Building and Managing Systems

PART IV

Part IV shows how to use the knowledge acquired in earlier chapters to analyze and design information system solutions to business problems. This part answers questions such as these: How can I develop a solution to an information system problem that provides genuine business benefits? How can the firm adjust to the changes introduced by the new system solution? What alternative approaches are available for building system solutions? What broader ethical and social issues should be addressed when building and using information systems?

Building Information Systems and Managing Projects

11

STUDENT LEARNING OBJECTIVES

After completing this chapter, you will be able to answer the following questions:

1. What are the core problem-solving steps for developing new information systems?
2. What are the alternative methods for building information systems?
3. What are the principal methodologies for modeling and designing systems?
4. How should information systems projects be selected and evaluated?
5. How should information systems projects be managed?

Chapter Outline

A NEW ORDERING SYSTEM FOR GIRL SCOUT COOKIES

Peanut Butter Petites, Caramel DeLites, Thin Mints—Girl Scout Cookies have been American favorites since the organization's first cookie drive in 1917. The Girl Scouts have been so successful selling cookies that cookie sales are a major source of funding for this organization. The Girl Scouts sell so many cookies that collecting, counting, and organizing the annual avalanche of orders has become a tremendous challenge.

The Girl Scouts' traditional cookie-ordering process depends on mountains of paperwork. During the peak sales period in January, each Girl Scout marked her sales on an individual order card and turned the card in to the troop leader when she was finished. The troop leader would transfer the information onto a five-part form and give this form to a community volunteer who tabulated the orders. From there, the orders data passed to a regional council headquarters, where they would be batched into final orders for the manufacturer, ABC Cookies. In addition to ordering, Girl Scout volunteers and troop members had to coordinate cookie deliveries, from the manufacturer to regional warehouses, to local drop-off sites, to each scout, and to the customers themselves.

The Patriots' Trail Girl Scout Council, representing 65 communities and 18,000 Girl Scouts in the greater Boston area, sold more than 1.6 million boxes of eight different cookie varieties in 2004 alone. According to its associate executive director Deborah Deacetis, the paperwork had become "overwhelming." "It changed hands too many times. There was a lot of opportunity for error, because of all the added columns, multiple prices per box, and calculations that had to be made by different people, all on deadline."

The Patriots' Trail Council first looked into building a computerized system using Microsoft Access database management and application development tools. But this alternative would have cost $25,000 to develop and would have taken at least three to four months to get the system up and running. It was too time consuming, complex, and expensive for the Girl Scouts. In addition to Microsoft Access software, the Girl Scouts would have to purchase a server to run the system, plus pay for networking and Web site maintenance services so the system could be made available on the Web.

After consulting with management consultants Dovetail Associates, the council selected Intuit's QuickBase for Corporate Workgroups. QuickBase is a hosted Web-based software service for small businesses and corporate workgroups. It is especially well suited for building simple database applications very quickly and does not require a great deal of training to use. QuickBase is customizable and designed to collect, organize, and share data among teams in many different locations.

A Dovetail consultant created a working QuickBase prototype with some basic functions for the Girl Scouts within a few hours. It only took two months to build, test, and implement the entire system using this software. The cost for developing the entire system was a fraction of the Microsoft Access solution. The Girl Scouts do not have to pay for any hardware, software, or networking services because QuickBase runs everything for them on its servers. QuickBase costs $500 per month for organizations with 100 users and $1,500 per month for organizations with up to 500 users. It is very easy to use.

The QuickBase solution eliminates paperwork and calculation errors by providing a clear central source of data for the entire council and easy online entry of cookie orders over the Web. Troop leaders collect the Girl Scouts' order cards and enter them directly into the QuickBase system using their home computers linked to the Web. With a few mouse clicks, the council office consolidates the unit totals and transmits the orders electronically to ABC Cookies.

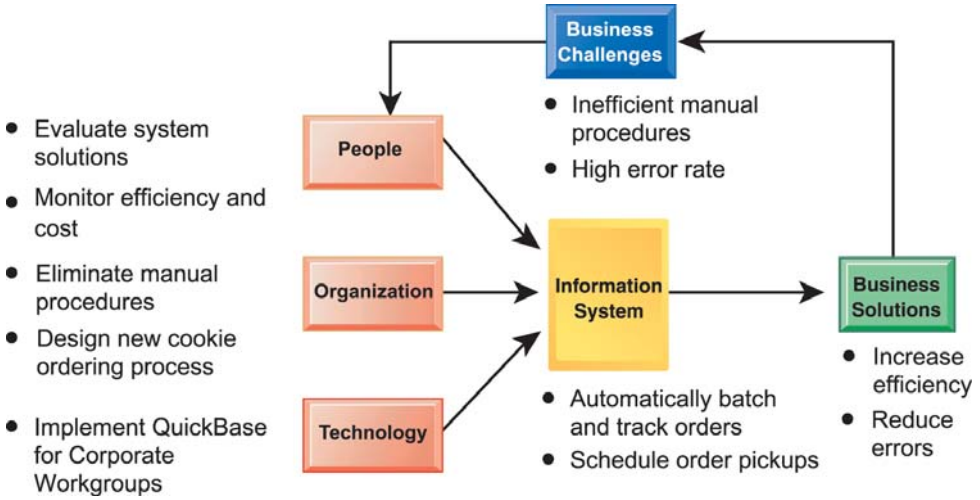
In the past the council relied on volunteers to handle their paperwork, dropping it off at the council office or mailing it in. "Now we have a way to actually watch the orders coming in," Deacetis notes. As local orders come in, local section leaders can track the data in real time.

The Patriots' Trail Girl Scout Council also uses the QuickBase system to manage the Cookie Cupboard warehouse, where volunteers pick up their cookie orders. Volunteers use the system to make reservations so that the warehouse can prepare the orders in advance, saving time and inventory management costs. The trucking companies that deliver cookie shipments now receive their instructions electronically through QuickBase so that they can create efficient delivery schedules.

Since its implementation, the QuickBase system has cut paperwork by more than 90 percent, reduced errors to 1 percent, and reduced the time spent by volunteers by 50 percent. The old system used to take two months to tally the orders and determine which Scouts should be rewarded for selling the most cookies. Now that time has been cut to 48 hours.

Sources: Intuit Inc., "QuickBase Customers: Patriots' Trail Girl Scouts," www.quickbase.com, accessed August 1, 2007 and "Girl Scouts Unite Behind Order Tracking," *Customer Relationship Management*, May 2005.

The experience of the Patriots' Trail Girl Scout Council illustrates some of the steps required to design and build new information systems. It also illustrates some of the benefits of a new system solution. The Girl Scouts had an outdated manual paper-based system for processing cookie orders that was excessively time consuming and error ridden. The Girl Scouts tried several alternative solutions before opting for a new ordering system based on the QuickBase software service. In this chapter, we will examine the Girl Scouts' search for a system solution as we describe each step of building a new information system using the problem-solving process.



HEADS UP

During your career, you will undoubtedly be asked to work on the development of a new system to solve an important challenge for your business. In the process of building new systems, you will face many choices about hardware and software, and you will have to think about how to redesign business processes and jobs in order to maximize the value derived from the new system. You will also have to manage the system-building project and any changes related to new technology and ways of working. But the most important decisions you will face involve understanding just exactly what it is you want the new system to do and what value it will bring to your firm. This chapter provides you with a methodology to guide you through the problem-solving process of building new information systems and information systems project management.

11.1 Problem Solving and Systems Development

We have already described the problem-solving process and how it helps us analyze and understand the role of information systems in business. This problem-solving process is especially valuable when we need to build new systems. A new information system is built as a solution to a problem or set of problems the organization perceives it is facing. The problem may be one in which managers and employees believe that the business is not performing as well as expected, or it may come from the realization that the organization should take advantage of new opportunities to perform more effectively.

Let's apply this problem-solving process to system building. Figure 11-1 illustrates the four steps we would need to take: (1) define and understand the problem, (2) develop alternative solutions, (3) choose the best solution, and (4) implement the solution.

Before a problem can be solved, it first must be properly defined. Members of the organization must agree that a problem actually exists and that it is serious. The problem must be investigated so that it can be better understood. Next comes a period of devising alternative solutions, then one of evaluating each alternative and selecting the best solution. The final stage is one of implementing the solution, in which a detailed design for the solution is specified, translated into a physical system, tested, introduced to the organization, and further refined as it is used over time.

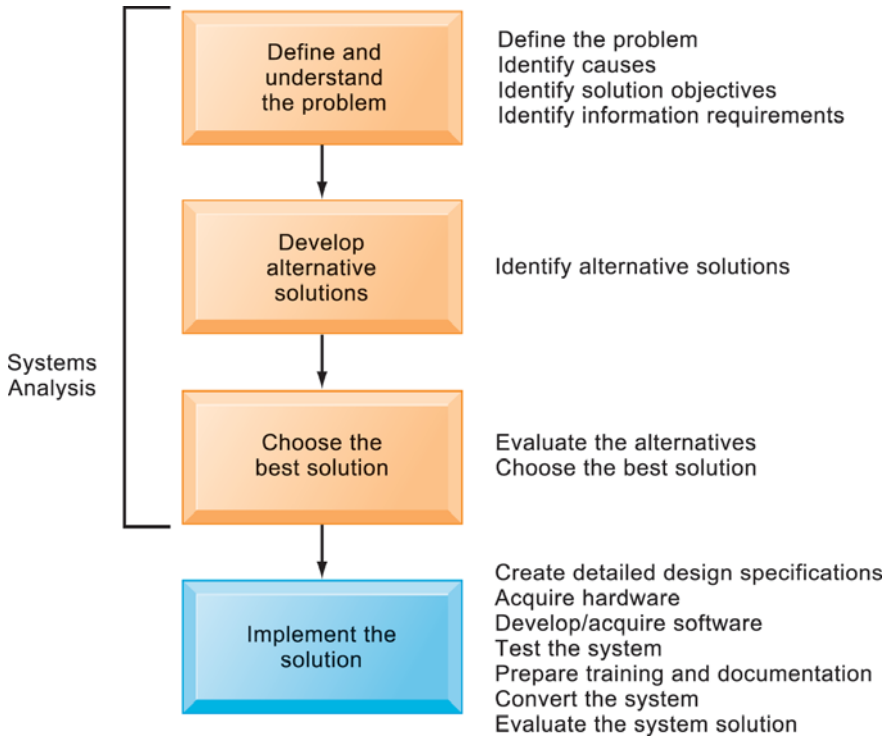


Figure 11-1
Developing an Information System Solution
Developing an information system solution is based on the problem-solving process.

In the information systems world, we have a special name for these activities. Figure 11-1 shows that the first three problem-solving steps, where we identify the problem, gather information, devise alternative solutions, and make a decision about the best solution, are called **systems analysis**.

DEFINING AND UNDERSTANDING THE PROBLEM

Defining the problem may take some work because various members of the company may have different ideas about the nature of the problem and its severity. What caused the problem? Why is it still around? Why wasn't it solved long ago? Systems analysts typically gather facts about existing systems and problems by examining documents, work papers, procedures, and system operations and by interviewing key users of the system.

Information systems problems in the business world typically result from a combination of people, organization, and technology factors. When identifying a key issue or problem, ask what kind of problem it is: Is it a people problem, an organizational problem, a technology problem, or a combination of these? What people, organizational, and technological factors contributed to the problem?

Once the problem has been defined and analyzed, it is possible to make some decisions about what should and can be done. What are the objectives of a solution to the problem? Is the firm's objective to reduce costs, increase sales, or improve relationships with customers, suppliers, or employees? Do managers have sufficient information for decision-making? What information is required to achieve these objectives?

At the most basic level, the **information requirements** of a new system identify who needs what information, where, when, and how. Requirements analysis carefully defines the objectives of the new or modified system and develops a detailed description of the functions that the new system must perform. A system designed around the wrong set of requirements will either have to be discarded because of poor performance or will need to undergo major modifications. Section 11.2 describes alternative approaches to eliciting requirements that help minimize this problem.

Let's return to our opening case about the Girl Scouts. The problem here is that the Girl Scout ordering process is heavily manual and cannot support the large number of volunteers and cookie orders that must be coordinated. As a result, cookie ordering is extremely inefficient with high error rates and volunteers spending excessive time organizing orders and deliveries.

Organizationally, the Girl Scouts are a voluntary organization distributed across a large area, with cookie sales as the primary source of revenue. The Scouts rely on volunteers with little or no business or computer experience for sales and management of orders and deliveries. They have almost no financial resources and volunteers are strapped for time. The Girl Scout cookie-ordering process requires many steps and coordination of multiple groups and organizations—individual Girl Scouts, volunteers, the council office, the cookie manufacturing factory, trucking companies, and the Cookie Cupboard warehouse.

The objectives of a solution for the Girl Scouts would be to reduce the amount of time, effort, and errors in the cookie-ordering process. Information requirements for the solution include the ability to rapidly total and organize order transactions for transmittal to ABC Cookies; the ability to track orders by type of cookie, troop, and individual Girl Scout; the ability to schedule deliveries to the Cookie Cupboard; and the ability to schedule order pickups from the Cookie Cupboard.

DEVELOPING ALTERNATIVE SOLUTIONS

What alternative solutions are possible for achieving these objectives and meeting these information requirements? The systems analysis lays out the most likely paths to follow given the nature of the problem. Some possible solutions do not require an information system solution but instead call for an adjustment in management, additional training, or refinement of existing organizational procedures. Some, however, do require modifications to the firm's existing information systems or an entirely new information system.

EVALUATING AND CHOOSING SOLUTIONS

The systems analysis includes a **feasibility study** to determine whether each proposed solution is feasible, or achievable, from a financial, technical, and organizational standpoint. The feasibility study establishes whether each alternative solution is a good investment, whether the technology needed for the system is available and can be handled by the firm's information systems staff, and whether the organization is capable of accommodating the changes introduced by the system.

