C H A P T E R 1

The Nature of Information Technology Projects

CHAPTER OBJECTIVES

Chapter 1 provides an overview of information technology project management (ITPM). After studying this chapter, you should understand and be able to:

- Describe the software crisis and how the often dismal track record for information technology (IT) projects provides a motivation for changing how we view and manage IT projects.
- Explain the socio-technical, project management, and knowledge management approaches that support ITPM.
- Define what an IT project is and describe its attributes.
- Define the discipline called project management.
- Describe the role and impact IT projects have on an organization.
- Identify the different roles and interests of project stakeholders.
- Describe the project life cycle, the systems development life cycle, and their relationship.
- Identify the Project Management Body of Knowledge (PMBOK) and its core knowledge areas.

GLOBAL TECHNOLOGY SOLUTIONS

Tim Williams placed the phone gently back in its cradle. He sat for a moment, not sure whether he felt excitement or sheer terror. Or, could it be he was feeling both? Kellie Matthews, his partner in Global Technology Solutions (GTS), had just told Tim that Husky Air, a business air charter company, was very interested in having them develop an information system. This was the moment Tim had been waiting for—

their first client! Before Husky Air will sign a contract, however, they need to know what GTS will deliver, how much it will cost, and when the project will be completed.

As the project's manager, Tim knows that getting this contract is important. Husky Air would be the company's first and, so far, only client. Tim also understands that a successful project could lead to other work with Husky Air. Moreover, a verbal or written recommendation would provide additional credibility to help GTS get its foot in the door with other potential clients.

While working together in the information services department at a large company, Tim and Kellie decided that a small, independent consulting firm could be successful developing smaller IT-based systems. The lure of being their own bosses and the potential for financial and personal rewards were too great to resist. Tim and Kellie cashed in their stock options, and GTS was born. They decided that Kellie would develop new business and manage the day-to-day operations of GTS, while Tim would deliver and manage the projects. New employees with specific skill sets would be hired as needed to support particular projects.

Although both Tim and Kellie had worked in IT for several years, neither of them had ever managed a consulting project before. Aside from the questions posed by Husky Air (What will you deliver? How much will it cost? How long will it take?), Tim felt a bit overwhelmed because he knew the success or failure of this project would have an immediate impact on the viability of the new firm.

Things to Think About:

1. If you were in Tim's shoes, what feelings do you think you would experi ence?
2. What questions would you have?
3. What might help reduce your anxiety and uncertainty as an inexperienced project manager?
4. Where do you begin a new project?

INTRODUCTION

The new millennium provides a vantage point from which to look back at our past and ahead to our future. Many people at the end of the century emptied their bank accounts, stockpiled food and water, and even went so far as to head for the hills for fear that computers would crash and civilization would fall into mass confusion. Fortunately, the reported Y2K computer-related problems were few and not especially critical. Was the problem hyped? Not really. Was there really a problem? Yes! Many companies spent millions of dollars to change and test the dates in their computer systems so the passing of January 1, 2000 would have no effect. Just ask one of the many IT professionals who worked hard and long on a Y2K project. You may even know, or be, one of those people.

What made the Y2K problem fascinating was that just about everyone was in this together and the project had an immovable deadline. As a result, the field of information technology received a great deal of attention in the media and the boardroom. Even though it was shortsightedness that created the problem, the few reported Y2K problems prompted some to believe this was the IT profession's shining hour. Moreover, the risks and costs associated with the Y2K problem captured the attention of senior management. As a result, people at different levels, including senior management, and in different functional areas became more involved with and interested in information technology. The good news is that the world of IT moved from the back office to the boardroom. The bad news is that IT may be doomed to repeat past mistakes.

After Y2K, it appeared that companies now had the time and money to start on their IT projects that had been on hold. Electronic commerce and the integration of enterprise resource planning (ERP) packages were at the top of the IT project list for many organizations. The demand for skilled IT professionals and project managers to head up these new initiatives had never been stronger. It seemed as though recruiters couldn't hire experienced professionals and university graduates fast enough to meet the demand.

Unfortunately, this golden time for IT did not last. The tragic events of September 2001 had a profound impact on the world and the global economy. As a result, many organizations were forced to make some difficult choices in order to survive. Seasoned professionals and new graduates who once commanded high salaries and choice assignments found themselves facing a tough job market. The bubble had burst. If nothing else, we learned that things can change quickly and without warning.

As you read this, think about what is going on in the field of IT right now. Is the demand for IT professionals and IT projects strong? Or, are there fewer jobs and projects available? If the demand for IT projects and professionals to work on these projects is strong, many organizations will probably have to choose from among projects that have been sitting on the backburner for some time. On the other hand, if time, money, and resources for many organizations are limited, then only a few, select IT projects can be funded.

In both good times and bad, senior management will make a certain level of funding available for IT projects. The budgeted amount will depend upon such things as the economy, competitor's actions within the industry, and the organization's strategic plan. Regardless whether an organization's budget for IT projects shrinks or grows, the resources available for any given period will be relatively fixed. Quite often the total funding requirements for the proposed projects will be greater than the available budget. As a result, any project that receives funding will do so at the expense of another project. The competition for funding IT projects proposed by the various business units within an organization will be especially keen when the budget is tight. Projects that do not receive any funding will either have to wait or fall by the wayside. Therefore, the decision to fund specific projects will always be an important management decision because it will have a major impact on the organization's performance.

The decision to fund or invest in an IT project should be based on the value that the completed project will provide the organization. Otherwise, what is the point of spending all that time, effort, and money? Although senior management must make the difficult decision as to which IT projects receive funding and which ones do not, others must plan and carry out the project work. Which situation is worse: Successfully building and implementing an information system that provides little or no value to the organization, or implementing an information system that *could have* provided value to the organization, but was developed or managed poorly? It is probably moot. In either situation everyone with a direct or indirect interest in the project's outcome loses.

How This Book Is Organized

The goal of this book is to help you to plan and manage information technology projects.

We will focus on a number of different theories, but the main focus will be on applying the methods, tools, techniques, and processes for planning and managing an IT project from start to finish. If you are a project manager (or will be one soon), this book will help you to understand and apply project management principles in order to better manage your IT project. If you are just starting out in the field, this book will help you to understand the big picture of what an IT project is all about. This knowl-edge will help you to become a better team member and prepare you for the next sev-eral progressions in your career.

Many of the principles of project management can be applied to just about any project, but IT projects are unique in several ways. Throughout the text, we will dis-cuss what makes IT projects different from other types of projects and how the prin-ciples and methods of system development can be integrated to define the IT project management discipline. Although many of the concepts for developing an informa-tion system will be integrated throughout, this is not a systems analysis and design text. More specifically, we will not delve too deeply into the systems analysis and design techniques that are used during systems development. We will leave that for other books and classes.

The remainder of this chapter provides a foundation for project initiation by pro-viding an understanding of the nature of information technology projects. Before get-ting too involved with definitions and concepts, however, it is important to understand the motivation behind IT project management. In the next section we will focus on the software crisis, which for many people has become a call to arms for more effective management of IT projects. Then, we will introduce and define projects and project management. Subsequently, we will look at the relationship between the project life cycle and the systems development life cycle. At the end of the chapter, you will be introduced to the nine areas that make up the Project Management Body of Knowledge (PMBOK) that will be integrated throughout the remaining chapters of this text.

