Part One

The Context of Systems Development Projects

This is a practical book about information systems development methods. All businesses and organizations develop information systems. You can be assured that you will play some role in the systems analysis and design for those systems—either as a customer or user of those systems or as a developer of those systems. Systems analysis and design is about business problem solving and computer applications. The methods you will learn in this book can be applied to a wide variety of problem domains, not just those involving the computer.

Before we begin, we assume you've completed an introductory course in computer-based information systems. Many of you have also completed one or more programming courses (using technologies such as *Access, Java, C/C++*, or *Visual Basic*). That will prove helpful, since systems analysis and design precedes and/or integrates with those activities. But don't worry—we'll review all the necessary principles on which systems analysis and design is based.

Part One focuses on the big picture. Before you learn about specific activities, tools, techniques, methods, and technology, you need to understand this big picture. As you explore the context of systems analysis and design, we will introduce many ideas, tools, and techniques that are not explored in great detail until later in the book. Try to keep that in mind as you explore the big picture.

Systems development isn't magic. There are no secrets for success, no perfect tools, techniques, or methods. To be sure, there are skills that can be mastered. But the complete and consistent application of those skills is still an art.

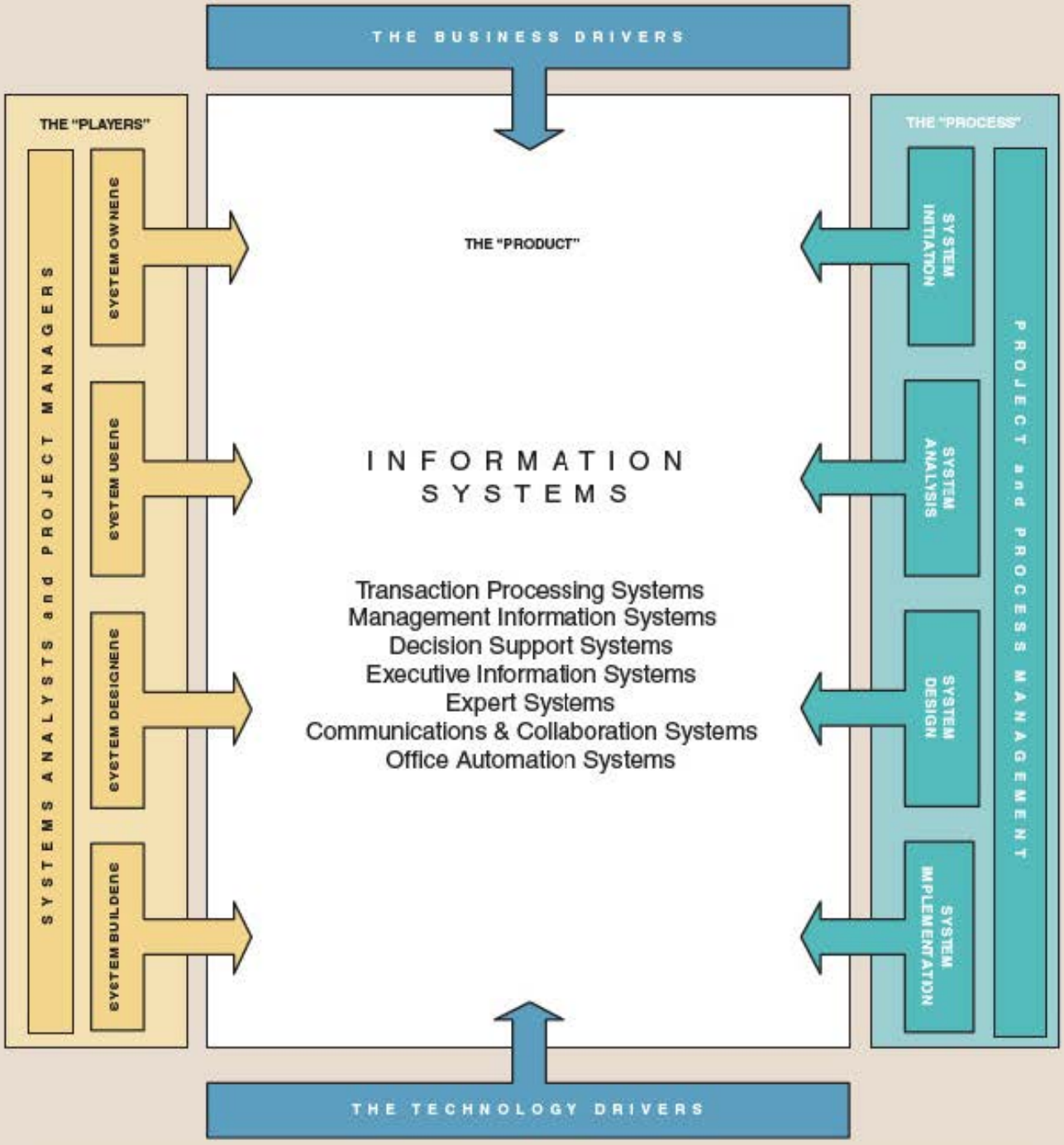
We start in Part One with fundamental concepts, philosophies, and trends that provide the context of systems analysis and design methods—in other words, the basics! If you understand these basics, you will be better able to apply, with confidence, the practical tools and techniques you will learn in Parts Two through Four. You will also be able to adapt to new situations and methods.

Four chapters make up this part. Chapter 1, "The Context of Systems Analysis and Design Methods," introduces you to the *participants* in systems analysis and design with special emphasis on the modern systems analyst as the facilitator of systems work. You'll also learn about the relationships between systems analysts, end users, managers, and other information systems professionals. Finally, you'll learn to prepare yourself for a career as an analyst (if that is your goal). Regardless, you will understand how you will interact with this important professional.

Chapter 2, "Information System Building Blocks," introduces the *product* we will teach you how to build—*information systems*. Specifically, you will learn to examine information systems in terms of common building blocks, KNOWLEDGE, PROCESSES, and COMMUNICATIONS—each from the perspective of different participants or stakeholders. A visual matrix framework will help you organize these building blocks so that you can see them applied in the subsequent chapters.

Chapter 3, "Information Systems Development," introduces a high-level (meaning general) process for information systems development. This is called a *systems development life cycle*. We will present the life cycle in a form in which most of you will experience it—a *systems development methodology*. This methodology will be the context in which you will learn to use and apply the systems analysis and design methods taught in the remainder of the book.

Chapter 4, "Project Management," introduces project management techniques. All systems projects are dependent on the principles that are surveyed. This chapter introduces two modeling techniques for project management: *Gantt* and *PERT*. These tools help you schedule activities, evaluate progress, and adjust schedules.



CHAPTER 1 HOME PAGE Each chapter in this book begins with a "home page" similar to the one above. The home page is something of a chapter map, a visual framework for systems thinking applicable to that chapter. Chapter 1 focuses on (1) the players in the systems game, (2) business drivers of interest to business players, (3) technology drivers and enablers of interest to the technical players, and (4) the process used to develop systems. We will also examine the critical role played by systems analysts in facilitating an understanding of how all four perspectives must come together.

1 The Context of Systems Analysis and Design Methods

Chapter Preview and Objectives

This is a book about systems analysis and design as applied to information systems and computer applications. No matter what your chosen occupation or position in any business, you will likely participate in systems analysis and design. Some of you will become *systems analysts,* the key players in systems analysis and design activities. The rest of you will work *with* systems analysts as projects come and go in your organizations. This chapter introduces you to information systems from four different perspectives. You will understand the context for systems analysis and design methods when you can:

- Define *information system* and name seven types of information system applications.
- Identify different types of *stakeholders* who use or develop information systems, and give examples of each.
- Define the unique role of *systems analysts* in the development of information systems.
- Identify those *skills* needed to successfully function as an information systems analyst.
- Describe current *business drivers* that influence information systems development.
- Describe current *technology drivers* that influence information systems development.
- Briefly describe a simple *process* for developing information systems.

Introduction

It is Bob Martinez's first week at work as an analyst/programmer. Fresh out of college with a degree in computer information systems technology, Bob is eager to work with information systems in the real world. His employer is SoundStage Entertainment Club, one of the fastest-growing music and video clubs in America. SoundStage is just beginning systems analysis and design work on a reengineering of their member services information system. Bob has been appointed to the project team.

This morning was the kickoff meeting for the project, a meeting that included the vice president of member services, director of the audio club, director of the game club, director of marketing, director of customer services, and director of warehouse operations. With that lineup Bob was glad to mainly keep silent at the meeting and rely on his boss, Sandra Shepherd, a senior systems analyst. He was amazed at how well Sandra was able to speak the language of each of the participants and to explain the plans for the new information system in terms they could understand and with benefits they could appreciate. Bob had thought that being just out of college he would know more about cutting-edge technology than most of his co-workers. But Sandra seemed to understand everything about e-commerce and using mobile technologies plus many things of which Bob was only vaguely aware. He made a note to read up on ERP systems as that had come up in the discussion. By the end of the meeting Bob had a new appreciation for the job of systems analyst and of all the things he had yet to learn.

system a group of interrelated components that function together to achieve a desired result.

A Framework for Systems Analysis and Design

information system (IS) an arrangement of people, data, processes, and **information technology** that interact to collect, process, store, and provide as output the information needed to support an organization.

information technology (IT) a contemporary term that describes the combination of computer technology (hardware and software) with telecommunications technology (data, image, and voice networks).

transaction processing system (TPS) an information system that captures and processes data about business transactions.

management information system (MIS) an information system that provides for management-oriented reporting based on transaction processing and operations of the organization.

As its title suggests, this is a book about *systems analysis and design methods*. In this chapter, we will introduce the subject using a simple but comprehensive visual framework. Each chapter in this book begins with a *home page* (see page 4) that quickly and visually shows which aspects of the total framework we will be discussing in the chapter. We'll build this visual framework slowly over the first four chapters to avoid overwhelming you with too much detail too early. Thereafter, each chapter will highlight those aspects of the full framework that are being taught in greater detail in that chapter.

Ultimately, this is a book about "analyzing" business requirements for information systems and "designing" *information systems* that fulfill those business requirements. In other words, the *product* of systems analysis and design is an information system. That product is visually represented in the visual framework as the large rectangle in the center of the picture.

A **system** is a group of interrelated components that function together to achieve a desired result. For instance, you may own a home theater system made up of a DVD player, receiver, speakers, and display monitor.

Information systems (IS) in organizations capture and manage data to produce useful information that supports an organization and its employees, customers, suppliers, and partners. Many organizations consider information systems to be essential to their ability to compete or gain competitive advantage. Most organizations have come to realize that *all* workers need to participate in the development of information systems. Therefore, information systems development is a relevant subject to you regardless of whether or not you are studying to become an information systems professional.

Information systems come in all shapes and sizes. They are so interwoven into the fabric of the business systems they support that it is often difficult to distinguish between business systems and their support information systems. Suffice it to say that information systems can be classified according to the functions they serve. **Transaction processing systems (TPSs)** process business transactions such as orders, time cards, payments, and reservations. **Management information systems (MISs)** use the transaction data to produce information needed by managers to run the business.

Decision support systems (DSSs) help various decision makers identify and choose between options or decisions. **Executive information systems (EISs)** are tailored to the unique information needs of executives who plan for the business and assess performance against those plans. **Expert systems** capture and reproduce the knowledge of an expert problem solver or decision maker and then simulate the "thinking" of that expert. **Communication and collaboration systems** enhance communication and collaboration between people, both internal and external to the organization. Finally, **office automation systems** help employees create and share documents that support day-to-day office activities.

decision support system (DSS) an information system that either helps to identify decision-making opportunities or provides information to help make decisions.

executive information system (EIS) an information system that supports the planning and assessment needs of executive managers.

As illustrated in the chapter home page, information systems can be viewed from various perspectives, including:

- The players in the information system (the "team").
- The business drivers influencing the information system.
- The technology drivers used by the information system.
- The process used to develop the information system.

Let's examine each of these perspectives in the remaining sections of the chapter.

The Players—System Stakeholders

Let's assume you are in a position to help build an information system. Who are the **stakeholders** in this system? Stakeholders for information systems can be broadly classified into the five groups shown on the left-hand side of Figure 1-1. Notice that each stakeholder group has a different perspective of the same information system. The *systems analyst* is a unique stakeholder in Figure 1-1. The systems analyst serves as a facilitator or coach, bridging the communications gap that can naturally develop between the nontechnical system owners and users and the technical system designers and builders.

expert system an information system that captures the expertise of workers and then simulates that expertise to the benefit of nonexperts.

All the above stakeholders have one thing in common—they are what the U.S. Department of Labor calls **information workers.** The livelihoods of information workers depend on decisions made based on information. Today, more than 60 percent of the U.S. labor force is involved in producing, distributing, and using information. Let's examine the five groups of information workers in greater detail.

communications and collaboration system an information system that enables more effective communications between workers, partners, customers, and suppliers to enhance their ability to collaborate.

Let's briefly examine the perspectives of each group. But before we do so, we should point out that these groups actually define "roles" played in systems development. In practice, any individual person may play more than one of these roles. For example, a system owner might also be a system user. Similarly, a systems analyst may also be a system designer, and a system designer might also be a system builder. Any combination may work.

office automation system an information system that supports the wide range of business office activities that provide for improved work flow between workers.

> Systems Owners

For any information system, large or small, there will be one or more **system owners.** System owners usually come from the ranks of management. For medium to large information systems, system owners are usually middle or executive managers. For smaller systems, system owners may be middle managers or supervisors. System owners tend to be interested in the bottom line—how much will the system cost? How much value or what benefits will the system return to the business? Value and benefits can be measured in different ways, as noted in the margin checklist.

stakeholder any person who has an interest in an existing or proposed information system. Stakeholders may include both technical and nontechnical workers. They may also include both internal and external workers.

> Systems Users

System users make up the vast majority of the information workers in any information system. Unlike system owners, system users tend to be less concerned with costs and benefits of the system. Instead, as illustrated in Figure 1-1, they are concerned with the functionality the system provides to their jobs and the system's ease of learning and ease of use. Although users have become more technology-literate over the years,

information worker any person whose job involves creating, collecting, processing, distributing, and using information.

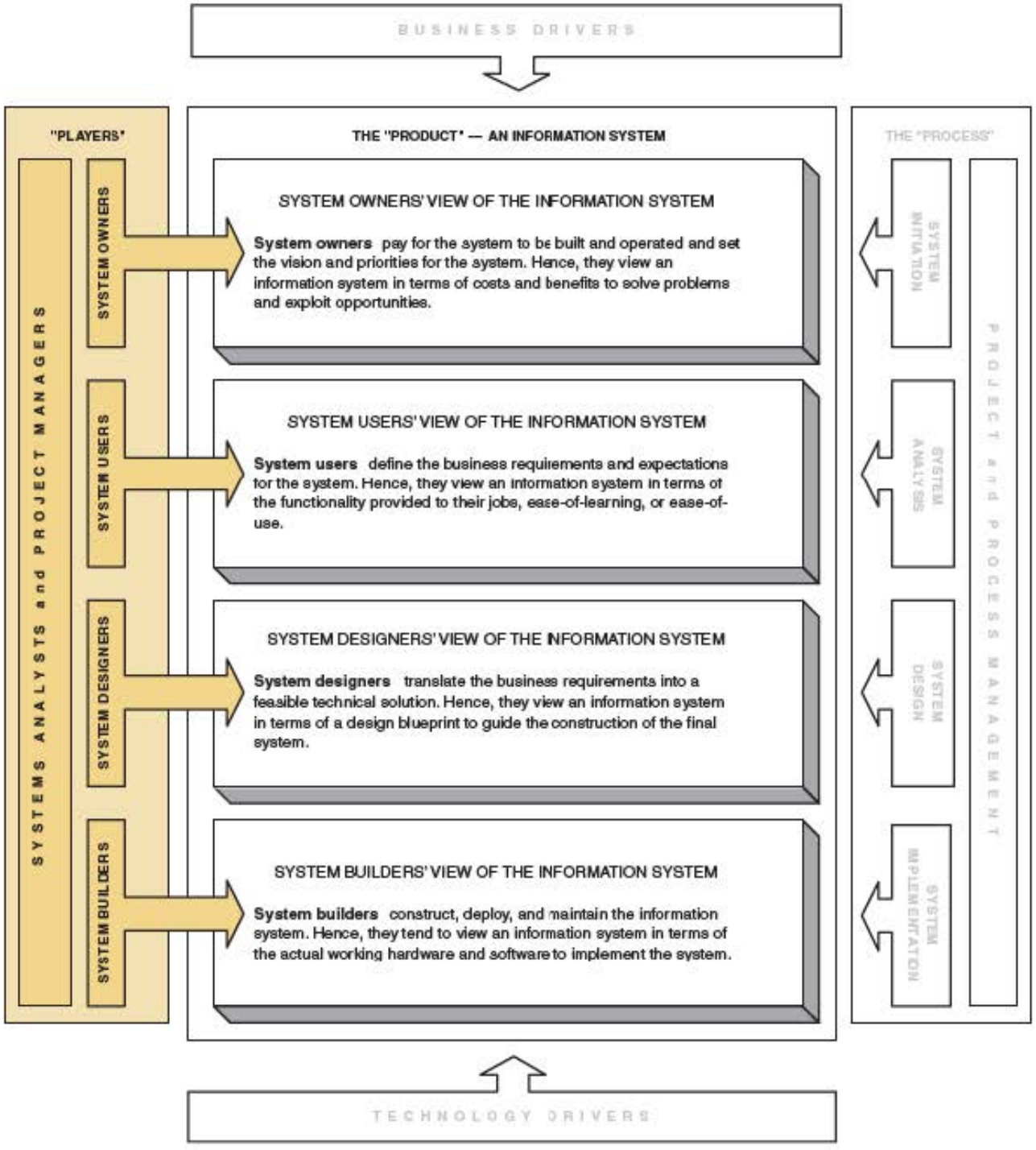


FIGURE 1-1 Stakeholders' Perspective of an Information System

system owner an information system's sponsor and executive advocate, usually responsible for funding the project of developing, operating, and maintaining the information system.

their primary concern is to get the job done. Consequently, discussions with most users need to be kept at the business requirements level as opposed to the technical requirements level. Much of this book is dedicated to teaching you how to effectively identify and communicate business requirements for an information system.

There are many classes of system users. Each class should be directly involved in any information system development project that affects them. Let's briefly examine these classes.

Internal System Users Internal system users are employees of the businesses for which most information systems are built. Internal users make up the largest percentage of information system users in most businesses. Examples include:

- *Clerical and service workers*—perform most of the day-to-day transaction processing in the average business. They process orders, invoices, payments, and the like. They type and file correspondence. They fill orders in the warehouse. And they manufacture goods on the shop floor. Most of the fundamental data in any business is captured or created by these workers, many of whom perform manual labor in addition to processing data. Information systems that target these workers tend to focus on transaction processing speed and accuracy.
- *Technical and professional staff*—consists largely of business and industrial specialists who perform highly skilled and specialized work. Examples include lawyers, accountants, engineers, scientists, market analysts, advertising designers, and statisticians. Because their work is based on well-defined bodies of knowledge, they are sometimes called **knowledge workers.** Information systems that target technical and professional staff focus on data analysis as well as generating timely information for problem solving.
- *Supervisors, middle managers, and executive managers*—are the decision makers. Supervisors tend to focus on day-to-day problem solving and decision making. Middle managers are more concerned with tactical (short-term) operational problems and decision making. Executive managers are concerned with strategic (long-term) planning and decision making. Information systems for managers tend to focus entirely on information access. Managers need the right information at the right time to identify and solve problems and make good decisions.

External System Users The Internet has allowed traditional information system boundaries to be extended to include other businesses or direct consumers as system users. These external system users make up an increasingly large percentage of system users for modern information systems. Examples include:

- *Customers*—any organizations or individuals that purchase our products and services. Today, our customers can become direct users of our information systems when they can directly execute orders and sales transactions that used to require intervention by an internal user. For example, if you purchased a company's product via the Internet, you became an external user of that business's sales information system. (There was no need for a separate internal user of the business to input your order.)
- *Suppliers*—any organizations from which our company may purchase supplies and raw materials. Today, these suppliers can interact directly with our company's information systems to determine our supply needs and automatically create orders to fill those needs. There is no longer always a need for an internal user to initiate those orders to a supplier.
- *Partners*—any organizations from which our company purchases services or with which it partners. Most modern businesses contract or outsource a number of basic services such as grounds maintenance, network management, and many others. And businesses have learned to partner with other businesses to more quickly leverage strengths to build better products more rapidly.
- *Employees*—those employees who work on the road or who work from home. For example, sales representatives usually spend much of their time on the road. Also, many businesses permit workers to telecommute (meaning "work from home") to reduce costs and improve productivity. As mobile or remote users, these employees require access to the same information systems as those needed by internal users.

POSSIBLE VALUES AND BENEFITS OF INFORMATION SYSTEMS

Increased Business Profit
Reduced Business Costs
Costs and Benefits of the System
Increased Market Share
Improved Customer Relations
Increased Efficiency
Improved Decision Making
Better Compliance with Regulations
Fewer Mistakes
Improved Security
Greater Capacity

system user a "customer" who will use or is affected by an information system on a regular basis—capturing, validating, entering, responding to, storing, and exchanging data and information.

knowledge worker any worker whose responsibilities are based on a specialized body of knowledge.

remote user a user who is not physically located on the premises but who still requires access to information systems.

mobile user a user whose location is constantly changing but who requires access to information systems from any location.

External system users are increasingly referred to as **remote users** and **mobile users.** They connect to our information systems through laptop computers, handheld computers, and smart phones—either wired or wireless. Designing information systems for these devices presents some of the most contemporary of challenges that we will address in this book.

> Systems Designers

system designer a technical specialist who translates system users' business requirements and constraints into technical solutions. She or he designs the computer databases, inputs, outputs, screens, networks, and software that will meet the system users' requirements.

System designers are technology specialists for information systems. As Figure 1-1 shows, system designers are interested in information technology choices and in the design of systems that use chosen technologies. Today's system designers tend to focus on technical specialties. Some of you may be educating yourselves to specialize in one of these technical specialties, such as:

- *Database administrators*—specialists in database technologies who design and coordinate changes to corporate databases.
- *Network architects*—specialists in networking and telecommunications technologies who design, install, configure, optimize, and support local and wide area networks, including connections to the Internet and other external networks.
- *Web architects*—specialists who design complex Web sites for organizations, including public Web sites for the Internet, internal Web sites for organizations (called *intranets*), and private business-to-business Web sites (called *extranets*).
- *Graphic artists*—relatively new in today's IT worker mix, specialists in graphics technology and methods used to design and construct compelling and easy-to-use interfaces to systems, including interfaces for PCs, the Web, handhelds, and smart phones.
- *Security experts*—specialists in the technology and methods used to ensure data and network security (and privacy).
- *Technology specialists*—experts in the application of specific technologies that will be used in a system (e.g., a specific commercial software package or a specific type of hardware).

> Systems Builders

system builder a technical specialist who constructs information systems and components based on the design specifications generated by the system designers.

System builders (again, see Figure 1-1) are another category of technology specialists for information systems. Their role is to construct the system according to the system designers' specifications. In small organizations or with small information systems, systems designers and systems builders are often the same people. But in large organizations and information systems they are often separate jobs. Some of you may be educating yourselves to specialize in one of their technical specialties, such as:

- *Applications programmers*—specialists who convert business requirements and statements of problems and procedures into computer languages. They develop and test computer programs to capture and store data and to locate and retrieve data for computer applications.
- *Systems programmers*—specialists who develop, test, and implement operating systems–level software, utilities, and services. Increasingly, they also develop reusable software "components" for use by applications programmers (above).
- *Database programmers*—specialists in database languages and technology who build, modify, and test database structures and the programs that use and maintain them.
- *Network administrators*—specialists who design, install, troubleshoot, and optimize computer networks.
- *Security administrators*—specialists who design, implement, troubleshoot, and manage security and privacy controls in a network.

- *Webmasters*—specialists who code and maintain Web servers.
- *Software integrators*—specialists who integrate software packages with hardware, networks, and other software packages.

Although this book is not directly intended to educate the system builder, it is intended to teach system designers how to better communicate design specifications to system builders.

> Systems Analysts

As you have seen, system owners, users, designers, and builders often have very different perspectives on any information system to be built and used. Some are interested in generalities, while others focus on details. Some are nontechnical, while others are very technical. This presents a communications gap that has always existed between those who need computer-based business solutions and those who understand information technology. The **systems analyst** bridges that gap. You can (and probably will) play a role as either a systems analyst or someone who works with systems analysts.

systems analyst a specialist who studies the problems and needs of an organization to determine how people, data, processes, and information technology can best accomplish improvements for the business.

As illustrated in Figure 1-1, their role intentionally overlaps the roles of all the other stakeholders. For the system owners and users, systems analysts identify and validate business problems and needs. For the system designers and builders, systems analysts ensure that the technical solution fulfills the business needs and integrate the technical solution into the business. In other words, systems analysts *facilitate* the development of information systems through interaction with the other stakeholders.

There are several legitimate, but often confusing, variations on the job title we are calling "systems analyst." A *programmer/analyst* (or *analyst/programmer*) includes the responsibilities of both the computer programmer and the systems analyst. A *business analyst* focuses on only the nontechnical aspects of systems analysis and design. Other synonyms for "systems analyst" are systems consultant, business analyst, systems architect, systems engineer, information engineer, information analyst, and systems integrator.

Some of you will become systems analysts. The rest of you will routinely work with systems analysts who will help you solve your business and industrial problems by creating and improving your access to the data and information needed to do your job. Let's take a closer look at systems analysts as the key facilitators of information systems development.

The Role of the Systems Analyst Systems analysts understand both business and computing. They study business problems and opportunities and then transform business and information requirements into specifications for information systems that will be implemented by various technical specialists including computer programmers. Computers and information systems are of value to a business only if they help solve problems or effect improvements.

Systems analysts initiate *change* within an organization. Every new system changes the business. Increasingly, the very best systems analysts literally change their organizations—providing information that can be used for competitive advantage, finding new markets and services, and even dramatically changing and improving the way the organization does business.

The systems analyst is basically a *problem solver*. Throughout this book, the term *problem* will be used to describe many situations, including:

- Problems, either real or anticipated, that require corrective action.
- Opportunities to improve a situation despite the absence of complaints.
- Directives to change a situation regardless of whether anyone has complained about the current situation.

The systems analyst's job presents a fascinating and exciting challenge to many individuals. It offers high management visibility and opportunities for important

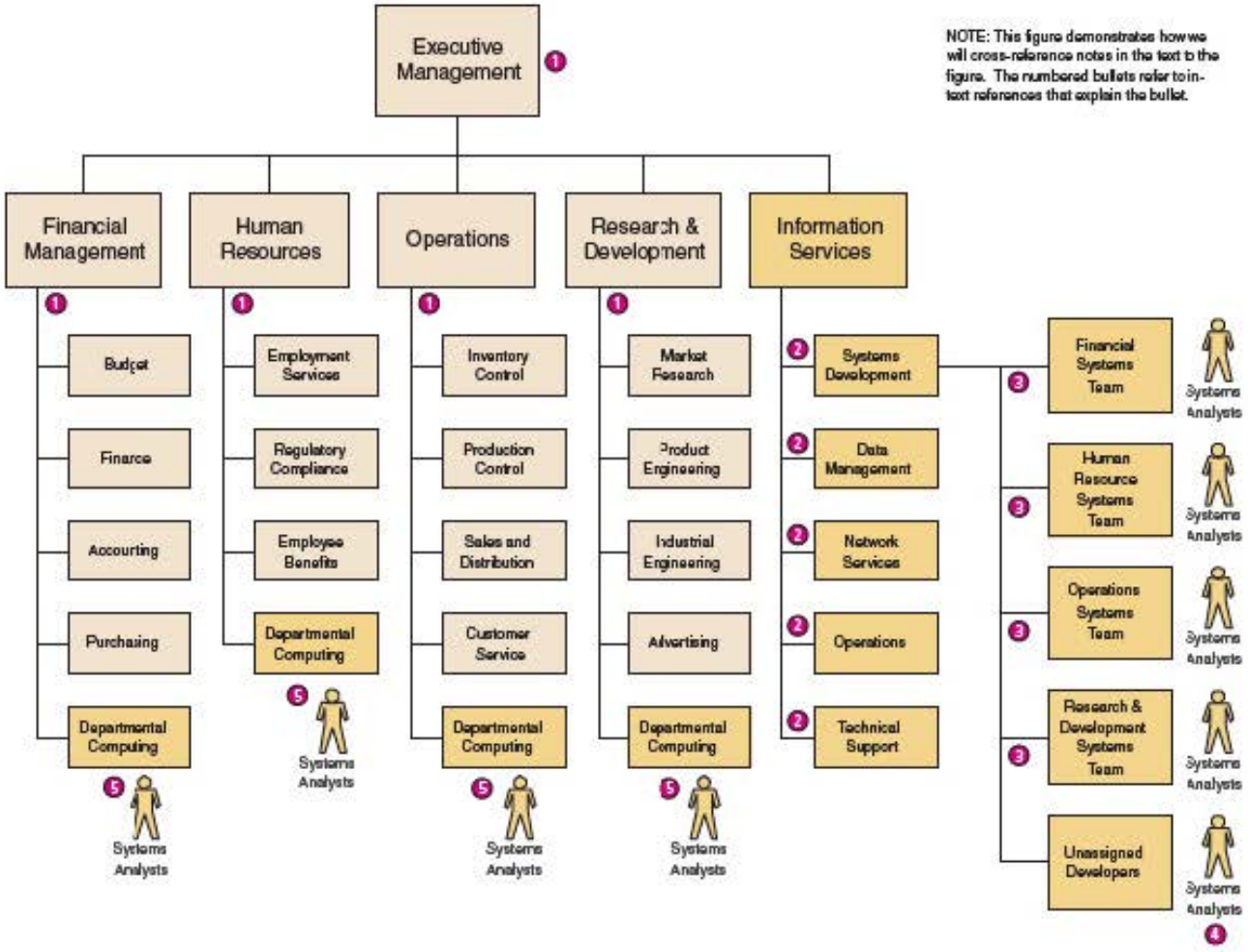


FIGURE 1-2 Systems Analysts in a Typical Organization

decision making and creativity that may affect an entire organization. Furthermore, this job can offer these benefits relatively early in your career (compared to other entry-level jobs and careers).

Where Do Systems Analysts Work? Every business organizes itself uniquely. But certain patterns of organization seem to reoccur. Figure 1-2 is a representative organization chart. The following numbered bullets cross-reference and emphasize key points in the figure:

1. System owners and system users are located in the functional units and subunits of the business, as well as in the executive management.
2. System designers and builders are usually located in the information systems unit of the business. Most systems analysts work also for the information services unit of an organization.
3. As shown in the figure, systems analysts (along with systems designers and builders) may be permanently assigned to a team that supports a specific business function (e.g., financial systems).

Numbers 2 and 3 above represent a traditional approach to organizing systems analysts and other developers. Numbers 4 and 5 below represent strategies intended to emphasize either efficiency or business expertise. All of the strategies can be combined in a single organization.

The Next Generation: Career Prospects for Systems Analysts

Next Generations

Many of you are considering or preparing for a career as a systems analyst. The life of a systems analyst is both challenging and rewarding. But what are the prospects for the future? Do organizations need systems analysts? Will they need them in the foreseeable future? Is the job changing for the future, and if so, how? These questions are addressed in this box.

According to the U.S. Department of Labor, computer-related jobs account for 5 out of the 20 fastest-growing occupations in the economy. What's more, these fastest-growing computer-related occupations pay better than many other jobs.

In 2002, 468,000 workers were classified as systems analysts. By 2012, that number will grow to 653,000, an increase of 39%. This means that at least 185,000 new systems analysts must be educated and hired (not including those needed to replace the ones who retire or move into managerial positions or other occupations). The need is increasing because industry needs systems analysts to meet the seemingly endless demand for more information systems and software applications. As some programming jobs are being out-sourced to independent contractors and other countries, the need grows even greater for skilled systems analysts, who can create solid design specifications for remote development teams. Opportunities for success will be the greatest for the most educated, qualified, skilled, and experienced analysts.

What happens to the successful systems analyst? Does a position as a systems analyst lead to any other careers? Indeed, there are many career paths. Some analysts leave the information systems field and join the user community. Their experience with developing business applications, combined with their total systems perspective, can make experienced analysts unique business specialists. Alternatively, analysts can become project managers, information systems managers, or technical specialists (for databases, telecommunications, microcomputers, and so forth). Finally, skilled systems analysts are often recruited by the consulting and outsourcing industries. The career path opportunities are virtually limitless.

As with any profession, systems analysts can expect change. While it is always dangerous to predict changes, we'll take a shot at it. We believe that organizations will become increasingly dependent on external sources for their systems analysts—consultants and outsourcers. This will be driven by such factors as the complexity and rapid change of technology, the desire to accelerate systems development, and the continued difficulty in recruiting, retaining, and retraining skilled systems analysts (and other information technology professionals). In many cases, internally employed systems analysts will manage projects through consulting or outsourcing agreements.

We believe that an increasing percentage of tomorrow's systems analysts will not work in the information systems department. Instead, they will work directly for a business unit within an organization. This will enable them to better serve their users. It will also give users more power over what systems are built and supported.

Finally, we also believe that a greater percentage of systems analysts will come from noncomputing backgrounds. At one time most analysts were computer specialists. Today's computer graduates are becoming more business-literate. Similarly, today's business and noncomputing graduates are becoming more computer-literate. Their full-time help and insight will be needed to meet demand and to provide the business background necessary for tomorrow's more complex applications.

4. Systems analysts (along with system designers and builders) may also be pooled and temporarily assigned to specific projects for any business function as needed. (Some organizations believe this approach yields greater efficiency because analysts and other developers are always assigned to the highest-priority projects regardless of business area expertise.)
5. Some systems analysts may work for smaller, departmental computing organizations that support and report to their own specific business functions. (Some organizations believe this structure results in systems analysts that develop greater expertise in their assigned business area to complement their technical expertise.)

All of the above strategies can, of course, be reflected within a single organization.

Regardless of where systems analysts are assigned within the organization, it is important to realize that they come together in *project teams*. Project teams are usually created and disbanded as projects come and go. Project teams must also include appropriate representation from the other stakeholders that we previously discussed (system owners, system users, system designers, and system builders). Accordingly, we will emphasize team building and teamwork throughout this book.

Skills Needed by the Systems Analyst For those of you with aspirations of becoming a systems analyst, this section describes the skills you will need to develop. This book introduces many systems analysis and design concepts, tools, and techniques. But you will also need skills and experiences that neither this book nor your systems analysis and design course can fully provide.

When all else fails, the systems analyst who remembers the basic concepts and principles of "systems thinking" will still succeed. No tool, technique, process, or methodology is perfect in all situations! But concepts and principles of systems thinking will always help you adapt to new and different situations. This book emphasizes systems thinking.

Not too long ago, it was thought that the systems analyst's only real tools were paper, pencil, and a flowchart template. Over the years, several tools and techniques have been developed to help the systems analyst. Unfortunately, many books emphasize a specific class of tools that is associated with one methodology or approach to systems analysis and design. In this book, we propose a "toolbox" approach to systems analysis and design. As you read this book, your toolbox will grow to include many tools from different methodologies and approaches to systems analysis and design. Subsequently, you should pick and use tools based on the many different situations you will encounter as an analyst—the right tool for the right job!

In addition to having formal systems analysis and design skills, a systems analyst must develop or possess other skills, knowledge, and traits to complete the job. These include:

- *Working knowledge of information technologies*—The analyst must be aware of both existing and emerging information technologies. Such knowledge can be acquired in college courses, professional development seminars and courses, and in-house corporate training programs. Practicing analysts also stay current through disciplined reading and participation in appropriate professional societies. (To get started, see the Suggested Readings at the end of this and subsequent chapters.)
- *Computer programming experience and expertise*—It is difficult to imagine how systems analysts could adequately prepare business and technical specifications for a programmer if they didn't have some programming experience. Most systems analysts need to be proficient in one or more high-level programming languages.
- *General knowledge of business processes and terminology*—Systems analysts must be able to communicate with business experts to gain an understanding of their problems and needs. For the analyst, at least some of this knowledge comes only by way of experience. At the same time, aspiring analysts should avail themselves of every opportunity to complete basic business literacy courses available in colleges of business. Relevant courses may include financial accounting, management or cost accounting, finance, marketing, manufacturing or operations management, quality management, economics, and business law.
- *General problem-solving skills*—The systems analyst must be able to take a large business problem, break down that problem into its parts, determine problem causes and effects, and then recommend a solution. Analysts must avoid the tendency to suggest the solution before analyzing the problem. For aspiring analysts, many colleges offer philosophy courses that teach

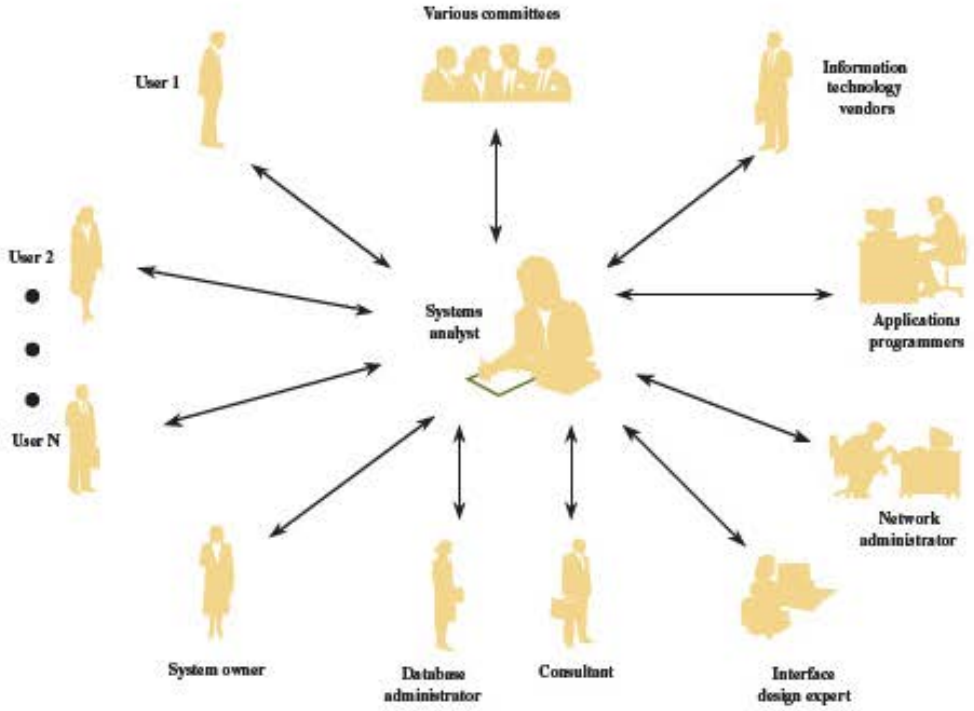


FIGURE 1-3
The Systems Analyst as a Facilitator

problem-solving skills, critical thinking, and reasoning. These "soft skills" will serve an analyst well.

- *Good interpersonal communication skills*—An analyst must be able to communicate effectively, both orally and in writing. Almost without exception, your communications skills, not your technical skills, will prove to be the single biggest factor in your career success or failure. These skills are learnable, but most of us must force ourselves to seek help and work hard to improve them. Most schools offer courses such as business and technical writing, business and technical speaking, interviewing, and listening—all useful skills for the systems analyst. These skills are taught in Chapter 6.
- *Good interpersonal relations skills*—As illustrated in Figure 1-3, systems analysts interact with all stakeholders in a systems development project. These interactions require effective interpersonal skills that enable the analyst to deal with group dynamics, business politics, conflict, and change. Many schools offer valuable interpersonal-skills development courses on subjects such as teamwork, principles of persuasion, managing change and conflict, and leadership.
- *Flexibility and adaptability*—No two projects are alike. Accordingly, there is no single, magical approach or standard that is equally applicable to all projects. Successful systems analysts learn to be flexible and to adapt to unique challenges and situations. Our aforementioned toolbox approach is intended to encourage flexibility in the use of systems analysis and design tools and methods. But you must develop an attitude of adaptability to properly use any box of tools.
- *Character and ethics*—The nature of the systems analyst's job requires a strong character and a sense of right and wrong. Analysts often gain access to sensitive or confidential facts and information that are not meant for public disclosure. Also, the products of systems analysis and design are usually considered the intellectual property of the employer. There are several standards for computer ethics. One such standard, from the Computer Ethics Institute, is called "The Ten Commandments of Computer Ethics" and is shown in Figure 1-4.

FIGURE 1-4 Ethics for Systems Analysts

The Ten Commandments of Computer Ethics

1. Thou shalt not use a computer to harm other people.
2. Thou shalt not interfere with other people's computer work.
3. Thou shalt not snoop around in other people's computer files.
4. Thou shalt not use a computer to steal.
5. Thou shalt not use a computer to bear false witness.
6. Thou shalt not copy or use proprietary software for which you have not paid.
7. Thou shalt not use other people's computer resources without authorization or proper compensation.
8. Thou shalt not appropriate other people's intellectual output.
9. Thou shalt think about the social consequences of the program you are writing or the system you are designing.
10. Thou shalt always use a computer in ways that insure consideration and respect for your fellow humans.

Source: Computer Ethics Institute.

> External Service Providers

external service provider (ESP) a systems analyst, system designer, or system builder who sells his or her expertise and experience to other businesses to help those businesses purchase, develop, or integrate their information systems solutions; may be affiliated with a consulting or services organization.

Those of you with some computing experience may be wondering where consultants fit in our taxonomy of stakeholders. They are not immediately apparent in our visual framework. But they are there! Any of our stakeholder roles may be filled by internal or external workers. Consultants are one example of an **external service provider (ESP).** Most ESPs are systems analysts, designers, or builders who are contracted to bring special expertise or experience to a specific project. Examples include technology engineers, sales engineers, systems consultants, contract programmers, and systems integrators.

> The Project Manager

project manager an experienced professional who accepts responsibility for planning, monitoring, and controlling projects with respect to schedule, budget, deliverables, customer satisfaction, technical standards, and system quality

We've introduced most of the key players in modern information systems development—systems owners, users, designers, builders, and analysts. We should conclude by emphasizing the reality that these individuals must work together as a team to successfully build information systems and applications that will benefit the business. Teams require leadership. For this reason, usually one or more of these stakeholders takes on the role of **project manager** to ensure that systems are developed on time, within budget, and with acceptable quality. As Figure 1-1 indicates, most project managers are experienced systems analysts. But in some organizations, project managers are selected from the ranks of what we have called "system owners." Regardless, most organizations have learned that project management is a specialized role that requires distinctive skills and experience.

Business Drivers for Today's Information Systems

Another way to look at our information system product is from the perspective of business drivers. Using Figure 1-5, let's now briefly examine the most important business trends that are impacting information systems. Many trends quickly become fads, but here are some business trends we believe will influence systems development in the

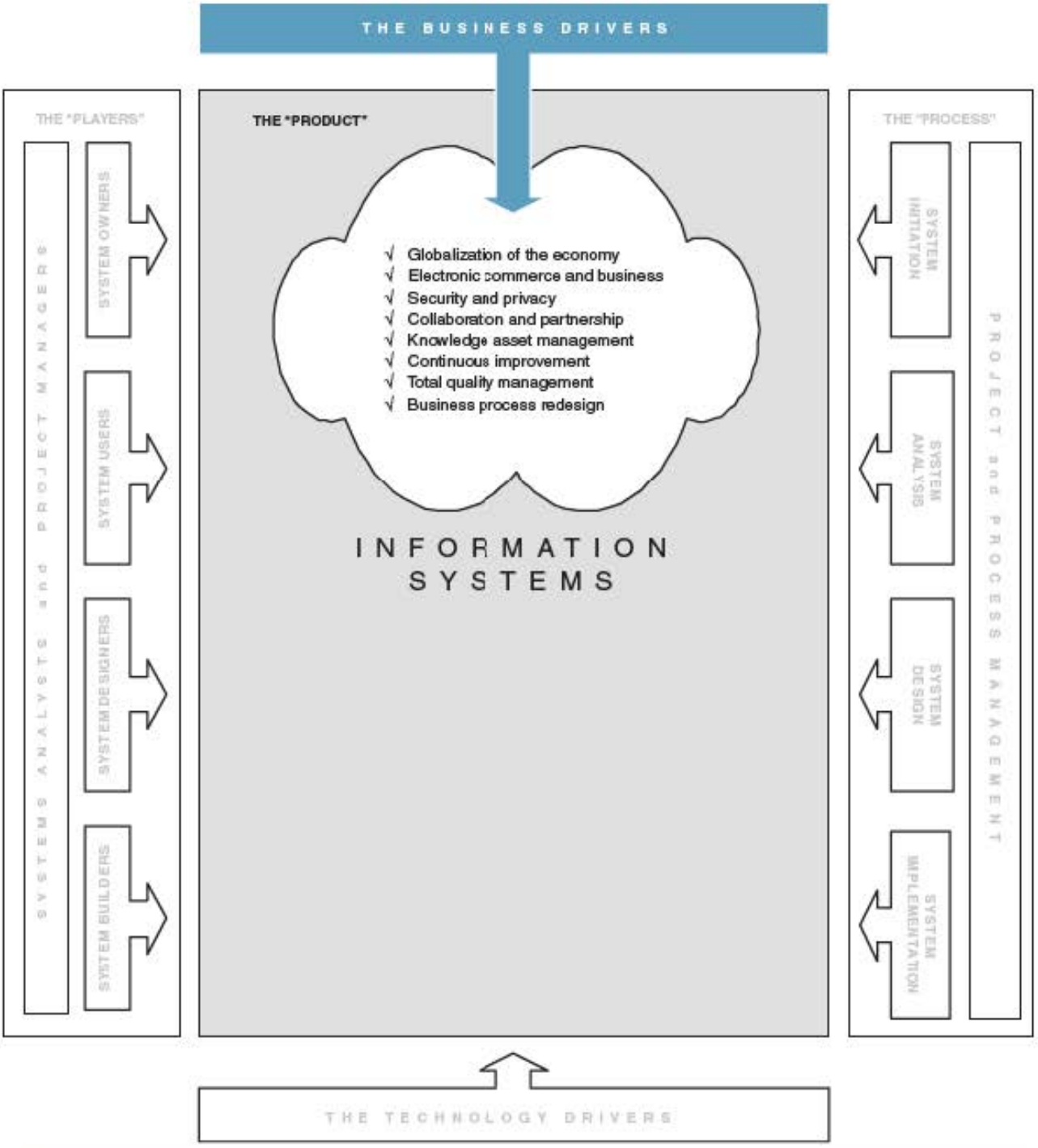


FIGURE 1-5 Business Drivers for an Information System

coming years. Many of these trends are related and integrated such that they form a new business philosophy that will impact the way everyone works in the coming years.

> Globalization of the Economy

Since the 1990s, there has been a significant trend of economic globalization. Competition is global, with emerging industrial nations offering lower-cost or higher-quality

alternatives to many products. American businesses find themselves with new international competitors. On the other hand, many American businesses have discovered new and expanded international markets for their own goods and services. The bottom line is that most businesses were forced to reorganize to operate in this global economy.

How does economic globalization affect the players in the systems game? First, information systems and computer applications must be internationalized. They must support multiple languages, currency exchange rates, international trade regulations, and different business cultures and practices. Second, most information systems ultimately require information consolidation for performance analysis and decision making. The aforementioned language barriers, currency exchange rates, transborder information regulations, and the like, complicate such consolidation. Finally, there exists a demand for players who can communicate, orally and in writing, with management and users that speak different languages, dialects, and slang. Opportunities for international employment of systems analysts should continue to expand.

> Electronic Commerce and Business

electronic commerce (e-commerce) the buying and selling of goods and services by using the Internet.

electronic business (e-business) the use of the Internet to conduct and support day-to-day business activities.

In part due to the globalization of the economy, and in part because of the pervasiveness of the Internet, businesses are changing or expanding their business model to implement **electronic commerce (e-commerce)** and **electronic business (e-business)**. The Internet is fundamentally changing the rules by which business is conducted. We live in a world where consumers and businesses will increasingly expect to conduct commerce (business transactions) using the Internet. But the impact is even more substantive. Because people who work in the business world have become so comfortable with "surfing the Web," organizations are increasingly embracing the Web interface as a suitable architecture for conducting day-to-day business *within* the organization.

There are three basic types of e-commerce- and e-business-enabled information systems applications:

- Marketing of corporate image, products, and services is the simplest form of electronic commerce application. The Web is used merely to "inform" customers about products, services, and policies. Most businesses have achieved this level of electronic commerce.
- *Business-to-consumer (B2C)* electronic commerce attempts to offer new, Web-based channels of distribution for traditional products and services. You, as a typical consumer, can research, order, and pay for products directly via the Internet. Examples include Amazon.com (for books and music) and E-trade.com (for stocks and bonds). Both companies are businesses that were created on the Web. Their competition, however, includes traditional businesses that have added Web-based electronic commerce front ends as an alternative consumer option (such as Barnes and Noble and Merrill Lynch). Figure 1-6 illustrates a typical B2C Web storefront.
- *Business-to-business (B2B)* electronic commerce is the real future. This is the most complex form of electronic commerce and could ultimately evolve into electronic business—the complete, paperless, and digital processing of virtually all business transactions that occur within and between businesses.

One example of B2B electronic commerce is electronic procurement. All businesses purchase raw materials, equipment, and supplies—frequently tens or hundreds of millions of dollars worth per year. B2B procurement allows employees to browse electronic storefronts and catalogs, initiate purchase requisitions and work orders, route requisitions and work orders electronically for expenditure approvals, order the goods and services, and pay for the delivered goods and completed services—all

FIGURE 1-6 An Electronic Commerce Storefront

without the traditional time-consuming and costly paper flow and bureaucracy. Figure 1-7 illustrates a sample Web-based procurement storefront.

Largely due to the trend toward these e-business and e-commerce applications, most new information systems applications are being designed for an Internet architecture. Not that long ago, we were redesigning most applications to operate within a *Windows* user interface. Today, we increasingly see applications designed to run within an Internet browser such as *Internet Explorer* or *Netscape*. The choice of a desktop operating system, such as *Windows, Macintosh,* or *Linux,* is becoming less important than the availability of the browser itself.

> Security and Privacy

As the digital economy continues to evolve, citizens and organizations alike have developed a heightened awareness of the security and privacy issues involved in today's economy. Security issues tend to revolve around business continuity; that is, "How will the business continue in the event of a breach or disaster—any event that causes a disruption of business activity?" Additionally, businesses must ask themselves, "How can the business protect its digital assets from outside threats?" It is true that these questions ultimately come down to technology; however, the concerns have become fundamental business concerns.

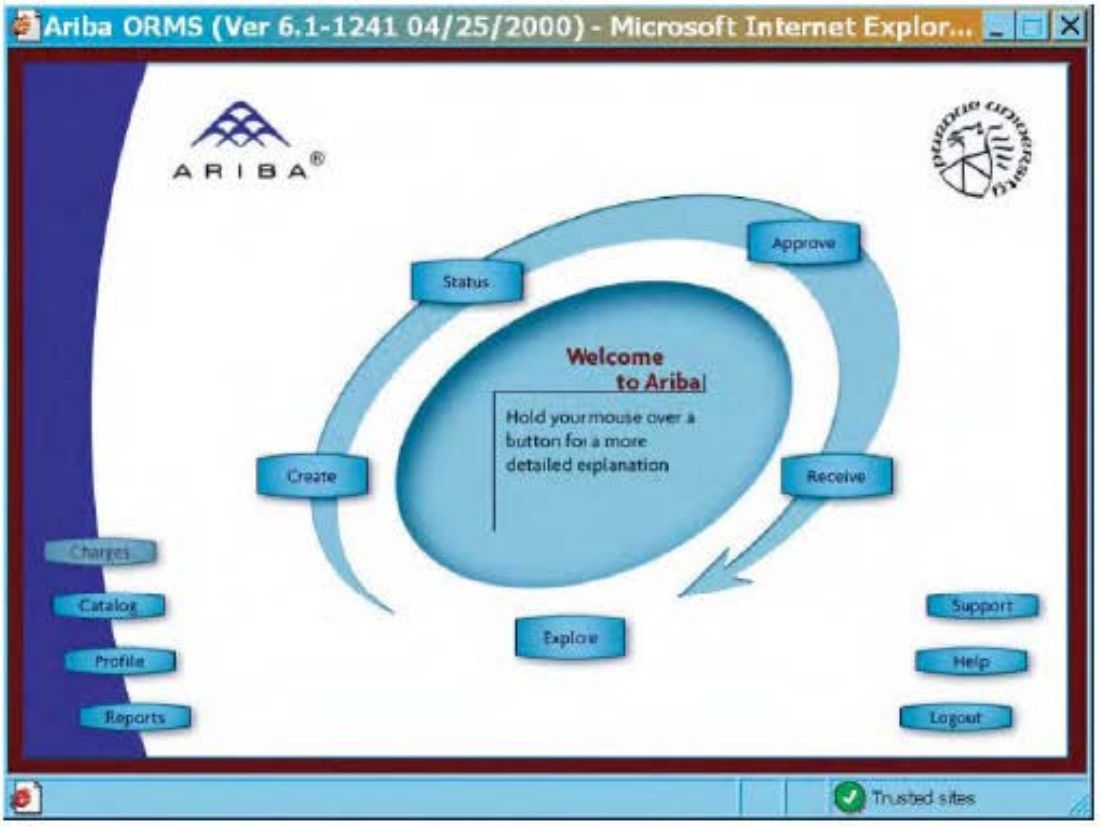


FIGURE 1-7 An Electronic Commerce Procurement Storefront

Related to security is the issue of privacy. Consumers are increasingly demanding privacy in the digital economy. Governments are regulating privacy issues, and the regulations will likely become more stringent as the digital economy continues to evolve. Go to your favorite commercial Web sites. Almost every business now has a privacy policy. Consumer groups are beginning to analyze and monitor such privacy policies, holding companies accountable and lobbying governments for stricter regulations and enforcement.

As information systems are developed and changed, you will increasingly be expected to incorporate more stringent security and privacy controls. In the global economy, you will need to become sensitive to a wide array of regulations that vary considerably from one country to another. Certainly, security and privacy mechanisms will be subject to the same internal audits that have become routine in systems that support or interact with financial systems.

> Collaboration and Partnership

Collaboration and partnership are significant business trends that are influencing information systems applications. Within organizations, management is emphasizing the need to break down the walls that separate organization departments and functions. Management speaks of "cross-functional" teams that collaborate to address common business goals from interdisciplinary perspectives. For example, new product design used to be the exclusive domain of engineers. Today, new product design typically involves a cross-functional team of representatives from many organizational units, such as engineering, marketing, sales, manufacturing, inventory control, distribution, and, yes, information systems.

Similarly, the trend toward collaboration extends beyond the organization to include other organizations—sometimes even competitors. Organizations choose to directly collaborate as partners in business ventures that make good business sense. Microsoft and Oracle sell competitive database management systems. But Microsoft and Oracle also partner to ensure that Oracle applications will operate on a Microsoft database. Both companies benefit financially from such cooperation.

In a similar vein, businesses have learned that it can be beneficial for their information systems to interoperate with one another. For example, while Wal-Mart could generate its own restocking orders for merchandise and send them to its suppliers, it makes more sense to integrate their respective inventory control systems. Suppliers can monitor Wal-Mart's inventory levels directly and can automatically initiate business-to-business transactions to keep the shelves stocked with their merchandise. Both companies benefit. (Of course, this also raises the aforementioned issue of requirements for good security.)

data raw facts about people, places, events, and things that are of importance in an organization. Each fact is, by itself, relatively meaningless.

> Knowledge Asset Management

What is knowledge? *Knowledge* is the result of a continuum of how we process raw data into useful information. Information systems collect raw **data** by capturing business facts (about products, employees, customers, and the like) and processing business transactions. Data gets combined, filtered, organized, and analyzed to produce **information** to help managers plan and operate the business. Ultimately, information is refined by people to create **knowledge** and expertise. Increasingly, organizations are asking themselves, "How can the company manage and share knowledge for competitive advantage? And as workers come and go, how can the workers' knowledge and expertise be preserved within the organization?"

information data that has been processed or reorganized into a more meaningful form for someone. Information is formed from combinations of data that hopefully have meaning to the recipient.

knowledge data and information that are further refined based on the facts, truths, beliefs, judgments experiences, and expertise of the recipient. Ideally information leads to *wisdom*.

Thirty years of data processing and information systems have resulted in an enormous volume of data, information, and knowledge. All three are considered critical *business* resources, equal in importance to the classic economic resources of land, labor, and capital.

The need for knowledge asset management impacts information systems on a variety of fronts. Although we have captured (and continue to capture) a great amount of data and information in information systems, they are loosely integrated in most organizations—indeed, redundant and contradictory data and information are common in information systems. As new information systems are built, we will increasingly be expected to focus on integration of the data and information that can create and preserve knowledge in the organizations for which we work. This will greatly complicate systems analysis and design. In this book, we plan to introduce you to many tools and techniques that can help you integrate systems for improved knowledge management.

business processes tasks that respond to business events (e.g., an order). Business processes are the work, procedures, and rules required to complete the business tasks, independent of any information technology used to automate or support them.

> Continuous Improvement and Total Quality Management

Information systems automate and support **business processes.** In an effort to continuously improve a business process, **continuous process improvement (CPI)** examines a business process to implement a series of small changes for improvement. These changes can result in cost reductions, improved efficiencies, or increased value and profit. Systems analysts are both affected by continuous process improvements and expected to initiate or suggest such improvements while designing and implementing information systems.

continuous process improvement (CPI) the continuous monitoring of business processes to effect small but measurable improvements in cost reduction and value added.

Another ongoing business driver is **total quality management (TQM).** Businesses have learned that quality has become a critical success factor in competition. They have also learned that quality management does not begin and end with the products and services sold by the business. Instead, it begins with a culture that recognizes that

total quality management (TQM) a comprehensive approach to facilitating quality improvements and management within a business.

everyone in the business is responsible for quality. TQM commitments require that every business function, including information services, identify quality indicators, measure quality, and make appropriate changes to improve quality.

Information systems, and hence systems analysts, are part of the TQM requirement. Our discussions with college graduate recruiters suggest that an "obsessive" attitude toward quality management will become an essential characteristic of successful systems analysts (and all information technology professionals). Throughout this book, continuous process improvement and total quality management will be an underlying theme.

> Business Process Redesign

As stated earlier, many information systems support or automate business processes. Many businesses are learning that those business processes have not changed substantially in decades and that those business processes are grossly inefficient and/or costly. Many business processes are overly bureaucratic, and all their steps do not truly contribute value to the business. Unfortunately, information systems have merely automated many of these inefficiencies. Enter business process redesign!

business process redesign (BPR) the study, analysis, and redesign of fundamental business processes to reduce costs and/or improve value added to the business.

Business process redesign (BPR) involves making substantive changes to business processes across a larger system. In effect, BPR seeks to implement more substantial changes and improvements than does CPI. In a BPR, business processes are carefully documented and analyzed for timeliness, bottlenecks, costs, and whether or not each step or task truly adds value to the organization (or, conversely, adds only bureaucracy). Business processes are then redesigned for maximum efficiency and lowest possible costs.

So how does BPR affect information systems? There are two basic ways to implement any information system—build it or buy it. In other words, you can write the software yourself, or you can purchase and implement a commercial software package. In both cases, BPR figures prominently. In writing your own software, it is useful to redesign business processes before writing the software to automate them. This way, you avoid automating underlying inefficiencies. Alternatively, in purchasing software packages, most businesses have discovered it is easier to redesign the business processes to work with the software package than to attempt to force (and even cripple) the software package to work with existing business processes.

Technology Drivers for Today's Information Systems

Advances in information technology can also be drivers for information systems (as suggested in Figure 1-8). In some cases, outdated technologies can present significant problems that drive information system development projects. In other cases, newer technologies present business opportunities. Let's examine several technologies that are influencing today's information systems.

