

# CHAPTER SIX OUTLINE

## STUDENT LEARNING OUTCOMES

1. List the seven steps in the systems development life cycle (SDLC) and associated activities for each step.
2. Describe the four systems development methodologies.
3. Define the role of outsourcing.
4. List and describe the three different forms of outsourcing.
5. Describe business process outsourcing (BPO).
6. Describe prototyping and profile an example of a prototype.
7. Describe the advantages of prototyping.

## INDUSTRY PERSPECTIVE

- 284 **The Life Cycle of Boise Office Solutions**
- 287 **Making Your Software Work**
- 298 **Do Your Homework!**
- 302 **St. Jude Children's Research Hospital Cures**

## FEATURES

- 283 **On Your Own**  
Analyzing Business Requirements
- 285 **Team Work**  
Writing Test Conditions
- 295 **Global Perspective**  
Outsourcing Help Desks Offshore
- 296 **Team Work**  
How Many Are There?
- 299 **Global Perspective**  
lastminute.com
- 304 **On Your Own**  
Build vs. Buy
- 311 **Electronic Commerce**  
Finding Freeware and Shareware on the Internet

## WEB SUPPORT

[www.mhhe.com/haag](http://www.mhhe.com/haag)

- Using your computer for more than work
- Protecting your computer investment
- Searching for freeware and shareware

# CHAPTER SIX

## Systems Development Phases, Tools, and Techniques

### OPENING CASE STUDY MERCEDES-BENZ ONLINE BUILT-TO-ORDER TRUCKS

“Which driver’s cabs and radios are available for a 24-ton rig?” In the past, such a question would have cost Mercedes-Benz dealers a lot of time leafing through big manuals. Today, the Mercedes-Benz Web site offers orientation in the multitude of equipment variations with the Mercedes-Benz customer advisory system, developed by CAS Software AG.

The online program, called Truck Online Configurator (TOC), required much more development time than was required to simply display the finished online system internally for the automobile and transporter divisions. The TOC was given a completely new user guide for Web site visitors that orients itself on the different demands of the site visitors in three ways: through technical features, trade solution, and transport task. Interested customers are able to compile their own vehicle preference by entering the required technical details, like the type of drive, wheelbase, or engine performance needed. Or the interested customer can select by trade solution (where he or she simply tells the TOC that a truck is required for the transportation of frozen foods, for example) and a list of suitable models is offered. These can then be viewed in video sequences or as a 360° view. The third alternative for visitors to the site is to let the interested cus-

tomers find the desired commercial vehicle by entering the transport task, that is, by feeding the TOC with the necessary approximate power per mile per year or the distance profile. On every single page, numerous information windows or buttons are opened containing explanatory text and graphics.

CAS had only five months in which to produce this complex and extensive application, from the definition of the specialist requirements, to the technical specifications, right up to the development, testing, and installation of the software. This tight deadline required a risk-driven project management, which meant that individual phases of the project would overlap. While a few developers, together with the project managers of Mercedes-Benz, were clarifying the technical requirements of the individual functions, other parts of the TOC were already being implemented, providing feedback for the developers. This approach to system development is called rapid application development (RAD).

The CAS solution was virtually tailor-made for this project. The finished programs run in an extremely fast and stable manner. These were the central, technical requirements of Mercedes-Benz for the TOC: The company wanted an application which boasted a maximum availability but with minimum response times.

Thanks to using the rapid application development (RAD) process, CAS was able to create a very fast high-quality application.<sup>1</sup>

## Introduction

Have you ever wondered why businesses develop information systems (IT systems) in the first place, or how a business knows when it's time to replace an old information system? Almost any position you take in a company today will require you to work with information systems because they are by now one of the most important elements of any business. Information systems supply the support structure for meeting companies' strategies and goals. Typically, new systems are created because employees request the systems to help them perform their work. For example, a marketing manager might request a system to provide product information and track customer sales information (CRM activities) or a human resource manager might request a new system to track employee vacation and sick days.

Billions of dollars are spent each year on the acquisition, design, development, implementation, and maintenance of information systems. The ongoing need for safe, secure, and reliable system solutions is the consequence of companies' increasing dependence on information technology to provide services and develop products, administer daily activities, and perform short- and long-term management functions.

Systems developers must ensure all the business's needs and requirements are met when developing information systems, establish uniform privacy and security practices for system users, and develop acceptable implementation strategies for the new systems. This chapter focuses on the many factors that must come together to develop a successful information system.

## The Systems Development Life Cycle

The *systems development life cycle (SDLC)* is a structured step-by-step approach for developing information systems. There are literally hundreds of different activities associated with each phase in the SDLC. Typical activities include determining budgets, gathering business requirements, designing models, and writing detailed user documentation. The activities you perform during each systems development project will vary depending on the type of system you're building and the tools you use to build it. Since we can't possibly cover them all in this brief introduction, we have chosen a few of the more important SDLC activities that you might perform on a systems development project. Figure 6.1 displays the seven phases in the SDLC along with the associated activities discussed in this chapter.

As we introduce you to the SDLC in this section, we'll focus on how the overall process works (see Figure 6.2), key activities within each phase, roles you may play in a project, and opportunities you can capitalize on to ensure that your systems development effort is a success.

### PHASE 1: PLANNING

In the *planning phase* of the systems development life cycle you create a solid plan for developing your information system. The following are the three primary activities you'll perform during the planning phase:

1. *Define the system to be developed:* You must identify and select the system for development or determine which system is required to support the strategic goals of your organization. Organizations typically track all the proposed systems

SDLC Phase	Activities
1. Planning	<ul style="list-style-type: none"> <li>Define the system to be developed</li> <li>Set the project scope</li> <li>Develop the project plan including tasks, resources, and timeframes</li> </ul>
2. Analysis	<ul style="list-style-type: none"> <li>Gather the business requirements for the system</li> </ul>
3. Design	<ul style="list-style-type: none"> <li>Design the technical architecture required to support the system</li> <li>Design system models</li> </ul>
4. Development	<ul style="list-style-type: none"> <li>Build the technical architecture</li> <li>Build the database and programs</li> </ul>
5. Testing	<ul style="list-style-type: none"> <li>Write the test conditions</li> <li>Perform the testing of the system</li> </ul>
6. Implementation	<ul style="list-style-type: none"> <li>Write detailed user documentation</li> <li>Provide training for the system users</li> </ul>
7. Maintenance	<ul style="list-style-type: none"> <li>Build a help desk to support the system users</li> <li>Provide an environment to support system changes</li> </ul>

Figure 6.1

The Systems Development Life Cycle (SDLC) and Associated Activities

and prioritize them based on business impact or critical success factors. A **critical success factor (CSF)** is a factor critical to your organization's success. This allows the business to strategically decide which systems to build.

2. *Set the project scope:* You must define the project's scope and create a project scope document for your systems development effort. The **project scope** clearly defines the high-level system requirements. Scope is often referred to as the 10,000-foot view of the system or the most basic definition of the system. Setting the project scope is important for many reasons; most important it helps you avoid scope creep and feature creep. **Scope creep** occurs when the scope of the project increases beyond its original intentions. **Feature creep** occurs when developers add extra features that were not part of the initial requirements. A **project scope document** is a written definition of the project scope and is usually no longer than a paragraph.
3. *Develop the project plan:* You must develop a detailed project plan for your entire systems development effort. The **project plan** defines the what, when, and who questions of systems development including all activities to be performed, the individuals, or resources, who will perform the activities, and the time required to complete each activity. The detailed project plan is the guiding force behind ensuring the on-time delivery of a complete and successful information system.

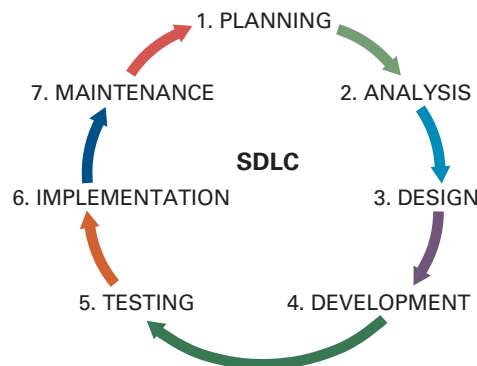


Figure 6.2

The Systems Development Life Cycle (SDLC)

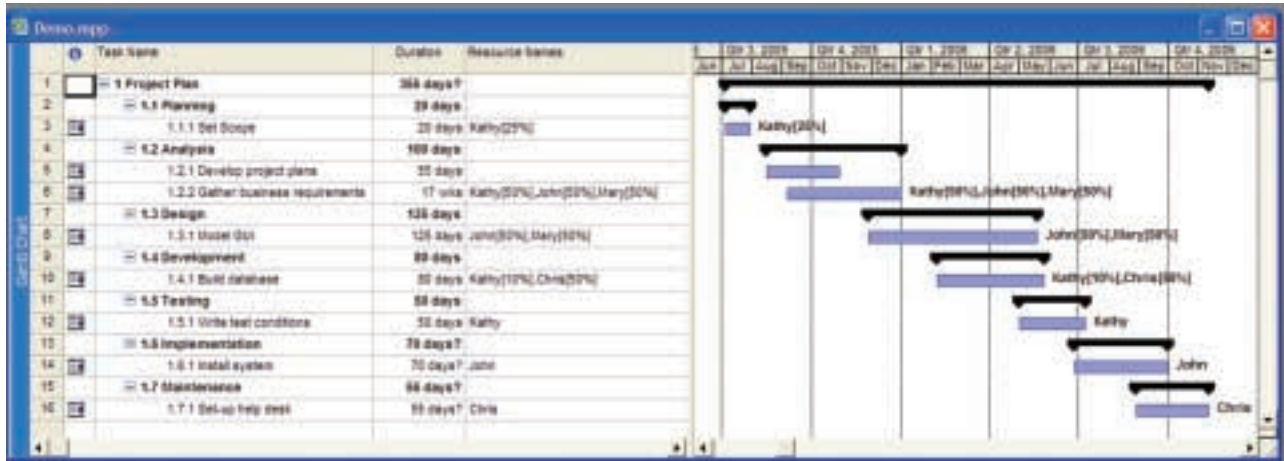


Figure 6.3

A Sample Project Plan

Figure 6.3 displays a sample project plan. A **project manager** is an individual who is an expert in project planning and management, defines and develops the project plan, and tracks the plan to ensure all key project milestones are completed on time. **Project milestones** represent key dates by which you need a certain group of activities performed. For example, completing the planning phase might be a project milestone. If all your project milestones are being completed on time, this is a good indication that your project is on schedule and will be successful.

To learn more about project planning and management, visit the Web site that supports this text at [www.mhhe.com/haag](http://www.mhhe.com/haag).