THE SOFTWARE CRISIS

Although IT is becoming more reliable, faster, and less expensive, the costs, complex-ities, and risks of IT projects continue to increase. In 1995, a consulting firm called The Standish Group conducted a survey of 365 IT managers. The widely cited report, appropriately called *CHAOS,* was startling.

For example, although the United States spent over $250 billion each year on IT application development projects, 31 percent of these projects were canceled before completion. Almost 53 percent were completed, but they were over-budget and over-schedule and did not meet the original specifications. The average cost overrun for a medium-size company surveyed was 182 percent of the original estimate, while the average schedule overrun was 202 percent. That is, the results of the survey, summarized in Table 1.1, suggest that a medium-sized project originally estimated to cost about $1 million and to take a year to develop, actually cost $1,820,000, tookjust over two years to complete, and only included 65 percent of the envisioned features and functions! Sadly, 48 percent of the IT managers surveyed believed there were more failures at the time than five and ten years earlier.

Why IT Projects Fail

The *CHAOS* report also provides some interesting insight as to why some projects succeed while others fail. According to the survey, user involvement, executive

Table 1.1 Summary of the CHAOS Study Results

Company Size	*Average Cost of Develop-ment*	*Average Cost Overruns*	*Average Schedule Overrun*	*Original Features and Functions Included*	*Successful Projects* [a]	*Challenged Projects* [b]	*Impaired Projects* [c]
Large	$2,322,000	178%	230%	42%	9%	61.5%	29.5%
Medium	$1,331,000	182%	202%	65%	16.2%	46.7%	37.1%
Small	$ 434,000	214%	239%	74%	28%	50.4%	21.6%

[a] Completed on-time and on-budget

[b] Completed, but over-budget, over schedule, and includes fewer features and functions than originally envisioned

[c] Cancelled before completion

SOURCE: Adapted from The Standish Group, *CHAOS* (West Yarmouth, MA: 1995), http://www.standishgroup .com /visitor/chaos.htm.

management support, and a clear statement of requirements ranked at the top of the list of factors essential for IT project success. On the other hand, lack of user involvement and incomplete requirements appear to be the two main factors for projects being challenged or canceled before completion.

Tables 1.1 and 1.2 summarize some of the key findings of the *CHAOS* report. First, larger projects report a success rate of only 9 percent and appear to be much more risky than medium and smaller projects. Technology, business models, and cycle times are changing too quickly to develop systems that take much more than a year to complete. This data also supports the need to break up large projects into smaller, more manageable ones that can be completed in less than a year. Companies such as Sears, Roebuck and Co., for example, have new, stricter IT project deadlines that require all web-based projects be completed within three months (Hoffman and King 2000).

In addition, one can look at the project factors for successful and not-so-successful projects to see what may be happening on those projects. User involvement leads the list as the most important factor in project success. This should come as no surprise since the client's expertise is needed to identify problems and opportunities and to define requirements. Moreover, active participation by the client keeps them interested in and excited about the project. Individuals will also begin to take ownership of a project if they feel that they have a stake in the project's success or failure. Effective communication between the techies and non-techies allows for a clearer definition of the project's goals and requirements. Working together, developers and users have more realistic expectations because they themselves set those expectations together. Management is then more compelled to support a popular project.

On the other hand, lack of user input or involvement ranks at or near the top in factors affecting challenged and impaired projects. One can almost picture the chain of events. Without close support of the key users, the project team will have a difficult time understanding the goals of the project and defining the requirements. As a result, suspicion and hostility may arise, and there can easily be an "us versus them" situation. Without effective communication and a clear direction, changes to the project's requirements always seem to appear, and both groups may set unrealistic expectations. Chaos sets in. Management begins to find fewer reasons to support an

Table 1.2 Summary of Factor Rankings for Successful, Challenged, and Impaired Projects

Rank	Factors for Successful Projects	Factors for Challenged Projects	Factors for Impaired Projects
1	User involvement	Lack of user input	Incomplete requirements
2	Executive management support	Incomplete requirements	Lack of user involvement
3	Clear statement of requirements	Changing requirements & specifications	Lack of resources
4	Proper planning	Lack of executive support	Unrealistic expectations
5	Realistic expectations	Technology incompetence	Lack of executive support
6	Smaller project milestones	Lack of resources	Changing requirements specifications
7	Competent staff	Unrealistic expectations	Lack of planning
8	Ownership	Unclear objectives	Didn't need it any longer
9	Clear vision & objectives	Unrealistic time frames	Lack of IT management
10	Hard-working, focused team	New technology	Technology illiteracy

SOURCE: Adapted from The Standish Group, *CHAOS* (West Yarmouth, MA: 1995), http://www.standishgroup .com /visitor/chaos.htm.

unpopular project and more and more resources are diverted away from it. The project is barely successful, or a failure.

Improving the Likelihood of Success

How can we improve the chances for IT project success and avoid repeating past mistakes? Here are three approaches that will be focal points throughout this book.

A Socio-Technicnl Approach In the past, organizations have attempted to improve the chances of IT project success by focusing on the tools, techniques, and methodologies of IT development. A purely technical approach, however, focuses attention on the technology. We can easily end up developing an application that no one asked for or needs. Applications to support electronic commerce, supply chain management, and integration require that at least equal attention be paid to the organizational side. The days of being good order takers are over. We can no longer be content with defining a set of user requirements, disappearing for several months, and then knocking on the user's door when it is time to deliver the new system. IT professionals must understand the business and be actively creative in applying the technology in ways that bring value to the organization. Similarly, the clients must become stakeholders in the project. This means actively seeking and encouraging their participation, involvement, and vision. The successful application of technology and the achievement of the project's goal must be an equal responsibility of the developers and users.

TAXPAYERS PAY $50 BILLION A YEAR FOR IRS MISTAKES

A *Computerworld* investigation reports that delays in overhauling the federal tax system have cost the U.S. government approximately $50 billion a year in uncol-lected taxes. Although the Internal Revenue Service (IRS) had spent hundreds of millions of dollars in an attempt to modernize its computer systems, critics claim that much of that money has been wasted because of mismanagement and primitive development practices. Government and private groups believe that there are several reasons for the problems:

- Failure to redesign the business processes before beginning systems development
- No overall systems architecture or development plan
- Primitive and sometimes "chaotic" software devel opment methodologies
- Failure to manage information systems as invest ments
- Lack of information security

Both Congress and the General Accounting Office have directed the IRS to carry out the following recommendations:

- Put in place a rigorous process for selecting, priori tizing, controlling, and evaluating major information systems investments
- Improve system development practices from ad hoc to ones that can be repeated and improve the likeli hood of success
- Develop organization-wide plans that focus on an integrated systems architecture, security, data archi tecture, and configuration management

SOURCE: Adapted from Gary H. Anthes, "IRS Project Failures Cost Taxpayers $50B Annually," *Computerworld,* October 14, 1996, http://www.computerworld.eom/news/1996/story/0,11280,10332 ,00.html.