A written systems proposal report describes the costs and benefits, and advantages and disadvantages of each alternative solution. Which solution is best in a financial sense? Which works best for the organization? The systems analysis will detail the costs and benefits of each alternative and the changes that the organization will have to make to use the solution effectively. We provide a detailed discussion of how to determine the business value of systems and manage change in the following section. On the basis of this report, management will select what it believes is the best solution for the company.

The Patriots' Trail Girl Scouts had three alternative solutions. One was to streamline existing processes, continuing to rely on manual procedures. However, given the large number of Girl Scouts and cookie orders, as well as relationships with manufacturers and shippers, redesigning and streamlining a manual ordering and delivery process would not have provided many benefits. The Girl Scouts needed an automated solution that accurately tracked thousands of order and delivery transactions, reduced paperwork, and created a central real-time source of sales data that could be accessed by council headquarters and individual volunteers.

A second alternative was to custom-build a cookie ordering system using Microsoft Access. This alternative was considered too time consuming, expensive, and technically challenging for the Girl Scouts. It required $25,000 in initial programming costs, plus the

purchase of hardware and networking equipment to run the system and link it to the Internet, as well as trained staff to run and maintain the system.

The third alternative was to rapidly create a system using an application service provider. QuickBase provides templates and tools for creating simple database systems in very short periods, provides the hardware for running the application and Web site, and can be accessed by many different users over the Web. This solution does not require the Girl Scouts to purchase any hardware, software, or networking technology or to maintain any information system staff to support the system. This last alternative was the most feasible for the Girl Scouts.

IMPLEMENTING THE SOLUTION

The first step in implementing a system solution is to create detailed design specifications. **Systems design** shows how the chosen solution should be realized. The system design is the model or blueprint for an information system solution and consists of all the specifications that will deliver the functions identified during systems analysis. These specifications should address all of the technical, organizational, and people components of the system solution. Table 11.1 lists the types of specifications that would be produced during system design.

A Dovetail Associates consultant elicited information requirements and created a design for the new Girl Scout cookie system. Table 11.2 shows some of the design specifications for the new system.

Completing Implementation

In the final steps of implementing a system solution, the following activities would be performed:

- *Hardware selection and acquisition*. System builders select appropriate hardware for the application. They would either purchase the necessary computers and networking hardware or lease them from a technology provider.

TABLE 11.1

System Design Specifications

Output	Medium and Content Timing
Input	Flow Data entry
User interface	Feedback and error handling
Database	Logical data model Volume and speed requirements File and record specifications
Processing	Program logic and computations
Manual procedures	What activities, who, when, how, and where
Security and controls	Access controls Input, processing, and output controls
Conversion	Testing method Conversion strategy
Training and documentation	Training modules and platforms Systems, user, and operations documentation
Organizational changes	Process design Organizational structure changes

TABLE 11.2

Design Specifications for the Girl Scout Cookie System

Output	Online reports
	Hard copy reports
	Online queries
	Order transactions for ABC Cookies
	Delivery tickets for the trucking firm
Input	Order data entry form
	Troop data entry form
	Girl Scout data entry form
	Shipping/delivery data entry form
User interface	Graphical Web interface
Database	Database with cookie order file, delivery file, troop contact file
Processing	Calculate order totals by type of cookie and number of boxes
	Track orders by troop and individual Girl Scout
	Schedule pickups at the Cookie Cupboard
	Update Girl Scout and troop data for address and member changes
Manual procedures	Girl Scouts take orders with paper forms
	Troop leaders collect order cards from Scouts and enter the order data online
Security and controls	Online passwords
	Control totals
Conversion	Input Girl Scout and troop data
	Transfer factory and delivery data
	Test system
Training and documentation	System guide for users
	Online practice demonstration
	Online training sessions
	Training for ABC Cookies and trucking companies to accept data and instructions automatically from the Girl Scout system
Organizational changes	Job design: Volunteers no longer have to tabulate orders
	Process design: Take orders on manual cards but enter them online into the system
	Schedule order pickups from the Cookie Cupboard online

- *Software development and programming*. Software is custom programmed in-house or purchased from an external source, such as an outsourcing vendor, an application software package vendor, or an application service provider.

The Girl Scouts did not have to purchase additional hardware or software. QuickBase offers templates for generating simple database applications. Dovetail consultants used the QuickBase tools to rapidly create the software for the system. The system runs on QuickBase servers.

- *Testing*. The system is thoroughly tested to ensure it produces the right results. The **testing** process requires detailed testing of individual computer programs, called **unit testing**, as well as **system testing**, which tests the performance of the information system as a whole. **Acceptance testing** provides the final certification that the system is ready to be used in a production setting. Information systems tests are evaluated by users and reviewed by management. When all parties are satisfied that the new system meets their standards, the system is formally accepted for installation.

The systems development team works with users to devise a systematic test plan. The **test plan** includes all of the preparations for the series of tests we have just described. Figure 11-2 shows a sample from a test plan that might have been used for the Girl Scout cookie system. The condition being tested is online access of an existing record for a specific Girl Scout troop.

- *Training and documentation*. End users and information system specialists require training so that they will be able to use the new system. Detailed **documentation** showing how the system works from both a technical and end-user standpoint must be prepared.

The Girl Scout cookie system provides an online practice area for users to practice entering data into the system by following step-by-step instructions. Also available on the Web is a step-by-step instruction guide for the system that can be downloaded and printed as a hard copy manual.

- ***Conversion*** is the process of changing from the old system to the new system. There are three main conversion strategies: the parallel strategy, the direct cutover strategy, and the phased approach strategy.

In a **parallel strategy**, both the old system and its potential replacement are run together for a time until everyone is assured that the new one functions correctly. The old system remains available as a backup in case of problems. The **direct cutover strategy** replaces the old system entirely with the new system on an appointed day, carrying the risk that there is no system to fall back on if problems arise. A **phased approach** introduces the system in stages (such as first introducing the modules for ordering Girl Scout cookies and then introducing the modules for transmitting orders and instructions to the cookie factory and shipper).

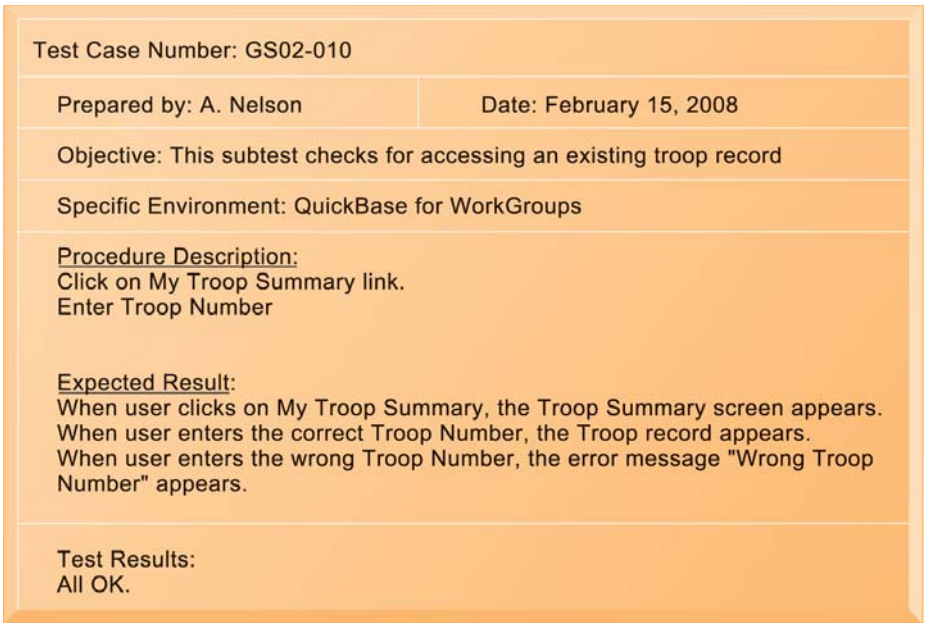


Figure 11-2
A Sample Test Plan for the Girl Scout Cookie System
When developing a test plan, it is imperative to include the various conditions to be tested, the requirements for each condition tested, and the expected results. Test plans require input from both end users and information systems specialists.

- *Production and maintenance*. After the new system is installed and conversion is complete, the system is said to be in **production**. During this stage, users and technical specialists review the solution to determine how well it has met its original objectives and to decide whether any revisions or modifications are in order. Changes in hardware, software, documentation, or procedures to a production system to correct errors, meet new requirements, or improve processing efficiency are termed **maintenance.**

The Girl Scouts continued to improve and refine their QuickBase cookie system. In 2005, the system was made more efficient for users with slow Internet connections. Other recent enhancements include capabilities for paying for orders more rapidly, entering troop information and initial orders without waiting for a specified starting date, and receiving online confirmation for reservations to pick up orders from the Cookie Cupboard.

Managing the Change

Developing a new information systems solution is not merely a matter of installing hardware and software. The business must also deal with the organizational changes that the new solution will bring about—new information, new business processes, and perhaps new reporting relationships and decision-making power. A very well-designed solution may not work unless it is introduced to the organization very carefully. The process of planning change in an organization so that it is implemented in an orderly and effective manner is so critical to the success or failure of information system solutions that we devote the next section to a detailed discussion of this topic.

To manage the transition from the old manual cookie-ordering processes to the new system, the Girl Scouts would have to inform troop leaders and volunteers about changes in cookie-ordering procedures, provide training, and provide resources for answering any questions that arose as parents and volunteers started using the system. They would need to work with ABC Cookies and their shippers on new procedures for transmitting and delivering orders.

11.2 Alternative Systems-Building Approaches

There are alternative methods for building systems using the basic problem-solving model we have just described. These alternative methods include the traditional systems lifecycle, prototyping, end-user development, application software packages, and outsourcing.

TRADITIONAL SYSTEMS DEVELOPMENT LIFECYCLE

The **systems development lifecycle (SDLC)** is the oldest method for building information systems. The lifecycle methodology is a phased approach to building a system, dividing systems development into a series of formal stages, as illustrated in Figure 11-3. Although systems builders can go back and forth among stages in the lifecycle, the systems lifecycle is predominantly a "waterfall" approach in which tasks in one stage are completed before work for the next stage begins.

This approach maintains a very formal division of labor between end users and information systems specialists. Technical specialists, such as system analysts and programmers, are responsible for much of the systems analysis, design, and implementation work; end users are limited to providing information requirements and reviewing the technical staff's work. The lifecycle also emphasizes formal specifications and paperwork, so many documents are generated during the course of a systems project.

The systems lifecycle is still used for building large complex systems that require rigorous and formal requirements analysis, predefined specifications, and tight controls over the systems-building process. However, this approach is also time consuming and expensive to use. Tasks in one stage are supposed to be completed before work for the next stage begins. Activities can be repeated, but volumes of new documents must be generated and steps retraced if requirements and specifications need to be revised. This encourages

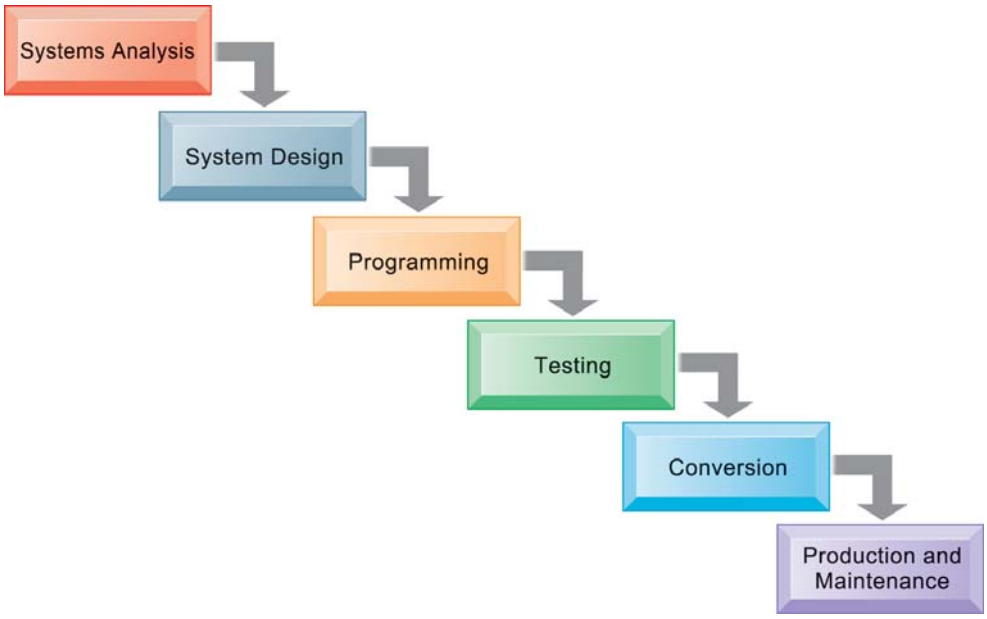


Figure 11-3
The Traditional Systems Development Lifecycle
The systems development lifecycle partitions systems development into formal stages, with each stage requiring completion before the next stage can begin.

freezing of specifications relatively early in the development process. The lifecycle approach is also not suitable for many small desktop systems, which tend to be less structured and more individualized.

PROTOTYPING

Prototyping consists of building an experimental system rapidly and inexpensively for end users to evaluate. The prototype is a working version of an information system or part of the system, but it is intended as only a preliminary model. Users interact with the prototype to get a better idea of their information requirements, refining the prototype multiple times. The chapter-opening case describes how Dovetail Associates used QuickBase to create a prototype that helped the Patriots' Trail Girl Scout Council refine their specifications for their cookie ordering system. When the design is finalized, the prototype will be converted to a polished production system. Figure 11-4 shows a four-step model of the prototyping process.

Step 1: *Identify the user's basic requirements*. The system designer (usually an information systems specialist) works with the user only long enough to capture the user's basic information needs.

