, prevents users from performing a different action, prevents items from being removed from the task list before they are accomplished, and sends additional reminder messages until the task is performed.

Another control advantage of BPMS is its built-in audit trail. The process monitoring and tracking systems, which document and link all actions and process steps in the order they occur in a process log, make it easy to track everything that takes place. This allows the auditor to continuously audit the business processes while they are active and afterward.

PROTOTYPING

Prototyping is a systems design approach in which a simplified working model of a system is developed. Developers who use prototyping still go through the SDLC discussed in Chapter 22, but prototyping allows them to condense and speed up some analysis and design tasks. Prototyping helps capture user needs and helps developers and users make conceptual and physical design decisions.

prototyping - An approach to systems design in which a simplified working model, or prototype, of an IS is developed.

UNUM Life Insurance wanted to use image processing to link systems and users. When top management had a hard time getting middle managers to understand how the system would work and the issues involved in the change, they had a prototype prepared. After using it, the managers grasped the possibilities and issues. Up to that point, they thought image processing meant replacing file cabinets.

As shown in Figure 23-1, a prototype is developed using four steps. The first is to meet with users to agree on the size and scope of the system and to decide what the system should and should not include. Developers and users also determine decision-making and transaction processing outputs as well as the inputs and data needed to produce the outputs. The emphasis is on *what* output should be produced rather than *how* it should be produced. The developer must ensure that users' expectations are realistic and that their basic information requirements can be met. The designer uses the information requirements to develop cost, time, and feasibility estimates for alternative AIS solutions.

The second step is to develop an initial prototype. The emphasis is on low cost and rapid development. Nonessential functions, controls, exception handling, input validation, and processing speed are ignored in the interests of simplicity, flexibility, and ease of use. Users need to see and use tentative data entry screens, menus, and source documents; respond to prompts; query the system; judge response times; and issue commands. The developer demonstrates the finished prototype and asks users to provide feedback on what they like and dislike, which is much easier to do than imagining what they want in a system. Even a simple system that is not fully functional demonstrates features better than diagrams, drawings, or verbal explanations.

In the third step, developers use the feedback to modify the system and return it to the users. Trial usage and modification continues until users are satisfied that the system meets their needs. A typical prototype goes through four to six iterations.

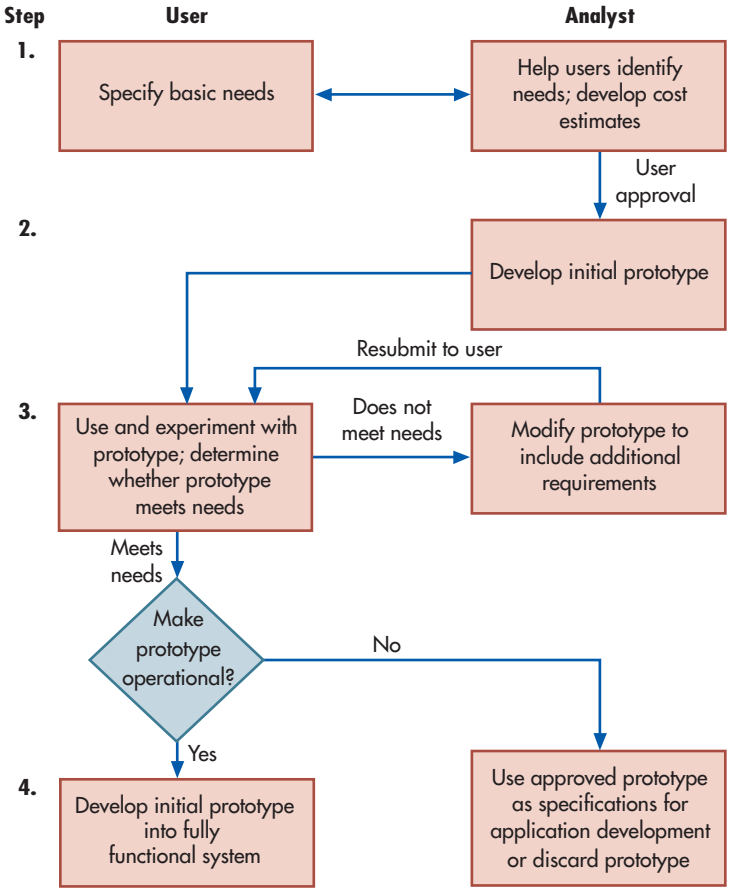

FIGURE 23-1
The Steps for Developing a System from a Prototype

The fourth step is to use the system. An approved prototype is typically used in one of two ways. Half of all prototypes are turned into fully functional systems, referred to as **operational prototypes**. To make the prototype operational, the developer incorporates the things ignored in step one, provides backup and recovery, and integrates the prototype with other systems. **Nonoperational (throwaway) prototypes** are used several ways. System requirements identified during prototyping can be used to develop a new system. The prototype can be used as the initial prototype for an expanded system designed to meet the needs of many different users. When an unsalvageable prototype is discarded, the company potentially saves itself years of development work and lots of money by avoiding the traditional SDLC process.

operational prototypes - Prototypes that are further developed into fully functional systems.

nonoperational (throwaway) prototypes - Prototypes that are discarded, but the system requirements identified from the prototypes are used to develop a new system.

WHEN TO USE PROTOTYPING Prototyping is appropriate when there is a high level of uncertainty, it is unclear what questions to ask, the AIS cannot be clearly visualized, or there is a high likelihood of failure. Good candidates for prototyping include decision support systems, executive information systems, expert systems, and information retrieval systems. Prototyping is less appropriate for large or complex systems that serve major organizational components or cross-organizational boundaries or for developing standard AIS components, such as accounts receivable or inventory management. Table 23-3 shows the conditions that make prototyping an appropriate design methodology.

ADVANTAGES OF PROTOTYPING Prototyping has the following advantages:

- ***Better definition of user needs.*** Prototyping generally requires intensive involvement from end users, resulting in well-defined user needs.
- ***Higher user involvement and satisfaction.*** Because users' requirements are met, there is less risk that the AIS will not be used. Early user involvement helps to build a climate of acceptance rather than skepticism and criticism.