A Project-Management Approach One suggestion of the *CHAOS* study was the need for better project management. But, isn't building an information system a project? Haven't organizations used project management in the past? And aren't they using project management now? While many organizations have applied the principles and tools of project management to IT projects, many more—even today—build systems on an ad hoc basis. Success or failure of an IT project depends largely on who is, or is not, part of the project team. Applying project management principles and tools across the entire organization, however, should be part of a **methodology**—the step-by-step activities, processes, tools, quality standards, controls, and deliverables that are defined for the entire project. As a result, project success does not depend primarily on the team, but more on the set of processes and infrastructure in place. A common set of tools and controls also provides a common language across projects and the ability to compare projects throughout the organization.

In addition, other reasons for project management to support IT projects include:

- *Resources*—When developing or purchasing an information system, all IT projects are capital projects that require cash and other organizational resources. Projects must be estimated accurately, and cost and schedules must be controlled effectively. Without the proper tools, techniques, meth ods, and controls in place, the project will drain or divert resources away from other projects and areas of the organization. Eventually, these uncon trolled costs could impact the financial stability of the organization.
- *Expectations*—Today, organizational clients expect IT professionals to deliver quality products and services in a professional manner. Timely sta tus updates and communication, as well as sound project management prac tices, are required.
- *Competition*—Internal and external competition has never been greater. An internal IT department's services can easily be outsourced if the quality or

COUNTER THINKING?

Many people find it easier to avoid failure than accept it. Yet, failure can be helpful and, at times, even desirable. Failure can be a valuable experience because one can learn more from failure than from success since the benefits of taking risks often outweigh the consequences of failure. In addition, Harold Kerzner makes three points about failure:

1. A company is not taking enough business risks if its projects are 100 percent successful.
2. Terminating a project early can be viewed as suc cessful if the resources originally dedicated to the project can be reassigned to more profitable activities or the technology needed for the project does not exist or cannot be invented cost-effectively within a reasonable time period.
3. Excellence in project management requires a continuous stream of successfully managed projects. But you can still have project failures.

SOURCE: Adapted from Alan S. Horowitz, "The Sweet Smell of Failure," *Computer-world,* http://www.computerworld.com/home/online9676.nsf/all/980209; Harold Kerzner, *In Search of Excellence in Project Management: Successful Practices in High Performance Organizations* (New York: John Wiley, 1998).

cost of providing IT services can be bettered outside the organization. Today, competition among consultants is increasing as they compete for business and talent.

- *Efficiency and Effectiveness*—Peter Drucker, the well-known management guru, defined **efficiency** as doing the thing right and **effectiveness** as doing the right thing. Many companies report that project management allows for shorter development time, lower costs, and higher quality. Just using project management tools, however, does not guarantee success. Project management must become accepted and supported by all levels within the organization, and continued commitment in terms of training, compensation, career paths, and organizational infrastructure must be in place. This support will allow the organization to do the right things and do them right.

A Knowledge-Management Approach A socio-technical approach and a commitment to project management principles and practices are important for success. However, excellence in IT project management for an individual or an organization takes time and experience. **Knowledge management** is a relatively new area. It is a systematic process for acquiring, creating, synthesizing, sharing, and using information, insights, and experiences to transform ideas into business value. Although many organizations today have knowledge management initiatives under way, and spending on knowledge management systems is expected to increase, many others believe that knowledge management is just a fad or a buzzword.

What about learning from experience? Experience can be a great teacher. These experiences and the knowledge gained from these experiences, however, are often fragmented throughout the organization. Chances are that if you encounter what appears to be a unique problem or situation, someone else in your organization has already dealt with that problem, or one very similar. Wouldn't it be great to just ask that person what they did? What the outcome was? And, would they do it again the same way? Unfortunately, that person could be on the other side of the world or down the hall—and you may not even know!

Knowledge and experience, in the form of lessons learned, can be documented and made available through the technologies accessible today, technologies such as the World Wide Web or local versions of the web called intranets. **Lessons learned** that document both reasons for success and failure can be valuable assets if maintained and used properly. A person who gains experience is said to be more mature. Similarly, an organization that learns from its experiences can be more mature in its processes by taking those lessons learned and creating **best practices**—simply, doing things in the most efficient and effective manner. In terms of managing IT projects, managing knowledge in the form of lessons learned can help an organization develop best practices that allow all of the project teams within the organization to do the right things and then to do them right. As summarized in the *CHAOS* report:

> There is one aspect to be considered in any degree of project failure. All success is rooted in either luck or failure. If you begin with luck, you learn nothing but arrogance. However, if you begin with failure and learn to evaluate it, you also learn to succeed. Failure begets knowledge. Out of knowledge you gain wisdom, and it is with wisdom that you can become truly successful (Standish Group 1995, 4).

| THE CONTEXT OF PROJECT MANAGEMENT

What Is a Project?

Although the need for effectively managing projects has been introduced, we still require a working definition of a project and project management. The Project Management Institute (PMI), an organization that was founded in 1969, has grown to become the leading non-profit professional association in the area of project management. In addition, PMI establishes many project management standards and provides seminars, educational programs, and professional certification. It also maintains the *Guide to the Project Management Body of Knowledge (PMBOK Guide).* The *PMBOK Guide* (Project Management Institute 2000) provides widely used definitions for **project** and **project management.**

> A *project* is a temporary endeavor undertaken to accomplish a unique purpose (4).

> *Project management* is the application of knowledge, skills, tools, and techniques to project activities in order to meet or exceed stakeholder needs and expectations from a project (6).

Attributes of a Project Projects can also be viewed in terms of their attributes: time frame, purpose, ownership, resources, roles, risks and assumptions, interdependent tasks, organizational change, and operating in an environment larger than the project itself.

Time Frame Because a project is a temporary endeavor, it must have a definite beginning and end. Many projects begin on a specific date and the date of completion is estimated. Some projects, on the other hand, have an immovable date when the project must be completed. In this case, it is necessary to work backwards to determine the date when the project must start. Keep in mind that your career should not consist of a single project, but a number of projects.

Purpose Projects are undertaken to accomplish something. An IT project can produce any number of results—a system, a software package, or a recommendation based on a study. Therefore, a project's goal must be to produce something tangible and of value to the organization. A project must have a goal to drive the project in terms of defining the work to be done, its schedule, and its budget, and to provide the project team with a clear direction.

Because it sets expectations that will directly influence the client's level of satisfaction, the project's goal must be clearly defined and agreed upon. The definition for project management suggests that project activities *must meet or exceed stakeholder needs and expectations.* Expectations and needs, however, cannot be met if the project's goal is not achieved. It is, therefore, important to keep in mind that a project should only be undertaken to provide some kind of value to the organization. Moreover, a specific and measurable project goal can be evaluated after the project is completed.

Ownership The project must provide something of value to an individual or group who will own the project's product after it is completed. Determining who owns this product is not *always* easy. For example, different groups may fight over who does or does not own the system, the data, the support, and the final cost of imp/ementi'ng and maintaining the system. Although a project may have *many* ***stakeholders*** (i.e., *people* or groups wfio nave a vested interest in the project's outcome), a project should have a clearly defined sponsor. The **sponsor** may be the end user, *C(f$t6(ftef, or the client who (tag the abttity and* desire to provide direction, funding, and other resources to the project.