Step 2: *Develop an initial prototype*. The system designer creates a working prototype quickly, using tools for rapidly generating software.

Step 3: *Use the prototype*. The user is encouraged to work with the system to determine how well the prototype meets his or her needs and to make suggestions for improving the prototype.

Step 4: *Revise and enhance the prototype*. The system builder notes all changes the user requests and refines the prototype accordingly. After the prototype has been revised, the cycle returns to step 3. Steps 3 and 4 are repeated until the user is satisfied.

Prototyping is especially useful in designing an information system's user interface. Because prototyping encourages intense end-user involvement throughout the systems-development process, it is more likely to produce systems that fulfill user requirements.

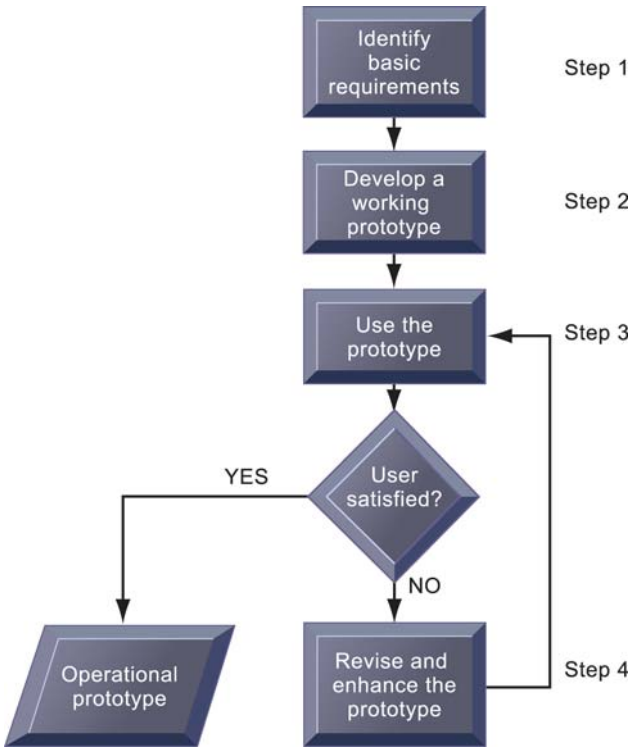


Figure 11-4
The Prototyping Process
The process of developing a prototype consists of four steps. Because a prototype can be developed quickly and inexpensively, systems builders can go through several iterations, repeating steps 3 and 4, to refine and enhance the prototype before arriving at the final operational one.

However, rapid prototyping may gloss over essential steps in systems development, such as thorough testing and documentation. If the completed prototype works reasonably well, management may not see the need to build a polished production system. Some hastily constructed systems do not easily accommodate large quantities of data or a large number of users in a production environment.

END-USER DEVELOPMENT

End-user development allows end users, with little or no formal assistance from technical specialists, to create simple information systems, reducing the time and steps required to produce a finished application. Using fourth-generation languages, graphics languages, and PC software tools, end users can access data, create reports, and develop entire information systems on their own, with little or no help from professional systems analysts or programmers.

For example Elie Tahari Ltd., a leading designer of women's fashions, uses InformationBuilders Inc.'s WebFOCUS software to enable authorized users to obtain self-service reports on orders, inventory, sales, and finance. Sales executives use the system to view their accounts, to determine what merchandise is selling, and to see what customers have ordered. Users can also create ad-hoc reports by themselves to obtain specific pieces of information or more detailed data (Information Builders, 2007).

On the whole, end-user-developed systems are completed more rapidly than those developed with conventional programming tools. Allowing users to specify their own business needs improves requirements gathering and often leads to a higher level of user involvement and satisfaction with the system. However, fourth-generation tools still cannot replace conventional tools for some business applications because they cannot easily handle the processing of large numbers of transactions or applications with extensive procedural logic and updating requirements.

End-user development also poses organizational risks because systems are created rapidly, without a formal development methodology, testing, and documentation. To help

organizations maximize the benefits of end-user applications development, management should require cost justification of end-user information system projects and establish hardware, software, and quality standards for user-developed applications.

PURCHASING SOLUTIONS: APPLICATION SOFTWARE PACKAGES AND OUTSOURCING

Chapter 4 points out that the software for most systems today is not developed in-house but is purchased from external sources. Firms may choose to purchase a software package from a commercial vendor, rent the software from a service provider, or outsource the development work to another firm. Selection of the software or software service is often is based on a **Request for Proposal (RFP)**, which is a detailed list of questions submitted to external vendors to see how well they meet the requirements for the proposed system.

Application Software Packages

Most new information systems today are built using an application software package or preprogrammed software components. Many applications are common to all business organizations—for example, payroll, accounts receivable, general ledger, or inventory control. For such universal functions with standard processes that do not change a great deal over time, a generalized system will fulfill the requirements of many organizations.

If a software package can fulfill most of an organization's requirements, the company does not have to write its own software. The company saves time and money by using the prewritten, predesigned, pretested software programs from the package.

Many packages include capabilities for customization to meet unique requirements not addressed by the package software. **Customization** features allow a software package to be modified to meet an organization's unique requirements without destroying the integrity of the packaged software. However, if extensive customization is required, additional programming and customization work may become so expensive and time consuming that it negates many of the advantages of software packages. If the package cannot be customized, the organization will have to adapt to the package and change its procedures.

Outsourcing

If a firm does not want to use its internal resources to build or operate information systems, it can outsource the work to an external organization that specializes in providing these services. Software service providers, which we describe in Chapter 4, are one form of outsourcing. An example would be the Girl Scouts leasing the software and hardware from QuickBase to run their cookie-ordering system. Subscribing companies use the software and computer hardware of the service provider as the technical platform for their systems. In another form of outsourcing, a company would hire an external vendor to design and create the software for its system, but that company would operate the system on its own computers.

The outsourcing vendor might be domestic or in another country. Domestic outsourcing is driven primarily by the fact that outsourcing firms possess skills, resources, and assets which their clients do not have. Installing a new supply chain management system in a very large company might require hiring an additional 30–50 people with specific expertise in supply chain management software licensed, say, from Manugistics or another vendor. Rather than hire permanent new employees, most of whom would need extensive training in the software package, and then release them after the new system is built, it makes more sense, and is often less expensive, to outsource this work for a 12 month period.

In the case of offshore outsourcing, the decision tends to be much more cost-driven. A skilled programmer in India or Russia earns about U.S. $10,000 per year, compared to $70,000 per year for a comparable programmer in the United States. The Internet and low-cost communications technology have drastically reduced the expense and

difficulty of coordinating the work of global teams in faraway locations. In addition to cost savings, many offshore outsourcing firms offer world-class technology assets and skills.

For example, Pinnacle West Capital Corporation, which sells and delivers electricity and energy-related services to one million customers in the western United States, turned to outsourcing to reduce operational costs. It contracted with the Indian software and service provider Wipro Ltd. to handle its application development. Wipro develops Pinnacle West's applications, services system enhancements, and provides 24-hour system support. Outsourcing to Wipro helped Pinnacle West accomplish 12 months of development work in 7 months while reducing computer processing and application development costs (Wipro, 2007).

There is a very strong chance that at some point in your career, you'll be working with offshore outsourcers or global teams. Your firm is most likely to benefit from outsourcing if it takes the time to evaluate all the risks and to make sure outsourcing is appropriate for its particular needs. Any company that outsources its applications must thoroughly understand the project, including its requirements, method of implementation, source of expected benefits, cost components, and metrics for measuring performance.

Many firms underestimate costs for identifying and evaluating vendors of information technology services, for transitioning to a new vendor, for improving internal software development methods to match those of outsourcing vendors, and for monitoring vendors to make sure they are fulfilling their contractual obligations. Outsourcing offshore incurs additional costs for coping with cultural differences that drain productivity and dealing with human resources issues, such as terminating or relocating domestic employees. These hidden costs undercut some of the anticipated benefits from outsourcing. Firms should be especially cautious when using an outsourcer to develop or to operate applications that give it some type of competitive advantage.

Figure 11-5 shows best- and worst-case scenarios for the total cost of an offshore outsourcing project. It shows how much hidden costs affect the total project cost. The best case reflects the lowest estimates for additional costs, and the worst case reflects the highest estimates for these costs. As you can see, hidden costs increase the total cost of an offshore outsourcing project by an extra 15 to 57 percent. Even with these extra costs, many firms will benefit from offshore outsourcing if they manage the work well.

TOTAL COST OF OFFSHORE OUTSOURCING

Cost of outsourcing contract			**$10,000,000**	
Hidden Costs	Best Case	Additional Cost ($)	Worst Case	Additional Cost ($)
1. Vendor selection	0.2%	20,000	2%	200,000
2. Transition costs	2%	200,000	3%	300,000
3. Layoffs & retention	3%	300,000	5%	500,000
4. Lost productivity/cultural issues	3%	300,000	27%	2,700,000
5. Improving development processes	1%	100,000	10%	1,000,000
6. Managing the contract	6%	600,000	10%	1,000,000
Total additional costs		**1,520,000**		**5,700,000**
	Outstanding Contract ($)	Additional Cost ($)	Total Cost ($)	Additional Cost
Total cost of outsourcing (TCO) best case	10,000,000	1,520,000	11,520,000	15.2%
Total cost of outsourcing (TCO) worst case	10,000,000	5,700,000	15,700,000	57.0%

Figure 11-5
Total Cost of Offshore Outsourcing
If a firm spends $10 million on offshore outsourcing contracts, that company will actually spend 15.2 percent in extra costs even under the best-case scenario. In the worst-case scenario, where there is a dramatic drop in productivity along with exceptionally high transition and layoff costs, a firm can expect to pay up to 57 percent in extra costs on top of the $10 million outlay for an offshore contract.

RAPID APPLICATION DEVELOPMENT FOR E-BUSINESS

Technologies and business conditions are changing so rapidly that agility and scalability have become critical elements of system solutions. Companies are adopting shorter, more informal development processes for many of their e-commerce and e-business applications, processes that provide fast solutions that do not disrupt their core transaction processing systems and organizational databases. In addition to using software packages, application service providers, and other outsourcing services, they are relying more heavily on fast-cycle techniques, such as joint application design (JAD), prototypes, and reusable standardized software components that can be assembled into a complete set of services for e-commerce and e-business.

The term **rapid application development (RAD)** refers to the process of creating workable systems in a very short period of time. RAD includes the use of visual programming and other tools for building graphical user interfaces, iterative prototyping of key system elements, the automation of program code generation, and close teamwork among end users and information systems specialists. Simple systems often can be assembled from prebuilt components (see Section 11.3). The process does not have to be sequential, and key parts of development can occur simultaneously.

Sometimes a technique called **joint application design (JAD)** will be used to accelerate the generation of information requirements and to develop the initial systems design. JAD brings end users and information systems specialists together in an interactive session to discuss the system's design. Properly prepared and facilitated, JAD sessions can significantly speed up the design phase and involve users at an intense level.

11.3 Modeling and Designing Systems

We have just described alternative methods for building systems. There are also alternative methodologies for modeling and designing systems. The two most prominent are structured methodologies and object-oriented development.

STRUCTURED METHODOLOGIES

Structured methodologies have been used to document, analyze, and design information systems since the 1970s. **Structured** refers to the fact that the techniques are step by step, with each step building on the previous one. Structured methodologies are top-down, progressing from the highest, most abstract level to the lowest level of detail—from the general to the specific.

Structured development methods are process-oriented, focusing primarily on modeling the processes, or actions, that capture, store, manipulate, and distribute data as the data flow through a system. These methods separate data from processes. A separate programming procedure must be written every time someone wants to take an action on a particular piece of data. The procedures act on data that the program passes to them.

The primary tool for representing a system's component processes and the flow of data between them is the **data flow diagram (DFD)**. The data flow diagram offers a logical graphic model of information flow, partitioning a system into modules that show manageable levels of detail. It rigorously specifies the processes or transformations that occur within each module and the interfaces that exist between them.

Figure 11-6 shows a simple data flow diagram for a mail-in university course registration system. The rounded boxes represent processes, which portray the transformation of data. The square box represents an external entity, which is an originator or receiver of information located outside the boundaries of the system being modeled. The open rectangles represent data stores, which are either manual or automated inventories of data. The arrows represent data flows, which show the movement between processes, external entities, and data stores. They always contain packets of data with the name or content of each data flow listed beside the arrow.

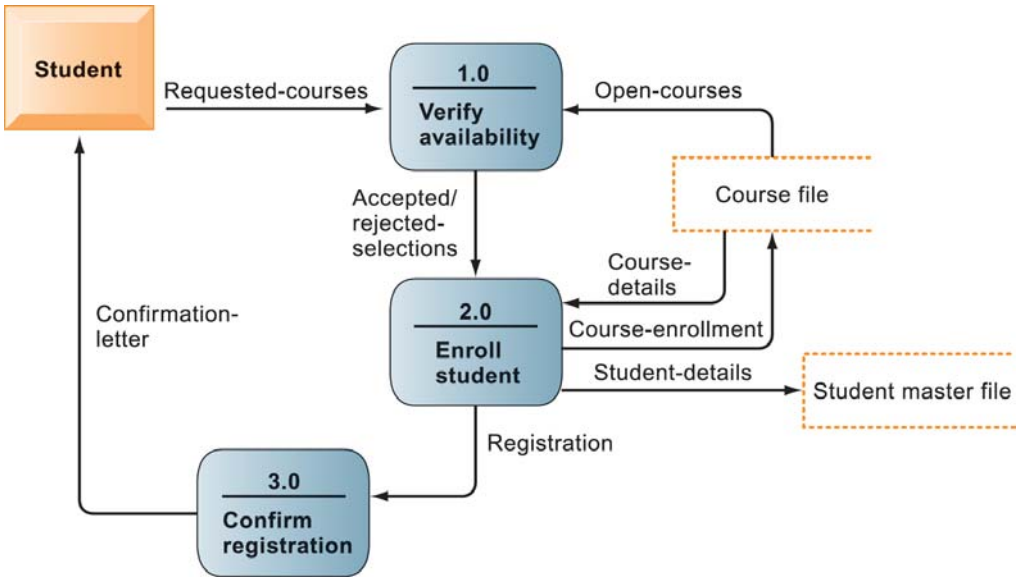


Figure 11-6
Data Flow Diagram for Mail-in University Registration System
The system has three processes: Verify availability (1.0), Enroll student (2.0), and Confirm registration (3.0). The name and content of each of the data flows appear adjacent to each arrow. There is one external entity in this system: the student. There are two data stores: the student master file and the course file.