TABLE 23-3 Conditions That Favor the Use of Prototyping

Users' needs are not understood, change rapidly, or evolve as the system is used.
System requirements are hard to define.
System inputs and outputs are not known.
The task to be performed is not well structured.
Designers are uncertain about what technology to use.
The system is crucial and needed quickly.
The risk associated with developing the wrong system is high.
User reactions are especially important development considerations.
Many design strategies must be tested.
The design staff has little experience developing the system or application.
The system will be used infrequently (processing efficiency is not a major concern).

- ***Faster development time.*** Prototypes are often functioning after a few days or weeks, allowing users to immediately evaluate the system. John Hancock Mutual Life Insurance developed an executive information system prototype in one month, as described in Focus 23-2.
- ***Fewer errors.*** The users test each version of the prototype, so errors are detected and eliminated early. It is also easier to identify and terminate infeasible systems before a great deal of time and expense is incurred.
- ***More opportunity for changes.*** Users can suggest changes until the system is exactly what they want.
- ***Less costly.*** Prototype systems can be developed for 20% of the cost of traditional systems. One utility company claimed a 13-to-1 improvement in development time over traditional methods when prototyping was used to develop 10 major applications.

DISADVANTAGES OF PROTOTYPING Prototyping has the following disadvantages:

- ***Significant user time.*** Users must devote significant time to working with the prototype and providing feedback. It may require more involvement and commitment than users are willing to give.
- ***Less efficient use of system resources.*** Prototype development does not always achieve resource efficiency, sometimes resulting in poor performance and reliability as well as high maintenance and support costs.
- ***Inadequate testing and documentation.*** Developers may shortchange testing and documentation because users are testing the prototype during development.
- ***Negative behavioral reactions.*** These can occur when requests for improvements are not made, there are too many iterations, or a prototype that users are invested in is thrown away.
- ***Never-ending development.*** This occurs when prototyping is not managed properly and the prototype is never completed due to recurring iterations and revision requests.

AGILE METHODOLOGIES

One of the problems with the traditional SDLC approach, often called the waterfall approach, is that it assumes requirements do not change as a system is designed. Figure 22-2 shows the waterfall nature of this development and the reasons why a developer would return to a prior SDLC phase. However, the reality is that in many systems development projects, requirements do change because users often do not understand all the things they want the software to do. That is, they discover more and different requirements as the software development process proceeds. Furthermore, as technology changes so fast, more things can be accomplished with technology. These rapid changes in requirements and technology created the need for shorter product life cycles that were often not compatible with traditional software development methodologies.

What many system developers wanted was a more iterative development methodology—one that embraced changes in requirements rather than restricted them and that was better

agile development - A guiding philosophy and a set of principles for developing information systems in an unknown, rapidly changing environment.

scrum methodology - A software development methodology where a team works together in an intense but relatively short iterative and incremental scrum process to reach a common development goal, with team members meeting daily in face-to-face communication, until development is concluded.

scrum development - A process that embraces customers frequently changing their minds about what they need or want. Scrum development focuses on flexibility, responding to new requirements, adapting to evolving changes in technology, and quickly delivering a system the customer can evaluate.

product owner - The customer who is responsible for making sure the scrum team produces what is needed. They write the user stories and prioritize backlog items so the scrum team knows what to develop next.

user stories - A description of something a user wants to include in the system written by the product owner.

product backlog - Items waiting to be developed that are prioritized by the product owner.

scrum team - A small group of up to 9 cross-functional developers responsible for developing, testing, and delivering software at the end of a scrum sprint. The team determines a sprint's major goals and deliverables.

sprint - A pre-determined time period where the team works on high priority items in the product backlog. A sprint's scope is frozen and desired changes are added to the product backlog. Sprints begin with a planning event to determine goals and deliverables and ends with a review to see if they were achieved. The incremental software developed is presented to the customer.

scrum master - Scrum facilitator who makes sure scrum practices are followed, promotes self-organization within the team, holds daily team meetings, works with the product owner to make sure the product backlog is properly maintained, and removes any impediments that affect the team's ability to achieve its goals and produce the sprint's deliverables.

able to deal with a lack of predictability in software development. They wanted a development process designed to produce frequent versions of a working system, with each succeeding iteration including more subsets of what users wanted in their system. In other words, they wanted to embrace smaller, incremental changes rather than a one-time, massive change in the software. These new processes require users and developers to work closely together with frequent feedback between the two groups to produce the desired system.

In 2001, proponents of a more iterative development process met and came up with **agile development**, a guiding philosophy and a set of principles for developing information systems in an unknown, rapidly changing environment. According to the agile development philosophy, it is more important to:

1. Respond to change than to follow a plan or predefined schedule.
2. Emphasize individuals and interactions than processes, tools, and development controls.
3. Emphasize customer collaboration than contract negotiations.
4. Produce quality software than to produce good software documentation.

There are a number of development methodologies that make use of agile development principles. Three of the most important are discussed here: Scrum, Extreme Programming, and the Unified Process.

SCRUM Scrum's name is derived from the game of rugby. After a penalty, players from both teams line up across from each other in a tightly-packed formation and the ball is thrown into the gap between the teams. What follows is an intense and relatively short struggle between the two teams in order to gain possession of the ball. Like the rugby scrum, the **scrum methodology** involves a software development team that works together in an intense but relatively short process to reach a common development goal. The iterative and incremental scrum process continues to repeat itself, with team members meeting daily in face-to-face communication, until development is concluded.

Those involved in **scrum development** embrace the fact that the intended users, called the client or customer, will frequently change their minds about what they need or want. They accept the unpredictability associated with the customer not fully understanding or being able to define all their system needs. Instead, they focus on flexibility, responding to new requirements, adapting to evolving changes in technology and market conditions, and quickly delivering a system that the customer can evaluate.

In the scrum process there are three main roles: product owner, team members, and scrum master. The **product owner** is the customer who is responsible for making sure the scrum team produces what is needed. The product owner works with system users and writes **user stories** (a description of something a user wants to include in the system), places them in the **product backlog** (items waiting to be developed), and prioritizes the backlog items so the team knows what to develop next. The product owner spends much of his or her time focusing on the business side of the development project, working with users to identify their business needs, and serving as the communication bridge between the users and the scrum team.