Resources IT projects require time, money, people, and technology. Resources provide the means for achieving a project's goal and also act as a constraint. For example, the project's **scope,** or work to be accomplished, is determined directly by the project's goal—that is, if we know what we have to accomplish, we can then figure out how to accomplish it. If the project sponsor asks that an additional feature be added to the system, however, this request will undoubtedly require additional resources in terms of more work on the part of the project team. The use of a project resource has an associated cost that must be included in the overall cost of the project.

In the past, computer technology was relatively more expensive than the labor needed to develop a system. Today, the labor to build a system is relatively more expensive than the technology. As IT salaries increase, the cost of IT projects will become even more expensive. Therefore, if team members must do additional work, their time and the costs associated with time spent doing unscheduled work must be added to the project's schedule and budget. In other words, if scope increases, the schedule and budget of a project must increase accordingly. If the project's schedule and resources are fixed, then the only way to decrease the cost or schedule of the project may be to reduce the project's scope. Scope, schedule, and budget must remain in a sort of equilibrium to support a particular project goal. This relationship, sometimes referred to as the **triple constraint,** is illustrated in Figure 1.1. It should be a consideration whenever making a decision that affects the project's goal, scope, schedule, or budget.

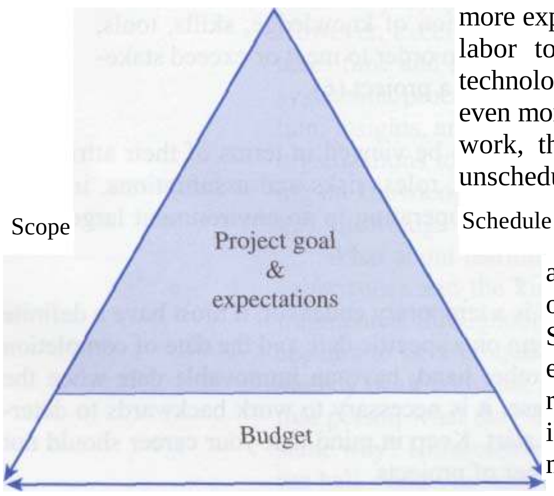

Figure 1.1 The Scope, Schedule, and Budget Relationship—the Triple Constraint

Roles Today, IT projects require different individuals with different skill sets. Although these skills may be different on different projects, a typical project may include the following:

- *Project Manager*—The project manager is the team leader and is responsi ble for ensuring that all of the project management and technical develop ment processes are in place and are being carried out within a set of specific requirements, defined processes, and quality standards.
- *Project Sponsor*—The project sponsor may be the client, customer, or orga nizational manager who will act as a champion for the project and provide organizational resources and direction when needed.
- *Subject Matter Expert(s) (SME)*—The subject matter expert may be a user or client who has specific knowledge, expertise, or insight in a specific functional area needed to support the project. For example, if the organiza tion wishes to develop a system to support tax decisions, having a tax expert on the project team who can share his/her knowledge will be more productive than having the technical people try to learn everything about tax accounting while developing the system.
- *Technical Expert(s) (TE)*—Technical expertise is needed to provide a tech nical solution to an organizational problem. Technical experts can include systems analysts, network specialists, programmers, graphic artists, train ers, and so forth. Regardless of their job title, these individuals are respon sible for defining, creating, and implementing the technical and organizational infrastructure to support the product of the IT project.

Risks and Assumptions All projects have an element of risk, and some projects entail more risk than others. Risk can arise from many sources, both internal and external to the project team. For example, **internal risks** may arise from the estimation process or from the fact that a key member of the project team could leave in the middle of the project. **External risks,** on the other hand, could arise from dependencies on other contractors or vendors. **Assumptions** are what we use to estimate scope, schedule, and budget and to assess the risks of the project. There are many unknown variables associated with projects, and it is important to identify and make explicit all of the risks and assumptions that can impact the IT project.

Interdependent Tasks Project work requires many interdependent tasks. For example, a network cannot be installed until the hardware is delivered, or certain requirements cannot be incorporated into the design until a key user is interviewed. Sometimes the delay of one task can affect other subsequent, dependent tasks. The project's schedule may slip, and the project may not meet its planned deadline.

Organizational Change Projects are planned organizational change. Change must be understood and managed because implementation of the IT project will change the way people work. The potential for resistance, therefore, exists, and a system that is a technical success could end up being an organizational failure.

Operating in an Environment Larger than the Project Itself Organizations choose projects for a number of reasons, and the projects chosen can impact the organization (Laudon and Laudon 1996). It is important that the project manager and team understand the company's culture, environment, politics, and the like.

These organizational variables will influence the selection of projects, the IT infrastructure, and the role of IT within the organization. For example, a small, family-owned manufacturing company may have a completely different corporate culture, strategy, and structure than a start-up electronic commerce company. As a result, the projects selected, the technical infrastructure, and the role of IT for each organization will be different. The project team must understand both the technical and organizational variables so that the project can be aligned properly with the structure and strategy of the organization. Moreover, understanding the organizational variables can help the project team understand the political climate within the organization and identify potential risks and issues that could impede the project.

THE PROJECT LIFE CYCLE AND IT DEVELOPMENT

The **project life cycle** (PLC) is a collection of logical stages or phases that maps the life of a project from its beginning to its end in order to define, build, and deliver the product of a project—that is, the information system. Each phase should provide one or more deliverables. A **deliverable** is a tangible and verifiable product of work (i.e., project plan, design specifications, delivered system, etc.). Deliverables at the end of each phase also provide tangible benefits throughout the project and serve to define the work and resources needed for each phase.

Projects should be broken up into phases to make the project more manageable and to reduce risk. **Phase exits, stage gates, or kill points** are the phase-end review of key deliverables that allow the organization to evaluate the project's performance and to take immediate action to correct any errors or problems. Although the deliverables at the end of a stage or phase usually are approved before proceeding to the next stage, **fast tracking** or starting the next phase before approval is obtained can sometimes reduce the project's schedule. Overlapping of phases can be risky and should only be done when the risk is deemed acceptable.

Like all living things, projects have life cycles where they are born, grow, peak, decline, and then terminate (Gido and Clements 1999; Meredith and Mantel 2000). Although project life cycles may differ depending upon the industry or project, all project life cycles will have a beginning, a middle, and an end (Rosenau 1998; Gido and Clements 1999). Figure 1.2 provides a generic life cycle that describes the common phases or stages shared by most projects.

Define Project Goal

Defining the project's overall goal should be the first step of the project. This goal should focus on providing business value to the organization. A well-defined goal gives the project team a clear focus and drives the other phases of the project. In addition, most projects seem to share the following characteristics:

- The effort, in terms of cost and staffing levels, is low at the start of the project, but then increases as the project work is being done, and then decreases at the end as the project is completed.
- Risk and uncertainty are the highest at the start of a project. Once the goal of the project is defined and the project progresses, the probability of suc cess should increase.