This data flow diagram shows that students submit registration forms with their names, identification numbers, and the numbers of the courses they wish to take. In process 1.0 the system verifies that each course selected is still open by referencing the university's course file. The file distinguishes courses that are open from those that have been canceled or filled. Process 1.0 then determines which of the student's selections can be accepted or rejected. Process 2.0 enrolls the student in the courses for which he or she has been accepted. It updates the university's course file with the student's name and identification number and recalculates the class size. If maximum enrollment has been reached, the course number is flagged as closed. Process 2.0 also updates the university's student master file with information about new students or changes in address. Process 3.0 then sends each student applicant a confirmation-of-registration letter listing the courses for which he or she is registered and noting the course selections that could not be fulfilled.

Through leveled data flow diagrams, a complex process can be broken down into successive levels of detail. An entire system can be divided into subsystems with a high-level data flow diagram. Each subsystem, in turn, can be divided into additional subsystems with lower-level data flow diagrams, and the lower-level subsystems can be broken down again until the lowest level of detail has been reached. **Process specifications** describe the transformation occurring within the lowest level of the data flow diagrams, showing the logic for each process.

In structured methodology, software design is modeled using hierarchical structure charts. The **structure chart** is a top-down chart, showing each level of design, its relationship to other levels, and its place in the overall design structure. The design first considers the main function of a program or system, then breaks this function into subfunctions, and decomposes each subfunction until the lowest level of detail has been reached. Figure 11-7 shows a high-level structure chart for a payroll system. If a design has too many levels to fit onto one structure chart, it can be broken down further on more detailed structure charts. A structure chart may document one program, one system (a set of programs), or part of one program.

OBJECT-ORIENTED DEVELOPMENT

Structured methods treat data and processes as logically separate entities, whereas in the real world such separation seems unnatural. Different modeling conventions are used for analysis (the data flow diagram) and for design (the structure chart).

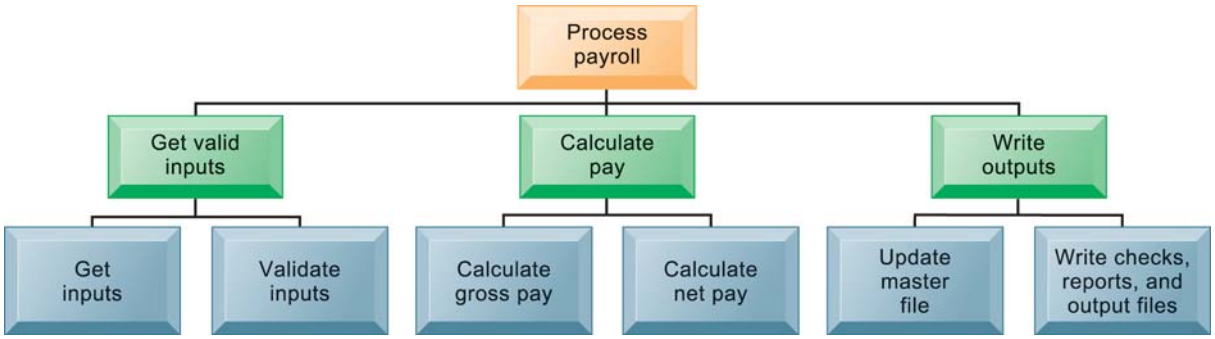


Figure 11-7 High-Level Structure Chart for a Payroll System
This structure chart shows the highest or most abstract level of design for a payroll system, providing an overview of the entire system.

Object-oriented development addresses these issues. Object-oriented development uses the object, which we introduced in Chapter 4, as the basic unit of systems analysis and design. An object combines data and the specific processes that operate on those data. Data encapsulated in an object can be accessed and modified only by the operations, or methods, associated with that object. Instead of passing data to procedures, programs send a message for an object to perform an operation that is already embedded in it. The system is modeled as a collection of objects and the relationships among them. Because processing logic resides within objects rather that in separate software programs, objects must collaborate with each other to make the system work.

Object-oriented modeling is based on the concepts of *class* and *inheritance*. Objects belonging to a certain class, or general categories of similar objects, have the features of that class. Classes of objects in turn inherit all the structure and behaviors of a more general class and then add variables and behaviors unique to each object. New classes of objects are created by choosing an existing class and specifying how the new class differs from the existing class, instead of starting from scratch each time.

We can see how class and inheritance work in Figure 11-8, which illustrates the relationships among classes concerning employees and how they are paid. Employee is the common ancestor, or superclass, for the other three classes. Salaried, Hourly, and Temporary are subclasses of Employee. The class name is in the top compartment, the attributes for each class are in the middle portion of each box, and the list of operations is in the bottom portion of each box. The features that are shared by all employees (ID, name, address, date hired, position, and pay) are stored in the Employee superclass, whereas each subclass stores features that are specific to that particular type of employee. Specific to Hourly employees, for example, are their hourly rates and overtime rates. A solid line from the subclass to the superclass is a generalization path showing that the subclasses Salaried, Hourly, and Temporary have common features that can be generalized into the superclass Employee.

Object-oriented development is more iterative and incremental than traditional structured development. During systems analysis, systems builders document the functional requirements of the system, specifying its most important properties and what the proposed system must do. Interactions between the system and its users are analyzed to identify objects, which include both data and processes. The object-oriented design phase describes how the objects will behave and how they will interact with one other. Similar objects are grouped together to form a class, and classes are grouped into hierarchies in which a subclass inherits the attributes and methods from its superclass.

The information system is implemented by translating the design into program code, reusing classes that are already available in a library of reusable software objects and adding new ones created during the object-oriented design phase. Implementation may also involve

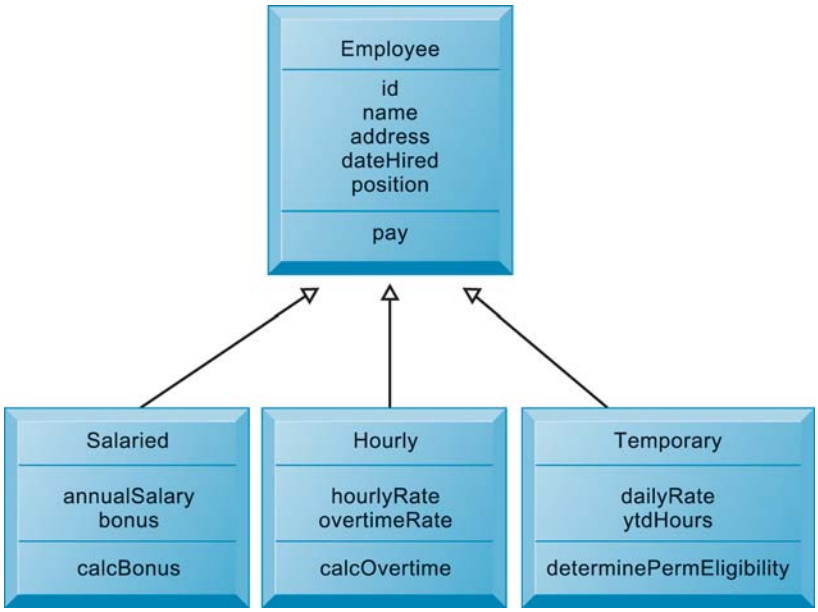


Figure 11-8
Class and Inheritance
This figure illustrates how classes inherit the common features of their superclass.

the creation of an object-oriented database. The resulting system must be thoroughly tested and evaluated.

Because objects are reusable, object-oriented development could potentially reduce the time and cost of writing software if organizations reuse software objects that have already been created as building blocks for other applications. New systems can be created by using some existing objects, changing others, and adding a few new objects.

Component-Based Development and Web Services

To further expedite software creation, groups of objects have been assembled into software components for common functions, such as a graphical user interface or online ordering capability, and these components can be combined to create large-scale business applications. This approach to software development is called **component-based development**. Businesses are using component-based development to create their e-commerce applications by combining commercially available components for shopping carts, user authentication, search engines, and catalogs with pieces of software for their own unique business requirements.

Chapter 4 introduced Web services as loosely coupled, reusable software components based on extensible markup language (XML) and other open protocols and standards that enable one application to communicate with another with no custom programming required. In addition to supporting internal and external integration of systems, Web services provide nonproprietary tools for building new information system applications or enhancing existing systems. For example, eDiets.com, an online provider of diet information and packaged meals to dieters, used an address-verification Web service from StrikeIron.com to quickly build its system for shipping meals to the right address (Babcock, 2006).

COMPUTER-AIDED SOFTWARE ENGINEERING (CASE)

Computer-aided software engineering (CASE)—sometimes called computer-aided systems engineering—provides software tools to automate the methodologies we have just described to reduce the amount of repetitive work in systems development. CASE tools provide automated graphics facilities for producing charts and diagrams, screen and report generators, data dictionaries, extensive reporting facilities, analysis and checking tools, code generators, and documentation generators. CASE tools also contain features for validating design diagrams and specifications.

CASE tools facilitate clear documentation and coordination of team development efforts. Team members can share their work by accessing each other's files to review or modify what has been done. Modest productivity benefits are achieved if the tools are used properly. Many CASE tools are PC based, with powerful graphical capabilities.

11.4 Project Management

Your company might have developed what appears to be an excellent system solution. Yet when the system is in use, it does not work properly or it doesn't deliver the benefits that were promised. If this occurs, your firm is not alone. There is a very high failure rate among information systems projects because they have not been properly managed. The Standish Group consultancy, which monitors IT project success rates, found that only 29 percent of all technology investments were completed on time, on budget, and with all features and functions originally specified (Levinson, 2006). Firms may have incorrectly assessed the business value of the new system or were unable to manage the organizational change required by the new technology. That's why it's essential to know how to manage information systems projects and the reasons why they succeed or fail.

PROJECT MANAGEMENT OBJECTIVES

A **project** is a planned series of related activities for achieving a specific business objective. Information systems projects include the development of new information systems, enhancing existing systems, or projects for replacing or upgrading the firm's information technology (IT) infrastructure.

Project management refers to the application of knowledge, skills, tools, and techniques to achieve specific targets within specified budget and time constraints. Project management activities include planning the work, assessing risk, estimating resources required to accomplish the work, organizing the work, acquiring human and material resources, assigning tasks, directing activities, controlling project execution, reporting progress, and analyzing the results. As in other areas of business, project management for information systems must deal with five major variables: scope, time, cost, quality, and risk.

Scope defines what work is or is not included in a project. For example, the scope of a project for a new order processing system might include new modules for inputting orders and transmitting them to production and accounting but not any changes to related accounts receivable, manufacturing, distribution, or inventory control systems. Project management defines all the work required to complete a project successfully, and should ensure that the scope of a project not expand beyond what was originally intended.

Time is the amount of time required to complete the project. Project management typically establishes the amount of time required to complete major components of a project. Each of these components is further broken down into activities and tasks. Project management tries to determine the time required to complete each task and establish a schedule for completing the work.

Cost is based on the time to complete a project multiplied by the daily cost of human resources required to complete the project. Information systems project costs also include the cost of hardware, software, and work space. Project management develops a budget for the project and monitors ongoing project expenses.

Quality is an indicator of how well the end result of a project satisfies the objectives specified by management. The quality of information systems projects usually boils down to improved organizational performance and decision making. Quality also considers the accuracy and timeliness of information produced by the new system and ease of use.

Risk refers to potential problems that would threaten the success of a project. These potential problems might prevent a project from achieving its objectives by increasing time and cost, lowering the quality of project outputs, or preventing the project from being completed altogether. We discuss the most important risk factors for information systems projects later in this section.

SELECTING PROJECTS: MAKING THE BUSINESS CASE FOR A NEW SYSTEM

Companies typically are presented with many different projects for solving problems and improving performance. There are far more ideas for systems projects than there are resources. You will need to select the projects that promise the greatest benefit to the business.

Determining Project Costs and Benefits

As we pointed out earlier, the systems analysis includes an assessment of the economic feasibility of each alternative solution—whether each solution represents a good investment for the company. In order to identify the information systems projects that will deliver the most business value, you'll need to identify their costs and benefits and how they relate to the firm's information systems plan.

Table 11.3 lists some of the more common costs and benefits of systems. **Tangible benefits** can be quantified and assigned a monetary value. **Intangible benefits**, such as more efficient customer service or enhanced decision making, cannot be immediately quantified. Yet systems that produce mainly intangible benefits may still be good investments if they produce quantifiable gains in the long run.

To determine the benefits of a particular solution, you'll need to calculate all of its costs and all of its benefits. Obviously, a solution where costs exceed benefits should be rejected. But even if the benefits outweigh the costs, some additional financial analysis is required to determine whether the investment represents a good return on the firm's invested capital. Capital budgeting methods, such as net present value, internal rate of return (IRR), or accounting rate of return on investment (ROI), would typically be employed to evaluate the proposed information system solution as an investment. You can find out more about how these capital budgeting methods are used to justify information system investments in our Learning Tracks.