## PHASE 2: ANALYSIS

Once your organization has decided which system to develop, you can move into the analysis phase. The **analysis phase** of the systems development life cycle involves end-users and IT specialists working together to gather, understand, and document the business requirements for the proposed system. The following are the two primary activities you'll perform during the analysis phase:

1. **Gathering the business requirements:** **Business requirements** are the detailed set of employee requests that the system must meet to be successful. The business requirements drive the entire system. A sample business requirement might state, "The CRM system must track all customer sales by product, region, and sales representative." The business requirement states what the system must do from the business perspective. Gathering business requirements is similar to performing an investigation. You must talk to everyone who has a claim in using the new system to find out what is required. An extremely useful way to gather business requirements is to perform a joint application development session. During a **joint application development (JAD)** session users and IT specialists meet, sometimes for several days, to define and review the business requirements for the system.
2. **Prioritize the requirements:** Once you define all the business requirements, you prioritize them in order of business importance and place them in a formal comprehensive document, the **requirements definition document**. The users receive the requirements definition document for their sign-off. **Sign-off** is the

## ON YOUR OWN

### ANALYZING BUSINESS REQUIREMENTS

You have been hired to build an employee tracking system for a new coffee shop. Review the following business requirements and highlight any potential issues.

- All employees must have a unique employee ID.
- The system must track employee hours worked based on employee last name.
- Employees must be scheduled to work a minimum of 8 hours per day.
- Employee payroll is calculated by multiplying the employee's hours worked by \$7.25.
- Managers must be scheduled to work morning shifts.
- Employees cannot be scheduled to work over 8 hours per day.
- Servers cannot be scheduled to work morning, afternoon, or evening shifts.
- The system must allow managers to change and delete employees from the system.

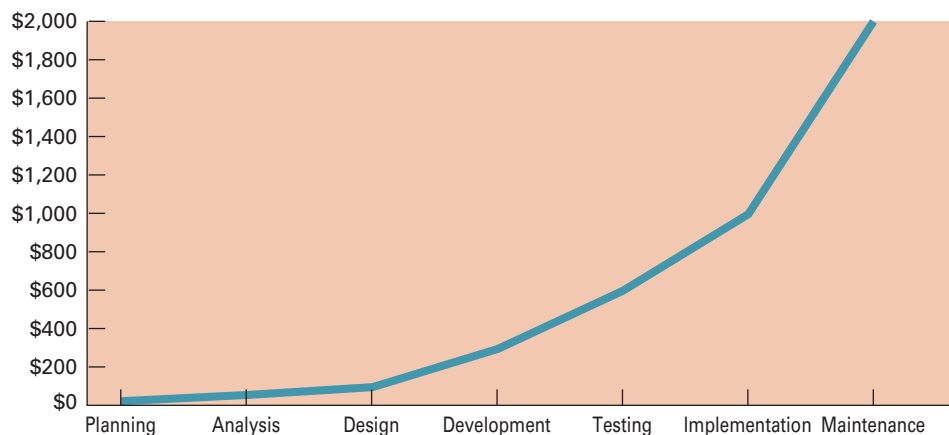


Figure 6.4

The Cost of Finding Errors

users' actual signatures indicating they approve all the business requirements.

Typically, one of the first major milestones in the project plan is the users' sign-off on business requirements.

One of the key things to think about when you are reviewing business requirements is the cost to the company of fixing errors if the business requirements are unclear or inaccurate. An error found during the analysis phase is relatively inexpensive to fix; all you typically have to do is change a Word document. An error found during later phases, however, is incredibly expensive to fix because you have to change the actual system. Figure 6.4 displays how the cost to fix an error grows exponentially the later the error is found in the SDLC.

### PHASE 3: DESIGN

The primary goal of the *design phase* of the systems development life cycle is to build a technical blueprint of how the proposed system will work. During the analysis phase, end users and IT specialists work together to develop the business requirements for the proposed system from a logical point of view. That is, during analysis you document business requirements without respect to technology or the technical infrastructure that

### THE LIFE CYCLE OF BOISE OFFICE SOLUTIONS

Boise Office Solutions is a premier multinational distributor of office and technology products, office furniture, and paper, with annual sales totaling \$3.5 billion. The company, headquartered in Itasca, Illinois, has customers ranging in size from small organizations to multinational corporations. Boise Office Solutions has operations throughout the United States, Australia, New Zealand, Canada, and Mexico, and also serves customers in Europe.

Boise already had its own unique system life cycle process, but it lacked many of the traditional systems

development life cycle (SDLC) capabilities. The company began looking at the SDLC to help facilitate designs and incorporate them into its own life cycle process. The main interest was in creating diagrams, as well as generating code and creating good design. By standardizing an SDLC, any member of the development team can walk over to another developer's desk, jump right in, and they can work together on a project. Everyone uses the same process, with the same look and feel.<sup>2</sup>

will support the system. As you move into design, the project team turns its attention to the system from a physical or technical point of view. You take the business requirements generated during the analysis phase and define the supporting technical architecture in the design phase. The following are the primary activities you'll perform during the design phase:

1. *Design the technical architecture:* The **technical architecture** defines the hardware, software, and telecommunications equipment required to run the system. Most systems run on a computer network with each employee having a workstation and the application software running on a server. The telecommunications requirements encompass access to the Internet and the ability for end users to connect remotely to the server. You typically explore several different technical architectures before choosing the final technical architecture.
2. *Design the system model:* **Modeling** is the activity of drawing a graphical representation of a design. You model everything you build including screens, reports, software, and databases (with E-R diagrams as we described in *Extended Learning Module C*). There are many different types of modeling activities performed during the design phase including a graphical user interface screen design. The **graphical user interface (GUI)** is the interface to an information system. **GUI screen design** is the ability to model the information system screens for an entire system. You must decide many things when modeling a GUI, including the placement of items on the screen and the number of items contained in a drop-down list. You base your decisions as to how and where to display menu items on whatever is easiest for the users to use. If the menu items are placed incorrectly on the GUI, users could waste a significant amount of time just searching the GUI to find the correct item.

### PHASE 4: DEVELOPMENT

During the **development phase** of the systems development life cycle, you take all your detailed design documents from the design phase and transform them into an actual system. This phase marks the point at which you go from physical design to physical im-

# TEAM WORK

## WRITING TEST CONDITIONS

Your manager has asked you to test the cut and paste functionality for a word processing application. Write 10 detailed test conditions using the template below. Be sure to think about cutting and pasting such things as different fonts, varying font sizes, bold and italic

fonts, graphics, etc. We have provided a sample for you. Once you have completed your test conditions, estimate how many test conditions would be required to completely test the cut and paste functionality for Microsoft Word.

Test Condition Number	Date	Tester Name	Test Condition	Expected Result	Actual Result	Pass/Fail
1	5/27/2005	McLeod	Highlight text	Highlighted text	Highlighted text	Pass

plementation. The following are the two main activities you'll perform during the development phase:

1. *Build the technical architecture:* For you to build your system, you must first build the platform on which the system is going to operate. In the development phase, you purchase and implement equipment necessary to support the technical architecture you designed during the design phase.
2. *Build the database and programs:* Once the technical architecture is built, you initiate and complete the creation of supporting databases and writing the software required for the system. These tasks are usually undertaken by IT specialists, and it may take months or even years to design and create the databases and write all the software.

## PHASE 5: TESTING

The *testing phase* of the systems development life cycle verifies that the system works and meets all the business requirements defined in the analysis phase. Testing is critical. The following are the primary activities you'll perform during the testing phase:

1. *Write the test conditions:* You must have detailed test conditions to perform an exhaustive test. **Test conditions** are the detailed steps the system must perform along with the expected results of each step. The tester will execute each test condition and compare the expected results with the actual results to verify that the system functions correctly. Each time the actual result is different from the expected result, a "bug" is generated, and the system goes back to development for a "bug fix." A typical systems development effort has hundreds or thousands of test conditions. You must execute and verify all of these test conditions to ensure the entire system functions correctly.

2. *Perform the testing of the system:* You must perform many different types of tests when you begin testing your new system. A few of the more common tests include:
  - **Unit testing**—tests individual units or pieces of code for a system.
  - **System testing**—verifies that the units or pieces of code written for a system function correctly when integrated into the total system.
  - **Integration testing**—verifies that separate systems can work together.
  - **User acceptance testing (UAT)**—determines if the system satisfies the business requirements and enables users to perform their jobs correctly.

## PHASE 6: IMPLEMENTATION

During the *implementation phase* of the systems development life cycle you distribute the system to all the users and they begin using the system to perform their everyday jobs. The following are the two primary activities you'll perform during the implementation phase:

1. *Write detailed user documentation:* When you install the system, you must also provide employees with **user documentation** that highlights how to use the system. Users find it extremely frustrating to have a new system without documentation.
2. *Provide training for the system users:* You must also provide training for the users who are going to use the new system. You can provide several different types of training, and two of the most popular are online training and workshop training. **Online training** runs over the Internet or off a CD or DVD. Employees perform the training at any time, on their own computers, at their own pace. This type of training is convenient because they can set their own schedule to undergo the training. **Workshop training** is held in a classroom environment and is led by an instructor. Workshop training is most suitable for difficult systems for which employees need one-on-one time with an individual instructor.

You also need to choose the implementation method that best suits your organization, project, and employees to ensure a successful implementation. When you implement the new system, you have four implementation methods you can choose from:

1. **Parallel implementation** uses both the old and new systems until you're sure that the new system performs correctly.
2. **Plunge implementation** discards the old system completely and immediately uses the new system.
3. **Pilot implementation** has only a small group of people using the new system until you know it works correctly and then the remaining people are added to the system.
4. **Phased implementation** installs the new system in phases (e.g., accounts receivable, then accounts payable) until you're sure it works correctly and then the remaining phases of the new system are implemented.

## PHASE 7: MAINTENANCE

Maintaining the system is the final phase of any systems development effort. During the *maintenance phase* of the systems development life cycle, you monitor and support the new system to ensure it continues to meet the business goals. Once a system is in place, it must change as your business changes. Constantly monitoring and supporting the new

## INDUSTRY PERSPECTIVE

### MAKING YOUR SOFTWARE WORK

Faulty software installations cost companies billions of dollars. Here are three tips from Mercury Interactive, a software testing company, on how to avoid the common pitfalls of implementation.