> Networks and the Internet

Scott McNealy, Sun Computer's charismatic CEO, is often cited as stating, "The network has become the computer." Few would argue that today's information systems are installed on a network architecture consisting of local and wide area networks. These networks include mainframe computers, network servers, and a variety of desktop, laptop, and handheld client computers. But today, the most pervasive networking technologies are based on the Internet. Some of the more relevant Internet technologies that you need to become aware of, if not develop some basic skill with, are described in the following list. (For now, don't be intimidated by these terms—we

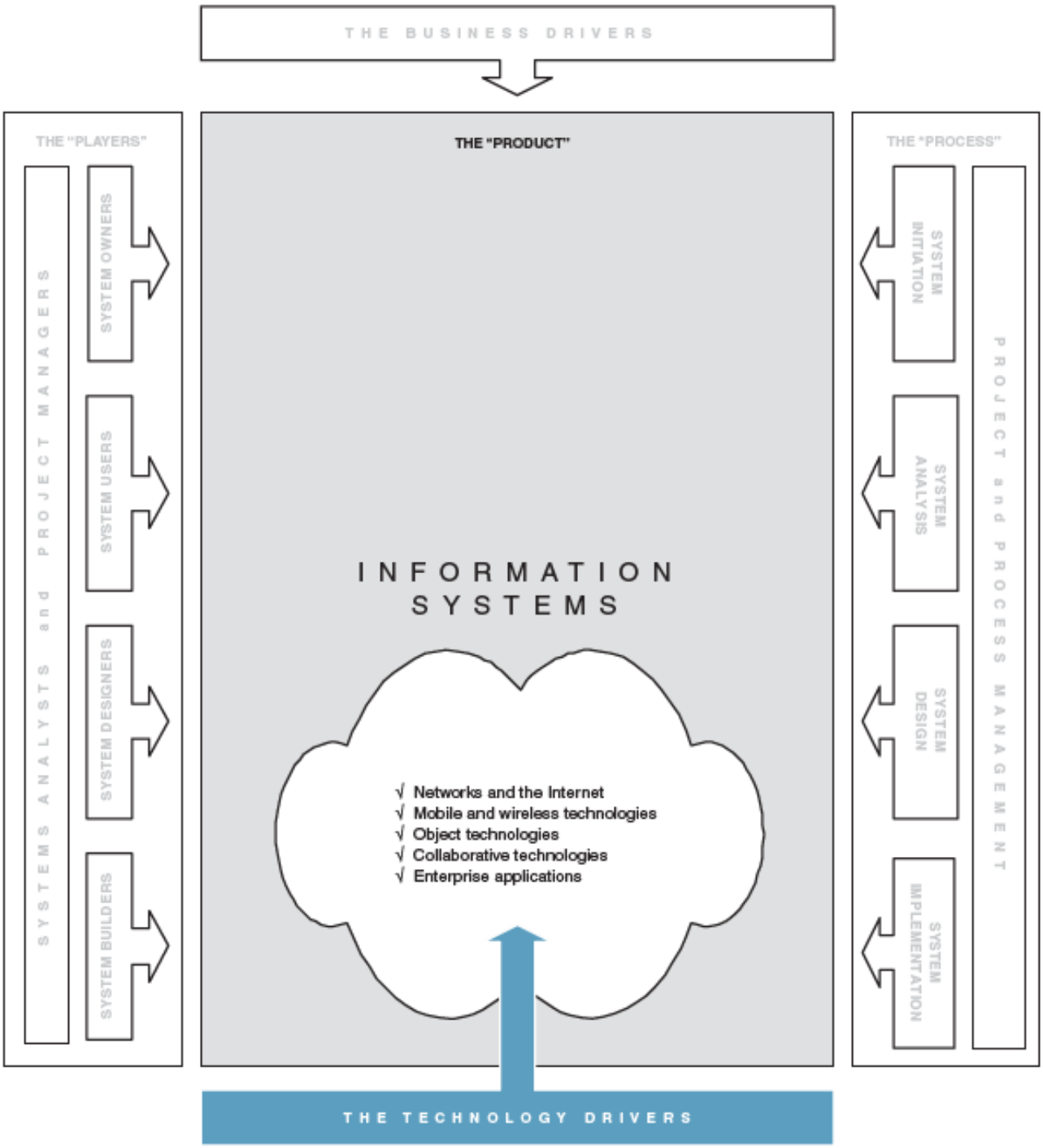


FIGURE 1-8 Technology Drivers for an Information System

will be teaching you more about these technologies and how to design systems that use them throughout this textbook.)

- *xHTML* and *XML* are the fundamental languages of Web page authoring and Internet application development. Extensible Hypertext Markup Language (xHTML) is the emerging second-generation version of HTML, the language used to construct Web pages. Extensible Markup Language (XML) is the language used to effectively transport data content along with its proper inter-

pretation over the Internet. Introductory xHTML and XML courses have become core requirements in most information systems and information technology college curricula.

- *Scripting* languages are simple programming languages designed specifically for Internet applications. Examples include *Perl, VBScript,* and *JavaScript.* These languages are increasingly taught in college Web development and programming courses.
- Web-specific programming languages such as *Java* and *Cold Fusion* have emerged to specifically address construction of complex, Web-based applications that involve multiple servers and Web browsers. These languages are also becoming prevalent in college programming curricula.
- *Intranets* are essentially private Internets designed for use by employees of an organization. They offer the look and feel of the Internet; however, security and firewalls restrict their use to employees.
- *Extranets,* like intranets, are private Internets. But extranets are for use between specific organizations. Only the employees of those identified businesses can access and use the extranet. For example, an automotive manufacturer such as Chevrolet might set up an extranet for the sole use of its dealers. Through this extranet, the manufacturer can communicate information about parts, problems, sales incentives, and the like.
- *Portals* (in corporations) are "home pages" that can be customized to the specific needs of different individuals who use them. For example, portal technology can define Web pages that provide appropriate information and applications for different roles in the same company. Each individual's role determines which information and applications that person can use from her or his Web page. Examples of roles include "customer," "supplier," and different types of "employee." Portals can also effectively integrate public Internet, private intranet, and extranet content into each individual user's home page.
- *Web services* are the latest rage. Web services are reusable, Web-based programs that can be called from any other Internet program. For example, let's say you need to write a program to accept credit card payments over the Web. Sure, you could write, debug, and test the credit card validation program yourself. But an alternative approach would be to purchase the right to use a credit card validation program over the Web. By doing so, you need not maintain responsibility for the credit card validation code. You need only "call" the Web service from your program, much as you would call an internal subroutine. Of course, you will pay for the privilege of using Web services since somebody had to write the original Web service program.

These are but a few of the network and Internet technologies that you should seek out during your education. But you must recognize the volatility of the Internet and accept that these and other technologies will emerge and disappear frequently in the near future.

Wireless Handheld

> Mobile and Wireless Technologies

Mobile and wireless technologies are poised to significantly change the next generation of information systems. Handheld computers, or *personal data assistants* (*PDAs,* such as the HP *iPaq,* Palm, and RIM *BlackBerry®*), have become common in the ranks of information workers. These devices are increasingly including wireless capabilities (see margin photo) that provide Web access and e-mail. *Cell phones* are also increasingly featuring Internet and e-mail capabilities. And now, integrated devices such as *smart phones* are emerging that integrate the capabilities of PDAs and cell phones into a single device (see margin photo). For those who prefer separate devices, technologies like *Bluetooth* are emerging to allow the separate devices to interoperate as one logical device while preserving each one's form factors and advantages.

Additionally, laptop computers are increasingly equipped with wireless and mobile capabilities to allow information workers to more easily move between locations while preserving connectivity to information systems. All of these technical trends will significantly impact the analysis and design of new information systems. Increasingly, wireless access must be assumed. And the limitations of mobile devices and screen sizes must be accommodated in an information system's design. This textbook will teach and demonstrate tools and techniques to deal with the design of emerging mobile applications.

Smart Phone

> Object Technologies

Today, most contemporary information systems are built using **object technologies.** Today's pervasive programming languages are object-oriented. They include *C++*, *Java*, *Smalltalk*, and *Visual Basic .NET*. Object technologies allow programmers to build software from software parts called *objects*. (We will get into more specifics about objects later in this book.) Object-oriented software offers two fundamental advantages over nonobject software. First, objects are reusable. Once they are designed and built, objects can be reused in multiple information systems and applications. This reduces the time required to develop future software applications. Second, objects are extensible. They can be changed or expanded easily without adversely impacting any previous applications that used them. This reduces the lifetime costs of maintaining and improving software.

object technology a software technology that defines a system in terms of objects that consolidate data and behavior (into objects). Objects become reusable and extensible components for the software developers.

The impact of object technology is significant in the world of systems analysis and design. Prior to object technologies, most programming languages were based on so-called *structured methods*. Examples include *COBOL* (the dominant language), *C*, *FORTRAN*, *Pascal*, and *PL/1*. It is not important at this time for you to be able to differentiate between structured and object technologies and methods. Suffice to say, structured methods are inadequate to the task of analyzing and designing systems that will be built using object technologies. Accordingly, **object-oriented analysis and design** methods have emerged as the preferred approach for building *most* contemporary information systems. For this reason, we will integrate object-oriented analysis and design tools and techniques throughout this book to give you a competitive advantage in tomorrow's job market.

object-oriented analysis and design a collection of tools and techniques for systems development that will utilize object technologies to construct a system and its software.

At the same time, structured tools and techniques are still important. Databases, for example, are still commonly designed using structured tools. And structured tools are still preferred by many systems analysts for analyzing and designing work flows and business processes. Thus, we will also teach you several popular structured tools and techniques for systems analysis and design.

It is easy to become a devout disciple of one analysis and design strategy, such as structured analysis and design or object-oriented analysis and design. You should avoid doing so! We will advocate both and teach you when and how to combine structured and object-oriented tools and techniques for systems analysis and design. As we write this chapter, this approach—called **agile development**—is gaining favor among experienced analysts who have become weary of overly prescriptive methods that usually insist that you use only *one* methodology's tools and processes. At the risk of oversimplifying agile methods, think of it as assembling a toolbox of different tools and techniques—structured, object-oriented, and others—and then selecting the best tool or technique for whatever problems or need you encounter as a systems analyst.

agile development a systems development strategy wherein the system developers are given the flexibility to select from a variety of appropriate tools and techniques to best accomplish the tasks at hand. Agile development is believed to strike an optimal balance between productivity and quality for systems development.

> Collaborative Technologies

Another significant technology trend is the use of collaborative technologies. Collaborative technologies are those that enhance interpersonal communications and teamwork. Four important classes of collaborative technologies are e-mail, instant messaging, groupware, and work flow.

Everybody knows what e-mail is. But e-mail's importance in information systems development is changing. Increasingly, modern information systems are *e-mail-enabled;*

that is, e-mail capabilities are built right into the application software. There is no need to switch to a dedicated e-mail program such as *Outlook*. The application merely invokes the user's or organization's default e-mail program to enable relevant messages to be sent or received.

Related to e-mail technology is instant messaging (e.g., AOL's *Instant Messenger* and Microsoft's *MSN Messenger Service*). Instant messaging was popularized in public and private "chat rooms" on the Internet. But instant messaging is slowly being incorporated into enterprise information systems applications as well. For example, instant messaging can implement immediate response capabilities into a help system for a business application. Imagine being able to instantly send and receive messages with the corporate help desk when using a business application. The productivity and service-level implications are significant.

Finally, groupware technolcgy allows teams of individuals to collaborate on projects and tasks regardless of their physical location. Examples of groupware technologies include Lotus's *SameTime* and Microsoft's *NetMeeting*. Using such groupware allows multiple individuals to participate in meetings and share software tools across a network. As with e-mail and instant messaging, groupware capabilities can be built into appropriate business applications.

Clearly, systems analysts and system designers must build these innovative collaborative technologies into their applications.

> Enterprise Applications

Virtually all organizations, large and small, require a core set of enterprise applications to conduct business. As shown in Figure 1-9, for most businesses the core applications include financial management, human resource management, marketing and sales, and operations management (inventory and/or manufacturing control). At one time, most organizations custom-built most or all of these core enterprise applications. But today, these enterprise applications are frequently purchased, installed, and configured for the business and integrated into the organization's business processes. Why? Because these core enterprise applications in different organizations or industries tend to be more alike than they are different.

systems integration the process of buiding a unified information system out of diverse components of purchased software, custom-built software, hardware, and networking.

Today, these "internal" core applications are being supplemented with other enterprise applications that integrate an organization's business processes with those of its suppliers and customers. These applications, called *customer relationship management* and *supply chain management*, are also illustrated in Figure 1-9.

The trend toward the use of purchased enterprise applications significantly impacts systems analysis and design. Purchased and installed enterprise applications are never sufficient to meet all of the needs for information systems in any organization. Thus, systems analysts and other developers are asked to develop value-added applications to meet additional needs of the business. But the purchased and installed enterprise applications become a technology constraint. Any custom application must properly integrate with and interface to the purchased enterprise applications. This is often called **systems integration**, and this is the business and systems environment into which most of you will graduate. Let's briefly explore some of the more common enterprise applications and describe their implications for systems analysis and design.

enterprise resource planning (ERP) a software application that fully integrates information systems that span most or all of the basic, core business functions (including transaction processing and management information for those business functions).

Enterprise Resource Planning (ERP) As previously noted, the core business information system applications in most businesses were developed in-house incrementally over many years. Each system had its own files and databases with loose and awkward integration of all applications. During the 1990s, businesses tried very hard to integrate these legacy information systems, usually with poor results. Organizations would have probably preferred to redevelop these core business applications (see Figure 1-9 again) from scratch as a single *integrated* information system. Unfortunately, few if any businesses had enough resources to attempt this. Recognizing that the basic applications needed by most businesses were more similar than different, the software industry developed a solution—**enterprise resource planning (ERP)**

REPRESENTATIVE ERP VENDORS

SSA
Oracle/PeopleSoft
SAP AG (the Market Leader)

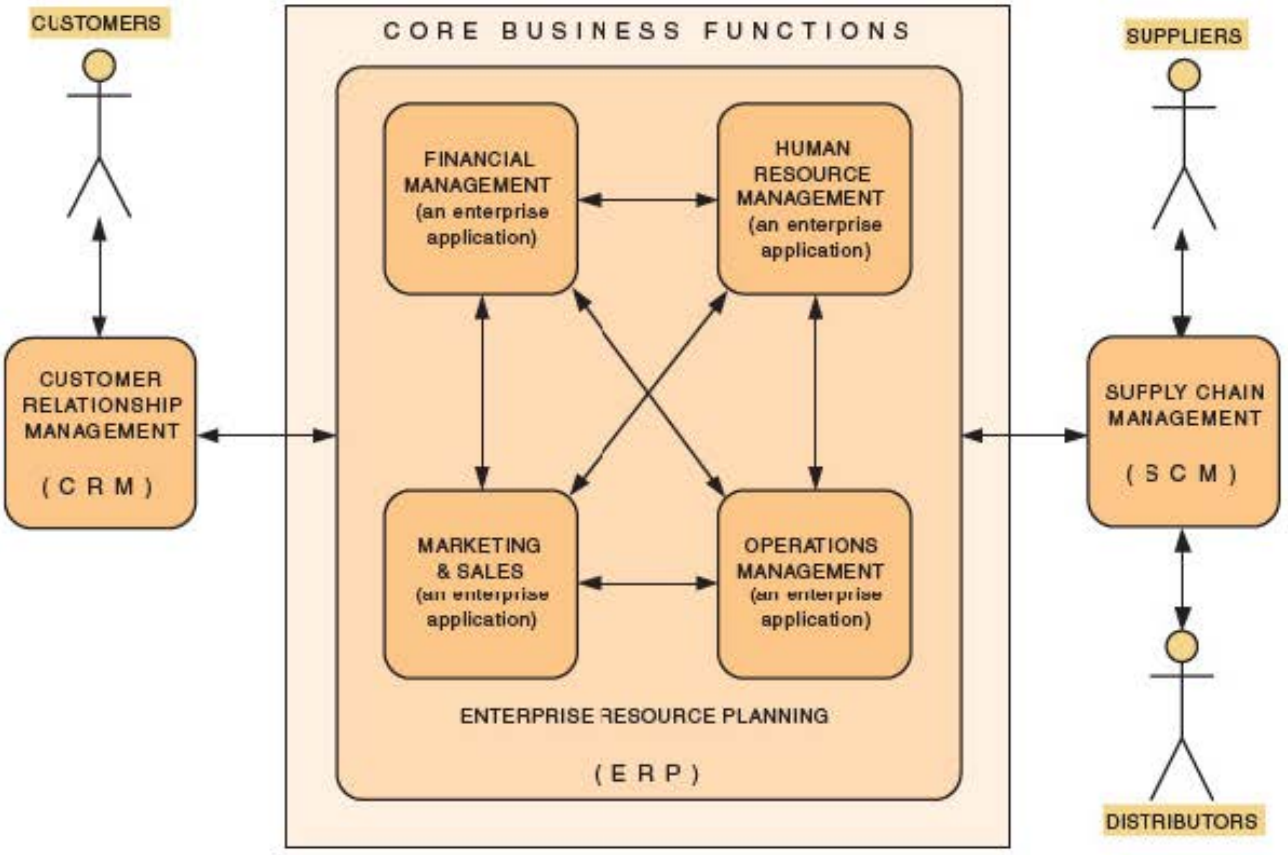


FIGURE 1-9 Enterprise Applications

applications. An ERP solution is built around a common database shared by common business functions. Examples of ERP software vendors are listed in the margin.

An ERP solution provides the core information system functions for the entire business. But usually an organization must redesign its business processes to fully exploit and use an ERP solution. Most organizations must still supplement the ERP solution with custom software applications to fulfill business requirements that are unique to the industry or business. For most organizations, an ERP implementation and integration represents the single largest information system project ever undertaken by the organization. It can cost tens of millions of dollars and require a small army of managers, users, analysts, technical specialists, programmers, and consultants.

ERP applications are significant to systems analysts for several reasons. First, systems analysts may be involved in the decision to select and purchase an ERP solution. Second, and more common, systems analysts are frequently involved in the customization of the ERP solution, as well as the redesign of business processes to use the ERP solution. Third, if custom-built applications are to be developed within an organization that uses an ERP core solution, the ERP system's architecture significantly impacts the analysis and design of the custom application that must coexist and interoperate with the ERP system.

REPRESENTATIVE SCM VENDORS

i2 Technologies
Manugistics
SAP
SCT

Supply Chain Management Today, many organizations are expending effort on enterprise applications that extend support beyond their core business functions. Companies are extending their core business applications to interoperate with their suppliers and distributors to more efficiently manage the flow of raw materials and products between their respective organizations. These **supply chain management (SCM)** applications utilize the Internet as a means for integration and communications.

supply chain management (SCM) a software application that optimizes business processes for raw material procurement through finished product distribution by directly integrating the logistical information systems of organizations with those of their suppliers and distributors.

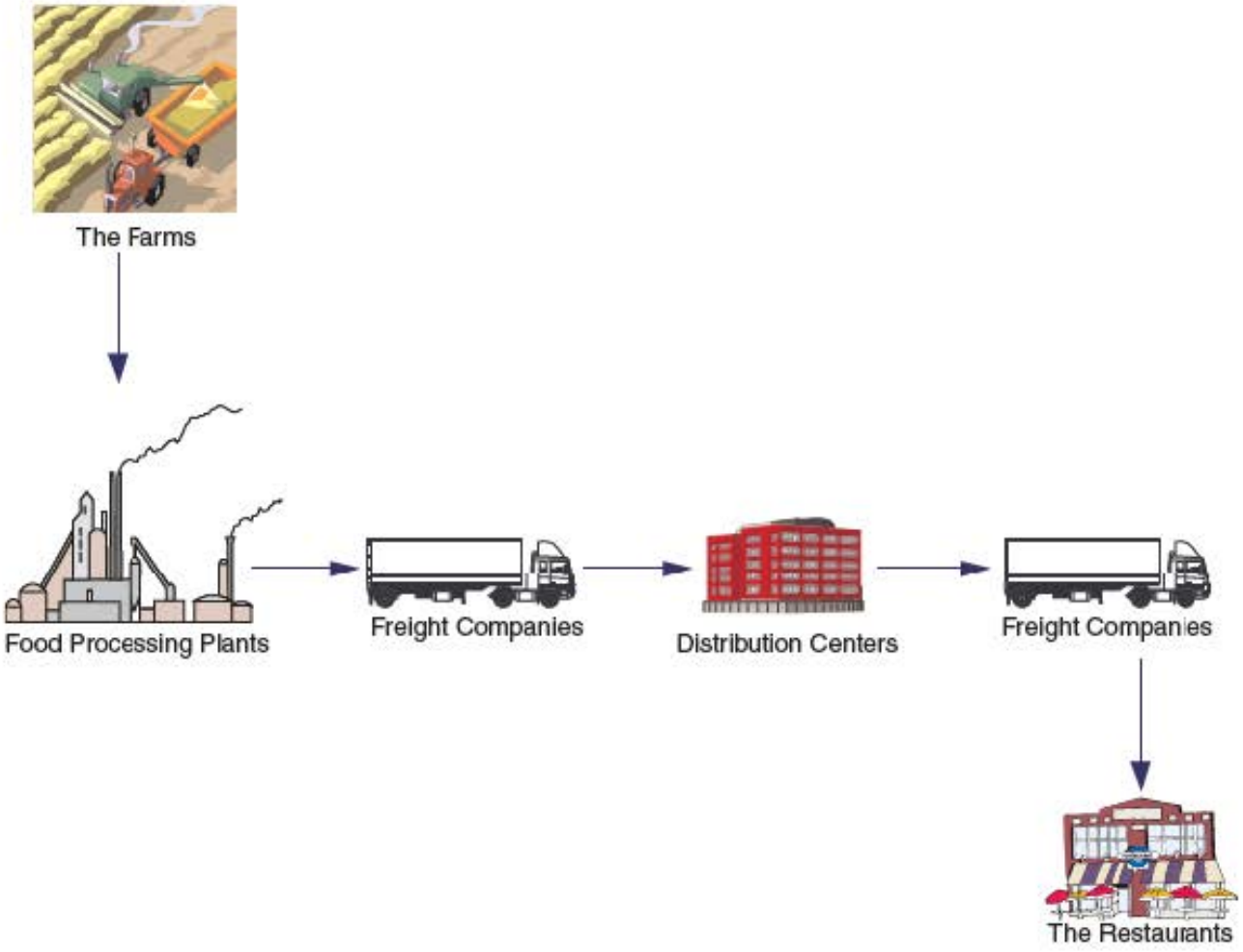


FIGURE 1-10 Supply Chain

REPRESENTATIVE CRM VENDORS

BroadVision
E.piphany
Kana
Amdocs
Oracle/PeopleSoft
Siebel (the Market Leader)
SAP

For example, Figure 1-10 demonstrates a logical supply chain ending at restaurants belonging to a franchise (e.g., Outback, Red Lobster, Wendy's). Notice that this supply chain includes many businesses and carriers to achieve its final result—ensuring that the restaurants have adequate food supplies to do business. Any delays or problems in any single link of this supply chain will adversely affect one and all. For that reason, several of these businesses will implement supply chain management using SCM software technology to plan, implement, and manage the chain. Examples of supply chain management vendors are listed in the margin. (It should be noted that several ERP application vendors are extending ERP software applications to include SCM capabilities. The SCM market is due for a shakeout because there are too many vendors for all to succeed.)

SCM applications are significant to systems analysts for the same reasons as stated for ERP applications. As an analyst, you may be involved in the evaluation and selection of an SCM package. Or you may be expected to implement and perhaps customize such packages to meet the organization's needs. And again, you may expect to participate in redesigning existing business processes to work appropriately with the SCM solution.

customer relationship management (CRM) a software application that provides customers with access to a business's processes from initial inquiry through postsale service and support.

Customer Relationship Management Many companies have discovered that highly focused customer relationship management can create loyalty that results in increased sales. Thus, many businesses are implementing **customer relationship management (CRM)** solutions that enable customer self-service via the Internet.

The theme of all CRM solutions is a focus on the "customer." CRM is concerned with not only providing effective customer inquiry responses and assistance but also helping the business better profile its customer base for the purpose of improving customer relations and marketing. Examples of CRM vendors are listed in the margin. As was the case with SCM technologies, many ERP vendors are developing or acquiring CRM capabilities to complement and extend their ERP solutions. And as with SCM, the larger number of players will likely be reduced through acquisition and attrition.

REPRESENTATIVE EAI VENDORS

BEA Systems
IBM (MQSeries)
Mercator Software
TIBCO Software

CRM technology impacts systems analysts in precisely the same ways as those we described for ERP and SCM technology. In many businesses, new applications must interface with a core, CRM enterprise application.

Enterprise Application Integration Many companies face the significant challenge of integrating their existing legacy systems with new applications such as ERP, SCM, and CRM solutions. Any company that wants to do business across the Internet will also have to meet the challenge of integrating its systems with those of other organizations and their different systems and technologies. To meet this challenge, many organizations are looking at enterprise application integration software. **Enterprise application integration (EAI)** involves linking applications, whether purchased or developed in house, so that they can transparently interoperate with one another. This is illustrated conceptually in Figure 1-11. Some vendors offering EAI tools are listed in the margin.

enterprise application integration (EAI) the process and technologies used to link applications to support the flow of data and information between those applications. EAI solutions are usually based on **middleware.**

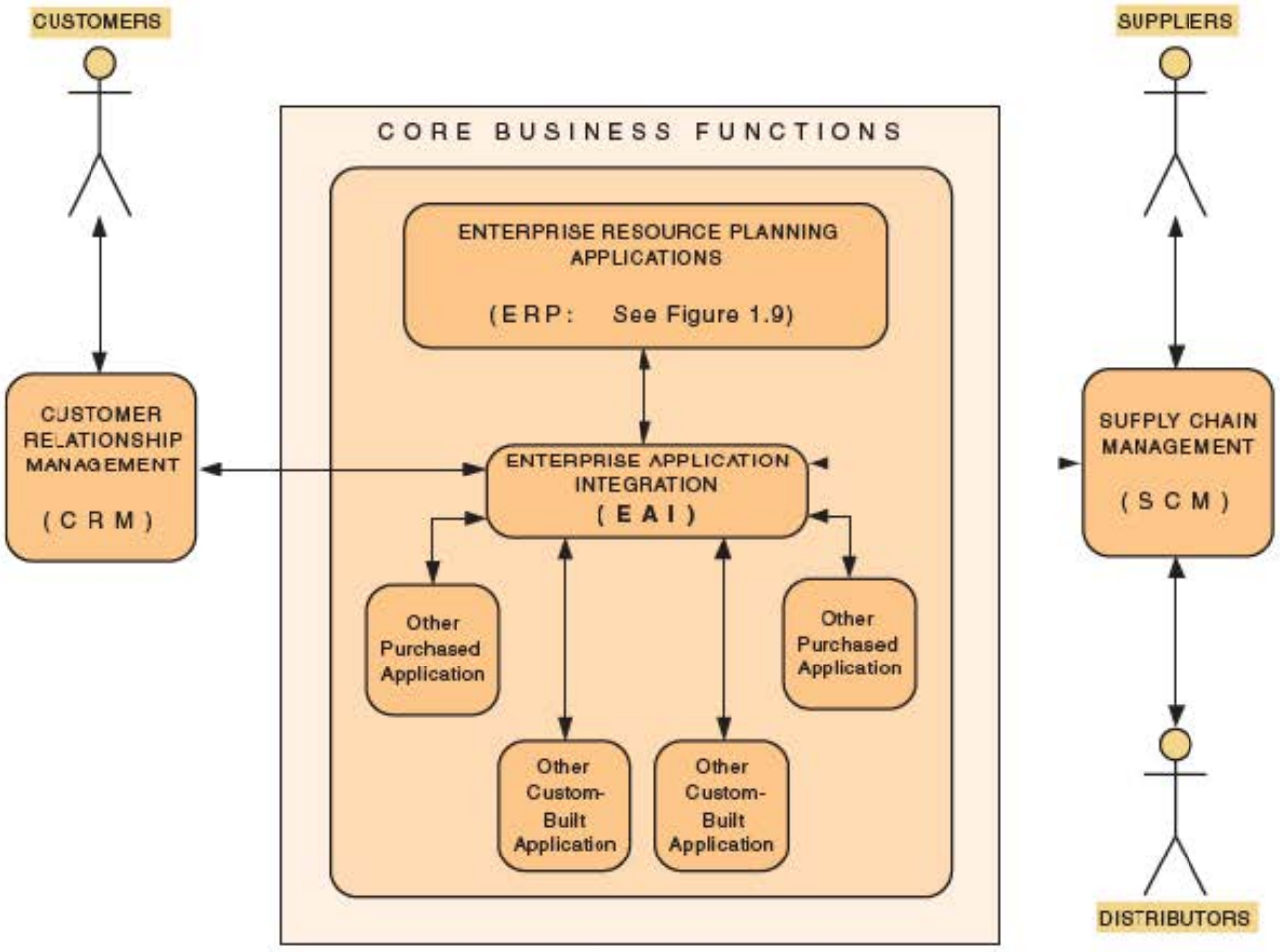


FIGURE 1-11 Enterprise Application Integration

middleware software (usually purchased) used to translate and route data between different applications.

Again, this market is rapidly expanding and contracting. The tools are used to define and construct communication pipelines between differing applications and technologies.

Today, as any new information system is developed, it must be integrated with all the information systems that preceded it. These "legacy" information systems may have been purchased or built in-house. Regardless, systems analysts and other developers must consider application integration for any new information system to be developed. And EAI technologies are at the core of the integration requirements.

A Simple System Development Process

Thus far you have learned about different types of information systems, the players involved in developing those systems, and several business and technology drivers that influence the development of information systems. In this section you will learn about another information system perspective, the "process" for developing an information system.

system development process a set of activities, methods, best practices, deliverables, and automated tools that stakeholders use to develop and maintain information systems and software.

Most organizations have a formal **system development process** consisting of a standard set of processes or steps they expect will be followed on any system development project. While these processes may vary greatly for different organizations, a common characteristic can be found: Most organizations' system development process follows a problem-solving approach. That approach typically incorporates the following general problem-solving steps:

1. Identify the problem.
2. Analyze and understand the problem.
3. Identify solution requirements and expectations.
4. Identify alternative solutions and choose the best course of action.
5. Design the chosen solution.
6. Implement the chosen solution.
7. Evaluate the results. (If the problem is not solved, return to step 1 or 2 as appropriate.)

Figure 1-12 adds a system development process perspective that we will use (with appropriate refinements) throughout this book as we study the development process, tools, and techniques. For the sake of simplicity our initial problem-solving approach establishes four stages or phases that must be completed for any system development project—system initiation, system analysis, system design, and system

Our Simplified System Development Process	General Problem-Solving Steps
System initiation	1. Identify the problem. (Also plan for the solution of the problem.)
System analysis	2. Analyze and understand the problem. 3. Identify solution requirements and expectations.
System design	4. Identify alternative solutions and choose the best course of action. 5. Design the chosen solution.
System implementation	6. Implement the chosen solution. 7. Evaluate the results. (If the problem is not solved, return to step 1 or 2 as appropriate.)

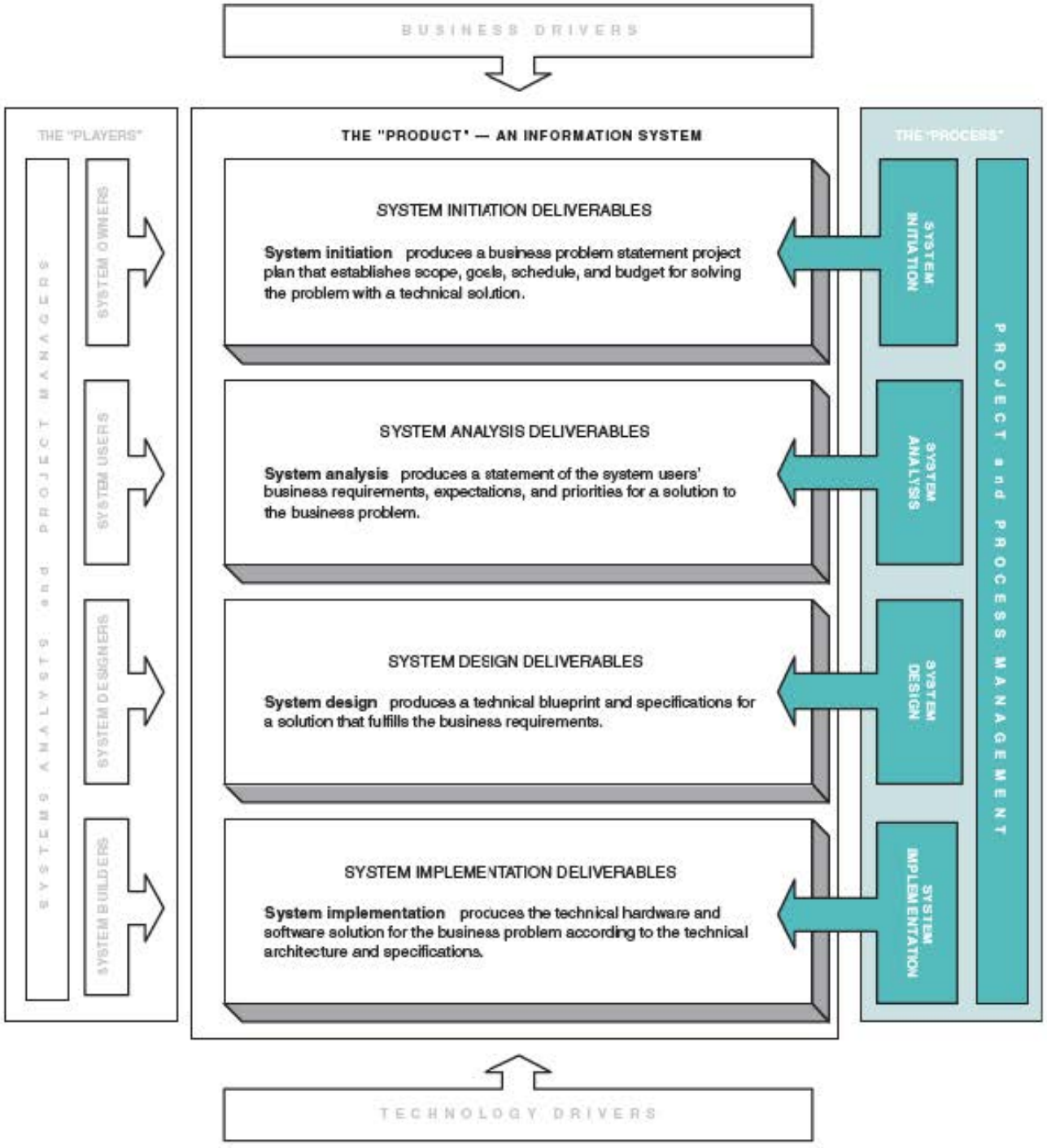


FIGURE 1-12 Systems Development and Problem Solving

implementation. The table on the previous page shows the correlation between the above general problem-solving steps and our process.

It is important to note that any system development process must be managed on a project-by-project basis. You learned earlier that at least one stakeholder accepts responsibility as the project manager for ensuring that the system is developed on time, within budget, and with acceptable quality. The activity of managing a project is referred to as **project management.** Accordingly, in Figure 1-12 we have added a process for project management. Also, to ensure that all projects are

project management the activity of defining, planning, directing, monitoring, and controlling a project to develop an acceptable system within the allotted time and budget.

process management the ongoing activity that defines, improves, and coordinates the use of an organization's chosen methodology (the "process") and standards for all system development projects.

managed according to the same development process, we have included **process management** as an ongoing activity. Notice that project and process management overlap all of the process phases.

Let's *briefly* examine our system development process in Figure 1-12 to expand your understanding of each phase and activity in the process. Given a problem to be solved or a need to be fulfilled, what will we do during system initiation, analysis, design, and implementation? Also, who will be involved in each phase?

> System Initiation

Information system projects are usually complicated. They require a significant time, effort, and economic investment. The problems to be solved are frequently stated vaguely, which means that the initial envisioned solution may be premature. For these reasons, system projects should be carefully planned. System initiation establishes project scope and the problem-solving plan. Thus, as shown in Figure 1-12, we see that **system initiation** establishes the project scope, goals, schedule, and budget required to solve the problem or opportunity represented by the project. Project scope defines the area of the business to be addressed by the project and the goals to be achieved. Scope and goals ultimately impact the resource commitments, namely, schedule and budget, that must be made to successfully complete the project. By establishing a project schedule and budget against the *initial* scope and goals, you also establish a *baseline* against which all stakeholders can accept the reality that any future changes in scope or goals *will* impact the schedule and budget.

system initiation the initial planning for a project to define initial business scope, goals, schedule, and budget.

Figure 1-12 also shows that project managers, system analysts, and system owners are the primary stakeholders in a system analysis. This book will teach you many tools and techniques for initiating a system project and establishing a suitable project plan.

> System Analysis

system analysis the study of a business problem domain to recommend improvements and specify the business requirements and priorities for the solution.

The next step in our system development process is **system analysis.** System analysis is intended to provide the project team with a more thorough understanding of the problems and needs that triggered the project. As such, the business area (scope of the project—as defined during system initiation) may be studied and analyzed to gain a more detailed understanding of what works, what doesn't, and what's needed. As depicted in Figure 1-12, the system analysis requires working with system users to clearly define business requirements and expectations for any new system that is to be purchased or developed. Also, business priorities may need to be established in the event that schedule and budget are insufficient to accomplish all that is desired.

Recall the business drivers discussed earlier in the chapter. These (and future) business drivers most closely affect system analysis, which often defines business requirements in response to the business drivers. For example, we discussed a current trend toward e-business and e-commerce. This business driver may influence the business requirement for any information system, leading us to establish project goals to conduct all business transactions on the Web.

The completion of a system analysis often results in the need to update many of the deliverables produced earlier, during system initiation. The analysis may reveal the need to revise the business scope or project goals—perhaps we now feel the scope of the project is too large or too small. Accordingly, the schedule and budget for the project may need to be revised. Finally, the feasibility of the project itself becomes questionable. The project could be canceled or could proceed to the next phase.

As shown in Figure 1-12, project managers, system analysts, and system users are the primary stakeholders in a system analysis. Typically, results must be summarized and defended to the system owners, who will pay to design and implement an information system to fulfill the business requirements. This book will teach you

many tools and techniques for performing a system analysis and documenting user requirements.

> System Design

Given an understanding of the business requirements for an information system, we can now proceed to **system design.** During system design we will initially need to explore alternative technical solutions. Rarely is there only one solution to any problem. For example, today most companies need to choose between purchasing a solution that is good enough and building a custom solution in-house. (We'll explore options such as this throughout this book.)

system design the specification or construction of a technical, computer-based solution for the business requirements identified in a system analysis. (Note: Increasingly, the design takes the form of a working prototype.)

Once a technical alternative is chosen and approved, the system design phase develops the technical blueprints and specifications required to implement the final solution. These blueprints and specifications will be used to implement required databases, programs, user interfaces, and networks for the information system. In the case where we choose to purchase software instead of build it, the blueprints specify how the purchased software will be integrated into the business and with other information systems.

Recall the technology drivers discussed in the last section of the chapter. These (and future) technology drivers most closely impact the system design process and decisions. Many organizations define a common information technology architecture based on these technology drivers. Accordingly, all system designs for new information systems must conform to the standard IT architecture.

As depicted in Figure 1-12, project managers, system analysts, and system designers are the primary stakeholders in a system design. This book will teach you many tools and techniques for performing a system design.

> System Implementation

The final step in our simple system development process is **system implementation.** As shown in Figure 1-12, system implementation constructs the new information system and puts it into operation. It is during system implementation that any new hardware and system software are installed and tested. Any purchased application software and databases are installed and configured. And any custom software and databases are constructed using the technical blueprints and specifications developed during system design.

system implementation the construction, installation, testing, and delivery of a system into production (meaning day-to-day operation).

As system components are constructed or installed, they must be individually tested. And the complete system must also be tested to ensure that it works properly and meets user requirements and expectations. Once the system has been fully tested, it must be placed into operation. Data from the previous system may have to be converted or entered into start-up databases, and system users must be trained to properly use the system. Finally, some sort of transition plan from older business processes and information systems may have to be implemented.

And once again, as depicted in Figure 1-12, project managers, system analysts, and system builders are the primary stakeholders in a system implementation. While this book will teach you some of the tools and techniques for performing a system implementation, most of these methods are taught in programming, database, and networking courses. This book emphasizes system initiation, analysis, and design skills, but it will also teach you the unique system implementation tools and techniques that are most commonly performed by systems analysts and, therefore, are not typically covered in these other information technology courses.

> System Support and Continuous Improvement

We would be remiss not to briefly acknowledge that implemented information systems face a lifetime of support and continuous improvement. But where is that shown in Figure 1-12? It is there! But it is subtle.

Implemented information systems are rarely perfect. Your users will find errors (bugs) and you will discover, on occasion, design and implementation flaws that require attention and fixes. Also, business and user requirements constantly change. Thus, there will be a need to continuously improve any information system until the time it becomes obsolete. So where does system support and change fit into our development process?

A change made for system support or improvement is merely another project, sometimes called a *maintenance* or *enhancement* project. Such a project should follow the exact same problem-solving approach defined for any other project. The only difference is the effort and budget required to complete the project. Many of the phases will be completed much more quickly, especially if the original stakeholders properly documented the system as initially developed. Of course, if they did not, a system improvement project can quickly consume much greater time, effort, and money. Much of what we will teach you in this book is intended to help you appropriately document information systems to significantly reduce lifetime costs of supporting and improving your information systems.

Learning Roadmap

Each chapter will provide guidance for self-paced instruction under the heading "Learning Roadmap." Recognizing that different students and readers have different backgrounds and interests, we will propose appropriate learning paths—most within this book, but some beyond the scope of this book.

Most readers should proceed directly to Chapter 2 because the first four chapters provide much of the context for the remainder of the book. Several recurring themes, frameworks, and terms are introduced in those chapters to allow you to define your own learning path from that point forward. This chapter focused on information systems from four different perspectives:

- The *players*—both developers and users of information systems.
- The *business drivers* that currently influence information systems.
- The *technology drivers* that currently influence information systems.
- The *process* of developing information systems.

Chapter 2 will take a closer look at the *product* itself—information systems—from an architectural perspective appropriate for systems development. We will define how different players and development stages view an information system.

Looking further ahead, Chapter 3 more closely examines the *process* of systems development. Chapter 4 completes the introduction to systems analysis and design methods by examining the *management* of systems development.

Summary

1. Information systems in organizations capture and manage data to produce useful information that supports an organization and its employees, customers, suppliers, and partners.
2. Information systems can be classified according to the functions they serve, including:
   a. Transaction processing systems that process business transactions such as orders, time cards, payments, and reservations.
   b. Management information systems that use transaction data to produce information needed by managers to run the business.

c. Decision support systems that help various decision makers identify and choose between options or decisions.
d. Executive information systems that are systems tailored to the unique information needs of executives who plan for the business and assess performance against the plans.
e. Expert systems that are systems that capture and reproduce the knowledge of an expert problem solver or decision maker and then simulate the "thinking" of that expert.
f. Communication and collaboration systems that enhance communication and collaboration between people, both internal and external to the organization.
g. Office automation systems that help employees create and share documents that support day-to-day office activities.

3. Information systems can be viewed from various perspectives, including from the perspective of the "players," the "business drivers" influencing the information system, the "technology drivers" used by the information system, and the "process" used to develop the information system.
4. Information workers are the stakeholders in information systems. Information workers include those people whose jobs involve the creation, collection, processing, distribution, and use of information. They include:

a. System owners, the sponsors and chief advocates of information systems.
b. System users, the people who use or are impacted by the information system on a regular basis. Geographically, system users may be internal or external.
c. System designers, technology specialists who translate system users' business requirements and constraints into technical solutions.
d. System builders, technology specialists who construct the information system based on the design specifications.
e. Systems analysts, who facilitate the development of information systems and computer applications. They coordinate the efforts of the owners, users, designers, and builders. Frequently, they may play one of those roles as well. Systems analysts perform systems analysis and design.

5. In addition to having formal systems analysis and design skills, a systems analyst must develop or possess the following skills, knowledge, and traits:

a. Working knowledge of information technologies.
b. Computer programming experience and expertise.
c. General knowledge of business processes and terminology.
d. General problem-solving skills.
e. Good interpersonal communication skills.
f. Good interpersonal relations skills.
g. Flexibility and adaptability.
h. Character and ethics.

6. Any stakeholder role may be filled by an internal or external worker referred to as an external service provider (ESP). Most ESPs are systems analysts, designers, or builders who are contracted to bring special expertise or experience to a specific project.
7. Most information systems projects involve working as a team. Usually one or more of the stakeholders (team members) takes on the role of project manager to ensure that the system is developed on time, within budget, and with acceptable quality. Most project managers are experienced systems analysts.
8. Business drivers influence information systems. Current business drivers that will continue to influence the development of information systems include:

a. Globalization of the economy.
b. Electronic commerce and business.
c. Security and privacy.
d. Collaboration and partnership.
e. Knowledge asset management.
f. Continuous improvement and total quality management.
g. Business process redesign.

9. Information technology can be a driver of information systems. Outdated technologies can present problems that drive the need to develop new systems. Newer technologies such as the following are influencing today's information systems:

a. Networks and the Internet:

i) *xHTML* and *XML* are the fundamental languages of Web page authoring and Internet application development. Extensible Hypertext Markup Language (xHTML) is the emerging second-generation version of HTML, the language used to construct Web pages. Extensible Markup Language (XML) is the language used to effectively

transport data content along with its proper interpretation over the Internet.
ii) *Scripting* languages are simple programming languages designed specifically for Internet applications.
iii) Web-specific programming languages such as *Java* and *Cold Fusion* have emerged to specifically address construction of complex, Web-based applications that involve multiple servers and Web browsers.
iv) *Intranets* are essentially private Internets designed for use by employees of an organization. They offer the look and feel of the Internet; however, security and firewalls restrict their use to employees.
v) *Extranets*, like intranets, are private Internets. But extranets are for use between specific organizations. Only the employees of those identified businesses can access and use the extranet.
vi) *Portals* (in corporations) are "home pages" that can be customized to the specific needs of different individuals who use them. For example, portal technology can define Web pages that provide appropriate information and applications for different roles in the same company. Each individual's role determines which information and applications that person can use from her or his Web page.
vii) Web services are reusable, Web-based programs that can be called from any other Internet program.

b. Mobile and wireless technologies—Increasingly, wireless access must be assumed. And the limitations of mobile devices and screen sizes must be accommodated in an information system's design. All of the following technical trends will significantly impact the analysis and design of new information systems:

i) Handheld computers, or *personal data assistants* (such as the HP *iPaq*, Palm, and RIM *BlackBerry*) have become common in the ranks of information workers. These devices are increasingly including wireless capabilities that provide Web access and e-mail
ii) *Cell phones* are also increasingly featuring Internet and e-mail capabilities.
iii) Integrated devices such as *smart phones* are emerging that integrate the capabilities of PDAs and cell phones into a single device.
iv) Technologies like *Bluetooth* are emerging to allow separate devices to interoperate as one logical device while preserving each one's form factors and advantages.

c. Object technologies—Most contemporary information systems are built using object technologies. Object technologies allow programmers to build software from software parts called objects. Object-oriented software offers the advantage of reusability and extensibility.
d. Collaborative technologies—Collaborative technologies are those that enhance interpersonal communications and teamwork. Four important classes of collaborative technologies are e-mail, instant messaging, groupware, and work flow.
e. Enterprise applications—Virtually all organizations, large and small, require a core set of enterprise applications to conduct business. For most businesses the core applications include financial management, human resource management, marketing and sales, and operations management (inventory and/or manufacturing control). At one time, most organizations custom-built most or all of these core enterprise applications. But today, these enterprise applications are frequently purchased, installed, and configured for the business and integrated into the organization's business processes. These "internal" core applications are being supplemented with other enterprise applications that integrate an organization's business processes with those of its suppliers and customers. These applications are called customer relationship management (CRM) and supply chain management (SCM). Enterprise application integration (EAI) involves linking applications, whether purchased or developed in-house, so that they can transparently interoperate with one another.

10. Many organizations have a formal systems development process consisting of a standard set of processes or steps they expect will be followed on any systems development project. Systems development processes tend to mirror general problem-solving approaches. This chapter presented a simplified system development process that is composed of the following phases:

a. System initiation—the initial planning for a project to define initial business scope, goals, schedule, and budget.

b. System analysis—the study of a business process domain to recommend improvements and specify the business requirements and priorities for the solution.
c. System design—the specification or construction of a technical, computer-based solution for the business requirements identified in system analysis.
d. System implementation—the construction, installation, testing, and delivery of a system into operation.

11. Information systems face a lifetime of support and continuous improvement. A change made for system support or improvement is merely another project, sometimes called a maintenance or enhancement project. These projects follow the exact same problem-solving approach defined for any other project, but they require less effort and budget.
12. Sequential development requires that each development process (phase) be completed—one after the other. This approach is referred to as the waterfall approach. An alternative development approach is iterative (or incremental) development. This approach requires completing enough analysis, design, and implementation as is necessary to fully develop a part of the new system. Once that version of the system is implemented, the strategy is to then perform some additional analysis, design, and implementation in order to release the next version of the system. These iterations continue until all parts of the entire information system have been developed.

Review Questions

1. Why are information systems (IS) essential in organizations?
2. Why do systems analysts need to know who the stakeholders are in the organization?
3. Who are the typical stakeholders in an information system? What are their roles?
4. Please explain what the consequences are if an information system lacks a system owner.
5. What are the differences between internal users and external users? Give examples.
6. What are the differences between the role of system analysts and the role of the rest of the stakeholders?
7. What kind of knowledge and skills should a system analyst possess?
8. In addition to the business and computing knowledge that system analysts should possess, what are the other essential skills that they need to effectively complete their jobs?
9. Why are good interpersonal communication skills essential for system analysts?
10. What are some of the business drivers for today's information systems?
11. What are the differences between electronic commerce (e-commerce) and electronic business (e-business)?
12. What are the differences between information and knowledge?
13. What are the most important technology drivers for today's information systems?
14. What are the four steps in a system development process? What happens in each step?
15. Why is system initiation essential in the system development process?

Problems and Exercises

1. Assume you are a systems analyst who will be conducting a requirements analysis for an individually owned brick-and-mortar retail store with a point-of-sale system. Identify who the typical internal and external users might include.
2. Assume you are a systems analyst for a consulting company and have been asked to assist the chief executive officer (CEO) of a regional bank. The bank recently implemented a plan to reduce the number of staff, including loan officers, as a strategy to maintain profitability. Subsequently, the bank has experienced chronic problems with backlogged loan requests because of the limited number of loan officers who are able to review and approve or disapprove loans. The CEO of the bank is interested in solutions that would allow the approval process to move faster without increasing the number of loan officers, and has

engaged your company to come up with suggestions. What is one type of system that you might recommend to the bank?

3. How do communication and collaboration systems improve efficiency and effectiveness? What are some of the communication and collaboration systems that are being used by an increasing number of organizations?
4. Identify the type of information system that clerical workers in an organization would typically use and why.
5. As information systems increase in complexity and comprehensiveness, ethical issues regarding accessing and using data from these systems are also increasing. What are some of these ethical issues?
6. What are business to consumer (B2C) and business to business (B2B) Web applications, and what are some examples of each type?
7. While system development processes and methodologies can vary greatly, identify and briefly explain the "generic" phases of the system development process that are described in the textbook and which must be completed for any project. You are a contractor with a systems integration company.
8. Your company has a contract with a local firm to link all of their systems so they can transparently work together. Their applications include a number of existing legacy systems, which were built at different times by different developers using a variety of languages and platforms, as well as several newer contemporary applications. What is the term for this type of linking? What type of tool would you most likely use, and what are some examples of these tools?
9. Your company has asked you to develop a new Web-based system to replace its existing legacy system. There will be very little change in business requirements and functionality from the existing legacy system. Suggest which system development process you might use and why.
10. You recently joined a retail sales company which has recently bought out and assimilated a commercial industrial supply house. You have been asked to lead a project to develop a consolidated inventory-tracking system. Suggest which system development process you might use and why.
11. Your company president sits down beside you just before a meeting is to begin and tells you that people keep saying the customer needs to install a CRM, but doesn't really know what it is. The company president then asks you to explain it in nontechnical terms in the next 30 seconds.
12. Industry studies indicate that mobile and wireless technology has become one of the major technology drivers for designing new information systems. Why is this the case and what is the impact?
13. Briefly explain the impact of Web services on Web development. Give some examples of Web services.
14. Identify in which phase of the development process the following activities belong:

    a. Development of the technical blueprint or design document.
    b. Project scheduling.
    c. Integration testing.
    d. Interviewing system users to define business requirements.

15. What are the two most important advantages of object-oriented software technologies over structured software technologies?

Projects and Research

1. Research the average and/or median salaries for IT professionals. You can use a variety of methods to find this information, such as searching the Web for online sites that publish the results of salary surveys for IT professionals. You can also look at classified ads in newspapers, trade magazines, and/or online.

    a. Is there a significant difference between typical salaries for system analysts, designers and developers?
    b. Roughly, what is the difference in the typical salaries for these different groups?
    c. What do you think are the reasons for the difference?
    d. Is there a gender gap in the salaries of IT professionals? Discuss any trends that you found, and the implications.

2. Contact the chief information officers (CIOs) or top IT managers of several local or regional organizations. Ask them about the process or

methodology they use for system development in their organizations, and why they utilize that particular approach.

a. Describe and compare the different approaches that you have found.
b. Which approach do you believe to be the most effective approach?
c. Why?

3. Career choices and personal skills:

a. At this point in your education, if you had to choose between becoming a systems analyst, systems designer, or systems builder, which one would you choose?
b. Why?
c. Now divide a piece of paper into two columns. On one side, list the personal skills and traits you think are most important for each of these three groups of systems analysts, designers, and builders. In the second column, list at least five skills and traits that you feel to be your strongest ones, then map them to the skills and traits you listed for each of the three groups. With which group do you have the most skills and traits in common?
d. Is this group the same one as the one you would choose in Question 3a? Why do you think this is (or is not) the case?

4. Your school library should have journals and periodicals dating back at least several decades, or may subscribe to online research services which do. Look at several recent articles in information technology journals regarding systems analysis, as well as several articles from at least 25 years ago.

a. Compare the recent articles to the older ones. Do there appear to be significant differences in the typical knowledge, skills, abilities, and/or experience that systems analysts needed 25 years ago compared to now?
b. If you found some differences, what are the ones that you consider most important?
c. What do you think are some of the reasons for these changes?
d. Now get out your crystal ball and look into the future 25 years from now. What differences do you predict between the systems analysts of today and those in 25 years?

5. Search the Web for several articles and information on ethical issues related to information technology professionals.

a. What articles did you find?
b. Based on your research, which ethical issues do you think systems analysts might face during their careers?
c. Pick a particular ethical issue and explain the steps you would take if you were confronted with this issue.
d. What would you do if you found your best friend and co-worker had committed a serious ethical violation? Facts to consider: The violation may never be discovered, but it will cost your company many thousands of dollars in higher costs over the next several years. Your company has a stringent policy of firing employees who commit serious ethical violations. Make sure to explain your reasons for the action(s) you would take, if any.

6. Search the Web or business periodicals in your library such as Forbes Magazine for information on three or four chief information officers of large companies or organizations.

a. Which industry sector, companies, and CIOs did you find?
b. For each CIO that you researched, what was their predominant experience prior to becoming a CIO; that is, did they have an information technology background, a business background, or both?
c. For each CIO, what is their level of education?
d. How many years has each been a CIO, and for approximately how many different companies has each one worked?
e. Based upon your research, what knowledge and skills does a CIO need in order to be successful? Why?

Minicases

1. What do you think will be possible technologically 10 years from now? How about 20 or 30 years from now? Research a new and interesting technology that is in the research and development stage. Prepare a presentation using a movie clip and PowerPoint on this technology and present it to the class. Submit a short paper on the impacts this new technology might have on society and/or businesses.
2. Consider outsourcing: It is many times the case that at least part of the development process is outsourced. In fact, project leaders today must be

capable of handling geographically diverse teams as well as timeline and resource constraints. Outsourcing brings to the table increased efficiency and economic gains to the societies that are interacting. However, these gains are not quickly realized, and the negative impacts on a society that is outsourcing can be significant from a jobs perspective. Dr. Mankiw, as an economic advisor to President Bush, publicly touted the benefits of outsourcing and was deeply criticized for his stance. Do you think that it is good or bad for a business to outsource work? Do you think there are ethical dilemmas for companies who outsource? Find at least two articles on the impacts of outsourcing, and bring them to share with the class.

3. You are a network administrator, and as part of your job, you monitor employee e-mails. You discover that your boss is cutting corners on a system that your company is developing in order to finish the project more quickly and to stay under budget. There is a flaw in the system as a result, and this flaw will cause a network crash if more than 20 people are on the network at a time. The client expects approximately 12 people on the network at any given time. You are sure, as apparently your boss is, that the customer will not find out until well after the project is accepted (if ever). What do you do?
4. A systems analyst must be both technically proficient and capable of successful customer communication. Developing a good system requires a complete understanding of user requirements. Many times, users don't know what is available (technologically) or even what they would like from a system. What are characteristics of good communication?

Team and Individual Exercises

1. Get together into small groups of two. The first person will decide on a task that he/she wishes to be completed—for instance, sharpen a pencil or write down the name of the professor. It should be simple and straightforward. That person is to communicate on paper using only diagrams and no words (verbal or written) what he/she wishes to be done, and give it to person number two. Person number two should then complete the requested task.
2. What did you discover from this exercise? How long did it take until the second person understood what the first person was asking for? Was there miscommunication? Write down your thoughts and observations, and share them with the class.
3. Individual exercise: Imagine a really cool technology. The sky is the limit, and anything is possible. How does this technology impact your life? Does it impact business?
4. Individual exercise: Think back on the last time someone told you something *couldn't* be done. What was it? Did you listen to them? Why or why not?

Suggested Readings

Ambler, Scott. *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process.* New York: John Wiley & Sons, 2002. This book has significantly shaped our thinking about the software development process. Those of you who are critical of the "extreme-programming" movement need not fear that our enthusiasm for this suggested reading indicates an endorsement of extreme programming. We simply like the sanity that Scott brings to the process of systems and software development through the use of flexible methods within the context of an iterative process. We will reference this book in several chapters.

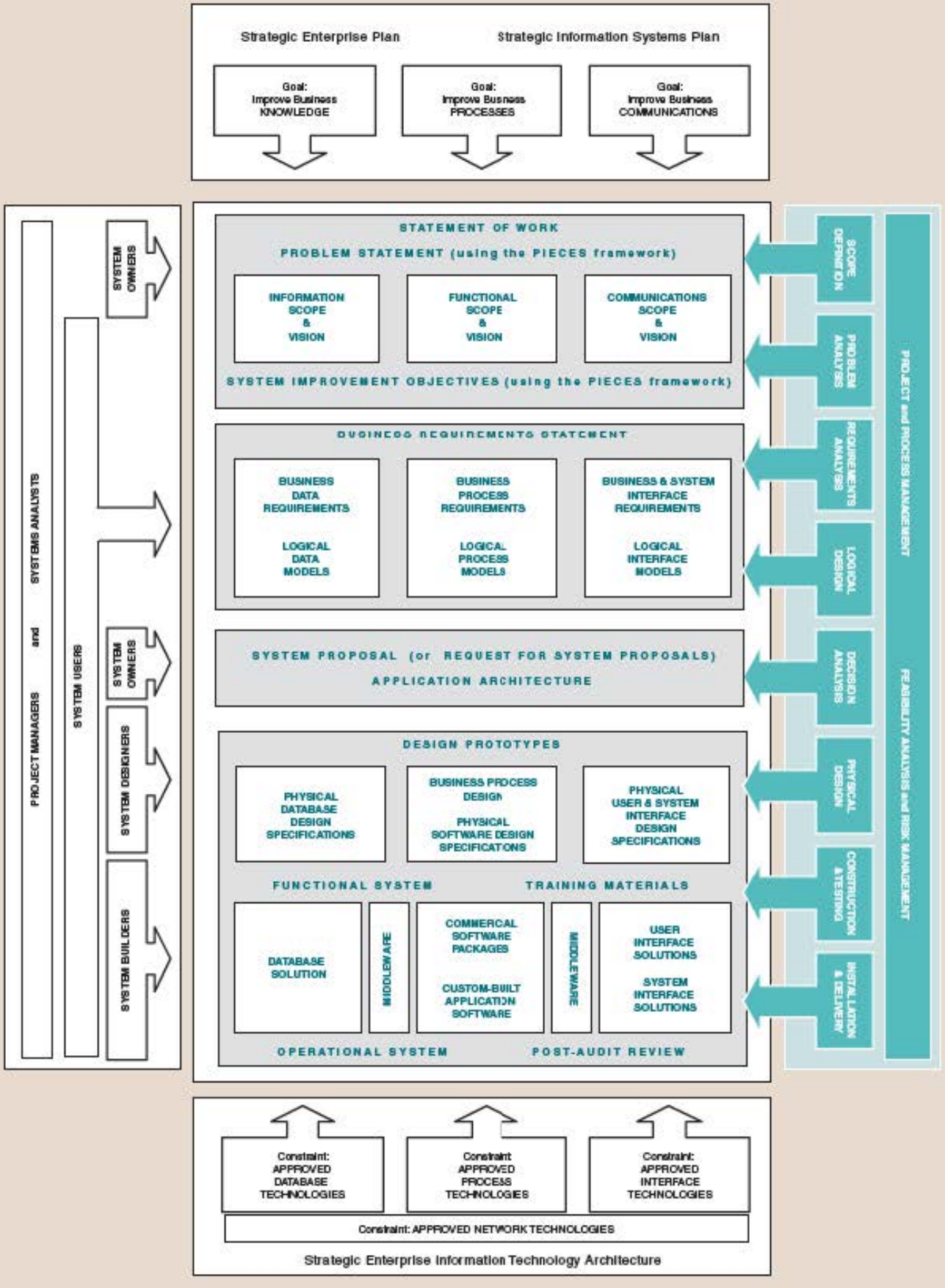
Ernest, Kallman; John Grillo; and James Linderman. *Ethical Decision Making and Information Technology: An Introduction with Cases,* 2nd ed. Burr Ridge, IL: McGraw-Hill/Irwin, 1995. This is an excellent textbook for teaching ethics in an MIS curriculum. It is a collection of case studies that can complement a systems analysis and design course.