The **scrum team** is a small group of up to 9 cross-functional developers responsible for developing, testing, and delivering software at the end of a scrum sprint. A **sprint** is a pre-determined time period, usually between one and four weeks, that the team works on one or more high priority items in the product backlog. A sprint's scope is frozen and anyone desiring changes must ask the product owner to add them to the product backlog. Each sprint begins with a planning event where the team determines the sprint's goals and deliverables and ends with a sprint review to see if they were achieved and to determine how to improve the next sprint. The incremental software developed during the sprint is presented to the customer.

The **scrum master** is the scrum facilitator that ensures scrum practices are followed and helps the team self-organize. She acts as a buffer between the team and any distracting influences by removing impediments that affect the team's ability to achieve its goals and produce the sprint's deliverables. The scrum master works with the product owner to make sure the product backlog is properly maintained. The scrum master holds a brief daily scrum meeting

with all team members where they report the progress made the day before, what they will work on that day, and what help they need from the scrum master to remove an impediment.

EXTREME PROGRAMMING **Extreme programming** (**XP**) is a software development methodology designed to produce higher-quality software more productively by taking the beneficial elements and best practices of traditional software development to "extreme" levels. Some of the more important characteristics of XP include.

extreme programming (XP) - A software development methodology designed to produce higher-quality software more productively by taking the beneficial elements and best practices of traditional software development to "extreme" levels.

- Uses two-person programming teams.
- Recognizes that changes to systems requirements are a normal and accepted occurrence in software development that should be expected and planned for as time passes and system needs are better understood. XP embraces changes and abandons the idea that a stable set of requirements can be produced during the initial planning phase of development.
- Uses short development cycles, culminating in frequent software releases, so that there are many checkpoints that allow new customer requirements to be introduced. Makes many small, incremental changes, rather than a few big changes, allowing the customer more control over the software development process. It reduces the cost of changes and improves development productivity.
- Starts with the simplest solution and only adds extra functionality as needed. Trying to design and code uncertain or complex future requirements might delay crucial features and risks spending resources on features that might not be needed or desired.
- Uses code to communicate thoughts about complex or hard to understand programming problems. Often, clear and concise code can better explain a problem than a written or oral description of the problem. Other programmers can give feedback on this code by coding their thoughts about the problem.
- Requires programmers to listen to customer needs and understand their business processes so they can provide customer feedback about the technical aspects of how a problem can or cannot be solved.
- Tests extensively every piece of code written before developing additional features to eliminate as many coding flaws as possible. Programmers create as many automated tests as possible to try to break the code. **Unit tests** help determine whether a given feature works as intended. **Acceptance tests** are used to verify that code satisfies the customer's actual requirements. System-wide **integration tests** are used to check for incompatible interfaces between code segments.
- Requires frequent and prompt communication and feedback between customers, developers, and testers. Customers communicate their needs and help develop the acceptance tests that occur every few weeks so they can easily correct design flaws and steer development. Developers estimate the costs and the time required to implement new requirements and communicate them to customers. Developers need feedback from testers to correct errors or weaknesses found during testing.
- Organizes system logic to prevent system dependencies so that changes in one part of the system will not affect other parts of the system.

unit tests - Help determine whether a given feature works as intended.

acceptance tests - Used to verify that code satisfies the customer's actual requirements.

integration tests - Used to check for incompatible interfaces between code segments.

UNIFIED PROCESS The **Unified Process** is a software development framework with four phases: inception, elaboration, construction, and transition. The last three phases are divided into a series of iterations of a predetermined length. Each incremental iteration contains additional functions or an improved version of the previously developed software. There are several versions, including Agile Unified Process, OpenUP, and the most popular—the Rational Unified Process.

Unified Process - A development framework with four phases: inception, elaboration, construction, and transition. The last three phases are divided into a series of iterations of a predetermined length. Each incremental iteration contains additional functions or an improved version of the previously developed software.

Inception is the shortest phase; if it takes too long it usually indicates that there are too many or too detailed up-front specifications. In the inception phase, analysts define the project's scope, identify preliminary key requirements and risks, determine the project's feasibility, and make the business case for developing the project.

In the elaboration phase, which is the second longest and considered by some to be the most important, analysts do most of the analysis and design activities for the project. They develop detailed user requirements and determine how to address known risk factors, with the

executable architecture baseline - A partial implementation of the system that includes all significant architecture components and demonstrates that the architecture supports key system functionality and will produce the desired performance and scalability at an acceptable cost.

most important risks addressed first. They determine the system's architecture and validate it using an **executable architecture baseline**, which is a partial implementation of the system that includes all significant architecture components and demonstrates that the architecture supports key system functionality and will produce the desired performance and scalability at an acceptable cost. The phase concludes with a plan for the Construction phase.

Construction, the longest phase, is where the system is coded and built, using the foundation created in the elaboration phase. The system is built and implemented in a series of short iterations, each of which results in an executable software release. Each new iteration is based on a use case. The construction phase ends with a beta version of the software that will be deployed during the transition phase.

In the transition phase, the system is made available to system users. System conversions and user training also take place during transition. The transition phase often includes several iterations, and user feedback during the initial transition phases is used to refine the system in the later transition phases. The transition phase is complete when the system meets user expectations and acceptance test criteria are satisfied.

COMPUTER-AIDED SOFTWARE ENGINEERING

computer-aided software (or systems) engineering (CASE) - Integrated package of tools that skilled designers use to help plan, analyze, design, program, and maintain an IS.

Computer-aided software (or systems) engineering (CASE) is an integrated package of tools that skilled designers use to help plan, analyze, design, program, and maintain an information system. CASE software typically has tools for strategic planning, project and system management, database design, screen and report layout, and automatic code generation. Many companies use CASE tools. Florida Power's $86 million customer information system was created using Accenture's CASE tool.

CASE tools provide a number of important advantages:

- ***Improved productivity.*** CASE can generate bug-free code from system specifications and can automate repetitive tasks. A programmer at Baptist Medical System used CASE to develop a system in one week that was estimated to take four months. Sony reported that CASE increased their productivity by 600%.
- ***Improved program quality.*** CASE tools simplify the enforcement of structured development standards, check the internal accuracy of the design, and detect inconsistencies.
- ***Cost savings.*** Savings of 80% to 90% are reported. At DuPont, an application estimated to require 27 months at a cost of $270,000 was finished in 4 months for $30,000. More than 90% of the code was generated directly from design specifications.
- ***Improved control procedures.*** CASE tools encourage system controls, security measures, and system auditability and error-handling procedures early in the design process.
- ***Simplified documentation.*** CASE automatically documents the system as the development progresses.