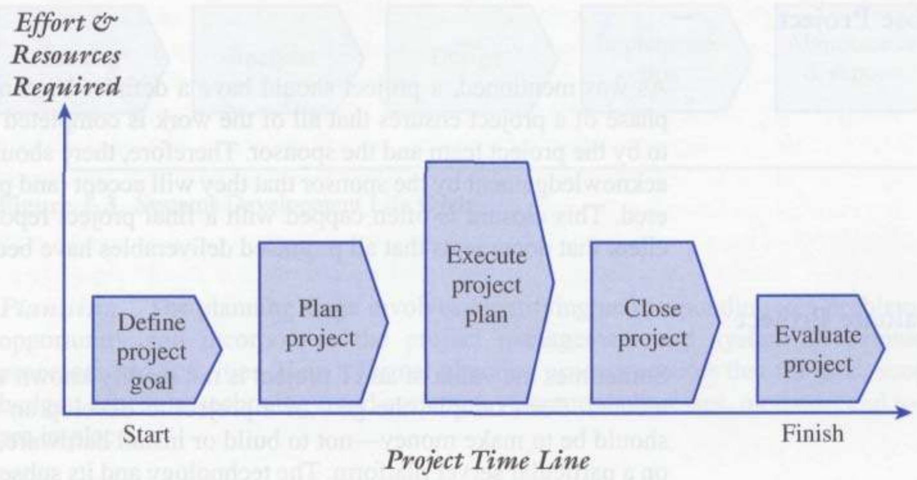

Figure 1.2 A Generic Project Life Cycle

- The ability for stakeholders to influence the scope and cost of the project is highest at the beginning of the project. The cost of changing the scope and correcting errors becomes more expensive as the project progresses.

Plan Project

Once the project's goal has been defined, developing the project plan is a much easier task. A project plan essentially answers the following questions:

- What are we going to do?
- Why are we going to do it?
- How are we going to do it?
- Who is going to be involved?
- How long will it take?
- How much will it cost?
- What can go wrong and what can we do about it?
- How did we estimate the schedule and budget?
- Why did we make certain decisions?
- How will we know if we are successful?

In addition, the deliverables, tasks, resources, and time to complete each task must be defined for each phase of the project. This initial plan, called a **baseline plan,** defines the agreed upon scope, schedule, and budget and is used as a tool to gauge the project's performance throughout the life cycle.

Execute Project Plan

After the project's goal and plan have been defined, it's time to put the plan into action. As work on the project progresses, scope, schedule, budget, and people must be actively managed to ensure that the project achieves its goal. The project's progress must be documented and compared to the project's baseline plan. In addition, project performance must be communicated to all of the project's stakeholders. At the end of this phase, the project team implements or delivers a completed product to the organization.

Close Project

As was mentioned, a project should have a definite beginning and end. The closing phase of a project ensures that all of the work is completed as planned and as agreed to by the project team and the sponsor. Therefore, there should be some kind of formal acknowledgement by the sponsor that they will accept (and pay for!) the product delivered. This closure is often capped with a final project report and presentation to the client that documents that all promised deliverables have been completed as specified.

Evaluate Project

Sometimes the value of an IT project is not readily known when the system is implemented. For example, the goal of a project to develop an electronic commerce site should be to make money—not to build or install hardware, software, and web pages on a particular server platform. The technology and its subsequent implementation are only a means to an end. Therefore, the goal of the electronic commerce site may be to produce $250,000 within six months. As a result, evaluating whether the project met its goal can be made only after the system has been implemented.

However, the project can be evaluated in other ways as well. The project team should document its experiences in terms of lessons learned—those things that it would do the same and those things it would do differently on the next project, based on its current project experiences. This post mortem should be documented, stored electronically, and shared throughout the organization. Subsequently, many of these experiences can be translated into best practices and integrated into future projects.

In addition, both the project team and the project itself should be evaluated at the end of the project. The project manager may evaluate each project team member's performance in order to provide feedback and as part of the organization's established merit and pay raise processes and procedures. Often, however, an outside third party, such as a senior manager or partner, may audit the project to determine whether the project was well-managed, provided the promised deliverables, followed established processes, and met specific quality standards. The project team and project manager may also be evaluated in terms of whether they acted in a professional and ethical manner.

The IT Product Life Cycle

Although projects follow a project life cycle, information systems development follows a product life cycle. The most common product life cycle in IT is the **Systems Development Life Cycle** (SDLC), which represents the sequential phases or stages an information system follows throughout its useful life. The SDLC establishes a logical order or sequence in which the system development activities occur and indicates whether to proceed from one system development activity to the next (McConnell 1996). Although there is no generally accepted version of the SDLC, the life cycle depicted in Figure 1.3 includes the generally accepted activities and phases associated with systems development. Keep in mind that these concepts are generally covered in great detail in system analysis and design books and courses. For some, this may be a quick review, while for others it will provide a general background for understanding how IT project management and information system development activities support one another.

Planning, analysis, design, implementation, and maintenance and support are the five basic phases in the systems development life cycle.

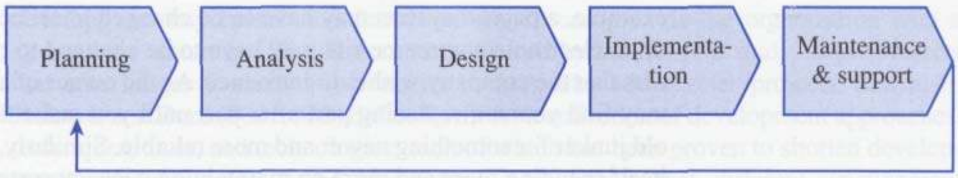

Figure 1.3 Systems Development Life Cycle

Planning The planning stage involves identifying and responding to a problem or opportunity and incorporates the project management and system development processes and activities. Here a formal planning process ensures that the goal, scope, budget, schedule, technology, and system development processes, methods, and tools are in place.

Analysis The analysis phase attempts to delve into the problem or opportunity more fully. For example, the project team may document the current system to develop an "as is" model to understand the system currently in place. In general, systems analysts will meet with various stakeholders (users, managers, customers, etc.) to learn more about the problem or opportunity. This work is done to identify and document any problems or bottlenecks associated with the current system. Generally, the "as is" analysis is followed by a requirements analysis. Here the specific needs and requirements for the new system are identified and documented. Requirements can be developed through a number of means—interviewing, joint applications development (JAD), conducting surveys, observing work processes, and reading company reports. Using process-oriented, data-oriented, and/or object-oriented modeling techniques, the current system, user requirements, and logical design of the future system called the "to be" system are represented and documented (Dennis and Haley 2000).

Design During the design phase, the project team uses the requirements and "to be" logical models as input for designing the architecture to support the new information system. This architecture includes designing the network, hardware configuration, databases, user interface, and application programs.

Implementation Implementation includes the development or construction of the system, testing, and installation. In addition, training, support, and documentation must be in place.

Maintenance and Support Although maintenance and support may not be a true phase of the current project, it is still an important consideration. Once the system has been implemented, it is said to be in production. Changes to the system, in the form of maintenance and enhancements, are often requested to fix any discovered errors (i.e., bugs) within the system, to add any features that were not incorporated into the original design, or to adjust to a changing business environment. Support, in terms of a call center or help desk, may also be in place to help users on an as-needed basis.