TABLE 11.3

Costs and Benefits of Information Systems

IMPLEMENTATION COSTS	INTANGIBLE BENEFITS
Hardware	Improved asset utilization
Telecommunications	Improved resource control
Software	Improved organizational planning
Personnel costs	Increased organizational flexibility
OPERATIONAL COSTS	More timely information
Computer processing time	More information
Maintenance	Increased organizational learning
Operating staff	Legal requirements attained
User time	Enhanced employee goodwill
Ongoing training costs	Increased job satisfaction
Facility costs	Improved decision making
TANGIBLE BENEFITS	Improved operations
Increased productivity	Higher client satisfaction
Lower operational costs	Better corporate image
Reduced workforce	
Lower computer expenses	
Lower outside vendor costs	
Lower clerical and professional costs	
Reduced rate of growth in expenses	
Reduced facility costs	
Increased sales	

Some of the tangible benefits obtained by the Girl Scouts were increased productivity and lower operational costs resulting from automating the ordering process and from reducing errors. Intangible benefits include enhanced volunteer job satisfaction and improved operations.

The Information Systems Plan

An **information systems plan** shows how specific information systems fit into a company's overall business plan and business strategy. Table 11.4 lists the major components of such a plan. The plan contains a statement of corporate goals and specifies how information technology will help the business attain these goals. The report shows how general goals will be achieved by specific systems projects. It identifies specific target dates and milestones that can be used later to evaluate the plan's progress in terms of how many objectives were actually attained in the time frame specified in the plan. The plan indicates the key management decisions concerning hardware acquisition; telecommunications; centralization/decentralization of authority, data, and hardware; and required organizational change.

The plan should describe organizational changes, including management and employee training requirements; changes in business processes; and changes in authority, structure, or management practice. When you are making the business case for a new information system project, you show how the proposed system fits into that plan.

Portfolio Analysis

Once you have determined the overall direction of systems development, **portfolio analysis** will help you evaluate alternative system projects. Portfolio analysis inventories all of the firm's information systems projects and assets, including infrastructure, outsourcing contracts, and licenses. This portfolio of information systems investments can be described as having a certain profile of risk and benefit to the firm (see Figure 11-9), similar to a financial portfolio. Each information systems project carries its own set of risks and benefits. Firms try to improve the return on their information system portfolios by balancing the risk and return from their systems investments.

Obviously, you begin first by focusing on systems of high benefit and low risk. These promise early returns and low risks. Second, high-benefit, high-risk systems should be examined; low-benefit, high-risk systems should be totally avoided; and low-benefit, low-risk systems should be reexamined for the possibility of rebuilding and replacing them with more desirable systems having higher benefits. By using portfolio analysis, management can determine the optimal mix of investment risk and reward for their firms, balancing riskier, high-reward projects with safer, lower-reward ones.

The U.S. Army's Office of the CIO/G6, which oversees an annual IT budget of more than $7 billion and manages over 1500 systems and programs, uses portfolio analysis to

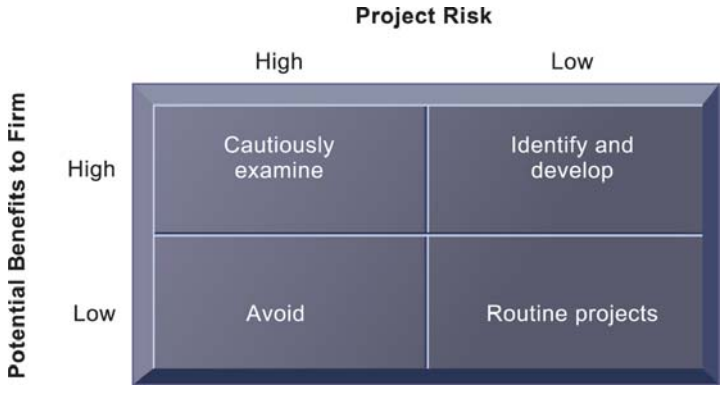


Figure 11-9
A System Portfolio
Companies should examine their portfolio of projects in terms of potential benefits and likely risks. Certain kinds of projects should be avoided altogether and others developed rapidly. There is no ideal mix. Companies in different industries have different information systems needs.

TABLE 11.4

Information Systems Plan

1. **Purpose of the Plan**
   - Overview of plan contents
   - Current business organization and future organization
   - Key business processes
   - Management strategy
2. **Strategic Business Plan Rationale**
   - Current situation
   - Current business organization
   - Changing environments
   - Major goals of the business plan
   - Firm's strategic plan
3. **Current Systems**
   - Major systems supporting business functions and processes
   - Current infrastructure capabilities
     - Hardware
     - Software
     - Database
     - Telecommunications and the Internet
   - Difficulties meeting business requirements
   - Anticipated future demands
4. **New Developments**
   - New system projects
     - Project descriptions
     - Business rationale
     - Applications' role in strategy
   - New infrastructure capabilities required
     - Hardware
     - Software
     - Database
     - Telecommunications and the Internet
5. **Management Strategy**
   - Acquisition plans
   - Milestones and timing
   - Organizational realignment
   - Internal reorganization
   - Management controls
   - Major training initiatives
   - Personnel strategy
6. **Implementation of the Plan**
   - Anticipated difficulties in implementation
   - Progress reports
7. **Budget Requirements**
   - Requirements
   - Potential savings
   - Financing
   - Acquisition cycle

inventory, evaluate, and rank its IT investments. Portfolio analysis helped the Office identify redundant systems and ensure that its IT investments provide needed capabilities. ("Winning the IT Portfolio Battle," 2007).

Another method for evaluating alternative system solutions is a **scoring model**. Scoring models give alternative systems a single score based on the extent to which they meet selected objectives. Table 11.5 shows part of a simple scoring model that could have been used by the Girl Scouts in evaluating their alternative systems. The first column lists the criteria that decision makers use to evaluate the systems. Table 11.5 shows that the Girl Scouts attach the most importance to capabilities for sales order processing, ease of use, ability to support users in many different locations, and low cost. The second column in Table 11.5 lists the weights that decision makers attached to the decision criteria. Columns 3 and 5 show the percentage of requirements for each function that each alternative system meets. Each alternative's score is calculated by multiplying the percentage of requirements met for each function by the weight attached to that function. The QuickBase solution has the highest total score.

MANAGING PROJECT RISK AND SYSTEM-RELATED CHANGE

Some systems development projects are more likely to run into problems or to suffer delays because they carry a much higher level of risk than others. The level of project risk is influenced by project size, project structure, and the level of technical expertise of the information systems staff and project team. The larger the project—as indicated by the

TABLE 11.5

Example of a Scoring Model for the Girl Scouts Cookie System

Criteria	Weight	Microsoft Access System (%)	Microsoft Access System Score	QuickBase System (%)	QuickBase System Score
1.0 Order processing					
1.1 Online order entry	5	67	335	83	415
1.2 Order tracking by troop	5	81	405	87	435
1.3 Order tracking by individual Girl Scout	5	72	360	80	400
1.4 Reserving warehouse pickups	3	66	198	79	237
Total order processing			1,298		1,487
2.0 Ease of use					
2.1 Web access from multiple locations	5	55	275	92	460
2.2 Short training time	4	79	316	85	340
2.3 User-friendly screens and data entry forms	4	65	260	87	348
Total ease of use			851		1,148
3.0 Costs					
3.1 Software costs	3	51	153	65	195
3.2 Hardware (server) costs	4	57	228	90	360
3.3 Maintenance and support costs	4	42	168	89	356
Total costs			549		911
Grand Total			2,698		3,546

dollars spent, project team size, and how many parts of the organization will be affected by the new system—the greater the risk. Very large scale systems projects have a failure rate that is 50 to 75 percent higher than that for other projects because such projects are complex and difficult to control. Risks are also higher for systems where information requirements are not clear and straightforward or the project team must master new technology.

Implementation and Change Management

Dealing with these project risks requires an understanding of the implementation process and change management. A broader definition of **implementation** refers to all the organizational activities working toward the adoption and management of an innovation, such as a new information system. Successful implementation requires a high level of user involvement in a project and management support.

If users are heavily involved in the development of a system, they have more opportunities to mold the system according to their priorities and business requirements, and more opportunities to control the outcome. They also are more likely to react positively to the completed system because they have been active participants in the change process. Incorporating user knowledge and expertise leads to better solutions.

The relationship between end users and information systems specialists has traditionally been a problem area for information systems implementation efforts because of differing backgrounds, interests, and priorities. These differences create a **user-designer communications gap**. Information systems specialists often have a highly technical orientation to problem solving, focusing on technical solutions in which hardware and software efficiency is optimized at the expense of ease of use or organizational effectiveness. End users prefer systems that are oriented toward solving business problems or facilitating organizational tasks. Often the orientations of both groups are so at odds that they appear to speak in different tongues.

These differences are illustrated in Table 11.6, which depicts the typical concerns of end users and technical specialists (information systems designers) regarding the development of a new information system. Communication problems between end users and designers are a major reason why user requirements are not properly incorporated into information systems and why users are driven out of the implementation process.

If an information systems project has the backing and commitment of management at various levels, it is more likely to receive higher priority from both users and the technical information systems staff. Management backing also ensures that a systems project receives sufficient funding and resources to be successful. Furthermore, to be enforced effectively, all the changes in work habits and procedures and any organizational realignments associated with a new system depend on management backing.

The Interactive Session on People demonstrates the importance of user involvement in designing and developing a successful solution. Dorfman Pacific, a manufacturer of

TABLE 11.6
The User-Designer Communications Gap

User Concerns	Designer Concerns
Will the system deliver the information I need for my work?	How much disk storage space will the master file consume?
How quickly can I access the data?	How many lines of program code will it take to perform this function?
How easily can I retrieve the data?	How can we cut down on CPU time when we run the system?
How much clerical support will I need to enter data into the system?	What is the most efficient way of storing the data?
How will the operation of the system fit into my daily business schedule?	What database management system should we use?

headwear and handbags, could not effectively expand its business because it was hampered by an outdated warehouse system and heavily manual processes. It decided to implement a new wireless warehouse that changed the way it worked. As you read this case, try to identify the problem this organization was facing, what alternative solutions were available to management, and how well the chosen solution worked.

Controlling Risk Factors

There are strategies you can follow to deal with project risk and increase the chances of a successful system solution. If the new system involves challenging and complex technology, you can recruit project leaders with strong technical and administrative experience. Outsourcing or using external consultants are options if your firm does not have staff with the required technical skills or expertise.

Large projects benefit from appropriate use of **formal planning and tools** for documenting and monitoring project plans. The two most commonly used methods for documenting project plans are Gantt charts and PERT charts. A Gantt chart lists project activities and their corresponding start and completion dates. The **Gantt chart** visually represents the timing and duration of different tasks in a development project as well as their human resource requirements (see Figure 11-10). It shows each task as a horizontal bar whose length is proportional to the time required to complete it.

Although Gantt charts show when project activities begin and end, they don't depict task dependencies, how one task is affected if another is behind schedule, or how tasks should be ordered. That is where **PERT charts** are useful. PERT stands for Program Evaluation and Review Technique, a methodology developed by the U.S. Navy during the 1950's to manage the Polaris submarine missile program. A PERT chart graphically depicts project tasks and their interrelationships. The PERT chart lists the specific activities that make up a project and the activities that must be completed before a specific activity can start, as illustrated in Figure 11-11 on page 394.

The PERT chart portrays a project as a network diagram consisting of numbered nodes (either circles or rectangles) representing project tasks. Each node is numbered and shows the task, its duration, the starting date, and the completion date. The direction of the arrows on the lines indicates the sequence of tasks and shows which activities must be completed before the commencement of another activity. In Figure 11-11, the tasks in nodes 2, 3, and 4 are not dependent on each other and can be undertaken simultaneously, but each is dependent on completion of the first task.

Project Management Software Commercial software tools are available to automate the creation of Gantt and PERT charts and to facilitate the project management process. Project management software typically features capabilities for defining and ordering tasks, assigning resources to tasks, establishing starting and ending dates for tasks, tracking progress, and facilitating modifications to tasks and resources. The most widely used project management tool today is Microsoft Project.

Overcoming User Resistance

You can overcome user resistance by promoting user participation (to elicit commitment as well as to improve design), by making user education and training easily available, and by providing better incentives for users who cooperate. End users can become active members of the project team, take on leadership roles, and take charge of system installation and training.

You should pay special attention to areas where users interface with the system, with sensitivity to ergonomics issues. **Ergonomics** refers to the interaction of people and machines in the work environment. It considers the design of jobs, health issues, and the end-user interface of information systems. For instance, if a system has a series of complicated online data entry screens that are extremely difficult or time-consuming to work with, users will reject the system if it increases their work load or level of job stress.