FOCUS 23-2 Prototyping at John Hancock

John Hancock Mutual Life Insurance was dissatisfied with the traditional development process. Too often, after development the typical user reaction was, "I may have said this is what I wanted, but it isn't." To counter this problem, Hancock used prototyping to develop an executive information system (EIS) that would obtain data quickly and easily from the existing system.

The development team included IBM consultants, users, systems analysts, and programmers. The prototyping process was highly interactive, and continual user involvement eliminated many misunderstandings. Programming started immediately, preparing sample screens for the first user interviews. Developers showed the users how the system would work and gave them a chance to try the screens. Almost immediately, users could determine whether what they said they wanted was what they needed.

The EIS prototype, which took a month to build, allowed top management to query current and historical financial data and measurements. Top managers who were skeptical when the project began were impressed by how much the team was able to accomplish in a single month.

Some of the more serious problems with CASE technology include the following:

- ***Incompatibility.*** Some CASE tools do not interact effectively with other systems.
- ***Cost.*** CASE technology is expensive, putting it out of the reach of many small companies.
- ***Unmet expectations.*** A Deloitte & Touche survey indicated that only 37% of CIOs using CASE believe they achieved the expected benefits.

Summary and Case Conclusion

A company can use different strategies to obtain a new AIS. First, as the quality and quantity of vendor-written software increases, more companies are purchasing it. Second, IS departments develop the software or allow end users to develop it. Third, some companies buy software and modify it themselves or ask the vendor to modify it so it meets company needs. Fourth, companies outsource data processing activities.

There are many ways to speed up or improve the development process. One way is business process management, which is a systematic approach to continuously improving and optimizing an organization's business processes.

A second way is to design a prototype, a simplified working model of a system. A prototype is quickly and inexpensively built and is given to users to "test-drive" so they can decide what they like and dislike about the system. Their reactions and feedback are used to modify the system, which is again given to the users to test. This iterative process of trial usage and modification continues until the users are satisfied that the system adequately meets their needs.

A third way to improve the development process is to use CASE tools to plan, analyze, design, program, and maintain an information system. They are also used to enhance the efforts of managers, users, and programmers in understanding information needs.

Ann considered the different strategies and eliminated several of them. She decided against outsourcing because she believes her team can do a better and faster job developing the system than an outsourcer could. Ann does not think prototyping would be effective because Shoppers Mart needs a large and complex system that would serve the needs of many users in many functional areas. Ann narrowed her options down to purchasing a system or designing one in-house. If Shoppers Mart develops its own software, Ann will investigate the various CASE and BPMS packages on the market to see whether they will add value to the development process.

No matter which approach she chooses, Ann wants to facilitate as much end-user development as is practical and useful. Ann will make the final decision during the conceptual design phase (Chapter 24). To gather the information she needs to decide whether to purchase software, Ann prepares and sends an RFP to vendors asking them to propose software and hardware to meet the company's needs identified during systems analysis.

KEY TERMS

commercial software 756
turnkey system 756
Software-as-a-Service (SaaS) providers 756
request for proposal (RFP) 756
benchmark problem 757
point scoring 757
requirement costing 758
custom software 759
end-user computing (EUC) 759
help desk 761
outsourcing 762
business process reengineering (BPR) 764
business process management (BPM) 764
business process management systems (BPMS) 764
prototyping 765
operational prototypes 766
nonoperational (throwaway) prototypes 766
agile development 768
scrum methodology 768
scrum development 768
product owner 768
user stories 768
product backlog 768
scrum team 768
sprint 768
scrum master 768
extreme programming (XP) 769
unit tests 769
acceptance tests 769
integration tests 769
Unified Process 769
executable architecture baseline 770
computer-aided software (or systems) engineering (CASE) 770

AIS in Action

CHAPTER QUIZ

1. Which of the following is not one of the difficulties software developers have experienced using the traditional systems development life cycle?
a. AIS development projects are backlogged for years.
b. Changes are usually not possible after requirements have been frozen.
c. The AIS developed may not meet their needs.
d. All are difficulties with the traditional SDLC.

2. Companies that buy rather than develop an AIS must still go through the systems development life cycle.
a. true
b. false

3. Which of the following statements is false?
a. As a general rule, companies should buy rather than develop software if they can find commercial software that meets their needs.
b. As an AIS increases in size and complexity, there is a greater likelihood that commercial software can be found that meets user needs.
c. A company should not attempt to develop its own custom software unless experienced, in-house programming personnel are available and the job can be completed less expensively on the inside.
d. As a general rule, a company should develop custom software only when it will provide a significant competitive advantage.

4. When a company is buying large and complex systems, vendors are invited to submit systems for consideration. What is such a solicitation called?
a. request for quotation
b. request for system
c. request for proposal
d. good-faith estimate

5. To compare system performance, a company can create a data processing task with input, processing, and output jobs. This task is performed on the systems under consideration, and the processing times are compared. The AIS with the lowest time is the most efficient. What is this process called?
a. benchmarking
b. requirements costing
c. point scoring
d. performance testing

6. Which of the following statements is true?
a. Because the AIS is so crucial, companies never outsource parts of the AIS.
b. Most mainframe outsourcing contracts are for two to three years and cost thousands of dollars a year.
c. Outsourcers often buy the client's computers and hire all or most of its information systems employees.
d. Only companies struggling to survive and wanting a quick infusion of cash from selling their hardware use outsourcing.

7. Which of the following is not a benefit of outsourcing?
a. It offers a great deal of flexibility because it is relatively easy to change outsourcers.
b. It can provide access to the expertise and special services provided by outsourcers.
c. It allows companies to move to a more sophisticated level of computing at a reasonable cost.
d. It is a cost-effective way to handle the peaks and valleys found in seasonal businesses.

8. Which of the following is a true statement about prototyping?
a. In the early stages of prototyping, system controls and exception handling may be sacrificed in the interests of simplicity, flexibility, and ease of use.
b. A prototype is a scaled-down, first-draft model that is quickly and inexpensively built and given to users to evaluate.

c. The first step in prototyping is to identify system requirements.
d. All of the statements are true.