1. *Keep it simple*—The trouble with implementing new software is rarely in the software itself. The problems occur when companies purchase software and then attempt to customize it to meet their needs. These unique one-off implementations create complexity and problems. Try to minimize customization of purchased software.
2. *Set clear goals*—You must develop a clear understanding of what the business objectives are for the software and after implementation ascertain that the software is adding the expected value.
3. *Heed the old adage that less is more*—The history of IT has been ever faster, more comprehensive, and more efficient software. The temptation to go for new systems is obvious, but organizations should focus on improving the quality of their existing software first to get the biggest bang for their buck.<sup>3</sup>

system involves making minor changes (for example, new reports or information retrieval) and reviewing the system to be sure that it continues to move your organization toward its strategic goals. The following are the two primary activities you'll perform during the maintenance phase:

1. *Build a help desk to support the system users*: One of the best ways to support users is to create a help desk. A **help desk** is a group of people who respond to users' questions. Typically, users have a phone number for the help desk they call whenever they have issues or questions about the system. Providing a help desk that answers user questions is a terrific way to provide comprehensive support for users using new systems.
2. *Provide an environment to support system changes*: As changes arise in the business environment, you must react to those changes by assessing their impact on the system. It might well be that the system needs to be adapted or updated to meet the ever-changing needs of the business environment. If so, you must modify the system to support the new business environment.

## Systems Development Methodologies

Most systems development is a chaotic activity, often characterized by the phrase “code and fix.” The software is written without much of an underlying plan, and the design of the system is cobbled together from many short-term decisions. This actually works pretty well if your system is small, but as your system grows it becomes increasingly difficult to add new features to the system. Furthermore, bugs become increasingly prevalent and difficult to fix.

Today, systems are so large and complex that teams of architects, analysts, developers, testers, and users must work together to create the millions of lines of custom-written code that drive these systems. For this reason, developers have come up with a number of different systems development life cycle methodologies to organize and make sense of the process—such as, the waterfall, rapid application development (RAD), extreme programming, and agile methodologies.

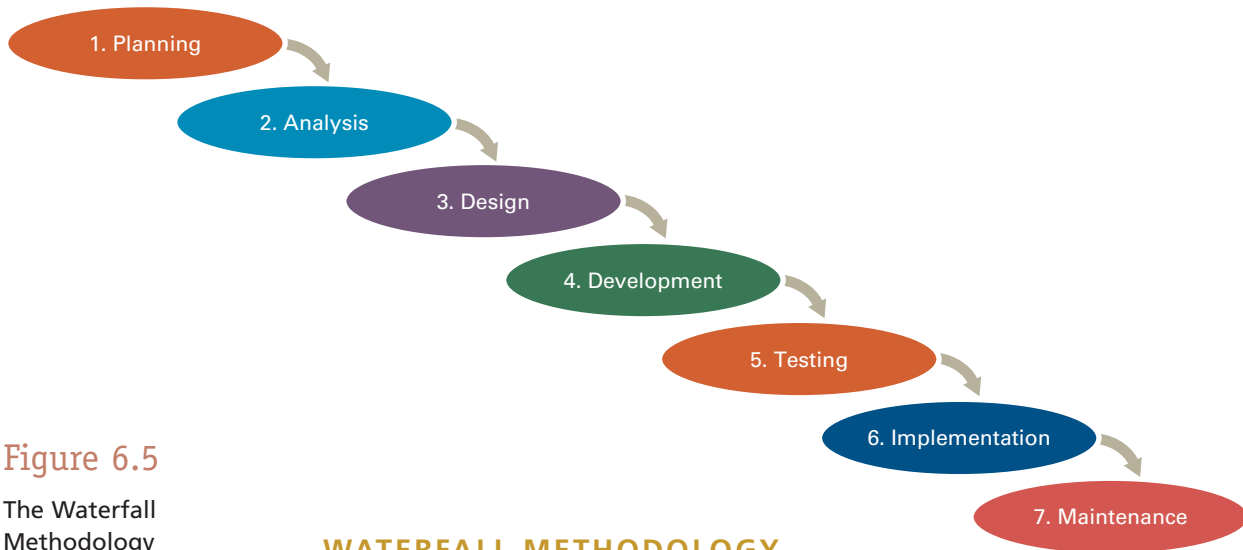


Figure 6.5

The Waterfall Methodology

### WATERFALL METHODOLOGY

The traditional *waterfall methodology* is a sequential, activity-based process in which one phase in the SDLC is followed by another, from planning through implementation. The waterfall methodology is one of the oldest software development methodologies and has been around for over 30 years (see Figure 6.5). The success rate for software development projects pursuing this approach is only about 1 in 10. The primary reason for such a low success rate for the waterfall methodology is that it does not accommodate the level of uncertainty characteristic of and thus the creativity often required to complete complex software development projects including:

- **The business problem:** Accurately defining and articulating the business problem in terms of what the business users actually require. It may take persistence and more than one stab at the problem to distinguish between what users want and what they really need.
- **The plan:** Managing cost, resource, and time constraints. What happens to the schedule if a programmer quits? How will a schedule delay in a specific phase impact the total cost of the project?
- **The solution:** Defining the appropriate IT infrastructure that is flexible, cost-efficient, scalable, and reliable. Finalizing the optimal IT infrastructure is not likely to happen via a step-by-step approach, when so many factors must converge from all directions.

The problem with the waterfall methodology is that it assumes users can specify all business requirements in advance. Unfortunately, business requirements change as the business changes, which calls for considerable feedback and reiterative consultation for all business requirements as the development process proceeds. You can think of software as “soft” and it must be easily changed and manipulated to meet the changing dynamics of an organization. As business problems evolve over time, so must the software. For this reason, it is counterproductive to define all requirements precisely upfront since, by the time the software goes into production, which can be several months or even years after completing the initial analysis phase, chances are the business problem will have changed as well as the business.

### RAPID APPLICATION DEVELOPMENT METHODOLOGY

In response to the fast pace of the economy, rapid application development has become a popular route for accelerating systems development. The *rapid application develop-*

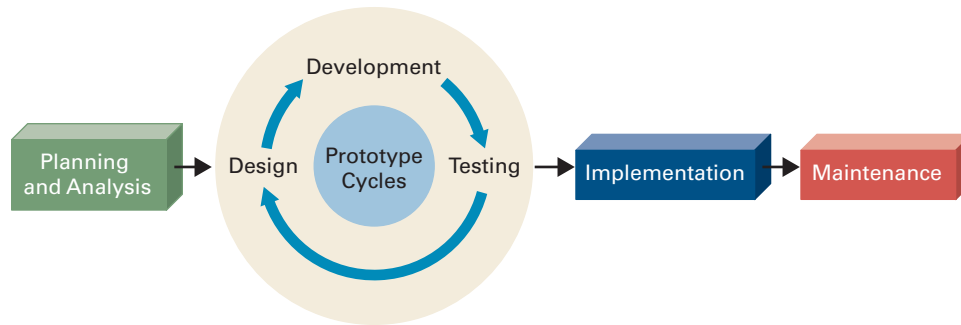


Figure 6.6

The Rapid Application Development Methodology

**ment (RAD)** (also called **rapid prototyping methodology**) emphasizes extensive user involvement in the rapid and evolutionary construction of working prototypes of a system to accelerate the systems development process (see Figure 6.6). The fundamentals of RAD include:

- Focus initially on creating a prototype (i.e., a working model) that looks and acts like the desired system.
- Actively involve system users in the analysis, design, and development phases.
- Accelerate collecting the business requirements through an interactive and iterative construction approach.

A **prototype** is a smaller-scale representation or working model of the users' requirements or a proposed design for an information system. The prototype is an essential part of the analysis phase when using the RAD methodology.

## EXTREME PROGRAMMING METHODOLOGY

The **extreme programming (XP) methodology** breaks a project into tiny phases and developers cannot continue on to the next phase until the current phase is complete. XP is a lot like a jigsaw puzzle; there are many small pieces. Individually, the pieces make no sense, but when they are combined together an organization can gain visibility into the entire system. The primary difference between the waterfall and XP methodologies is that XP divides its phases into iterations. For example, the waterfall approach develops the entire system, whereas XP develops the system in iterations (see Figure 6.7).

Microsoft Corporation developed Internet Explorer and Netscape Communications Corporation developed Communicator using extreme programming. Both companies did a nightly compilation (called a *build*) of the entire project, bringing together all the current components. They established release dates and expended considerable effort

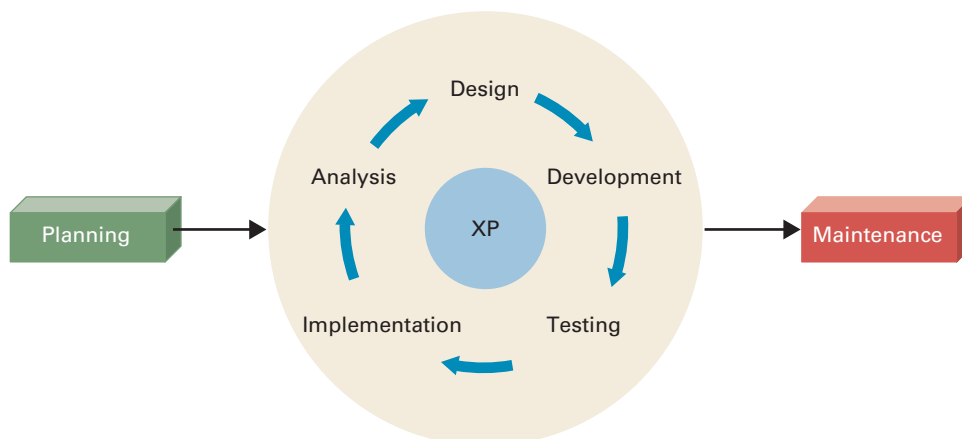


Figure 6.7

The Extreme Programming Methodology

to involve customers in each release. The extreme programming approach allowed both Microsoft and Netscape to manage millions of lines of code as specifications changed and evolved over time. Most important, both companies frequently held user design reviews and strategy sessions to solicit and incorporate user feedback.<sup>4</sup>

XP is a significant departure from traditional software development methodologies, and many organizations in different industries have developed successful software using it. One of the reasons for XP's success is that it stresses customer satisfaction. XP empowers developers to respond to changing customer and business requirements, even late in the systems development life cycle, and emphasizes teamwork. Managers, customers, and developers are all part of a team dedicated to delivering quality software. XP implements a simple, yet effective, way to enable group style development. The XP methodology supports quickly being able to respond to changing requirements and technology.

### AGILE METHODOLOGY

The *agile methodology*, a form of XP, aims for customer satisfaction through early and continuous delivery of useful software components. Agile is similar to XP but with less focus on team coding and more on limiting project scope. An agile project sets a minimum number of requirements and turns them into a deliverable product. Agile means what it sounds like: fast and efficient; small and nimble; lower cost; fewer features; shorter projects.

The Agile Alliance, a group of software developers, has made its mission to improve software development processes. Its manifesto includes the following tenets:

- Satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.<sup>5</sup>

### Outsourcing

You have two primary choices as to who will build your IT system. Here we're not talking about personal productivity software you can buy at a local computer store, but large software packages that may cost millions of dollars. First, you can choose *insourcing*, which involves using IT specialists within your organization to develop the system. Second, you can choose *outsourcing*, which is the delegation of specific work to a third party for a specified length of time, at a specified cost, and at a specified level of service.