Gartner Group IT Symposium and Expo (annual). Our university's management information unit has long subscribed to the Gartner Group's service that reports on industry trends, the probabilities for success of trends and technologies, and suggested strategies for information technology transfer. Gartner research has played a significant, ongoing role in helping us to chart business and technology drivers as described in this chapter. We have also been fortunate to be able to attend Gartner's annual IT symposium. Gartner Group reports and symposiums have influenced each edition of the book. For more information about the Gartner Group, see www.gartner.com.

Gause, Donald, and Gerald Weinberg. *Are Your Lights On? How to Figure Out What the Problem REALLY Is.* New York: Dorset House Publishing, 1990. Yes, this is not a recent book, but neither are the fundamentals of problem solving. Here's a short and easy-to-read book about general problem solving. You can probably read the entire book in one night, and it could profoundly improve your problem-solving potential as a systems analyst (or, for that matter, any other profession).

Levine, Martin. *Effective Problem Solving,* 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 1994. This is another older book, but as we stated before, problem-solving methods are timeless. At only 146 pages, this title can serve as an excellent professional reference.

Weinberg, Gerald. *Rethinking Systems Analysis and Design.* New York: Dorset House Publishing, 1988. Don't let the date fool you. This is one of the best and most important books on this subject ever written. This book may not teach any of the popular systems analysis and design methods of our day, but it challenges the reader to leap beyond those methods to consider something far more important—the people side of systems work. Dr. Weinberg's theories and concepts are presented in the context of dozens of delightful fables and short experiential stories. We are grateful to him for our favorite systems analysis fable of all time, "The Three Ostriches."


Strategic Enterprise Plan
Strategic Information Systems Plan
Goal: Improve Business KNOWLEDGE
Goal: Improve Business PROCESSES
Goal: Improve Business COMMUNICATIONS
PROJECT MANAGERS and SYSTEMS ANALYSTS
SYSTEM USERS
SYSTEM OWNERS
SYSTEM OWNERS
SYSTEM DESIGNERS
SYSTEM BUILDERS
STATEMENT OF WORK
PROBLEM STATEMENT (using the PIECES framework)
INFORMATION SCOPE & VISION
FUNCTIONAL SCOPE & VISION
COMMUNICATIONS SCOPE & VISION
SYSTEM IMPROVEMENT OBJECTIVES (using the PIECES framework)
BUSINESS REQUIREMENTS STATEMENT
BUSINESS DATA REQUIREMENTS
LOGICAL DATA MODELS
BUSINESS PROCESS REQUIREMENTS
LOGICAL PROCESS MODELS
BUSINESS & SYSTEM INTERFACE REQUIREMENTS
LOGICAL INTERFACE MODELS
SYSTEM PROPOSAL (or REQUEST FOR SYSTEM PROPOSALS)
APPLICATION ARCHITECTURE
DESIGN PROTOTYPES
PHYSICAL DATABASE DESIGN SPECIFICATIONS
BUSINESS PROCESS DESIGN
PHYSICAL SOFTWARE DESIGN SPECIFICATIONS
PHYSICAL USER & SYSTEM INTERFACE DESIGN SPECIFICATIONS
FUNCTIONAL SYSTEM
TRAINING MATERIALS
DATABASE SOLUTION
MIDDLEWARE
COMMERCIAL SOFTWARE PACKAGES
CUSTOM-BUILT APPLICATION SOFTWARE
MIDDLEWARE
USER INTERFACE SOLUTIONS
SYSTEM INTERFACE SOLUTIONS
OPERATIONAL SYSTEM
POST-AUDIT REVIEW
SCOPE DEFINITION
PROBLEM ANALYSIS
REQUIREMENTS ANALYSIS
LOGICAL DESIGN
DECISION ANALYSIS
PHYSICAL DESIGN
CONSTRUCTION & TESTING
INSTALLATION & DELIVERY
PROJECT and PROCESS MANAGEMENT
FEASIBILITY ANALYSIS and RISK MANAGEMENT
Constraint: APPROVED DATABASE TECHNOLOGIES
Constraint: APPROVED PROCESS TECHNOLOGIES
Constraint: APPROVED INTERFACE TECHNOLOGIES
Constraint: APPROVED NETWORK TECHNOLOGIES
Strategic Enterprise Information Technology Architecture

3 Information Systems Development

Chapter Preview and Objectives

This chapter more closely examines the systems development process that was first introduced in Chapter 1. Successful systems development is governed by some fundamental, underlying principles that we introduce in this chapter. We also introduce a basic, representative systems development methodology as a disciplined approach to developing information systems. Although such an approach will not guarantee success, it will greatly improve the chances of success. You will know that you understand information systems development when you can:

- Describe the motivation for a standard systems development process in terms of the Capability Maturity Model (CMM) for quality management.
- Differentiate between the system life cycle and a system development methodology.
- Describe 10 basic principles of systems development.
- Define problems, opportunities, and directives—the triggers for systems development projects.
- Describe the PIECES framework for categorizing problems, opportunities, and directives.
- Describe the essential phases of systems development. For each phase, describe its purpose, inputs, and outputs.
- Describe cross life-cycle activities that overlap multiple system development phases.
- Describe typical, alternative "routes" through the essential phases of systems development. Describe how routes may be combined or customized for different types of projects.
- Describe various automated tools for systems development.

Introduction

Work is getting underway at SoundStage Entertainment Club for the systems analysis and design of their member services information system. But the more Bob Martinez learns about the system, the more confused he gets. Bob can recall some of his programming assignments in college. Most of them were just a page or two of bulleted points listing required features. It was pretty easy to get your head around that. But the new SoundStage system will involve tracking member contacts and purchase requirements, promotions, sales, shipments, inventory, multiple warehouses, Web sites, and more. Bob wonders how they will even list all the requirements, let alone keep them straight. How will they know what data they need to track? How will they know what every piece of programming needs to do? He mentioned that to his boss, Sandra. She said it was all about following "the methodology." He remembered something about methodology from a systems analysis class. At the time it seemed like a lot of unnecessary work. But he is starting to see now that on a large project, following an established method may be the only path that is safe to travel.

The Process of Systems Development

systems development process a set of activities, methods, best practices, deliverables, and automated tools that stakeholders (from Chapter 1) use to develop and continuously improve information systems and software (from Chapters 1 and 2).

This chapter introduces a focus on information systems development. We will examine a **systems development process.** Notice we did not say "the" process—there are as many variations on the process as there are experts and authors. We will present one representative process and use it consistently throughout this book. The chapter home page shows an expanded number of phases compared to the home page of Chapter 1. This is because we have factored the high-level phases such as system analysis and system design from Chapter 1 into multiple phases and activities. We have also refined the size and place of the stakeholder roles to reflect "involvement" as opposed to emphasis or priority. And we have edited and enhanced the building blocks to indicate deliverables and artifacts of system development. All of these modifications will be explained in due time.

Why do organizations embrace standardized processes to develop information systems? Information systems are a complex product. Recall from Chapter 2 that an information system includes data, process, and communications building blocks and technologies that must serve the needs of a variety of stakeholders. Perhaps this explains why as many as 70 percent or more of information system development projects have failed to meet expectations, cost more than budgeted, and are delivered much later than promised. The Gartner Group suggests that "consistent adherence to moderately rigorous methodology guidelines can provide 70 percent of [systems development] organizations with a productivity improvement of at least 30 percent within two years."[1]

Increasingly, organizations have no choice but to adopt and follow a standardized systems development process. First, using a consistent process for systems development creates efficiencies that allow management to shift resources between projects. Second, a consistent methodology produces consistent documentation that reduces lifetime costs to maintain the systems. Finally, the U.S. government has mandated that any organization seeking to develop software for the government must adhere to certain quality management requirements. A consistent process promotes quality. And many other organizations have aggressively committed to total quality management goals in order to increase their competitive advantage. In order to realize quality and productivity improvements, many organizations have turned to project and process management frameworks such as the Capability Maturity Model, discussed in the next section.

[1]Richard Hunter, "AD Project Portfolio Management," *Proceedings of the Gartner Group IT98 Symposium/Expo* (CD-ROM). The Gartner Group is an industry watchdog and research group that tracks and projects industry trends for IT managers.

> The Capability Maturity Model

It has been shown that as an organization's information system development process matures, project timelines and cost decrease while productivity and quality increase. The Software Engineering Institute at Carnegie Mellon University has observed and measured this phenomenon and developed the **Capability Maturity Model (CMM)** to assist all organizations to achieve these benefits. The CMM has developed a wide following, both in industry and government. Software evaluation based on CMM is being used to qualify information technology contractors for most U.S. federal government projects.

Capability Maturity Model (CMM) a standardized framework for assessing the maturity level of an organization's information systems development and management processes and products. It consists of five levels of maturity.

The CMM framework for systems and software is intended to help organizations improve the maturity of their systems development processes. The CMM is organized into five maturity levels (see Figure 3-1):

- *Level 1—Initial:* This is sometimes called anarchy or chaos. At this level, system development projects follow no consistent process. Each development team uses its own tools and methods. Success or failure is usually a function of the skill and experience of the team. The process is unpredictable and not repeatable. A project typically encounters many crises and is frequently over budget and behind schedule. Documentation is sporadic or not consistent from one project to the next, thus creating problems for those who must maintain a system over its lifetime. Almost all organizations start at Level 1.
- *Level 2—Repeatable:* At this level, project management processes and practices are established to track project costs, schedules, and functionality. The focus is on project management. A system development process is always followed, but it may vary from project to project. Success or failure is still a function of the skill and experience of the project team; however, a concerted effort is made to repeat earlier project successes. Effective project management practices lay the foundation for standardized processes in the next level.

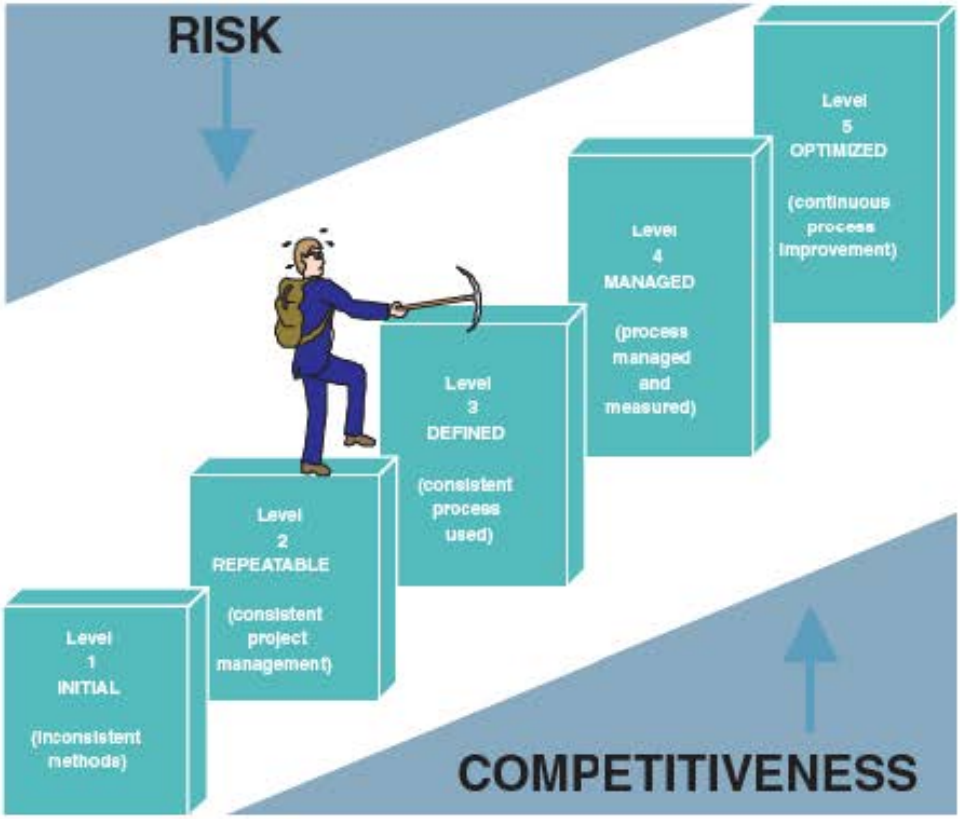


FIGURE 3-1
The Capability Maturity Model (CMM)

TABLE 3-1 Impact of System Development "Process" on Quality

CMM Project Statistics for a Project Resulting in 200,000 Lines of Code

Organization's CMM Level	Project Duration (months)	Project Person-Months	Number of Defects Shipped	Median Cost ($ millions)	Lowest Cost ($ millions)	Highest Cost ($ millions)
1	30	600	61	5.5	1.8	100+
2	18.5	143	12	1.3	.96	1.7
3	15	80	7	.728	.518	.933

Source: Master Systems, Inc.

system development methodology a formalized approach to the systems development process; a standardized process that includes the activities, methods, best practices, deliverables, and automated tools to be used for information systems development.

- *Level 3—Defined:* In this level, a standard **system development** process (sometimes called a *methodology*) is purchased or developed. All projects use a tailored version of this process to develop and maintain information systems and software. As a result of using the standardized process for all projects, each project results in consistent and high-quality documentation and deliverables. The process is stable, predictable, and repeatable.
- *Level 4—Managed:* In this level, measurable goals for quality and productivity are established. Detailed measures of the standard system development process and product quality are routinely collected and stored in a database. There is an effort to improve individual project management based on this collected data. Thus, management seeks to become more proactive than reactive to systems development problems (such as cost overruns, scope creep, schedule delays, etc.). Even when a project encounters unexpected problems or issues, the process can be adjusted based on predictable and measurable impacts.
- *Level 5—Optimizing:* In this level, the standardized system development process is continuously monitored and improved based on measures and data analysis established in Level 4. This can include changing the technology and best practices used to perform activities required in the standard system development process, as well as adjusting the process itself. Lessons learned are shared across the organization, with a special emphasis on eliminating inefficiencies in the system development process while sustaining quality. In summary, the organization has institutionalized continuous systems development process improvement.

It is very important to recognize that each level is a prerequisite for the next level.

Currently, many organizations are working hard to achieve at least CMM Level 3 (sometimes driven by a government or organizational mandate). A central theme to achieving Level 3 (Defined) is the use of a standard process or methodology to build or integrate systems. As shown in Table 3-1, an organization can realize significant improvements in schedule and cost by institutionalizing CMM Level 3 process improvements.[2]

system life cycle the factoring of the lifetime of an information system into two stages, (1) systems development and (2) systems operation and maintenance—first you build it; then you use and maintain it. Eventually, you cycle back to redevelopment of a new system.

> Life Cycle versus Methodology

The terms *system life cycle* and *system development methodology* are frequently and incorrectly interchanged. Most system development processes are derived from a natural **system life cycle.** The system life cycle just happens. Figure 3-2 illustrates two

[2]White Paper, "Rapidly Improving Process Maturity: Moving Up the Capability Maturity Model through Outsourcing" (Boston: Keane, 1997, 1998, p.11).

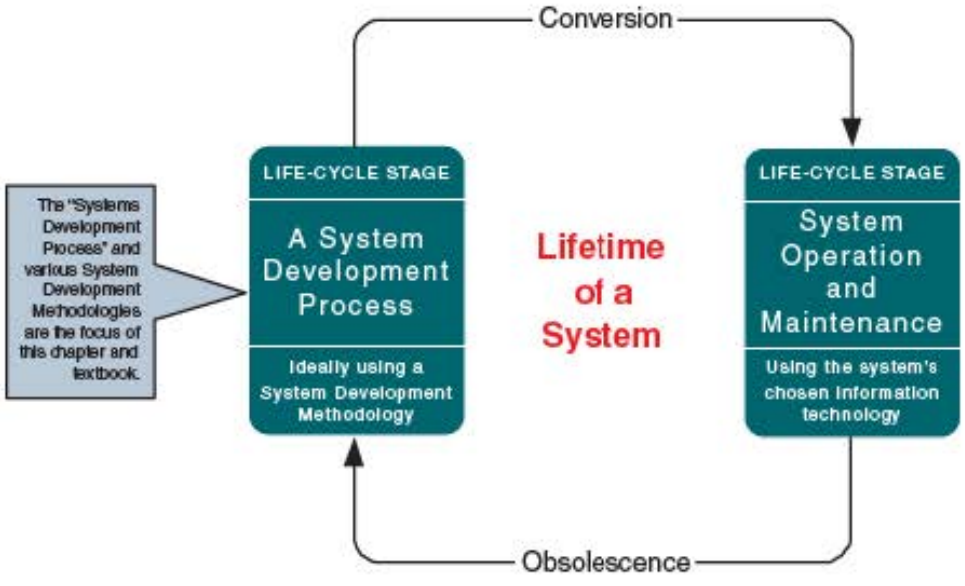


FIGURE 3-2
The System Life Cycle

life-cycle stages. Notice that there are two key events that trigger a change from one stage to the other:

- When a system cycles from development to operation and maintenance, a *conversion* must take place.
- At some point in time, *obsolescence* occurs (or is imminent) and a system cycles from operation and maintenance to redevelopment.

Actually, a system may be in more than one stage at the same time. For example, one version may be in *operation and support* while the next version is in *development.*

So how does this contrast with a systems development methodology? A systems development methodology "executes" the systems development stage of the system life cycle. Each individual information system has its own system life cycle. The methodology is the standard process to build and maintain that system and all other information systems through their life cycles. Consistent with the goals of the CMM, methodologies ensure that:

- A consistent, reproducible approach is applied to all projects.
- There is reduced risk associated with shortcuts and mistakes.
- Complete and consistent documentation is produced from one project to the next.
- Systems analysts, designers, and builders can be quickly reassigned between projects because all use the same process.
- As development teams and staff constantly change, the results of prior work can be easily retrieved and understood by those who follow.

Methodologies can be purchased or homegrown. Why purchase a methodology? Many information system organizations can't afford to dedicate staff to the development and continuous improvement of a homegrown methodology. Methodology vendors have a vested interest in keeping their methodologies current with the latest business and technology trends. Homegrown methodologies are usually based on generic methodologies and techniques that are well documented in books and on Web sites. Examples of system development methodologies are listed in the margin on the following page. You should be able to research most of them on the Web. Many of their underlying methods will be taught in this textbook.

Throughout this book, we will use a methodology called ***FAST,*** which stands for *F*ramework for the *A*pplication of *S*ystems *T*hinking. *FAST* is not a real-world commercial methodology. We developed it as a composite of the best practices we've

FAST a hypothetical methodology used throughout this book to demonstrate a representative systems development process. The acronym's letters stand for *F*ramework for the *A*pplication of *S*ystems *T*hinking.

REPRESENTATIVE SYSTEM DEVELOPMENT METHODOLOGIES

Architected Rapid Application Development (Architected RAD)
Dynamic Systems Development Methodology (DSDM)
Joint Application Development (JAD)
Information Engineering (IE)
Rapid Application Development (RAD)
Rational Unified Process (RUP)
Structured Analysis and Design (old, but still occasionally encountered)
eXtreme Programming (XP)
Note: There are many commercial methodologies and software tools (sometimes called *methodware*) based on the above general methodologies.

encountered in many commercial and reference methodologies. Unlike many commercial methodologies, *FAST* is not prescriptive. That is to say, *FAST* is an agile framework that is flexible enough to provide for different types of projects and strategies. Most important, *FAST* shares much in common with both the book-based and the commercial methodologies that you will encounter in practice.

> Underlying Principles for Systems Development

Before we examine the *FAST* methodology, let's introduce some general principles that should underlie all systems development methodologies.

Principle 1: Get the System Users Involved Analysts, programmers, and other information technology specialists frequently refer to "my system." This attitude has, in part, created an "us versus them" conflict between technical staff and their users and management. Although analysts and programmers work hard to create technologically impressive solutions, those solutions often backfire because they don't address the real organization problems. Sometimes they even introduce new organization problems. For this reason, system user involvement is an absolute necessity for successful systems development. Think of systems development as a partnership between system users, analysts, designers, and builders. The analysts, designers, and builders are responsible for systems development, but they must engage their owners and users, insist on their participation, and seek agreement from all stakeholders concerning decisions that may affect them.

Miscommunication and misunderstandings continue to be a significant problem in many systems development projects. However, owner and user involvement and education minimize such problems and help to win acceptance of new ideas and technological change. Because people tend to resist change, information technology is often viewed as a threat. The best way to counter that threat is through constant and thorough communication with owners and users.

Principle 2: Use a Problem-Solving Approach A system development methodology is, first and foremost, a problem-solving approach to building systems. The term *problem* is broadly used throughout this book to include (1) real problems, (2) opportunities for improvement, and (3) directives from management. The classical problem-solving approach is as follows:

1. Study and understand the problem, its context, and its impact.
2. Define the requirements that must be met by *any* solution.
3. Identify candidate solutions that fulfill the requirements, and select the best solution.
4. Design and/or implement the chosen solution.
5. Observe and evaluate the solution's impact, and refine the solution accordingly.

Systems analysts should approach all projects using some variation of this problem-solving approach.

Inexperienced or unsuccessful problem solvers tend to eliminate or abbreviate one or more of the above steps. For example, they fail to completely understand the problem, or they prematurely commit to the first solution they think of. The result can range from (1) solving the wrong problem, to (2) incorrectly solving the problem, (3) picking the wrong solution, or (4) picking a less-than-optimal solution. A methodology's problem-solving process, when correctly applied, can reduce or eliminate these risks.

Principle 3: Establish Phases and Activities All methodologies prescribe phases and activities. The number and scope of phases and activities vary from author to author, expert to expert, methodology to methodology, and business to business. The chapter home page at the beginning of this chapter illustrates the eight phases of our *FAST* methodology in the context of your information systems framework. The phases

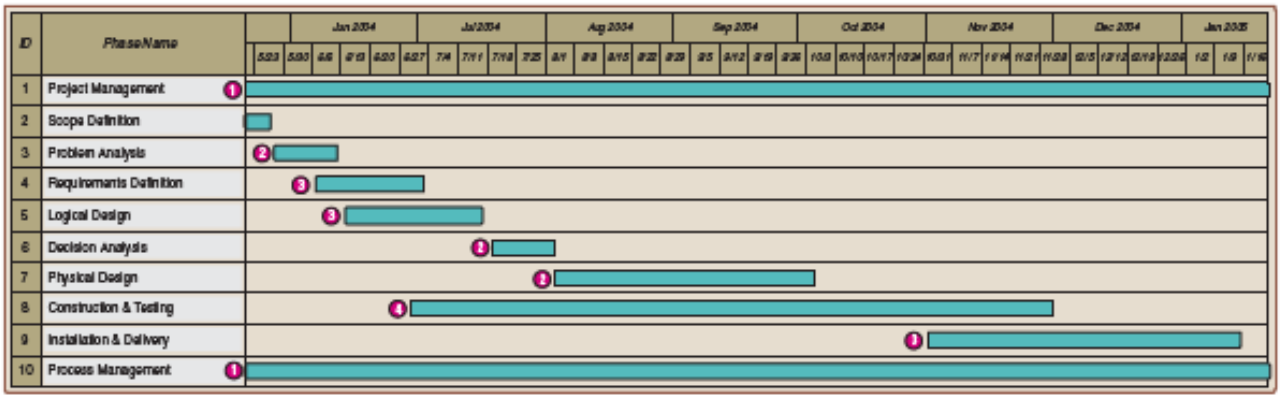


FIGURE 3-3 Overlap of System Development Phases and Activities

are listed on the far right-hand side of the illustration. In each phase, the focus is on those building blocks and on stakeholders that are aligned to the left of that phase.

The phases are: scope definition, problem analysis, requirements analysis, logical design, decision analysis, physical design and integration, construction and testing, and installation and delivery. Each of these phases will be discussed later in this chapter. These phases are not absolutely sequential. The phases tend to overlap one another, as illustrated in Figure 3-3. Also, the phases may be customized to the special needs of a given project (e.g., deadlines, complexity, strategy, resources). In this chapter, we will describe each customization as alternative routes through the methodology and problem-solving process.

Principle 4: Document throughout Development When do you document the programs you write? Be honest. We must confess that, like most students, we did our documentation after we wrote the programs. We knew better, but we postdocumented anyway. That just does not work in the business world. In medium to large organizations, system owners, users, analysts, designers, and builders come and go. Some will be promoted; some will have extended medical leaves; some will quit the organization; and still others will be reassigned. To promote good communication between constantly changing stakeholders, documentation should be a working by-product of the entire systems development effort.

Documentation enhances communications and acceptance. Documentation reveals strengths and weaknesses of the system to multiple stakeholders. It stimulates user involvement and reassures management about progress. At the same time, some methodologies have been criticized for expecting too much documentation that adds little value to the process or resulting system. Our *FAST* methodology advocates a balance between the value of documentation and the effort to produce it. Experts call this *agile modeling*.

Principle 5: Establish Standards In a perfect world, all information systems would be integrated such that they behave as a single system. Unfortunately, this never happens because information systems are developed and replaced over a very long period of time. Even organizations that purchase and install an enterprise resource planning (ERP) product usually discover that there are applications and needs that fall outside the ERP system. Systems integration has become critical to the success of any organization's information systems.

To achieve or improve systems integration, organizations turn to standards. In many organizations, these standards take the form of enterprise information technology architecture. An IT architecture sets standards that serve to direct technology solutions and information systems to a common technology vision or configuration.

An information technology architecture typically standardizes on the following (note: it is not important that you know what all these sample technologies are):

- *Database technology*—What database engine(s) will be used (e.g., *Oracle,* IBM *DB2,* Microsoft *SQL Server*)? On what platforms will they be operated (e.g., *UNIX, Linux, Windows XP, MVS*)? What technologies will be used to load data into online transaction processing (OLTP) databases, operational data stores, and data warehouses (i.e., Extract Transform and Load [ETL])?
- *Software technology*—What application development environment(s)/language(s) will be used to write software (e.g., IBM's *Websphere* with *Java,* Microsoft's *Visual Studio .NET* with *Visual Basic .NET, Visual C++,* and/or *Visual C#;* Syebase's *Powerbuilder,* Oracle's *Oracle Forms*)?
- *Interface technology*—How will user interfaces be developed—with *MS Windows* components or Web languages and components (e.g., an *xHTML* editor such as Macromedia's *Dreamweaver,* a portal engine such as IBM's *Websphere*)? How will data be exchanged between different information systems (e.g., a data broker such as IBM's *MQ Messaging,* an XML-based data exchange, or a custom programmed interface)?

Notice how these architectural questions closely correspond to the technology drivers in your information system model.

In the absence of an IT architecture, each information system and application may be built using radically different technologies. Not only does this make it difficult to integrate applications, but it creates resource management problems—IT organizations cannot as easily move developers between projects as priorities change or emergencies occur because different teams are staffed with technical competencies based on the various technologies used and being used to develop information systems. Creating an enterprise IT architecture and driving projects and teams to that architecture make more sense.

As new technologies emerge, an IT architecture must change. But that change can be managed. The chief technology officer (CTO) in an organization is frequently charged with technology exploration and IT architecture management. Given that architecture, all information systems projects are constrained to implement new systems that conform to the architecture (unless otherwise approved by the CTO).

Principle 6: Manage the Process and Projects Most organizations have a system development process or methodology, but they do not always use it consistently on projects. Both the process and the projects that use it must be managed. **Process management** ensures that an organization's chosen process or management is used consistently on and across all projects. Process management also defines and improves the chosen process or methodology over time. **Project management** ensures that an information system is developed at minimum cost, within a specified time frame, and with acceptable quality (using the standard system development process or methodology). Effective project management is essential to achieving CMM Level 2 success. Use of a repeatable process gets us to CMM Level 3. CMM Levels 4 and 5 require effective process management. Project management can occur without a standard process, but in mature organizations all projects are based on a standardized and managed process.

process management an ongoing activity that documents, teaches, oversees the use of, and improves an organization's chosen methodology (the "process") for systems development. Process management is concerned with phases, activities, deliverables, and quality standards that should be consistently applied to all projects.

project management the process of scoping, planning, staffing, organizing, directing, and controlling a project to develop an information system at minimum cost, within a specified time frame, and with acceptable quality.

Process management and project management are influenced by the need for quality management. Quality standards are built into a process to ensure that the activities and deliverables of each phase will contribute to the development of a high-quality information system. They reduce the likelihood of missed problems and requirements, as well as flawed designs and program errors (bugs). Standards also make the IT organization more agile. As personnel changes occur, staff can be relocated between projects with the assurance that every project is following an understood and accepted process.

Principle 7: Justify Information Systems as Capital Investments Information systems are capital investments, just like a fleet of trucks or a new building. System owners commit to this investment. Notice that the initial commitment occurs early in a project, when system owners agree to sponsor and fund the project. Later (during the phase called *decision analysis*), system owners recommit to the more costly technical decisions. In considering a capital investment, two issues must be addressed:

1. For any problem, there are likely to be several possible solutions. The systems analyst and other stakeholders should not blindly accept the first solution suggested. The analyst who fails to look at alternatives may be doing the business a disservice.
2. After identifying alternative solutions, the systems analyst should evaluate each possible solution for feasibility, especially for **cost-effectiveness.** Cost-effectiveness is measured using a technique called *cost-benefit analysis.*

Like project and process management, cost-benefit analysis is performed throughout the system development process.

A significant advantage of the phased approach to systems development is that it provides several opportunities to reevaluate cost-effectiveness, risk, and feasibility. There is often a temptation to continue with a project only because of the investment already made. In the long run, canceled projects are usually much less costly than implemented disasters. This is extremely important for young analysts to remember.

Most system owners want more from their systems than they can afford or are willing to pay for. Furthermore, the scope of most information system projects increases as the analyst learns more about the business problems and requirements as the project progresses. Unfortunately, most analysts fail to adjust estimated costs and schedules as the scope increases. As a result, the analyst frequently and needlessly accepts responsibility for cost and schedule overruns.

Because information systems are recognized as capital investments, system development projects are often driven by enterprise planning. Many contemporary information technology business units create and maintain a **strategic information systems plan.** Such a plan identifies and prioritizes information system development projects. Ideally, a strategic information systems plan is driven by a **strategic enterprise plan** that charts a course for the entire business.

cost-effectiveness the result obtained by striking a balance between the lifetime costs of developing, maintaining, and operating an information system and the benefits derived from that system. Cost-effectiveness is measured by cost-benefit analysis.

strategic information systems plan a formal strategic plan (3 to 5 years) for building and improving an information technology infrastructure and the information system applications that use that infrastructure.

strategic enterprise plan a formal strategic plan (3 to 5 years) for an entire business that defines its mission, vision, goals, strategies, benchmarks, and measures of progress and achievement. Usually, the strategic enterprise plan is complemented by strategic business unit plans that define how each business unit will contribute to the enterprise plan. The information systems plan (above) is one of those unit-level plans.

Principle 8: Don't Be Afraid to Cancel or Revise Scope There is an old saying: "Don't throw good money after bad." In other words, don't be afraid to cancel a project or revise scope, regardless of how much money has been spent so far—cut your losses. To this end, we advocate a **creeping commitment** approach to systems development.[3] With the creeping commitment approach, multiple feasibility checkpoints are built into any systems development methodology. At each checkpoint feasibility is reassessed. All previously expended costs are considered sunk (meaning not recoverable). They are, therefore, irrelevant to the decision. Thus, the project should be reevaluated at each checkpoint to determine if it remains feasible to continue investing time, effort, and resources into the project. At each checkpoint, the analyst should consider the following options:

- Cancel the project if it is no longer feasible.
- Reevaluate and adjust the costs and schedule if project scope is to be increased.
- Reduce the scope if the project budget and schedule are frozen and not sufficient to cover all project objectives.

The concept of sunk costs is more or less familiar to most financial analysts, but it is frequently forgotten or not used by the majority of systems analysts, most system users, and even many system owners.

creeping commitment a strategy in which feasibility and risks are continuously reevaluated throughout a project. Project budgets and deadlines are adjusted accordingly.

[3]Thomas Gildersleeve, *Successful Data Processing Systems Analysis*, 2nd ed. (Englewood Cliffs, NJ: Prentice Hall, 1985), pp. 5–7.

risk management the process of identifying, evaluating, and controlling what might go wrong in a project before it becomes a threat to the successful completion of the project or implementation of the information system. Risk management is driven by risk analysis or assessment.

In addition to managing feasibility throughout the project, we must manage risk. **Risk management** seeks to balance risk and reward. Different organizations are more or less averse to risk, meaning that some are willing to take greater risks than others in order to achieve greater rewards.

Principle 9: Divide and Conquer Whether you realize it or not, you learned the divide-and-conquer approach throughout your education. Since high school, you've been taught to outline a paper before you write it. Outlining is a divide-and-conquer approach to writing. A similar approach is used in systems development. We divide a system into subsystems and components in order to more easily conquer the problem and build the larger system. In systems analysis, we often call this *factoring*. By repeatedly dividing a larger problem (system) into more easily managed pieces (subsystems), the analyst can simplify the problem-solving process. This divide-and-conquer approach also complements communication and project management by allowing different pieces of the system to be communicated to different and the most appropriate stakeholders.

The building blocks of your information system framework provide one basis for dividing and conquering an information system's development. We will use this framework throughout the book.

Principle 10: Design Systems for Growth and Change Businesses change over time. Their needs change. Their priorities change. Accordingly, information systems that support a business must change over time. For this reason, good methodologies should embrace the reality of change. Systems should be designed to accommodate both growth and changing requirements. In other words, well-designed information systems can both scale up and adapt to the business. But regardless of how well we design systems for growth and change, there will always come a time when they simply cannot support the business.

System scientists describe the natural and inevitable decay of all systems over time as *entropy*. As described earlier in this section, after a system is implemented it enters the *operations and maintenance* stage of the life cycle. During this stage the analyst encounters the need for changes that range from correcting simple mistakes, to redesigning the system to accommodate changing technology, to making modifications to support changing user requirements. Such changes direct the analyst and programmers to rework formerly completed phases of the life cycle. Eventually, the cost of maintaining the current system exceeds the costs of developing a replacement system—the current system has reached entropy and becomes obsolete.

But system entropy can be managed. Today's tools and techniques make it possible to design systems that can grow and change as requirements grow and change. This book will teach you many of those tools and techniques. For now, it's more important to simply recognize that flexibility and adaptability do not happen by accident—they must be built into a system.

We have presented 10 principles that should underlie any methodology. These principles are summarized in the margin and can be used to evaluate any methodology, including our *FAST* methodology.

PRINCIPLES OF SYSTEMS DEVELOPMENT

Get the System Users Involved.
Use a Problem-Solving Approach.
Establish Phases and Activities.
Document throughout Development.
Establish Standards.
Manage the Process and Projects.
Justify Information Systems as Capital Investments.
Don't Be Afraid to Cancel or Revise Scope.
Divide and Conquer.
Design Systems for Growth and Change.

A Systems Development Process

In this section we'll examine a logical process for systems development. (Reminder: *FAST* is a hypothetical methodology created for learning purposes.) We'll begin by studying types of system projects and how they get started. Then we'll introduce the eight *FAST* phases. Finally, we'll examine alternative variations, or "routes" through the phases, for different types of projects and development strategies.

> Where Do Systems Development Projects Come From?

System owners and system users initiate most projects. The impetus for most projects is some combination of **problems, opportunities,** and **directives.** To simplify this discussion, we will frequently use the term *problem* to collectively refer to problems, opportunities, and directives. Accordingly, *problem solving* refers to solving problems, exploiting opportunities, and fulfilling directives.

problem an undesirable situation that prevents the organization from fully achieving its mission, vision, goals, and/or objectives.

There are far too many potential system problems to list them all in this book. However, James Wetherbe developed a useful framework for classifying problems.[4] He calls it *PIECES* because the letters of each of the six categories, when put together, spell the word "pieces." The categories are:

opportunity a chance to improve the organization even in the absence of an identified problem.

- **P** the need to correct or improve *performance.*
- **I** the need to correct or improve *information* (and data).
- **E** the need to correct or improve *economics,* control costs, or increase profits.
- **C** the need to correct or improve *control* or security.
- **E** the need to correct or improve *efficiency* of people and processes.
- **S** the need to correct or improve *service* to customers, suppliers, partners, employees, and so on.

directive a new requirement that's imposed by management, government, or some external influence.

Figure 3-4 expands on each of the PIECES categories.

The categories of the PIECES framework are neither exhaustive nor mutually exclusive—they overlap. Any given project is usually characterized by one or more categories, and any given problem or opportunity may have implications with respect to more than one category. But PIECES is a practical framework (used in *FAST*), not just an academic exercise. We'll revisit PIECES several times in this book.

Projects can be either planned or unplanned. A *planned project* is the result of one of the following:

- An *information systems strategy plan* has examined the business as a whole to identify those system development projects that will return the greatest strategic (long-term) value to the business.
- A *business process redesign* has thoroughly analyzed a series of business processes to eliminate redundancy and bureaucracy and to improve efficiency and value added. Now it is time to redesign the supporting information system for those redesigned business processes.

The opposite of planned projects are *unplanned projects*—those that are triggered by a specific problem, opportunity, or directive that occurs in the course of doing business. Most organizations have no shortage of unplanned projects. Anyone can submit a proposed project based on something that is happening in the business. The number of unplanned-project proposals can easily overwhelm the largest information systems organization; therefore, they are frequently screened and prioritized by a **steering committee** of system owners and IT managers to determine which requests get approved. Those requests that are not approved are **backlogged** until resources become available (which sometimes never happens).

steering committee an administrative body of system owners and information technology executives that prioritizes and approves candidate system development projects.

Both planned and unplanned projects go through the same essential system development process. Let's now examine the project phases in somewhat greater detail.

> The *FAST* Phases

FAST, like most methodologies, consists of phases. The number of phases will vary from one methodology to another. In Chapter 1 you were introduced to the four classic phases of the system development life cycle. The *FAST* methodology employs

backlog a repository of project proposals that cannot be funded or staffed because they are a lower priority than those that have been approved for system development. Note that priorities change over time; therefore, a backlogged project might be approved at some future date.

[4] James Wetherbe and Nicholas P. Vitalari, *Systems Analysis and Design: Traditional, Best Practices,* 4th ed. (St. Paul, MN: West Publishing, 1994), pp. 196–199. James Wetherbe is a respected information systems educator, researcher, and consultant.

The PIECES Problem-Solving Framework and Checklist

The following checklist for problem, opportunity, and directive identification uses Wetherbe's PIECES framework. Note that the categories of PIECES are not mutually exclusive; some possible problems show up in multiple lists. Also, the list of possible problems is not exhaustive. The PIECES framework is equally suited to analyzing both manual and computerized systems and applications.

PERFORMANCE

A. Throughput – the amount of work performed over some period of time.
B. Response times – the average delay between a transaction or request, and a response to that transaction or request.

INFORMATION (and Data)

A. Outputs
   1. Lack of any information
   2. Lack of necessary information
   3. Lack of relevant information
   4. Too much information – "information overload"
   5. Information that is not in a useful format
   6. Information that is not accurate
   7. Information that is difficult to produce
   8. Information is not timely to its subsequent use
B. Inputs
   1. Data is not captured
   2. Data is not captured in time to be useful
   3. Data is not accurately captured – contains errors
   4. Data is difficult to capture
   5. Data is captured redundantly – same data captured more than once
   6. Too much data is captured
   7. Illegal data is captured
C. Stored data
   1. Data is stored redundantly in multiple files and/or databases
   2. Same data items have different values in different files (poor data integration)
   3. Stored data is not accurate
   4. Data is not secure to accident or vandalism
   5. Data is not well organized
   6. Data is not flexible – not easy to meet new information needs from stored data
   7. Data is not accessible

ECONOMICS

A. Costs
   1. Costs are unknown
   2. Costs are untraceable to source
   3. Costs are too high
B. Profits
   1. New markets can be explored
   2. Current marketing can be improved
   3. Orders can be increased

CONTROL (and Security)

A. Too little security or control
   1. Input data is not adequately edited
   2. Crimes (e.g., fraud, embezzlement) are (or can be) committed against data
   3. Ethics are breached on data or information – refers to data or information getting to unauthorized people
   4. Redundantly stored data is inconsistent in different files or databases
   5. Data privacy regulations or guidelines are being (or can be) violated
   6. Processing errors are occurring (either by people, machines, or software)
   7. Decision-making errors are occurring
B. Too much control or security
   1. Bureaucratic red tape slows the system
   2. Controls inconvenience customers or employees
   3. Excessive controls cause processing delays

EFFICIENCY

A. People, machines, or computers waste time
   1. Data is redundantly input or copied
   2. Data is redundantly processed
   3. Information is redundantly generated
B. People, machines, or computers waste materials and supplies
C. Effort required for tasks is excessive
D. Material required for tasks is excessive

SERVICE

A. The system produces inaccurate results
B. The system produces inconsistent results
C. The system produces unreliable results
D. The system is not easy to learn
E. The system is not easy to use
F. The system is awkward to use
G. The system is inflexible to new or exceptional situations
H. The system is inflexible to change
I. The system is incompatible with other systems

FIGURE 3-4 The PIECES Framework for Problem Identification

eight phases to better define periodic milestones and the deliverables. The grid below compares the *FAST* phases to the classic phases. As you can see, both sets of phases cover the same ground, but *FAST* is more detailed.

FAST Phases	Classic Phases			
	Project Initiation	System Analysis	System Design	System Implementation
Scope definition	X			
Problem analysis	X	X		
Requirements analysis		X		
Logical design		X		
Decision analysis	(a system analysis transition phase)			
Physical design and integration			X	
Construction and testing			X	X
Installation and delivery				X

Figure 3-5 illustrates the phases of the *FAST* methodology. Each phase produces deliverables that are passed to the next phase. And documentation accumulates as you complete each phase. Notice that we have included an iconic representation of the building blocks to symbolize this accumulation of knowledge and work-in-process artifacts during system development. Notice also that a project starts with some combination of PROBLEMS, OPPORTUNITIES, and DIRECTIVES from the user community (the green arrow) and finishes with a WORKING BUSINESS SOLUTION (the red arrow) for the user community.

Figure 3-6 shows the *FAST* methodology from the perspective of your information system building blocks that you learned in Chapters 1 and 2. The phases are on the right-hand side. The deliverables are built around the building blocks for knowledge, processes, and communications. The stakeholders are on the left-hand side. Notice how we have expanded and duplicated some stakeholders to reflect their involvement opposite the phases in which they primarily participate.

> NOTE: The remainder of this section briefly describes each of the eight basic phases. Throughout this discussion, we will be referring to the process flowchart in Figure 3-5, as well as the building blocks view of the process in Figure 3-6. Also throughout the discussion, all terms printed in SMALL CAPS refer to phases, prerequisites (inputs), and deliverables (outputs) shown in Figures 3-5 and 3-6.

Scope Definition The first phase of a typical project is SCOPE DEFINITION. The purpose of the scope definition phase is twofold. First, it answers the question, "Is this problem worth looking at?" Second, and assuming the problem *is* worth looking at, it establishes the size and boundaries of the project, the project vision, any constraints or limitations, the required project participants, and, finally, the budget and schedule.

In Figure 3-6, we see that the participants in the scope definition phase primarily include SYSTEM OWNERS, PROJECT MANAGERS, and SYSTEM ANALYSTS. System users are generally excluded because it is too early to get into the level of detail they will eventually bring to the project.

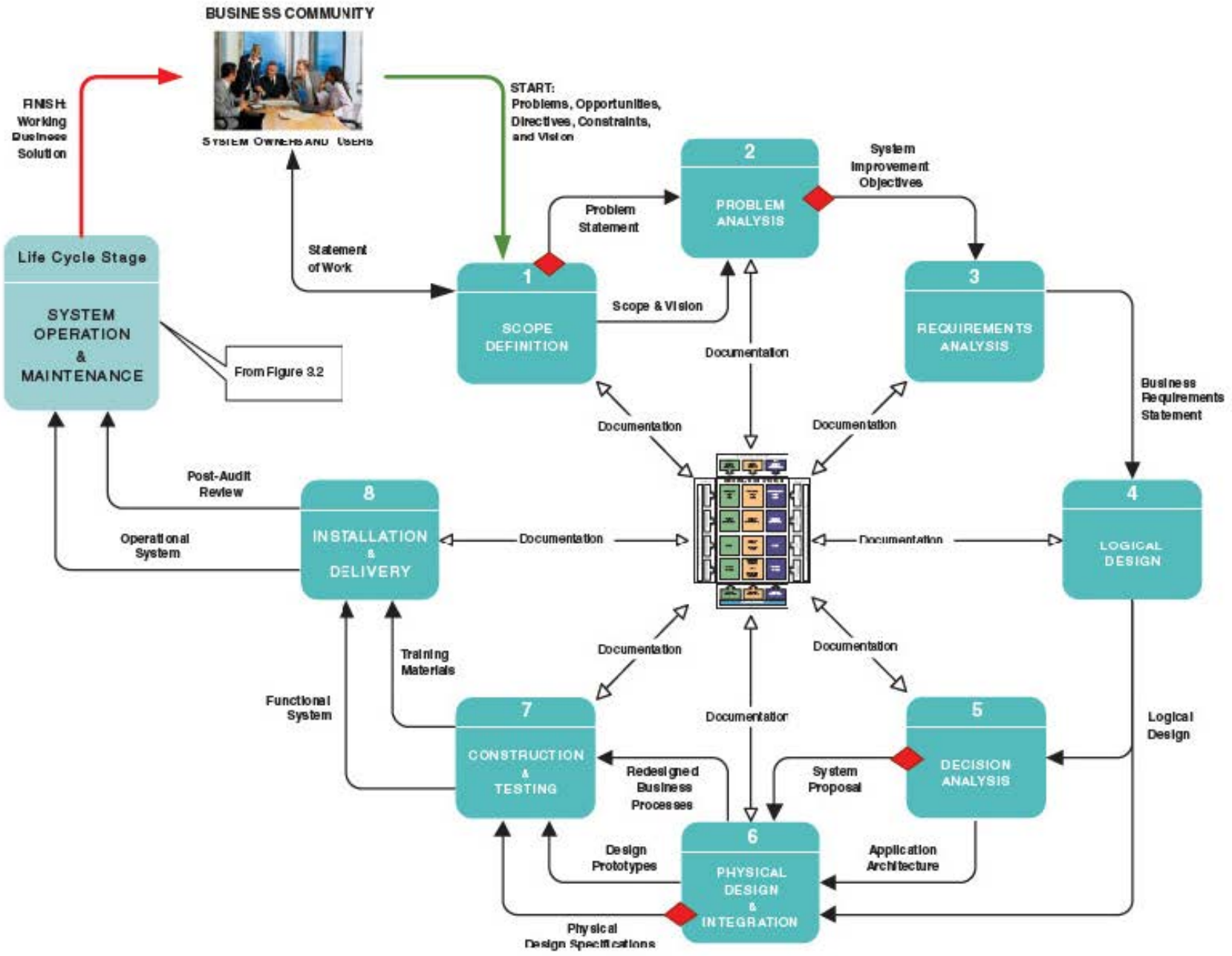


FIGURE 3-5 Process View of System Development

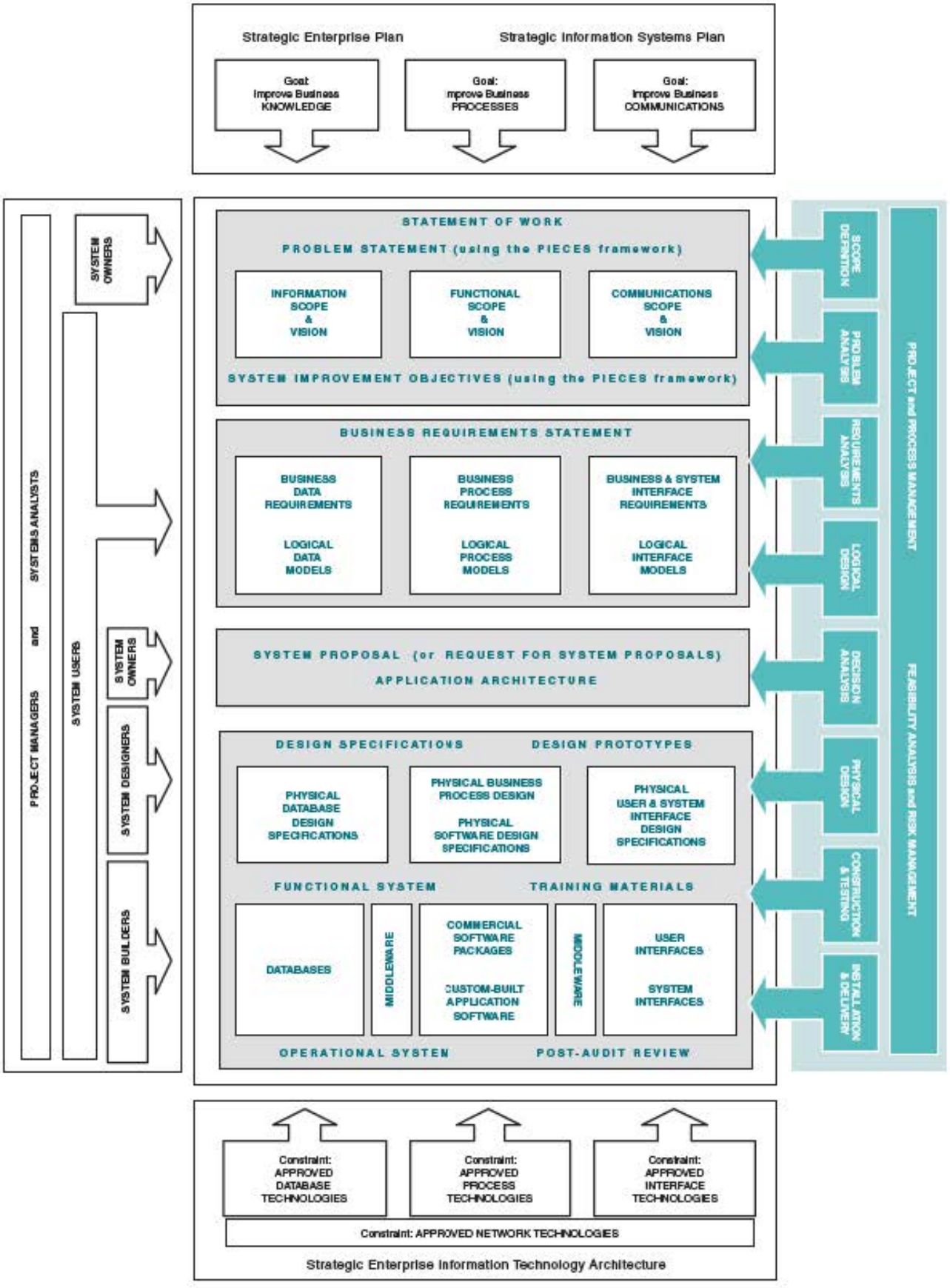


FIGURE 3-6 Building Blocks View of System Development

problem statement a statement and categorization of problems, opportunities, and directives; may also include constraints and an initial vision for the solution. Synonyms include *preliminary study and feasibility assessment.*

In Figure 3-5, we see that the scope definition phase is triggered by some combination of PROBLEMS, OPPORTUNITIES, and DIRECTIVES (to which we will add CONSTRAINTS and VISION). There are several deliverables or outcomes of a scope definition. One important outcome is a PROBLEM STATEMENT, a succinct overview of the problems, opportunities, and/or directives that triggered the project. The PIECES framework provides an excellent outline for a **problem statement.** The goal here is not to solve the problems, opportunities, and directives but only to catalog and categorize them. We should also identify any **constraints** that may impact the proposed project. Examples of constraints include budget limits, deadlines, human resources available or not available, business policies or government regulations, and technology standards. Finally, the system owners should be asked for at least a high-level vision for the system improvements they are seeking

constraint any factor, limitation, or restraint that may limit a solution or the problem-solving process.

Given a basic understanding of problems, opportunities, directives, constraints, and vision, we need to establish initial scope. Thus, an initial SCOPE STATEMENT is another important outcome of this phase. Scope defines how big we think the project is. Your information system building blocks provide a useful framework for defining scope. Figure 3-6 illustrates that scope and vision can be defined in terms of INFORMATION, FUNCTIONS, and INTERFACES. Scope can, and frequently does, change during a project. But by documenting initial scope, you establish a baseline for controlling **scope creep** on both the budget and the schedule.

scope creep a common phenomenon wherein the requirements and expectations of a project increase, often without regard to the impact on budget and schedule.

Given the initial problem and scope statements for the project, the analyst can staff the project team, estimate the budget for system development, and prepare a schedule for the remaining phases. Ultimately, this phase concludes with a "go or no-go" decision from system owners. Either the system owners agree with the proposed scope, budget, and schedule for the project, or they must reduce scope (to reduce costs and time) or cancel the project. This feasibility checkpoint is illustrated in Figure 3-5 as a diamond.

statement of work a contract with management and the user community to develop or enhance an information system; defines vision, scope, constraints, high-level user requirements, schedule, and budget. Synonyms include *project charter, project plan,* and *service-level agreement.*

The final and most important deliverable is a STATEMENT OF WORK. A **statement of work** is a contract or agreement to develop the information system. It consolidates the problem statement, scope statement, and schedule and budget for all parties who will be involved in the project.

Problem Analysis There is always an existing system, regardless of whether it currently uses information technology. The PROBLEM ANALYSIS phase studies the existing system and analyzes the findings to provide the project team with a more thorough understanding of the problems that triggered the project. The analyst frequently uncovers new problems and answers the most important question, "Will the benefits of solving these problems exceed the costs of building the system to solve these problems?"

Once again, Figure 3-6 provides a graphical overview of the problem analysis phase in terms of your information system building blocks. Notice that the participants still include the SYSTEM OWNERS but that this phase begins to actively involve the SYSTEM USERS as well. The system users are the business subject matter experts in any project. (Notice the intentional expansion of the system users' perspective to overlap many phases—remember principle 1: "Get the system users involved.") Of course, PROJECT MANAGERS and SYSTEM ANALYSTS are always involved in all phases of a project.

As shown in Figure 3-5, the prerequisites for the problem analysis phase are the SCOPE and PROBLEM STATEMENTS as defined and approved in the scope definition phase. The deliverable of the problem analysis phase is a set of SYSTEM IMPROVEMENT OBJECTIVES derived from a thorough understanding of the business problems. These objectives do not define inputs, outputs, or processes. Instead, they define the business criteria on which any new system will be evaluated. For instance, we might define a system improvement objective as any of the following:

Reduce the time between order processing and shipping by three days.
Reduce bad credit losses by 45 percent.
Comply with new financial aid federal qualification requirements by January 1.

Think of system improvement objectives as the *grading criteria* for evaluating any new system that you might eventually design and implement. System improvement objectives may be presented to system owners and users as a written recommendation or an oral presentation.

Depending on the complexity of the problem and the project schedule, the team may or may not choose to formally document the existing system. Such documentation frequently occurs when the business processes are considered dated or overly bureaucratic. Documentation of the existing system is sometimes called an "AS IS" BUSINESS MODEL. The as-is model may be accompanied by analysis demonstrating inefficiencies, bottlenecks, or other problems related to the business processes.

Every existing system has its own terminology, history, culture, and nuances. Learning those aspects of the system is an important by-product of this phase. From all of the information gathered, the project team gains a better understanding of the existing system's problems and opportunities. After reviewing the findings, the system owners will either agree or disagree with the recommended system improvement objectives. And consistent with the creeping commitment principle, we include another go or no-go feasibility checkpoint (the red diamond) at the end of the phase. The project can be either:

- Canceled if the problems are deemed no longer worth solving.
- Approved to continue to the next phase.
- Reduced or expanded in scope (with budget and schedule modifications) and then approved to continue to the next phase.

Requirements Analysis Given system owner approval to continue from the problem analysis phase, now you can design a new system, right? No, not yet! What capabilities should the new system provide for its users? What data must be captured and stored? What performance level is expected? Careful! This requires decisions about *what* the system must do, *not how* it should do those things. The REQUIREMENTS ANALYSIS phase defines and prioritizes the *business* requirements. Simply stated, the analyst approaches the users to find out what they need or want out of the new system, carefully avoiding any discussion of technology or technical implementation. This is perhaps the most important phase of systems development. Errors and omissions in requirements analysis result in user dissatisfaction with the final system and costly modifications.

Returning again to Figure 3-6, notice that the participants primarily include both SYSTEM USERS (which may include owners who will actually *use* the system) and SYSTEMS ANALYSTS. PROJECT MANAGERS are also involved. SYSTEM DESIGNERS are omitted from this phase in order to prevent premature attention to technology solutions. The building blocks can themselves provide the framework for defining many business requirements, including BUSINESS DATA REQUIREMENTS, BUSINESS PROCESS REQUIREMENTS, and BUSINESS AND SYSTEM INTERFACE REQUIREMENTS. Because the business requirements are intended to solve problems, the PIECES framework can also provide a useful outline, this time for a requirements statement.

In Figure 3-5, we see that the SYSTEM IMPROVEMENT OBJECTIVES from the problem analysis phase are the prerequisite to the requirements analysis phase. The deliverable is a BUSINESS REQUIREMENTS STATEMENT. Again, this requirements statement does not specify any technical possibilities or solutions. The requirements statement may be a document as small as a few pages, or it may be extensive with a page or more of documentation per requirement.

To produce a business requirements statement, the systems analyst works closely with system users to identify needs and priorities. This information is collected by way of interviews, questionnaires, and facilitated meetings. The challenge to the team is to validate those requirements. The system improvement objectives provide the "grading key" for business requirements: *Does each requirement contribute to meeting one or more system improvement objectives?* Chapters 6

and 7 will introduce systems analysis tools and techniques for identifying and documenting user requirements.

Typically, requirements must also be prioritized. Priorities serve two purposes. First, if project timelines become stressed, requirements priorities can be used to rescope the project. Second, priorities can frequently be used to define iterations of design and construction to create staged releases or versions of the final product.

The requirements analysis phase should never be skipped or shortchanged. One of the most common complaints about new systems and applications is that they don't really satisfy the users' needs. This usually happens when system designers and builders become preoccupied with a technical solution before fully understanding the business needs. System designers and builders are dependent on competent systems analysts to work with users to define and document complete and accurate business requirements before applying any technology.

system model a picture of a system that represents reality or a desired reality. System models facilitate improved communication between system users, system analysts, system designers, and system builders.

Logical Design Business requirements (above) are usually expressed in words. Systems analysts have found it useful to translate those words into pictures called **system models** to validate the requirements for completeness and consistency. (Figure 3-5 is an example of a common system model called a *data flow diagram.*) System modeling implements a timeless concept: "A picture is worth a thousand words."

The LOGICAL DESIGN PHASE translates business requirements into system models. The term **logical design** should be interpreted as "technology independent," meaning the pictures illustrate the system independent of any possible technical solution—hence, they model business requirements that must be fulfilled by any technical solution we might want to consider.

logical design the translation of business user requirements into a system model that depicts only the business requirements and not any possible technical design or implementation of those requirements. Common synonyms include *conceptual design* and *essential design*, the latter of which refers to modeling the "essence" of a system, or the "essential requirements" independent of any technology. The antonym of logical design is *physical design* (defined later in this chapter).

Different methodologies require or recommend different amounts and degrees of system modeling or logical design. Prescriptive methodologies like *structured analysis and design, information engineering,* and the *Rational Unified Process (RUP)* usually require that many types and/or instances of system models be drawn in various levels of detail. Fortunately, computer-automated tools are available to assist the systems analyst in these drawing tasks. Alternatively, agile methodologies like *architected rapid application development* and *extreme programming* recommend "just enough modeling." This so-called *agile modeling* seeks to prevent the project from degenerating into a condition called **analysis paralysis.** This textbook leans toward agile methods but recognizes that complex problems may best be solved using more prescriptive approaches.

analysis paralysis a satirical term coined to describe a common project condition in which excessive system modeling dramatically slows progress toward implementation of the intended system solution.

In Figure 3-6, we see that the participants include SYSTEM ANALYSTS (who draw the models) and SYSTEM USERS (who validate the models). PROJECT MANAGERS are always included to ensure that modeling meets standards and does not deter overall project progress. We can draw (1) LOGICAL DATA MODELS that depict data and information requirements, (2) LOGICAL PROCESS MODELS that depict business processes requirements, and (3) LOGICAL INTERFACE MODELS that depict business and system interface requirements.[5]

In Figure 3-5, we see that the prerequisite to logical design is the BUSINESS REQUIREMENTS STATEMENT from the previous phase. In practice, the requirements analysis and logical design phases almost always have considerable overlap. In other words, as business requirements are identified and documented, they can be modeled. The deliverables of logical design are the LOGICAL SYSTEM MODELS AND SPECIFICATIONS themselves. Depending on the methodology used, the level of detail in the specifications will vary. For example, we may define a business rule that specifies the legitimate values for a data attribute such as *Credit Rating* or a rule that specifies the business policy for a *Credit Check.*

[5] Those of you already familiar with *object-oriented* modeling should note that object models tend to blur the boundaries of our framework somewhat, but the framework can still be applied since the problem to be solved is still driven by the three fundamental business goals illustrated in our framework. This will be demonstrated in the object-oriented analysis and design chapters of this book.

Before we move on to the next phase, we should note that the SCOPE DEFINITION, PROBLEM ANALYSIS, REQUIREMENTS ANALYSIS, and LOGICAL DESIGN PHASES are collectively recognized by most experts as *system analysis*. Some experts would also include our next phase, DECISION ANALYSIS. But we consider it to be a system analysis to system design transition phase because it makes the transition from the business concerns of system owners and users to the technology concerns of system designers and builders. And of course, systems analysts are the common thread that ensures continuity as we make this transition. Let's examine the transition.