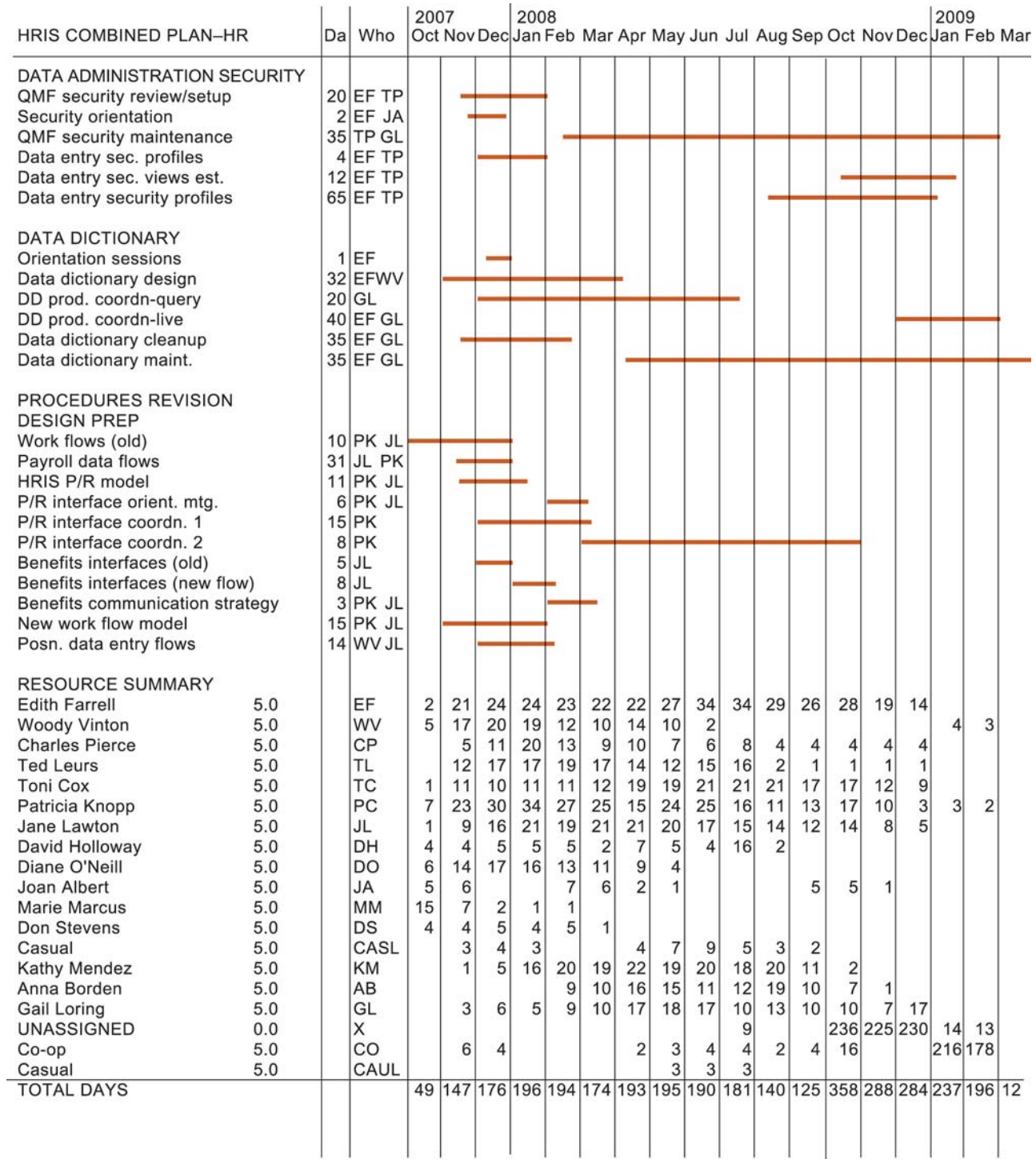


Figure 11-10
A Gantt chart
The Gantt chart in this figure shows the task, person-days, and initials of each responsible person, as well as the start and finish dates for each task. The resource summary provides a good manager with the total person-days for each month and for each person working on the project to manage the project successfully. The project described here is a data administration project.

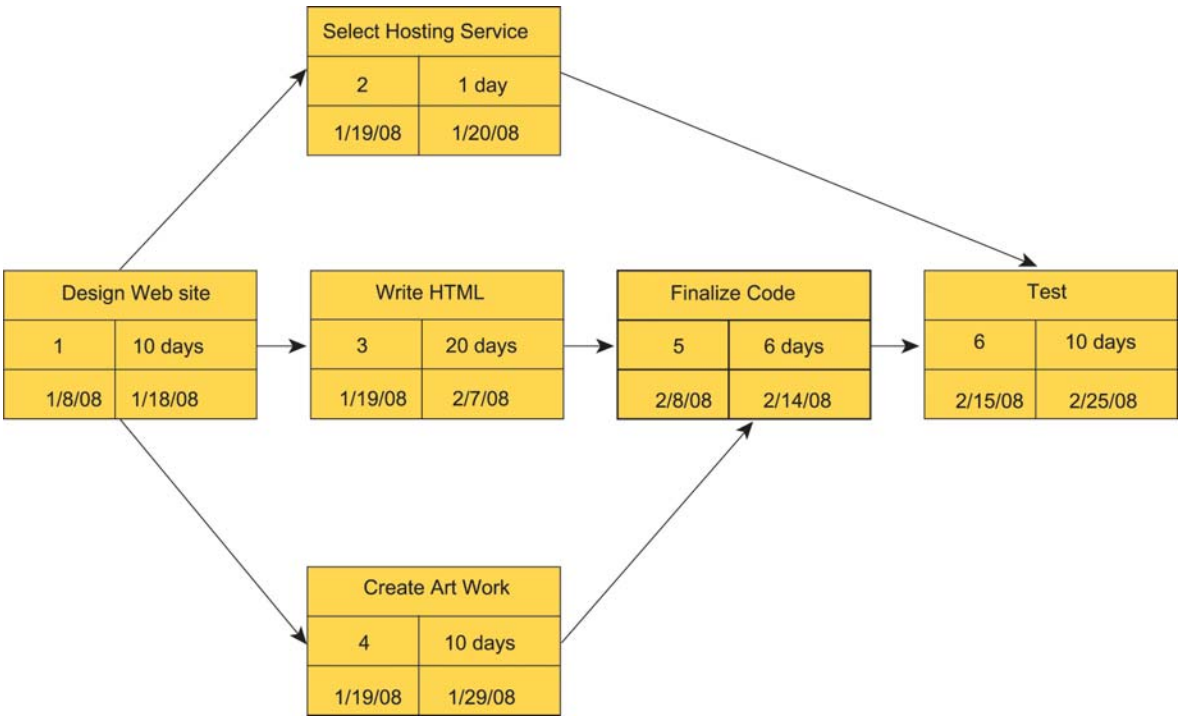


Figure 11-11
A PERT chart
This is a simplified PERT chart for creating a small Web site. It shows the ordering of project tasks and the relationship of a task with preceding and succeeding tasks.

Users will be more cooperative if organizational problems are solved prior to introducing the new system. In addition to procedural changes, transformations in job functions, organizational structure, power relationships, and behavior should be identified during systems analysis using an **organizational impact analysis**.

You will see some of these project management issues at work in the Interactive Session on Organizations, which describes the experience of Maine's Department of Human Services in implementing a new information system for processing Medicaid claims. As you read this case, try to identify the problem this organization was facing, what alternative solutions were available to management, and how well the chosen solution worked.

MANAGING PROJECTS ON A GLOBAL SCALE

As globalization proceeds, companies will be building many more new systems that are global in scale, spanning many different units in many different countries. The project management challenges for global systems are similar to those for domestic systems, but they are complicated by the international environment. User information requirements, business processes, and work cultures differ from country to country. It is difficult to convince local managers anywhere in the world to change their business processes and ways of working to align with units in other countries, especially if this might interfere with their local performance.

Involving people in change, assuring them that change is in the best interests of the company and their local units, is a key tactic for convincing users to adopt global systems and standards. Information systems projects should involve users in the design without giving up control over the project to parochial interests.

INTERACTIVE SESSION: PEOPLE Dorfman Pacific Rolls Out a New Wireless Warehouse

You may not have heard of Dorfman Pacific, but you've probably seen its hats on celebrities featured in People and *InStyle* magazines. Dorfman Pacific, based in Stockton, California, has been manufacturing and distributing headwear and handbags for over 85 years. The company's philosophy has been to keep up with fashion trends while offering quality products with strong customer service, on-time deliveries, and competitive prices.

Traditionally, Dorfman served the mom-and-pop sector of the retail market. The company's warehouse processes reflected this. Warehouse activities relied on paper-based processes and tacit knowledge of the facility and Dorfman Pacific's customers.

In the 1980s and 1990s, Dorfman Pacific started adding big-box stores like Wal-Mart and JC Penney to its roster of customers. Such stores quickly came to represent half of Dorfman's business. More significantly, the large retailers had a much greater appetite for thousands of different items and box types.

Serving retailers like Wal-Mart with a paper-based order-picking process in a 100,000-square-foot warehouse was stressful and an ineffective means of doing business. During seasonal peaks in demand, Dorfman had to hire extra workers and pay hefty overtime wages to satisfy the demand. The extra wages amounted to $250,000 every year. The company's IT systems were spread out over various functional areas and did little to support a transparent inventory.

Dorfman eventually increased its warehouse space to 275,000 square feet, but the space alone was not enough to overcome flawed business processes. Top management at Dorfman realized that major changes were necessary if the company was going to expand its operations successfully. In 2001, CEO Douglass Highsmith committed to a complete overhaul of the technology in the warehouse. He wanted to eliminate the paper systems and replace them with wireless technology.

The traditional order-fulfillment process at Dorfman began with a warehouse worker, called a picker, receiving a paper pick ticket from a supervisor. The picker then drove a forklift to the area of the warehouse where he or she expected to find the bin that stored the product on the ticket. The worker manually picked boxes off of the shelf and then brought them to a packing area to be boxed, labeled, and loaded onto a truck. The warehouse was really set up only for picking, which left the remaining order-fulfillment processes as afterthoughts.

Confusing the process were bins that were labeled manually, making them difficult to read, and boxes that sometimes held more than one product. Additionally, each picker had his or her own preferred path to performing picking duties. The inefficiencies of these practices were magnified by special orders. The company's ERP system offered little help because it did not integrate well with other systems. Mark Dulle, Dorfman's IT Services Director, recognized that picking by order wasn't going to work in an era of expansion.

Dorfman approached the change as a business project rather than an IT project. A cross-functional team consisting of an outside consultant as project manager and managers from distribution, purchasing, customer service, and sales worked on the transformation. The IT department took responsibility for choosing hardware, installing the hardware and software for the wireless warehouse, and appointing an administrator for the new warehouse management system.

Highsmith's goals were to reduce labor costs and create the most efficient way for a streamlined warehouse staff to pick products with the smallest error rate. A successful implementation required a number of steps. First, the project team sought to learn everything it could about how Dorfman's 25,000 products were received, replenished, picked, packed, and shipped. This study included measuring the dimension and weight of each product, as well as the size of every bin and storage shelf, and determining whether products were stored in the correct places.

Next, Dorfman brought in Texas Bar Code Systems to test the feasibility of a wireless system in the warehouse. The project would have been fruitless if wireless signals did not function properly amidst the warehouse's concrete walls, steel doors, and metal storage shelves. The testing also helped to determine where the best wireless access points were located. Dorfman's warehouse required an unusually high number of access points, fifteen, because the floor space expansions over the years created an irregular layout, which was dense with inventory.

Dulle led the effort to revamp Dorfman's IT infrastructure, including replacing all of the old networking cables and switches with the most advanced networking technology available. He also reconfigured the ERP system and installed a new warehouse management system from HighJump Software complete with wireless capabilities and the

ability to sort through warehouse and shipping data. To this system, which was based on a wireless LAN, Dulle added bar-coding equipment from Zebra Technologies, integration software, durable mobile computers, and additional computers mounted on forklifts.

With these components in place, paper was no longer necessary. The new ERP system and the warehouse management system used software to manage the picking, packing, and shipping processes. Pickers carrying mobile devices receive data telling them where to go, what to pick, and where to bring the merchandise using the most efficient route.

Dorfman employees had to change the way they worked. The new warehouse management system required a different warehouse floor configuration as well as new ways to pick, pack and ship products. Dorfman took the job of selling the new systems to its workers very seriously, convincing them that the wireless warehouse would improve their lives and their job performance.

Once the new warehouse system was deployed, pickers armed with wireless scanning devices could be assured that the bar-code-labeled bins to which they were directed contained only one product type each. Tracking inventory became seamless. According to Dulle, Dorfman can now handle twice the number of orders during peak seasons and labor costs are down almost 30 percent. Eliminating the need for temporary workers and overtime has saved the company $250,000 and counting.

Sources: Thomas Wailgum, "How to Take Your Warehouse Wireless," and "Wireless—Five Steps to a Successful Wireless Rollout," *CIO Magazine*, February 1, 2007; Jim Fulcher, "Rise of User-Friendly Devices Propels Strategic Use of Wireless Technology," *Manufacturing Business Technology*, February 18, 2007; Lisa M. Kempfer, "Hats Off to Wireless," *Material Handling Management*, January 2007; and "Hats-Off: Dorfman Pacific Implements Symbol Enterprise Mobility Solution for Paperless Warehouse Operations," www.symbol.com, September 13, 2006.

CASE STUDY QUESTIONS

1. Compare Dorfman Pacific's old and new order-picking processes. Diagram the processes.
2. What role did end users play in developing Dorfman's wireless warehouse system? What would have happened to the project if users hadn't been so involved? Explain your answer.
3. What types of system-building methods and tools did Dorfman use for building its wireless warehouse system?
4. How did the new system change the way Dorfman ran its business?
5. What problems did the new system solve? Was it successful?

MIS IN ACTION

Use your Web-searching capabilities to answer the following questions.

1. What are some of the components of a wireless warehouse system?
2. What companies manufacture these components?
3. What other businesses or organizations have implemented wireless wirehouses?
4. If you were implementing a wireless warehouse, what potential problems would worry you most?

One tactic is to permit each country unit in a global corporation to develop one transnational application first in its home territory, and then throughout the world. In this manner, each major country systems group is given a piece of the action in developing a transnational system, and local units feel a sense of ownership in the transnational effort. On the downside, this assumes the ability to develop high-quality systems is widely distributed, and that, a German team, for example, can successfully implement systems in France and Italy. This will not always be the case.

A second tactic is to develop new transnational centers of excellence, or a single center of excellence. These centers draw heavily from local national units, are based on multinational teams, and must report to worldwide management. Centers of excellence

INTERACTIVE SESSION: ORGANIZATIONS What Went Wrong with Maine's New Medicaid System?