9. Which of the following is not an advantage of prototyping?
a. better definition of user needs
b. adequately tested and documented systems
c. higher user involvement and satisfaction
d. faster development time

10. When is it most appropriate to use prototyping?
a. when there is little uncertainty about the AIS
b. when it is clear what users' needs are
c. when the final AIS cannot be clearly visualized because the decision process is still unclear
d. when there is a very low likelihood of failure

COMPREHENSIVE PROBLEM

In 1991, telemarketers placed 18 million calls per day; by 2003, it was 104 million. President Bush announced the Do Not Call Registry by saying, "Unwanted telemarketing calls are intrusive, they are annoying, and they're all too common. When Americans are sitting down to dinner, or a parent is reading to his or her child, the last thing they need is a call from a stranger with a sales pitch." Congress appropriated $18.1 million to fund the program, which made it a federal offense for telemarketers to call anyone on the list. Within 72 hours, more than 10 million phone numbers were added to the Do Not Call list when people accessed the donotcall.gov website or called the toll-free number. The Do Not Call Registry was hailed as one the most successful IT projects in the history of government.

Identify the benefits and risks of the FTC purchasing a prewritten software system, developing the system in-house, and outsourcing the system to an external vendor. What approach do you think the FTC should have used?

(*Source:* Adapted from Alice Dragoon, "How the FTC Rescued the Dinner Hour," *CIO* [June 1, 2004]: 59–64.)

DISCUSSION QUESTIONS

23.1 What is the accountant's role in the computer acquisition process? Should the accountant play an active role, or should all the work be left to computer experts? In what aspects of computer acquisition might an accountant provide a useful contribution?

23.2 In a Midwest city of 45,000, a computer was purchased, and in-house programmers began developing programs. Four years later, only one incomplete and poorly functioning application had been developed, none of the software met users' minimum requirements, and the hardware and the software frequently failed. Why do you think the city was unable to produce quality, workable software? Would the city have been better off purchasing commercial software? Could the city have found commercial software that met its needs? Why, or why not?

23.3 You are a systems consultant for Ernst, Price, and Deloitte, CPAs. At your country club's annual golf tournament, Frank Fender, an automobile dealer, describes a proposal from Turnkey Systems and asks for your opinion. The commercial software system will handle inventories, receivables, payroll, accounts payable, and general ledger accounting. Turnkey personnel would install the $40,000 system and train Fender's employees. Identify the major themes you would touch on in responding to Fender. Identify the advantages and disadvantages of using a turnkey system with commercial software for the organization's accounting system.

23.4 Sara Jones owns a rapidly growing retail store that faces stiff competition due to poor customer service, late and error-prone billing, and inefficient inventory control. For the company's growth to continue, its AIS must be upgraded, but Sara is not sure what the company wants the AIS to accomplish. Sara has heard about prototyping but does not know what it is or whether it would help. How would you explain prototyping to Sara? Include an explanation of its advantages and disadvantages as well as when its use is appropriate.

23.5 Clint Grace has been in business more than 30 years and has definite ideas about how his 10 retail stores should be run. He is financially conservative and is reluctant to make expenditures that do not have a clear financial payoff. Store profitability has declined sharply, and customer dissatisfaction is high. Store managers never know how much inventory is on hand and when purchases are needed until a shelf is empty. Clint asks you to determine why profitability has declined and to recommend a solution. You determine that the current AIS is inefficient and unreliable and that company processes and procedures are out-of-date. You believe the solution is to redesign the systems and business processes using BPM. What are some challenges you might face in redesigning the system? How will you present your recommendations to Clint?

PROBLEMS

23.1 Match the terms with their definitions:

____ **1.**	scrum master	a.	Software and hardware sold as a package; the user can begin using the system after the vendor installs it
____ **2.**	turnkey system	b.	Company that rents cloud-based software applications that customers can access via the Internet
____ **3.**	executable architecture baseline	c.	Document that asks vendors to bid on a system to meet specified needs
____ **4.**	scrum development	d.	Comparing systems by executing input, processing, and output tasks on different computer systems and evaluating the results
____ **5.**	nonoperational prototypes	e.	Evaluating the overall merits of vendor proposals by assigning a weight to each evaluation criterion based on its importance
____ **6.**	integration tests	f.	Hands-on development, use, and control of computer-based information systems by users
____ **7.**	unified process	g.	Hiring an outside company to handle an organization's data processing activities
____ **8.**	point scoring	h.	Systematic approach to gradually and continuously improving business processes
____ **9.**	prototyping	i.	Systems design approach where a simplified working model of an AIS is developed
____**10.**	user stories	j.	Prototype is discarded, but the system requirements identified are used to develop a new system
____**11.**	Software-as-a-Service	k.	Philosophy and principles for IS development in an unknown, rapidly changing environment
____**12.**	unit tests	l.	Development focus is flexibility, responding to new requirements, adapting to changes, and quickly delivering a system to evaluate
____**13.**	business process reengineering	m.	Customer is responsible for making sure the scrum team produces what is needed

____**14.**	product owner	n.	Description of what a product owner wants to include in the system
____**15.**	business process management	o.	Pre-determined time period where the team works on high priority items in the product backlog
____**16.**	sprint	p.	Person who makes sure scrum practices are followed, holds daily team meetings, and works with product owners
____**17.**	scrum methodology	q.	Produces higher quality software by taking best software development practices to very high levels
____**18.**	outsourcing	r.	Helps determine whether a given feature works as intended
____**19.**	benchmark problem	s.	Checks for incompatible interfaces between code segments
____**20.**	extreme programming	t.	Development framework with four phases: inception, elaboration, construction, and transition
____**21.**	agile development	u.	Partial system implementation that includes all significant architecture components
____**22.**	request for proposal	v.	Integrated package of tools that skilled designers use to plan, analyze, design, program, and maintain an IS
		w.	Thorough analysis and redesign of business processes and systems to achieve dramatic improvements; often a drastic, one-time-event
		x.	Prototypes that are further developed into fully functional systems
		y.	Development methodology where teams work in an intense, short, iterative, incremental process to reach a common development goal
		z.	Items waiting to be developed; prioritized by the product owner

23.2 This chapter describes several different agile methodologies for system development. Select one of the methodologies and conduct a search (using written materials, the Internet, electronic databases, etc.) for one or more companies that successfully used that methodology to develop an information system. Per your professor's instructions, prepare an oral or written summary of the successful implementation. Include in your summary the nature of the system, the approach used to develop the system, and a description of what it does and how it has helped the company that developed it.