Decision Analysis Given business requirements and the logical system models, there are usually numerous alternative ways to design a new information system to fulfill those requirements. Some of the pertinent questions include the following:

- How much of the system should be automated with information technology?
- Should we purchase software or build it ourselves (called the *make-versus-buy decision*)?
- Should we design the system for an internal network, or should we design a Web-based solution?
- What information technologies (possibly emerging) might be useful for this application?

These questions are answered in the DECISION ANALYSIS phase of the methodology. The purpose of this phase is to (1) identify candidate technical solutions, (2) analyze those candidate solutions for feasibility, and (3) recommend a candidate system as the target solution to be designed.

In Figure 3-6, we see that the decision analysis phase is positioned halfway through the development process. Half the building blocks are positioned higher, and half are positioned lower. This is consistent with the decision analysis phase's role as a transition from analysis to design—and from business concerns of SYSTEM USERS to those of SYSTEM DESIGNERS (and, ultimately, system builders). Designers (the technical experts in specific technologies) begin to play a role here along with system users and SYSTEM ANALYSTS. Analysts help to define and analyze the alternatives. Decisions are made regarding the technologies to be used as part of the application's architecture. Ultimately, SYSTEM OWNERS will have to approve or disapprove the approved decisions since they are paying for the project.

Figure 3-5 shows that a decision analysis is triggered by validated business requirements plus any logical system models and specifications that expand on those requirements. The project team solicits ideas and opinions for technical design and implementation from a diverse audience, possibly including IT software vendors. Candidate solutions are identified and characterized according to various criteria. It should be noted that many modern organizations have information technology and architecture standards that constrain the number of candidate solutions that might be considered and analyzed. (The existence of such standards is illustrated at the bottom of your information system building blocks model in Figure 3-6.) After the candidate solutions have been identified, each one is evaluated by the following criteria:

- *Technical feasibility*—Is the solution technically practical? Does our staff have the technical expertise to design and build this solution?
- *Operational feasibility*—Will the solution fulfill the user's requirements? To what degree? How will the solution change the user's work environment? How do users feel about such a solution?
- *Economic feasibility*—Is the solution cost-effective (as defined earlier in the chapter)?
- *Schedule feasibility*—Can the solution be designed and implemented within an acceptable time period?
- *Risk feasibility*—What's the probability of a successful implementation using the technology and approach?

The project team is usually looking for the *most* feasible solution—the solution that offers the best combination of technical, operational, economic, schedule, and risk feasibility. Different candidate solutions may be most feasible on a single criterion; however, one solution will usually prove *most* feasible based on all of the criteria.

The key deliverable of the decision analysis phase is a SYSTEM PROPOSAL. This proposal may be written and/or presented verbally. Several outcomes are possible. The creeping commitment feasibility checkpoint (again, the red diamond) may result in any one of the following options:

- Approve and fund the system proposal for design and construction (possibly including an increased budget and timetable if scope has significantly expanded).
- Approve or fund one of the alternative candidate solutions.
- Reject all the candidate solutions and either cancel the project or send it back for new recommendations.
- Approve a reduced-scope version of the proposed solution.

Optionally, the decision analysis phase may also produce an APPLICATION ARCHITECTURE for the approved solution. Such a model serves as a high-level blueprint (like a simple house floor plan) for the recommended or approved proposal.

Before we move on, you may have noticed in Figure 3-6 a variation on the SYSTEM PROPOSAL deliverable called a REQUEST FOR SYSTEM PROPOSALS (or RFP). This variation is for a recommendation to purchase the hardware and/or software solution as opposed to building it in-house. We'll defer any further discussion of this option until later in the chapter when we discuss the commercial package integration variation of our basic process.

Physical Design and Integration Given approval of the SYSTEM PROPOSAL from the decision analysis phase, you can finally design the new system. The purpose of the PHYSICAL DESIGN AND INTEGRATION phase is to transform the business requirements (represented in part by the LOGICAL SYSTEM MODELS) into PHYSICAL DESIGN SPECIFICATIONS that will guide system construction. In other words, physical design addresses greater detail about *how* technology will be used in the new system. The design will be constrained by the approved ARCHITECTURAL MODEL from the previous phase. Also, design requires adherence to any internal technical design standards that ensure completeness, usability, reliability, performance, and quality.

physical design the translation of business user requirements into a system model that depicts a technical implementation of the users' business requirements. Common synonyms include *technical design* or, in describing the output, *implementation model*. The antonym of physical design is *logical design* (defined earlier in this chapter).

Physical design is the opposite of logical design. Whereas logical design dealt exclusively with business requirements independent of any technical solution, physical design represents a specific technical solution. Figure 3-6 demonstrates the physical design phase from the perspective of your building blocks. Notice that the design phase is concerned with technology-based views of the system: (1) PHYSICAL DATABASE DESIGN SPECIFICATIONS, (2) PHYSICAL BUSINESS PROCESS and SOFTWARE DESIGN SPECIFICATIONS, and (3) PHYSICAL USER AND SYSTEM INTERFACE SPECIFICATIONS. The SYSTEM DESIGNER and SYSTEM ANALYST (possibly overlapping roles for some of the same individuals) are the key participants; however, certain aspects of the design usually have to be shared with the SYSTEM USERS (e.g., screen designs and work flow). You may have already had some exposure to physical design specifications in either programming or database courses.

There are two extreme philosophies of physical design.

- *Design by specification*—Physical system models and detailed specifications are produced as a series of written (or computer-generated) blueprints for construction.
- *Design by prototyping*—Incomplete but functioning applications or subsystems (called *prototypes*) are constructed and refined based on feedback from users and other designers.

In practice, some combination of these extremes is usually performed.

No new information system exists in isolation from other existing information systems in an organization. Consequently, a design must also reflect system integration concerns. The new system must be integrated both with other information systems

and with the business's processes themselves. Integration is usually reflected in physical system models and design specifications.

In summary, Figure 3-5 shows that the deliverables of the physical design and integration phase include some combination of PHYSICAL DESIGN MODELS AND SPECIFICATIONS, DESIGN PROTOTYPES, and REDESIGNED BUSINESS PROCESSES. Notice that we have included one final go or no-go feasibility checkpoint for the project (the red diamond). A project is rarely canceled after the design phase unless it is hopelessly over budget or behind schedule. On the other hand, scope could be decreased to produce a minimum acceptable product in a specified time frame. Or the schedule could be extended to build a more complete solution in multiple versions. The project plan (schedule and budget) would need to be adjusted to reflect these decisions.

It should be noted that in modern methodologies, there is a trend toward merging the design phase with our next phase, construction. In other words, the design and construction phases usually overlap.

Construction and Testing Given some level of PHYSICAL DESIGN MODELS AND SPECIFICATIONS (and/or DESIGN PROTOTYPES), we can begin to construct and test system components for that design. Figure 3-5 shows that the primary deliverable of the CONSTRUCTION AND TESTING phase is a FUNCTIONAL SYSTEM that is ready for implementation. The purpose of the construction and testing phase is twofold: (1) to build and test a system that fulfills business requirements and physical design specifications, and (2) to implement the interfaces between the new system and existing systems. Additionally, FINAL DOCUMENTATION (e.g., help systems, training manuals, help desk support, production control instructions) will be developed in preparation for training and system operation. The construction phase may also involve installation of purchased software.

Your information system framework (Figure 3-6) identifies the relevant building blocks and activities for the construction phase. The focus is on the last row of building blocks. The project team must construct or install:

- DATABASES—Databases may include *online transaction processing (OLTP)* databases to support day-to-day business transactions, *operational data stores (ODS)* to support day-to-day reporting and queries, and *data warehouses* to support data analysis and decision support needs.
- COMMERCIAL SOFTWARE PACKAGES and/or CUSTOM-BUILT SOFTWARE—Packages are installed and customized as necessary. Application programs are constructed according to the physical design and/or prototypes from the previous phase. Both packages and custom software must be thoroughly tested.
- USER AND SYSTEM INTERFACES—User interfaces (e.g., Windows and Web interfaces) must be constructed and tested for usability and stability. System-to-system interfaces must be either constructed or implemented using application integration technologies. Notice that MIDDLEWARE (a type of system software) is often used to integrate disparate database, software, and interface technologies. We'll talk more about middleware in the design unit of this book.

Figure 3-6 also identifies the participants in this phase as SYSTEM BUILDERS, SYSTEM ANALYSTS, SYSTEM USERS, and PROJECT MANAGERS. SYSTEM DESIGNERS may also be involved to clarify design specifications.

You probably already have some experience with part of this activity—application programming. Programs can be written in many different languages, but the current trend is toward the use of visual and object-oriented programming languages such as *Java*, *C++*, or *Visual Basic*. As components are constructed, they are typically demonstrated to users in order to solicit feedback.

One of the most important aspects of construction is conducting tests of both individual system components and the overall system. Once tested, a system (or version of a system) is ready for INSTALLATION AND DELIVERY.

Installation and Delivery What's left to do? New systems usually represent a departure from the way business is currently done; therefore, the analyst must provide

for a smooth transition from the old system to the new system and help users cope with normal start-up problems. Thus, the INSTALLATION AND DELIVERY phase serves to deliver the system into operation (sometimes called *production*).

In Figure 3-5, the FUNCTIONAL SYSTEM from the construction and testing phase is the key input to the INSTALLATION AND DELIVERY phase. The deliverable is an OPERATIONAL SYSTEM. SYSTEM BUILDERS install the system from its development environment into the production environment. SYSTEM ANALYSTS must train SYSTEM USERS, write various user and production control manuals, convert existing files and databases to the new databases, and perform final system testing. Any problems may initiate rework in previous phases thought to be complete. System users provide continuous feedback as new problems and issues arise. Essentially, the installation and delivery phase considers the same building blocks as the construction phase.

To provide a smooth transition to the new system, a conversion plan should be prepared. This plan may call for an abrupt cutover, where the old system is terminated and replaced by the new system on a specific date. Alternatively, the plan may run the old and new systems in parallel until the new system has been deemed acceptable to replace the old system.

The installation and delivery phase also involves training individuals who will use the final system and developing documentation to aid the system users. The implementation phase usually includes some form of POST-AUDIT REVIEW to gauge the success of the completed systems project. This activity promotes continuous improvement of the process and future project management.

System Operation and Maintenance Once the system is placed into operation, it will require ongoing **system support** for the remainder of its useful, productive lifetime. System support consists of the following ongoing activities:

system support the ongoing technical support for users of a system, as well as the maintenance required to deal with any errors, omissions, or new requirements that may arise

- *Assisting users*—Regardless of how well the users have been trained and how thorough and clear the end-user documentation is, users will eventually require additional assistance as unanticipated problems arise, new users are added, and so forth.
- *Fixing software defects (bugs)*—Software defects are errors that slipped through the testing of software. These are inevitable, but they can usually be resolved, in most cases, by knowledgeable support.
- *Recovering the system*—From time to time, a system failure may result in a program "crash" and/or loss of data. Human error or a hardware or software failure may cause this. The systems analyst or technical support specialists may then be called on to recover the system—that is, to restore a system's files and databases and to restart the system.
- *Adapting the system to new requirements*—New requirements may include new business problems, new business requirements, new technical problems, or new technology requirements.

Eventually, we expect that the user feedback and problems, or changing business needs, will indicate that it is time to start over and reinvent the system. In other words, the system has reached entropy, and a new project to create an entirely new system development process should be initiated.

cross life-cycle activity any activity that overlaps multiple phases of the system development process. Examples include *fact-finding, documentation, presentation, estimation, feasibility analysis, project and process management, change management,* and *quality management.*

> Cross Life-Cycle Activities

System development also involves a number of **cross life-cycle activities.** These activities, listed in the margin definition, are not explicitly depicted in Figure 3-5, but they are vital to the success of any project. Let's briefly examine each of these activities.

fact-finding the formal process of using research, interviews, meetings, questionnaires, sampling, and other techniques to collect information about system problems, requirements, and preferences. It is also called *information gathering* or *data collection.*

Fact-Finding There are many occasions for **fact-finding** during a project. Fact-finding is most crucial to the early phases of a project. It is during these phases that

the project team learns about a business's vocabulary, problems, opportunities, constraints, requirements, and priorities. But fact-finding is also used during the decision analysis, physical design, construction and testing, and installation and delivery phases—only to a lesser extent. It is during these latter phases that the project team researches technical alternatives and solicits feedback on technical designs, standards, and working components.

documentation the ongoing activity of recording facts and specifications for a system for current and future reference.

Documentation and Presentation Communication skills are essential to the successful completion of any project. In fact, poor communication is frequently cited as the cause of project delays and rework. Two forms of communication that are common to systems development projects are **documentation** and **presentation.**

Clearly, documentation and presentation opportunities span all the phases. In Figure 3-7, the black arrows represent various instances of documentation of a phase. The red arrows represent instances where presentations are frequently required. Finally, the green arrows represent the storage of documentation and other artifacts of systems development in a **repository.** A repository saves documentation for reuse and rework as necessary.

presentation the ongoing activity of communicating findings, recommendations, and documentation for review by interested users and managers. Presentations may be either written or verbal.

repository a database and/or file directory where system developers store all documentation, knowledge, and artifacts for one or more information systems or projects. A repository is usually automated for easy information storage, retrieval, and sharing.

Feasibility Analysis Consistent with our creeping commitment approach to systems development, **feasibility analysis** is a cross life-cycle activity. Different measures of **feasibility** are applicable in different phases of the methodology. These measures include technical, operational, economic, schedule, and risk feasibility, as described when we introduced the decision analysis phase. Feasibility analysis requires good **estimation** techniques.

feasibility analysis the activity by which feasibility is measured and assessed.

feasibility a measure of how beneficial the development of an information system would be to an organization.

Process and Project Management Recall that the CMM considers systems development to be a *process* that must be managed on a project-by-project basis. For this reason and others, process management and project management are ongoing, cross life-cycle activities. Both types of management were introduced earlier, but their definitions are repeated in the margin on page ••• for your convenience. **Process management** defines the methodology to be used on every project—think of it as the *recipe* for building a system. **Project management** is concerned with administering a single instance of the process as applied to a single project.

Failures and limited successes of systems development projects often outnumber successful projects. Why is that? One reason is that many systems analysts are unfamiliar with, or undisciplined in how to properly apply, tools and techniques of systems development. But most failures are attributed to poor leadership and management. This mismanagement results in unfulfilled or unidentified requirements, cost overruns, and late delivery.

estimation the calculated prediction of the costs and effort required for system development. A somewhat facetious synonym is *guesstimation*, usually meaning that the estimation is based on experience or empirical evicence but is lacking in rigor—in other words, a *guess*.

> Sequential versus Iterative Development

The above discussion of phases might lead you to assume that systems development is a naturally sequential process, moving in a one-way direction from phase to phase. Such sequential development is, in fact, one alternative. This approach is depicted in part (a) of Figure 3-8. In the figure we have used the four classic phases rather than the eight *FAST* phases in the interest of simplicity. This strategy requires that each phase be "completed" one after the other until the information system is finished. In reality, the phases may somewhat overlap one another in time. For example, some system design can be started prior to the completion of system analysis. Given its waterfall-like visual appearance, this approach is often called the **waterfall development approach.**

The waterfall approach has lost favor with most modern system developers. A more popular strategy, shown in part (b) of Figure 3-8, is commonly referred to as the **iterative development approach,** or incremental development process. This

process management an ongoing activity that documents, teaches, oversees the use of, and improves an organization's chosen methodology (the "process") for systems development. Process management is concerned with phases, activities, deliverables, and quality standards that should be consistently applied to all projects.

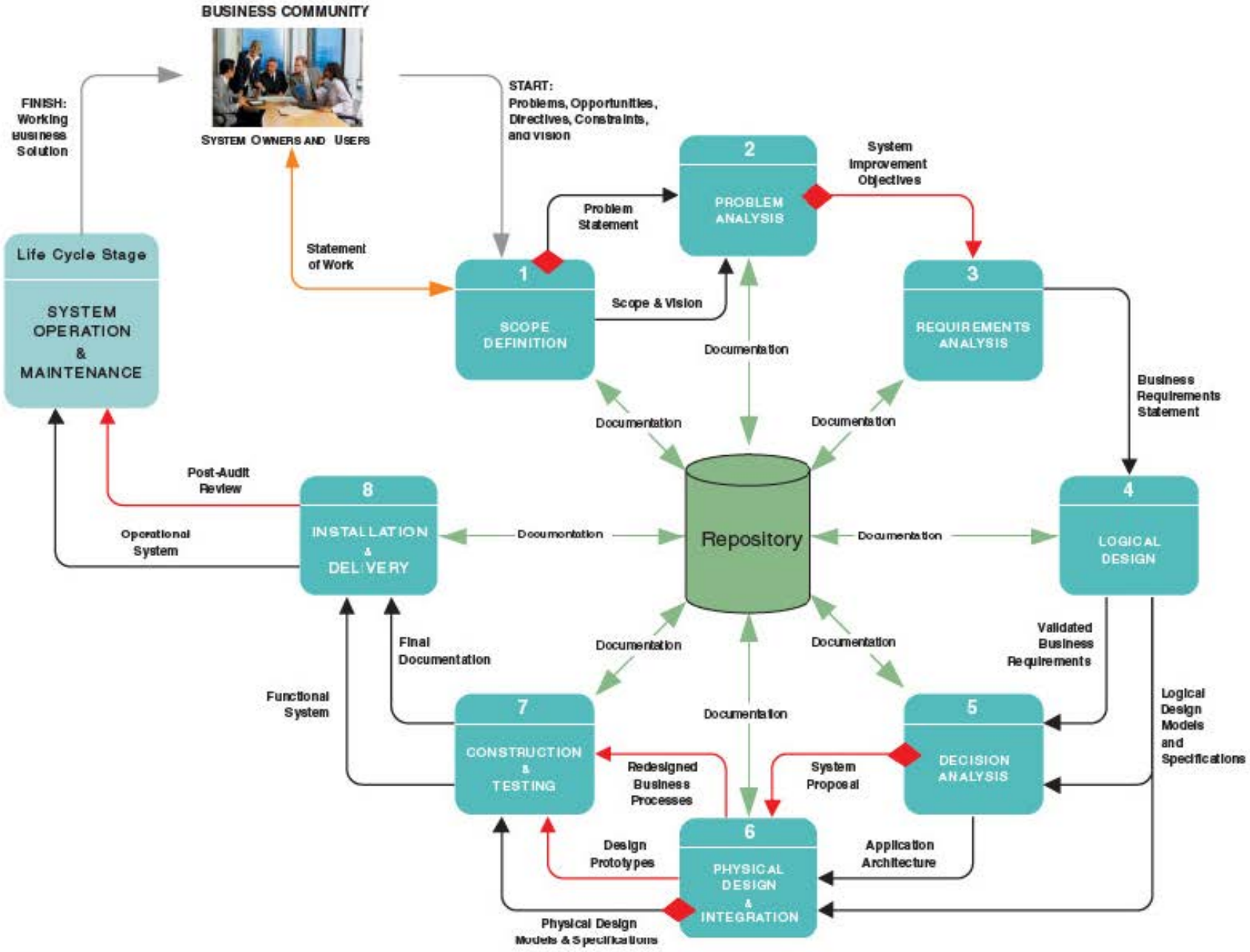


FIGURE 3-7 System Development Documentation, Repository, and Presentation

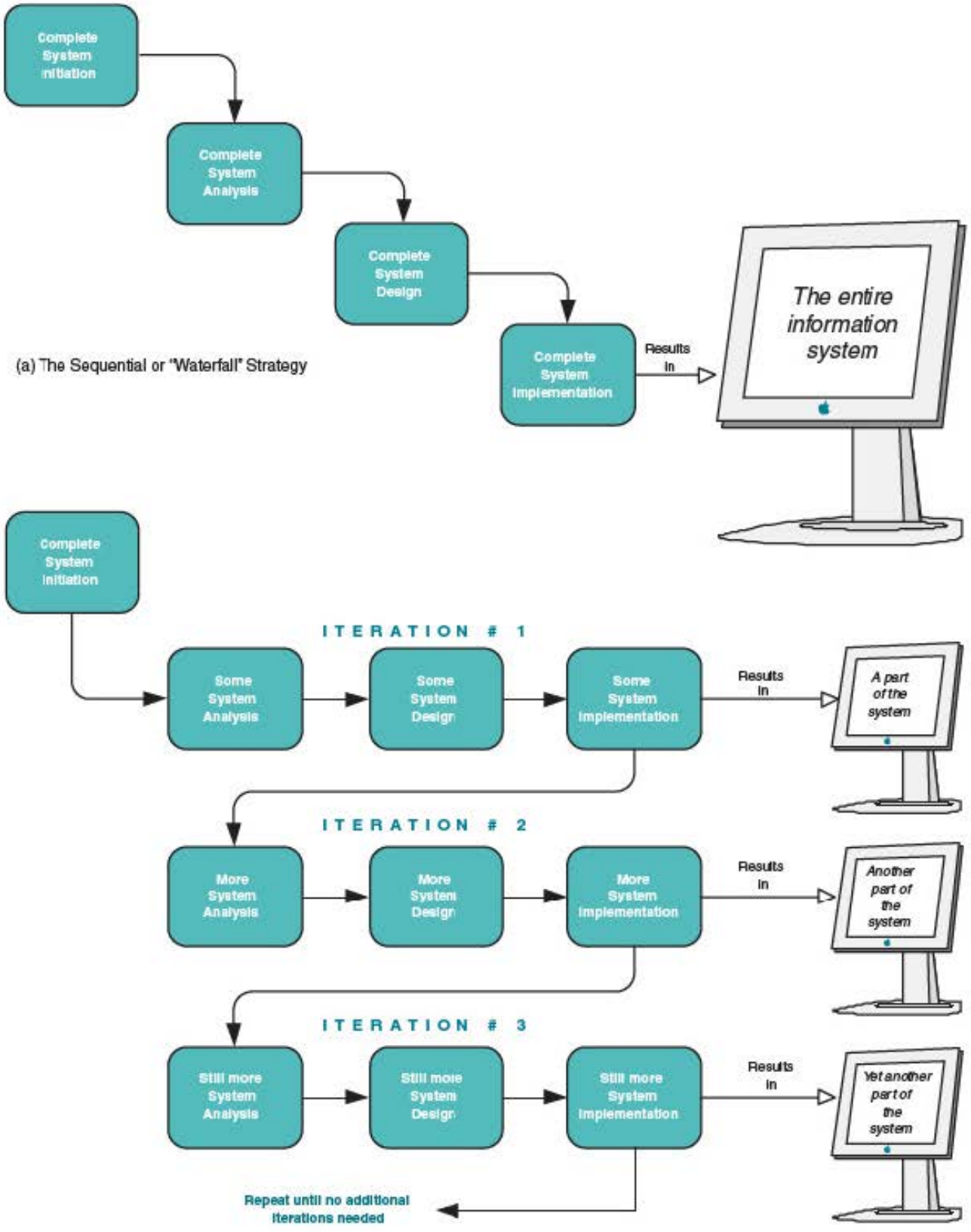


FIGURE 3-8 Sequential versus Iterative Systems Development Approach

project management the process of scoping, planning, staffing, organizing, directing, and controlling a project to develop an information system at minimum cost, within a specified time frame, and with acceptable quality.

approach requires completing enough analysis, design, and implementation to be able to fully develop a *part* of the new system and place it into operation as quickly as possible. Once that version of the system is implemented, the strategy is to then perform some additional analysis, design, and implementation to release the next version of the system. These iterations continue until all parts of the entire information system have been implemented. The popularity of this iterative and incremental process can be explained simply: System owners and users have long complained about the excessive time required to develop and implement information systems using the waterfall approach. The iterative approach allows versions of useable information to be delivered in regular and shorter time frames. This results in improved customer (system owner and user) satisfaction.

Alternative Routes and Strategies

waterfall development approach an approach to systems analysis and design that completes each phase one after another and only once.

iterative development approach an approach to systems analysis and design that completes that entire information system in successive iterations. Each iteration does some analysis, some design, and some construction. Synonyms include incremental and spiral.

Given any destination, there are many routes to that destination and many modes of transport. You could take the superhighway, highways, or back roads, or you could fly. Deciding which route is best depends on your goals and priorities. Do you want to get there fast, or do you want to see the sights? How much are you willing to spend? Are you comfortable with the mode of travel? Just as you would pick your route and means to a travel destination, you can and should pick a route and means for a systems development destination.

So far, we've described a basic set of phases that comprise our *FAST* methodology. At one time, a "one size fits all" methodology was common for most projects; however, today a variety of types of projects, technologies, and development strategies exist—one size no longer fits all projects! Like many contemporary methodologies, *FAST* provides alternative routes and strategies to accommodate different types of projects, technology goals, developer skills, and development paradigms.

In this section, we will describe several *FAST* routes and strategies. Before we do so, examine Figure 3-9 on the following page. The figure illustrates a taxonomy or classification scheme for methodological strategies. Notice the following:

- Methodologies and routes can support the option of either *building software solutions* in-house or *buying a commercial software solution* from a software vendor. Generally, many of the same methods and techniques are applicable to both options.
- Methodologies may be either very *prescriptive* ("Touch all the bases; follow all the rules") or relatively *adaptive* ("Change as needed within certain guidelines").
- Methodologies can also be characterized as *model-driven* ("Draw pictures of the system") or *product-driven* ("Build the product and see how the users react").
- Model-driven methodologies are rapidly moving to a focus on the *object-oriented* technologies being used to construct most of today's systems (more about this later). Earlier model-driven approaches emphasized either process modeling *or* data modeling.
- Finally, product-driven approaches tend to emphasize either rapid *prototyping* or writing program *code* as soon as possible (perhaps you've heard the term *extreme programming*).

So many strategies! Which should you choose? A movement is forming known as *agile methods*. In a nutshell, advocates of agile methods suggest that system analysts and programmers should have a tool box of methods that include tools and techniques from all of the above methodologies. They should choose their tools and techniques based on the problem and situation. *FAST* is an agile methodology. It advocates the integrated use of tools and techniques from many methodologies, applied in the context of repeatable processes (as in CMM Level 3). That said, let's examine some of the route variations and strategies for the *FAST* process. As we

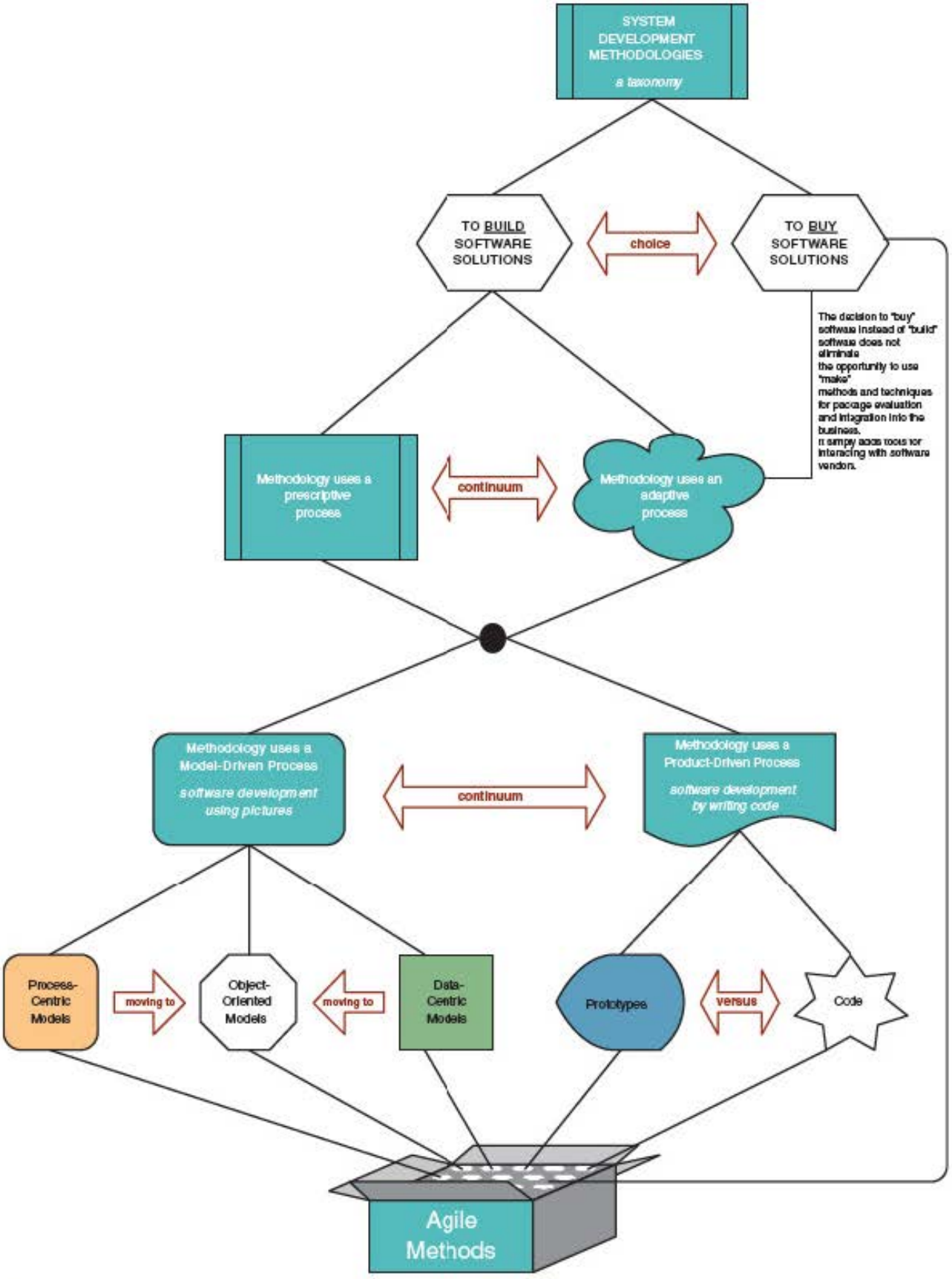


FIGURE 3-9 A Taxonomy for System Development Methodologies and Strategies

navigate through each route, we will use red typefaces and arrows to highlight those aspects of the route that differ from the basic route you've already learned.

> The Model-Driven Development Strategy

One of the oldest and most commonly used approaches to analyzing and designing information systems is based on system modeling. As a reminder, a system model is a picture of a system that represents reality or a desired reality. System models facilitate improved communication between system users, system analysts, system designers, and system builders. In the *FAST* methodology, system models are used to illustrate and communicate the KNOWLEDGE, PROCESS, OR INTERFACE building blocks of information systems. This approach is called **model-driven development.**

model-driven development a system development strategy that emphasizes the drawing of system models to help visualize and analyze problems, define business requirements, and design information systems.

The model-driven development route for *FAST* is illustrated in Figure 3-10. The model-driven approach does not vary much from the basic phases we described earlier. We call your attention to the following notes that correspond to the numbered bullets:

1. System models may exist from the project that created the current system. Be careful! These models are notorious for being out of date. But they can still be useful as a point of departure.
2. Earlier you learned that it is important to define scope for a project. One of the simplest ways to communicate scope is by drawing MODELS THAT SHOW SCOPE DEFINITION. Scope models show which aspects of a problem are within scope and which aspects are outside scope. This is sometimes called a *context diagram* or context model.
3. Some system modeling techniques call for extensive MODELS OF THE EXISTING SYSTEM to identify problems and opportunities for system improvement. This is sometimes called the *as-is system model.* Modeling of the current system has waned in popularity today. Many project managers and analysts view it as counterproductive or of little value added. The exception is modeling of as-is business processes for the purpose of business process redesign.
4. The requirements statement is one of the most important deliverables of system development. It sometimes includes MODELS THAT DEPICT HIGH-LEVEL BUSINESS REQUIREMENTS. One of the most popular modeling techniques today is called *use case* (introduced in Chapter 7). Use cases identify requirements and track their fulfillment through the life cycle.
5. Most model-driven techniques require that analysts document business requirements with **logical models** (defined earlier). Business requirements are frequently expressed in LOGICAL MODELS THAT DEPICT MORE DETAILED USER REQUIREMENTS. They show only *what* a system must be or must do. They are implementation *in*dependent; that is, they depict the system independent of any possible technical implementation. Hence, they are useful for depicting and validating business requirements.
6. As a result of the decision analysis phase, the analyst may produce system MODELS THAT DEPICT APPLICATION ARCHITECTURE. Such models illustrate the planned technical implementation of a system.
7. Many model-driven techniques require that analysts develop MODELS THAT DEPICT PHYSICAL DESIGN SPECIFICATIONS (defined earlier in this chapter). Recall that **physical models** show not only what a system is or does but also *how* the system is implemented with technology. They are implementation *de*pendent because they reflect technology choices and the limitations of those technology choices. Examples include database schemas, structure charts, and flowcharts. They serve as a blueprint for construction of the new system.
8. New information systems must be interwoven into the fabric of an organization's business processes. Accordingly, the analyst and users may develop MODELS OF REDESIGNED BUSINESS PROCESSES.

logical model a pictorial representation that depicts *what* a system is or does. Synonyms include essential model, conceptual model, and business model.

physical model a technical pictorial representation that depicts what a system is or does and *how* the system is implemented. Synonyms include implementation model and technical model.

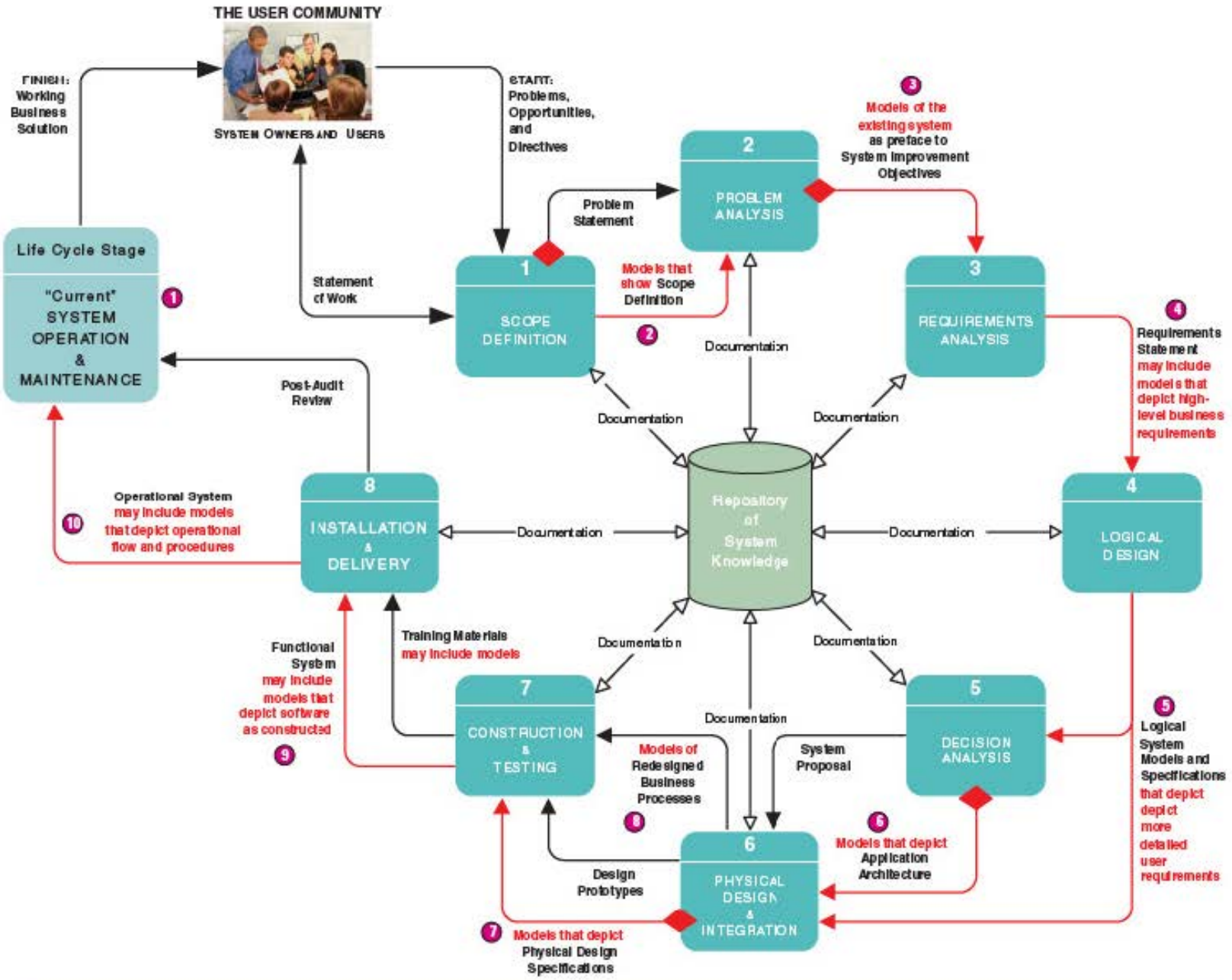


FIGURE 3-10 The Model-Driven System Development Strategy

9. Construction translates the physical system models into software. In some cases, automated tools exist to automatically translate software into PHYSICAL MODELS THAT DEPICT SOFTWARE CONSTRUCTED. This is called *reverse engineering*.
10. Finally, the operational system may include MODELS THAT DEPICT FLOW AND PROCEDURE. For example, system models may document backup and recovery procedures.

In summary, system models can be produced as a portion of the deliverables for most phases. Model-driven approaches emphasize system modeling. Once implemented, the system models serve as documentation for any changes that might be needed during the operation and support stage of the life cycle.

The model-driven approach is believed to offer several advantages and disadvantages, as listed below:

Advantages

- Requirements specification tends to be more thorough and better documented.
- Business requirements and system designs are easier to validate with pictures than words.
- It is easier to identify, conceptualize, and analyze alternative technical solutions.
- Design specifications tend to be more sound, stable, adaptable, and flexible because they are model based and more thoroughly analyzed *before* they are built.
- Systems can be constructed more correctly the first time when built from thorough and clear model based specifications. Some argue that code-generating software can automatically generate skeleton or near-complete code from good system models.

Disadvantages

- It is time-consuming. It takes time to collect the facts, draw the models, and validate those models. This is especially true if users are uncertain or imprecise about their system requirements.
- The models can only be as good as the users' understanding of those requirements.
- Pictures are not software—some argue that this reduces the users' role in a project to passive participation. Most users don't get excited about pictures. Instead, they want to see working software, and they gauge project progress by the existence of software (or its absence).
- The model-driven approach is considered by some to be inflexible—users must fully specify requirements before design, design must fully document technical specifications before construction, and so forth. Some view such rigidity as impractical.

process modeling a process-centered technique popularized by the *structured analysis and design* methodology that used models of business process requirements to derive effective software designs for a system. Structured analysis introduced a modeling tool called the *data flow diagram* to illustrate the flow of data through a series of business processes. Structured design converted data flow diagrams into a process model called *structure charts* to illustrate a top-down software structure that fulfills the business requirements.

Model-driven development is most effective for systems for which requirements are well understood and which are so complex that they require large project teams to complete. The approach also works well when fulfillment of user expectations and quality is more important than cost and schedule.

There are several different model-driven techniques. They differ primarily in terms of the types of models that they require the systems analyst to draw and validate. Let's briefly examine three of the most popular model-driven development techniques that will be taught in this book. Please note that we are introducing only the techniques here, not the models. We'll teach the models themselves later, in the "how to" chapters.

Process Modeling **Process modeling** was founded in the structured analysis and design methodologies in 1978. While structured analysis and design has lost favor as a

methodology, process modeling remains a viable and important technique. Recall that your information system building blocks include several possible focuses: KNOWLEDGE, PROCESSES, and INTERFACES. Process modeling focuses on the PROCESS column of building blocks. *Flowcharts* are one type of process model (used primarily by SYSTEM BUILDERS) that you may have encountered in a programming course. Process modeling has enjoyed something of a renaissance with the emergence of business process redesign (introduced in Chapter 1).

Data flow diagrams and structure charts have contributed significantly to reducing the communications gap that often exists between nontechnical system owners and users and technical system designers and builders. Process modeling is taught in this book.

Data Modeling Recall that KNOWLEDGE improvement is a fundamental goal and set of building blocks in your framework. Knowledge is the product of *information*, which in turn is the product of *data*. **Data modeling** methods emphasize the knowledge building blocks, especially data. In the data modeling approach, emphasis is placed on diagrams that capture business data requirements and translate them into database designs. Arguably, data modeling is the most widely practiced system modeling technique. Hence, it will be taught in this book.

data modeling a data-centered technique used to model business data requirements and design database systems that fulfill those requirements. The most frequently encountered data models are *entity relationship diagrams*.

Object Modeling Object modeling is the result of technical advancement. Today, most programming languages and methods are based on the emergence of object technology. While the concepts of object technology are covered extensively throughout this book, a brief but oversimplified introduction is appropriate here.

For the past 30 years, techniques like process and data modeling deliberately separated the concerns of PROCESSES from those of DATA. In other words, process and data models were separate and distinct. Because virtually all systems included processes *and* data, the techniques were frequently used in parallel and the models had to be carefully synchronized. Object techniques are an attempt to eliminate the separation of concerns, and hence the need for synchronization of data and process concerns. This has given rise to **object modeling** methods.

object modeling a technique that attempts to merge the data and process concerns into singular constructs called *objects*. Object models are diagrams that document a system in terms of its objects and their interactions. Object modeling is the basis for *object-oriented analysis and design* methodologies.

Business objects correspond to real things of importance in the business such as *customers* and the *orders* they place for *products*. Each object consists of *both* the data that describes the object and the processes that can create, read, update, and delete that object. With respect to your information system building blocks, object-oriented analysis and design (OOAD) significantly changes the paradigm. The DATA and PROCESS columns (and, arguably, the INTERFACE column as well) are essentially merged into a single OBJECT column. The models then focus on identifying objects, building objects, and assembling appropriate objects, as with *Legos*, into useful information systems.

The current popularity of object technology is driving the interest in object models and OOAD. For example, most of today's popular operating systems like Microsoft *Windows* and Apple *Mac/OS* have object-oriented user interfaces ("point and click," using objects such as windows, frames, drop-down menus, radio buttons, checkboxes, scroll bars, and the like). Web user interfaces like Microsoft *Internet Explorer* and Netscape *Navigator* are also based on object technology. Object programming languages such as *Java*, *C++*, *C#*, *Smalltalk*, and *Visual Basic .NET* are used to construct and assemble such object-oriented operating systems and applications. And those same languages have become the tools of choice for building next-generation information system applications. Not surprisingly, object modeling techniques have been created to express business and software requirements and designs in terms of objects. This edition of this book extensively integrates the most popular object modeling techniques to prepare you for systems analysis and design that ultimately produces today's object-based information systems and applications.

> The Rapid Application Development Strategy

rapid application development (RAD) a system development strategy that emphasizes speed of development through extensive user involvement in the rapid, iterative, and incremental construction of a series of functioning **prototypes** of a system that eventually evolves into the final system (or a version).

In response to the faster pace of the economy in general, **rapid application development (RAD)** has become a popular route for accelerating systems development. The basic ideas of RAD are:

- To more actively involve system users in the analysis, design, and construction activities.
- To organize systems development into a series of focused, intense workshops jointly involving SYSTEM OWNERS, USERS, ANALYSTS, DESIGNERS, and BUILDERS.
- To accelerate the requirements analysis and design phases through an iterative construction approach.
- To reduce the amount of time that passes before the users begin to see a working system.

prototype a small-scale, representative, or working model of the users' requirements or a proposed design for an information system. Any given prototype may omit certain functions or features until such time as the prototype has sufficiently evolved into an acceptable implementation of requirements.

The basic principle behind prototyping is that users know what they want when they see it working. In RAD, a **prototype** eventually evolves into the final information system. The RAD route for *FAST* is illustrated in Figure 3-11. Again, the red text and flows indicate the deviations from the basic *FAST* process. We call your attention to the following notes that correspond to the numbered bullets:

1. The emphasis is on reducing time in developing applications and systems; therefore, the initial problem analysis, requirements analysis, and decision analysis phases are consolidated and accelerated. The deliverables are typically abbreviated, again in the interest of time. The deliverables are said to be INITIAL, meaning "expected to change" as the project progresses.

   After the above initial analysis, the RAD uses an iterative approach, as discussed earlier in the chapter. Each iteration emphasizes only enough new functionality to be accomplished within a few weeks.
2. LOGICAL AND PHYSICAL DESIGN SPECIFICATIONS are usually significantly abbreviated and accelerated. In each iteration of the cycle, only some design specifications will be considered. While some system models may be drawn, they are selectively chosen and the emphasis continues to be on rapid development. The assumption is that errors can be caught and fixed in the next iteration.
3. In some, but rarely all, iterations, some business processes may need to be redesigned to reflect the likely integration of the evolving software application.
4. In each iteration of the cycle, SOME DESIGN PROTOTYPES or SOME PARTIAL FUNCTIONAL SYSTEM elements are constructed and tested. Eventually, the completed application will result from the final iteration through the cycle.
5. After each prototype or partial functional subsystem is constructed and tested, system users are given the opportunity to experience working with that prototype. The expectation is that users will clarify requirements, identify new requirements, and provide BUSINESS FEEDBACK on design (e.g., ease of learning, ease of use) for the next iteration through the RAD cycle.
6. After each prototype or functioning subsystem is constructed and tested, system analysts and designers will review the application architecture and design to provide TECHNICAL FEEDBACK and direction for the next iteration through the RAD cycle.
7. Based on the feedback, systems analysts will identify REFINED SYSTEM IMPROVEMENT OBJECTIVES and/or BUSINESS REQUIREMENTS. This analysis tends to focus on revising or expanding objectives and requirements and identifying user concerns with the design.
8. Based on the feedback, systems analysts and system designers will identify a REFINED APPLICATION ARCHITECTURE and/or DESIGN CHANGES.
9. Eventually, the system (or a version of the system) will be deemed worthy of implementation. This CANDIDATE RELEASE VERSION OF THE FUNCTIONAL SYSTEM is system tested and placed into operation. The next version of the system may continue iterating through the RAD cycle.

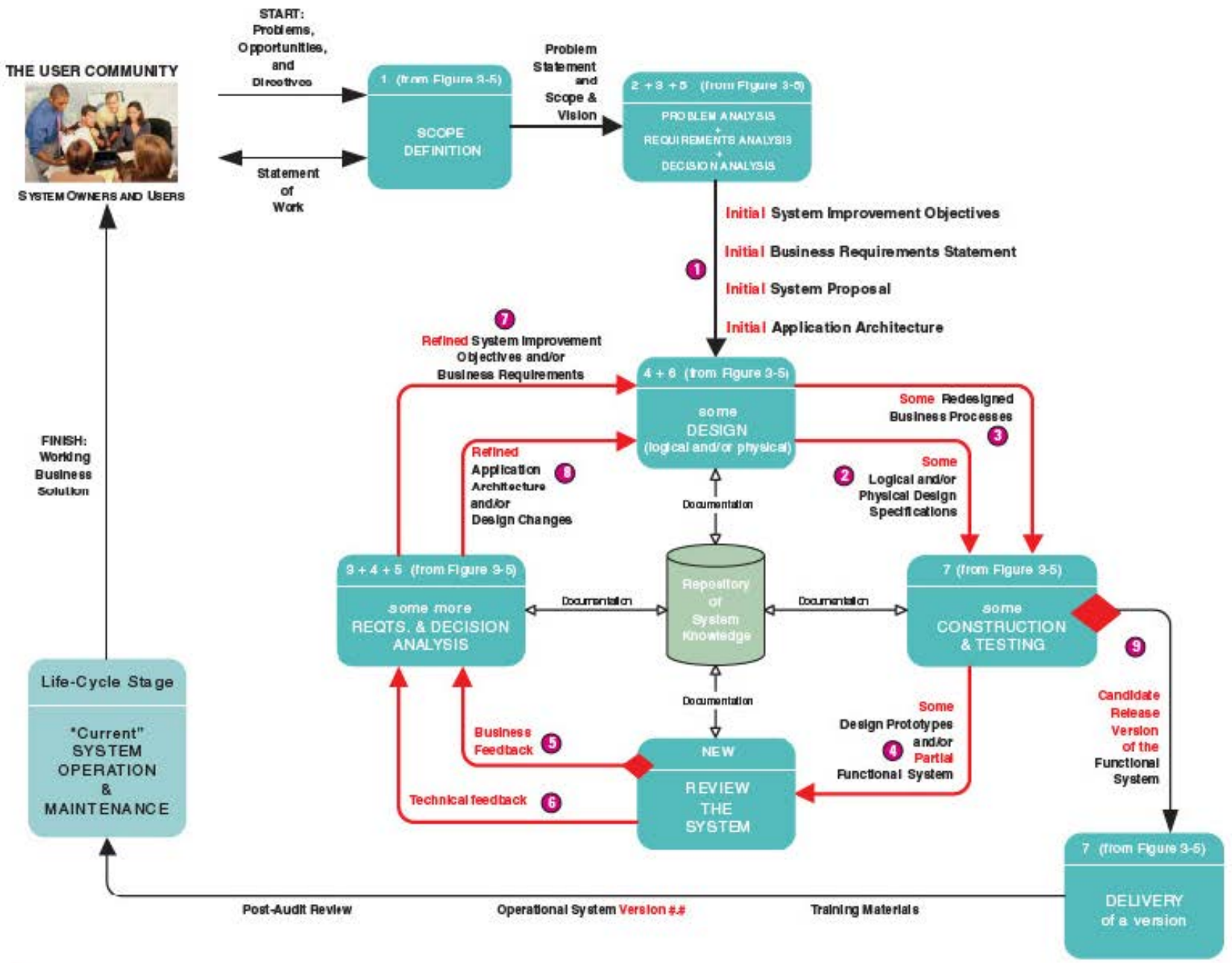


FIGURE 3-11 The Rapid Application Development (RAD) Strategy

timeboxing the imposition of a nonextendable period of time, usually 60 to 90 days, by which the first (or next) version of a system must be delivered into operation.

Although not a rigid requirement for RAD, the duration of the prototyping loop can be limited using a technique called **timeboxing.** Timeboxing seeks to deliver an operational system to users and management on a regular, recurring basis. Advocates of timeboxing argue that management and user enthusiasm for a project can be enhanced and sustained because a working version of the system is implemented on a regular basis.

The RAD approach offers several advantages and disadvantages:

Advantages

- It is useful for projects in which the user requirements are uncertain or imprecise.
- It encourages active user and management participation (as opposed to a passive reaction to nonworking system models). This increases end-user enthusiasm for the project.
- Projects have higher visibility and support because of the extensive user involvement throughout the process.
- Users and management see working, software-based solutions more rapidly than they do in model-driven development.
- Errors and omissions tend to be detected earlier in prototypes than in system models.
- Testing and training are natural by-products of the underlying prototyping approach.
- The iterative approach is a more natural process because change is an expected factor during development.

Disadvantages

- Some argue that RAD encourages a "code, implement, and repair" mentality that increases lifetime costs required to operate, support, and maintain the system.
- RAD prototypes can easily solve the wrong problems since problem analysis is abbreviated or ignored.
- A RAD-based prototype may discourage analysts from considering other, more worthy technical alternatives.
- Sometimes it is best to throw a prototype away, but stakeholders are often reluctant to do so because they see this as a loss of time and effort in the current product.
- The emphasis on speed can adversely impact quality because of ill-advised shortcuts through the methodology.

RAD is most popular for small- to medium-size projects. We will demonstrate prototyping and RAD techniques in appropriate chapters of this book.

> The Commercial Application Package Implementation Strategy

commercial application package a software application that can be purchased and customized (within limits) to meet the business requirements of a large number of organizations or a specific industry. A synonym is *commercial off-the-shelf (COTS) system.*

Sometimes it makes more sense to buy an information system solution than to build one in-house. In fact, many organizations increasingly expect to build software in-house only when there is a competitive advantage to be gained. And for many core applications such as human resources, financials, procurement, manufacturing, and distribution, there is little competitive value in building your own system—hence a **commercial application package** is purchased. Accordingly, our *FAST* methodology includes a commercial software package route.

The ultimate commercial solution is enterprise resource planning, or ERP (defined in Chapter 1). ERP solutions provide *all* of the core information system applications for an entire business. For most organizations, an ERP implementation represents the single largest information system project ever undertaken by the organization. It can cost tens or hundreds of millions of dollars and require a small army of managers, users, analysts, technical specialists, programmers, consultants, and contractors.

The *FAST* methodology's route for commercial application package integration is not really intended for ERP projects. Indeed, most ERP vendors provide their own implementation methodology (and consulting partners) to help their customers implement such a massive software solution. Instead, our *FAST* methodology provides a route for implementing all other types of information system solutions that may be purchased by a business. For example, an organization might purchase a commercial application package for a single business function such as accounting, human resources, or procurement. The package must be selected, installed, customized, and integrated into the business and its other existing information systems.

The basic ideas behind our commercial application package implementation route are:

- Packaged software solutions must be carefully selected to fulfill business needs—"You get what you ask and pay for."
- Packaged software solutions not only are costly to purchase but can be costly to implement. In fact, the package route can actually be more expensive to implement than an in-house development route.
- Software packages must usually be customized for and integrated into the business. Additionally, software packages usually require the redesign of existing business processes to adapt to the software.
- Software packages rarely fulfill all business requirements to the users' complete satisfaction. Thus, some level of in-house systems development is necessary in order to meet the unfulfilled requirements.

The commercial application package implementation route is illustrated in Figure 3-12. Once again, the red typeface and arrows indicate differences from the basic *FAST* process. We call your attention to the following notes that correspond to the numbered bullets:

1. It should be noted that the decision to purchase a package is determined in the problem analysis phase. The red diamond represents the "make versus buy" decision. The remainder of this discussion assumes that a decision to buy has been approved.
2. Problem analysis usually includes some initial TECHNOLOGY MARKET RESEARCH to identify what package solutions exist, what features are in the software, and what criteria should be used to evaluate such application packages. This research may involve software vendors, IT research services (such as the Gartner Group), or consultants.
3. After defining business requirements, the requirements must be communicated to the software vendors who offer viable application solutions. The business (and technical) requirements are formatted and communicated to candidate software vendors as either a **REQUEST FOR PROPOSAL (RFP)** or a **REQUEST FOR QUOTATION (RFQ).** The double-ended arrow implies that there may need to be some clarification of requirements and criteria.
4. Vendors submit PROPOSALS or QUOTATIONS for their application solutions. These proposals are evaluated against the business and technical requirements specified in the RFP. The double-ended arrow indicates that claimed features and capabilities must be validated and in some instances clarified. This occurs during the decision analysis phase.
5. A CONTRACT AND ORDER is negotiated with the winning vendor for the software and possibly for services necessary to install and maintain the software.
6. The vendor provides the BASELINE COMMERCIAL APPLICATION software and documentation. Services for installation and implementation of the software are frequently provided by the vendor or its service providers (certified consultants).
7. When an application package is purchased, the organization must nearly always change its business processes and practices to work efficiently with the package. The need for REDESIGNED BUSINESS PROCESSES is rarely greeted with enthusiasm,

request for proposal (RFP) a formal document that communicates business, technical, and support requirements for an application software package to vendors that may wish to compete for the sale of that application package and services.

request for quotation (RFQ) a formal document that communicates business, technical, and support requirements for an application software package to a single vendor that has been determined as being able to supply that application package and services.

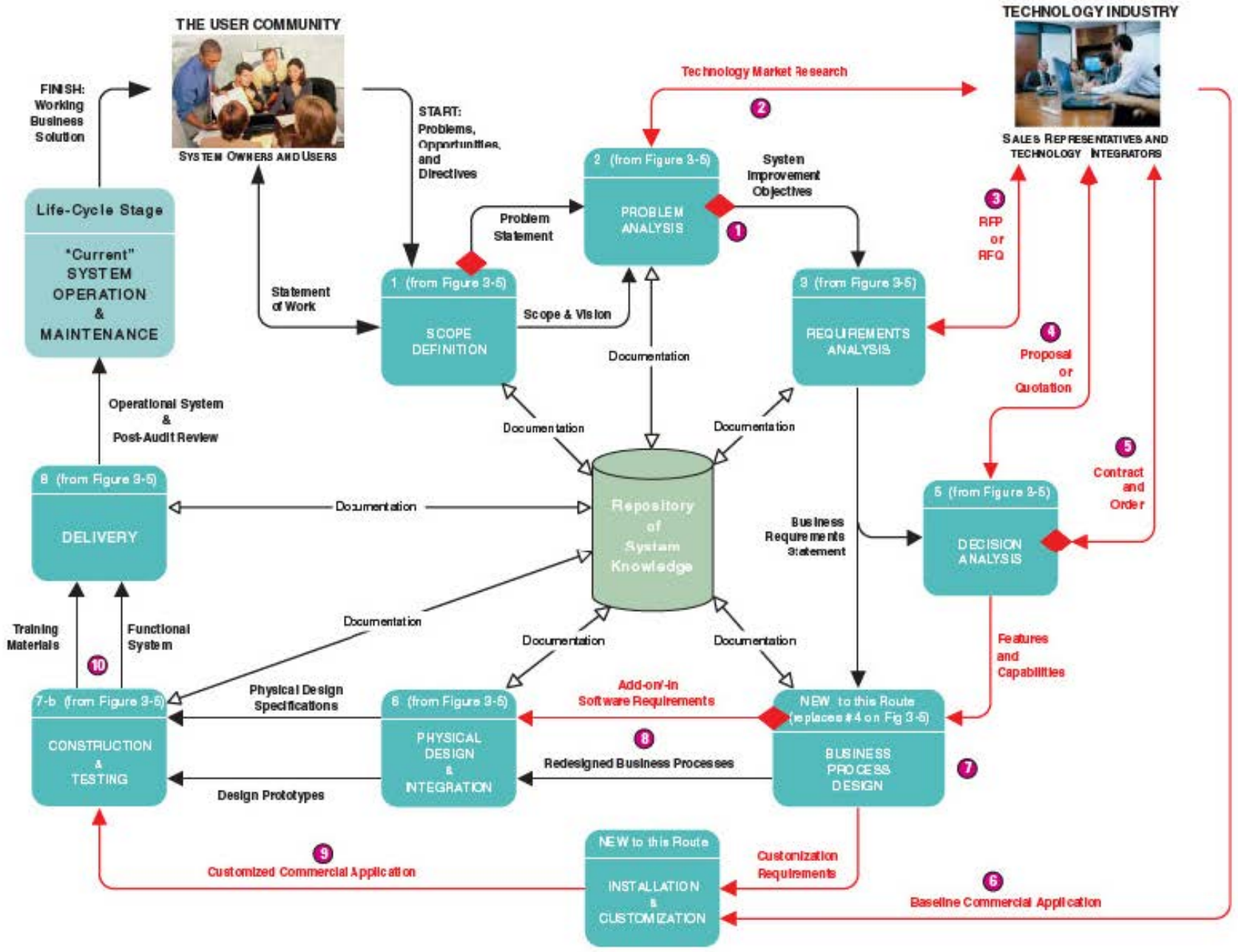


FIGURE 3-12 The Commercial Application Package Implementation Strategy

but they are usually necessary. In many cases, the necessary changes are not wrong—they are just different and unfamiliar. System users tend to be uncomfortable with changing the way they have always done something.

8. An application package rarely meets all business requirements upon installation. Typically, a **gap analysis** must be performed to determine which business requirements are not fulfilled by the package's capabilities and features. For requirements that will not be fulfilled, the following options exist:
   - Request customizations of the package within allowable limits as specified by the software vendor. Most commercial application packages allow the purchaser to set specific options, preferences, and defined values and ranges for certain parameters. Within limits, these customizations allow you to "personalize" the package to the business accounting and business practices. Such necessary CUSTOMIZATION REQUIREMENTS need to be specified.
   - Define ADD-ON SOFTWARE REQUIREMENTS. Add-on software requirements specify programs that must be designed and constructed to augment the commercial application package and deliver additional functionality. It should be noted that add-on programs carry some risk that they may have to be modified in the future when a new version of the software becomes available. But this risk is nominal, and most organizations take the risk in order to provide additional functionality.
   - Define ADD-IN SOFTWARE REQUIREMENTS. Add-in software requirements are very dangerous! They specify changes to the actual commercial application package to meet business requirements. In other words, users are requesting that changes be made to the purchased software, its database, or its user interfaces. At best, such changes can make version upgrades extremely difficult and prohibitively expensive. At worst, such changes can invalidate technical support from the vendor. (And most vendors encourage keeping versions relatively current by canceling technical support on older versions.) Changing program code and database structures should be discouraged. Insistence by users is often a symptom of unwillingness to adapt business processes to work with the application. Many organizations prohibit changes to application packages and force users to adapt in order to preserve their upgrade path.
9. The BASELINE COMMERCIAL APPLICATION is installed and tested. Allowable changes based on options, preferences, and parameters are completed and tested. Note: These customizations within the limits specified by the software vendor will typically carry forward to version upgrades. In most instances the vendor has provided for this level of CUSTOMIZED COMMERCIAL APPLICATION.
10. Any add-on (or add-in) software changes are designed and constructed to meet additional business requirements. The system is subsequently tested and placed into operation using the same activities described in the basic *FAST* process.

gap analysis a comparison of business and technical requirements for a commercial application package against the capabilities and features of a specific commercial application package for the purpose of defining the requirements that cannot be met.