Eventually, the system becomes part of the organizational infrastructure and becomes known as a legacy system. At this point, the system becomes very similar to a car. Let's say you buy a brand new car. Over time, the car becomes less and less new, and parts have to be replaced as they wear out. Although, a system does not wear out like a car, changes to the system are required as the organization changes. For example, a payroll system may have to be changed to reflect changes in the tax laws, or an electronic commerce site may have to be changed to reflect a new line of products that the company wishes to introduce. As the owner of an older or classic car, you may find yourself replacing part after part until you make the decision to trade in the old junker for something newer and more reliable. Similarly, an organization may find itself spending more and more on maintaining a legacy system. Eventually, the organization will decide that it is time to replace this older system with a newer one that will be more reliable, require less maintenance, and better meets its needs. Subsequently, a new life cycle begins.

Putting the SDLC into Practice

There are basically two ways to implement the SDLC. Today, an IT project will follow either a structured approach or a newer approach called Rapid Applications Development (RAD).

Structured Approach to Systems Development A structured approach to systems development has been around since the 1960s and 1970s when large mainframe applications were developed. These applications were built when (1) systems were relatively simple and independent from each other, (2) computer hardware was relatively more expensive than the labor, and (3) development and programming tools were primitive compared to today (Satzinger, Jackson, Burd 2002).

The Waterfall method in Figure 1.4 follows the SDLC in a very sequential and structured way. Planning overhead is minimized because it is all done up-front and a tangible system is not produced until the end of the life cycle (McConnell 1996). The idea of a waterfall is a metaphor for a cascading of activities from one phase to the next. This approach stresses a sequential and logical flow of development activities. For example, the design activities begin only after the analysis and requirement activities are complete. Subsequently, the actual development, or programming activities, will not start until the design phase is complete. Although one can go back to a previous stage, it is not always easy or desirable.

This approach is suitable when developing structured systems and assumes, or at least hopes, that the requirements defined in the analysis stage do not change very much over the remainder of the project. In addition, because it will provide a solid structure that can minimize wasted effort, this method may work well when the project team is inexperienced or less technically competent (McConnell 1996).

Rapid Applications Development (RAD) On the other hand, one can take a less-structured approach to developing systems. Today, taking less time to conceive,

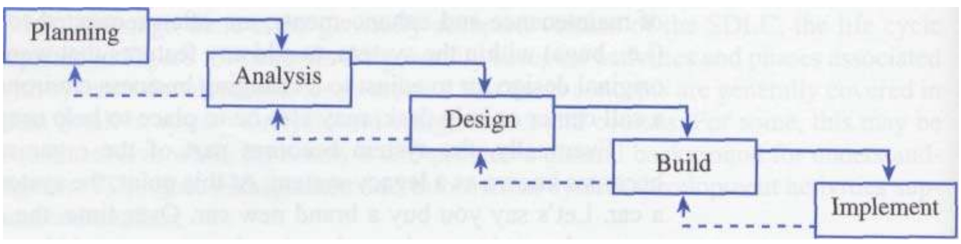

Figure 1.4 Waterfall Model

develop, and implement an information system can provide the organization with a competitive advantage. In addition, as evidenced by the *CHAOS* study, larger projects that take longer to develop are riskier than smaller and shorter projects. Satzinger, Jackson, and Burd (2002, 533) define RAD as "a collection of development approaches, techniques, tools, and technologies, each of which has been proven to shorten development schedules under some conditions." This means that different development approaches, tools, techniques, and so forth can be mixed and matched depending on the project. For some projects, it means that the Waterfall approach is the most appropriate; however, RAD often follows one of the following iterative approaches:

- *Prototyping*—Prototyping is an iterative approach to systems development where the developer and user work together very closely to develop a partially or fully functional system as soon as possible, usually within a few days or weeks. The prototype application will go through a number of iterations as functional requirements are defined or changed. This approach is most useful when the requirements of the new system are difficult to define or when working with a new technology where the capabilities of that technology are unknown or not understood very well. A prototype may be either a throwaway system or a fully usable system. A throwaway prototype may be used to discover or refine system requirement specifications that can be used as a model for developing the real system. On the other hand, the prototype may become the actual system after it has gone through a number of refinements over time.
- *Spiral Development*—Another way to expedite the SDLC is the spiral approach first proposed by Barry Boehm (1988). The spiral model provides a risk-oriented approach where a software project is broken up into a number of miniprojects where each addresses one or more major risks until all major risks have been addressed (McConnell 1996). A risk can be defined as a poorly understood requirement or architecture or as a potential problem with the technology or system performance. The basic idea is to begin development of the system on a small scale where risks can be identified. Once identified, the development team then develops a plan for addressing these risks and evaluates various alternatives. Next, deliverables for the iteration are identified, developed, and verified before planning and committing to the next iteration. Subsequently, completing each iteration brings the project closer to a fully functional system. Reviews after each iteration provide a means of controlling the overall risk of the project. Major problems or challenges will surface early in the project and, therefore, provide the potential to reduce the total cost of the project. The disadvantages to the spiral development approach center on its complexity (Satzinger, Jackson, Burd 2002). These types of projects are more complex to manage because many people may be working on a number of different parallel activities.
- *Extreme Programming (XP)*—Kent Beck introduced the idea of XP in the mid-1990s. Under XP, the system is transferred to the users in a series of versions called releases. A release may be developed using several iterations and should be developed and tested within a few weeks or months. Each release is a working system that only includes one or several functions that are part of the full system specifications. XP includes a number of activities where the user requirements are first documented as a user story. The user stories are then documented using an object-oriented model

called the class diagram, and the release is developed over the course of several iterations. A set of acceptance tests for each user story is then developed. Releases that pass the acceptance test are then considered complete. XP provides the continuous testing and integration of different software modules and components while supporting active user involvement. In addition, XP often incorporates team programming where two programmers work together on the same workstation. Small teams of developers often work in a common room where workstations are positioned in the middle and workspace for each team member is provided around the perimeter. Developers often are prohibited from working more than 40 hours a week in order to avoid burnout and the mistakes that often occur because of fatigue (Satzinger, Jackson, Burd 2002).

The PLC versus the SDLC

You may be still wondering about the difference between the project life cycle and the systems development life cycle. Although they may seem to be quite similar, the difference is that the product life cycle focuses on the processes of managing a project, while the SDLC focuses on creating and implementing a product—the information system. In this text we will focus primarily on the PLC, although the SDLC and the particular approach we choose will have a direct bearing on the project's scope (i.e., the deliverables that the project team must provide) and the work activities needed to produce those deliverables. Consequently, the number of activities, their sequence, time-to-complete, and resources required will directly determine the project's schedule and budget.

As illustrated in Figure 1.5, the SDLC is really part of the PLC because many of the activities for developing the information system occur during the execution phase. The last two stages of the PLC, closing and evaluating the project, occur after the implementation of the information system.