Outsourcing, as we know it, grew up as the industrial revolution spread and companies specialized. The oldest form of outsourcing is specialized machinery. The original cotton and woolen mills that defined the start of the industrial revolution were self-contained entities. Mechanics made everything on site except the bricks and mortar for the buildings. Over time, people discovered that they could sell equipment to mills by designing and making cheaper and better equipment. They succeeded and gave birth to the outsourcing industry.

You may argue that this is not outsourcing. However, it meets the criteria: An outside party provides a product or service designed to meet the buyer's specification; and the buyer is motivated because the seller is offering specialized expertise and/or a lower price. Manufacturing is by far the biggest user of outsourcing and has done so with great success for many years. Industries such as automobiles, aircraft, and footwear are major outsourcers.

With competitive pressures to cut costs and reduce time-to-market, many organizations are looking to outsource their IT systems development. The Outsourcing Research Council recently completed a study indicating that human resources (HR) is a top outsourcing area for many companies. Fifty percent of the companies surveyed said they were already outsourcing some or all of their payroll processing and another 38 percent said they were considering it.<sup>6</sup>

Energizer, the world's largest manufacturer of batteries and flashlights, outsourced its HR operations to ADP, one of the top HR outsourcing companies. Energizer currently has more than 3,500 employees and 2,000 retired employees who all require multiple HR services. ADP provides Energizer with centralized call centers, transaction processing services, and Web-based employee self-service systems. Energizer's vice president of Human Resources, Peter Conrad, stated, "ADP was clearly the most capable, and offered the kind of one-stop shopping our company was looking for." For several of the systems provided by ADP employee usage topped over 80 percent in the first six months the systems were active.<sup>7</sup>

Another high profile company relying heavily on outsourcing some of its IT functions is Dell. By the end of 2005, Dell plans to increase the number of staff at its Indian call center and software development operations to 10,000. Dell opened its first customer contact center in Bangalore, India, in 2001, initially providing technical support for customers in the United States. It set up another contact center in Hyderabad, India, in 2004, and a third in Mohali, India, during 2005. The company has also set up a product team in Bangalore that develops and tests some of Dell's enterprise server and storage products and a software development center that serves Dell's internal information systems requirements.

According to PricewaterhouseCoopers' survey of CEOs from 452 of the countries' fastest growing companies, "Businesses that outsource are growing faster, larger, and more profitably than those that do not. In addition, most of those involved in outsourcing say they are saving money and are highly satisfied with their outsourcing service providers." Figure 6.8 lists common areas for outsourcing opportunities across some industries.<sup>8</sup>

Industry	Outsourcing Opportunities
Banking and finance	Check and electronic payment processing, credit report issuance, delinquency management, securities and trades processing
Insurance	Claims reporting and investigation, policy administration, check processing, risk assessment
Telecommunications	Invoice and bill production, transaction processing
Health care	Electronic data interchange, database management, accounting
Transportation	Ticket and order processing
Government	Loan processing, Medicaid processing
Retail	Electronic payment processing

Figure 6.8

Outsourcing Opportunities across Industries

The main reasons behind the rapid growth of the outsourcing industry include the following:

- **Globalization:** As markets open worldwide, competition heats up. Companies may engage outsourcing service providers to deliver international services.
- **The Internet:** Barriers to entry, such as lack of capital, are dramatically reduced in the world of e-business. New competitors enter the market daily.
- **Growing economy and low unemployment rate:** Building a competitive workforce is much harder and more expensive.
- **Technology:** Technology is advancing at such an accelerated rate that companies often lack the resources, workforce, or expertise to keep up.
- **Deregulation:** As private industries such as telecommunications and energy deregulate, markets open and competition increases.

The many benefits associated with outsourcing include:

- Increased quality and efficiency of a process, service, or function.
- Reduced operating expenses.
- Outsourcing noncore processes or nonrevenue-producing areas allows businesses to focus resources on their core profit-generating competencies.
- Reduced exposure to risks involved with large capital investments.
- Access to outsourcing service provider's economies of scale.
- Tap into the outsourcing services provider's expertise and best-in-class practices.
- Access to advanced technologies.
- Increased flexibility with the ability to respond quickly to changing market demands.
- Avoid costly outlay of capital funds.
- Reduced headcount and associated overhead expense.
- Reduced frustration and expense related to hiring and retaining employees in an exceptionally tight job market.
- Reduced time to market for products or services.

IT outsourcing today represents a significant opportunity for your organization to capitalize on the intellectual resources of other organizations by having them take over and perform certain business functions in which they have more expertise than the users in your company. Information technology outsourcing enables organizations to keep up with market and technology advances—with less strain on human and financial resources, and more assurance that the IT infrastructure will keep pace with evolving business priorities.

The outsourcing process is both similar to and quite different from the systems development life cycle. It's different in that you turn over much of the design, development, testing, implementation, and maintenance steps to another organization. It's similar in that your organization begins with planning and defining the project scope. It's during one of these phases that your organization may come to understand that it needs a particular system but it cannot be developed through insourcing.

Outsourcing involves telling another organization what you want. What you *want* is essentially the logical requirements for a proposed system, and you convey that information by developing a request for proposal. A **request for proposal (RFP)** is a formal document that describes in detail your logical requirements for a proposed system and invites outsourcing organizations (or “vendors”) to submit bids for its development. An RFP is the most important document in the outsourcing process. For systems of great

size, your organization may create an RFP that's hundreds of pages long and requires months of work to complete.

It's vitally important that you take all the time you need to create a complete and thorough RFP. Eventually, your RFP will become the foundation for a legal and binding contract into which your organization and the vendor will enter. For your organization, the ability to develop a complete and thorough RFP means that you completely understand what you have and what you want. For the vendors, a complete and thorough RFP makes it easier to propose a system that will meet most, if not all, your needs.

## OUTSOURCING OPTIONS

IT outsourcing for software development can take one of four forms (see Figure 6.9):

1. Purchasing existing software.
2. Purchasing existing software and paying the publisher to make certain modifications.
3. Purchasing existing software and paying the publisher for the right to make modifications yourself.
4. Outsourcing the development of an entirely new and unique system for which no software exists.

Which option should you choose? Every organization has to track financial information, for example, and there are several different systems they can purchase that help them perform this activity. Building a financial system for your organization would probably be a waste of your time and money since there are several good systems already built that will likely meet your organizational needs. So option 1 would be appropriate, purchasing existing software, such as Oracle Financials, an application suite your organization can buy that tracks all your organizational financial information.

In a geopolitical sense, there are furthermore three types of outsourcing:

1. **Onshore outsourcing** is the process of engaging another company in the same country for services. Much of the current jargon relating to outsourcing is based on the United States as the reference point. Thus, "onshore" outsourcing typically means contracting a U.S. company to provide business services.



Figure 6.9

Major Forms of Outsourcing Systems Development

2. **Nearshore outsourcing** is contracting an outsourcing arrangement with a company in a nearby country. Often this country will share a border with the native country. Again, this term is often used with the United States as the frame of reference. In this case, nearshore outsourcing will take place in either Canada or in Mexico.
3. **Offshore outsourcing** is contracting with a company that is geographically far away. For many companies, certain IT services, such as application development, maintenance, and help desk support, fall within the category of functions that are ideal for offshore outsourcing.

In today's high-tech world, the terms "onshore," "nearshore," and "offshore" refer only to the physical location. Thanks to the advancement in technology, terrestrial and satellite links, videoconferencing, online chats, e-mails, and telephones, a delivery system works as a single virtual unit, regardless of where people are actually located.

Although onshore and nearshore outsourcing are important forms in the outsourcing industry, when outsourcing is spoken about nowadays, it is usually offshore outsourcing that is being referenced.

**OFFSHORE OUTSOURCING** From a humble beginning as a mere cost-cutting concept, offshore outsourcing has gradually moved ahead and established itself as a successful business model by rendering not only cost-effective but also sophisticated and highly efficient quality services. According to International Data Corporation (IDC), U.S.-based companies tripled their offshore outsourcing spending from \$5.5 billion in 2000 to more than \$17.6 billion in 2005. The offshore outsourcing trend has overcome all barriers of political turmoil, language problems, and culture differences, and has proved that no matter in which part of the world your outsourcer resides, what really matters is industry-standard, high-quality service together with decisive cost advantage.<sup>9</sup>

Since the mid-1990s, major U.S. companies have been sending significant portions of their software development work offshore—primarily to vendors in India, but also to vendors in China, Eastern Europe (including Russia), Ireland, Israel, and the Philippines. The big selling point for offshore outsourcing to these countries is "good work done cheap." A programmer who earns \$63,000 per year in the United States is paid as little as \$5,000 per year overseas (see Figure 6.10). Companies can easily realize cost savings of 30 to 50 percent through offshore outsourcing and still get the same, if not better, quality of service.

Stories about U.S. companies outsourcing work offshore to India have been reported for years; however, it is becoming increasingly apparent that Romania, Bulgaria, Russia, China, Ghana, the Philippines, and dozens of other countries are also clamoring for and getting business from the United States. The reality is that offshore outsourcing is a growing trend. According to a recent study from Meta Group, the worldwide offshore outsourcing market will grow 20 percent annually through 2008. Meta also claims that offshoring growth will outpace outsourcing in general and predicts that the average enterprise will offshore 60 percent of application development by 2008 or 2009.<sup>10</sup>

What types of functions or projects make good candidates for offshore outsourcing? Data conversions and system migrations with well-defined requirements and specifications

Figure 6.10

Typical Salary Ranges for Computer Programmers

Country	Salary Range per Year
China	\$ 5,000–\$ 9,000
India	\$ 6,000– 10,000
Philippines	\$ 6,500– 11,000
Russia	\$ 7,000– 13,000
Ireland	\$21,000– 28,000
Canada	\$25,000– 50,000
United States	\$60,000– 90,000

### OUTSOURCING HELP DESKS OFFSHORE

There are two classes of help desk–support customers—high paying and low paying. Today, vendors are tailoring support based on the amount of money customers pay and are transferring the lower-paying customers to offshore support sites. More and more companies, however, are finding that offshore help desks are causing their customers major problems.

Jim Miller, chief technology officer at Creditex Inc., a Wall Street–based credit derivative trading firm, recently requested help with a Microsoft Exchange server that supports users in New York and London. The Microsoft help desk in India answered Jim’s call and the technician on the other end of the line merely consulted online Microsoft knowledge-base articles, some-

thing Miller said he could have done himself. The technician told him that the only way to rectify the problem was to rebuild the whole machine and reinstall Exchange, an effort that Miller said would have entailed “a huge investment in time, the loss of significant data and e-mail down for an extended period of time.”