The commercial application package strategy offers its own advantages and disadvantages:

Advantages

- New systems can usually be implemented more quickly because extensive programming is not required.
- Many businesses cannot afford the staffing and expertise required to develop in-house solutions.
- Application vendors spread their development costs across all customers that purchase their software. Thus, they can invest in continuous

Disadvantages

- A successful COTS implementation is dependent on the long-term success and viability of the application vendor—if the vendor goes out of business, you can lose your technical support and future improvements.
- A purchased system rarely reflects the ideal solution that the business might achieve with an in-house-developed system that could be

improvements in features, capabilities, and usability that individual businesses cannot always afford.
- The application vendor assumes responsibility for significant system improvements and error corrections.
- Many business functions are more similar than dissimilar for all businesses in a given industry. For example, business functions across organizations in the health care industry are more alike than different. It does not make good business sense for each organization to "reinvent the wheel."

customized to the precise expectations of management and the users.
- There is almost always at least some resistance to changing business processes to adapt to the software. Some users will have to give up or assume new responsibilities. And some people may resent changes they perceive to be technology-driven instead of business-driven.

Regardless, the trend toward purchased commercial application packages cannot be ignored. Today, many businesses require that a package alternative be considered prior to engaging in any type of in-house development project. Some experts estimate that by the year 2005 businesses will purchase 75 percent of their new information system applications. For this reason, we will teach systems analysis tools and techniques needed by system analysts to function in this environment.

> Hybrid Strategies

The *FAST* routes are not mutually exclusive. Any given project may elect to or be required to use a combination of, or variation of, more than one route. The route to be used is always selected during the *scope definition* phase and is negotiated as part of the *statement of work.* One strategy that is commonly applied to both model-driven and rapid application development routes is an incremental strategy. Figure 3-13 illustrates one possible implementation of an incremental strategy in combination with rapid application development. The project delivers the information system into operation in four stages. Each stage implements a version of the final system using a RAD route. Other variations on routes are possible.

> System Maintenance

All routes ultimately result in placing a new system into operation. System maintenance is intended to guide projects through the operation and support stage of their life cycle—which could last decades! Figure 3-14 places system maintenance into perspective. System maintenance in *FAST* is not really a unique route. As illustrated in the figure, it is merely a smaller-scale version of the same *FAST* process (or route) that was used to originally develop the system. The figure demonstrates that the starting point for system maintenance depends on the problem to be solved. We call your attention to the following numbered bullets in the figure:

1. Maintenance and reengineering projects are triggered by some combination of user and technical feedback. Such feedback may identify new problems, opportunities, or directives.
2. The maintenance project is initiated by a SYSTEM CHANGE REQUEST that indicates the problems, opportunities, or directives.
3. The simplest fixes are SOFTWARE BUGS (errors). Such a project typically jumps right into a RECONSTRUCTION AND RETESTING phase and is solved relatively quickly.
4. Sometimes a DESIGN FLAW in the system becomes apparent after implementation. For example, users may frequently make the same mistake due to a confusing screen design. For this type of maintenance project, the PHYSICAL DESIGN AND INTEGRATION phase would need to be revisited, followed of course by the construction and delivery phases.

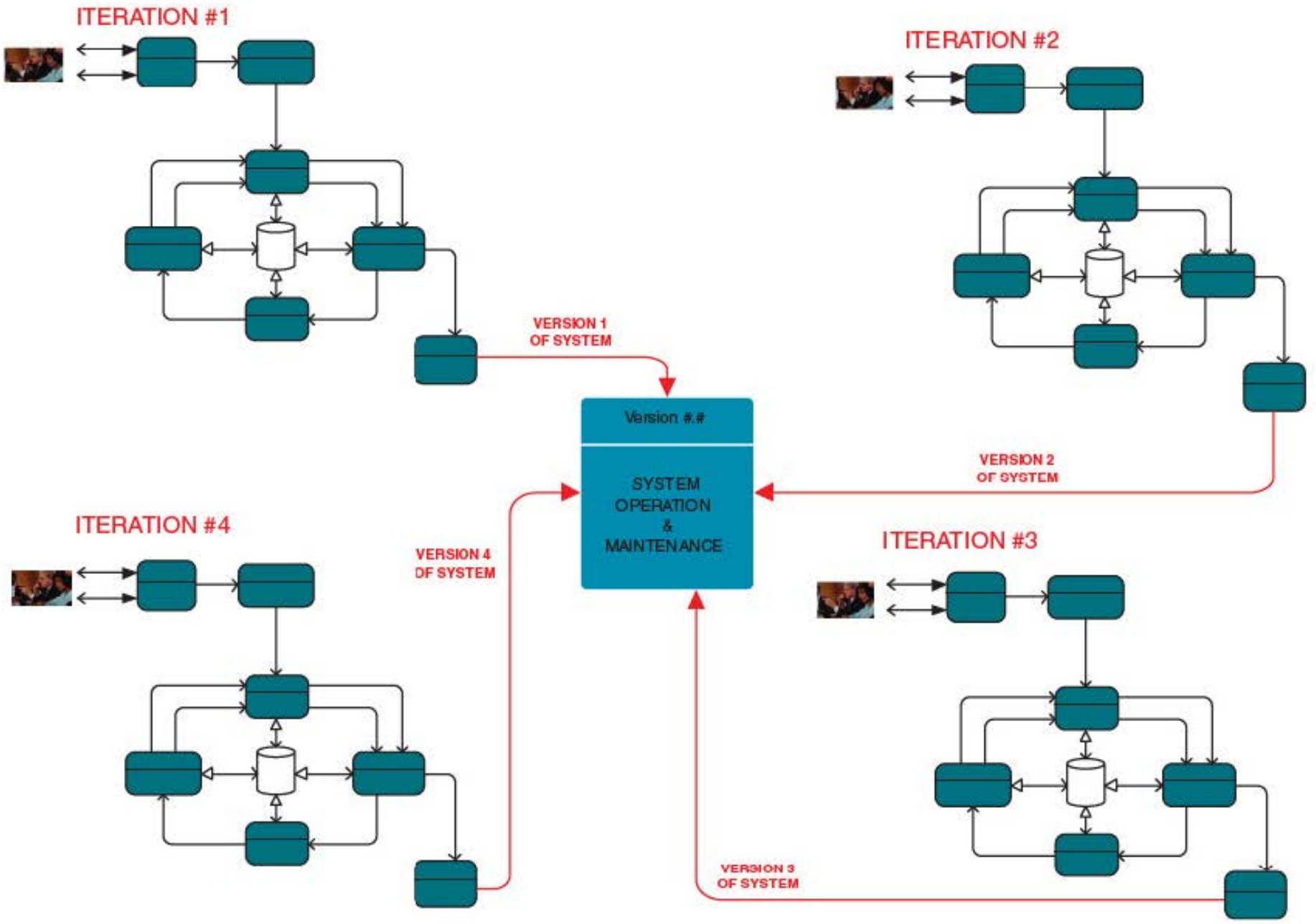


FIGURE 3-13 An Incremental Implementation Strategy

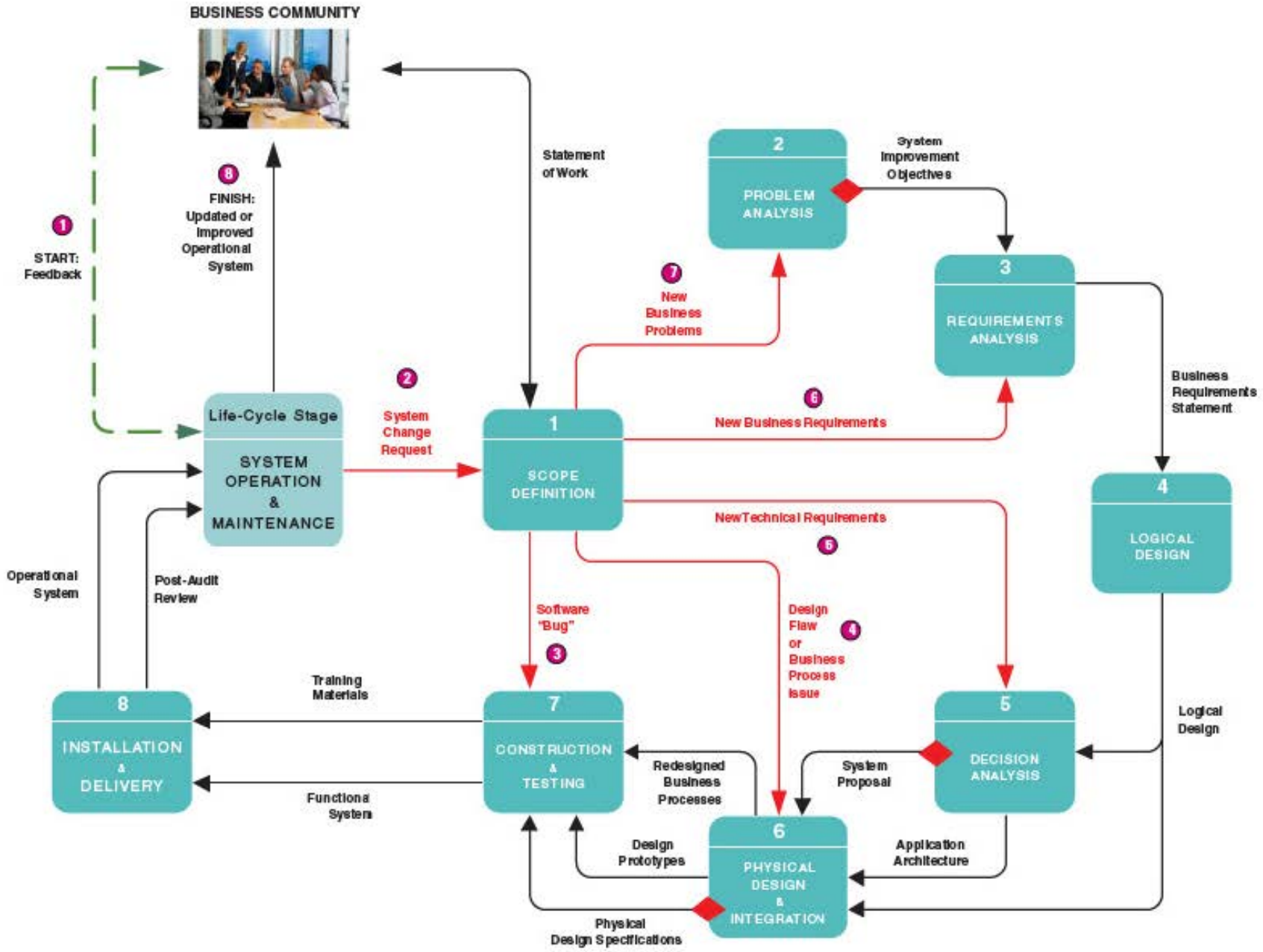


FIGURE 3-14 A System Maintenance Perspective

In some cases a BUSINESS PROCESS ISSUE may become apparent. In this case, only the business process needs to be redesigned.

5. On occasion, NEW TECHNICAL REQUIREMENTS might dictate a change. For example, an organization may standardize on the newest version of a particular database management system such as *SQL Server* or *Oracle*. For this type of project, the DECISION ANALYSIS phase may need to be revisited to first determine the risk and feasibility of converting the existing, operational database to the new version. As appropriate, such a project would subsequently proceed to the physical design, construction, and delivery phases as necessary.
6. Businesses constantly change; therefore, business requirements for a system also change. One of the most common triggers for a reengineering project is a NEW (or revised) BUSINESS REQUIREMENT. Given the requirement, the REQUIREMENTS ANALYSIS phase must be revisited with a focus on the impact of the new requirement on the existing system. Based on requirements analysis, the project would then proceed to the logical design, decision analysis, physical design, construction, and delivery phases.

   Time out! By now, you've noticed that maintenance and reengineering projects initiate in different phases of the basic methodology. You might be concerned that repeating these phases would require excessive time and effort. Keep in mind, however, that the scope of maintenance and reengineering projects is much, much smaller than the original project that created the operational system. Thus, the work required in each phase is much less than you might have guessed.
7. Again, as businesses change, significant NEW BUSINESS PROBLEMS, opportunities, and constraints can be encountered. In this type of project, work begins with the PROBLEM ANALYSIS phase and proceeds as necessary to the subsequent phases.
8. In all cases, the final result of any type of maintenance or reengineering project is an updated operational business system that delivers improved value to the system users and owners. Updates may include revised programs, databases, interfaces, or business processes.

As described earlier in the chapter, we expect all systems to eventually reach entropy. The business and/or technical problems and requirements have become so troublesome as to warrant "starting over."

That completes our introduction to the systems development methodology and routes. Before we end this chapter, we should introduce the role of automated tools that support systems development.

Automated Tools and Technology

You may be familiar with the old fable of the cobbler (shoemaker) whose own children had no shoes. That situation is not unlike the one faced by some systems developers. For years we've been applying information technology to solve our users' business problems; however, we've sometimes been slow to apply that same technology to our own problem—developing information systems. In the not-too-distant past, the principal tools of the systems analyst were paper, pencil, and flowchart template.

Today, entire suites of automated tools have been developed, marketed, and installed to assist systems developers. While system development methodologies do not always require automated tools, most methodologies do benefit from such technology. Some of the most commonly cited benefits include:

- Improved productivity—through automation of tasks.
- Improved quality—because automated tools check for completeness, consistency, and contradictions.
- Better and more consistent documentation—because the tools make it easier to create and assemble consistent, high-quality documentation.

- Reduced lifetime maintenance—because of the aforementioned system quality improvements combined with better documentation.
- Methodologies that really work—through rule enforcement and built-in expertise.

Chances are that your future employer is using or will be using this technology to develop systems. We will demonstrate the use of various automated tools throughout this textbook. There are three classes of automated tools for developers:

- Computer-aided systems modeling.
- Application development environments.
- Project and process managers.

Let's briefly examine each of these classes of automated tools.

REPRESENTATIVE CASE TOOLS

Computer Associates' *Erwin*
Oracle's *Designer 2000*
Popkin's *System Architect*
Rational ROSE
Visible Systems' *Visible Analyst*

> Computer-Assisted Systems Engineering

Systems developers have long aspired to transform information systems and software development into engineering-like disciplines. The terms *systems engineering* and *software engineering* are based on a vision that systems and software development can and should be performed with engineering-like precision and rigor. Such precision and rigor are consistent with the model-driven approach to systems development. To help systems analysts better perform system modeling, the industry developed automated tools called **computer-assisted software engineering (CASE)** tools. Think of CASE technology as software that is used to design and implement other software. This is very similar to the computer-aided design (CAD) technology used by most contemporary engineers to design products such as vehicles, structures, machines, and so forth. Representative modeling products are listed in the margin. Most modeling products run on personal computers, as depicted in Figure 3-15.

computer-assisted software engineering (CASE) the use of automated software tools that support the drawing and analysis of system models and associated specifications. Some CASE tools also provide prototyping and code generation capabilities.

CASE Repositories At the center of any true CASE tool's architecture is a developer's database called a **CASE repository.** The repository concept was introduced earlier (see Figure 3-7).

Around the CASE repository is a collection of tools, or facilities, for creating system models and documentation.

CASE repository a systems developers' database where developers can store system models, detailed descriptions and specifications, and other products of systems development. Synonyms data include data *dictionary* and *encyclopedia*.

CASE Facilities To use the repository, the CASE tools provide some combination of the following facilities, illustrated in Figure 3-16:

- *Diagramming tools* are used to draw the system models required or recommended in most system development methodologies. Usually, the shapes on one system model can be linked to other system models and to detailed descriptions (see next item below).
- *Dictionary tools* are used to record, delete, edit, and output detailed documentation and specifications. The descriptions can be associated with shapes appearing on system models that were drawn with the diagramming tools.
- *Design tools* can be used to develop mock-ups of system components such as inputs and outputs. These inputs and outputs can be associated with both the aforementioned system models and the descriptions.
- *Quality management tools* analyze system models, descriptions and specifications, and designs for completeness, consistency, and conformance to accepted rules of the methodologies.
- *Documentation tools* are used to assemble, organize, and report on system models, descriptions and specifications, and prototypes that can be reviewed by system owners, users, designers, and builders.
- *Design and code generator tools* automatically generate database designs and application programs or significant portions of those programs.
- *Testing tools* simulate transactions and data traffic, measure performance, and provide configuration management of test plans and test scripts.

forward engineering a CASE tool capability that can generate initial software or database code directly from system models.

reverse engineering a CASE tool capability that can automatically generate initial system models from software or database code.

Forward and Reverse Engineering As previous stated, CASE technology automates system modeling. Today's CASE tools provide two distinct ways to develop system models—**forward engineering** and **reverse engineering.** Think of reverse

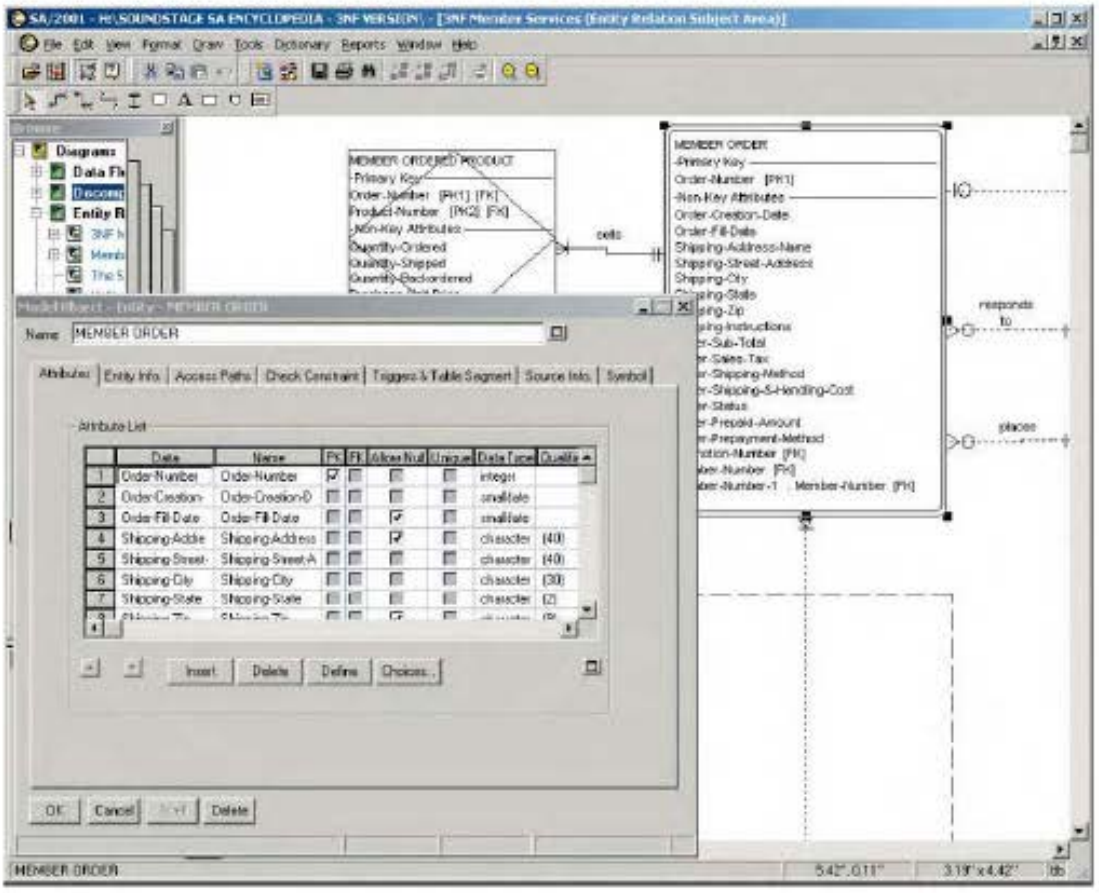


FIGURE 3-15 Screen Capture of *System Architect* CASE Tool

engineering as allowing you to generate a flowchart from an existing program and of forward engineering as allowing you to generate a program directly from a flowchart. CASE tools that allow for bidirectional, forward, and reverse engineering are said to provide for "round-trip engineering." For example, you reverse engineer a poorly designed system into a system model, edit and improve that model, and then forward engineer the new model into an improved system.

> Application Development Environments

The emphasis on speed and quality in software development has resulted in RAD approaches. The potential for RAD has been amplified by the transformation of programming language compilers into complete **application development environments (ADEs).** ADEs make programming simpler and more efficient. Indeed, most programming language compilers are now integrated into an ADE. Examples of ADEs (and the programming languages they support, where applicable) are listed in the margin.

Application development environments provide a number of productivity and quality management facilities. The ADE vendor provides some of these facilities. Third-party vendors provide many other facilities that can integrate into the ADE.

- *Programming languages* or *interpreters* are the heart of an ADE. Powerful debugging features and assistance are usually provided to help programmers quickly identify and solve programming problems.
- *Interface construction tools* help programmers quickly build the user interfaces using a component library.

REPRESENTATIVE ADEs

IBM's *Websphere (Java)*
Borland's *J Builder (Java)*
Macromedia's *Cold Fusion*
Microsoft's *Visual Studio.NET (VB .NET, C#, C++.NET)*
Oracle's *Developer*
Sybase's *Powerbuilder*

application development environment (ADE) an integrated software development tool that provides all the facilities necessary to develop new application software with maximum speed and quality. A common syncnym is *integrated development environment (IDE).*

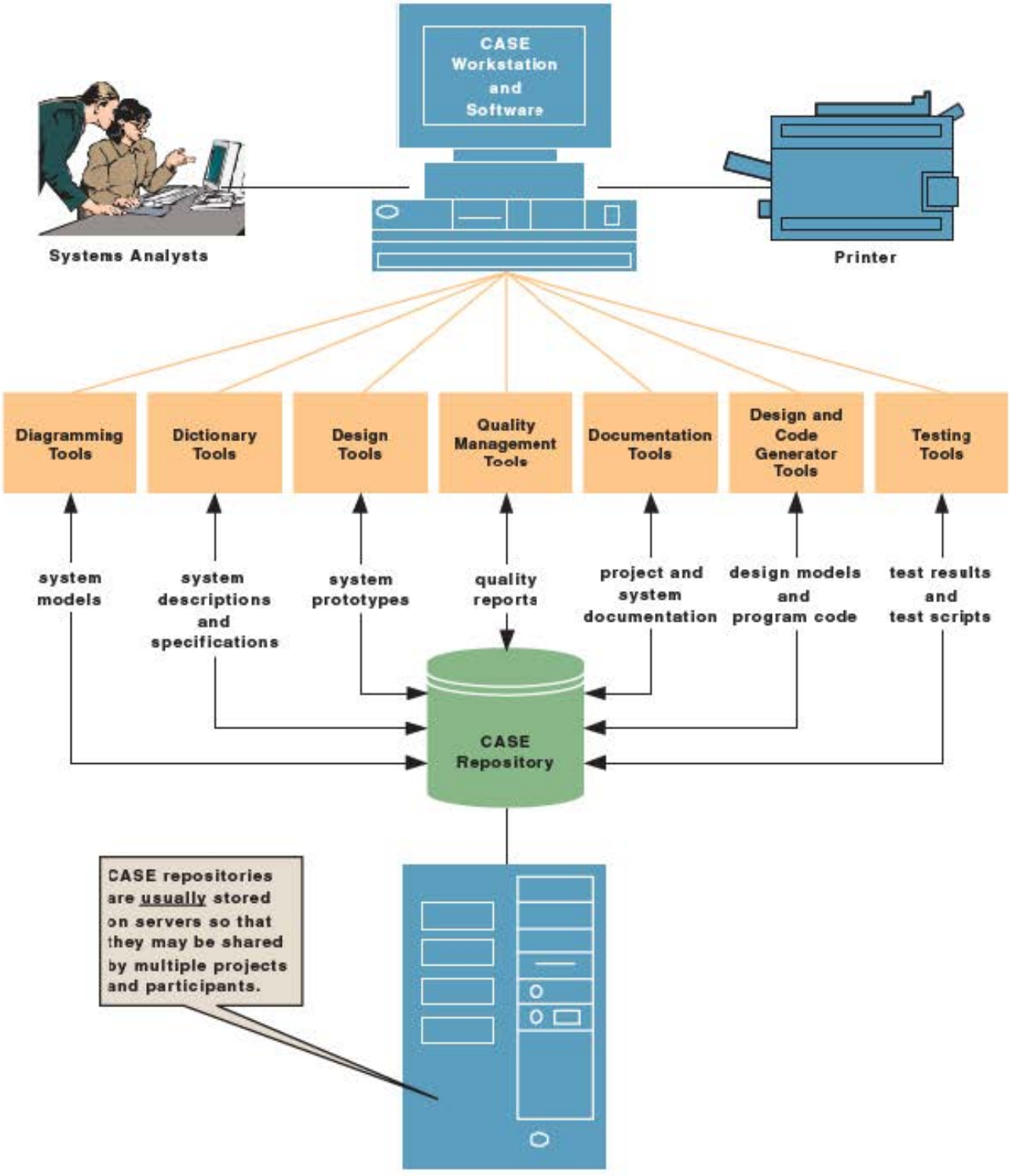


FIGURE 3-16 CASE Tool Architecture

- *Middleware* is software that helps programmers integrate the software being developed with various databases and computer networks.
- *Testing tools* are used to build and execute test scripts that can consistently and thoroughly test software.
- *Version control* tools help multiple programmer teams manage multiple versions of a program, both during development and after implementation.
- *Help authoring tools* are used to write online help systems, user manuals, and online training.
- *Repository links* permit the ADE to integrate with CASE tool products as well as other ADEs and development tools.

process manager application an automated tool that helps to document and manage a methodology and routes, its deliverables, and quality management standards. An emerging synonym is *methodware*.

> Process and Project Managers

A third class of automated tools helps us manage the system development methodology and projects that use the methodology. While CASE tools and ADEs are intended to support analysis, design, and construction of new information systems and software, **process manager application** and **project manager application** tools are intended to support cross life-cycle activities. Microsoft's *Project* and Niku's *Open Workbench* and *Project Manager* are examples of automated project management tools. Because process and project management is the subject of the next chapter, you'll learn more about these automated tools in that chapter.

project manager application an automated tool that helps to plan system development activities (preferably using the approved methodology), estimate and assign resources (including people and costs), schedule activities and resources, monitor progress against schedule and budget, control and modify schedule and resources, and report project progress.

Learning Roadmap

This chapter, along with the first two, completes the minimum context for studying systems analysis and design. We have described that context in terms of the three Ps—the participants (the stakeholders; Chapter 1), the product (the information system; Chapter 2), and the process (the system development; Chapter 3). Armed with this understanding, you are now ready to study systems analysis and/or design methods.

For many of you, Chapter 4, "Project Management," will provide a more complete context for studying systems analysis and design. It builds on Chapter 3 by emphasizing a variety of management issues related to the system development process. These include methodology management, resource management, management of expectations, change management, and others.

For those of you who skip the project and process management chapter, your next assignment will depend on whether your goal is:

- A comprehensive survey of systems development—We recommend you continue your sequential path to Chapter 5, "Systems Analysis." In that chapter you will study in greater depth the scope definition, problem analysis, requirements analysis, and logical design phases that were introduced in Chapter 3.
- The study of systems analysis techniques—Again, we recommend you continue your sequential path to Chapter 5, "Systems Analysis." Chapter 5 will provide a context for subsequently studying the tools and techniques of systems analysis.
- The study of systems design techniques—You might still want to quickly skim or review Chapter 5, "Systems Analysis," for context. Then continue your detailed study in Chapter 12, "System Design." In that chapter, you will learn more about the process and strategies for system design and construction of information systems.

Summary

1. A systems development process is a set of activities, methods, best practices, deliverables, and automated tools that stakeholders use to develop and continuously improve information systems and software.
2. The Capability Maturity Model (CMM) is a framework for assessing the maturity level of an organization's information systems development and management processes and products. It defines the need for a system development process.
3. A system life cycle divides the life of an information system into two stages, *systems development* and *systems operation and maintenance.*
4. A systems development methodology is a process for the system development stage. It defines a set of activities, methods, best practices, deliverables, and automated tools that systems developers and project managers are to use to develop and maintain information systems and software.
5. The following principles should underlie all systems development methodologies:
   a. Get the system users involved.
   b. Use a problem-solving approach.
   c. Establish phases and activities.
   d. Document throughout development.
   e. Establish standards.
   f. Manage the process and projects.
   g. Justify information systems as capital investments.
   h. Don't be afraid to cancel or revise scope.
   i. Divide and conquer.
   j. Design systems for growth and change.
6. System development projects are triggered by problems, opportunities, and directives:
   a. Problems are undesirable situations that prevent the organization from fully achieving its purpose, goals, and/or objectives.
   b. Opportunities are chances to improve the organization even in the absence of specific problems.
   c. Directives are new requirements that are imposed by management, government, or some external influence.
7. Wetherbe's PIECES framework is useful for categorizing problems, opportunities, and directives. The letters of the PIECES acronym correspond to Performance, Information, Economics, Control, Efficiency, and Service.
8. Traditional, basic systems development phases include:
   a. Scope definition
   b. Problem analysis
   c. Requirements analysis
   d. Logical design
   e. Decision analysis
   f. Physical design and integration
   g. Construction and testing
   h. Installation and delivery
9. Cross life-cycle activities are activities that overlap many or all phases of the methodology. They may include:
   a. Fact-finding, the formal process of using research, interviews, meetings, questionnaires, sampling, and other techniques to collect information about systems, requirements, and preferences.
   b. Documentation, the activity of recording facts and specifications for a system for current and future reference. Documentation is frequently stored in a repository, a database where systems developers store all documentation, knowledge, and products for one or more information systems or projects.
   c. Presentation, the activity of communicating findings, recommendations, and documentation for review by interested users and managers. Presentations may be either written or verbal.
   d. Feasibility analysis, the activity by which feasibility, a measure of how beneficial the development of an information system would be to an organization, is measured and assessed.
   e. Process management, the ongoing activity that documents, manages the use of, and improves an organization's chosen methodology (the "process") for systems development.
   f. Project management, the activity of defining, planning, directing, monitoring, and controlling a project to develop an acceptable system within the allotted time and budget.
10. There are different routes through the basic systems development phases. An appropriate route is selected during the scope definition phase. Typical routes include:
    a. Model-driven development strategies, which emphasize the drawing of diagrams to help visualize and analyze problems, define

business requirements, and design information systems. Alternative model-driven strategies include:

i) Process modeling
ii) Data modeling
iii) Object modeling

b. Rapid application development (RAD) strategies, which emphasize extensive user involvement in the rapid and evolutionary construction of working prototypes of a system to accelerate the system development process.
c. Commercial application package implementation strategies, which focus on the purchase and integration of a software package or solution to support one or more business functions and information systems.
d. System maintenance, which occurs after a system is implemented and lasts throughout the system's lifetime. Essentially, system maintenance executes a smaller-scale version of the development process with different starting points depending on the type of problem to be solved.

11. Automated tools support all systems development phases:

a. Computer-aided systems engineering (CASE) tools are software programs that automate or support the drawing and analysis of system models and provide for the translation of system models into application programs.
i) A CASE repository is a system developers' database. It is a place where developers can store system models, detailed descriptions and specifications, and other products of systems development.
ii) Forward engineering requires that the systems analyst draw system models, either from scratch or from templates. The resulting models are subsequently transformed into program code.
iii) Reverse engineering allows a CASE tool to read existing program code and transform that code into a representative system model that can be edited and refined by the systems analyst.

b. Application development environments (ADEs) are integrated software development tools that provide all the facilities necessary to develop new application software with maximum speed and quality.
c. Process management tools help us document and manage a methodology and routes, its deliverables, and quality management standards.
d. Project management tools help us plan system development activities (preferably using the approved methodology), estimate and assign resources (including people and costs), schedule activities and resources, monitor progress against schedule and budget, control and modify schedule and resources, and report project progress.

Review Questions

1. Explain why having a standardized system development process is important to an organization.
2. How are system life cycle and system development methodology related?
3. What are the 10 underlying principles for systems development?
4. Why is documentation important throughout the development process?
5. Why are process management and project management necessary?
6. What is risk management? Why is it necessary?
7. Which stakeholders initiate most projects? What is the impetus for most projects?
8. Who are the main participants in the scope definition? What are their goals in the scope definition?
9. What are the three most important deliverables in scope definition?
10. Who are the main participants in the requirements analysis phase? Why are they the main participants?
11. What feasibility analyses are made in the decision analysis?
12. What is model-driven development?
13. Why is model-driven development popular?
14. What is rapid application development (RAD)?
15. What benefits can RAD bring to the system development process?
16. What is computer-assisted software engineering (CASE)? List some examples of CASE.

Problems and Exercises

1. The Capability Maturity Model (CMM) was developed by the Software Engineering Institute at Carnegie Mellon, and is widely used by both the private and public sectors. What is the purpose of the CMM framework and how does it achieve this?
2. List the five maturity levels, and briefly describe each of them.
3. Table 3-1 in the textbook illustrates the difference in a typical project's duration, person-months, quality, and cost, depending upon whether an organization's system development process is at CMM level 1, 2, or 3. Between which two CMM levels does an organization gain the greatest benefit in terms of percentage of improvement? What do you think is the reason for this?
4. *Systems development methodology* and *system life cycle* are two terms that are frequently used and just as frequently misused. What is the difference between the two terms?
5. Describe how using a systems development methodology is in line with CMM goals and can help an organization increase its maturity level.
6. A number of underlying principles are common to all systems development methodologies. Identify these underlying principles and explain them.
7. The PIECES framework was developed by James Wetherbe as a means to classify problems. Identify the categories, then categorize the following problems using the PIECES framework:

   a. Duplicate data is stored throughout the system.
   b. There is a need to port an existing application to PDA devices.
   c. Quarterly sales reports need to be generated automatically.
   d. Employees can gain access to confidential portions of the personnel system.
   e. User interfaces for the inventory system are difficult and confusing, resulting in a high frequency of incorrect orders.

8. Each phase of a project includes specific deliverables that must be produced and delivered to the next phase. Using the textbook's hypothetical *FAST* methodology, what are the deliverables for the requirements analysis, logical design, and physical design/integration phases?
9. Scope definition is the first phase of the *FAST* methodology, and it is either the first phase or part of the first phase of most methodologies. What triggers the scope phase, which stakeholders are involved in this phase, what two essential questions need to be answered, and what three important deliverables come out of this phase?
10. The requirements analysis phase is an essential part of a system development methodology. According to the *FAST* methodology, which stakeholders typically participate in this phase? What is the primary focus of requirements analysis? What is *not* the focus? How should each proposed requirement be evaluated? What critical error must be avoided?
11. In the *FAST* methodology, as well as most system methodologies, system owners and system designers do not participate in the requirements analysis phase. What do you think the reason is for this?
12. What is the essential purpose of the logical design phase? How does it accomplish this? How are technological solutions incorporated in this phase? What are some common synonyms for this phase used by other methodologies? Who are the typical participants in this phase? What is agile modeling and what is its purpose? What are the deliverables coming out of this phase? In terms of the development team, what critical transition takes place by the end of this phase?
13. What is the essential purpose of the physical design phase? Who must be involved in this phase, and who may be involved? What are the two philosophies of physical design on the different ends of the continuum, and how are they different? Is this a likely phase in which a project might be canceled? With what other phase is there likely to be overlap, and what do you think is the reason for this?
14. A customer has engaged your software development company to develop a new order-processing system. However, the time frames are very tight and inflexible for delivery of at least the basic part of the new system. Further, user requirements are sketchy and unclear. What are two system development strategies that might be advantageous to use in this engagement?
15. What is the potential downside to using the strategies described in the preceding question?

Projects and Research

1. The Software Engineering Institute (SEI) at Carnegie Mellon University has developed a series of related Capability Maturity Models (CMMs). You can read about these different CMM products at SEI's Web site (http://www.sei.cmu.edu).

   a. Identify the current CMM products being maintained or developed by SEI.
   b. What are their differences and similarities?
   c. If you were to rate your organization, or an organization with which you are familiar, using the CMM described in the textbook, at which level would it be? Why?
   d. What steps would you recommend that your organization take in order to advance to the next CMM level?
   e. Do you feel that the time, cost, and resources to advance to the next level would be worth the perceived benefits for your organization? Why or why not?

2. You are a new project manager and have been assigned responsibility for an enterprise information systems project that touches every division in your organization. The chief executive officer stated at project initiation that successfully implementing this project was the number 1 priority of your organization. The project is in midst of the requirements analysis phase. While it is on schedule, you notice that attendance of the system users and owners at meetings on requirements has been dropping. A more experienced project manager has told you not to worry, that this is normal. Should you be concerned?

3. There are many different systems development methodologies in use, each with its own terminology, and number and scope of phases. Search the Web for information on two or three of these other systems development methodologies, then do the following:

   a. Note the systems development methodologies that you found. When, by whom, and why were they developed?
   b. What phases and terminology do they employ?
   c. Draw a matrix comparing their phases to the textbook's *FAST* methodology.
   d. What significant differences did you find?
   e. Do you see any advantages or disadvantages in any of the methodologies that you found compared to the *FAST* methodology?

4. The *PIECES* framework, which was developed by James Wetherbe and is described in the textbook, is intended to be a framework for classifying problems, opportunities, and directives.

   Contact one of the systems analysts for your organization, your school, and/or another organization. Ask them about the information systems used in their organization and to describe what the problems are in general terms. Select three of these systems:

   a. Describe the systems you selected, their problems, and the organizations that use them.
   b. Use the *PIECES* framework in the textbook to categorize each system's problems.
   c. Describe the *PIECES* category or categories you found for each problem. Did each problem generally have one or more categories associated with it?
   d. For the systems used by different organizations, what commonality did you find in the categories of problems? If you found a great deal of commonality of categories, do you think this is significant or just coincidental?
   e. Where in the systems development life cycle do you think the PIECES framework would be of the greatest value?

5. Computer-assisted software engineering (CASE) tools can significantly help developers improve productivity, quality, and documentation. Conduct an informal survey of about a dozen information technology departments regarding whether they use CASE tools, and if so, what type. Also, find out how long they have been using the CASE tools and whether they are used for all or just some projects. Add any other questions you may find meaningful. Try to split your survey between public and private sector agencies, and/or large and small organizations.

   a. What types of organizations did you survey?
   b. What did you ask?
   c. What were the results?
   d. Given the limited and informal nature of this survey, were you able to find any patterns or trends?
   e. Based upon your readings and your survey, what are your feelings regarding the use of CASE tools?

6. Projects at times are canceled or abandoned, sometime by choice, sometimes not. Research the Web for articles on project abandonment strategies, and select two of them.

a. What articles did you select?
b. What are their central themes, findings, and recommendations regarding project abandonment?
c. Compare and contrast their findings and their recommendations. Which strategy would you choose, if any?
d. Do you think that abandoning a project is always avoidable and/or always represents a failure?

Minicases

1. Interview at least two project managers. What are their experiences with *scope creep?*
2. George is the CEO of a major corporation that has been trying to develop a program that captures the keystrokes of employees on their computers. The project is currently $100,000 over budget and behind schedule and will require at least another $50,000 and six months to complete. The CEO wants to continue the project because so much has already been invested in it. What is your recommendation? Why?
3. Beatrice is an excellent manager—she is very capable of managing the bureaucratic process and following the business rules in her corporation. She is a "by the book" person who can always be counted on to do things "right." Will Beatrice make a good *project manager?* Why or why not?
4. A company is trying to decide between using an off-the-shelf program or developing a custom program for inventory management. The off-the-shelf product is less expensive than the custom solution and still has most of the needed functionality. The CEO believes that the missing capability can be addressed through tweaking the program once it is purchased. As the CIO of the company, what are your concerns and recommendations to the CEO?

Team and Individual Exercises

1. (Team) Hold team meetings using different communications mediums. Examples: phone, e-mail, virtual environment. What did you notice about the impact of the technology on the meeting? Was the productivity of the meetings the same for each medium? How did the team feel about the impact of the technologies on the team relationships?
2. (Individual) The analysis and design of an information system, done well, often eliminates jobs in a company. Economic theory strongly supports the creation of new jobs when this happens, but generally there is a time lag between the structural loss of jobs and the creation of new jobs. How do you feel about that?
3. (Team or Individual) Visit a neonatal intensive care unit at a highly regarded medical center (such as UC San Francisco). Write a short paper on your thoughts of the involved technology and the impact it has on people's lives. At the professor's discretion, share with the class.

Suggested Readings

Ambler, Scott. *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process.* New York: John Wiley & Sons, 2002. This is the definitive book on agile methods and modeling.

Application Development Trends (monthly periodical). Framingham, MA: Software Productivity Group, Inc. This is our favorite periodical for keeping up with the latest trends in methodology and automated tools. Each month features several articles on different topics and products.

DeMarco, Tom. *Structured Analysis and System Specification.* Englewood Cliffs, NJ: Prentice Hall, 1978. This is the classic book on structured systems analysis, a process-centered, model-driven methodology.

Gane, Chris. *Rapid Systems Development.* Englewood Cliffs, NJ: Prentice Hall, 1989. This book presents a nice overview of RAD that combines model-driven development and prototyping in the correct balance.

Gildersleeve, Thomas. *Successful Data Processing Systems Analysis,* 2nd ed. Englewood Cliffs, NJ: Prentice Hall,

1985. We are indebted to Gildersleeve for the creeping commitment approach. The classics never become obsolete.

Jacobson, Ivar; Grady Booch; and James Rumbaugh. *The Unified Software Development Process.* Reading, MA: Addison-Wesley, 1999. The Rational Unified Process is a currently popular example of a model-driven, object-oriented methodology.

McConnell, Steve. *Rapid Development.* Redmond, WA: Microsoft Press, 1996. Chapter 7 of this excellent reference book provides what may be the definitive summary of system development life cycle and methodology variations that we call "routes" in our book.

Orr, Ken. *The One Minute Methodology.* New York: Dorsett House, 1990. Must reading for those interested in exploring the need for methodology. This very short book can be read in one sitting. It follows the satirical story of an analyst's quest for the development silver bullet, "the one minute methodology."

Paulk, Mark C.; Charles V. Weber; Bill Curtis; and Mary Beth Chrissis. *The Capability Maturity Model: Guidelines for Improving the Software Process.* Reading, MA: Addison-Wesley, 1995. This book fully describes version 1.1 of the Capability Maturity Model. Note that version 2.0 was under development at the time we were writing this chapter.

Wetherbe, James. *Systems Analysis and Design: Best Practices,* 4th ed. St. Paul, MN: West, 1994. We are indebted to Wetherbe for the PIECES framework.

Strategic Enterprise Plan
Strategic Information Systems Plan
Goal: Improve Business KNOWLEDGE
Goal: Improve Business PROCESSES
Goal: Improve Business COMMUNICATIONS
PROJECT MANAGERS and SYSTEMS ANALYSTS
SYSTEM USERS
SYSTEM OWNERS
SYSTEM OWNERS
SYSTEM DESIGNERS
SYSTEM BUILDERS
STATEMENT OF WORK
PROBLEM STATEMENT (using the PIECES framework)
INFORMATION SCOPE & VISION
FUNCTIONAL SCOPE & VISION
COMMUNICATIONS SCOPE & VISION
SYSTEM IMPROVEMENT OBJECTIVES (using the PIECES framework)
BUSINESS REQUIREMENTS STATEMENT
BUSINESS DATA REQUIREMENTS
LOGICAL DATA MODELS
BUSINESS PROCESS REQUIREMENTS
LOGICAL PROCESS MODELS
BUSINESS & SYSTEM INTERFACE REQUIREMENTS
LOGICAL INTERFACE MODELS
SYSTEM PROPOSAL (or REQUEST FOR SYSTEM PROPOSALS)
ARCHITECTURAL MODEL
DESIGN PROTOTYPES
PHYSICAL DATABASE DESIGN SPECIFICATIONS
BUSINESS PROCESS DESIGN
PHYSICAL SOFTWARE DESIGN SPECIFICATIONS
PHYSICAL USER & SYSTEM INTERFACE DESIGN SPECIFICATIONS
FUNCTIONAL SYSTEM
TRAINING MATERIALS
DATABASE SOLUTION
MIDDLEWARE
COMMERCIAL SOFTWARE PACKAGES
CUSTOM-BUILT APPLICATION SOFTWARE
MIDDLEWARE
USER INTERFACE SOLUTIONS
SYSTEM INTERFACE SOLUTIONS
OPERATIONAL SYSTEM
POST-AUDIT REVIEW
SCOPE DEFINITION
PROBLEM ANALYSIS
REQUIREMENTS ANALYSIS
LOGICAL DESIGN
DECISION ANALYSIS
PHYSICAL DESIGN
CONSTRUCTION & TESTING
INSTALLATION & DELIVERY
PROJECT and PROCESS MANAGEMENT
FEASIBILITY ANALYSIS and RISK MANAGEMENT
FACT-FINDING TECHNIQUES: Sampling Research Observation Questionnaire Interview Prototyping JAD
Constraint: APPROVED DATABASE TECHNOLOGIES
Constraint: APPROVED PROCESS TECHNOLOGIES
Constraint: APPROVED INTERFACE TECHNOLOGIES
Constraint APPROVED NETWORKTECHNOLOGIES
Strategic Enterprise Information Technology Architecture

4 Project Management

Chapter Preview and Objectives

Project management skills are greatly in demand in the information technology community. Project management is a natural extension of the previous chapter's introduction to system development. This chapter provides a process-centric survey of key project management tools and techniques as they apply to systems analysis and design. You will know that you understand the basics of project management when you can:

- Define the terms *project* and *project management* and differentiate between project and process management.
- Describe the causes of failed information systems and technology projects.
- Describe the basic competencies required of project managers.
- Describe the basic functions of project management.
- Differentiate between PERT and Gantt charts as project management tools.
- Describe the role of project management software as it relates to project management tools.
- Describe eight activities in project management.
- Define *joint project planning* and its role in project management.
- Define *scope* and write a *statement of work* to document scope.
- Use a *work breakdown structure* to decompose a project into tasks.
- Estimate tasks' durations and specify intertask dependencies on a PERT chart.
- Assign resources to a project and produce a project schedule with a Gantt chart.
- Assign people to tasks and direct the team effort.
- Use critical path analysis to adjust schedule and resource allocations in response to schedule and budget deviations.
- Manage user expectations of a project and adjust project scope.

Introduction

Bob Martinez was in the office of his boss, Sandra Shepherd, discussing the SoundStage Member Services system project.

"This sure is a big project," Bob said, "bigger than anything I've ever worked on before. How will you make sure it stays on track?"

"Well, first we have to get consensus on the scope of the project and document assumptions and constraints," Sandra answered. "We also have to negotiate the project budget and schedule. Then we identify all the tasks that need to be performed. The *FAST* methodology is our template, but we always customize it for each project. We have to plan each task and analyze how its work and its own schedule fit in with the overall project. Then we assign people and other resources to each task. As the project goes on we have to manage the process to make sure everything stays on schedule."

"Wow!" Bob replied. "On some of the semester group projects I did in college, we just kind of dived right in with the work. If we got behind we just pulled a couple of all-nighters."

"Believe me," Sandra said, "you don't want to be around when the finger pointing starts on a real systems project that is behind schedule or over budget. That's something a couple of all-nighters won't fix. We have a long road ahead of us, and we want to plan this as carefully as possible."

What Is Project Management?

Most of you are familiar with Murphy's Law, which suggests, "If anything can go wrong, it will." Murphy has motivated numerous pearls of wisdom about projects, machines, people, and why things go wrong. This chapter will teach you strategies, tools, and techniques for project management as applied to information systems projects.

project manager the person responsible for supervising a systems project from initiation to conclusion. Successful project managers possess a wide range of technical, management, leadership, and communication skills.

The demand for **project managers** in the information systems community is strong. Typically, IS project managers come from the ranks of experienced IS developers such as systems analysts. While it is unlikely that your first job responsibilities will include project management, you should immediately become aware of project management processes, tools, and techniques. Eventually you will combine this knowledge with development experience plus your own observation of project managers to form the basis for your own career opportunities in project management.

Before we can define *project management*, we should first define *project*. There are as many definitions as there are authors, but we like the definition put forth by Wysocki, Beck, and Crane:

project a sequence of activities that must be completed on time, within budget, and according to specification.

> A **project** is a [temporary] sequence of unique, complex, and connected activities that have one goal or purpose and that must be completed by a specific time, within budget, and according to specification.[1]

The keywords are underlined to draw your attention to some key aspects of the definition. As applied to information system development, we note the following:

- A system development process or methodology, such as *FAST*, defines a sequence of activities, mandatory and optional.
- Every system development project is unique; that is, it is different from every other system development project that preceded it.
- The activities that comprise system development are relatively complex. They require the skills that you are learning in this book, and they require that you be able to adapt concepts and skills to changing conditions and unanticipated events.

[1] Robert K. Wysocki, Robert Beck, Jr., and David B. Crane, *Effective Project Management: How to Plan, Manage, and Deliver Projects on Time and within Budget* (New York: John Wiley & Sons, 1995), p. 38.

- By now, you've already learned that the phases and activities that make up a system development methodology are generally sequential. While some tasks may overlap, many tasks are dependent on the completion of other tasks.
- The development of an information system represents a goal. Several objectives may need to be met to achieve that goal.
- Although many information system development projects do not have absolute deadlines or specified times (there are exceptions), they are notoriously completed later than originally projected. This is becoming less acceptable to upper management given the organizationwide pressures to reduce cycle times for products and business processes.
- Few information systems are completed within budget. Again, upper management is increasingly rejecting this tendency.
- Information systems must satisfy the business, user, and management expectations according to specifications (which we call *requirements* throughout this book).

For any systems development project, effective **project management** is necessary to ensure that the project meets the deadline, is developed within an acceptable budget, and fulfills customer expectations and specifications. You learned in Chapter 3 that project management is a cross life-cycle activity because it overlaps all phases of any systems development methodology.

project management the process of scoping, planning, staffing, organizing, directing, and controlling the development of an acceptable system at a minimum cost within a specified time frame.

Corporate rightsizing has changed the structure and culture of most organizations and, hence, project management. More flexible and temporary interdepartmental teams that are given greater responsibility and authority for the success of organizations have replaced rigid hierarchical command structures and permanent teams. Contemporary system development methodologies depend on having teams that include both technical and nontechnical users, managers, and information technologists all directed to the project goal. These dynamic teams require leadership and project management.

Organizations take different approaches to project management. One approach is to appoint a project manager from the ranks of the team (once it has been formed). Alternatively, many organizations believe that successful project managers apply a unique body of knowledge and skills that must be learned. These organizations tend to hire and/or develop professional project managers who are then assigned to one or more projects.

The prerequisite for good project management is a well-defined system development process. In Chapter 3, we introduced the Capability Maturity Model (CMM) as a framework for quality management that is based on a sound systems development process. In Chapter 3 we differentiated between project and process management. Project management was defined above. **Process management** is an ongoing activity that documents, manages the use of, and improves an organization's chosen methodology (the "process") for systems development. Process management is concerned with the activities, deliverables, and quality standards to be applied to *all* projects. The scope of process management is all projects, whereas the scope of project management is a single project. This chapter will focus on project management.

process management the activity of documenting, managing, and continually improving the process of systems development.

> The Causes of Failed Projects

What causes a project to succeed or fail? Chapter 3 introduced 10 basic principles of systems development that are critical success factors for all projects. See Chapter 3 for a review of those principles. From a project management perspective, a project is considered a success if:

- The resulting information system is acceptable to the customer.
- The system is delivered "on time."
- The system is delivered "within budget."
- The system development process had a minimal impact on ongoing business operations.

Not all projects meet these criteria, and as a result, not all projects are successful. Failures and limited successes far outnumber successful information systems. Project mismanagement can undermine the best application of the systems analysis and design methods taught in this book. We can develop an appreciation for the importance of project management by studying the mistakes of some project managers. Let's examine some project mismanagement problems and consequences:

- *Failure to establish upper-management commitment to the project*—Sometimes commitment changes during a project.
- *Lack of organization's commitment to the system development methodology*—Many system development methodologies do little more than collect dust.
- *Taking shortcuts through or around the system development methodology*—Project teams often take shortcuts for one or more of the following reasons:
  - The project gets behind schedule, and the team wants to catch up.
  - The project is over budget, and the team wants to make up costs by skipping steps.
  - The team is not trained or skilled in some of the methodology's activities and requirements, so it skips them.
- *Poor expectations management*—All users and managers have expectations of the project. Over time, these expectations may change. This can lead to two undesirable situations:
  - **Scope creep** is the unexpected growth of user expectations and business requirements for an information system as the project progresses. The schedule and budget can be adversely affected by such changes.
  - **Feature creep** is the uncontrolled addition of technical features to a system under development without regard to schedule or budget.
- *Premature commitment to a fixed budget and schedule*—You can rarely make accurate estimates of project costs and schedule before completing a detailed problem analysis or requirements analysis. Premature estimates are inconsistent with the creeping commitment approach introduced in Chapter 3.
- *Poor estimating techniques*—Many systems analysts estimate by making a best-calculated estimate and then doubling that number. This is not a scientific approach.
- *Overoptimism*—Systems analysts and project managers tend to be optimists. As project schedules slip, they respond, "No big deal. We can make it up later." They fail to recognize that certain tasks are dependent on other tasks. Because of these dependencies, a schedule slip in one phase or activity will cause corresponding slips in many other phases and activities, thus contributing to cost overruns.
- *The mythical man-month*[2]—As the project gets behind schedule, project leaders frequently try to solve the problem by assigning more people to the team. It doesn't work! There is no linear relationship between time and number of personnel. The addition of personnel usually creates more communication problems, causing the project to get even further behind schedule.
- *Inadequate people management skills*—Managers tend to be thrust into management positions and are not prepared for management responsibilities. This problem is easy to identify. No one seems to be in charge; customers don't know the status of the project; teams don't meet regularly to discuss and monitor progress; team members aren't communicating with one another; the project is always said to be "95 percent complete."
- *Failure to adapt to business change*—If the project's importance changes during the project, or if the management or the business reorganizes,

scope creep the unexpected and gradual growth of requirements during an information systems project.

feature creep the uncontrolled addition of technical features to a system.

[2]Fred Brooks, *The Mythical Man-Month* (Reading, MA: Addison-Wesley, 1975).

projects should be reassessed for compatibility with those changes and their importance to the business.

- *Insufficient resources*—This could be due to poor estimating or to other priorities, or it could be that the staff resources assigned to a project do not possess the necessary skills or experience.
- *Failure to "manage to the plan"*—Various factors may cause the project manager to become sidetracked from the original project plan.

Ultimately, *the* major cause of project failure is that most project managers were not educated or trained to be project managers. Just as good programmers don't always go on to become good systems analysts, good systems analysts don't automatically perform well as project managers. To be a good project manager, you should be educated and skilled in the "art of project management."

> The Project Management Body of Knowledge

The Project Management Institute was created as a professional society to guide the development and certification of *professional* project managers. The institute created the Project Management Body of Knowledge (PMBOK) for the education and certification of professional project managers. This chapter's content was greatly influenced by the PMBOK.

Project Manager Competencies Good project managers possess a core set of competencies. Table 4-1 summarizes these competencies. Some of these competencies can be taught, in courses, books, and professional workshops; however, some come only with professional experience in the field. There are two basic premises of project management competencies: First, individuals cannot manage a process they

TABLE 4-1 Project Manager Competencies

Competency	Explanation	How to Obtain?
	Business Achievement Competencies	
Business awareness	Ties every systems project to the mission, vision, and goals of the organization.	Business experience
Business partner orientation	Keeps managers and users involved throughout a systems project.	Business experience
Commitment to quality	Ensures that every systems project contributes to the quality expectation of the organization as a whole.	Business experience
	Problem-Solving Competencies	
Initiative	Demonstrates creativity, calculated risks, and persistence necessary to get the job done.	Business experience
Information gathering	Skillfully obtains the factual information necessary to analyze, design, and implement the information system.	Chapter 6 in this book plus business experience
Analytical thinking	Can assess and select appropriate system development processes and use project management tools to plan, schedule, and budget for system development.	This chapter
	Can solve problems through the analytical approach of decomposing systems into their parts and then reassembling the parts into improved systems.	Chapters 8, 9, and 11 in this book plus business experience
Conceptual thinking	Understands systems theory and applies it to systems analysis and design of information systems.	Chapters 2 and 5–11 in this book

continued

TABLE 4-1 (Continued)

Competency	Explanation	How to Obtain?
	Influence Competencies	
Interpersonal awareness	Understands, recognizes, and reacts to interpersonal motivations and behaviors.	Can be learned in courses but requires business experience
Organizational awareness	Understands the politics of the organization and how to use them in a project.	Business experience
Anticipation of impact	Understands implications of project decisions and manages expectations and risk.	Introduced in this chapter but requires business experience
Resourceful use of influence	Skillfully obtains cooperation and consensus of managers, users, and technologists to solutions.	Business experience
	People Management Competencies	
Motivating others	Coaches and directs individuals to overcome differences and achieve project goals as a team.	Business experience
Communication skills	Communicates effectively, both orally and in writing, in the context of meetings, presentations, memos, and reports.	Can be learned in courses but usually requires business experience
Developing others	Ensures that project team members receive sufficient training, assignments, supervision, and performance feedback required to complete projects.	Business experience
Monitoring and controlling	Develops the project plan, schedule, and budget and continuously monitors progress and makes adjustments when necessary.	Tools and techniques taught in this chapter, but requires project experience
	Self-Management Competencies	
Self-confidence	Consistently makes and defends decisions with a strong personal confidence in the process and/or facts.	Business experience
Stress management	Works effectively under pressure or adversity.	Business experience
Concern for credibility	Consistently and honestly delivers on promises and solutions. Maintains technical or business currency in the field as appropriate.	Business experience
Flexibility	Capable of adjusting process, management style, or decision making based on situations and unanticipated problems.	Business experience

Source: Adapted from Robert K. Wysocki, Robert Beck, Jr., and David B. Crane, *Effective Project Management: How to Plan, Manage, and Deliver Projects on Time and within Budget* (New York: John Wiley & Sons, 1995).

have never used. Second, managers must have an understanding of the business and culture that provides a context for the project.

Project Management Functions The basic functions of a project manager have been studied and refined by management theorists for many years. These functions include scoping, planning, staffing, organizing, scheduling, directing, controlling, and closing:

- *Scoping*—Scope defines the boundaries of the project. A project manager must scope project expectations and constraints in order to plan activities, estimate costs, and manage expectations.
- *Planning*—Planning identifies the tasks required to complete the project. This is based on the manager's understanding of the project scope and the methodology used to achieve the goal.
- *Estimating*—Each task that is required to complete the project must be estimated. How much time will be required? How many people will be needed?

What skills will be needed? What tasks must be completed before other tasks are started? Can some of the tasks overlap? How much will it cost? These are all estimating issues. Some of these issues can be resolved with the project modeling tools that will be discussed later in this chapter.

- *Scheduling*—Given the project plan, the project manager is responsible for scheduling all project activities. The project schedule should be developed with an understanding of the required tasks, task duration, and task prerequisites.
- *Organizing*—The project manager should make sure that members of the project team understand their own individual roles and responsibilities as well as their reporting relationship to the project manager.
- *Directing*—Once the project has begun, the project manager must direct the team's activities. Every project manager must demonstrate people management skills to coordinate, delegate, motivate, advise, appraise, and reward team members.
- *Controlling*—Perhaps the manager's most difficult and important function is controlling the project. Few plans will be executed without problems and delays. The project manager must monitor and report progress against goals, schedule, and costs and make appropriate adjustments when necessary.
- *Closing*—Good project managers always assess successes and failures at the conclusion of a project. They learn from their mistakes and plan for continuous improvement of the systems development process.

All the above functions are dependent on ongoing interpersonal *communication* among the project manager, the team, and other managers.

Project Management Tools and Techniques—PERT and Gantt Charts The PMBOK includes tools and techniques that support project managers. Two such tools are PERT and Gantt charts.

PERT, which stands for *Project Evaluation and Review Technique*, was developed in the late 1950s to plan and control large weapons development projects for the U.S. Navy. A **PERT chart** is a graphical network model that depicts a project's tasks and the relationships between those tasks. A sample PERT chart is illustrated in Figure 4-1. PERT was developed to make clear the *interdependence* between project tasks before those tasks are scheduled. The boxes represent project tasks (we used phases from Chapter 3). (The content of the boxes can be adjusted to show various project attributes such as schedule and actual start and finish times.) The arrows indicate that one task is dependent on the start or completion of another task.

PERT chart a graphical network model used to depict the interdependencies between a project's tasks.

The Gantt chart, first conceived by Henry L. Gantt in 1917, is the most commonly used project scheduling and progress evaluation tool. A **Gantt chart** is a simple horizontal bar chart that depicts project tasks against a calendar. Each bar represents a named project task. The tasks are listed vertically in the left-hand column. The horizontal axis is a calendar time line. Figure 4-2 illustrates a phase-level Gantt chart, once again based on Chapter 3. We used the same project that was illustrated in Figure 4-1.

Gantt chart a bar chart used to depict project tasks against a calendar.

Gantt charts offer the advantage of clearly showing *overlapping* tasks, that is, tasks that can be performed at the same time. The bars can be shaded to clearly indicate percentage completion and project progress. The figure demonstrates which phases are ahead of and behind schedule at a glance. The popularity of Gantt charts stems from their simplicity—they are easy to learn, read, prepare, and use.