The state of Maine provides medical coverage for over 260,000 of its residents through its Medicaid program, called MaineCare. Healthcare providers submit claims to MaineCare in order to be paid for the services they provide to Medicaid patients. As the 1990s drew to a close, Maine, like many other states, began planning for a complete overhaul of its Medicaid claims processing systems to comply with the Health Insurance Portability and Accountability Act of 1996 (HIPAA). HIPAA was enacted to standardize the management of patient health and records, and, most notably the protection of patient privacy. HIPAA provided a deadline of October 1, 2002 to meet its patient privacy and security standards.

The Medicaid program, as outlined by the federal government, was becoming increasingly complex as new services were added, each with codes and subcodes assigned to them. The state also wanted to offer providers access to patient eligibility and claim status data online in the hopes of reducing the volume of calls to the state Bureau of Medical Services, which ran Medicaid under the Department of Human Services (DHS).

At the time, Maine was processing over 100,000 Medicaid claims per week on a Honeywell mainframe that dated back to the 1970s. The system was not capable of supporting HIPAA requirements or the online access that the state wished to implement. The state's IT department decided that a completely new system would be more cost-effective and easier to maintain than an upgrade of the old system. DHS believed a new custom-built system would be more flexible because they could make it rule-based in order to accommodate frequent changes in Medicaid rules.

For such a large and significant project, the DHS enlisted a private contractor to work with its IT staff. CNSI, which had never before designed a Medicaid claims system, received the contract for its $15 million proposal. The deal called for CNSI to deploy the new processing system by the HIPAA compliance deadline, which was 12 months away. The system debuted on January 21, 2005, almost 27 months later, failing on many levels.

Shortly after its rollout, the new system was rejecting claims much more frequently than the old system had. Most of the rejected claims were being held up as suspended, a designation usually applied to claim forms that contained errors. The suspended file grew quickly, causing millions of dollars in claims to be held back.

Within two months, 300,000 claims were frozen. The number would eventually reach nearly 650,000. The Bureau of Medical Services could keep up neither with the number of phone calls nor the processing of the suspended claims. The 65 members of the DHS/CNSI team worked feverishly to fix glitches in the error-prone claims software, but their efforts were accompanied by a lack of regard for critical management guidelines. Meanwhile, some providers who weren't getting paid were forced to turn away Medicaid patients or even shut down their operations. Others sought bank loans to keep their practices fluid. Maine began issuing interim payments to providers that were based on their average claims, the only backup plan the state had in place.

Providers were having difficulty filling out the lengthy new claim forms correctly. The incorrect forms were among those claims that were flagged for the suspended file. CNSI recommended that all providers re-enroll so that their information would be complete according to the new system's requirements. The department chose, instead, to transfer existing information from the old system, which was considered incomplete by the new system. Only 13 people were on staff to handle customer service calls from 7,000 providers. The disaster cost the state an additional $30 million. More than a year after rolling out the new system, Maine was the only state that still had not satisfied the HIPAA requirements.

The project team had difficulty obtaining input from the Medicaid experts on staff at the Bureau of Medical Services, forcing the project team to make judgments about Medicaid rules and requirements without them. The team then had to reprogram parts of the system once the Medicaid experts became available. Although the federal body that runs Medicaid extended the HIPAA deadline to October 1, 2003, the DHS team still had an uphill fight. Sensing that they would never catch up, the team began to take shortcuts.

When the DHS launched the new claims processing system in January 2005, it made a clean break from the legacy system. There was no backup or parallel system to support the deployment because the legacy system was incompatible with the new code numbers and electronic claim forms, and a parallel system was not feasible economically or technically. The malfunctioning new system was the only one available for claims processing work. The Bureau of Medical Services could not remedy the botched claims as fast they were coming in, creating a crisis in Maine's healthcare system.

Later in 2005, Maine hired XWave, a project management consultant specializing in integration for

over $860,000 to right the ship. The state added an Oracle database specialist, who filled the role of operations manager and worked to improve communications and prioritize troubleshooting.

Dr. Laureen Biczak was appointed the organization's Medicaid expert. Beginning in January 2006, questions from providers were filtered to Medicaid specialists working under Biczak if they were business-process issues, or to the IT department if they were hardware or software issues. With this filtering system in effect, Maine was able to reduce the claim suspension rate to about 15%.

By the fall of 2006 the project cost between $30 million and $40 million over the initial bid of $15 million. In the spring of 2007, Maine decided to cut its losses. The state reached an agreement to have CNSI continue maintaining the system and add functionality until a new vendor was chosen by June 2008.

CNSI lost around $10 million on the project. Its president, B. Chatterjee, believed that CNSI and the state bore responsibility for half of the problems, and providers who made mistakes on their claim forms could be blamed for the other half. He also insisted that Maine is better positioned to go forward than other states that updated their existing systems rather than building state-of-the-art new systems. CNSI received additional state Medicaid contracts in the wake of its work with Maine.

Sources: Adam Wilson, "Glitches Delay State Project," *The Olympian*, July 13, 2007; Patty Enrado, "Maine Medicaid Cuts Losses on $50M of IT," www.healthcareitnews.com, April 1, 2007 and "CMS Investigating New Maine Medicaid System," www.healthcareitnews.com, October 1, 2006; "Maine Struggles with Medicaid Billing System," www.fiercehealthit.com, November 26, 2006; and Allan Holmes, "Maine's Medicaid Mistakes," *CIO Magazine*, April 15, 2006.

CASE STUDY QUESTIONS

1. How important are information systems for Maine's Department of Human Services? Analyze the impact of its faulty Medicaid claims processing system.
2. Evaluate the risks of the Medicaid claims processing system project and key risk factors.
3. Classify and describe the problems the Maine Department of Human Services faced in implementing its new Medicaid claims processing system. What people, organization, and technology factors caused these problems?
4. Describe the steps you would have taken to control the risk in the Maine Medicaid project. If you were in charge of managing this project, what else would you have done differently to increase chances for success?

MIS IN ACTION

Visit the Office of MaineCare Services on the Web at www.maine.gov/bms/ and then answer the following questions:

1. What services are available through MaineCare online?
2. What information is available regarding MaineCare and its compliance with HIPAA standards?
3. What information is available for providers who have had or are still having problems using MaineCare's claims processing system?
4. How easy is it to obtain this information?

perform the business and systems analysis and accomplish all design and testing. Implementation, however, and pilot testing are rolled out to other parts of the globe. Recruiting a wide range of local groups to transnational centers of excellence helps send the message that all significant groups are involved in the design and will have an influence.

11.5 Hands-On MIS

The projects in this section give you hands-on experience designing and building an employee training and skills tracking system for a real world company, designing and building a customer system for auto sales, and analyzing Web site information requirements.

ACHIEVING OPERATIONAL EXCELLENCE: DESIGNING AN EMPLOYEE TRAINING AND SKILLS TRACKING SYSTEM AND DATABASE

Software skills: Database design, querying, and reporting
Business skills: Employee training and skills tracking

This project requires you to perform a systems analysis and then design a system solution using database software.

Dirt Bikes promotes itself as a "learning company." It pays for employees to take training courses or college courses to help them advance in their careers. As employees move on, their job positions become vacant and Dirt Bikes must quickly fill them to maintain its pace of production. Dirt Bikes's human resources staff would like to find a way to quickly identify qualified employees who have the training to fill vacant positions. Once the company knows who these employees are, it has a better chance of filling open positions internally rather than paying to recruit outsiders. Dirt Bikes would like to track each employee's years of education and the title and date completed of training classes that each employee has attended.

Dirt Bikes currently cannot identify such employees. Its existing employee database is limited to basic human resources data, such as employee name, identification number, birth date, address, telephone number, marital status, job position, and salary. A portion of this database is illustrated here. You can find some sample records from this database on the Laudon Web site. Dirt Bikes's human resources staff keeps skills and training data in paper folders.

tblEmployee : Table

Emp_SS#	Last_Name	First_Name	Street	City	State	Zip	Telephone	Birth_Date	Marital
012-34-5678	Munoz	Luis	11 Ram Ridge	Carbondale	CO	81623	970-945-4338	8/29/1971	M
123-45-6789	Lattimore	Joseph	302 Garden St.	Carbondale	CO	81623	970-945-7002	7/3/1972	S
234-56-7890	Artis	Patricia	45 William St.	Basalt	CO	81621	970-945-0087	4/11/1973	S
345-67-8901	Renaldo	Carlos	793 Ridge Rd.	Basalt	CO	81621	970-945-1883	5/8/1969	M
456-78-9012	Norwick	Robert	10 Webb St.	Carbondale	CO	81623	970-945-5654	9/1/1970	M
567-89-0123	O'Donnell	James	56 Mountainview	Carbondale	CO	81623	970-945-3021	2/9/1969	M
678-90-1234	Morrissey	Richard	93 Pond Drive	Basalt	CO	81621	970-945-2838	1/30/1960	D
789-01-2345	Kamp	Timothy	39 Brook Dr.	Basalt	CO	81621	970-945-5944	6/27/1964	M
890-12-3456	Franz	George	8 Powder Tr.	Carbondale	CO	81623	970-945-9304	9/21/1964	S
901-23-4567	Collins	Dawn	9 Young Ave.	Glenwood Sprin	CO	81602	970-945-2648	11/3/1974	S
987-65-4321	Stokes	Brian	5 Saddle Tr.	Carbondale	CO	81623	970-945-8943	5/14/1963	D

Record: 1 of 11

Prepare a systems analysis report describing Dirt Bikes's problem and a system solution that can be implemented using PC database software. Then use the database software to develop a simple system solution. Your report should include the following:

- Description of the problem and its organizational and business impact.
- Proposed solution and solution objectives.
- Information requirements to be addressed by the solution.
- People, organization, and technology issues to be addressed by the solution, including changes in business processes.

On the basis of the requirements you have identified, design the solution using database software and populate it with at least 10 records per table. Consider whether you can use or modify the existing employee database in your design. Print out the design for each table in your new application. Use the system you have created to create queries and reports that would be of most interest to management, such as which employees have college education or which employees have training in project management or advanced computer-aided design [CAD] tools.

If possible, use electronic presentation software to summarize your findings for management.

IMPROVING DECISION MAKING: USING DATABASE SOFTWARE TO DESIGN A CUSTOMER SYSTEM FOR AUTO SALES

Software skills: Database design, querying, reporting, and forms
Business skills: Sales lead and customer analysis

This project requires you to perform a systems analysis and then design a system solution using database software.

Ace Auto Dealers specializes in selling new vehicles from Subaru. The company advertises in local newspapers and also is listed as an authorized dealer on the Subaru Web site and other major Web sites for auto buyers. The company benefits from a good local word-of-mouth reputation and name recognition and is a leading source of information for Subaru vehicles in the Portland, Oregon, area.

When a prospective customer enters the showroom, he or she is greeted by an Ace sales representative. The sales representative manually fills out a form with information such as the prospective customer's name, address, telephone number, date of visit, and model and make of the vehicle in which the customer is interested. The representative also asks where the prospect heard about Ace—whether it was from a newspaper ad, the Web, or word of mouth—and this information is noted on the form also. If the customer decides to purchase an auto, the dealer fills out a bill of sale form.

Ace does not believe it has enough information about its customers. It cannot easily determine which prospects have made auto purchases, nor can it identify which customer touch points have produced the greatest number of sales leads or actual sales so it can focus advertising and marketing more on the channels that generate the most revenue. Are purchasers discovering Ace from newspaper ads, from word of mouth, or from the Web?

Prepare a systems analysis report detailing Ace's problem and a system solution that can be implemented using PC database management software. Then use database software to develop a simple system solution. Your systems analysis report should include the following:

- Description of the problem and its organizational and business impact.
- Proposed solution, solution objectives, and solution feasibility.
- Costs and benefits of the solution you have selected. The company has a PC with Internet access and the full suite of Microsoft Office desktop productivity tools.
- Information requirements to be addressed by the solution.
- People, organization, and technology issues to be addressed by the solution, including changes in business processes.

On the basis of the requirements you have identified, design the database and populate it with at least 10 records per table. Consider whether you can use or modify the existing customer database in your design. Print out the database design. Then use the system you have created to generate queries and reports that would be of most interest to management. Create several prototype data input forms for the system and review them with your instructor. Then revise the prototypes.

ACHIEVING OPERATIONAL EXCELLENCE: ANALYZING WEB SITE DESIGN AND INFORMATION REQUIREMENTS

Software skills: Web browser software
Business skills: Information requirements analysis, Web site design

Visit the Web site of your choice and explore it thoroughly. Prepare a report analyzing the various functions provided by that Web site and its information requirements. Your report should answer these questions: What functions does the Web site perform? What data does it use? What are its inputs, outputs, and processing? What are some of its other design specifications? Does the Web site link to any internal systems or systems of other organizations? What value does this Web site provide the firm?

LEARNING TRACKS

The following Learning Tracks provide content relevant to topics covered in this chapter:

1. Capital Budgeting Methods for Information System Investments
2. Enterprise Analysis (Business Systems Planning) and Critical Success Factors (CSFs)
3. Unified Modeling Language (UML)

Review Summary

1 What are the core problem-solving steps for developing new information systems? The core problem-solving steps for developing new information systems are: (1) define and understand the problem, (2) develop alternative solutions, (3) evaluate and choose the solution, and (4) implement the solution. The third step includes an assessment of the technical, financial, and organizational feasibility of each alternative. The fourth step entails finalizing design specifications, acquiring hardware and software, testing, providing training and documentation, conversion, and evaluating the system solution once it is in production.

2 What are the alternative methods for building information systems? The systems lifecycle requires that information systems be developed in formal stages. The stages must proceed sequentially and have defined outputs; each requires formal approval before the next stage can commence. The system lifecycle is rigid and costly but nevertheless useful for large projects.

Prototyping consists of building an experimental system rapidly and inexpensively for end users to interact with and evaluate. The prototype is refined and enhanced until users are satisfied that it includes all of their requirements and can be used as a template to create the final system. End-user-developed systems can be created rapidly and informally using fourth-generation software tools. End-user development can improve requirements determination and reduce application backlog.