Miller took a pass on the advice and asked the technician to place him back in the call queue, only to end up back in India four times. Finally, Creditex “escalated” its request for help to Microsoft support managers and finally received a call from a knowledgeable technician in Dallas who resolved the problem in 15 minutes.<sup>11</sup>

and minimal end-user interaction with the development team are typical projects taken offshore. Naturally, the company must be willing to allow its application code to be located offsite during development. Application development projects are also good offshore candidates. From a SDLC perspective, offshore work is most beneficial in the development and testing phases where end-user interaction is limited, and the task is well defined. For stable applications, most maintenance activities can be performed remotely so application maintenance is also a good candidate for offshore outsourcing. With the right communication infrastructure and a clear understanding of your company’s business language requirements, call center or help desk functions can also be moved offshore.

### THE ADVANTAGES AND DISADVANTAGES OF OUTSOURCING

Making the decision to outsource may be critical one way or the other to your organization’s success. Thus far in our discussion of outsourcing, we’ve alluded to some advantages and disadvantages of outsourcing. Following is a summary of the major advantages and disadvantages of outsourcing the systems development process, in order to help you make the important outsourcing decision.

**ADVANTAGES** Your organization may benefit from outsourcing because outsourcing allows you to:

- *Focus on unique core competencies:* By outsourcing systems development efforts that support noncritical business functions, your organization can focus on developing systems that support important, unique core competencies.
- *Exploit the intellect of another organization:* Outsourcing allows your organization to obtain intellectual capital by purchasing it from another organization. Often you won’t be able to find individuals with all the expertise required to develop a system. Outsourcing allows you to find those individuals with the expertise you need to get your system developed and implemented.

# TEAM WORK

## HOW MANY ARE THERE?

Assume your company is looking to outsource its payroll activities including calculating the payroll and generating the paychecks each month. Search the Internet to find different companies that offer this outsourcing

service. Fill in the following table comparing the advantages and disadvantages of the different companies you find. Discuss which one you would choose and why.

Company Name	Advantages	Disadvantages

- *Better predict future costs:* When you outsource a function, whether systems development or some other business function, you know the exact costs.
- *Acquire leading-edge technology:* Outsourcing allows your organization to acquire leading-edge technology without having to acquire technical expertise and bear the inherent risks of choosing the wrong technology.
- *Reduce costs:* Outsourcing is often seen as a money saver for organizations. Reducing costs is one of the important reasons organizations outsource.
- *Improve performance accountability:* Outsourcing involves delegating work to another organization at a specified level of service. Your organization can use this specified level of service to guarantee that it gets exactly what it wants from the vendor.

**DISADVANTAGES** Outsourcing may *not* be a beneficial option for you because it:

- *Reduces technical know-how for future innovation:* Outsourcing is a way of exploiting the intellect of another organization, so it can also mean that your organization will no longer possess that expertise internally. If you outsource because you don't have the necessary technical expertise today, you'll probably have to outsource for the same reason tomorrow.
- *Reduces degree of control:* Outsourcing means giving up control. No matter what you choose to outsource, you are in some way giving up control over that function.
- *Increases vulnerability of your strategic information:* Outsourcing systems development involves telling another organization what information you use and

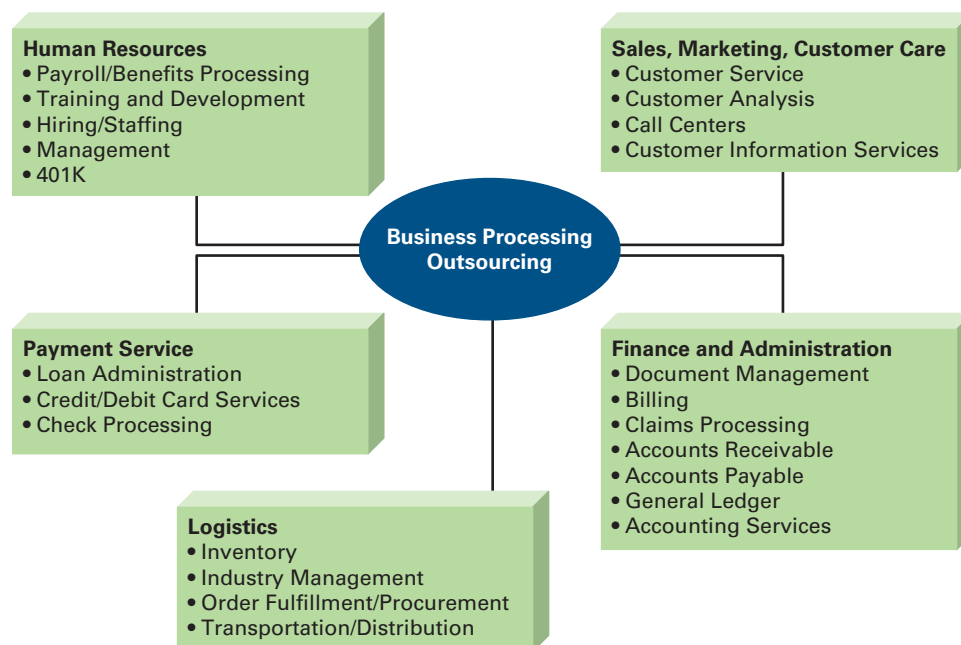
how you use that information. In doing so, you could be giving away strategic information and secrets.

- *Increases dependency on other organizations:* As soon as you start outsourcing, you immediately begin depending on another organization to perform many of your business functions.

## BUSINESS PROCESS OUTSOURCING (BPO)

Today's world economy is driving many companies to seek increasingly aggressive ways to meet the demands of shareholders, investors, and employees. In a quest for higher profits and increased competitive advantage, you would be hard-pressed to find an executive or manager not motivated by lower costs and better productivity. Additionally, most companies are focusing on value-added, customer-centric functions and striving to reduce investments in operations that don't directly affect the bottom line. What are major corporations doing to achieve all this? One strategy is business process outsourcing. **Business process outsourcing (BPO)** is using a contractual service to completely manage, deliver, and operate one or more (typically IT—or call center-intensive) business processes or functions. Figure 6.11 displays a few of the typical BPO business processes or functions.

IT services have been successfully outsourced for many years as an effective way for companies to lower cost and raise productivity. Business process management uses technology to break down barriers between traditional functional silos, such as those found in finance, order fulfillment, and call centers. The goal is to redesign the process, reduce unnecessary steps, and eliminate redundancies. Outsourcing uses skills and resources of specialized outside service providers to perform many of these critical, yet noncore activities. In short, BPO means examining the processes that make up the business and its functional units and then working with specialized service providers to both reengineer and outsource these processes. Thus, BPO lowers costs and raises productivity at the same time, speeding implementation and ensuring that the intended benefits really hit the bottom line.



**Figure 6.11**  
Business Processes and  
Functions of Business  
Process Outsourcing

## INDUSTRY PERSPECTIVE

### DO YOUR HOMEWORK!

The complexity of today's IT systems and the technical demands placed on an organization and its IT department are leading many companies down the path of business process outsourcing.

However, achieving success is no simple matter. Not only is it important to know when to outsource, it's also essential to know how to use outsourcing to your full advantage. Here are some questions you should ask before embracing BPO:

- **Does the outsourcing opportunity match the organization's business needs?** If an outsourcing initiative can create a strategic advantage, then it's worth pursuing.
- **Will outsourcing improve performance?** A successful outsourcing initiative translates into service that is better than it would be if an organization handled the tasks internally.
- **Are the financial projections accurate?** Take a critical look at the numbers, particularly those generated by a vendor, and try to spot assumptions, oversimplifications, or just plain misleading figures.

There can also be dramatic savings in technology investments, such as enterprise resource planning (ERP) systems and call centers. Because of the transactional nature of BPO functions, BPO service providers themselves make heavy investments in these technologies, thereby saving their clients the burden of purchasing, implementation, ongoing support, and maintenance of ERP software and call center technology.

Ideal processes for BPO are ones that can be standardized or automated. The financial services industry is particularly suitable for BPO on account of its repetitive and transaction-oriented nature. Financial services is one of the major markets for BPO, particularly its payroll and call center functions. Financial services pioneered BPO, and leads other business sectors in spending in this outsourcing area. Merrill Lynch decided that its IT help desk would be a good candidate for BPO. Merrill Lynch needed a BPO service provider to manage the support function for approximately 11,000 users in the United States and the United Kingdom. The help desk was broken down into a number of components: a help desk for routine troubleshooting; a service to deal with hardware failure; desk-side support for investment managers; and associated procurement processes, asset management, and reporting functions. Merrill Lynch was able to reduce its IT costs by 20 percent using a BPO service provider.<sup>12</sup>

## Prototyping

**Prototyping** is the process of building a model that demonstrates the features of a proposed product, service, or system. A prototype, then, is simply a model of a proposed product, service, or system. If you think about it, people prototype all the time. Automobile manufacturers build prototypes of cars to demonstrate safety features, aerodynamics, and comfort. Building contractors construct models of homes and other structures to show layout and fire exits.

In systems development, prototyping can be a valuable tool for you. Prototyping is an iterative process in which you build a model from basic business requirements, have

## LASTMINUTE.COM

Launched in October 1998, lastminute.com, the online provider of last minute travel and leisure solutions, has grown to become one of Europe's most popular and successful e-commerce sites, offering a range of travel, entertainment, and gift services.

In order to allow team members to effectively manage the design, modeling, quality assurance, and deployment aspects of software development projects, lastminute.com needed to select a systems development methodology. After it went through an extensive two-month evaluation period, lastminute.com chose to implement an agile methodology—a stable, well-

crafted methodology that really appealed to its developers. The methodology makes no demands on how lastminute.com structures its code; therefore, it was able to begin using it right from the start. The software development teams at lastminute.com are focused on specific business areas such as flights, hotels, and leisure. lastminute.com anticipates that the agile systems development methodology will help it to quickly and effectively deliver further functionality in these key business areas to enhance customer experience and generate increased revenues.<sup>13</sup>

other users review the prototype and suggest changes, and further refine and enhance the prototype to include suggestions. Most notably, prototyping is a dynamic process that allows you to see, work with, and evaluate a model and suggest changes to that model to increase the likelihood of success of the proposed system. Prototyping is a common tool in extreme programming, RAD, and the agile methodology.