Gantt and PERT charts are not mutually exclusive. Gantt charts are more effective when you are seeking to communicate schedule. PERT charts are more effective when you want to study the relationships between tasks.

Project Management Software Project management software is routinely used to help project managers plan projects, develop schedules, develop budgets, monitor progress and costs, generate reports, and effect change. Representative automated project management tools are listed in the margin.

PROJECT MANAGEMENT SOFTWARE

Niku's *Project Manager*
Artemis *International Solutions Corporation's 9000*
Computer Associates' *AllFusion Process Management Suite*
Microsoft's *Project*
Primavera's *Project Planner* and *Project Manager*
C/S Solutions' *Risk +*.

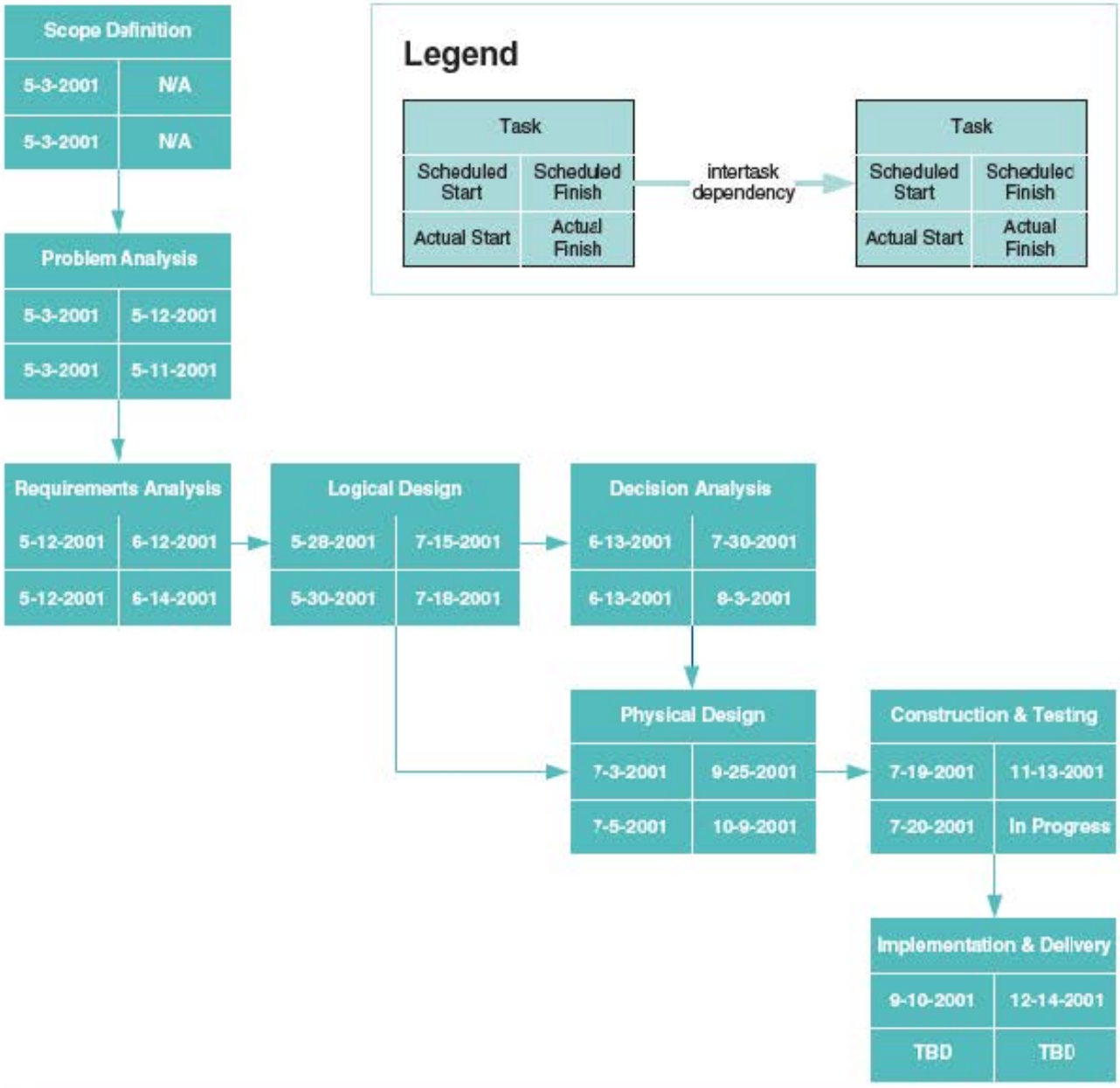


FIGURE 4-1 A PERT Chart

We will teach you project modeling and management techniques in the context of project management software. We used Microsoft *Project* because that tool is frequently available to students and institutions at special academic prices through their college bookstore. Microsoft *Project*, like most project management software tools, supports both PERT and Gantt charts.

Figure 4-3(a) illustrates one possible Microsoft *Project* Gantt chart for the SoundStage Member Services project. We call your attention to the following numbered bullets:

1. The black bars are *summary tasks* that represent project *phases* that are further broken down into other tasks.
2. The red bars indicate tasks that have been determined to be critical to the schedule, meaning that any extension to the duration of those tasks will delay other tasks and the project as a whole. We'll talk more about critical tasks later.

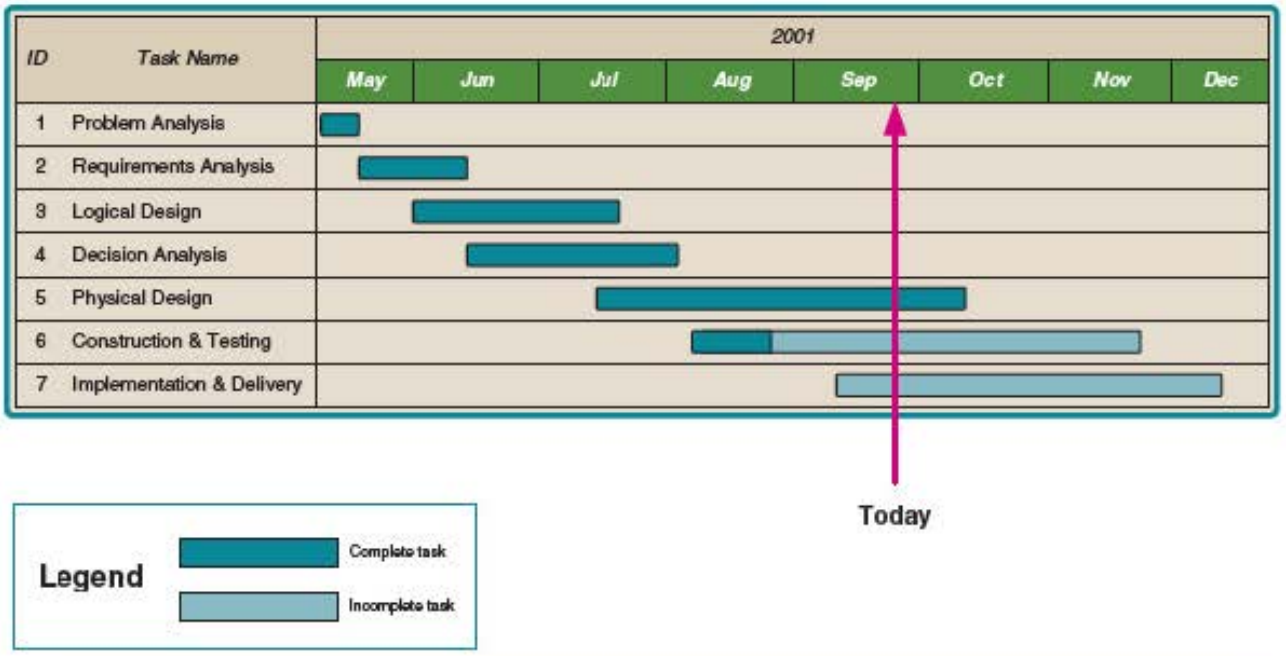


FIGURE 4-2 A Gantt Chart

3. The blue bars indicate tasks that are not critical to the schedule, meaning they have some slack time during which delays will not affect other tasks and the project as a whole.
4. The red arrows indicate prerequisites between two critical tasks. (The blue arrows indicate prerequisites between two noncritical tasks.)
5. The teal diamonds indicate milestones—events that have no duration. They signify the end of some significant task or deliverable.

Figure 4-3(b) shows a Microsoft *Project* PERT chart based on the same project plan that was illustrated in the Gantt chart. The contents of each cell in the task rectangles are able to be customized in Microsoft *Project*.

The Project Management Life Cycle

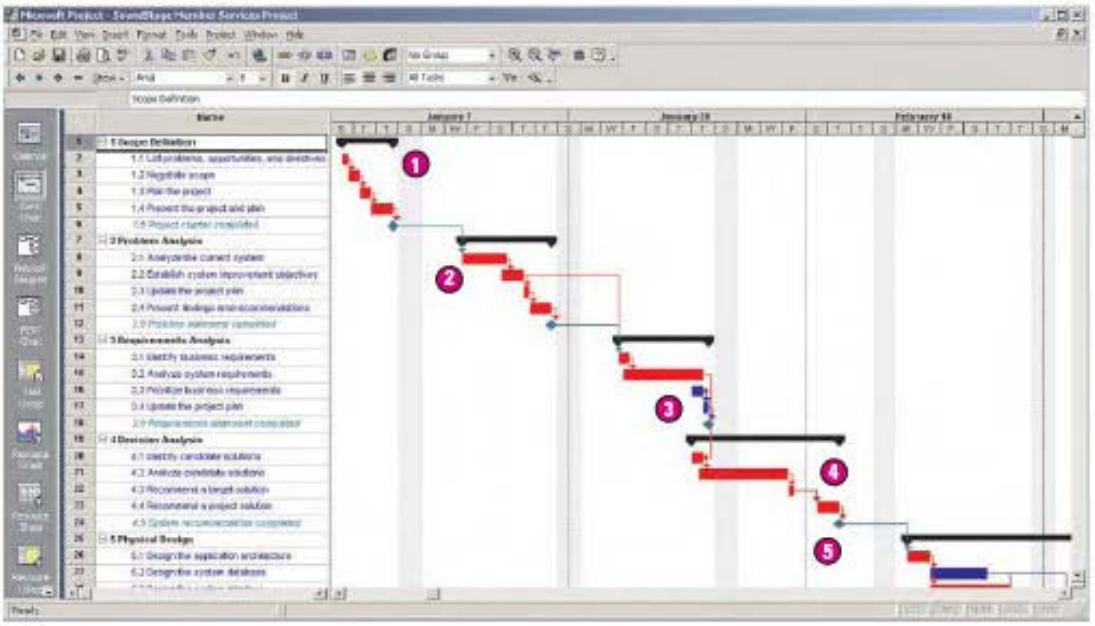
Recall from Chapter 3 that the Capability Maturity Model defines a framework for assessing the quality of an organization's information systems development activities. CMM Level 1 is defined as "initial" and is characterized by the lack of any consistent project or process management function. The first stage of maturity improvement is to implement a consistent project management function—called CMM Level 2. In this section we introduce a project management life cycle representative of CMM Level 2 maturity.

Figure 4-4 illustrates a project management process or life cycle. Recall that project management is a cross life-cycle activity; that is, project management activities overlap all the system development phases that were introduced in Chapter 3. The illustrated project management activities correspond to classic management functions: scoping, planning, estimating, scheduling, organizing, directing, controlling, and closing.

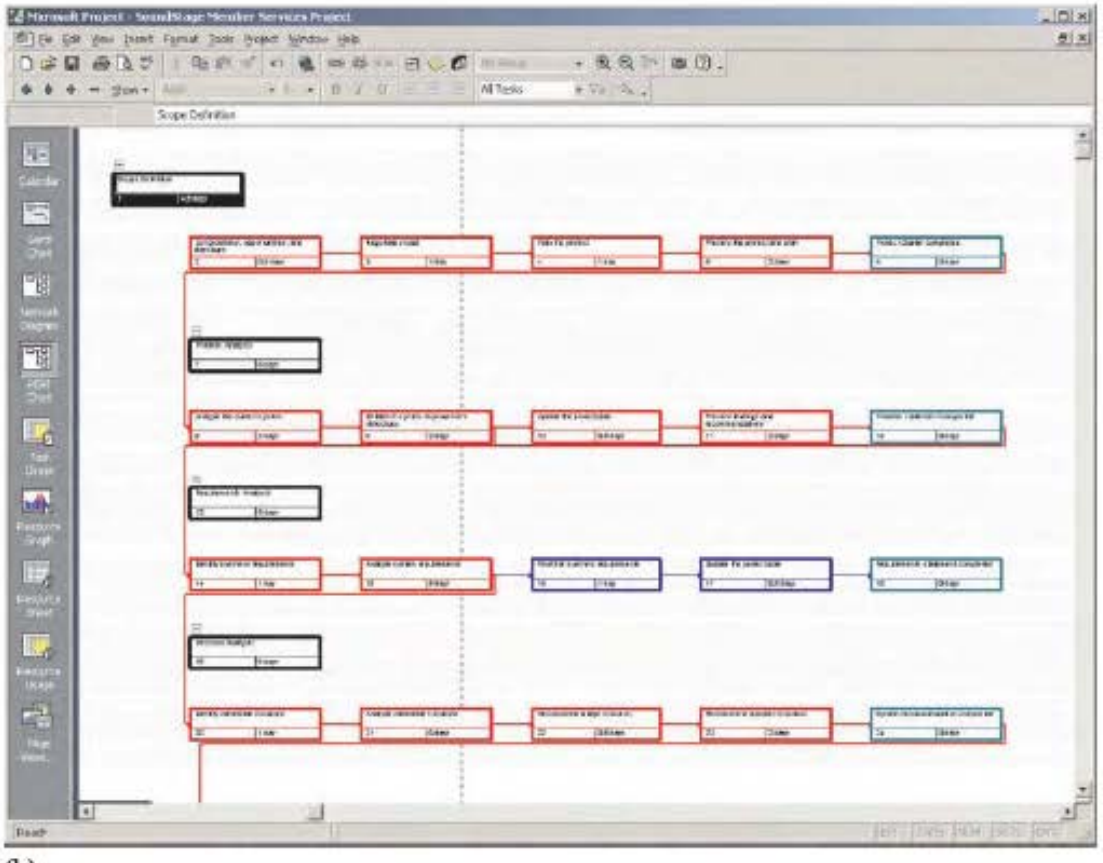
The project management process shown in Figure 4-4 incorporates a joint project planning (JPP) technique.[3] **Joint project planning (JPP)** is a strategy wherein all

joint project planning (JPP) a strategy in which all stakeholders attend an intensive workshop aimed at reaching consensus on project decisions.

[3]Wysocki, Beck, and Crane. *Effective Project Management: How to Plan, Manage, and Deliver Projects on Time and within Budget*, p. 38.



(a)



(b)

FIGURE 4-3 Microsoft *Project* Gantt and PERT Charts

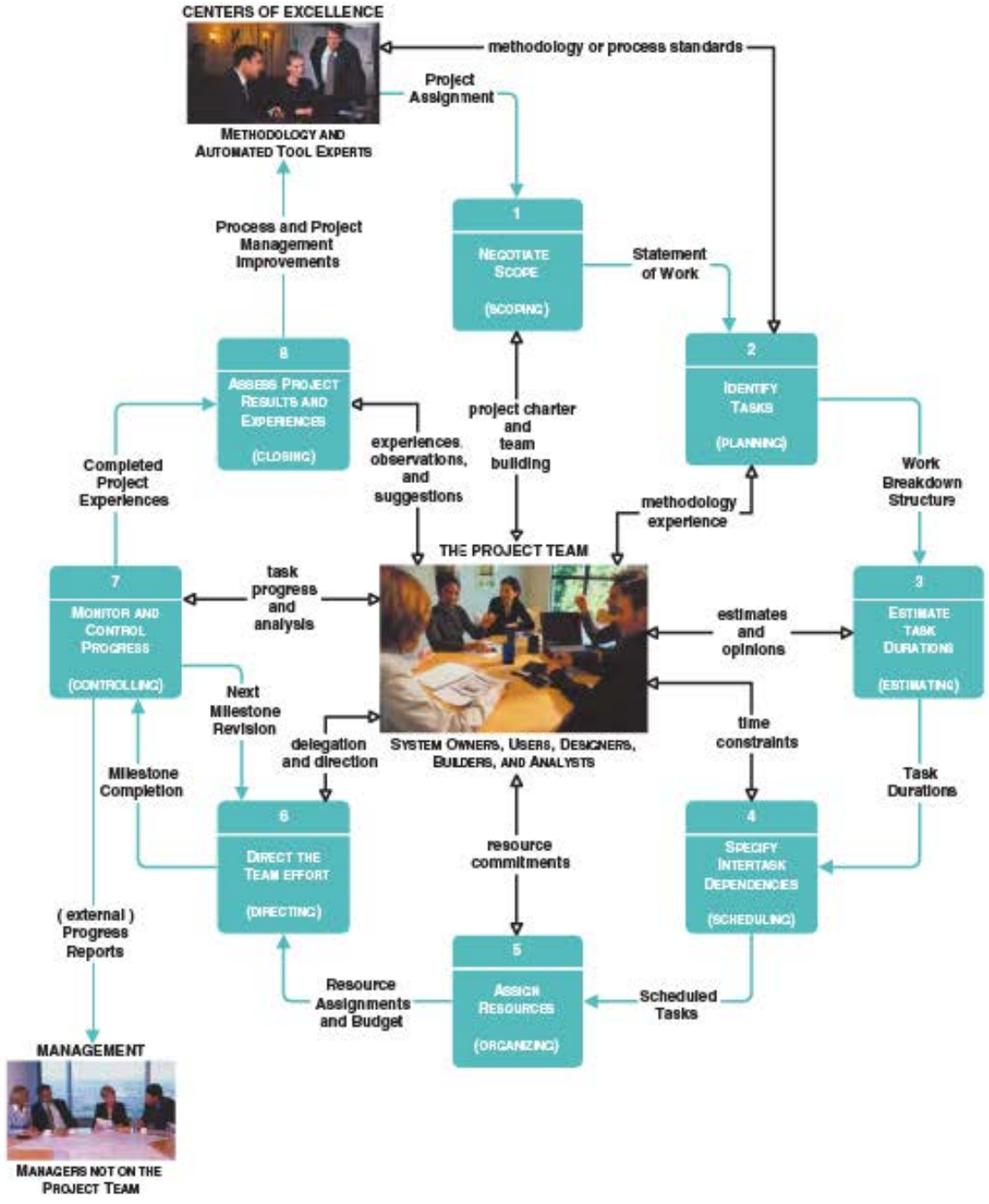


FIGURE 4-4 A Project Management Life Cycle

stakeholders in a project (meaning system owners, users, analysts, designers, and builders) participate in a one- to three-day project management workshop, the result of which is consensus on project scope, schedule, resources, and budget. (Subsequent workshops or meetings may be required to adjust scope, budget, and schedule.) Notice that in JPP, the project team is *actively* involved in all inputs and deliverables of all project management activities.

In the following subsections, we will review each of the illustrated project management activities and discuss how to use appropriate project management tools and techniques.

> Activity 1—Negotiate Scope

Perhaps the most important prerequisite to effective project management occurs at the beginning. All parties must agree to the project scope before any attempt is made to identify and schedule tasks or to assign resources (people) to those tasks. Scope defines the expectations of a project, and expectations ultimately determine satisfaction and degrees of success. Accordingly, the negotiation of project scope is a necessary activity in the project management life cycle. What is scope? **Scope** defines the boundaries of a project—the parts of the business that are to be studied, analyzed, designed, constructed, implemented, and ultimately improved. Scope also defines the aspects of a system that are considered outside the project. The answers to five basic questions influence the negotiation of project scope:

scope the boundaries of a project—the areas of a business that a project may (or may not) address.

- *Product*—What do you want?
- *Quality*—How good do you want it to be?
- *Time*—When do you want it?
- *Cost*—How much are you willing to pay for it?
- *Resources*—What resources are you willing or able to bring to the table?

Negotiation of the above factors is a give-and-take activity that includes much iteration. The deliverable is an agreed-on **statement of work** that describes the work to be performed during the project. In consulting engagements, the statement of work has become a commonly used contract between the consultant and client. This approach works equally well for internal system development projects to establish a contract between business management and the project manager and team. According to Keane, Inc., a leading project management consulting firm,

statement of work a narrative description of the work to be performed as part of a project. Common synonyms include *scope statement, project definition, project overview,* and *document of understanding.*

> The statement of work affirms that the project manager understands who is really in charge of the effort, who is controlling the purse strings, what is the formal and informal organization within which the project will be developed, who are the "kings and queens" that have interest, and other similar but mainly nontechnical issues. It establishes a firm business relationship between the project manager and both the customer and the extended project team.[4]

An outline for a typical statement-of-work document is shown in Figure 4-5. The size of the document will vary in different organizations. It may be as small as one to two pages, or it may run several pages.

> Activity 2—Identify Tasks

Given the project scope, the next activity is to identify project tasks. Tasks identify the work to be done. Typically, this work is defined in a top-down, outline manner. In Chapter 3, you learned about system development routes and their phases. But phases are too large and complex for planning and scheduling a project. We need to break them down into activities and tasks until each task represents a manageable amount of work that can be planned, scheduled, and assigned. Some experts advocate decomposing tasks until the tasks represent an amount of work that can be completed in two weeks or less.

Ultimately, the project manager will determine the level of detail in the outline; however, most system development methodologies decompose phases for you—into suggested activities and tasks. These activities and tasks are not necessarily carved in stone; that is, most methodologies allow for some addition, deletion, and changing of activities and tasks based on the unique nature of each project. One popular tool used to identify and document project activities and tasks is a work breakdown structure. A **work breakdown structure (WBS)** is a hierarchical decomposition of the project into phases, activities, and tasks.

work breakdown structure (WBS) a graphical tool used to depict the hierarchical decomposition of a project into phases, activities, and tasks.

[4]Updated and revised by Donald H. Plumbet, *Productivity Management: Keane's Project Management Approach for Systems Development* (Boston: Keane, Inc., 1995), p. 5.

STATEMENT OF WORK

I. Purpose
II. Background
  A. Problem, opportunity, or directive statement
  B. History leading to project request
  C. Project goal and objectives
  D. Product description
III. Scope
(notice the use of the information system building blocks)
  A. Stakeholders
  B. Knowledge
  C. Processes
  D. Communications
IV. Project Approach
  A. Route
  B. Deliverables
V. Managerial Approach
  A. Team-building considerations
  B. Manager and experience
  C. Training requirements
  D. Meeting schedules
  E. Reporting methods and frequency
  F. Conflict management
  G. Scope management
VI. Constraints
  A. Start date
  B. Deadlines
  C. Budget
  D. Technology
VII. Ballpark Estimates
  A. Schedule
  B. Budget
VIII. Conditions of Satisfaction
  A. Success criteria
  B. Assumptions
  C. Risks
IX. Appendixes

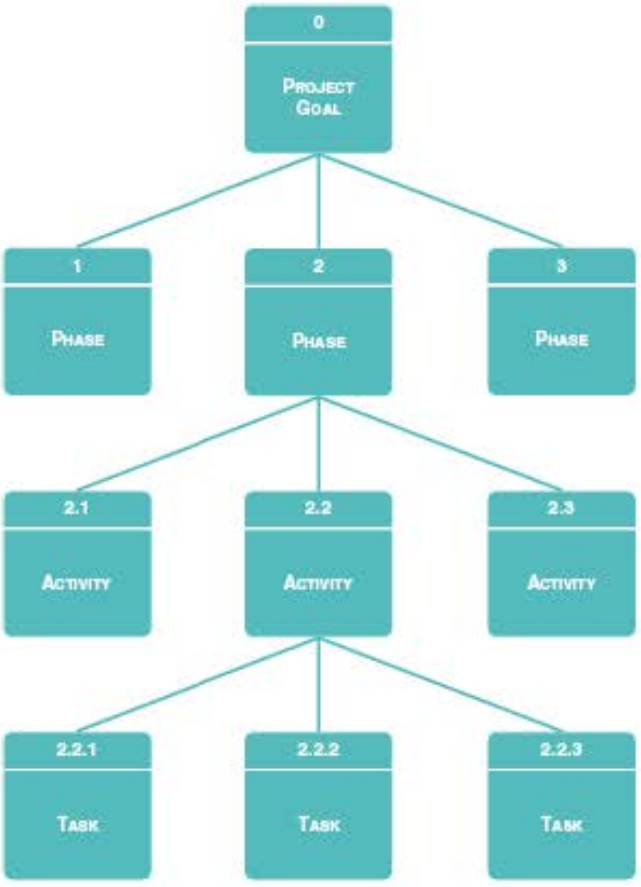
FIGURE 4-5
An Outline for a Statement of Work

Work breakdown structures can be drawn using top-down hierarchy charts similar to organization charts (Figure 4-6). In Microsoft *Project*, a WBS is depicted using a simple outline style, indentation of activities and tasks on the Gantt chart "view" of the project. Microsoft *Project* also offers a military numbering scheme to represent hierarchical decomposition of a project as follows:

1. Phase 1 of the project
  1.1 Activity 1 of Phase 1
    1.1.1 Task 1 of Activity 1 in Phase 1
    1.1.2 Task 2 of Activity 1 in Phase 1
  1.2 Activity 2 of Phase 1 . . .
2. Phase 2 of the project . . .

If you reexamine Figure 4-3(a), you will notice that Microsoft *Project* provides a column for the WBS in the Gantt chart. Also notice its use of the indentation and numbering to differentiate between tasks and subtasks.

FIGURE 4-6
A Graphical Work Breakdown Structure



milestone an event signifying the completion of a major project deliverable.

We may want to include in a WBS special tasks called **milestones.** These are events that signify the accomplishment or completion of major deliverables during a project. In information systems projects, an example of a milestone might be the completion of all the tasks associated with producing a major project deliverable such as a requirements statement (see Chapter 3). It might be useful to distinguish milestones from other tasks in a WBS by using special formatting, such as italics.

> Activity 3—Estimate Task Durations

Given a work breakdown structure with a suitable level of detail, the project manager must estimate duration for each task. Duration of any task is a random variable subject to factors such as the size of the team, number of users, availability of users, aptitudes of users, complexity of the business system, information technology architecture, experience of team personnel, time committed to other projects, and experience with other projects.

Most system development methodologies not only define tasks but also provide baseline estimates for task duration. The project manager must adjust these baselines into reasonable estimates for each unique project.

In Microsoft *Project,* all phases, activities, and tasks of a methodology are simply called *tasks.* The work breakdown structure then consists of both summary and primitive tasks. A *summary task* is one that consists of other tasks (such as phases and activities). A *primitive task* is one that does not consist of any other tasks. It is the primitive tasks for which the project manager must estimate duration. (Like most project management software, Microsoft *Project* will automatically calculate the duration of all summary tasks based on the estimated durations of their component primitive tasks.)

For those primitive tasks that are not milestones, we must estimate duration. In estimating task duration, it is important to understand the concept of *elapsed time.* Elapsed time takes into consideration two important factors with respect to people:

- *Efficiency*—No worker performs at 100 percent efficiency. Most people take coffee breaks, lunch breaks, restroom breaks, and time to read their e-mail, check their calendars, participate in nonproject work, and even engage in idle conversation. Experts differ on just how productive the average worker is, but one commonly used figure is 75 percent.
- *Interruptions*—People experience phone calls, visitors, and other unplanned interruptions that increase the time required for project work. This is variable for different workers. Interruptions can consume as little as 10 percent of a worker's day or as much as 50 percent.

Why is this important? Given a task that could be completed in 10 hours with 100 percent efficiency and no interruptions, and assuming a worker efficiency of 75 percent and 15 percent interruptions, the true estimate for the task would be

$$10 \text{ hours} \div 0.75 \text{ efficiency} = 13.3 \text{ hours} \div (1.00 - 0.15 \text{ interruptions})$$
$$= 15.7 \text{ hours}$$

There are many techniques for estimating task duration. For the sake of demonstration, we offer the following classic technique:

1. *Estimate the minimum amount of time it would take to perform the task.* We'll call this the **optimistic duration (OD).** The optimistic duration assumes that even the most likely interruptions or delays, such as occasional employee illnesses, will *not* happen.
2. *Estimate the maximum amount of time it would take to perform the task.* We'll call this the **pessimistic duration (PD).** The pessimistic duration assumes that nearly anything that can go wrong will go wrong. All possible interruptions or delays, such as labor strikes, illnesses, inaccurate specification of requirements, equipment delivery delays, and underestimation of the system's complexity, are assumed to be inevitable.
3. *Estimate the* **expected duration (ED)** *that will be needed to perform the task.* Don't just take the median of the optimistic and pessimistic durations. Attempt to identify interruptions or delays that are most likely to occur, such as occasional employee illnesses, inexperienced personnel, and occasional training.
4. *Calculate the* **most likely duration (D),** as follows:

$$D = \frac{(1 \times OD) + (4 \times ED) + (1 \times PD)}{6}$$

where 1, 4, and 1 are default weights used to calculate a weighted average of the three estimates.

optimistic duration (OD) the estimated minimum amount of time needed to complete a task.

pessimistic duration (PD) the estimated maximum amount of time needed to complete a task.

expected duration (ED) the estimated amount of time required to complete a task.

most likely duration (D) an estimated amount of time required to complete a task, based on a weighted average of optimistic, pessimistic, and expected durations.

Developing OD, PD, and ED estimates can be tricky and require experience. Several techniques are used in estimating. Three of the most common techniques are:

- *Decomposition*—a simple technique wherein a project is decomposed into small, manageable pieces that can be estimated based on historical data of past projects and similarly complex pieces.
- *COCOMO* (pronounced like "Kokomo")—a model-based technique wherein standard parameters based on prior projects are applied to the new project to estimate duration of a project and its tasks.
- *Function points*—a model-based technique wherein the "end product" of a project is measured based on number and complexity of inputs, outputs, files, and queries. The number of function points is then compared to projects that had a similar number of function points to estimate duration.

Some automated project management tools, such as *CS/10000* and *Cost•Xpert*, provide expert system technology that makes these estimates for you based on your answers to specific questions.

Milestones (as defined in the previous subsection) have no duration. They simply happen. In Microsoft *Project*, milestones are designated by setting the duration to *zero*. (In the Gantt chart, those zero-duration tasks change from bars to diamonds.)

> Activity 4—Specify Intertask Dependencies

Given the duration estimates for all tasks, we can now begin to develop a project schedule. The project schedule depends not only on task durations but also on intertask dependencies. In other words, the start or completion of individual tasks may depend on the start or completion of other tasks. There are four types of intertask dependencies:

- *Finish-to-start (FS)*—The finish of one task triggers the start of another task.
- *Start-to-start (SS)*—The start of one task triggers the start of another task.
- *Finish-to-finish (FF)*—Two tasks must finish at the same time.
- *Start to finish (SF)*—The start of one task signifies the finish of another task.

Intertask dependencies can be established and depicted in both Gantt and PERT charts. Figure 4-7 illustrates how to enter intertask dependencies in the Gantt chart view in Microsoft *Project*. We call your attention to the following annotated bullets:

❶ Intertask dependencies may be entered in the Gantt chart view in the *Predecessors* column by entering the dependent tasks' row numbers. Note that a task can have zero, one, or many predecessors.

❷ Intertask dependencies may also be entered (or modified) by opening the *Task Information* dialogue box for a given task.

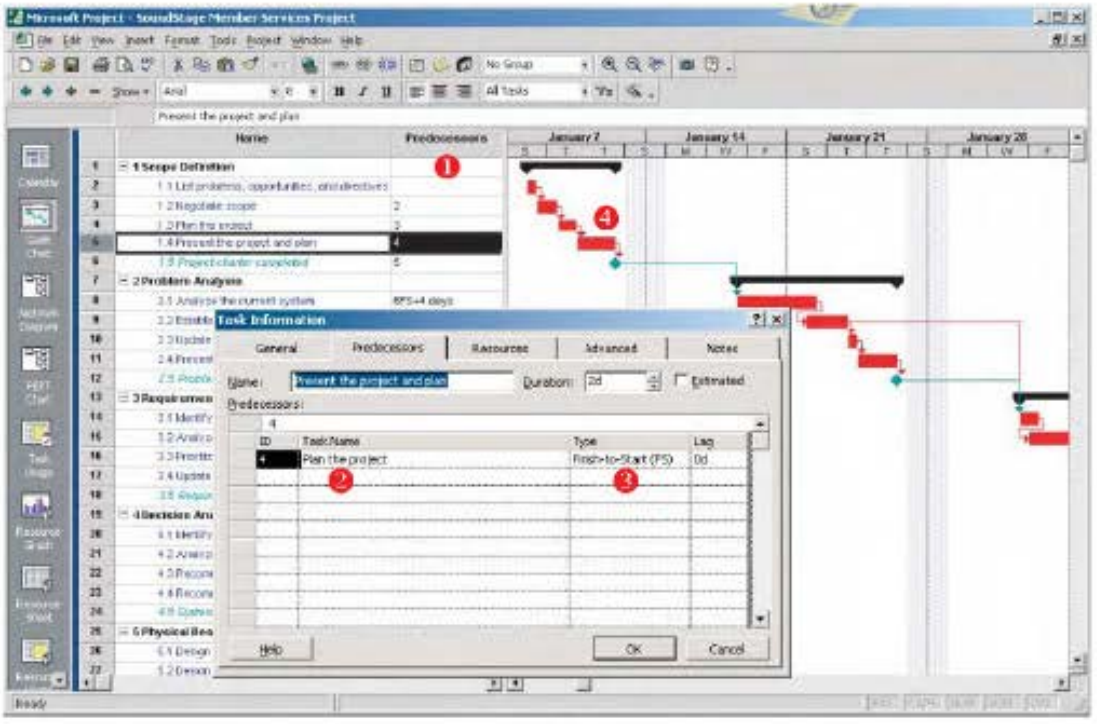


FIGURE 4-7 Entering Intertask Dependencies in Microsoft *Project*

3. The type of dependency can be entered in the *Task Information* dialogue box for any given dependent task.
4. Intertask dependencies are graphically illustrated in the Gantt chart as arrows between the bars that represent each task. Arrows may begin or terminate on the left side (to indicate a "start" dependency) or right side (to indicate a "finish" dependency).

Milestones (as defined earlier) almost always have several predecessors to signify those tasks that must be completed before the milestone has been achieved.

Given the start date for a project, the tasks to be completed, the task durations, and the intertask dependencies, the project can now be scheduled. There are two approaches to scheduling:

- **Forward scheduling** establishes a project start date and then schedules forward from that date. Based on the planned duration of required tasks, their interdependencies, and the allocation of resources to complete those tasks, a projected project completion date is calculated.
- **Reverse scheduling** establishes a project deadline and then schedules backward from that date. Tasks, their duration, interdependencies, and resources must be considered to ensure that the project can be completed by the deadline.

forward scheduling a project scheduling approach that establishes a project start date and then schedules forward from that date.

reverse scheduling a project scheduling strategy that establishes a project deadline and then schedules backward from that date.

Each task can be given its own start and finish dates. Like most project management tools, Microsoft *Project* actually builds the schedule for you as you enter the task durations and intertask dependencies (predecessors). On the Gantt chart, the task bars are expanded to reflect duration and shifted left and right to reflect start and end dates. Microsoft *Project* can also produce a traditional calendar view of the final schedule, as shown in Figure 4-8.

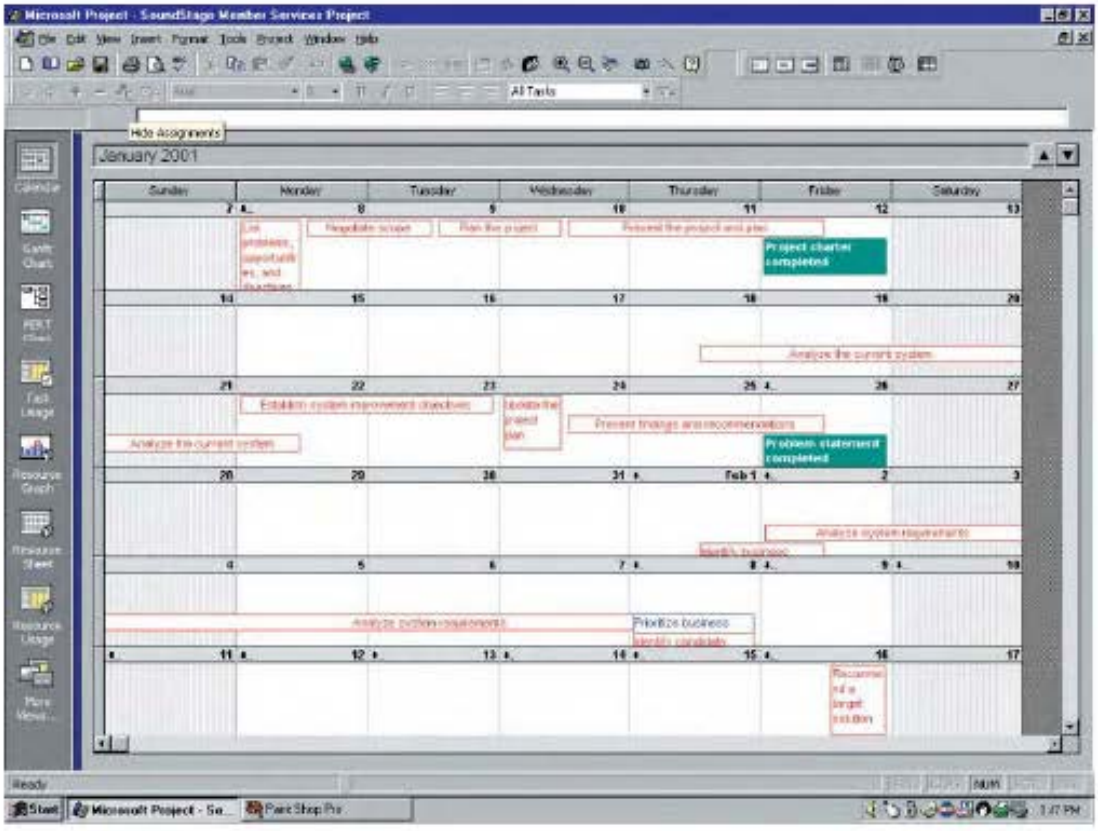


FIGURE 4-8 The Project Schedule in Calendar View

REPRESENTATIVE ROLES IN A PROJECT

Auditor
Business Analyst
Business Subject Matter Expert
Database Administrator
Executive Sponsor
Information Systems Manager
JAD Facilitator
JAD Scribe
Management Sponsor
Network Administrator
Programmer
Project Manager
System Modeler

> Activity 5—Assign Resources

The previous steps resulted in "a" schedule, but not "the" schedule! We have yet to consider the allocation of resources to the project. Resources include the following categories:

- *People*—includes all the system owners, users, analysts, designers, builders, external agents, and clerical help that will be involved in the project in any way.
- *Services*—includes services such as a quality review that may be charged on a per-use basis.
- *Facilities and equipment*—includes all rooms and technology that will be needed to complete the project.
- *Supplies and materials*—includes everything from pencils, paper, and notebooks to toner cartridges, and so on.
- *Money*—includes a translation of all of the above into budgeted dollars!

The availability of resources, especially people and facilities, can significantly alter the project schedule.

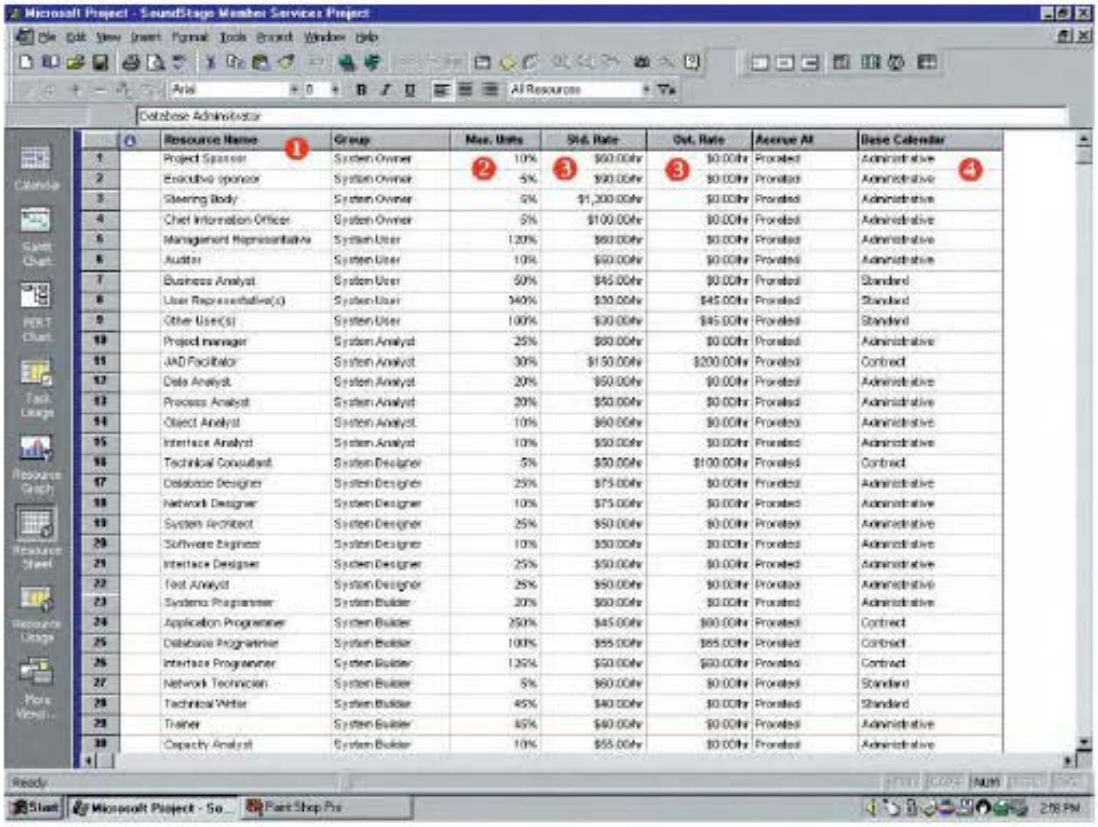
Most system development methodologies identify *people* resources required for each task in the form of roles. A role is not the same as a job title. Think of a role as a "hat" that someone wears because he or she possesses a certain skill(s). Any given individual may be capable of wearing many hats (thus playing many roles). Also, many people may possess the skills required to play a given role. The project manager's task is either to assign specific people to fill roles or to gain commitments from management to provide people to fill roles. Representative roles from the *FAST* methodology are listed in the margin.

In Microsoft *Project,* roles and assignments are specified in the *Resource Sheet* view, as shown in Figure 4.9(a). Predefined roles and resources may be available in the chosen methodology and route templates.

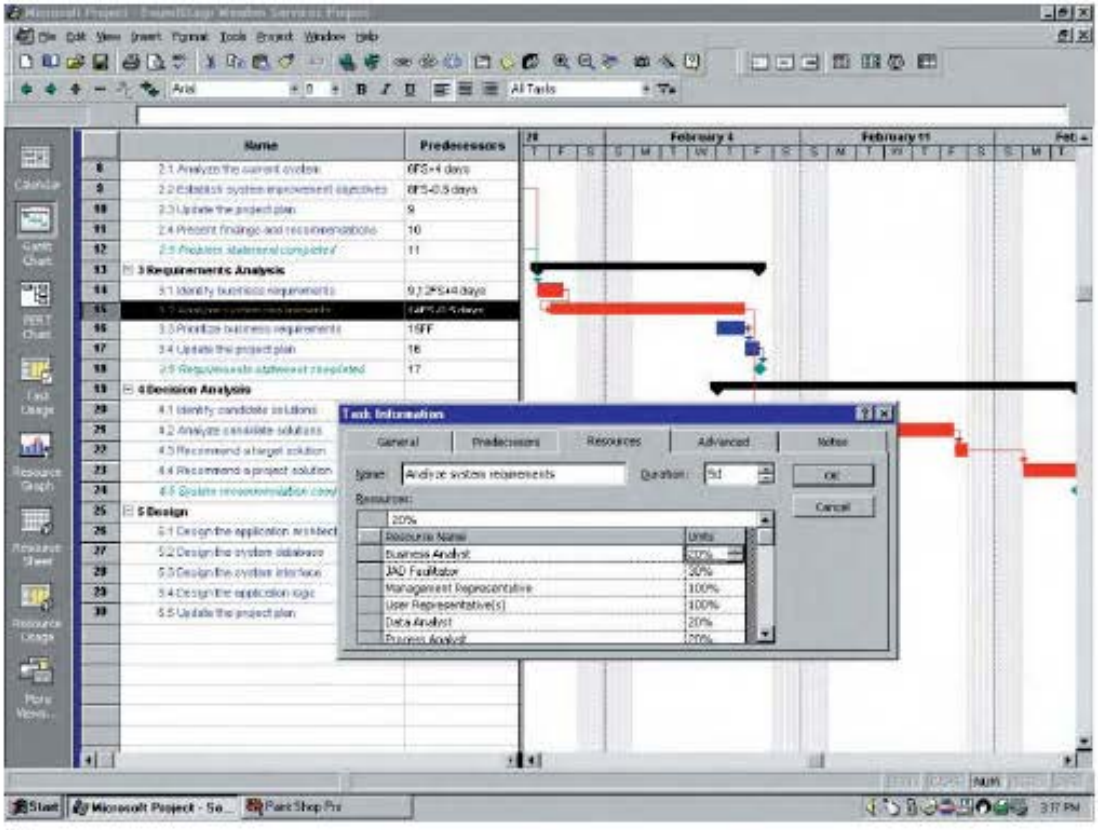
1. The project manager enters the names or titles of people (roles) in the *Resource Name* column. Resources may also include specific services, facilities, equipment, supplies, materials, and so forth.
2. Notice that the Resource Sheet provides a column for establishing what percentage of a resource will be allocated to the project. For example, a database administrator might be allocated one-quarter time (25 percent) to a project. Allocations greater than 100 percent indicate a need for more than one person to fill a given role in the project. By setting *Max. Units* to 250 percent for that resource, there would be a need for the equivalent of 2½ full-time programmers.
3. *Project* also allows the project manager to estimate the cost of each resource. These costs can be estimated based on company history, consulting contracts, or internal cost accounting standards. Notice that both standard and overtime costs can be estimated. These costs are usually based on standards to protect information about anyone's actual salary.
4. Each resource has a *calendar* that considers the standard workweek and holidays, as well as individual vacations and other commitments.

Given the resources, they now can be specifically assigned to tasks, as shown in Figure 4.9(b). As resources are assigned to the tasks, the project manager would specify the units of that resource that will be required to complete each assigned task. (This may be a *percentage* of a person's time needed for that task.)

As these resources are formally assigned, the schedule will be adjusted (which happens automatically in tools such as *Project*). If you enter the cost of resources, tools such as Microsoft *Project* will automatically calculate and maintain a budget based on the resources and schedule.



(a)



(b)

FIGURE 4-9 Defining and Assigning Project Resources

Assigning People to Tasks Recruiting the right team members can make or break a project. The following are guidelines for selecting and recruiting the team:

- *Recruit talented, highly motivated people.* Highly skilled and motivated team members are more likely to overcome project obstacles unaided and are more likely to meet project deadlines and produce quality work.
- *Select the best task for each person.* All workers have strengths and weaknesses. Effective project managers learn to exploit the strengths of team members and avoid assigning tasks to team members not skilled in those areas.
- *Promote team harmony.* Project managers should select team members who will work well together.
- *Plan for the future.* Junior personnel with potential to be mentored by project leaders must be considered. Although junior personnel might not be as productive as the seasoned veterans, project managers will need them and have to rely on them on future projects.
- *Keep the team size small.* By limiting the team size, communication overhead and difficulties will be reduced. A 2-person team has only 1 communication path; a 4-person team has 6 communication paths; and a 50-person team has at least 1,200 communication paths. The more communication paths there are, the greater the probability that there will be increased communication problems. By the same token the teams should be large enough to provide adequate backup and coverage in key skills if a team member is lost.

Resource Leveling So far, we have identified tasks, task durations, and intertask dependencies and assigned resources to each task to produce the project schedule. It is common to overallocate resources when assigning resources to tasks. *Overallocate* refers to the act of assigning more resources than are available.

For example, during a specific period in the project (day, week, etc.), a project manager may have assigned a specific person to work on multiple tasks that add up to more hours than the person has available to work during that period. This renders the overall schedule infeasible because the overallocated resource cannot reasonably complete all assigned tasks according to schedule. To correct this problem, project managers must use a technique called resource leveling. **Resource leveling** is a strategy used to correct resource overallocations by some combination of delaying or splitting tasks. Let's briefly explain both approaches.

resource leveling a strategy for correcting resource overallocations.

Delaying tasks is based on the concepts of *critical path* and *slack time*. When it comes to the project schedule, some tasks are more sensitive to schedule delays than others. For this reason, project managers must become aware of the critical path for a project. The **critical path** for a project is the sequence of dependent tasks that have the largest sum of *most likely durations*. The critical path determines the earliest possible completion date of the project. (We previously described how to estimate *most likely duration* for a task.) The critical path tasks have no slack time available—thus, any delay in completion of any of the tasks on the critical path will cause an overall delay in the completion of the entire project. The opposite of a critical task is one that has some slack time. The **slack time** available for any noncritical task is the amount of delay that can be tolerated between the starting time and the completion time of a task without causing a delay in the completion date of the entire project. Tasks that have slack time can get behind schedule by an amount less than or equal to the slack time without having any impact on the project's final completion date. The availability of slack time in certain tasks provides an opportunity to delay the start of the tasks to level resources while not affecting the project completion date. Of course, it may be necessary to delay a critical path task to level resources, unless you can split the task.

critical path the sequence of dependent tasks that determines the earliest completion date for a project.

slack time the amount of delay that can be tolerated between the starting time and the completion time of a task without causing a delay in the completion date of a project.

Splitting tasks involves breaking a task into multiple tasks to assign alternate resources to the tasks. Thus, a single task for which a resource was overallocated is now apportioned to two or more resources that are (presumably) not overallocated. Splitting tasks requires identifying and assigning new resources such as analysts, contractors, or consultants.

Resource leveling can be tedious to perform manually. For each resource, the project manager needs to know the total time available to the project for the resource, all task assignments made to the resource, and the sum of all durations of those task assignments over various time periods. All project management software tools, such as Microsoft *Project,* automatically determine critical *paths* and slack times. This enables those same software tools to track resource allocations and automatically perform resource leveling. It is extraordinarily rare for any modern project manager to manually level resource assignments.

Resource leveling will be an ongoing activity because the schedule and resource assignments are likely to change over the course of a project.

Schedule and Budget Given a schedule based on leveled resources and given the cost of each resource (e.g., cost per hour of a systems analyst or database administrator) the project manager can produce a printed (or Web-based) document that communicates the project plan to all concerned parties. Project management tools will provide multiple views of a project such as calendars, Gantt chart, PERT chart, resource and resource leveling reports, and budget reports. All that remains is to direct resources to the completion of project tasks and deliverables.

Communication The statement of work, timetable for major deliverables, and overall project schedule should be communicated to all parties involved in the project. This communication must also include a plan for reporting progress, both orally and in writing, the frequency of such communications, and a contact person and method for parties to submit feedback and suggestions. A corporate intranet can be an effective way to keep everyone informed of project progress and issues.

> Activity 6—Direct the Team Effort

All the preceding project management activities led to a master plan for the project. It's now time to execute that plan. There are several dimensions to directing the team effort. Tom Demarco states in his book *The Deadline: A Novel about Project Management* that the hardest job in management is people.

Few new project managers are skilled at supervising people. Most learn supervision through their own experiences as subordinates—things they liked and disliked about those who supervised them. This topic could easily take up an entire chapter. In the margin checklist, we provide a classic list of project supervision recommendations from *The People Side of Systems,* by Keith London.

As noted by Graham McLeod and Derek Smith, "Individuals brought together in a systems development team do not form a close-knit unit immediately." McLeod and Smith explain that teams go through stages of team development, as shown in Figure 4-10.

In *The One Minute Manager,* by Kenneth Blanchard and Spencer Johnson, a classic and indispensable aid to anyone managing people for the first time, the authors share the simple secrets of managing people and achieving success through the actions of subordinates.

Most young, and many experienced, managers have difficulty with the subtle arts of delegation and accountability. Worse still, they let subordinates reverse-delegate tasks back to the manager. This leads to poor time management and manager frustration. In *The One Minute Manager Meets the Monkey,* Kenneth Blanchard teams with William Oncken and Hal Burrows to help managers overcome this problem. The solution is based on Oncken's classic principle of "the care and feeding of monkeys." Monkeys are "problems" that managers delegate to their subordinates, who in turn attempt to reverse-delegate back to the manager. In this 125-page book the authors teach managers how to keep the monkeys on the subordinates' backs. Doing so increases the manager's available work time, accelerates task accomplishment by subordinates, and teaches subordinates how to take responsibility and solve their own problems.

10 HINTS FOR PROJECT LEADERSHIP

- Be Consistent.
- Provide Support.
- Don't Make Promises You Can't Keep.
- Praise in Public; Criticize in Private.
- Be Aware of Morale Danger Points.
- Set Realistic Deadlines.
- Set Perceivable Targets.
- Explain and Show, Rather Than Do.
- Don't Rely Just on Status Reports.
- Encourage a Good Team Spirit.

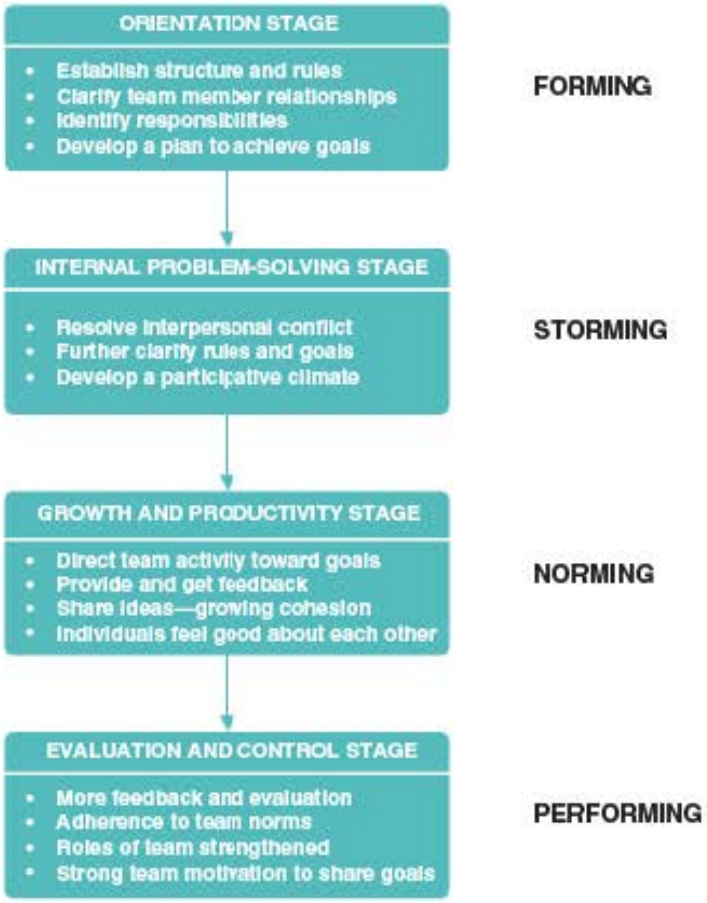


FIGURE 4-10

Stages of Team Maturity

Source: Adapted from Graham McLeod and Derek Smith, *Managing Information Technology Projects* (Cambridge, MA: Course Technology, 1996).

> Activity 7—Monitor and Control Progress

While executing the project, the project manager must control the project, that is, monitor its progress against the scope, schedule, and budget. The manager must report progress and, when necessary, adjust scope, schedule, and resources.

Progress Reporting Progress reporting should be frequent enough to establish accountability and control, but not so frequent as to become a burden and impediment to real project progress. For example, Keane, Inc., a consulting firm, recommends that progress reports or meetings occur every two weeks—consistent with the firm's project-planning strategy that decomposes projects into tasks that produce deliverables that require no more than 80 work hours.

Project progress reports can be verbal or written. Figure 4-11 illustrates a template for a written progress report. Project progress reports (or presentations) should be honest and accurate, even if the news is less than good. Project progress reports should report successes but should clearly identify problems and concerns such that they can be addressed before they escalate unto major issues or catastrophes.

As tasks are completed, progress can be recorded in Microsoft *Project* (see Figure 4-12). We call your attention to the following Gantt progress items:

1. All the tasks in the preliminary investigation phase are complete as indicated by the yellow lines that run the full length of each task bar. Notice that because all these tasks are complete, they are no longer critical—the bars have changed from red to blue.
2. In the problem analysis phase, only the first task, "Analyze the current system," is 100 percent complete.

PROJECT PROGRESS REPORT

I. Cover page
   A. Project name or identification
   B. Project manager
   C. Date of report
II. Summary of progress
   A. Schedule analysis
   B. Budget analysis
   C. Scope analysis
      (describe any changes that may have an impact on future progress)
   D. Process analysis
      (describe any problems encountered with strategy or methodology)
   E. Gantt progress chart(s)
III. Activity analysis
   A. Tasks completed since last report
   B. Current tasks and deliverables
   C. Short-term future tasks and deliverables
IV. Previous problems and issues
   A. Action item and status
   B. New or revised action items
      1. Recommendation
      2. Assignment of responsibility
      3. Deadline
V. New problems and issues
   A. Problems
      (actual or anticipated)
   B. Issues
      (actual or anticipated)
   C. Possible solutions
      1. Recommendation
      2. Assignment of responsibility
      3. Deadline
VI. Attachments
   (include relevant printouts from project management software)

FIGURE 4-11
Outline for a Progress Report

3. Notice that the "Establish system improvement objectives" task bar has a partial yellow line running 60 percent of its length. This indicates the task is about 60 percent complete. The task bar is still red because any delay in completing the task will threaten the project completion date.
4. All remaining tasks shown in the displayed chart have not been started. Actual progress will be recorded when the task is started, in process, or completed.
5. Progress for any given task is recorded in the task information dialogue box for that task. In this example, the project manager is recording 10 percent completion of the named task.

Microsoft *Project* also provides a number of preconfigured and customizable reports that can present useful project status information.

Change Management It is not uncommon for scope to grow out of control even when a properly completed statement of work was agreed on early in the planning process. We refer to scope growth as "change." As noted by Keane, Inc., "Change is frequently a point of contention between the customer and the information systems organization, because they disagree on whether a particular function is a change or a part of the initial agreement." The inevitability of scope change necessitates that we have a formal strategy and process to deal with change and its impact on schedule

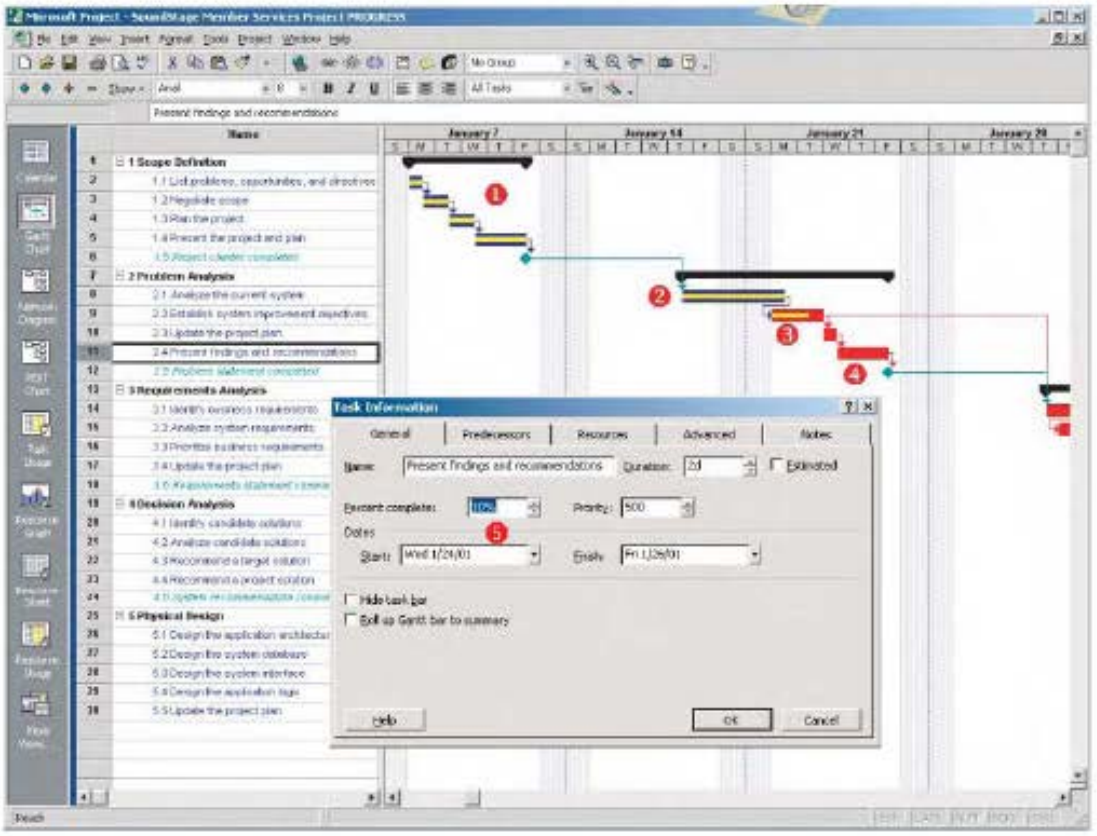


FIGURE 4-12 Progress Reporting on a Gantt Chart

change management a formal strategy wherein a process is established to facilitate changes that occur during a project.

and budget. **Change management** is intended to protect the project manager and team from being held accountable for schedule and budget overruns that were driven by changes in scope.