23.3 Search written materials, the Internet, and electronic databases for successful and failed information system implementations. Prepare an oral or written summary of a successful and a failed implementation. Include the approach used to acquire or develop the system.

23.4 Mark Mitton, the liaison to the IS department, has eliminated all but the best three systems. Mark developed a list of required features, carefully reviewed each system, talked to other users, and interviewed appropriate systems representatives. Mark used a point-scoring system to assign weights to each requirement. Mark developed Table 23-4 to help him select the best system.

REQUIRED

a. Use a spreadsheet to develop a point-scoring matrix and determine which system Mark should select.

TABLE 23-4 An Evaluation Matrix

Selection Criteria	Weight	System 1	System 2	System 3
Software				
Fulfillment of business needs	100	6	8	9
Acceptance in marketplace	30	6	7	6
Quality of documentation	50	7	9	8
Quality of warranty	50	4	8	7
Ease of use	80	7	6	5
Control features	50	9	7	9
Flexibility	20	4	5	9
Security features	30	4	4	8
Modularity	30	8	5	4
Integration with other software	30	8	9	6
Quality of support utilities	50	9	8	5
Vendor				
Reputation and reliability	10	3	9	6
Experience with similar systems	20	5	5	6
Installation assistance	70	9	4	6
Training assistance	35	4	8	6
Timeliness of maintenance	35	5	4	4
Hardware				
Internal memory size (RAM)	70	5	6	8
Hard-drive capacity	40	9	9	5
Graphics capabilities	50	7	7	8
Processing speed	30	8	8	5
Overall performance	40	9	4	4
Expandability	50	7	2	5
Support for network technology	30	3	4	7

b. Susan Shelton did not agree with Mark's weightings and suggested the following changes:

Flexibility	60
Reputation and reliability	50
Quality of support utilities	10
Graphics capability	10

When the changes are made, which vendor should Mark recommend?

c. Mark's manager suggested the following changes to Susan's weightings:

Reputation and reliability	90
Installation assistance	40
Experience with similar systems	40
Training assistance	65
Internal memory size	10

Will the manager's changes affect the decision about which system to buy?

d. What can you conclude about point scoring from the changes made by Susan and Mark's manager? Develop your own weighting scale to evaluate the commercial software. What other selection criteria would you use? Be prepared to discuss your results with the class.

e. What are the weaknesses of the point-scoring method?

23.5 Nielsen Marketing Research (NMR), with operations in 29 countries, produces and disseminates marketing information. Nielsen has been the primary supplier of decision support information for more than 70 years. NMR's most recognizable product is the Nielsen television ratings. Nielsen is one of the largest users of computer capacity in the United States. Its information system consistently ranks above average in efficiency for its industry. NMR hired IBM to evaluate outsourcing its information processing. NMR wanted to know whether outsourcing would allow it to concentrate on giving its customers value-added services and insights, increase its flexibility, promote rapid growth, and provide it with more real-time information.

REQUIRED

What are the benefits and risks of outsourcing for NMR? Do the benefits outweigh the risks? Explain your answer.

23.6 A large organization had 18 months to replace its old customer information system with a new one that could differentiate among customer levels and provide appropriate products and services on demand. The new system, which cost $1 million and was installed by the IS staff on time, did not work properly. Complex transactions were error-prone, some transactions were canceled and others were put on hold, and the system could not differentiate among customers. The system was finally shut down, and transactions were processed manually. New IS management was hired to build a new system and mend the strained relationship between operations and IS.

So what went wrong? IS couldn't—or wouldn't—say no to all the requests for systems enhancements. Eager to please top management, IS management ignored the facts and assured them they could build a scalable system that was on time and within budget. Another big mistake was a strict project schedule with little flexibility to deal with problems and unforeseen challenges. Developers never spoke up about any glitches they encountered along the way. More than a dozen people (including the CIO) lost their jobs because of their roles in this disaster.

REQUIRED

a. What could IS management have done differently to make this project successful?
b. What in-house development issues are demonstrated in this case?
c. How could the in-house issues have been addressed to prevent the system's failure?
d. Can we conclude from this case that organizations should not have custom software written for them? Explain your answer.

23.7 Meredith Corporation publishes books and magazines, owns and operates television stations, and has a real estate marketing and franchising service. Meredith has 11 different systems that do not communicate with each other. Management wants an executive information system that provides them with the correct and timely information they need to make good business decisions. Meredith has decided to use prototyping to develop the system.

REQUIRED

a. Identify three questions you would ask Meredith personnel to determine systems requirements. What information are you attempting to elicit from each question?
b. Explain how prototyping works. What would system developers do during the iterative process? Why would you want the fewest iterations possible?
c. Would you want the prototype to be operational or nonoperational? Why? If it were an operational prototype, what would have to happen? If it were a nonoperational prototype, how could the prototype be used?
d. Suppose the company decides the prototype system is not practical, abandons it, and takes some other approach to solving its information problem. Does that mean prototyping is not a valid systems development approach? Explain your answer.

23.8 Norcom, a division of a large manufacturer, needed a new distribution and customer service system. The project was estimated to take 18 months and cost $5 million. The project team consisted of 20 business and IT staff members. After two years, the CIO was fired, and the company hired a CIO with expertise in saving troubled projects. The new CIO said three grave errors were committed.

1. IT picked the wrong software using a very naïve request for proposal process.
2. IT did not formulate a project plan.
3. No one "owned" the project. The IT staff assumed the users owned the project, the users believed the IT staff owned it, and management believed the vendor owned it.

The CIO developed a 2,000-line plan to rescue the project. Three months later, the system failed, even with IT staff and consultants working on it day and night. The

failed system was to have been the company's preeminent system, but it could not even process customer orders correctly, resulting in complaints about late shipments and receiving the wrong goods.

After three years and $4 million, the new CIO polled the staff anonymously. Only two said the project could be saved, and they had staked their careers on the project. The message that the project was not worth saving was very hard for the CIO to give. It was likewise hard for the division president to receive it; he could not accept the idea of killing a project that cost so much money. He finally accepted the decision and all the ramifications involved, including corporate IT taking control of all IT operations at his division.