You can use prototyping to perform a variety of functions in the systems development process:

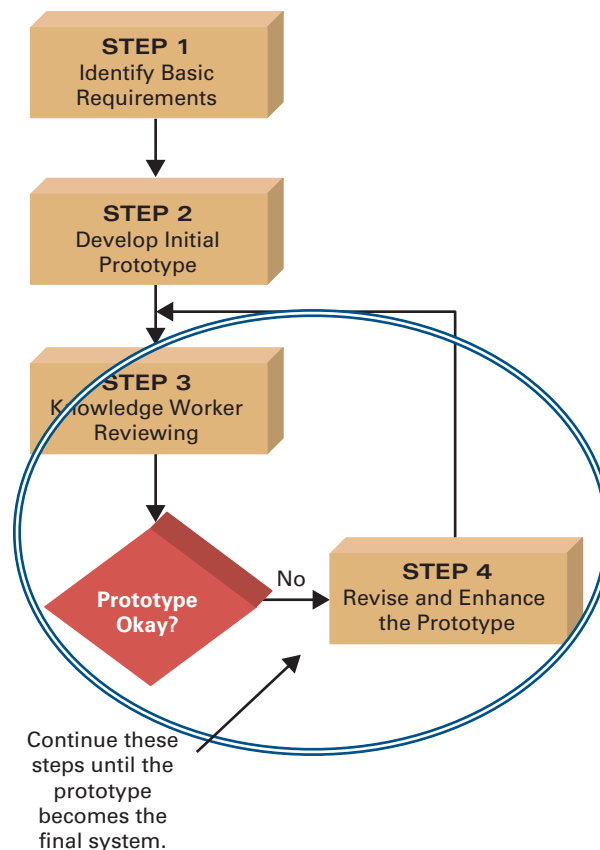
- *Gathering requirements:* Prototyping is a great requirements gathering tool. You start by simply prototyping the basic system requirements. Then you allow users to add more requirements (information and processes) as you revise the prototype.
- *Helping determine requirements:* In many systems development projects, users aren't sure what they really want. They simply know that the current system doesn't meet their needs. In this instance, you can use prototyping to help users determine their exact requirements.
- *Proving that a system is technically feasible:* Let's face it, there are some things to which you cannot apply technology, and knowing whether you can or not is often unclear while defining the scope of the proposed system. If you're uncertain about whether something can be done, prototype it first. A prototype you use to prove the technical feasibility of a proposed system is a ***proof-of-concept prototype***.
- *Selling the idea of a proposed system:* Many people resist changes in IT. The current system seems to work fine, and they see no reason to go through the process of developing and learning to use a new system. In this case, you have to convince them that the proposed system will be better than the current one. Because prototyping is relatively fast, you won't have to invest a lot of time to develop a prototype that can convince people of the worth of the proposed system. A prototype you use to convince people of the worth of a proposed system is a ***selling prototype***.

## THE PROTOTYPING PROCESS

Prototyping has proved an excellent tool in systems development. The prototyping process involves four steps (see Figure 6.12):

1. *Identify basic requirements:* During the first step, you gather the basic requirements for a proposed system. These basic requirements include input and output information desired and, perhaps, some simple processes. At this point, you're typically unconcerned with editing rules, security issues, or end-of-period processing (for example, producing W-2s for a payroll system at the end of the year).
2. *Develop initial prototype:* Having identified the basic requirements, you then set out to develop an initial prototype. Most often, your initial prototype will include only user interfaces, such as data entry screens and reports.
3. *User review:* Step 3 starts the truly iterative process of prototyping. When users first initiate this step, they evaluate the prototype and suggest changes or additions. In subsequent returns to step 3 (after step 4), they evaluate new versions of the prototype. It's important to involve as many users as possible during this iterative process. This will help resolve any discrepancies in such areas as terminology and operational processes.
4. *Revise and enhance the prototype:* The final sequential step in the prototyping process is to revise and enhance the prototype according to any user suggestions. In this step, you make changes to the current prototype and add any new requirements. Next, you return to step 3 and have the users review the new prototype; then step 4 again, revision, and so on.

**Figure 6.12**  
Prototyping Steps



Who uses prototyping and for what purpose determines how the prototyping process occurs. Most often, IT specialists use prototyping in the SDLC only to form a technical system blueprint; on the other hand, if end users develop prototypes to create their own system, they would cycle through steps 3 and 4 until the prototype becomes the final system.

Most often, IT specialists develop prototypes using special prototyping development tools. Many of these tools don't support the creation of a final system—you simply use them to build prototypes. Therefore, the finished prototype becomes a blueprint or technical design for the final system. In the appropriate stages of the SDLC, IT specialists implement the prototypes in another application development environment better suited to the development of whole systems.

## THE ADVANTAGES OF PROTOTYPING

- *Encourages active user participation*—First and foremost, prototyping encourages users to actively participate in the development process. As opposed to interviewing and reviewing documentation, prototyping allows users to see and work with working models of the proposed system.
- *Helps resolve discrepancies among users*—During the prototyping process, many users participate in defining the requirements for and reviewing the prototype. The word “many” is key. If at least several users participate in prototyping, you'll find it's much easier to resolve any discrepancies the users may encounter.
- *Gives users a feel for the final system*—Prototyping, especially for user interfaces, provides a feel for how the final system will look and work. When users understand the look and feel of the final system, they are more apt to see its potential for success.
- *Helps determine technical feasibility*—Proof-of-concept prototypes are great for determining the technical feasibility of a proposed system.
- *Helps sell the idea of a proposed system*—Prototypes can help break down resistance barriers and sell the system. Many people don't want new systems because the old one seems to work just fine, and they're afraid the new system won't meet their expectations and won't work properly. If you provide them with a working prototype that proves the new system will be successful, they will be more inclined to buy into it.

## THE DISADVANTAGES OF PROTOTYPING

- *Leads people to believe the final system will follow*—When a prototype is complete, many people believe that the final system will follow shortly. After all, they've seen the system at work in the form of a prototype—how long can it take to bring the system into production? Unfortunately, it may take months or years. You need to be sure that people understand that the prototype is only a model, not the final system missing only a few simple bells and whistles.
- *Gives no indication of performance under operational conditions*—Prototypes very seldom take all operational conditions into consideration. This problem surfaced for the Department of Motor Vehicles in a state on the East Coast. During prototyping, the system, which handled motor vehicle and driver registration for the entire state, worked fine for 20 workstations at two locations. When the system was finally installed for all locations (which included more than 1,200 workstations), the system spent all its time just managing communications traffic; it had absolutely no time to complete any transactions. This is potentially the most significant drawback to prototyping. You must prototype operational conditions as well as interfaces and processes.

## INDUSTRY PERSPECTIVE

### ST. JUDE CHILDREN'S RESEARCH HOSPITAL CURES

St. Jude Children's Research Hospital ([www.stjude.org](http://www.stjude.org)) works to find cures for children with catastrophic illnesses through research and treatment. Researchers at St. Jude continually conduct clinical trials to study the effects on patients of new combinations, types, and amounts of drugs and other medical therapies.

Before a new clinical trial can begin, each research team must establish a treatment regime—known as a clinical research protocol—and a protocol application/database to collect the unique patient data gathered during the study. Historically, protocol applications/databases were coded from scratch: a labor-intensive, time-consuming process that kept software development costs high while also hindering research efforts. An especially complex application/database could take from several months to a year for full life cycle development. Because of the long development cycles, many

of the protocol applications/databases failed to meet researchers' ongoing needs. Even "standard" industry practices were insufficient for this environment because requirements changed so rapidly that they were out of date before developers could produce a custom application/database.

The St. Jude team now uses the analysis and design features of the systems development life cycle to identify requirements—including baselining those necessary for compliance with the federal Food and Drug Administration (FDA) and Health Insurance Portability and Accountability Act (HIPAA). The new analysis and design methodologies also help improve communications between business analysts and developers by providing a way for analysts to clearly define and track requirements.<sup>14</sup>

- *Leads the project team to forgo proper testing and documentation*—You must thoroughly test and document all new systems. Unfortunately, many people believe they can forgo testing and documentation when using prototyping. After all, they've tested the prototype; why not use the prototype as the documentation for the system? Don't make this mistake.

## End-User Development

**End-user development (EUD)** is the collection of methods for the creation of important software applications by end users. End users are individuals who will use a system, who, although skilled in their own domain, are not IT or computer experts, and yet they know very well what they want from a system and are capable of being involved in prototyping, since they have basic IT skills.

End-user development of applications is an important source of organizational information systems. Applications developed by end users support a wide range of decision-making activities and contribute to business processing in a wide range of tasks. The major tools used for end-user system development have been spreadsheets and database management systems and Web development.

Rapidly gaining in acceptance is that end-user development can be a potent source of stress relief rather than a cause of stress. Rather than combating the trend toward end-user application development, IT staff should leverage it to offload solution building to end users. IT then frees its own scarce resources for complex, visible, infrastructure management tasks. A successful strategy for end-user development relies on two keys: (1) knowing which applications are good candidates and (2) providing end users with the right tools.

## WHICH APPLICATIONS FOR IT TO OFFLOAD

The following checklist helps IT staff to determine which applications are in IT's domain and which are good candidates for end-user development. IT delivers maximum value to the enterprise by focusing on high-cost, high-return applications with the following characteristics:

- Infrastructure-related
- Mission-critical including ERP, CRM, business intelligence and e-business
- Support large numbers of concurrent users, for example, call center applications

Other applications may be good candidates for end-user development.

## THE RIGHT TOOL FOR THE JOB

Requirements of end-user development tools and enterprise development tools (for IT analysts) are quite different. Ease of use is paramount for end-user development tools. That's because end users are not skilled programmers and might use the developmental tool so infrequently that they can forget commands that aren't intuitive. Therefore, end users must have development tools that:

- *Are easy to use:* This is essential for rapid, low-cost development. For application programs, specific characteristics of ease-of-use include:
  - Simple data entry
  - Error checking for value lists and ranges
  - Easy report generation, for example, by drag and drop
  - Ease of Web publishing
- *Support multiple platforms:* To minimize support requirements, end users should select one or two development tools that run on the range of hardware platforms and operating systems.
- *Offer low cost of ownership:* Cost factors include not only the tool's purchase price, but also training time, speed of application development, and required skill level. When assessing cost of ownership you should ask, "How easy is it to use after not having used it in a month or two?" and "How quickly can someone with subject matter expertise develop a simple application?"
- *Support a wide range of data types:* By its very nature, data is dynamic. Therefore, the toolset should support all the features normally found in database products.

However, not all end-user development is greatly accepted for the following reasons:

- *Version control:* One person creates a spreadsheet and distributes it to five others, each of whom modifies the formulas and passes the spreadsheet to others. Soon, scores of hundreds of people in the enterprise are working with a different version of the document.
- *Low quality:* It is estimated that 30 percent of spreadsheets contain errors, not only entry errors, which occur at the rate of one per each 100 cells, but also serious inaccuracies.
- *Loss of control:* IT groups traditionally struggle to attain a balance between centralized control and rapid response to workgroup needs. End-user development is sometimes seen as a relinquishing of control.