Changes can be the result of various events and factors, including:

- An omission in defining initial scope (as documented in the statement of work).
- A misunderstanding of the initial scope (the desired product is more complicated than originally communicated or perceived).
- An external event such as government regulations that create new requirements.
- Organizational changes, such as mergers, acquisitions, and partnerships, that create new business problems and opportunities (not to mention "players").
- Availability of better technology.
- Shifts in planned technology that force unexpected and significant changes to the business organization, culture, and/or processes.
- Management's desire to have the system do more than was originally requested or agreed to.
- Reduced funding for the project or imposition of an earlier deadline.

A change management system results in a collection of procedures for documenting a change request and defines the steps necessary to consider the change based on the expected impact of the change. Most change management systems require that a change request form be initiated by one or more project stakeholders (e.g., system owners, users, analysts, designers, or builders). Ideally, change requests are considered by a *change control board (CCB)*, which is responsible for approving or rejecting all change requests. The CCB's composition typically includes members

of the project team as well as outsiders who may have an interest or stake in the project. The CCB's decision should be based on impact analysis.

Feasibility impact analysis should assess the importance of the change to the business, the impact of the change on the project schedule, and the impact of the change on the project budget and long-term operating costs.

Ultimately, change management boils down to managing the expectations of the stakeholders. In the next section, we introduce a simple but conceptually sound framework for managing expectations and their impact on project schedule and budget.

Expectations Management Experienced project managers often complain that managing system owners' and users' expectations of a project is more difficult than managing cost, schedule, people, or quality. In this section we introduce a simple tool that we'll call an *expectations management matrix* that can help project managers deal with this problem. We first learned about this tool from Dr. Phil Friedlander, a consultant and trainer then with McDonnell Douglas. He attributes the matrix to "folklore" but also credits Jerry Gordon, of Majer, and Ron Leflour, a project management educator/trainer. Dr. Friedlander's paper (listed in the Suggested Readings for this chapter) is adapted in this text for this presentation.

Every project has goals and constraints when it comes to cost, schedule, scope, and quality. In an ideal world, each of these parameters could be optimized. Management often has that expectation. Reality suggests, however, that you can't optimize them all—you must strike a balance that is both feasible and acceptable to management. That is the purpose of the expectations management matrix. An **expectations management matrix** is a rule-driven tool for helping management understand the dynamics and impact of changing project parameters such as cost, schedule, scope, and quality.

expectations management matrix a tool used to understand the dynamics and impact of changing the parameters of a project.

The basic matrix, shown in Figure 4-13, consists of three rows and three columns (plus headings). The rows correspond to the measures of success in any project: cost, schedule, and scope and/or quality. The columns correspond to priorities: first, second, and third. To establish expectations, we assign names to the priorities as follows:

- *Maximize or minimize*—the measure of success that is determined to be the most important for a given project.
- *Constrain*—the second most important of the three measures of success in a project.
- *Accept*—the least important of the three measures in a project.

Most managers would, ideally, like to give equal priority to all three measures; experience suggests that the three measures tend to balance themselves naturally. For example, if you increase scope or quality requirements, the project will take more time and/or money. If you try to get any job done faster, you generally have to reduce scope or

PRIORITIES → / ↓MEASURES OF SUCCESS	Max or Min	Constrain	Accept
Cost			
Schedule			
Scope and/or Quality			

FIGURE 4-13
A Management Expectations Matrix

FIGURE 4-14
Management of Expectations for the Lunar Landing Project

PRIORITIES → / ↓MEASURES OF SUCCESS	Max or Min	Constrain	Accept
Cost • $20 billion (estimated)			X
Schedule • Dec 31, 1969 (deadline)		X	
Scope and/or Quality • Land a man on the moon • Get him back safely	X		

quality requirements or pay more money to compensate. The management expectations matrix helps (or forces) management to understand this through three simple rules:

1. For any project, you must record three Xs within the nine available cells.
2. No row may contain more than one X. In other words, a single measure of success must have one and only one priority.
3. No column may contain more than one X. In other words, there must be a first, second, and third priority.

Let's illustrate the tool using Dr. Friedlander's own example. In 1961 President John F. Kennedy established a major project—land a man on the moon and return him safely before the end of the decade. Figure 4-14 shows the realistic expectations of the project. Let's walk through the example:

1. The system owner (the public) had both scope and quality expectations. The scope (or requirement) was to successfully land a man on the moon. The quality measure was to return the man (or men) safely. Because the public would expect no less from the new space program, this had to be made the first priority. In other words, we had to maximize safety and minimize risk as a first priority. Hence, we record the X in column 1, row 3.
2. At the time of the project's inception, the Soviet Union was ahead in the race to space. This was a matter of national pride; therefore, the second priority was to get the job done by the end of the decade. We call this the project constraint—there is no need to rush the deadline, but we don't want to miss the deadline. Thus, we record the second X in column 2, row 2.
3. By default, the third priority had to be cost (estimated at $20 billion in 1961). By making cost the third priority, we are not stating that cost will not be controlled. We are merely stating that we may have to accept cost overruns to achieve the scope and quality requirement by the constrained deadline. We record the third X in column 3, row 1.

History records that we achieved the scope and quality requirement, and did so in 1969. The project actually cost well in excess of $30 billion, more than a 50 percent cost overrun. Did that make the project a failure? On the contrary, most people perceived the project as a grand success. The government managed the public's expectations of the project in realizing that maximum safety and minimum risk, plus meeting the deadline (beating the Soviets), was an acceptable trade-off for the cost overrun. The government brilliantly managed public opinion. Systems development project managers can learn a valuable lesson from this balancing act.

At the beginning of any project, the project manager should consider introducing the system owner to the expectations matrix concept and should work with the system owner to complete the matrix. For most projects, it would be difficult to record all the scope and quality requirements in the matrix. Instead, they would be listed in the statement of work. The estimated costs and deadlines could be recorded directly in the matrix.

The project manager doesn't establish the priorities; he or she merely enforces the rules of the matrix. This sounds easy, but it rarely is. Many managers are unwilling to be pinned down on the priorities—"Shouldn't we be able to maximize everything?" These managers need to be educated about the reason for the priorities. They need to understand the priorities if they cannot maximize all three measures. This leads to intelligent compromises instead of merely guesswork.

What if a system owner refuses to prioritize? The tool becomes less useful, except as a mechanism for documenting concerns before they become disasters. A system owner who refuses to set priorities may be setting the project manager up for a no-win performance review. And as Dr. Friedlander points out, "Those who do not 'believe' the principles [of the matrix] will eventually 'know' the truth. You do not have to believe in gravity, but you will hit the ground just as hard as the person who does."

Let's assume the management expectations matrix that conforms to the aforementioned rules. How does this help a project manager manage expectations? During the course of the average systems development project, priorities are not stable. Various factors such as the economy, government, and company politics can change the priorities. Budgets may become more or less constrained. Deadlines may become more or less important. Quality may become more important. And, most frequently, requirements increase. As already noted, these changing factors affect all the measures in some way. The trick is to manage expectations despite the ever-changing project parameters.

The technique is relatively straightforward. Whenever the "max/min measure" or the "constrain measure" begins to slip, it can result in a potential management expectations problem. For example, consider a project manager who is faced with the following priorities (see Figure 4-15):

1. Explicit requirements and quality expectations were established at the start of a project and given the highest priority.
2. An absolute maximum budget was established for the project.
3. The project manager agreed to shoot for the desired deadline, but the system owner(s) accepted the reality that if something must slip, it should be schedule.

Now suppose that during systems analysis, significant and unanticipated business problems are identified. The analysis of these problems has placed the project behind schedule. Furthermore, solving the new business problems substantially expands the user

PRIORITIES → / ↓MEASURES OF SUCCESS	Max or Min	Constrain	Accept
Cost		X	
Schedule			X
Scope and/or Quality	X		

FIGURE 4-15
A Typical Initial Expectations Matrix

FIGURE 4-16
Adjusting Expectations (a sample)

PRIORITIES → / ↓MEASURES OF SUCCESS	Max or Min	Constrain	Accept
Cost • Adjusted budget		X+ Increase budget	
Schedule • Adjusted deadline			X- Extend deadline
Scope and/or Quality • Adjusted scope	X+ Accept expanded requirements		

requirements for the new system. How does the project manager react? There should be no overreaction to the schedule slippage—schedule slippage was the "accept" priority in the matrix. The scope increase (in the form of several new requirements) is the more significant problem because the added requirements will increase the cost of the project. Cost is the constrained measure of success. As it stands, an expectations problem exists. The project manager needs to review the matrix with the system owner.

First, the system owner needs to be made aware of which measure or measures are in jeopardy and why. Then together, the project manager and system owner can discuss courses of action. Several courses of action are possible:

- The resources (cost and/or schedule) can be reallocated. Perhaps the system owner can find more money somewhere. All priorities would remain the same (noting, of course, the revised deadline based on schedule slippages already encountered during systems analysis).
- The budget might be increased, but it would be offset by additional planned schedule slippages. For instance, by extending the project into a new fiscal year, additional money might be allocated without taking any money from existing projects or uses. This solution is shown in Figure 4-16.
- The user requirements (or quality) might be reduced through prioritizing those requirements and deferring some of those requirements until version 2 of the system. This alternative would be appropriate if the budget cannot be increased.
- Finally, measurement priorities can be changed.

Only the system owner may initiate priority changes. For example, the system owner may agree that the expanded requirements are worth the additional cost. He or she may allocate sufficient funds to cover the requirements but may migrate priorities such that minimizing cost now becomes the highest priority (see Figure 4-17, step 1). But now the matrix violates a rule—there are two Xs in column 1. To compensate, we must migrate the scope and/or quality criterion to another column, in this case, the constrain column (see Figure 4-17, step 2). Expectations have been adjusted. In effect, the system owner is freezing growth of requirements and still accepting schedule slippage.

There are three final comments about priority changes. First, priorities may change more than once during a project. Expectations can be managed through any number of changes as long as the matrix is balanced (meaning it conforms to our

PRIORITIES → / ↓MEASURES OF SUCCESS	Max or Min	Constrain	Accept
Cost	X ← Step 1	→ X	
Schedule			X
Scope and/or Quality	X → Step 2	→ X	

FIGURE 4-17
Changing Priorities

rules). Second, expectation management can be achieved through any combination of priority changes and resource adjustments. Finally, system owners can initiate priority changes even if the project is on schedule. For example, government regulation might force an uncompromising deadline on an existing project. That would suddenly migrate our "accept" schedule slippages to "max constraint." The other Xs would have to be migrated to rebalance the matrix.

While the management expectations matrix is a simple tool, it may be one of the most effective.

Schedule Adjustments—Critical Path Analysis When it comes to the project schedule, some tasks are more sensitive to schedule delays than others. For this reason, project managers must become aware of the critical path and slack times for a project.

Understanding the critical path and slack time in a project is indispensable to the project manager. Knowledge of such project factors influences the people management decisions to be made by the project manager. Emphasis can and should be placed on the critical path tasks, and if necessary, resources might be temporarily diverted from tasks with slack time to help get critical tasks back on schedule.

The critical path and slack time for a project can be depicted on both Gantt and PERT charts; however, PERT charts are generally preferred because they more clearly depict intertask dependencies that define the critical path. Most project management software, including Microsoft *Project,* automatically calculates and highlights the critical path based on intertask dependencies combined with durations. It is useful, however, to understand how the critical path and slack times are calculated.

Consider the following hypothetical example. A project consists of the nine primitive tasks shown in Figure 4-18. The most likely duration (in days) for each task is recorded. There are four distinct sequences of tasks in a project. They are:

Path 1: $A \rightarrow B \rightarrow C \rightarrow D \rightarrow I$
Path 2: $A \rightarrow B \rightarrow C \rightarrow E \rightarrow I$
Path 3: $A \rightarrow B \rightarrow C \rightarrow F \rightarrow G \rightarrow I$
Path 4: $A \rightarrow B \rightarrow C \rightarrow F \rightarrow H \rightarrow I$

The total of most likely duration times for each path is calculated as follows:

Path 1: $3 + 2 + 2 + 7 + 5 = 19$
Path 2: $3 + 2 + 2 + 6 + 5 = 18$
Path 3: $3 + 2 + 2 + 3 + 2 + 5 = 17$
Path 4: $3 + 2 + 2 + 3 + 1 + 5 = 16$

In this example, path 1 is the *critical path* at 19 days. (Note: You can have multiple critical paths if they have the same total duration.)

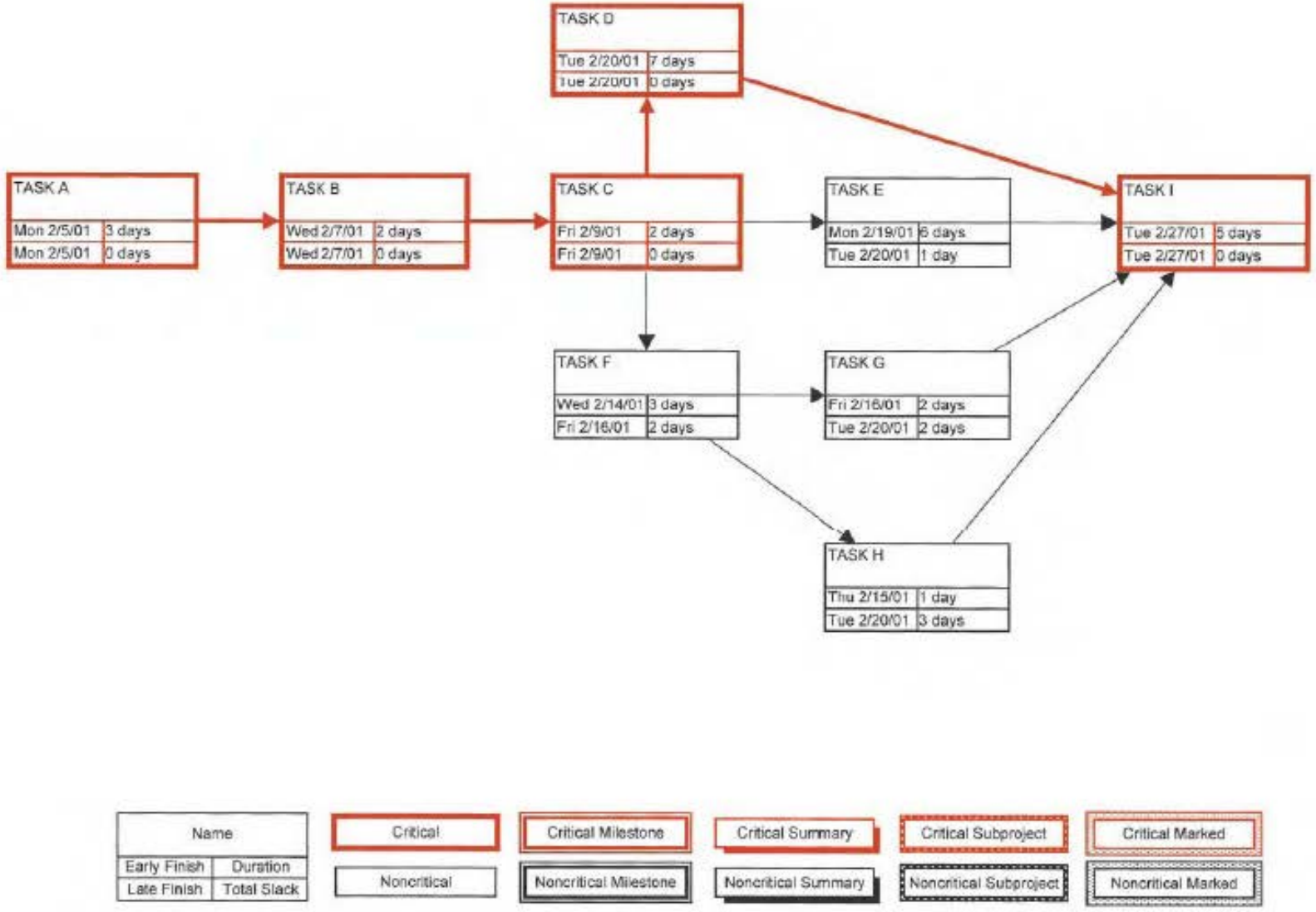


FIGURE 4-18 Critical Path Analysis

In this example, tasks E, F, and G are not on the critical path; they each have some slack time. For example, task E is included in a path that has one day less duration than the critical path; therefore, task E can get behind by as much as one day without adversely affecting the project completion date. Similarly, tasks F and G can *combine* for a maximum slack of two days without delaying the entire schedule.

In Figure 4-18, the critical path is shown in red. The tasks that have slack capacity are shown in black. Similarly, project management software also uses color to differentiate critical path tasks in a Gantt or PERT chart.

> Activity 8—Assess Project Results and Experiences

Project managers must learn from their mistakes! They should embrace continuous process improvement. This final activity involves soliciting feedback from project team members (including customers) concerning their project experiences and suggestions aimed at improving the project and process management of the organization. Project review(s) should be conducted to answer the following fundamental questions:

- Did the final product meet or exceed user expectations?
- Did the project come in on schedule?
- Did the project come in under budget?

The answers to these questions should be followed up with the basic question "Why or why not?" Subsequently, and based on the responses to the above questions, changes should be made to improve the system development and project management methods that will be used on future projects. Suggestions for improvements are communicated to "Centers for Excellence," which can modify standards and processes, as well as share useful ideas and experiences with other project teams that may solicit their help or expertise. Project assessments often contribute improvements to specific project deliverables (milestones), processes or tasks that created the deliverables, and the overall management of the project.

Learning Roadmap

Where you go from here depends on where you are coming from and where you want to go. If you are reading through the chapters sequentially, you should probably move on to Chapter 5, "Systems Analysis," to expand your understanding of systems analysis tasks, tools, and techniques. Alternatively, if you are enrolled in a system design–focused course, you might skip ahead to either Chapter 11, "Feasibility Analysis and the System Proposal" (which marks the end of systems analysis), or Chapter 12, "Systems Design" (which provides an in-depth look at the activities of system design, prototyping, and rapid application development).

Some instructors have deferred this project management chapter to the end of your course. If so, you may be interested in expanding your knowledge of project management tools, techniques, and methods. Some schools offer a project management course. If not, you may find that your systems analysis and design instructor might supervise you to complete an independent study course on the subject. If so, we direct you to two specific references at the end of this chapter as possible texts: (1) the Wysocki et al. book is well organized around the Project Management Body of Knowledge that we presented in our chapter, and (2) the McLeod and Smith book is especially comprehensive in its coverage of project management dimensions that we could not cover fully in our chapter.

Summary

1. A project is a (temporary) sequence of unique, complex, and connected activities that have one goal or purpose and that must be completed by a specific time, within budget, and according to specification.
2. Project management is the process of scoping, planning, staffing, organizing, directing, and controlling the development of an acceptable system at a minimum cost within a specified time frame.
3. Process management is an ongoing activity that documents, manages the use of, and improves an organization's chosen methodology (the "process") for systems development.
4. From a project management perspective, a project is considered a success if the resulting information system is acceptable to the customer, the system is delivered "on time" and "within budget," and the system development process had a minimal impact on ongoing business operations.
5. The Project Management Institute has created the Project Management Body of Knowledge (PMBOK) for the education and certification of professional project managers. It addresses:
   a. Project manager competencies.
   b. Project management functions.
   c. Tools and techniques such as:
      i) PERT charts, graphical network models that depict a project's tasks, and the relationships between those tasks.
      ii) Gantt charts, simple horizontal bar charts that depict project tasks against a calendar.
   d. Project management software.
6. Project management is a cross life-cycle activity; that is, project management tasks overlap all the system development phases. A project management process is essential to achieving CMM Level 2 maturity.
7. Joint project planning (JPP) is a strategy wherein all stakeholders in a project participate in a one- to three-day project management workshop, the result of which is consensus agreement on project scope, schedule, resources, and budget.
8. The tasks of project management include:
   a. Negotiate scope. Scope defines the boundaries of a project and is included in the statement of work, a narrative description of the work to be performed as part of a project.
   b. Identify tasks. A work breakdown structure (WBS) is a hierarchical decomposition of the project into its tasks and subtasks. Some tasks represent the completion of milestones or the completion of major deliverables during a project.
   c. Estimate task durations. There are many techniques and tools for estimating task durations.
   d. Specify intertask dependencies. The start or completion of individual tasks may be dependent on the start or completion of other tasks. These dependencies impact the completion of any project.
   e. Assign resources. The following resources may impact a project schedule: people, services, facilities and equipment, supplies and materials, and money.
      i) Such resources must be assigned to tasks to develop a schedule.
      ii) Resource leveling is a strategy used to correct resource overallocations by some combination of delaying or splitting tasks. Resource leveling requires knowledge of:
         (1) The critical path—that sequence of dependent tasks that have the largest sum of most likely durations. The critical path determines the earliest possible completion date of the project.
         (2) Slack time—the amount of delay that can be tolerated between the starting time and completion time of a task without causing a delay in the completion date of the entire project.
   f. Direct the team effort. One of the most important dimensions of directing the team effort is the supervision of people.
   g. Monitor and control progress. During the project, the project manager must monitor project progress against the scope, schedule, and budget and, when necessary, make adjustments to scope, schedule, and resources.
      i) Progress reporting is an essential control process that uses communication to keep a project within scope, on time, and within budget.
      ii) A complete project plan provides mechanisms and a process to manage requests for changes to scope. This is called change management.
      iii) Change management frequently requires that a project manager manage the expectations of management and users themselves. An expectations management matrix is a rule-driven tool for helping management understand the dynamics and impact of changing

project parameters such as cost, schedule, scope, and quality.

iv) Schedule adjustments are required when a project's scope changes or when other factors drive schedule or budget out of the projected range.

h. Assess project results and experiences. This final activity involves soliciting feedback from project team members (including customers) concerning their project experiences and suggestions aimed at improving the project and process management of the organization.

Review Questions

1. What is a project?
2. Of the many different reasons that projects fail, what is the major cause of project failure?
3. What is the difference between scope creep and feature creep?
4. What are the five main categories of competencies that a project manager should have?
5. Why are business achievement competencies important?
6. What are the basic project management functions?
7. What are PERT and Gantt charts? How do we decide which one to use?
8. What are the eight major activities in the project management life cycle?
   - Negotiate scope
   - Identify tasks
   - Estimate task durations
   - Specify intertask dependencies
   - Assign resources
   - Direct the team effort
   - Monitor and control progress
   - Assess project results and experiences
9. Why is negotiating scope important? What is the deliverable in the process of negotiating the scope?
10. What is a popular tool used to identify tasks in the project management life cycle?
11. What are the factors to consider in estimating task durations?
12. What are the differences between forward scheduling and reverse scheduling?
13. What are the categories of resources to be allocated to the project?
14. What should project managers do to manage changes that occur and/or are requested during a project?
15. Why is critical path analysis important?

Problems and Exercises

1. Assume you are a systems analyst and a proud member of a project team that has just completed a major project that spanned several years and that touched almost every business unit in your organization. The project was completed ahead of schedule and well within budget. Development and implementation went very smoothly with virtually no disruption of business operations. A postimplementation survey indicates that system users have been able to use the system with minimal training, although there have been some comments from the more vocal users that it wasn't quite what they expected and doesn't do some of the things they thought it would. Should the project be considered a success?
2. Executive management is concerned that some users are less than satisfied with the new system described in the preceding question and have assigned you to lead a postimplementation work group to determine the cause. Of the dozen project mismanagement problems described in the textbook, which ones do you think were most likely to have contributed to user dissatisfaction?
3. As a newly appointed project manager, you are eager to get started on your first project. What should your first activity be? How important is it? Who is typically involved? What questions do you need to make sure are answered? What's the ultimate outcome from this activity, and what is included in this deliverable?
4. You are the project manager of a medium-size project that is scheduled to take 10 months from project initiation on September 1st through delivery on June 30th. It is now April 1st, seven months since the project began, and the project is slightly behind schedule, by perhaps a week. Draw a Gantt chart (you may use the style shown in Figure 4.2 or another Gantt chart style if you prefer). Assume you are using the *FAST* methodology, and that project phases can overlap.
5. You are the project manager for a company that is building a behavioral health system for some of the counties in your state. The project is slightly ahead of schedule and there haven't been any significant problems to date. In reviewing some of the screens under construction, you are surprised to

find a number of features that were not part of the design. The system builder was one of your most talented and creative programmers. When you ask about these features, the builder proudly tells you that they add to the functionality of the system without taking any additional programming time, and that they were intended to be a surprise. You can see that the features definitely do add to the functionality of the system. The code has already been written for them—should you allow them to be included in the system, even though they were not part of the approved technical design?

6. The methodology used in your organization calls for change requests to be considered by a change control board (CCB). After some reflection and a discussion with the programmer, you have decided to submit a change request to the CCB to add the new features. In your presentation to the CCB, what reason might you give for the change request and what things should you take into consideration?
7. The CEO of your organization was so impressed with your last project that you have been given responsibility with a larger, even more important project. The CEO calls you in for a discussion regarding the importance of the project, and tells you that the very survival of the organization may hinge upon completing this project and rolling out the new system to customers before a certain date when a competitor is expected to complete a similar project. The company can afford to budget only up to a certain maximum, although if other, less critical projects-in-progress are delayed, there may be some additional funding available if absolutely necessary. Finally, in order to be a competitive product in the market, the new system must contain at least a certain minimum feature set, although more would be desirable, and the quality must be of the highest level. At the conclusion of this discussion, the CEO shakes your hand and wishes you good luck. Use the priorities set by the CEO to create an initial management expectations matrix.
8. Now suppose that during the course of this project, it becomes apparent that costs were significantly underestimated and the budget is rapidly becoming depleted. In addition, the head of marketing has picked up a trade magazine and read that your organization's main competitor is adding some really exciting features to their product without changing their release date. The budget overage is not the major problem; you know additional money can be allocated, although it may delay other projects. But you also know that your marketing stakeholders will be demanding that similar features be added to the system you are developing while keeping to the original schedule. This presents an expectations conflict since scope is the constrained measure of success. What should you do at this point?
9. Suppose the CEO decides that no matter what, the new features absolutely must be added in order for the new system to be competitive. What issues does this raise, and how would this be reflected in the expectations matrix?
10. You are working on the schedule for the system design phase and are trying to estimate the duration of a complex design task. From breaking this task down into smaller tasks similar to ones that you've had experience with on other projects, you estimate the task should normally take an expected duration (ED) of three workdays, given a typical 75 percent worker efficiency rate and 15 percent interruption factor. But you also know of some instances where absolutely nothing went right, and it took up to two full workweeks, or a pessimistic duration (PD) of 80 hours, to complete the design task. Using the classic technique described in the textbook, calculate the most likely duration of the task.
11. In the preceding question, what technique did you use to estimate the expected duration of the design task? Describe some of the other techniques you could use to estimate task duration.
12. During one phase of the project, you review the project schedule and realize that a member of your project team has been assigned multiple tasks that add up to more hours than the person has available to work during that period. What technique could you use to resolve this?
13. You have been asked to complete a project in shortest time possible. The project tasks, most likely duration (in days), and predecessors are shown below. What are the different paths (sequence of tasks) and the number of days for each? What is the critical path, that is, the shortest time in which the project can be completed? Is it actually important in the business world for project managers to understand critical path analysis, or is this just theoretical knowledge?

Tasks	Duration	Predecessors
A	2	None
B	2	None
C	1	None
D	4	A
E	5	B
F	1	C, D
G	6	A, E
H	4	F
I	7	G,H

14. As a new project manager in a rapidly growing organization, you have been asked to lead a project team for an important project. The scope of the project is not too broad, project time frames are somewhat on the tight side but definitely doable, and the budget is more than generous. In fact, you have been given the authority to hire as many people as you want for your project team. You estimate that 5 people would be about right for this type of project, 8 would provide a healthy amount of backup, and 10 could give you the resources to deliver an outstanding system in record time. What is something you might want to keep in mind before making your decision on how many people to hire?

Projects and Research

1. Projects fail, sometimes spectacularly. Search the Web for articles on major project failures; numerous articles should be readily available. Find and review articles on approximately 10 major project failures during the past decade, then do the following:
   a. List the project failures that you found, and describe them.
   b. What was the cost of each project failure?
   c. What were the consequences of each project failure?
   d. Categorize the reason(s) for each project's failure based upon the causes listed in this chapter.
   e. What were the most common causes for the project failures?
   f. In hindsight, how many of the project failures were avoidable?
   g. What is the most important lesson that new systems analysts can learn from these project failures?

2. The Project Management Institute (PMI) is one of the leading and perhaps *the* leading project management organization in the world. PMI created and maintains the "Project Management Body of Knowledge" (PMBOK), which is a de facto standard for project managers. PMI also certifies a project manager who passes its knowledge and experience requirements as a "Project Management Professional" (PMP).
   a. Go to the PMI Web site (*www.pmi.org*). What are the requirements to become certified as a PMP?
   b. Based upon your readings and experience, how important do you think the PMP certification is for the project manager? Do you think it is worth the investment?
   c. What about the organization that employs a PMP? Is the certification an assurance that the organization's projects will be completed successfully. How much more should an organization pay a project manager with a PMP certification?
   d. Professionals in many fields, such as medicine, engineering, accountancy, and law, are required to be licensed or certified. Do you think professional certification should be required before someone can manage a large project?

3. You work in the information technology division of a large law firm with offices throughout the state. One of the vice presidents of the company has asked you to manage the development of an automated case-tracking system for your company. The project, which is just beginning, is the first large project you have been asked to manage. You take your duties very seriously and want to do an exemplary job on this project.
   a. You are meeting with the vice president of the company to discuss the scope of the project. In your meeting, what questions need to be answered and negotiated in order to be able to determine the scope of the project?
   b. Once you have finished negotiating scope, the vice president has asked you to write a Statement of Work. What does the Statement of Work represent in this situation? How long should it be?
   c. Write a Statement of Work, using the outline shown in Figure 4-5 as an example. Assume that the vice president has given you carte blanche (although that will probably never happen in real life).

4. Project management software, such as Microsoft *Project,* have become commonplace. Many of them incorporate traditional tools, such as PERT and Gantt charts, which were developed decades ago.
   a. Conduct an informal survey of about a dozen project managers in industry. How many of them use project management software?
   b. For those who don't use project management software, what are their reasons for not using it?
   c. For those who do use project management software, which ones do they use? What are their opinions regarding the software they use?

d. Search on the Web for different project management programs. Which ones did you find?
e. Review their features and specifications. Do any of them appear to have unique features? Which one do you think is the most popular, or at least the one most widely used? Which one would you pick if cost was not a consideration?

5. You are managing the development of a case-tracking system project for your large law firm. The requirements phase of the project is almost complete, and preliminary design work has begun. The project is running several days behind schedule, which you don't consider serious, and it is within budget, but barely. Quality, in terms of requirements analysis, has generally been acceptable so far in your opinion, but some of the project team members have mentioned that they are not sure if certain issues have been fully resolved. Based upon this information, write a Project Progress Report, using the outline in Figure 4-11 as an example and following the guidelines described in Activity 7 in this chapter.
6. As part of continuous improvement, it is important for project managers and project teams to assess the results and their experiences once a project has been completed. There are numerous methods and techniques for doing this. Search on the Web for pertinent articles, using phrases such as project assessment, project postimplementation reports, and the like.
   a. What articles did you find?
   b. Describe the methods and techniques they suggest.
   c. Select the ones you feel are the most valuable, and explain why.
   d. Do you think that assessing project results can make a significant difference in the quality of future projects?

Minicases

1. You are on a team that is developing a Web site for a local business, Custom Car Care. There is a set schedule of four months for requirements analysis, development, and successful deployment. The team is on schedule, in week 8, and has just shown Debbie, the CEO of Custom Car Care, the prototype. Debbie is very happy with your work so far, but has some additional capabilities she would like added to the site. Although the additions are not in the previous time or cost estimate, she requires that you stay on schedule and within current budget. What do you do?
2. Alicia and John are a team coding a difficult and sizable program in Java. They have some experience with the language, but will have to learn a significant amount "on the fly." They have estimated that the project will take two months as the optimistic estimate, three months as the expected estimate, and four months as the pessimistic estimate. You are their project manager and must develop a contract for completion with the client for the code development. How much time should you allow in the contract for this deliverable?
3. Develop both a forward and backward schedule of tasks and timelines for a major project you are completing for a class. If there is discrepancy between the two schedules, err on the side of front-loading your tasks. Monitor your project timeline and keep track of the milestones as you complete the project. At the end of the project, submit your timeline and project notes to your professor, along with a copy of the class project. Did the schedule development and management of the project help you? Share with your class your experience.
4. In an interview with a project manager, find out how often personnel issues affect the successful (and on-time) completion of a project. How does this project manager deal with personal or family problems that distract or remove key members of a team?

Team and Individual Exercises

1. For the professor to direct: Create teams of four and designate one as the project manager. Assign them a challenging task with a short deadline. It should be doable for the class, but certainly not easy. Midway through the project, exchange one member per team so that each team has lost one member and gained one new member. Do not allow the team to converse with the member that was "hired away."

   Have the project manager document how they handled the situation, what problems arose, and how they would handle a team differently in the future (knowing that they could

lose a teammate at any time and without any notice).

2. (Team or Individual) For each of the class projects, develop a Rolling Wave timeline for completion. Write down everything you can think of that could go wrong and a contingency project plan. Advice: Front-load each project.
3. As a team, go out to lunch or dinner. Share some aspect of your life that your team may not know about. Find out something you didn't know about each of your teammates.

Suggested Readings

Blanchard, Kenneth, and Spencer Johnson. *The One Minute Manager.* New York: Berkley Publishing Group, 1981, 1982. Arguably, this is one of the best people management books ever written. Available in most bookstores, it can be read overnight and used for discussion material for the lighter side of project management (or any kind of management). This is must reading for all college students with management aspirations.

Blanchard, Kenneth; William Oncken, Jr.; and Hal Burrows. *The One Minute Manager Meets the Monkey.* New York: Simon & Schuster, 1988. A sequel to *The One Minute Manager,* this book effectively looks at the topic of delegation and time management. The monkey refers to Oncken's classic article, "Managing Management Time: Who's Got the Monkey?" as printed in the *Harvard Business Review* in 1974. The book teaches managers how to achieve results by helping their staff (their monkeys) solve their own problems.

Brooks, Fred. *The Mythical Man-Month.* Reading, MA.: Addison-Wesley, 1975. This is a classic set of essays on software engineering (also known as systems analysis, design, and implementation). Emphasis is on managing complex projects.

Catapult, Inc. *Microsoft Project 98: Step by Step.* Redmond, WA: Microsoft Press, 1997. An update for *Project 2000* is expected.

Demarco, Tom. *The Deadline: A Novel about Project Management.* New York: Dorset House Publishing, 1997. This would be an excellent companion to a project management text, especially for a graduate-level course. It demonstrates the "good, bad, and ugly" of project management, told as a story.

Duncan, William R., Director, and Standards Committee. *A Guide to the Project Management Body of Knowledge.* Upper Darby, PA: Project Management Institute, 1996. This is a concise overview of the generally accepted Project Management Body of Knowledge and practices used for certification of project managers.

Friedlander, Phillip. "Ensuring Software Project Success with Project Buyers," *Software Engineering Tools, Techniques, and Practices 2,* no. 6 (March/April 1992), pp. 26–29. We adapted our expectations management matrix from Dr. Friedlander's work.

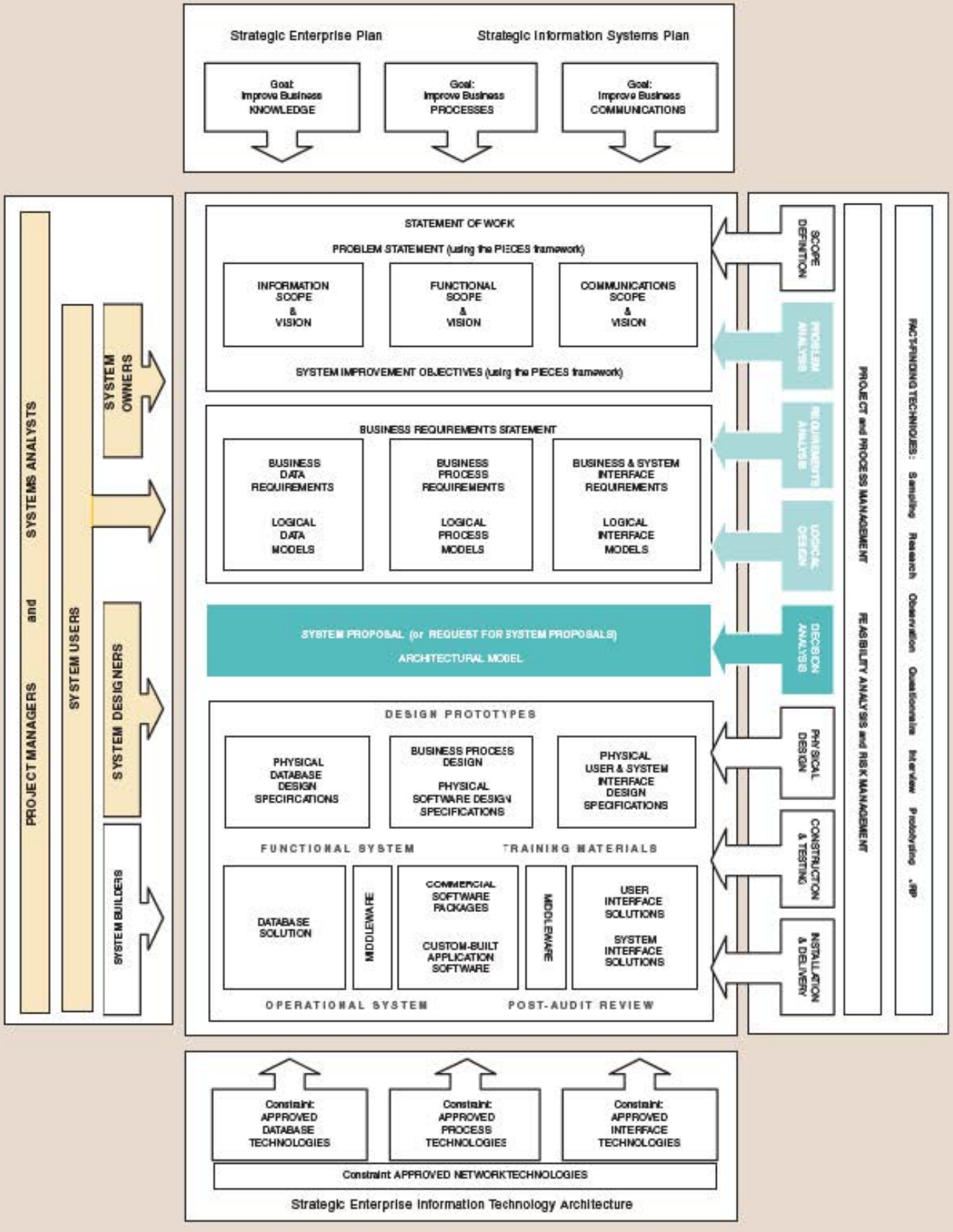
Kerzner, Harold. *Project Management: A Systems Approach to Planning, Scheduling, and Controlling,* 4th ed. New York: Van Nostrand Reinhold, 1989. Many experts consider this book to be the definitive work in the field of project management. Dr. Kerzner's seminars and courses on the subject are renowned.

London, Keith. *The People Side of Systems.* New York: McGraw-Hill, 1976. This is a timeless classic about various people aspects of systems work. Chapter 8, "Handling a Project Team," does an excellent job of teaching the leadership aspects of project management.

McLeod, Graham, and Derek Smith. *Managing Information Technology Projects.* Cambridge, MA: Course Technology, 1996. If you are looking for a good academic book for a course or independent study project to expand your knowledge of IT project and process management, this is it! The book provides a comprehensive treatment of virtually all dimensions of IT project and process management.

Roetzheim, William H., and Reyna A. Beasley. *Software Project Cost and Schedule Estimating: Best Practices.* Upper Saddle River, NJ: Prentice Hall, 1998. This is one of the more complete books on the subject of estimating techniques. Better still, the book includes evaluation copies of *Cost•Xpert, Risk•Xpert,* and *Strategy•Xpert* (™of Marotz, Inc.), software tools for estimating.

Wysocki, Robert K; Robert Beck, Jr; and David B. Crane. *Effective Project Management: How to Plan, Manage, and Deliver Projects on Time and within Budget,* 2nd ed. New York: John Wiley & Sons, 2000. Buy this book! This is our new benchmark for introducing project management. It is easy to read and worth its weight in gold. We were surprised how compatible the book is with past editions of our book, and our project management directions continue to be influenced by this work.


Strategic Enterprise Plan
Strategic Information Systems Plan
Goal: Improve Business KNOWLEDGE
Goal: Improve Business PROCESSES
Goal: Improve Business COMMUNICATIONS
PROJECT MANAGERS and SYSTEMS ANALYSTS
SYSTEM USERS
SYSTEM OWNERS
SYSTEM DESIGNERS
SYSTEM BUILDERS
STATEMENT OF WORK
PROBLEM STATEMENT (using the PIECES framework)
INFORMATION SCOPE & VISION
FUNCTIONAL SCOPE & VISION
COMMUNICATIONS SCOPE & VISION
SYSTEM IMPROVEMENT OBJECTIVES (using the PIECES framework)
BUSINESS REQUIREMENTS STATEMENT
BUSINESS DATA REQUIREMENTS
LOGICAL DATA MODELS
BUSINESS PROCESS REQUIREMENTS
LOGICAL PROCESS MODELS
BUSINESS & SYSTEM INTERFACE REQUIREMENTS
LOGICAL INTERFACE MODELS
SYSTEM PROPOSAL (or REQUEST FOR SYSTEM PROPOSALS)
ARCHITECTURAL MODEL
DESIGN PROTOTYPES
PHYSICAL DATABASE DESIGN SPECIFICATIONS
BUSINESS PROCESS DESIGN
PHYSICAL SOFTWARE DESIGN SPECIFICATIONS
PHYSICAL USER & SYSTEM INTERFACE DESIGN SPECIFICATIONS
FUNCTIONAL SYSTEM
TRAINING MATERIALS
DATABASE SOLUTION
MIDDLEWARE
COMMERCIAL SOFTWARE PACKAGES
CUSTOM-BUILT APPLICATION SOFTWARE
MIDDLEWARE
USER INTERFACE SOLUTIONS
SYSTEM INTERFACE SOLUTIONS
OPERATIONAL SYSTEM
POST-AUDIT REVIEW
SCOPE DEFINITION
PROBLEM ANALYSIS
REQUIREMENTS ANALYSIS
LOGICAL DESIGN
DECISION ANALYSIS
PHYSICAL DESIGN
CONSTRUCTION & TESTING
INSTALLATION & DELIVERY
PROJECT and PROCESS MANAGEMENT
FEASIBILITY ANALYSIS and RISK MANAGEMENT
FACT-FINDING TECHNIQUES: Sampling Research Observation Questionnaires Interviews Prototyping JRP
Constraint: APPROVED DATABASE TECHNOLOGIES
Constraint: APPROVED PROCESS TECHNOLOGIES
Constraint: APPROVED INTERFACE TECHNOLOGIES
Constraint APPROVED NETWORKTECHNOLOGIES
Strategic Enterprise Information Technology Architecture

11

Feasibility Analysis and the System Proposal

Chapter Preview and Objectives

Good systems analysts thoroughly evaluate alternative solutions before proposing change. In this chapter you will learn how to analyze and document those alternatives on the basis of four feasibility criteria: operational, technical, schedule, and economic. You will also learn how to make a system proposal in the form of a written report and a formal presentation. You will know that you understand the feasibility analysis and recommendation skills needed by the systems analyst when you can:

- Identify feasibility checkpoints in the system's life cycle.
- Identify alternative system solutions.
- Define and describe six types of feasibility and their respective criteria.
- Perform various cost-benefit analyses using time-adjusted costs and benefits.
- Write suitable system proposal reports for different audiences.
- Plan for a formal presentation to system owners and users.

Introduction

As all the analysis has been going on for the SoundStage Member Services system project, Bob Martinez has been getting more and more excited about it. The programmer in Bob would like to jump in and start coding the information system. But Sandra, his boss, had him research packaged solutions on the market. They were pricey. But then Sandra ran the numbers for the labor costs of in-house programming. Bob realized that the packaged solutions weren't that expensive relatively and could be put in place a whole lot faster. There would still be programming to do, because the packaged solutions would need to be customized to meet all their requirements.

The final decision on which solution to select would be made by the steering committee that was overseeing the project. Sandra said the executives on the steering committee were currently very budget conscious. They would be scrutinizing the numbers and would approve the project to continue only if it showed a solid return on investment. Bob would have a small part in the system proposal presentation. He rehearsed and studied up on the facts to make sure he was ready for any question. He didn't want to blow it. He was surprised to realize he was now glad that some of his college courses required him to dress in a business suit and make a formal presentation.

Feasibilty Analysis and the System Proposal

In today's business world, it is becoming increasingly apparent that analysts must learn to think like business managers. Computer applications are expanding at a record pace. Now more than ever, management expects information systems to pay for themselves. Information is a major capital investment that must be justified, just as marketing must justify a new product and manufacturing must justify a new plant or equipment. Systems analysts are called on more than ever to help answer the following questions: Will the investment pay for itself? Are there other investments that will return even more on their expenditure?

This chapter deals with feasibility analysis issues of interest to the systems analyst and users of information systems. It also emphasizes the importance of making recommendations to management in the form of a system proposal that is a formal written report and/or oral presentation. As is illustrated in the chapter home page, feasibility analysis is appropriate to the systems analysis phases but particularly important to the decision analysis phase. The system proposal represents the deliverable and presents the technical KNOWLEDGE, PROCESS, and COMMUNICATION solution.

> Feasibility Analysis—A Creeping Commitment Approach

feasibility the measure of how beneficial or practical an information system will be to an organization.

feasibility analysis the process by which feasibility is measured.

Let's begin with a formal definition of feasibility and feasibility analysis. **Feasibility** is the measure of how beneficial or practical the development of an information system will be to an organization. **Feasibility analysis** is the process by which feasibility is measured.

Feasibility should be measured throughout the life cycle. In earlier chapters we called this a *creeping commitment* approach to feasibility. The scope and complexity of an apparently feasible project can change after the initial problems and opportunities are fully analyzed or after the system has been designed. Thus, a project that is feasible at one point may become infeasible later.

Figure 11-1 shows feasibility checkpoints during the systems analysis phases of our life cycle. The checkpoints are represented by red diamonds. The diamonds indicate that a feasibility reassessment and management review should be conducted at the end of the prior phase (before the next phase). A project may be canceled or revised at any checkpoint, despite whatever resources have been spent.

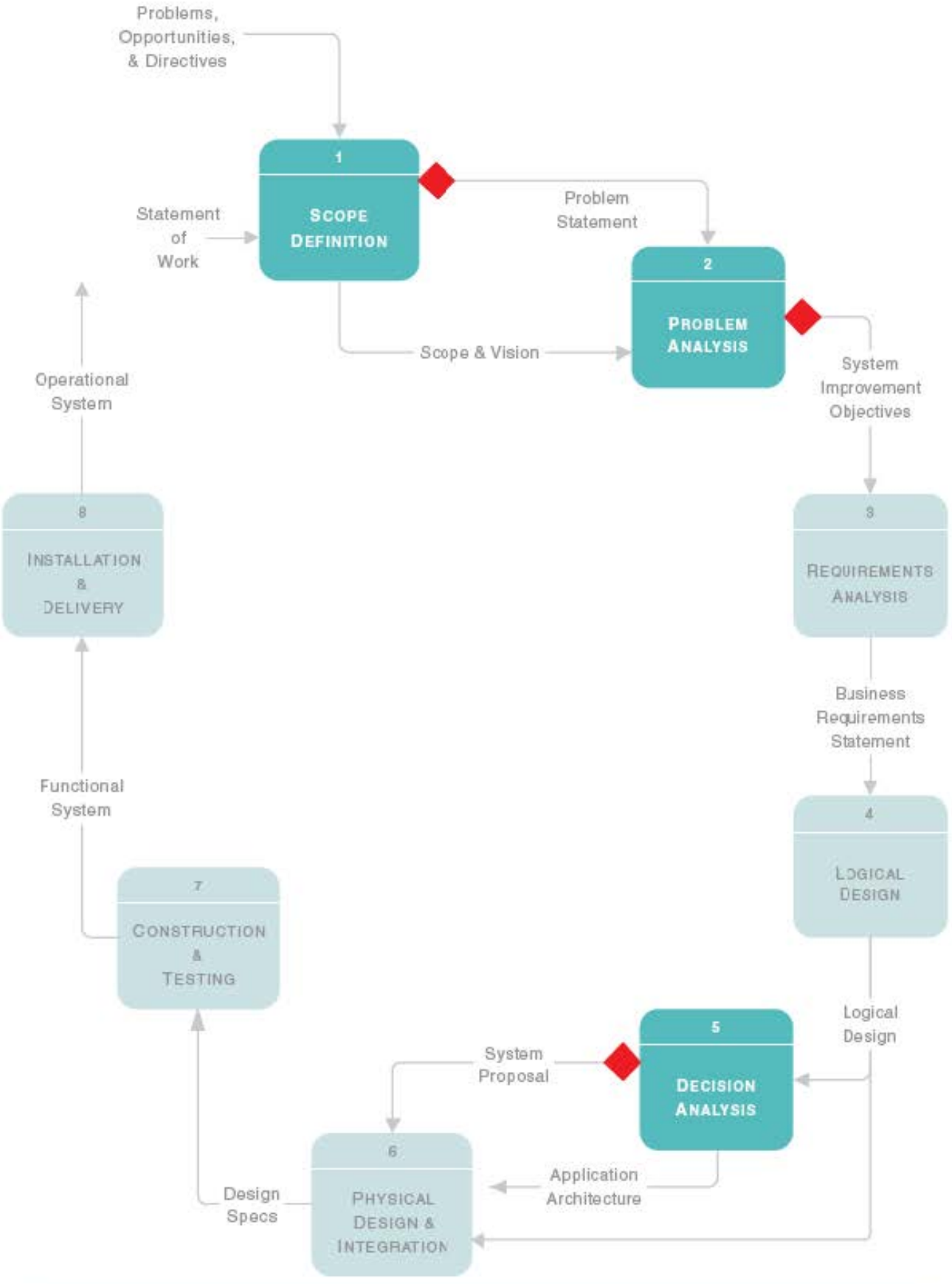


FIGURE 11-1 Feasibility Checkpoints during Systems Analysis

The idea of canceling a project is often difficult to face. A natural inclination may be to justify continuing a project based on the time and money that has already been spent. However, a fundamental principle of management is never to throw good money after bad—cut your losses and move on to a more feasible project. Deciding to cancel doesn't mean the costs already spent are not important. Costs must eventually be recovered if the investment is ever to be considered a success. Let's briefly examine the checkpoints in Figure 11-1.

> Systems Analysis—Scope Definition Checkpoint

The first feasibility analysis is conducted during the scope definition phase. At this early stage of the project, feasibility is rarely more than a measure of the urgency of the problem and the first-cut estimate of development costs. It answers the question, Do the problems (or opportunities) warrant the cost of a detailed study and analysis of the current system? Realistically, feasibility can't be accurately measured until the problems (and opportunities) and requirements are better understood.

After estimating the benefits of solving the problems and opportunities, analysts estimate the costs of developing the expected system. Experienced analysts routinely increase these costs by 50 to 100 percent (or more) because experience tells them the problems are rarely well defined and user requirements are typically understated.

> Systems Analysis—Problem Analysis Checkpoint

The next checkpoint occurs after a more detailed study and problem analysis of the current system. Because the problems are better understood, the analysts can make better estimates of development costs and of the benefits to be obtained from a new system. The minimum value of solving a problem is equal to the cost of that problem. For example, if inventory carrying costs are $35,000 over acceptable limits, then the minimum value of an acceptable information system would be $35,000. It is hoped an improved system will be able to do better than that; however, it must return this minimum value.

Development costs, at this point, are still just guesstimates. Analysts have yet to fully define user requirements or to specify a design solution to those requirements.

If the cost estimates significantly increase from the preliminary investigation phase to the problem analysis phase, the likely culprit is scope. Scope has a tendency to increase in many projects. If increased scope threatens feasibility, then scope might be reduced.

> Systems Design—Decision Analysis Checkpoint

The decision analysis phase represents a major feasibility analysis activity since it charts one of many possible implementations as the target for systems design.

Problems and requirements should be known by now. During the decision analysis phase, alternative solutions are defined in terms of their input/output methods, data storage methods, computer hardware and software requirements, processing methods, and people implications. The following list presents the typical range of options that can be evaluated by the analyst:

- Do nothing. Leave the current system alone. Regardless of management's opinion or your own opinion, this option should be considered and analyzed as a baseline option against which all others can and should be evaluated.
- Reengineer the (manual) business processes, not the computer-based processes. This may involve streamlining activities, reducing duplication and unnecessary tasks, reorganizing office layouts, and eliminating redundant and unnecessary forms and processes, among others.
- Enhance existing computer processes.
- Purchase a packaged application.
- Design and construct a new computer-based system.

After defining these options, each is analyzed for operational, technical, schedule, and economic feasibility. One alternative is recommended to system owners for approval and the basis for general and detailed design.

Six Tests for Feasibility

So far, we've defined feasibility and feasibility analysis, and we've identified feasibility checkpoints during systems analysis. Feasibility can be viewed from multiple perspectives. Below we present six categories of feasibility tests.

- *Operational feasibility* is a measure of how well a solution meets the identified system requirements to solve the problems and take advantage of the opportunities envisioned for the system.
- *Cultural (or political) feasibility* is a measure of how people feel about a solution and how well it will be accepted in a given organizational climate.
- *Technical feasibility* is a measure of the practicality of a specific technical solution and the availability of technical resources and expertise to implement and maintain it.
- *Schedule feasibility* is a measure of how reasonable the project timetable is.
- *Economic feasibility* is a measure of the cost-effectiveness of a project or solution.
- *Legal feasibility* is a measure of how well a solution can be implemented within existing legal and contractual obligations.

Actually, few systems are infeasible. Instead, different solution options tend to be more or less feasible than others. Let's take a closer look at the four feasibility criteria.

> Operational Feasibility

Operational feasibility is the measure of how well a proposed system solves the problems and takes advantage of the opportunities identified during the scope definition and problem analysis phases and how well it satisfies the system requirements identified in the requirements analysis phase. Operational feasibility also asks if, given what is now known about the problem and the cost of the solution, the problem is still worth solving. The PIECES framework (Chapter 3) can be used as the basis for analyzing the urgency of a problem or the effectiveness of a solution.

operational feasibility a measure of how well a solution meets the identified system requirements to solve the problems and take advantage of the opportunities envisioned for the system.

> Cultural (or Political) Feasibility

This is related to operational feasibility. But where operational feasibility deals more with how well the solution will meet system requirements, **cultural/political feasibility** deals with how the end users feel about the proposed system. You could say that operational feasibility evaluates whether a system *can* work, and cultural/political feasibility asks whether a system *will* work in a given organizational climate.

In an information age, knowledge is power. It is common for an information system to change the structure of how information is routed and controlled, changing to some extent the power structure of the organization. Some users and managers may feel threatened and fight implementation of the system.

Recognize that increasingly the culture of an organization is multicultural. Employees and divisions may have been merged in from different companies with widely varying perspectives on how work should be structured and what information systems should do and not do. With international organizations, the information system must also be accepted by multiple national cultures. The following questions address this concern:

- Does management support the system?
- How do the end users feel about their role in the new system?

cultural (or political) feasibility a measure of how well the solution will be accepted in a given organizational climate.

- What end users or managers may resist or not use the system? Can this problem be overcome? If so, how?
- How will the working environment of the end users change? Can or will end users and management adapt to the change?

> Technical Feasibility

technical feasibility a measure of the practicality of a technical solution and the availability of technical resources and expertise.

Today, very little is technically impossible. Consequently, technical feasibility looks at what is practical and reasonable. Technical feasibility addresses three major issues:

1. Is the proposed technology or solution practical?
2. Do we currently possess the necessary technology?
3. Do we possess the necessary technical expertise?

Is the Proposed Technology or Solution Practical? The technology for any defined solution is normally available. The question is whether that technology is mature enough to be easily applied to our problems. Some firms like to use state-of-the-art technology, but most firms prefer to use mature and proven technology. A mature technology has a larger customer base for obtaining advice concerning problems and improvements.

Do We Currently Possess the Necessary Technology? Assuming the solution's required technology is practical, we must next ask ourselves, Is the technology available in our information systems shop? If the technology is available, we must ask if we have the capacity. For instance, will our current printer be able to handle the new reports and forms required of a new system?

If the answer to either of these questions is no, then we must ask ourselves, Can we get this technology? The technology may be practical and available, and, yes, we need it. But we simply may not be able to afford it at this time. Although this argument borders on economic feasibility, it is truly technical feasibility. If we can't afford the technology, then the alternative that requires the technology is not practical and is technically infeasible!

Do We Possess the Necessary Technical Expertise? This consideration of technical feasibility is often forgotten during feasibility analysis. Even if a company has the technology, that doesn't mean it has the skills required to properly apply that technology. For instance, a company may have a database management system (DBMS). However, the analysts and programmers available for the project may not know that DBMS well enough to properly apply it. True, all information systems professionals can learn new technologies; however, that learning curve will impact the technical feasibility of the project—specifically, it will impact the schedule.

> Schedule Feasibility

schedule feasibility a measure of how reasonable a project timetable is.

Given the available technical expertise, are the project deadlines reasonable—that is, what is the **schedule feasibility** of the project? Some projects are initiated with specific deadlines. It is necessary to determine whether the deadlines are mandatory or desirable. For instance, a project to develop a system to meet new government reporting regulations may have a deadline that coincides with when the new reports must be initiated. Penalties associated with missing such a deadline may make meeting it mandatory. If the deadlines are desirable rather than mandatory, the analyst can propose alternative schedules.

It is preferable (unless the deadline is absolutely mandatory) to deliver a properly functioning information system two months late than to deliver an error-prone, useless information system on time! While missing deadlines can be problematic, developing inadequate systems can be disastrous. It's a choice between the lesser of two evils.

> Economic Feasibility

The bottom line in many projects is **economic feasibility.** During the early phases of the project, economic feasibility analysis amounts to little more than judging whether the possible benefits of solving the problem are worthwhile. Costs are practically impossible to estimate at that stage because the end user's requirements and alternative technical solutions have not been identified. However, as soon as specific requirements and solutions have been identified, the analyst can weigh the costs and benefits of each alternative. This is called a cost-benefit analysis. Cost-benefit analysis is discussed later in this chapter.

economic feasibility a measure of the cost-effectiveness of a project or solution.

> Legal Feasibility

Information systems have a legal impact. First of all, there are copyright restrictions. For any system that includes purchased components, one has to make sure that the license agreements are not violated. For one thing this means installing only licensed copies. But license agreements and copy protection can also restrict how you integrate the data and processes with other parts of the system. If you are working with contract programmers, the ownership of the program source code and nondisclosure agreements have to be worked out in advance.

Union contracts can add constraints to the information system on how workers are paid and how their work is monitored. Legal requirements for financial reporting must be met. System requirements for sharing data with partners could even run up against antitrust laws. Finally, many information systems today are international in scope. Some countries mandate where data on local employees and local transactions must be stored and processed. Countries differ on the number of hours that make up a workweek or how long employees break for lunch.

legal feasibility is a measure of how well a solution can be implemented within existing legal and contractual obligations.

> The Bottom Line

We have now discussed the fact that any alternative solution can be evaluated according to six criteria: operational, cultural/political, technical, schedule, economic, and legal feasibility. How does an analyst pick the best solution? It's not easy. Operational and economic issues often conflict. For example, the solution that provides the best operational impact for end users may also be the most expensive and, therefore, the least economically feasible. The final decision can be made only by sitting down with end users, reviewing the data, and choosing the best overall alternative.

Cost-Benefit Analysis Techniques

Economic feasibility has been defined as a cost-benefit analysis. How can costs and benefits be estimated? How can those costs and benefits be compared to determine economic feasibility? Most schools offer complete courses on these subjects—courses on financial management, financial decision analysis, and engineering economics and analysis. This section presents an overview of the techniques.

> How Much Will the System Cost?

Costs fall into two categories. There are costs associated with developing the system, and there are costs associated with operating a system. The former can be estimated from the outset of a project and should be refined at the end of each phase of the

project. The latter can be estimated only after specific computer-based solutions have been defined. Let's take a closer look at the costs of information systems.