The integration of project management and system development activities is one important component that distinguishes IT projects from other types of projects. A

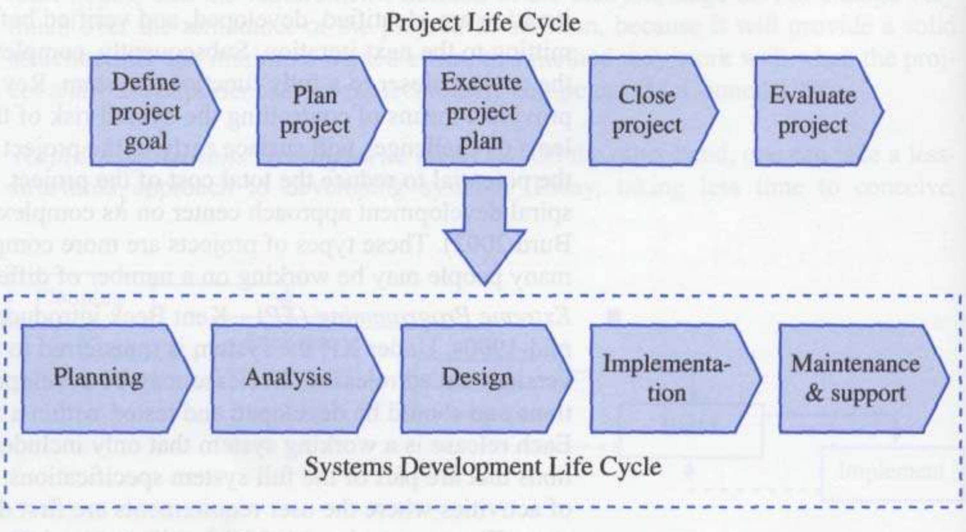

Figure 1.5 PLC and SDLC

methodology will be presented in Chapter 2 and will illustrate how the project life cycle and systems development life cycle can be combined to manage the process and product of IT projects. This methodology will provide a foundation for the concepts, processes, tools, and techniques throughout this text.

THE PROJECT MANAGEMENT BODY OF KNOWLEDGE (PMBOK)

As was mentioned earlier, the *Guide to the Project Management Body of Knowledge* is a document available from the Project Management Institute (PMI) — an international, nonprofit, professional organization with more than 55,000 members worldwide. The original document was published in 1987, and the updated version provides a basis for identifying and describing the generally accepted principles and practices of project management. However, as PMBOK is quick to point out, "generally accepted" does not mean these principles and practices work the same way on each and every project. It does mean that many people over time believe that these principles and practices are useful and have value. Determining what is appropriate is the responsibility of the team and comes from experience. (Perhaps experiences that can be documented and shared?)

This text will use the *PMBOK Guide* as a foundation but will also integrate a number of concepts and ideas that are part of the body of knowledge that makes up the field of information systems. Ideally, you will then understand not only what many IT project managers and organizations throughout the world think are important, but also the language and the processes.

PMI provides a certification in project management through the Project Management Professional (PMP) certification exam. This text can also help you prepare for the PMP certification exam. To pass, you must demonstrate a level of understanding and knowledge about project management, satisfy education and experience requirements, and agree to and adhere to a professional code of ethics.

Project Management Knowledge Areas

The *Guide to the Project Management Body of Knowledge* defines nine knowledge areas for understanding project management. These nine knowledge areas are illustrated in Figure 1 .6 and will be covered in more detail in later chapters.

- *Project Integration Management* — Integration focuses on coordinating the project plan's development, execution, and control of changes.
- *Project Scope Management* — A project's scope is the work to be completed by the project team. Scope management provides assurance that the pro ject's work is defined accurately and completely and that it is completed as planned. In addition, scope management includes ways to ensure that proper scope change procedures are in place.
- *Project Time Management* — Time management is important for developing, monitoring, and managing the project's schedule. It includes identifying the project's phases and activities and then estimating, sequencing, and assign ing resources for each activity to ensure that the project's scope and objec tives are met.
- *Project Cost Management* — Cost management assures that the project's budget is developed and completed as approved.

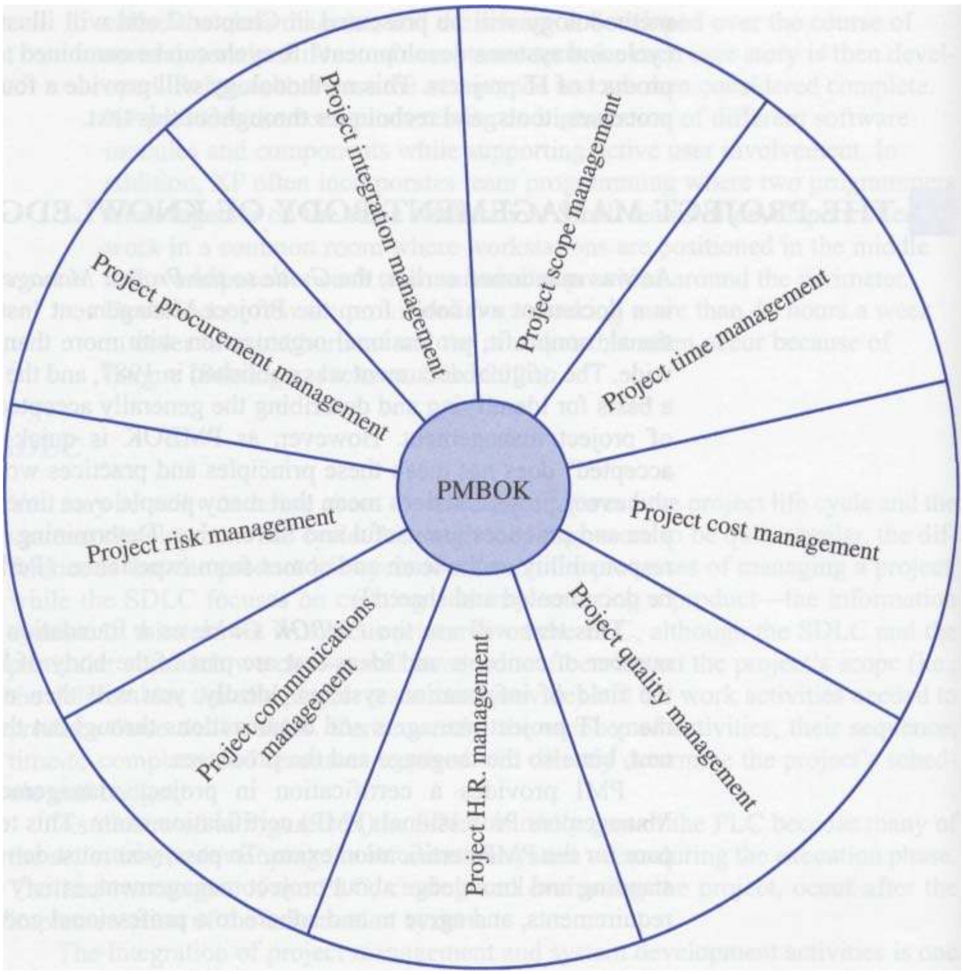

Figure 1.6 Project Management Body of Knowledge (PMBOK)

Project Quality Management—Quality management focuses on planning, developing, and managing a quality environment that allows the project to meet or exceed stakeholder needs or expectations.