## ON YOUR OWN

### BUILD VS. BUY

When an organization decides it needs a new information system, it has many acquisition/development choices, two of which are (1) purchasing the information system from a vendor and (2) building the system itself. Typically, a cost/benefit analysis forms the basis of this buy versus build decision. You must consider the following during the buy versus build decision:

- Are there any currently available products that fit the organization's needs?
- Are there features that are not available but important enough to warrant the expense of in-house development?
- Can the organization customize or modify an existing application or application suite to fit its needs?
- Is there a justification to purchase or develop based on the cost of acquisition?

Search the Internet to find other criteria not mentioned above that would help an organization with the build versus buy decision.

We need to stress to you that end-user development of applications is an important source of many organizational information systems. End-user development aims to empower users who are not professional programmers to develop or modify software.

Applications developed by end users support a wide range of decision-making activities and contribute to business processing in a wide range of tasks. End users will no longer depend solely on programmers, enabling the production of a wide range of applications in a shorter amount of time as well as the increased number and diversity of people creating applications will promote innovation.

## Summary: Student Learning Outcomes Revisited

**1. List the seven steps in the systems development life cycle (SDLC) and associated activities for each step.** The *systems development life cycle (SDLC)* is a structured step-by-step approach for developing information systems. The seven steps and associated activities are as follows:

1. *Planning*: Define system, set project scope, develop project plan.
2. *Analysis*: Gather business requirements.
3. *Design*: Design technical architecture, design system models.
4. *Development*: Build technical architecture, build database and programs.
5. *Testing*: Write the test conditions, perform testing.

**6. *Implementation***: Perform user training, write user documentation.

**7. *Maintenance***: Provide a help desk and support system changes.

**2. Describe the four systems development methodologies.** The traditional *waterfall methodology* is a sequential, activity-based process in which each phase in the SDLC is performed sequentially from planning through implementation. *Rapid application development (RAD)* (also called *rapid prototyping methodology*) emphasizes extensive user involvement in the rapid and evolutionary construction of working prototypes of a system to accelerate the systems development process. *Extreme programming (XP) methodology*

breaks a project into tiny phases, and developers cannot continue on to the next phase until the first phase is complete. The *agile methodology*, a form of XP, aims for customer satisfaction through early and continuous delivery of useful software components. Agile is similar to XP but with less focus on team coding and more on limiting project scope.

3. **Define the role of outsourcing.** *Outsourcing* is the delegation of specific work to a third party for a specified length of time, at a specified cost, and at a specified level of service.
4. **List and describe the three different forms of outsourcing.** *Onshore outsourcing* is the process of engaging another company within the same country for services. *Nearshore outsourcing* refers to contracting an outsourcing arrangement with a company in a nearby country. *Offshore outsourcing* refers to contracting with a company that is geographically far away. For many companies, certain IT services, such as application development, maintenance, and help desk support, fall within the category of functions that are ideal for offshore outsourcing.
5. **Describe business process outsourcing (BPO).** *Business process outsourcing (BPO)* is using a contractual service to completely manage, deliver, and operate one or more (typically IT—

or call center-intensive) business processes or functions.

6. **Describe prototyping and profile an example of a prototype.** *Prototyping* is the process of building a model that demonstrates the features of a proposed product, service, or system. A prototype, then, is simply a model of a proposed product, service, or system. Automobile manufacturers build prototypes of cars to demonstrate safety features, aerodynamics, and comfort. Building contractors construct models of homes and other structures to show layout and fire exits. And systems developers model, or prototype, a system.
7. **Describe the advantages of prototyping.** Prototyping encourages active knowledge worker participation in the development process. As opposed to interviewing and the reviewing of documentation, prototyping allows knowledge workers to see and work with working models of the proposed system. This gives users a feel for the system. Prototyping also helps resolve discrepancies among the many end users participating in defining the requirements for and reviewing the prototype. Finally, prototypes help determine the technical feasibility of a system, and prototypes can help sell users on a new system.

## CLOSING CASE STUDY ONE

### GETTING SOFTWARE DEVELOPMENT RIGHT

Effective software development is the backbone of today's most successful businesses. Seeking a competitive edge, organizations are expanding the roles of IT leaders, business leaders, and developers who participate in the strategic software development process. Together, these stakeholders are charged with maximizing the value of corporate information across their enterprises.

Still, market research shows that software development projects have a high risk of failure. Many projects suffer from delays and budget overruns. Others don't fulfill the original project scope and some wind up being canceled entirely. Industry regulations and service level agreements mean there's less margin for

experiment and error. This pressure can result in missed financial opportunities and squandered competitive advantages.

Poor project planning, poor cost estimating, poor measurements, poor milestone tracking, poor change control, and poor quality control are the six most frequent problems that can undermine software projects, according to industry experts. Their impact hits the bottom line—software errors cost the American economy nearly \$60 billion a year.

U.S. businesses spent \$234 billion in 2004 to buy or build software, and roughly 37 percent of that figure, or about \$86 billion, was spent on software built

in-house, according to Forrester Research Inc. of Cambridge, Massachusetts. With so much money at stake, organizations can't afford to embark on software projects that are doomed to delay or complete failure. So, it's time to redefine the way that software is conceived, designed, built, and delivered in order to meet quality and user goals on time and on budget.

Historically, programming has largely been an unpredictable art form in many organizations. Now it's time for software development to become a manageable, repeatable business process. Getting software "right" requires a lot of effort incorporating the people, processes, and technology involved in the development process. Working together, business people, IT operations people, and application developers must minimize gaps in communication and comprehension between their specialized roles. These communication gaps have only intensified with globalization (distributed development teams and the rise of offshoring and outsourcing) and the increased complexity of IT.

In a managed business process, the software development team is integrated into the overall fabric of the enterprise, ensuring their participation as an equal

partner in shaping business-driven IT initiatives. In fact, software development has the potential to dramatically improve the way business units and IT groups collaborate.<sup>15</sup>

### Questions

1. What are some ways that a company can help to minimize the poor project planning, poor cost estimating, poor measurements, poor milestone tracking, poor change control, and poor quality control of software development?
2. One specific problem in any software development project is testing. Describe some ways in which an organization can help ensure proper testing in software development.
3. Evaluate the SDLC phases that seem most important to you and outline specific areas of the software development process that deserve greatest attention.
4. Explain the cost of finding errors later rather than sooner and explain how more time spent in the analysis and the design phases could save money.

## CLOSING CASE STUDY TWO

### TO OUTSOURCE OR NOT TO OUTSOURCE

At first glance, retailer Sears, Roebuck & Co. and financial services company Huntington National Bank seem to have little in common. Over the past few years, however, the CIOs at both companies faced a similar challenge. To outsource or not to outsource, that was the question.

#### SEARS, ROEBUCK

CIO Gerald F. Kelly Jr. joined Sears, Roebuck & Co. in order to turn around the company's IT department as Sears focused on ways to compete with Target and Wal-Mart, among others. Kelly was informed in his first few days on the job that the company's aging and unreliable technical infrastructure was a big problem. There were signs that the company had not invested appropriately and had not kept the infrastructure at a level of capability consistent with business objectives. Among the signs were network failures, CPU failures,

extended recovery times and root cause analysis, and lack of redundancy, backup, and recovery.

Kelly knew Sears was a retail company, not a technology company. He also felt that infrastructure was largely a commodity. If he insourced this work, it would mean denying a number of employees access to the best opportunities for career growth (those that support the core mission). So he began to investigate alternatives that would minimize the time issue and maximize the economic potential of transforming the IT infrastructure.

So in the end the decision for Sears to outsource wasn't led by the usual "to outsource or not to outsource" debate. It was led by an analysis of where the company was headed. After nine months of analysis, Sears executives weighed the time and costs involved in insourcing the IT infrastructure versus outsourcing it against the effect outsourcing would have on opportu-

nities for affected employees. It seemed that outsourcing would allow Sears to get the maximum value out of its dispersed infrastructure, achieve the goal of a stable infrastructure faster, save a significant amount of money, avoid having to hire a large number of technical specialists—and free up remaining staff for developing new retail systems. Kelly decided to outsource the infrastructure work.

### HUNTINGTON NATIONAL BANK

Huntington National Bank already relied on an outsourcer for deposit and loan systems; the outsourcer was hoping to win some of the financial services BPO market. Joe Gottron, CIO for Huntington National Bank and a 16-year IBM veteran, understood intimately the nature of outsourcing relationships, which led him to some reservations. His main concern was that outsourcing agreements are loss leaders to start, with revenue built into the latter year. Gottron, however, was particularly interested in the possibilities of outsourcing business processes.

Once Gottron could compare internal benchmarks to the BPO and other outsourcing offers, it became clear that the bank's size was a major factor in the BPO decision. With 8,300 employees (500 of them in IT), Joe stated, "We're big enough to afford good talent but small enough to get in a room, figure things out and

make things happen." A larger financial services company, more geographically dispersed, with more people or more complexity, probably could have gotten more value out of the BPO proposition, Gottron says, especially in terms of costs. The sourcing decision process took 18 months. Huntington National Bank chose to reject the BPO proposal and keep IT in-house based upon executives' understanding of where IT improvements should be made, business alignment and project management concerns, and the fact that outsourcing would likely save little, if any, money in the long term and could introduce risk to the corporate culture.<sup>16,17</sup>

### Questions

1. Summarize the reasons for the outsourcing decisions of Sears, Roebuck & Co and Huntington National Bank.
2. Explain why Sears, Roebuck & Co. did not insource its IT development.
3. Explain why Huntington National Bank did not outsource its IT development.
4. Describe some of the factors that may have "forced" Sears to outsource.
5. Describe some of the factors that may have "forced" Huntington National Bank to insource.

## Key Terms and Concepts

Agile methodology, 290  
 Analysis phase, 282  
 Business process outsourcing (BPO), 297  
 Business requirement, 282  
 Critical success factor (CSF), 281  
 Design phase, 283  
 Development phase, 284  
 End-user development (EUD), 302  
 Extreme programming (XP) methodology, 289  
 Feature creep, 281  
 Graphical user interface (GUI), 284  
 GUI screen design, 284  
 Help desk, 287  
 Implementation phase, 286  
 Insourcing, 290  
 Integration testing, 286  
 Joint application development (JAD), 282

Maintenance phase, 286  
 Modeling, 284  
 Nearshore outsourcing, 294  
 Offshore outsourcing, 294  
 Online training, 286  
 Onshore outsourcing, 293  
 Outsourcing, 290  
 Parallel implementation, 286  
 Phased implementation, 286  
 Pilot implementation, 286  
 Planning phase, 280  
 Plunge implementation, 286  
 Project manager, 282  
 Project milestone, 282  
 Project plan, 281  
 Project scope, 281  
 Project scope document, 281

Proof-of-concept prototype, 299

Prototype, 289

Prototyping, 298

Rapid application development (RAD) (rapid prototyping) methodology, 288

Request for proposal (RFP), 292

Requirements definition document, 282

Scope creep, 281

Selling prototype, 299

Sign-off, 282

Systems development life cycle (SDLC), 280

System testing, 286

Technical architecture, 284

Test condition, 285

Testing phase, 285

Unit testing, 286

User acceptance testing (UAT), 286

User documentation, 286

Waterfall methodology, 288

Workshop training, 286

## Short-Answer Questions

1. What is the systems development life cycle (SDLC)?
2. What are the seven steps in the SDLC?
3. What is a critical success factor?
4. What is feature creep?
5. How does a project plan help the project manager do his or her job?
6. In which step in the SDLC do you define business requirements?
7. Why would a company outsource?
8. In which step in the SDLC do you build the technical architecture?
9. Why must you provide sign-off on the business requirements?
10. Will a project be successful if you miss business requirements?
11. What is business process outsourcing?
12. Why would you build a prototype?
13. What is a selling prototype?
14. What is an advantage of end-user development?
15. What is an advantage of prototyping?
16. How do onshore, nearshore, and offshore outsourcing differ?