REQUIRED

a. List the primary components of an RFP.
b. Identify possible components or deficiencies in Norcom's RFP that could have led the new CIO to claim that it was naïve or insufficient.
c. Identify possible approaches Norcom could have used to evaluate RFP responses.

23.9 Quickfix is rapidly losing business, and management wants to redesign its computer repair processes and procedures to decrease costs and increase customer service. Currently, a customer needing help calls one of five regional service centers. A customer service representative records the relevant customer information, finds the closest qualified technician, and calls the technician's cell phone to see whether the repair fits into his or her schedule. If not, the representative finds the next closest technician. When a technician is located, customer repair information is texted or e-mailed to the technician. The technician calls the customer and arranges to pick up the computer and replace it with a loaner. Making these arrangements takes one to two days and sometimes more if technicians are not available or do not promptly return calls.

If a broken computer cannot be quickly repaired, it is sent to a repair depot. These repairs take another four to seven days. If problems arise, it can take up to two weeks for an item to be repaired. When a customer calls to see whether the computer is ready, the service representative calls the technician to find out the status and calls the customer back. The repair process usually takes five phone calls between the customer, the service representative, and the technician.

Several problems with this process led to a significant drop in business: (1) it is time-consuming; (2) it is inconvenient for a customer to have a computer removed, a new one installed, and then the old one reinstalled; and (3) service representatives do not have immediate access to information about items being repaired. Quickfix decides to use BPM principles to redesign its business processes.

REQUIRED

a. Identify the repair processes that occur, and decide which should be redesigned.
b. Describe how the repair process can be redesigned to solve the problems identified.
c. What benefits can Quickfix achieve by redesigning the repair process?

CASE 23-1 Wong Engineering Corp.

Wong Engineering Corp (WEC) operates in 25 states and three countries. WEC faced a crucial decision: choosing network software that would maximize functionality, manageability, and end-user acceptance of the system. WEC developed and followed a four-step approach:

Step 1. ***Develop evaluation criteria.*** WEC organized a committee that interviewed users and developed the following evaluation criteria:

- Ease of use
- Scope of vendor support
- Ease of network management and administration
- Cost, speed, and performance
- Ability to access other computing platforms
- Security and control
- Fault tolerance and recovery abilities
- Ability to connect workstations to the network
- Global naming services

CASE 23-1 Continued

- Upgrade and enhancement options
- Vendor stability

WEC organized the criteria into the following four categories and prioritized them. Criteria vital to short-term and long-term business goals were given a 5. "Wish list" criteria were weighted a 3. Inapplicable criteria were given a 1.

1. Business criteria: overall business, economic, and competitive issues
2. Operational criteria: tactical issues and operating characteristics
3. Organizational criteria: networks' impact on the information systems structure
4. Technical criteria: hardware, software, and communications issues

Step 2. ***Define the operating environment.*** Several data-gathering techniques were used to collect information from which an information systems model was developed. The model revealed the need to share accounting, sales, marketing, and engineering data at three organizational levels: district, division, and home office. District offices needed access to centralized financial information to handle payroll. WEC needed a distributed network that allowed users throughout the organization to access company data.

Step 3. ***Identify operating alternatives.*** Using the criteria from step 1, committee members evaluated each commercial software package and then compared notes during a roundtable discussion.

Step 4. ***Test the software.*** The highest-scoring products were tested, and the product that fit the organization's needs the best was selected.

REQUIRED

Discuss the committee's role in the selection process. How should committee members be selected? What are the pros and cons of using a committee to make the selection?

a. What data-gathering techniques could WEC use to assess user needs? To select a vendor?
b. What is the benefit of analyzing the operating environment before selecting the software? What data-gathering techniques help a company understand the operating environment?
c. In selecting a system using the point-scoring method, how should the committee resolve scoring disputes? List at least two methods.
d. Should a purchase decision be made on the point-scoring process alone? What other procedure(s) should the committee employ in making the final selection?

AIS in Action Solutions

QUIZ KEY

1. Which of the following is not one of the difficulties software developers have experienced using the traditional systems development life cycle?

a. AIS development projects are backlogged for years. [Incorrect. This is one of the difficulties accountants have experienced using the traditional SDLC, but there is more than one correct answer.]
b. Changes are usually not possible after requirements have been frozen. [Incorrect. This is one of the difficulties accountants have experienced using the traditional SDLC, but there is more than one correct answer.]
c. The AIS developed may not meet their needs. [Incorrect. This is one of the difficulties accountants have experienced using the traditional SDLC, but there is more than one correct answer.]
► **d.** All are difficulties with the traditional SDLC. [Correct.]

2. Companies that buy rather than develop an AIS must still go through the systems development life cycle.

► **a.** true [Correct. Purchasing a system still requires a company to follow the systems development life cycle of analyzing, designing (conceptual and physical), and implementing a new system. Otherwise, the company risks not purchasing the right system for its needs.]
b. false [Incorrect.]

3. Which of the following statements is false?
   a. As a general rule, companies should buy rather than develop software if they can find commercial software that meets their needs. [Incorrect. This is a true statement, not a false statement. Purchasing software is generally less expensive than developing software in-house.]
   ► **b.** As an AIS increases in size and complexity, there is a greater likelihood that commercial software can be found that meets user needs. [Correct. This is a false statement. Large and complex systems need greater customization than smaller systems and thus are less likely to lend themselves to the one-size-fits-all approach of commercial software.]
   c. A company should not attempt to develop its own custom software unless experienced, in-house programming personnel are available and the job can be completed less expensively on the inside. [Incorrect. This is a true statement, not a false statement. Skilled in-house programmers and the promise of lower costs are essential if companies decide to develop their own custom software.]
   d. As a general rule, a company should develop custom software only when it will provide a significant competitive advantage. [Incorrect. This is a true statement, not a false statement. According to Arthur Little, companies should pursue custom software only when it provides a distinct competitive advantage.]

4. When a company is buying large and complex systems, vendors are invited to submit systems for consideration. What is such a solicitation called?
   a. request for quotation [Incorrect. A request for quotation asks for dollar bids on proposed systems or their components.]
   b. request for system [Incorrect. A request for system is not the terminology used to refer to inviting vendors to submit systems for consideration.]
   ► **c.** request for proposal [Correct. A request for proposal invites vendors to propose solutions to a company's needs.]
   d. good-faith estimate [Incorrect. A good-faith estimate provides a vendor's best guess on the cost of a proposal based on reliable parameters.]