The costs of developing an information system can be classified according to the phase in which they occur. Systems development costs are usually onetime costs that will not recur after the project has been completed. Many organizations have standard cost categories that must be evaluated. In the absence of such categories, the following list should help:

- *Personnel costs*—The salaries of systems analysts, programmers, consultants, data entry personnel, computer operators, secretaries, and the like, who work on the project make up the personnel costs. Because many of these individuals spend time on many projects, their salaries should be prorated to reflect the time spent on the projects being estimated.
- *Computer usage*—Computer time will be used for one or more of the following activities: programming, testing, conversion, word processing, maintaining a project dictionary, prototyping, loading new data files, and the like. If a computing center charges for usage of computer resources such as disk storage or report printing, the cost should be estimated.
- *Training*—If computer personnel or end users have to be trained, the training courses may incur expenses. Packaged training courses may be charged out on a flat fee per site, a student fee (such as $395 per student), or an hourly fee (such as $75 per class hour).
- *Supply, duplication, and equipment costs.*
- *Cost of any new computer equipment and software.*

Sample development costs for a typical solution are displayed in Figure 11-2. When analysts are estimating development costs, it is important that money be set aside for the possibility that a system will incur costs after it is operating. The lifetime benefits must recover both the developmental and the operating costs. Unlike system development costs, operating costs tend to recur throughout the lifetime of the system. The costs of operating a system over its useful lifetime can be classified as fixed or variable.

fixed cost a cost that occurs at a regular interval and at a relatively fixed rate.

Fixed costs occur at regular intervals but at relatively fixed rates. Examples of fixed operating costs include:

- Lease payments and software license payments.
- Prorated salaries of information systems operators and support personnel (although salaries tend to rise, the rise is gradual and tends not to change dramatically from month to month).

variable cost a cost that occurs in proportion to some usage factor.

Variable costs occur in proportion to some usage factor. Examples include:

- Costs of computer usage (e.g., CPU time used, terminal connect time used, storage used), which vary with the workload.
- Supplies (e.g., preprinted forms, printer paper used, punched cards, floppy disks, magnetic tapes, and other expendables), which vary with the workload.
- Prorated overhead costs (e.g., utilities, maintenance, and telephone service), which can be allocated throughout the lifetime of the system using standard techniques of cost accounting.

Sample operating cost estimates for a solution are also displayed in Figure 11-2.

> What Benefits Will the System Provide?

Benefits normally increase profits or decrease costs, both highly desirable characteristics of a new information system. As much as possible, benefits should be quantified in dollars and cents; they should also be classified as tangible or intangible.

tangible benefit a benefit that can be easily quantified.

Tangible benefits are those that can be easily quantified. Tangible benefits are usually measured in terms of monthly or annual savings or of profit to the firm. For example, consider the following scenario:

Estimated Costs for Client-Server System Alternative

DEVELOPMENT COSTS

Personnel:

2	Systems Analysts (400 hours/ea $50.00/hr)	$40,000
4	Programmer/Analysts (250 hours/ea $35.00/hr)	$35,000
1	GUI Designer (200 hours/ea $40.00/hr)	$8,000
1	Telecommunications Specialist (50 hours/ea $50.00/hr)	$2,500
1	System Architect (100 hours/ea $50.00/hr)	$5,000
1	Database Specialist (15 hours/ea $45.00/hr)	$675
1	System Librarian (250 hours/ea $15.00/hr)	$3,750

Expenses:

4	Smalltalk training registration ($3,500.00/student)	$14,000

New Hardware & Software:

1	Development Server	$18,700
1	Server software (operating system, misc.)	$1,500
1	DBMS server software	$7,500
7	DBMS client software ($950.00 per client)	$6,650

Total Development Costs: $143,275

PROJECTED ANNUAL OPERATING COSTS

Personnel:

2	Programmer/Analysts (125 hours/ea $35.00/hr)	$8,750
1	System Librarian (20 hours/ea $15.00/hr)	$300

Expenses:

1	Maintenance Agreement for server	$995
1	Maintenance Agreement for server DBMS software	$525
	Preprinted forms (15,000/year @ .22/form)	$3,300

Total Projected Annual Costs: **$13,870**

FIGURE 11-2 Costs for a Proposed Systems Solution

While processing student housing applications, we discover that considerable data is being redundantly typed and filed. An analysis reveals that the same data is typed seven times, requiring an average of 44 additional minutes of clerical work per application. The office processes 1,500 applications per year. That means a total of 66,000 minutes or 1,100 hours of redundant work per year. If the average salary of a secretary is $15 per hour, the cost of this problem and the benefit of solving the problem is $16,500 per year.

Alternatively, tangible benefits might be measured in terms of unit cost savings or profit. For instance, an alternative inventory valuation scheme may reduce inventory carrying cost by $0.32 per unit of inventory. Some examples of tangible benefits are listed in the margin.

TANGIBLE BENEFITS

Fewer Processing Errors
Increased Throughput
Decreased Response Time
Elimination of Job Steps
Increased Sales
Reduced Credit Losses
Reduced Expenses

Other benefits are intangible. **Intangible benefits** are those that are believed to be difficult or impossible to quantify. Unless these benefits are at least identified, it is entirely possible that many projects would not be feasible. Examples of intangible benefits are listed in the margin on the next page.

INTANGIBLE BENEFITS

Improved Customer Goodwill
Improved Employee Morale
Better Service to Community
Better Decision Making

intangible benefit a benefit that is believed to be difficult or impossible to quantify.

Unfortunately, if a benefit cannot be quantified, it is difficult to accept the validity of an associated cost-benefit analysis that is based on incomplete data. Some analysts dispute the existence of intangible benefits. They argue that all benefits are quantifiable; some are just more difficult to quantify than others. Suppose, for example, that improved customer goodwill is listed as a possible intangible benefit. Can we quantify goodwill? You might try the following analysis:

1. What is the result of customer ill will? The customer will submit fewer (or no) orders.
2. To what degree will a customer reduce orders? A user may find it difficult to specifically quantify this impact, but you could try to have the end user estimate the possibilities (or invent an estimate to which the end user can react). For instance:
   a. There is a 50 percent (.50) chance that the regular customer would send a few orders—fewer than 10 percent of all its orders—to competitors to test their performance.
   b. There is a 20 percent (.20) chance that the regular customer would send as many as half its orders (.50) to competitors, particularly those orders we are historically slow to fulfill.
   c. There is a 10 percent (.10) chance that a regular customer would send us an order only as a last resort. That would reduce that customer's normal business with us to 10 percent of its current volume (90 percent, or .90, loss).
   d. There is a 5 percent (.05) chance that a regular customer would choose not to do business with us at all (100 percent or 1.00 loss).
3. We can calculate an estimated business loss as follows:

$$\begin{aligned}\text{Loss} = {} & .50 \times (.10 \text{ loss of business}) \\ & + .20 \times (.50 \text{ loss of business}) \\ & + .10 \times (.90 \text{ loss of business}) \\ & + .50 \times (1.00 \text{ loss of business}) \\ = {} & .29 \\ = {} & 29\% \text{ statistically estimated loss of business}\end{aligned}$$

4. If the average customer does $40,000 per year of business, then we can expect to lose 29 percent, or $11,600, of that business. If we have 500 customers, this can be expected to amount to a total of $5,800,000.
5. Present this analysis to management, and use it as a starting point for quantifying the benefit.

> Is the Proposed System Cost-Effective?

There are three popular techniques for assessing economic feasibility, also called *cost-effectiveness:* payback analysis, return on investment, and net present value.

The choice of techniques should consider the audiences that will use them. Virtually all managers who have come through business schools are familiar with all three techniques. One concept that should be applied to each technique is the adjustment of cost and benefits to reflect the time value of money.

The Time Value of Money A concept shared by all three techniques is the **time value of money**—a dollar today is worth more than a dollar one year from now. You could invest that dollar today and, through accrued interest, have more than one dollar a year from now. Thus, you'd rather have that dollar today than in one year. That's why your creditors want you to pay your bills promptly—they can't invest what they don't have. The same principle can be applied to costs and benefits *before* a cost-benefit analysis is performed.

Some of the costs of a system will be accrued after implementation. Additionally, all benefits of the new system will be accrued in the future. Before cost-benefit analysis, these costs should be brought back to current dollars. An example should clarify the concept.

Suppose we are going to realize a benefit of $20,000 two years from now. What is the current dollar value of that $20,000 benefit? If the current return on investments is running about 10 percent, an investment of $16,528 today would give us our $20,000 in two years (we'll show you how to calculate this later). Therefore, the current value of the estimated benefit is $16,528—that is, we'd rather have $16,528 today than the promise of $20,000 two years from now.

Because projects are often compared against other projects that have different lifetimes, time-value analysis techniques have become the preferred cost-benefit methods for most managers. By time-adjusting costs and benefits, you can improve the following cost-benefit techniques.

Payback Analysis The **payback analysis** technique is a simple and popular method for determining if and when an investment will pay for itself. Because system development costs are incurred long before benefits begin to accrue, it will take some time for the benefits to overtake the costs. After implementation, you will incur additional operating expenses that must be recovered. Payback analysis determines how much time will elapse before accrued benefits overtake accrued and continuing costs. This period of time is called the **payback period.**

payback analysis a technique for determining if and when an investment will pay for itself.

payback period the period of time that will elapse before accrued benefits overtake accrued costs.

In Figure 11-3 we see an information system that will be developed at a cost of $418,040. The estimated net operating costs for each of the next six years are also recorded in the table. The estimated net benefits over the same six operating years are also shown. What is the payback period?

First, we need to adjust the costs and benefits for the time value of money (that is, adjust them to current dollar values). Here's how: The present value of a dollar in year *n* depends on something typically called a **discount rate.** The discount rate is a percentage similar to interest rates that you earn on your savings account. In most cases the discount rate for a business is the **opportunity cost** of being able to invest money in other projects, including the possibility of investing in the stock market, money market funds, bonds, and the like. Alternatively, a discount rate could represent

	A	B	C	D	E	F	G	H
4	Cash flow description	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6
5	Development cost:	($418,040)						
6	Operation & maintenance cost:		($15,045)	($16,000)	($17,000)	($18,000)	($19,000)	($20,000)
7	Discount factors for 12%:	1.000	0.893	0.797	0.712	0.636	0.567	0.507
8	Time-adjusted costs (adjusted to present value):	($418,040)	($13,435)	($12,752)	($12,104)	($11,449)	($10,773)	($10,140)
9	Cumulative time-adjusted costs over lifetime:	($418,040)	($431,475)	($444,227)	($456,331)	($467,779)	($478,552)	($488,693)
10								
11	Benefits derived from operation of new system:	$0	$150,000	$170,000	$190,000	$210,000	$230,000	$250,000
12	Discount factors for 12%:	1.000	$0.893	$0.797	$0.712	$0.636	$0.567	$0.507
13	Time-adjusted benefits (current of present value):	$0	$133,950	$135,490	$135,280	$133,560	$130,410	$126,750
14	Cumulative time-adjusted benefits over lifetime:	$0	$133,950	$269,440	$404,720	$538,280	$668,690	$795,440
15		0	1	2	3	4	5	6
16	Cumulative lifetime time-adjusted costs + benefits:	($418,040)	($297,525)	($174,787)	($51,611)	$70,501	$190,138	$306,748

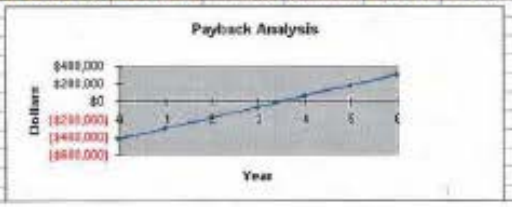


FIGURE 11-3
Payback Analysis for a Project

FIGURE 11-4 Partial Table For Present Value Of A Dollar

Periods	8%	9%	10%	11%	12%	13%	14%
1	0.926	0.917	0.909	0.901	0.893	0.885	0.877
2	0.857	0.842	0.826	0.812	0.797	0.783	0.769
3	0.794	0.772	0.751	0.731	0.712	0.693	0.675
4	0.735	0.708	0.683	0.659	0.636	0.613	0.592
5	0.681	0.650	0.621	0.593	0.567	0.543	0.519
6	0.630	0.596	0.564	0.535	0.507	0.480	0.456
7	0.583	0.547	0.513	0.482	0.452	0.425	0.400
8	0.540	0.502	0.467	0.434	0.404	0.376	0.351

what the company considers an acceptable return on its investments. This number can be learned by asking any financial manager, officer, or comptroller.

present value the current value of a dollar at any time in the future.

Let's say the discount rate for our sample company is 12 percent. The current value, actually called the **present value**, of a dollar at any time in the future can be calculated using the following formula:

$$PV_n = 1/(1 + i)^n$$

where PV_n is the present value of \$1.00 n years from now and i is the discount rate. Therefore, the present value of a dollar two years from now is

$$PV_2 = 1/(1 + .12)^2 = 0.797$$

Earlier we stated that a dollar today is worth more than a dollar a year from now. But it looks as if it is worth less. This is an illusion. The present value is interpreted as follows. If you have 79.7 cents today, it is better than having 79.7 cents two years from now. How much better? Exactly 20.3 cents better since that 79.7 cents would grow into one dollar in two years (assuming our 12 percent discount rate).

To determine the present value of any cost or benefit in year 2, you simply multiply 0.797 times the estimated cost or benefit. For example, the estimated operating expense in year 2 is \$16,000. The present value of this expense is \$16,000 × 0.797, or \$12,752 (rounded up). Fortunately, you don't have to calculate discount factors. There are tables similar to the partial one shown in Figure 11-4 that show the present value of a dollar for different time periods and discount rates. Simply multiply this number times the estimated cost or benefit to get the present value of that cost or benefit. More detailed versions of this table can be found in many accounting and finance books as well as in spreadsheet functions.

Better still, most spreadsheets include built-in functions for calculating the present value of any cash flow, be it cost or benefit. All the examples in this module were done with Microsoft *Excel*. The same tables can be prepared with *Lotus 1-2-3*. The beauty of a spreadsheet is that once the rows, columns, and functions have been set up, you simply enter the costs and benefits and let the spreadsheet discount the numbers to present value. (In fact, you can also program the spreadsheet to perform the cost-benefit analysis.)

In Figure 11-3, notice that we have brought all costs and benefits for our example back to present value. Also notice that the discount rate for year 0 is 1.000. Why? The present value of a dollar in year 0 is exactly \$1. In other words, if you hold a dollar today, it is worth exactly \$1.

Now that we've discounted the costs and benefits, we can complete our payback analysis. Look at the cumulative lifetime costs and benefits. The lifetime costs are gradually increasing over the six-year period because operating costs are being incurred. But also notice that the lifetime benefits are accruing at a much faster pace.

Lifetime benefits will overtake the lifetime costs between years 3 and 4. By charting the cumulative lifetime time-adjusted costs and benefits, we can estimate that the break-even point (when Costs + Benefits = 0) will occur approximately 3.5 years after the system begins operating.

Is this information system a good or bad investment? It depends. Many companies have a payback period guideline for all investments. In the absence of such a guideline, you need to determine a reasonable guideline before you determine the payback period. Suppose that the guideline states that all investments must have a payback period less than or equal to four years. Because our example has a payback period of 3.5 years, it is a good investment. If the payback period for the system were greater than four years, the information system would be a bad investment.

It should be noted that you can perform payback analysis without time-adjusting the costs and benefits. The result, however, would show a 2.8-year payback that looks more attractive than the 3.5-year payback that we calculated. Thus, non-time-adjusted paybacks tend to be overly optimistic and misleading.

Return-on-Investment Analysis The **return-on-investment (ROI) analysis** technique compares the lifetime profitability of alternative solutions or projects. The ROI for a solution or project is a percentage rate that measures the relationship between the amount the business gets back from an investment and the amount invested. The lifetime ROI for a potential solution or project is calculated as follows:

return-on-investment (ROI) analysis a technique that compares the lifetime profitability of alternative solutions.

$$\text{Lifetime ROI} = (\text{Estimated lifetime benefits} - \text{Estimated lifetime costs}) / \text{Estimated lifetime costs}$$

Let's calculate the lifetime ROI for the same systems solution we used in our discussion of payback analysis. Once again, all costs and benefits should be time-adjusted over a period of six years. The time-adjusted costs and benefits were presented in rows 9 and 16 of Figure 11-3. The estimated lifetime benefits minus estimated lifetime costs equal

$$\$795{,}440 - \$488{,}692 = \$306{,}748$$

Therefore, the lifetime ROI is

$$\text{Lifetime ROI} = \$306{,}748/\$488{,}692 = .628 = 63\%$$

This is a lifetime ROI, *not* an annual ROI. Simple division by the lifetime of the system (63 ÷ 6) yields an average ROI of 10.5 percent per year. This solution can be compared with alternative solutions. The solution offering the highest ROI is the best alternative. However, as was the case with payback analysis, the business may set a minimum acceptable ROI for all investments. If none of the alternative solutions meets or exceeds that minimum standard, then none of the alternatives is economically feasible. Once again, spreadsheets can greatly simplify ROI analysis through their built-in financial analysis functions.

As with payback analysis, we could have calculated the ROI without time-adjusting the costs and benefits. This would, however, result in a misleading 129.4 percent lifetime or a 21.6 percent annual ROI. Consequently, we recommend time-adjusting all costs and benefits to current dollars.

Net Present Value The **net present value** of an investment alternative is considered the preferred cost-benefit technique by many managers, especially those who have substantial business schooling. Once again, you initially determine the costs and benefits for each year of the system's lifetime. And once again, we need to adjust all the costs and benefits back to present dollar values.

net present value an analysis technique that compares the annual discounted costs and benefits of alternative solutions.

Figure 11-5 illustrates the net present value technique. Costs are represented by negative cash flows, while benefits are represented by positive cash flows. We have brought all costs and benefits for our example back to present value. Notice again that the discount rate for year 0 (used to accumulate all development costs) is 1.000 because the present value of a dollar in year 0 is exactly $1.

FIGURE 11-5
Net Present Value Analysis for a Project

	A	B	C	D	E	F	G	H	I	J
1	Net Present Value Analysis for Client-Server System Alternative									
2	(Numbers rounded to nearest $1)									
3										
4	Cash flow description	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5	Year 6	Total	
5	Development cost:	($418,040)								
6	Operation & maintenance cost:		($ 5,045)	($18,000)	($17,000)	($16,000)	($16,000)	($20,000)		
7	Discount factors for 12%:	1.000	0.893	0.797	0.712	0.636	0.567	0.507		
8	Present value of annual costs:	($418,040)	($ 3,435)	($12,752)	($12,104)	($11,448)	($10,773)	($10,140)		
9	Total present value of lifetime costs:								($488,692)	
10										
11	Benefits derived from operation of new	$0	$150,000	$170,000	$180,000	$210,000	$230,000	$250,000		
12	Discount factors for 12%:	1.000	0.893	0.797	0.712	0.636	0.567	0.507		
13	Present value of annual benefits:	$0	$133,950	$135,490	$135,280	$133,560	$130,410	$126,750		
14	Total present value of lifetime benefits:								$795,440	
15										
16	NET PRESENT VALUE OF THIS ALTERNATIVE:								$306,748	
17										

After discounting all costs and benefits, subtract the sum of the discounted costs from the sum of the discounted benefits to determine the net present value. If it is positive, the investment is good. If negative, the investment is bad. When comparing multiple solutions or projects, the one with the highest positive net present value is the best investment. (This works even if the alternatives have different lifetimes!) In our example the solution being evaluated yields a net present value of $306,748. This means that if we invest $306,748 at 12 percent for six years, we will make the same profit that we'd make by implementing this information systems solution. This is a good investment provided no other alternative has a net present value greater than $306,748.

Once again, spreadsheets can greatly simplify net present value analysis through their built-in financial analysis functions.

Feasibility Analysis of Candidate Systems

During the decision analysis phase of system analysis, the systems analyst identifies candidate system solutions and then analyzes those solutions for feasibility. We discussed the criteria and techniques for analysis in this chapter. In this section, we evaluate a pair of documentation techniques that can greatly enhance the comparison and contrast of candidate system solutions. Both use a matrix format. We have found these matrices useful for presenting candidates and recommendations to management.

> Candidate Systems Matrix

candidate systems matrix a tool used to document similarities and differences between candidate systems.

The first matrix allows us to compare candidate systems on the basis of several characteristics. The **candidate systems matrix** documents similarities and differences between candidate systems; however, it offers no analysis.

The columns of the matrix represent candidate solutions. Experienced analysts always consider multiple implementation options. At least one of those options should be the existing system because it serves as a baseline for comparing alternatives.

The rows of the matrix represent characteristics that differentiate the candidates. For purposes of this book, we based some of the characteristics on the information system building blocks. The breakdown is as follows:

- *Stakeholders*—Identify how the system will interact with people and other systems.
- *Knowledge*—Identify how data stores will be implemented (e.g., conventional files, relational databases, other database structures), how inputs will be

FIGURE 11-6 Candidate Systems Matrix Template

	Candidate 1 Name	Candidate 2 Name	Candidate 3 Name
Stakeholders			
Knowledge			
Processes			
Communications			

captured (e.g., online, batch, etc.), how outputs will be generated (e.g., on a schedule, on demand, printed, on screen, etc.).

- *Processes*—Identify how (manual) business processes will be modified, how computer processes will be implemented. For the latter, we have numerous options, including online versus batch processes and packaged versus built-in-house software.
- *Communications*—Identify how processes and data will be distributed. Once again, we might consider several alternatives—for example, centralized versus decentralized versus distributed (or duplicated) versus cooperative (client/server) solutions. Network distribution types and strategies will be discussed in Chapter 13.

The cells of the matrix document whatever characteristics help the reader understand the differences between options. Figure 11-6 illustrates the basic structure of the matrix.

Before considering any solutions, we must consider any constraints on solutions. Solution constraints take the form of architectural decisions intended to bring order and consistency to applications. For example, a technology architecture may restrict solutions to relational databases or client/server networks.

There are several approaches for identifying candidate solutions, including:

- *Recognizing users' ideas and opinions*—Throughout a systems project, users may suggest manual or technology-related solutions. They should be given consideration.
- *Consulting methodology and architecture standards*—Many organizations' development methodology and architecture standards may dictate how technology solutions are to be selected and what technology(ies) may be represented.
- *Brainstorming possible solutions*—Brainstorming is an effective technique for identifying possible solutions. It is particularly effective when done using an organized approach or framework, such as the IS building blocks or other IS characteristics. Brainstorming should encompass solutions that represent buy, build, and a combination of buy and build options.
- *Seeking references*—The analyst should solicit ideas and opinions from other persons and organizations that have implemented similar systems.
- *Browsing appropriate journals and periodicals*—Such literature may feature advertisements and articles concerning automation strategies, successes, failures, and technologies.

A combination of the above approaches could be used independently by the development team members to derive a number of possible alternative system solutions.

A sample, partially completed candidate systems matrix listing three of the five candidates is shown in Figure 11-7. In the figure, the matrix is used to provide

FIGURE 11-7 Sample Candidate Systems Matrix

Characteristics	Candidate 1	Candidate 2	Candidate 3	Candidate . . .
Portion of System Computerized Brief description of that portion of the system that would be computerized in this candidate.	COTS package Platinum Plus from Entertainment Software Solutions would be purchased and customized to satisfy Member Services required functionality.	Member Services and warehouse operations in relation to order fulfillment.	Same as candidate 2.	
Benefits Brief description of the business benefits that would be realized for this candidate.	This solution can be implemented quickly because it's a purchased solution.	Fully supports user-required business processes for SoundStage Inc. Plus more efficient interaction with member accounts.	Same as candidate 2.	
Servers and Workstations A description of the servers and workstations needed to support this candidate.	Technically, architecture dictates Pentium III, MS Windows 2000 class servers and workstations (clients).	Same as candidate 1.	Same as candidate 1.	
Software Tools Needed Software tools needed to design and build the candidate (e.g., database management system, emulators, operating systems, languages). Not generally applicable if applications software packages are to be purchased.	MS Visual C++ and MS Access for customization of package to provide report writing and integration.	MS Visual Basic 5.0 System Architect 2001 Internet Explorer	MS Visual Basic 5.0 System Architect 2001 Internet Explorer	
Application Software A description of the software to be purchased, built, accessed, or some combination of these techniques.	Package solution	Custom solution	Same as candidate 2.	
Method of Data Processing Generally some combination of online, batch, deferred batch, remote batch, and real time.	Client/Server	Same as candidate 1.	Same as candidate 1.	
Output Devices and Implications A description of output devices that would be used, special output requirements (e.g., network, preprinted forms, etc.), and output considerations (e.g., timing constraints).	(2) HP4MV department laser printers (2) HP5SI LAN laser printers	(2) HP4MV department laser printers (2) HP5SI LAN laser printers (1) PRINTRONIX bar code printer (includes software & drivers) Web pages must be designed to VGA resolution. All internal screens will be designed for SVGA resolution.	Same as candidate 2.	
Input Devices and Implications A description of input methods to be used, input devices (e.g., keyboard, mouse, etc.), special input requirements (e.g., new or revised forms from which data would be input), and input considerations (e.g., timing of actual inputs).	Keyboard & mouse	Apple "Quick Take" digital camera and software (15) PSC Quickscan laser bar code scanners (1) HP Scanjet 4C flatbed scanner Keyboard & mouse	Same as candidate 2.	
Storage Devices and Implications Brief descriptions of what data would be stored, what data would be accessed from existing stores, what storage media would be used, how much storage capacity would be needed, and how data would be organized.	MS SQL Server DBMS with 100GB arrayed capability.	Same as candidate 1.	Same as candidate 1.	

FIGURE 11-8 Feasibility Analysis Matrix Template

	Weighting	Candidate 1	Candidate 2	Candidate 3
Description				
Operational feasibility				
Cultural feasibility				
Technical feasibility				
Economic feasibility				
Schedule feasibility				
Legal feasibility				
Weighted score				

overview characteristics concerning the portion of the system to be computerized, the business benefits, and the software tools and/or applications needed. Subsequent pages would provide additional details concerning other characteristics such as those mentioned previously. Two columns can be similar except for their entries in one or two cells. Multiple pages would be used if we were considering more than three candidates. A simple word processing "table" template can be duplicated to create a candidate systems matrix.

> Feasibility Analysis Matrix

The second matrix complements the candidate systems matrix with an analysis and ranking of the candidate systems. It is called a **feasibility analysis matrix.**

feasibility analysis matrix a tool used to rank candidate systems.

The columns of the matrix correspond to the same candidate solutions as shown in the candidate systems matrix. Some rows correspond to the feasibility criteria presented in this chapter. Rows are added to describe the general solution and a ranking of the candidates. The general format is shown in Figure 11-8.

The cells contain the feasibility assessment notes for each candidate. Each row can be assigned a rank or score for each criterion (for operational feasibility, candidates can be ranked 1, 2, 3, etc.). After ranking or scoring all candidates on each criterion, a final ranking or score is recorded in the last row. Not all feasibility criteria are necessarily equal in importance; consequently, before assigning final rankings, candidates for which any criterion is deemed infeasible can be eliminated. In reality, this doesn't happen very often.

A completed feasibility analysis matrix is presented in Figure 11-9. In the figure, the feasibility assessment is provided for each candidate solution. In this example, a score is recorded directly in the cell for each candidate's feasibility criteria assessment. The weightings allow you to quantify the analysis. But be aware that any solution that is completely infeasible on any criteria should be eliminated. For instance, a solution that could be implemented only by violating contracts with suppliers could not be considered.

FIGURE 11-9 Sample Feasibility Analysis Matrix

	Wt	Candidate 1	Candidate 2	Candidate 3
Description		Purchase commercial off-the-shelf package for member services.	Write new application in-house using new company standard VB. NET and SQL Server database	Rewrite current in-house application using Powerbuilder.
Operational feasibility	15%	Supports only Member Services requirements. Current business process would have to be modified to take advantage of software functionality. Also, there is concern about security in the system. **Score: 60**	Fully supports user-required functionality. **Score: 100**	Fully supports user-required functionality. **Score: 100**
Cultural feasibility	15%	Possible user resistance to nonstandard user interface of proposed purchased package. **Score: 70**	No foreseeable problems **Score: 100**	No foreseeable problems **Score: 100**
Technical feasibility	20%	Current production release of Platinum Plus package is version 1.0 and has been on the market for only 6 weeks. Maturity of product is a risk, and company charges and additional monthly fee for technical support. Required to hire or train Java J2EE expertise to perform modifications for integration requirements. **Score: 50**	Solution requires writing application in VB. NET. Although current technical staff has only Powerbuilder experience, it should be relatively easy to find programmers with VB. NET experience. **Score: 95**	Although current technical staff is comfortable with Powerbuilder, management is concerned about acquisition of Powerbuilder by Sybase Inc. MS SQL Server is the current company standard for database, which competes with Sybase DBMS. We have no guarantee that future versions of Powerbuilder will "play well" with our current version of SQL Server. **Score: 60**
Economic feasibility	30%			
Cost to develop:		Approx. $350,000	Approx. $418,000	Approx. $400,000
Payback (discounted):		Approx. 4.5 years	Approx. 3.5 years	Approx. 3.3 years
Net present value:		Approx. $210,000	Approx. $307,000	Approx. $325,000
Detailed calculations:		See Attachment A **Score: 60**	See Attachment A **Score: 85**	See Attachment A **Score: 90**
Schedule feasibility	10%	Less than 3 months **Score: 95**	9–12 months **Score: 80**	9 months **Score: 85**
Legal feasibility	10%	No foreseeable problems **Score: 100**	No foreseeable problems **Score: 100**	No foreseeable problems **Score: 100**
Weighted score	100%	67	92.5	87.5

The System Proposal

Recall from Chapter 5 that the decision analysis phase involves identifying candidate solutions, analyzing those solutions, comparing and then selecting the best overall solution, and then recommending a solution. We've just learned how to do the first three tasks. Let's now learn about recommending a solution.

Recommending a solution involves producing a **system proposal.** This deliverable is usually a formal written report or oral presentation intended for system owners and users. Therefore, the systems analysts should be able to write a formal business report and make a business presentation without getting into technical issues or alternatives. Let's survey some important concepts of written reports and presentations.

system proposal a report or presentation of a recommended solution.

> Written Report

The written report is the most abused method used by analysts to communicate with system users. There is a tendency to generate large, voluminous reports that look impressive. Sometimes such reports are necessary, but often they are not. If a manager receives a 500-page technical report, the manager may skim it but not read it—and you can be certain it won't be studied carefully.

Length of the Written Report Trial and error has taught us about report size. The following are general guidelines on limiting report size:

- To executive-level managers—one or two pages.
- To middle-level managers—three to five pages.
- To supervisory-level managers—less than 10 pages.
- To clerk-level personnel—less than 50 pages.

It is possible to organize a larger report to include subreports for managers who are at different levels. These subreports are usually included as early sections in the report and summarize the report, focusing on the bottom line.

Organization of the Written Report There is a general pattern to organizing any report. Every report consists of both primary and secondary elements. *Primary elements* present the actual information that the report is intended to convey. Examples include the introduction and the conclusion.

While the primary elements present the actual information, all reports also contain secondary elements. *Secondary elements* package the report so that the reader can easily identify the report and its primary elements. Secondary elements also add a professional polish to the report.

As indicated in Figure 11-10, the primary elements can be organized in one of two formats: factual and administrative. The *factual format* is traditional and best suited to

FIGURE 11-10 Formats For Written Reports

Factual Format	Administrative Format
I. Introduction	I. Introduction
II. Methods and procedures	II. Conclusions and recommendations
III. Facts and details	III. Summary and discussion of facts and details
IV. Discussion and analysis of facts and details	IV. Methods and procedures
V. Recommendations	V. Final conclusion
VI. Conclusion	VI. Appendixes with facts and details

FIGURE 11-11 Secondary Elements for a Written Report

Letter of transmittal
Title page
Table of contents
List of figures, illustrations, and tables
Abstract or executive summary
(The primary elements—the body of the report in either the factual or administrative format—are presented in this portion of the report.)
Appendixes

readers who are interested in facts and details as well as conclusions. This is the format we would use to specify detailed requirements and design specifications to system users. But the factual format is not appropriate for most managers and executives.

The *administrative format* is a modern, results-oriented format preferred by many managers and executives. This format is designed for readers who are interested in results, not facts. It presents conclusions or recommendations first. Any reader can read the report straight through, until the point at which the level of detail exceeds the reader's interest.

Both formats include some common elements. The *introduction* should include four components: purpose of the report, statement of the problem, scope of the project, and a narrative explanation of the contents of the report. The *methods and procedures section* should briefly explain how the information contained in the report was developed—for example, how the study was performed or how the new system will be designed. The bulk of the report will be in the *facts section*. This section should be named to describe the type of factual data presented (e.g., "Existing Systems Description," "Analysis of Alternative Solutions," or "Design Specifications"). The *conclusion* should briefly summarize the report, verifying the problem statement, findings, and recommendations.

Figure 11-11 shows the secondary, or packaging, elements of the report and their relationship to the primary elements. Many of these elements are self-explanatory. We briefly discuss here those that may not be. No report should be distributed without a *letter of transmittal* to the recipient. This letter should be clearly visible, not inside the cover of the report. A letter of transmittal states what type of action is needed on the report. It can also call attention to any features of the project or report that deserve special attention. In addition, it is an appropriate place to acknowledge the help you've received from various people.

The *abstract* or *executive summary* is a one- or two-page summary of the entire report. It helps readers decide if the report contains information they need to know. It can also serve as the highest-level summary report. Virtually every manager reads these summaries. Most managers will read on, possibly skipping the detailed facts and appendixes.

Writing the Report Figure 11-12 illustrates the proper procedure for writing a formal report. Here are some guidelines to follow:

- *Paragraphs should convey a single idea.* They should flow nicely, one to the next. Poor paragraph structure can almost always be traced to outlining deficiencies.
- *Sentences should not be too complex.* The average sentence length should not exceed 20 words. Studies suggest that sentences longer than 20 words are difficult to read and understand.
- *Write in the active voice.* The passive voice becomes wordy and boring when used consistently.

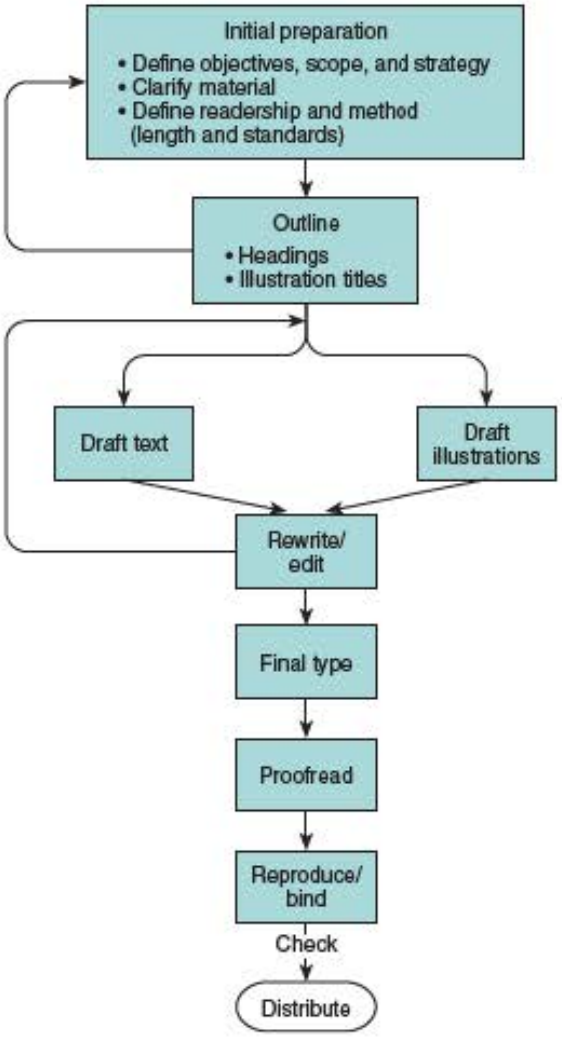


FIGURE 11-12
Steps in Writing a Report

Source: Copyright Keith London.

- *Eliminate jargon, big words, and deadwood.* For example, replace "DBMS" with "database management system," substitute "so" for "accordingly," try "useful" instead of "advantageous," and use "clearly" instead of "it is clear that."

Every businessperson should have a copy of *The Elements of Style* by William Strunk, Jr., and E. B. White. This classic paperback may set a record in value-to-cost ratio. Barely bigger than a pocket-size book, it is a gold mine of information.

> Formal Presentation

To communicate information to the many different people involved in a systems development project, a systems analyst is frequently required to make formal presentations. **Formal presentations** are special meetings used to sell new ideas and gain approval for new systems. They may also be used for any of these purposes: sell a new system, sell new ideas, sell change, head off criticism, address concerns, verify conclusions, clarify facts, and report progress. In many cases, a formal presentation may set up or supplement a more detailed written report.

formal presentation a special meeting used to sell new ideas and gain approval for new systems.

Effective and successful presentations require significant preparation. The time allotted to presentations is frequently brief; therefore, organization and format are critical issues. You cannot improvise and expect acceptance.

Presentations offer the advantage of impact through immediate feedback and spontaneous responses. The audience can respond to the presenter, who can use

FIGURE 11-13 Typical Outline and Time Allocation for an Oral Presentation

I. Introduction (one-sixth of total time available)
   A. Problem statement
   B. Work completed to date
II. Part of the presentation (two-thirds of total time available)
   A. Summary of existing problems and limitations
   B. Summary description of the proposed system
   C. Feasibility analysis
   D. Proposed schedule to complete project
III. Questions and concerns from the audience (Time here is not to be included in the time allotted for presentation and conclusion; it is determined by those asking the questions and voicing their concerns.)
IV. Conclusion (one-sixth of total time available)
   A. Summary of proposal
   B. Call to action (request for whatever authority you require to continue systems development)

emphasis, timed pauses, and body language to convey messages not possible with the written word. The disadvantage to presentations is that the material presented is easily forgotten because the words are spoken and the visual aids are transient. That's why presentations are often followed by a written report, either summarized or detailed.

Preparing for the Formal Presentation Presenters must know their audience. This is especially crucial when your presentation is trying to sell new ideas and a new system. The systems analyst is frequently thought of as the dreaded agent of change in an organization. As Machiavelli wrote in his classic book *The Prince*,

> There is nothing more difficult to carry out, nor more dangerous to handle, than to initiate a new order of things. For the reformer has enemies in all who profit by the old order, and only lukewarm defenders in all those who would profit from the new order, this lukewarmness arising partly from fear of their adversaries—and partly from the incredulity of mankind, who do not believe in anything new until they have had actual experience of it.[1]

People tend to be opposed to change. There is comfort in the familiar way things are today. Yet a substantial amount of the analyst's job is to bring about change—in methods, procedures, technology, and the like. A successful analyst must be an effective salesperson. It is entirely appropriate (and strongly recommended) for an analyst to formally study salesmanship. To effectively present and sell change, you must be confident in your ideas and have the facts to back them up. Again, preparation is the key!

First, define expectations of the presentation—for instance, that the goal is to seek approval to continue the project, that another goal is to confirm facts, and so forth. A presentation is a summary of ideas and proposals that is directed toward the presenter's expectations.

Executives are usually put off by excessive detail. To avoid this, a presentation should be carefully organized around the allotted time (usually 30 to 60 minutes). Although each presentation differs, the organization and time allocation suggested in Figure 11-13 provide an idea of how this works. This figure illustrates some typical

[1]Niccolo Machiavelli, *The Prince and Discourses*, trans. Luigi Ricci (New York: Random House, 1940, 1950). Reprinted by permission of Oxford University Press.

topics of an oral presentation and the amount of time to allow for each. Note that this particular outline is for a systems analysis presentation. Other types of presentations might be slightly different.

What else can you do to prepare for the presentation? Because of the limited time, use visual aids—predrawn flipcharts, overhead slides, Microsoft PowerPoint slides, and the like—to support your position. Just like a written paragraph, each visual aid should convey a single idea. When preparing pictures or words, use the guidelines shown in Figure 11-14.

Microsoft PowerPoint contains software guides called *wizards* to assist the most novice users with creating professional-looking presentations. The wizard steps the user through the development process by asking a series of questions and tailoring the presentation based on responses. To hold your audience's attention, consider distributing photocopies of the visual aids at the start of the presentation. This way, the audience doesn't have to take as many notes.

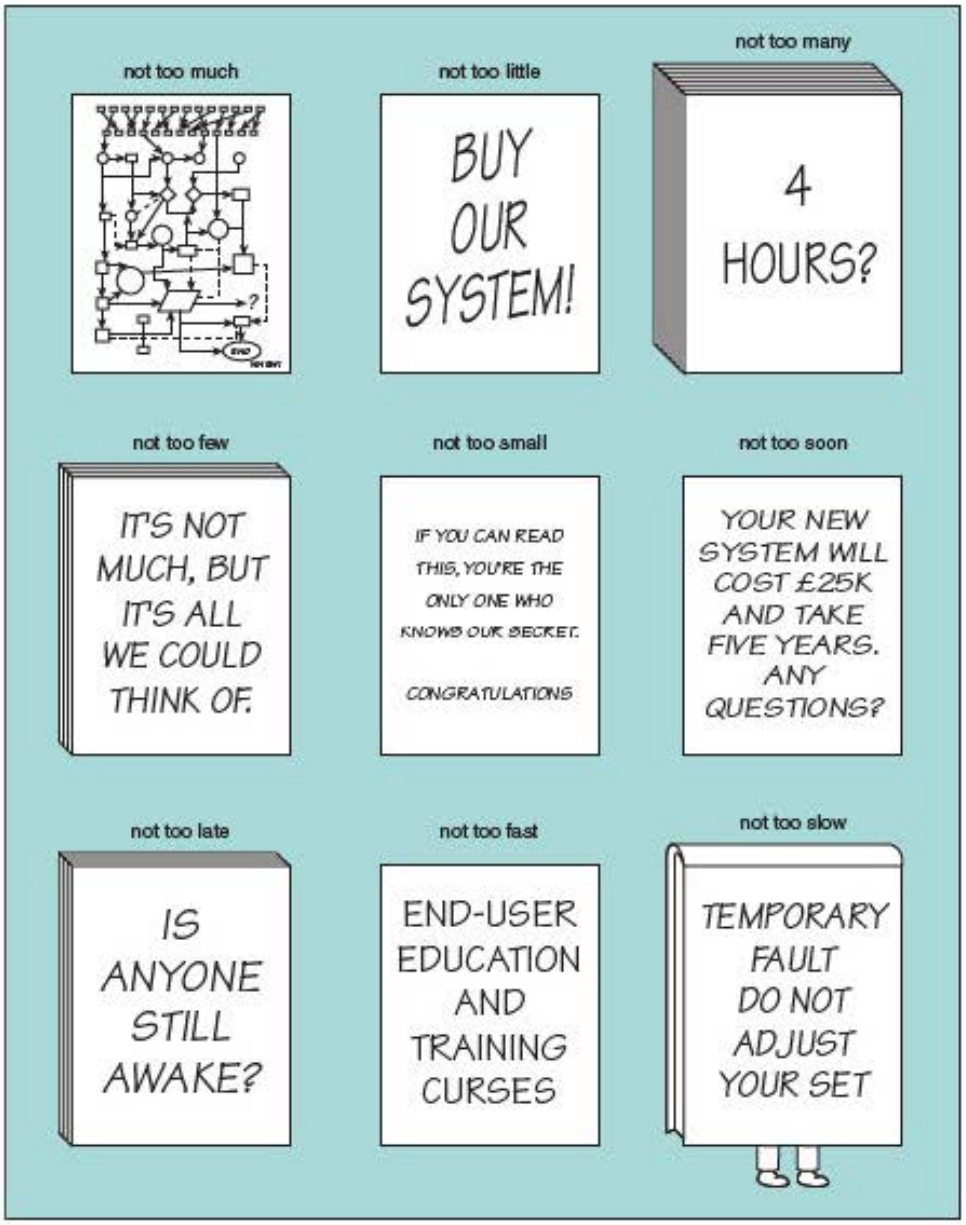


FIGURE 11-14
Guidelines for Visual Aids

Source: Copyright Keith London.

Finally, practice the presentation in front of the most critical audience you can assemble. Play your own devil's advocate, or, better yet, get somebody else to raise criticisms and objections. Practice your responses to these issues.

Conducting the Formal Presentation A few additional guidelines may improve the actual presentation:

- *Dress professionally.* The way you dress influences people. John T. Malloy's books, *Dress for Success* and *The Woman's Dress for Success Book*, are excellent reading for both wardrobe advice and the results of studies regarding the effects of clothing on management.
- *Avoid using the word "I" when making the presentation.* Use "you" and "we" to assign ownership of the proposed system to management.
- *Maintain eye contact with the group and keep an air of confidence.* If you don't show management that you believe in your proposal, why should management believe in it?
- *Be aware of your own mannerisms.* Some of the most common mannerisms include using too many hand gestures, pacing, and repeatedly saying "you know" or "OK." Although mannerisms alone don't contradict the message, they can distract the audience.

Sometimes while you are making a presentation, some members of the audience may not be listening. This lack of attention may take several forms. Some people may be engaged in competing conversations, some may be daydreaming, some may be busy glancing at their watches, some who are listening may have puzzled expressions, and some may show no expression. The following suggestions may prove useful for keeping people listening:

- *Stop talking.* The silence can be deafening. The best public speakers know how to use dramatic pauses for special emphasis.
- *Ask a question, and let someone in the audience answer it.* This involves the audience in the presentation and is a very effective way of stopping a competing conversation.
- *Try a little humor.* You don't have to be a talented comedian. But everybody likes to laugh. Tell a joke on yourself.
- *Use props.* Use some type of visual aid to make your point clearer. Draw on the chalkboard, illustrate on the back of your notes, or create a physical model to make the message easier to understand.
- *Change your voice level.* By making your voice louder or softer, you force the audience to listen more closely or make it easier for the audience to hear. Either way, you've made a change from what the audience was used to, and that is the best way to get and hold attention.
- *Do something unexpected.* Drop a book; toss your notes; jingle your keys. Doing the unexpected is almost always an attention grabber.

A formal presentation will usually include time for questions from the audience. This time is very important because it allows you to clarify any points that were unclear and draw additional emphasis to important ideas. It also allows the audience to interact with you. However, sometimes answering questions after a presentation may be difficult and frustrating. We suggest the following guidelines when answering questions:

- *Always answer a question seriously, even if you think it is a silly question.* Remember, if you make someone feel stupid for asking a "dumb" question, that person will be offended. Also, other members of the audience won't ask their questions for fear of the same treatment.
- *Answer both the individual who asked the question and the entire audience.* If you direct all your attention to the person who asked the question,

the rest of the audience will be bored. If you don't direct enough attention to the person who asked the question, that person won't be satisfied. Try to achieve a balance. If the question is not of general interest to the audience, answer it later with that specific person.

- *Summarize your answers.* Be specific enough to answer the question, but don't get bogged down in details.
- *Limit the amount of time you spend answering any one question.* If additional time is needed, wait until after the presentation is over.
- *Be honest.* If you don't know the answer to a question, admit it. Never try to bluff your way out of a question. The audience will eventually find out, and you will destroy your credibility. Instead, promise to find out and report back. Or ask someone in the audience to do some research and present the findings later.

Following Up the Formal Presentation As mentioned earlier, it is extremely important to follow up a formal presentation because the spoken word and impressive visual aids used in a presentation often do not leave a lasting impression. For this reason, most presentations are followed by written reports that provide the audience with a more permanent copy of the information that was communicated.

Chapter Review

1. Feasibility is a measure of how beneficial the development of an information system would be to an organization. Feasibility analysis is the process by which we measure feasibility. It is an ongoing evaluation of feasibility at various checkpoints in the life cycle. At any of these checkpoints, the project may be canceled, revised, or continued. This is called a creeping commitment approach to feasibility.
2. There are six feasibility tests: operational, cultural/political, technical, schedule, economic, and legal.
   a. Operational feasibility is a measure of problem urgency or solution acceptability. It includes a measure of how the end users and managers feel about the problems or solutions.
   b. Cultural (or political) feasibility is a measure of how people feel about a solution and how well it will be accepted.
   c. Technical feasibility is a measure of how practical solutions are and whether the technology is already available within the organization. If the technology is not available to the firm, technical feasibility also looks at whether it can be acquired.
   d. Schedule feasibility is a measure of how reasonable the project schedule or deadline is.
   e. Economic feasibility is a measure of whether a solution will pay for itself or how profitable a solution will be. For management, economic feasibility is the most important of our four measures.
   f. Legal feasibility is a measure of how well a solution can be implemented within existing legal and contractual obligations.
3. To analyze economic feasibility, you itemize benefits and costs. Benefits are either tangible (easy to measure) or intangible (hard to measure). To properly analyze economic feasibility, try to estimate the value of all benefits. Costs fall into two categories: development and operating.
   a. Development costs are onetime costs associated with analysis, design, and implementation of the system.
   b. Operating costs may be fixed over time or variable with respect to system usage.
4. Given the costs and benefits, economic feasibility is evaluated by the techniques of cost-benefit analysis. Cost-benefit analysis determines if a project or solution will be cost-effective—if lifetime benefits will exceed lifetime costs. There are three popular ways to measure cost-effectiveness: payback analysis, return-on-investment analysis, and net present value analysis.
   a. Payback analysis defines how long it will take for a system to pay for itself.
   b. Return-on-investment and net present value analyses determine the profitability of a system.

c. Net present value analysis is preferred because it can compare alternatives with different lifetimes.

5. A candidate systems matrix is a useful tool for documenting the similarities and differences between candidate systems being considered.
6. A feasibility analysis matrix is used to evaluate and rank candidate systems. Both the candidate systems matrix and the feasibility analysis matrix are useful for presenting the results of a feasibility analysis as part of a system proposal.
7. Written reports are the most common communications vehicle used by analysts. Reports consist of both primary and secondary elements. Primary elements contain factual information. Secondary elements package the report for ease of use. Reports may be organized in either the factual or administrative format. The factual format presents the details before conclusions; the administrative format reverses that order. Managers like the administrative format because it is results-oriented and gets right to the bottom-line question.
8. Formal presentations are a special type of meeting at which a person presents conclusions, ideas, or proposals to an interested audience. Preparation is the key to effective presentations.
9. The system proposal may be a formal written report or an oral presentation.

Review Questions

1. What does a creeping commitment approach to feasibility analysis mean?
2. What are the feasibility analysis checkpoints in the development cycle? What should be done at each checkpoint?
3. What are the objectives of the operational feasibility test?
4. Why is it important to find out how the end users and managers feel about the problem solution that the system analyst has identified?
5. When is usability analysis performed? What is the objective of the usability analysis?
6. What is the objective of the technical feasibility test?
7. What are the characteristics of development costs and operating costs? List three examples of each kind of cost.
8. List five examples of tangible benefits.
9. Why is the time-value-of-money concept an essential consideration when accessing economic feasibility?
10. What are the most commonly used techniques to determine the cost-effectiveness of a project?
11. For what are the candidate systems matrix and feasibility analysis matrix used?
12. For written reports, what is the difference between the factual format and the administrative format?
13. What are the steps in writing a report?
14. What are the advantages and disadvantages of presentations?
15. What should be done to follow up the formal presentation?

Problems and Exercises

1. The textbook describes a creeping commitment approach to feasibility.
   a. Explain this approach and why the textbook recommends it.
   b. What are the some of the changes or events that might occur which make this approach advisable?
   c. Should an organization cancel a project if it becomes infeasible?
2. The textbook describes three checkpoints for measuring feasibility.
   a. What are these checkpoints?
   b. Typically, how accurately can feasibility be determined at each checkpoint?
   c. Which checkpoint, if any, is the most critical one?
3. What are the four categories of feasibility tests, and what is the criteria each of them uses to measure feasibility?
4. You are a systems designer on a project which is getting close to finishing the systems design phase. A working prototype has been developed, and you've been tasked with doing a usability analysis. Draft a one- or two-page plan detailing your approach to conducting the usability analysis.
5. You are a systems analyst working in the IT shop of a medium-size organization with about 300 employees. The organization is in the system design phase of a project to develop an

electronic activity reporting system for all employees, replacing the current hard copy method. All of the work is being done in-house except for several consultants, who are providing ancillary services, such as IV&V. The application will use employees' existing desktops, although several dedicated servers will need to be acquired. The user interface is very intuitive, but the project calls for about a half day of training for all employees on policies and procedures for using the new application. The system is not using any new technology, and the IT technical staff have a great deal of expertise. Create a worksheet, detailing the estimated one-time development costs and ongoing operating costs. By the way, in your organization, salary and benefits for systems analysts average $40 per hour; you can use this as a basis for estimating salary and benefits for other classifications involved in the project.

6. In the project described above, it was noted that the electronic activity reporting system will be replacing the current manual system. Describe the tangible benefits that might be expected. Take a "best guess" approach, and calculate the annual savings to the organization. Show your assumptions in the calculations.
7. You are designing a Web-based system where your regional offices can submit their sales reports online instead of filling them out by hand and mailing them in. Three candidate solutions have been identified. Their estimated lifetime benefits and estimated lifetime costs are shown below. All have been time-adjusted over the projected five-year lifetime of each alternative.

	Estimated Lifetime Benefits	Estimated Lifetime Costs
Candidate Solution 1:	$640,000	$172,000
Candidate Solution 2:	$640,000	$160,000
Candidate Solution 3:	$640,000	$185,000

According to return-on-investment analysis, which candidate solution offers the highest ROI? If the organization sets a minimum lifetime ROI of 80 percent, which of these solutions is economically feasible?

8. What are the different techniques or methods for identifying candidate solutions? If you had to choose just one of these methods, which would it be and why?
9. You are working as a system designer for a company that manufactures heavy-duty power tools used by contractors. Every month, your regional sales and service centers batch together the hard copy repair orders for work performed under warranty. They are sent to headquarters, where they are run through a legacy mainframe batch process. A report is then generated, which the engineers analyze for signs of any problem trends in the new models. The company's CEO has decided that this process is far too slow in today's highly competitive business environment and wants to replace the legacy system as soon as possible with something more contemporary. Identify at least three candidate solutions, and describe them in a candidate systems matrix, using Figure 11-7 as an example.
10. Prepare a feasibility analysis matrix, using the candidate solutions you identified and described in the preceding question. Use Figure 11-9 as your template, but choose the weighting factors that you feel would be most appropriate in this situation. For purposes of this exercise, you may provide an estimate of the economic feasibility.
11. Once the feasibility analysis matrix has been completed, it is time to write the feasibility report. For this exercise, prepare a feasibility report to executive-level managers, using the appropriate format shown in Figure 11-10.
12. You have been asked to present the feasibility analysis and recommendation to the executive managers of every department in your organization at their weekly meeting. Prepare a set of PowerPoint slides to be used as a visual aid during your presentation.
13. Name at least 10 things you should *not* do if you want your presentation to be informative, persuasive, and well-received.

Projects and Research

1. Steve McConnell is an author who has written numerous books on software engineering and development. In his book *Rapid Development*, McConnell points out that in engineering, design is usually a much smaller part of the total project than the actual construction. He compares bridge building projects, where design is about 10 percent of the total effort and construction about 90 percent, to software development projects, where design is generally at least 50 percent of the total

project effort. Explore and expand on this theme of the unique differences in software engineering compared to other types of engineering, and summarize your analysis and findings in a one- to two-page paper. What do other software engineering leaders have to say on this topic?

2. You work as a system analyst in the headquarters of your state's highway patrol, which has field offices throughout the state. Currently, traffic accident reports are handwritten in the field by the highway patrol officers, reviewed by their sergeant, stored temporarily, then batched and sent monthly to headquarters. Each one is entered into a legacy mainframe system by key data operators, then after the reports from the patrol offices in each county have been input, a computer operator runs the edit program using JCL.

   Reports with major errors or omissions are rejected and returned to the county highway patrol office of origin for correction. After the edit program is completed for all the counties, an update program is run adding the monthly batch of traffic accident reports to the master file of reports. Statistical reports are generated quarterly and yearly. The entire process from the time the batches of reports are received to the point the master file is updated generally takes about three months. Executive management is interested in replacing the system with something that is more modern, less labor-intensive, more accurate, and easier for users to access and that will reduce turnaround time for preparing statistical reports. Your assignment, as a member of the project team, is to prepare the feasibility study report (FSR).

   a. What are some of the options or alternatives that you think should be considered? (Identify at least three in addition to "do nothing" or "maintain the status quo").
   b. Prepare a candidate systems matrix describing the characteristics of each of these alternatives, using the candidate systems matrix template shown in Figure 11-6.
   c. Expand the candidate systems matrix, using the template shown in Figure 11-7.
   d. Evaluate each of these alternatives for operational, technical, and schedule feasibility, using the techniques described in the textbook and using the template shown in Figure 11-9.

3. Based upon the scenario described in the preceding question:

   a. Prepare an estimated-costs worksheet for each alternative, using the format shown in Figure 11-2.
   b. Assess the economic feasibility of each alternative, using one of the three techniques described in the textbook. Which technique did you use and why?
   c. Add the economic feasibility analysis to the feasibility analysis matrix from the preceding question.
   d. Compare and score each of these alternatives. Use different weighting factors for each of the feasibility criteria than those used in the textbook.
   e. What weighting factors did you choose for the different criteria in your feasibility analysis matrix? Why?

4. Management was impressed by your excellent work on the feasibility analysis matrix and has asked you to prepare the system proposal report. Write a system proposal whose primary audience will be the midlevel business and IT managers, but which also will be read by the executive sponsor and chief information officer. Use the appropriate format shown in Figure 11-10.
5. Midlevel management was extremely impressed by your system proposal. They now want you to prepare and present a formal presentation to the top management of the department.

   a. Describe the steps you should go through to prepare for the formal presentation.
   b. Prepare a PowerPoint slide presentation, using the guidelines suggested in the textbook, or in other books and articles on the do's and don'ts of PowerPoint presentations.
   c. What do you consider to be the most critical thing to know in preparing for the formal presentation? Why

6. A wide variety of formats, templates, and methods exist for preparing system proposals and feasibility study reports. Search the Web to see what other tools and techniques you can find.

   a. Describe the formats that you found and their sources.
   b. Compare the different formats that you found to each other and to the one in the textbook. What are some of the differences?
   c. Do you think there is one format with clear-cut advantages over the others? If so, describe which one, and why you feel it is better.
   d. Create what you believe to be the ideal FSR template for your organization.

Minicases

The grocery store, Wow Munchies, from an earlier chapter is considering developing an online site for customers to purchase food. The owner of the store believes that this capability will enable the store to grab market share from nearby Fast Food Co., which has a Web site and delivers food to the customer. This site will allow customers to purchase any item that is currently in stock in the store. The store will *not* deliver the food, but will have the food bagged and ready for pickup at the time designated by the customer. Wow Munchies has a single storefront.

1. Conduct an operational feasibility study. Do you think the Web site will enable Wow Munchies to gain market share, as is its purpose? What factors will affect the operational success of this site? Submit your paper, supporting documents, charts, and any interviews you conducted.
2. Conduct a technical feasibility study. What would you recommend the company use in the creation and maintenance of their site (e.g. languages, specific host, encryption)? Why does your choice affect the feasibility of said site? Submit your paper, supporting documents, charts, and any interviews you conducted.
3. Conduct an economic feasibility study for the investment into an e-commerce site. What discount rate are you using? Why? Submit your paper, supporting documents, charts, and any interviews you conducted.
4. Develop a timeline and schedule feasibility study for completion of this Web site. Do you see any mitigating factors that might cause a delay in the timeline or deadline overrun? Submit both a short paper and a Gantt chart.

Team and Individual Exercises

1. Roundtable discussion: ROI analyses are often done with a consideration of a technology "lifetime" of 3–5 years. Do you think that technology has an impact on business after that time period? Why or why not?
2. Individual: In the last chapter, you discussed (in a roundtable format) the importance of knowledge and information on economic success. Every year, grants for college go unawarded because students do not apply for them (possibly because they do not know they exist). Research college grants available to you, and apply for at least one.
3. Team/class discussion: What does it mean to have an attitude for success? Do you think that people's belief in their own capabilities can influence their actual success? On the flip side, do you think that people's lack of belief in themselves will impact their ability to be successful? What can you do to further develop/enhance your own attitude for success?

Suggested Readings

Bovee, Courtland L., and John V. Thill. *Business Communications Today,* 2nd ed. New York: Random House, 1989.

Gildersleeve, Thomas R. *Successful Data Processing Systems Analysis,* 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 1985. This book provides an excellent chapter on cost-benefit analysis techniques. Chapter 5 discusses presentations. We are indebted to Gildersleeve for the creeping commitment concept.

Gore, Marvin, and John Stubbe. *Elements of Systems Analysis,* 4th ed. Dubuque, IA: Brown, 1988. The feasibility analysis chapter suggests an interesting matrix approach to identifying, cataloging, and analyzing the feasibility of alternative solutions for a system.

Smith, Randi Sigmund. *Written Communications for Data Processing.* New York: Van Nostrand Publishing, 1976.

Stuart, Ann. *Writing and Analyzing Effective Computer System Documentation.* New York: Holt, Rinehart and Winston, 1984.

Uris, Auren. *The Executive Deskbook,* 3rd ed. New York: Van Nostrand Reinhold, 1988.

Walton, Donald. *Are You Communicating? You Can't Manage without It.* New York: McGraw-Hill, 1989.

Wetherbe, James. *Systems Analysis and Design: Traditional, Structured, and Advanced Concepts and Techniques,* 2nd ed. St. Paul, MN: West, 1984. Wetherbe pioneered the PIECES framework for problem classification. In this chapter we extended that framework to analyze operational feasibility of solutions.