Application software packages eliminate the need for writing software programs when developing an information system. Application software packages are helpful if a firm does not have the internal information systems staff or financial resources to custom develop a system.

Outsourcing consists of using an external vendor to build (or operate) a firm's information systems. If it is properly managed, outsourcing can save application development costs or enable firms to develop applications without an internal information systems staff.

Rapid application design, joint application design (JAD), and reusable software components (including Web services) can be used to speed up the systems development process.

3 What are the principal methodologies for modeling and designing systems? The two principal methodologies for modeling and designing information systems are structured methodologies and object-oriented development. Structured methodologies focus on modeling processes and data separately. The data flow diagram is the principal tool for structured analysis and the structure chart is the principal tool for representing structured

software design. Object-oriented development models a system as a collection of objects that combine processes and data.

4 How should information systems projects be selected and evaluated? To determine whether an information system project is a good investment, one must calculate its costs and benefits. Tangible benefits are quantifiable, and intangible benefits cannot be immediately quantified but may provide quantifiable benefits in the future. Benefits that exceed costs should then be analyzed using capital budgeting methods to make sure they represent a good return on the firm's invested capital.

Organizations should develop information systems plans that describe how information technology supports the company's overall business plan and strategy. Portfolio analysis and scoring models can be used to evaluate alternative information systems projects.

5 How should information systems projects be managed? Information systems projects and the entire implementation process should be managed as planned organizational change using an organizational impact analysis. Management support and control of the implementation process are essential, as are mechanisms for dealing with the level of risk in each new systems project. Project risks are influenced by project size, project structure, and the level of technical expertise of the information systems staff and project team. Formal planning and control tools (including Gantt and PERT charts) track the resource allocations and specific project activities. Users can be encouraged to take active roles in systems development and become involved in installation and training. Global information systems projects should involve local units in the creation of the design without giving up control of the project to parochial interests.

Key Terms

Acceptance testing, 376
Component-based development, 385
Computer-aided software engineering (CASE), 385
Conversion, 376
Customization, 380
Data flow diagram (DFD), 382
Direct cutover, 376
Documentation, 376
End-user development, 379
Ergonomics, 392
Feasibility study, 373
Formal planning and tools, 392
Gantt chart, 392
Implementation, 391
Information requirements, 372
Information systems plan, 388
Intangible benefits, 387
Joint application design (JAD), 382
Maintenance, 377
Object-oriented development, 384
Organizational impact analysis, 394
Parallel strategy, 376
PERT charts, 392
Phased approach, 376
Portfolio analysis, 388
Process specifications, 383
Production, 377
Project, 386
Project management, 386
Prototyping, 378
Rapid application development (RAD), 382
Request for Proposal (RFP), 380
Scope, 386
Scoring model, 390
Structure chart, 383
Structured, 382
System testing, 376
Systems analysis, 372
Systems design, 374
Systems development lifecycle (SDLC), 377
Tangible benefits, 387
Test plan, 376
Testing, 376
Unit testing, 376
User-designer communications gap, 391

Review Questions

1. What are the core problem-solving steps for developing new information systems?
- List and describe the problem-solving steps for building a new system.
- Define information requirements and explain why they are important for developing a system solution.
- List the various types of design specifications required for a new information system.
- Explain why the testing stage of systems development is so important. Name and describe the three stages of testing for an information system.
- Describe the roles of documentation, conversion, production, and maintenance in systems development.

2. What are the alternative methods for building information systems?
- Define the traditional systems lifecycle and describe its advantages and disadvantages for systems building.
- Define information system prototyping and describe its benefits and limitations. List and describe the steps in the prototyping process.
- Define end-user development and explain its advantages and disadvantages.
- Describe the advantages and disadvantages of developing information systems based on application software packages.
- Define outsourcing. Describe the circumstances in which it should be used for building information systems. List and describe the hidden costs of offshore software outsourcing.
- Explain how businesses can rapidly develop e-business applications.

3. What are the principal methodologies for modeling and designing systems?
- Compare object-oriented and traditional structured approaches for modeling and designing systems.

4. How should information systems projects be selected and evaluated?
- Explain the difference between tangible and intangible benefits.
- List six tangible benefits and six intangible benefits.
- List and describe the major components of an information systems plan.
- Describe how portfolio analysis and scoring models can be used to establish the worth of systems.

5. How should information systems projects be managed?
- Explain the importance of implementation for managing the organizational change surrounding a new information system.
- Define the user-designer communications gap and explain the kinds of implementation problems it creates.
- List and describe the factors that influence project risk and describe strategies for minimizing project risks.
- Describe tactics for managing global projects.

Discussion Questions

1. Discuss the role of business end users and information system professionals in developing a system solution. How do both roles differ when the solution is developed using prototyping or end-user development?

2. It has been said that systems fail when systems builders ignore "people" problems. Why might this be so?

Video Case

You will find a video case illustrating some of the concepts in this chapter on the Laudon Web site along with questions to help you analyze the case.

Teamwork

Analyzing Web Site Requirements

With three or four of your classmates, visit the Web site of iTunes, MP3.com, the Internet Movie Database, or a company described in this text that uses the Web. Review the Web site for the company you select. Use what you have learned from the Web site and this chapter to prepare a report describing the functions of that Web site and some of its design specifications. If possible, use electronic presentation software to present your findings to the class.

BUSINESS PROBLEM-SOLVING CASE

Citizens National Bank Searches for a System Solution

Citizens National Bank of Texas is a private, full-service bank with headquarters in Waxahachie, Texas, and 200 employees that has operated independently since 1868. Citizens National serves businesses and consumers in Ellis County and other nearby counties, primarily in communities with populations of 25,000 or less. The bank counts total assets of $400 million and is growing annually at a rate of 12 percent. Since 1999, the number of branches has increased from four to fifteen, with locations in ten cities. Citizens National would like to increase its market share to at least 50 percent in eight counties south of the Dallas-Fort Worth area

A major part of Citizen National's strategy for continuing growth was to implement customer relationship management (CRM) software. The CRM strategy targeted the bank's two main contact points with customers: the bank's call center and its sales force. The call center receives around 4,000 calls per day, which are handled by between 10 and 20 customer service representatives. The sales force consists of 16 representatives, who are known by the title of relationship bankers. It is the relationship bankers that drive business for Citizens National. Their contacts with customers generate loan sales and deposits that make money for the bank.

In 2001 CEO Mark Singleton oversaw the adoption of a CRM package from Siebel Systems (Siebel is now owned by Oracle). The main goal for the implementation was to increase sales by raising the number of contacts relationship bankers were making and improving the tracking of these activities so that the bank could learn more from them. The CRM package promised additional benefits. The bank would be able to approve credit and loan applications more quickly. It would also finally have a method for storing the interactions between relationship bankers and customers electronically.

Electronic records were key for two reasons. Under the old paper system, a salesperson that left Citizens National could take records of customer interaction with him or her, leaving the bank with no information to maintain the relationship. The paper system also created too much information for Singleton and his branch managers to process effectively.

For Singleton, the decision to move to a CRM system was not a slam dunk. While he recognized the great value that automated systems provided to businesses, he placed even greater value on the person-to-person interactions between his relationship bankers and their customers. He feared that an overreaching CRM system might interfere with those interactions and diminish the relationship bankers' rapport with customers. The track record for the old fashioned way was impressive. For retail customers, the bank's cross-sale ratio was between 2 and 2.5, meaning that the average customer used at least two of the bank's products. The top commercial and personal customers were using between 6 and 7 products.

With this strength in mind, Singleton insisted that any CRM implementation at Citizens National be able to fortify the relationship bankers' knowledge of their customers and potential customers, including their previous interactions with the bank. The Siebel package, which had a price tag of $150,000, was supposed to fulfill this goal. The bank contracted with a local consulting firm, The Small Business Solution, to help install the package. The union of old-fashioned business sensibility with powerful enterprise software was a mismatch almost immediately. The approach of Citizens National toward nearly all business functions, from tracking customer leads to generating reports about them, was very basic. The Siebel software was simply too rich in features. The bank spent an inordinate amount of time switching off features that hindered productivity.

For example, Siebel had a complex module for handling customer support cases. It included capabilities for detailed managing of complaints from initial call through subsequent calls to options for resolving the issue. Customer complaints at Citizens National rarely went that far. A representative at the call center handled them immediately. In cases where a second interaction was necessary, the representative simply sent an e-mail to the employee responsible for carrying out the action.

Jim Davis, an expert on CRM from Deloitte Consulting, characterized the situation at Citizens National by saying, "The problem with Siebel is that it has everything." At Citizens National, the sheer size of the package was not the only issue. Employees found the software to be too complicated. They were surprised to learn, for example, that the system did not automatically generate potential business opportunities for customers on their records. They had to assign the potential transaction to the customer. Furthermore, bankers were not able to view multiple relationships between a customer and the bank on the same screen. The extra navigation was confusing and inefficient. Not surprisingly, the relationship bankers resisted the new system. It didn't make sense for them to change their tried-and-true methods simply because new software required change.

According to Davis, the disconnect between the relationship bankers and the new system was at the crux of the implementation's failure. The relationship bankers were the key employees; the system was intended to be of value to them, and, in turn, provide value to the bank. However, they found no incentive in the Siebel environment because their compensation was based on sales, and sales had become harder to make.

Citizens National also experienced compatibility issues between the database formats in Siebel and those used by the bank's core banking application, developed by Kirchman. The Kirchman software combined customer first and last names into one field, whereas Siebel had separate fields for first and last names. As a result, the two systems had difficulty exchanging information properly. The bank was forced to spend a significant amount of time fixing such compatibility issues, which negatively impacted its ability to serve customers.

Citizens National spent three years trying to make its Siebel CRM implementation work. In 2004, having derived no quantifiable benefit from the effort, the bank finally decided to cut its losses. In addition to the initial $150,000 purchase, it had spent $350,000 on solving integration issues. Singleton referred to the process as "a $500,000 education."

David Furney, president and CEO of The Small Business Solution, began searching for another CRM solution for Citizens National. Furney happened upon a hosted online database system from Intuit called QuickBase, which was used by small businesses and consumers, including the Patriots' Trail Girl Scout Council discussed in the chapter-opening case. Intuit was best known for its financial management applications such as Quicken and QuickBooks. The company did not have a defined reputation in the CRM market.

QuickBase included modules for databases, spreadsheets, and sales management, all of which could be easily manipulated for the bank's business functions. QuickBase was designed for organizing, tracking, and sharing information among team members in the workplace while encouraging progress by notifying workers via automated e-mails of updated files, new task assignments, and approaching deadlines. Intuit offered in the product ready-made applications for general purposes, such as project management, sales management, and marketing management, as well as more specific purposes, such as healthcare, IT, legal, and real estate.

Furney referred to QuickBase as "the ultimate in rapid application development" and "a kind of do-it-yourself application." Citizens National people, including Singleton, would be able to customize the package themselves rather than having to solicit help from the manufacturer or an IT specialist. To make changes in the Siebel system, the bank had to request help from Siebel. Because QuickBase was not programmed as a specific business application, businesses could modify its database structure to meet specific business functions. The Citizens National staff were able to make changes to QuickBase themselves, so the cost of ownership and maintenance fees were much lower.

QuickBase offered Citizens National flexibility that it did not have previously. Because the system was Web-based, the relationship bankers were able to use it anywhere that they had access to a browser. In addition to the cost of ownership savings, Citizens National saved

a small fortune with QuickBase, which started with a one-time fee of $249 for the first 10 users and increased by $3 per month for each additional user (the cost is now $249 per month plus $15 per month for every five additional users).

Singleton still had to deal with the fact that some of relationship bankers would have preferred to push technology aside altogether. To give the system its best chance of success, Singleton permitted these bankers to dictate their activity to administrative assistants. The assistants then entered the information into QuickBase for the bankers. Davis observed that this may not have been the best practice, but it was fairly common and, more importantly, very worthwhile if the bankers were devoting their time to making money for the bank rather than wasting it wrestling with technology. Citizens National made the transition easier additionally by implementing QuickBase using a phased rollout, starting with the call center.

Also central to the success of QuickBase at Citizens National was Furney's ability to integrate the system with the Kirchman core banking application. Furney configured QuickBase to upload new account information to the core system every night via an XML interface. Relationship bankers and management received daily updated access to all interactions and transactions, enabling them to track business in a way that was never possible previously. For the first time, Citizens National was able to completely track sales opportunities and, as Singleton said, "where we lost business, so we know where we need to make those extra 10 or 15 sales calls."

Sources: Doug Bartholomew, "A Banker's $500,000 Lesson in CRM," *Baseline Magazine*, February 26, 2007; Mark Singleton, as told to Colin Beasty, "Secret of My Success: Getting More for the Money," www.destinationcrm.com, March 1, 2006; "QuickBase: A Better Way to Work," www.quickbase.com, accessed March 30, 2007; "QuickBase: How It Works," www.quickbase.com, accessed March 30, 2007; and "Siebel Customer Relationship Management Applications," www.oracle.com, March 30, 2007.

Case Study Questions

1. What was the initial problem that Mark Singleton was trying to solve at Citizens National? How well did he apply the four steps of problem solving?
2. What was the business case for implementing a new system? What were some of the tangible benefits? What were some of the intangible benefits?
3. Why didn't the implementation of the Siebel CRM solution work out for Citizens National? What were the biggest factors? How would you classify these factors in terms of organization, technology, and people issues?
4. Was QuickBase a better solution for Citizens National? If so, why? What factors suggest that the bank ended up with the right approach and the right choice of product?
5. Based on this case study, what kind of organization do you think would benefit from using the Siebel CRM package? Give an example of such an organization and justify your choice. You may use the Web to research your answer, including Oracle's Web site.
6. Could Citizens National have made a better choice of software for its CRM system the first time around? Explain your answer.