5. To compare system performance, a company can create a data processing task with input, processing, and output jobs. This task is performed on the systems under consideration and the processing times are compared. The AIS with the lowest time is the most efficient. What is this process called?
   ► **a.** benchmarking [Correct. Benchmarking measures system performance by comparing processing times.]
   b. requirements costing [Incorrect. Requirement costing estimates the costs of purchasing or developing features that are not present in a particular AIS.]
   c. point scoring [Incorrect. Point scoring measures system performance by comparing each system based on weighted criteria.]
   d. performance testing [Incorrect. Performance testing is a general term applied to many types of comparison testing.]

6. Which of the following statements is true?
   a. Because the AIS is so crucial, companies never outsource parts of the AIS. [Incorrect. AIS functions are routinely outsourced.]
   b. Most mainframe outsourcing contracts are for two to three years and cost thousands of dollars a year. [Incorrect. Most mainframe outsourcing contracts are longer term (averaging 10 years) and cost hundreds of thousands to millions of dollars.]
   ► **c.** Outsourcers often buy the client's computers and hire all or most of its information systems employees. [Correct. Many large outsourcing deals involve purchasing the client's hardware and hiring the client's employees.]
   d. Only companies struggling to survive and wanting a quick infusion of cash from selling their hardware use outsourcing. [Incorrect. Many large and financially sound companies use outsourcing as a way to decrease costs and become even more profitable.]

7. Which of the following is not a benefit of outsourcing?

► **a.** It offers a great deal of flexibility because it is relatively easy to change outsourcers. [Correct. This is not a benefit of outsourcing. Because contracts are long term, outsourcers can be very inflexible as well as difficult and costly to change.]

b. It can provide access to the expertise and special services provided by outsourcers. [Incorrect. This is a benefit of outsourcing. Many companies cannot afford to retain information systems expertise on their payroll; therefore, outsourcing provides a less expensive way to acquire that expertise.]

c. It allows companies to move to a more sophisticated level of computing at a reasonable cost. [Incorrect. This is a benefit of outsourcing. Many companies cannot afford to maintain the most effective and sophisticated hardware; therefore, outsourcing provides a less expensive way to gain access to that hardware.]

d. It is a cost-effective way to handle the peaks and valleys found in seasonal businesses. [Incorrect. This is a benefit of outsourcing. For companies in cyclical industries, outsourcing provides an effective way to meet company needs during the busy times and to lower costs during the slow times of their business cycle.]

8. Which of the following is a true statement about prototyping?

a. In the early stages of prototyping, system controls and exception handling may be sacrificed in the interests of simplicity, flexibility, and ease of use. [Incorrect. This is a true statement. Prototyping provides simplicity, flexibility, and ease of use by sacrificing controls and exception handling. However, this is not the only true statement.]

b. A prototype is a scaled-down, first-draft model that is quickly and inexpensively built and given to users to evaluate. [Incorrect. This is a true statement. Prototypes are essentially rough-draft models. However, this is not the only true statement.]

c. The first step in prototyping is to identify system requirements. [Incorrect. This is a true statement. The first step in prototyping is to identify system requirements. However, this is not the only true statement.]

► **d.** All of the statements are true. [Correct.]

9. Which of the following is not an advantage of prototyping?

a. better definition of user needs [Incorrect. This is an advantage of prototyping. Because users can test-drive the model, they can give better feedback to the developers regarding their needs and requirements.]

► **b.** adequately tested and documented systems [Correct. This is not an advantage of prototyping. Because prototypes are developed so quickly, developers often neglect documentation and a full testing before the system becomes operational.]

c. higher user involvement and satisfaction [Incorrect. This is an advantage of prototyping. Prototyping success depends on high user involvement, which generally leads to greater user satisfaction.]

d. faster development time [Incorrect. This is an advantage of prototyping. Prototypes can be developed in a matter of days or weeks, whereas a more traditional approach can take a year or longer.]

10. When is it most appropriate to use prototyping?

a. when there is little uncertainty about the AIS [Incorrect. Prototyping is more effective when there is substantial uncertainty about how an AIS should work.]

b. when it is clear what users' needs are [Incorrect. Prototyping is more effective when users are uncertain of their needs and benefit from working on models to help them identify and solidify their needs.]

► **c.** when the final AIS cannot be clearly visualized because the decision process is still unclear [Correct. Prototyping is more effective when there is substantial uncertainty about how an AIS should work, look, and feel.]

d. when there is a very low likelihood of failure [Incorrect. Prototyping is the most effective when there is substantial uncertainty about whether a new system will work.]

COMPREHENSIVE PROBLEM SOLUTION

Identify the benefits and risks of the three courses of action facing the FTC: purchasing a prewritten software system, developing the system in-house, and outsourcing the system to an external vendor.

PURCHASING SOFTWARE

The primary benefit of purchasing software is greater availability and lower cost; because the product is sold to many companies, it can be sold at a lower price. The downside is that because the software is designed for as wide an audience as possible, it may not meet all the needs of the purchaser. In addition, software support is a problem if the vendor goes out of business.

DEVELOPING THE SYSTEM IN-HOUSE

The primary benefit of in-house development is that the system should meet the entity's needs. The drawbacks are that it occupies significant time and resources, it is usually a very complex process, and problems—such as poor requirements planning, insufficient staff, poor top-management support, inadequate communication, and a lack of cooperation between the developers and users—can easily derail a project.

OUTSOURCING THE SYSTEM

Outsourcing allows entities to devote time and resources to their core competencies instead of diverting attention to systems development. It also gives companies access to expertise at a much lower cost. Outsourcing can save 15% to 30% in overall systems development costs because of quicker development time, smoothing usage peaks and valleys, and facilitating corporate restructuring (downsizing). The risks involve a loss of control over the project, inflexible outsourcing contracts, and poor service. Entities can also lose their competitive advantage by not maintaining proprietary systems.

WHAT THE FTC DID

Given the short timeframe to implement such a large task, the FTC could not have built this system on its own. Instead, they outsourced it to AT&T, which had the expertise and staff to handle the system's analysis, design, and implementation. With AT&T's help, the project became one of the government's most successful IT projects.