## Assignments and Exercises

1. **SDLC AND THE REAL WORLD** Think of the seven steps in the SDLC and apply them to one of your daily activities—for example, getting dressed in the morning. First, you plan what you are going to wear. This will vary depending on what you are going to do that day which could mean shorts, formal wear, jeans, or other attire. Second, you analyze what you have in your closet compared to what you plan to wear. Third, you design the outfit. Fourth, you get the clothes out of the closet and assemble them on your bed. Fifth, you test the outfit to ensure it matches and fits your day's plans. Sixth, you put on the outfit. Seventh, you wear the outfit throughout the day adjusting it as needed.
2. **HOW CREATIVE ARE YOU?** You've been appointed as the manager of the design team for Sneakers-R-Us. Your first activity is to design the GUI for the main system. The only requirements you are given is that the colors must be bold and the following buttons must appear on the screen.
  - Order Inventory
  - Enter Sales
  - Schedule Employees
  - Tax Reports
  - Sales Reports
  - Employee Vacation and Sick Time
  - Administrative Activities

Create two different potential GUI screen designs for the main system. Provide a brief explanation of the advantages and disadvantages of each design.

3. **UNDERSTANDING END-USER DEVELOPMENT** The advantages and disadvantages of end-user development and of outsourcing are covered in this chapter. Compile a list of the different advantages and disadvantages of end-user development compared to outsourcing.
4. **MANAGING THE PROJECT PLAN** You are in the middle of an interview for your first job. The manager performing the interview asks you to explain why managing a project plan is critical to a project's success. The manager also wants to know what scope creep and feature creep are and how you would manage them during a project. Please write a one-page document stating how you would answer these questions during the interview.
5. **WHY PROTOTYPE?** You are in the middle of the design phase for a new system. Your manager does not understand why it's important to develop a prototype of a proposed system before building the actual system. In a one-page document, explain what potential problems would arise if you didn't develop a prototype and went straight to developing the system.
6. **BUSINESS REQUIREMENTS** Gathering accurate and complete business requirements is critical to the successful development of any system. Review the following requirements and explain any problems they might have.
  - The GUI must be red.
  - There should be three buttons labeled "Start" and "Stop."
  - Buttons 1 through 8 are required for the calculator function.
  - There should be a text field for the user name and a button for the user password.
7. **CONSTRUCTION AND THE SDLC** The systems development life cycle is often compared to the activities in the construction industry. Fill in the following chart listing some of the activities performed in building a house and how they relate to the different SDLC steps.

SDLC	Activities for Building a Home
Planning	
Analysis	
Design	
Development	
Testing	
Implementation	
Maintenance	

## Discussion Questions

1. Why is it important to develop a logical model of a proposed system before generating a technical architecture? What potential problems would arise if you didn't develop a logical model and went straight to developing the technical design?
2. If you view systems development as a question-and-answer session, another question you could ask is, "Why do organizations develop IT systems?" Consider what you believe to be the five most important reasons organizations develop IT systems. How do these reasons relate to topics in the first five chapters of this book?
3. When deciding how to staff a systems development project, what are some of the primary questions you must be able to answer to determine if you will outsource? What are some of the advantages and disadvantages of outsourcing versus insourcing?
4. Your company has just decided to implement a new financial system. Your company's financial

- needs are almost the same as those of all the other companies in your industry. Would you recommend that your company purchase an existing system or build a custom system? Would you recommend your company use end-user development or outsource the new system?
5. Why do you think system documentation is important? If you had to write system documentation for a word processing application, what would be a few of the main sections? Do you think it would be useful to test the documentation to ensure it's accurate? What do you think happens if you provide system documentation that is inaccurate? What do you think happens if you implement a new system without documentation?
  6. What would happen to an organization that refused to follow the systems development life cycle when building systems? If you worked for such an organization, what would you do to convince your manager to follow the systems development life cycle?
  7. There are seven phases in the systems development life cycle. Which one do you think is the hardest? Which one do you think is the easiest? Which one do you think is the most important? Which one do you think is the least important? If you had to skip one of the phases, which one would it be and why?
  8. What would happen to a systems development effort that decided to skip the testing phase? If you were working on this project, what would you do to convince your team members to perform the activities in the testing phase?
  9. If you were working on a large systems development project, and after reviewing the business requirements you were positive several were missing, would you still sign off on the requirements? If you did, what would happen?
  10. You are talking with another student who is complaining about having to learn about the systems development life cycle, because he is not going to work in an IT department. Would you agree with this student? What would you say to him to convince him that learning about the systems development life cycle is important no matter where he works?
  11. A company typically has many systems it wants to build, but unfortunately it usually doesn't have the resources to build all the systems. How does a company decide which systems to build?
  12. When you start working on a new system, one of your first activities is to define the project scope. Do you think this is an easy activity? Why is the project scope so important? Do you think everyone on the project should know the project's scope? What could happen if people on the project are not familiar with the project scope?
  13. People often think that a system is complete once it is implemented. Is this true? What happens after a system is implemented? What can you do to ensure the system continues to meet the knowledge workers' needs?
  14. Imagine your friends are about to start their own business and they have asked you for planning and development advice with respect to IT systems. What would you tell them? What if their business idea was completely Internet-based? What if their business idea didn't include using the Internet at all? Would your answers differ? Why or why not?

## → Finding Freeware and Shareware on the Internet

Even upgrading to the latest version of your existing software can make a real dent in your pocketbook. And after installing new software, you may find it simply doesn't meet your needs. That's when you notice that you can't return opened software; you can only exchange it for a new copy. Therefore, if it doesn't meet your needs you're out of luck with commercial consumer software.

An alternative to commercial software that you might consider is shareware or freeware. Shareware is sometimes called "try before you buy" software because users are permitted to try the software on their own computer system (generally for a limited time) without any cost or obligation. Then you make a payment if you decide you want to keep using the software beyond the evaluation (trial) period. Freeware is software available at no charge to users for as long as they choose to use the software.

### USING YOUR COMPUTER FOR MORE THAN WORK

By far the most popular freeware/shareware applications are games. The quality of these software titles is truly amazing for software that is free to download and begin playing immediately whenever you want. Shareware/freeware games are so numerous on the Internet that you'll often find games grouped by categories. Common categories are action/adventure, board, card, casino, educational, role-playing, simulation, sports, strategy and war, and word games.

Connect to the Internet and several sites that offer freeware and shareware games. Pick at least two games and download them. For each, answer the following questions:

- A. Is a description of the game provided?
- B. Are system requirements listed?
- C. Can you tell if the game is freeware or shareware without downloading it?
- D. Are any of the games you selected really commercial software that requires a purchase before you download the game?
- E. If the game is shareware, how long are you permitted to use it until registration is required?
- F. If the game is shareware, does the game cease to function after the free period is over?
- G. How can you tell without waiting that long?
- H. How long does it take to download the game?
- I. Is it worth it?

### PROTECTING YOUR COMPUTER INVESTMENT

Have you ever been frantically typing away, desperately trying to make an assignment deadline, when all of a sudden something goes wrong with your computer? If you're lucky, the problem is something easy to identify, so you correct the problem and go on about your work. At other times the solution eludes you. Most of the time these problems have nice logical explanations such as hardware or software conflicts or failures of some kind. In a few rare instances, the problem may have been caused intentionally by a computer virus, a program that someone develops with malicious intent to harm an IT system.

So how does a computer virus get into your system? Anytime you download software, open a file attachment to an e-mail, or read a file off a diskette from another computer, you stand the chance of contracting a computer virus. And access to the Internet increases your opportunity to download files from many different sources. So on every one of its computers virtually every company installs anti-virus protection software that scans new files for known viruses and purges them from the system. The catch is that traditional anti-virus software can find only viruses that it knows about. As new viruses come along, anti-virus software must be updated. The deviant minds that develop viruses seem to find more and better ways to infiltrate your system every day.

Connect to a site that allows you to download anti-virus software, download the software, and answer the following questions:

- A. Is the anti-virus software shareware, freeware, or traditional retail software?
- B. What viruses does the software detect?
- C. Does the software remove the virus as well as detect it?
- D. Are updates for the software available to detect new viruses? How often are they available? At what cost?
- E. Does the software detect viruses not yet created? How does it do that?
- F. Does the software site offer recommendations to reduce your chance of contracting a virus?
- G. Does the site tell you what to do if you have already contracted a virus?

### SEARCHING FOR SHAREWARE AND FREWARE

So maybe the shareware/freeware software concept appeals to you. You'd like to be able to try the software before you buy. If you want software such as screen savers or anti-virus software, you're in luck. But what if you want some shareware to help you compose music or keep track of your soccer team's schedule? Well, then you'll have to go searching for that software. You could use a general-purpose search engine such as Yahoo! and type in shareware and music or soccer. If you do this you will find a few shareware software titles to download. But suppose those few titles don't meet your needs.

Finding shareware/freeware titles can be daunting for two reasons. First, currently there are over 1 million shareware and freeware titles available to you. Unless a search engine is designed specifically for this type of software, you'll probably miss many of these titles using a general-purpose search engine. Second, most shareware/freeware developers don't have their own Web sites. As many don't develop their software as a business, they can't justify the cost of supporting their own Web sites. To address both of these challenges, Web sites have been created that maintain databases of thousands of shareware/freeware software titles. Most also include a search engine to help you navigate through these thousands of titles.

Find a site that maintains a database of freeware and shareware software. As you peruse it, answer the following questions:

- A. How does the site group the software?
- B. Can you search by operating system or platform?
- C. Does the site provide descriptions of the software?
- D. Can you search by file size?
- E. Are screen captures from the software provided?
- F. Are reviews and/or ratings of the software provided?
- G. When was the last update for the site?