Project Human Resource Management—People are the most important resource on a project. Human resource management focuses on creating and developing the project team as well as understanding and responding appropriately to the behavioral side of project management.

Project Communications Management—Communication management entails communicating timely and accurate information about the project to the project's stakeholders.

Project Risk Management—All projects face a certain amount of risk. Project risk management is concerned with identifying and responding appropriately to risks that can impact the project.

Project Procurement Management—Projects often require resources (people, hardware, software, etc.) that are outside the organization. Procurement management makes certain that these resources are acquired properly.

CHAPTER SUMMARY

This chapter provides an introduction to the text and to the area of information technology project management (ITPM). As evidenced by the *CHAOS* report published by The Standish Group, many IT projects are late and over-budget and include only a fraction of the functionality originally envisioned. Although many factors contribute to a project's success or failure, the product and process associated with the development of an information system must be actively managed. This management includes taking a socio-technical approach that focuses not only on the technology, but also on the organizational side. In addition, individuals and organizations can learn and share their experiences. These experiences, in the form of lessons learned, can be used to develop new ideas and best practices that can be implemented in an organization's systems development and project management policies and methods.

The *Guide to the Project Management Body of Knowledge (PMBOK Guide)* defines a project as a temporary endeavor undertaken to accomplish a unique purpose and project management as the application of knowledge, skills, tools, and techniques to project activities in order to meet or exceed stakeholder needs and expectations. Projects can also be viewed in terms of their attributes. These attributes include the project's time frame, purpose, ownership, resources, roles, risks and assumptions, tasks, and the impact the project will have on the organization. Projects also operate in an environment larger than the project itself. The company's culture, environment, politics, strategy, structure, policies, and processes can influence the selection of projects, the IT infrastructure, and the role of IT within the organization. Similarly, the selection of projects, the IT infrastructure, and the role of IT within the organization can influence the organizational variables.

The project life cycle (PLC) is a collection of logical stages or phases that maps the life of a project from its beginning to its end. It also helps in defining, building, and delivering the product of a project. Projects are broken up into phases to make the project more manageable and to reduce risk. In addition, each phase should focus on providing a deliverable—a tangible and verifiable product of work. A generic project life cycle was introduced. Its phases included (1) defining the project goal, (2) planning the project, (3) executing or carrying out the project, (4) closing the project, and (5) evaluating the project. Although projects follow a project life cycle, information systems development follows a product life cycle.

The Systems Development Life Cycle (SDLC) represents the sequential phases or stages an information system follows throughout its useful life. The SDLC described in this chapter includes the following phases: (1) planning, (2) analysis, (3) design, (4) implementation, (5) maintenance and support. In addition, the SDLC can be implemented using a structured approach (the Waterfall model) or by means of more iterative approaches. By following a rapid applications development (RAD) approach, systems developers can combine different approaches, tools, and techniques in order to shorten the time needed to develop an information system. The SDLC is really a component of the PLC, and choice of a particular approach for systems development will influence the activities, their sequence, and the estimated time to complete. In turn, this will directly impact the project's schedule and budget.

The *Guide to the Project Management Body Knowledge* outlines nine knowledge areas for understanding project management. These nine areas include: (1) project integration management, (2) project scope management, (3) project time management, (4) project cost management, (5) project quality management, (6) project human resources management, (7) project communications management, (8) project risk management, and (9) project procurement management. Along with a number of concepts and principles that make up the body of knowledge for information systems, these nine PMBOK areas will be integrated in the chapters throughout this text.

REVIEW QUESTIONS

1. Describe the software crisis in your own words.
2. How is a successful project defined in the *CHAOS* study?
3. How is a challenged project defined in the *CHAOS* study?
4. How is an impaired project defined in the *CHAOS* study?
5. Why are many IT projects late, over budget, and with fewer features and functions than originally envisioned?

"• What is the socio-technical approach to systems development?

What are the benefits to using a project management approach to developing information systems?

8. What is a methodology? What are the advantages of following a methodology when developing an information system?
9. How does sharing experiences in the form of les sons learned lead to best practices in managing and developing information systems?
10. What is a project?
11. What is project management?
12. What are the attributes of a project?
13. Describe the relationship among scope, schedule, and budget.
14. Describe the different roles and skill sets needed for a project.
15. Describe three risks that could be associated with an IT project.
16. Why should assumptions associated with a project be documented?
17. Discuss the statement: Projects operate in an envi ronment larger than the project itself.
18. Describe the project life cycle.
19. What are phase exits, stage gates, and kill points? What purpose do they serve?
20. What is fast tracking? When should fast tracking be used? When is fast tracking not appropriate?
21. Describe the Systems Development Life Cycle (SDLC).
22. Describe the Waterfall model for systems develop ment. When should the Waterfall model *be* used?
23. Describe the prototyping approach to systems development. When is prototyping appropriate?
24. Describe the Spiral approach for iterative develop ment. What advantages does this model have in comparison with the Waterfall model?
25. Describe extreme programming (XP). How does XP accelerate the SDLC?
26. What is knowledge management? Although many people believe knowledge cannot be managed, why do you think many companies are undertaking knowledge management initiatives?
27. Although the *Guide to the Project Management Body of Knowledge* describes the generally accepted principles and practices of project man agement, why wouldn't these principles and prac tices work for every project?

EXTEND YOUR KNOWLEDGE

I.
Using the web or library, find an article that describes either a successful or an unsuccessful IT project. Discuss whether any of the project factors listed in Table 1.2 had any bearing on the project. 2. Design a template that could be used by a project team to document its experiences and lessons learned. Describe or show how these experiences could be catalogued and shared with other members and other teams.

3. Using the web or library as a resource, write a one-page position paper on knowledge management. You should provide a definition of knowledge management and your opinion as to whether an organization should invest in a knowledge management initiative.

BIBLIOGRAPHY

Boehm, B. W. 1988. A Spiral Model of Software Development and Enhancement. *Computer* (May): 61-72. Dennis, A. and W. B. Haley. 2000. *Systems Analysis and Design: An Applied Approach.* New York: John Wiley. Gido, J. and J. P. Clements. 1999. *Successful Project Management.* Cincinnati, OH: South-Western College Publishing. Hoffman, T. and J. King. 2000. Y2K Freeze Melts In January Thaw. *Computer-world,* January 17. http://www.computerworld.com /home/print.nsf/all/000117E04A. Laudon, K. C. and J. P. Laudon. 1996. *Management Information Systems: Organization and Technology.* Upper Saddle River, NJ: Prentice Hall. McConnell, S. 1996. *Rapid Development: Taming Wild Software Schedules.* Redmond, WA: Microsoft Press.

Meredith, J. R. and S. J. Mantel, Jr. 2000. *Project Management: A Managerial Approach.* New York: John Wiley. Project Management Institute (PMI). 2000. *A Guide to the Project Management Body of Knowledge (PMBOK Guide).* Newtown Square, PA: PMI Publishing. Rosenau, M. D. J. 1998. *Successful Project Management.* New York: John Wiley. Satzinger, J. W, R. B. Jackson, and S. D. Burd. 2002. *Systems Analysis and Design in a Changing World.* Boston: Course Technology. Standish Group. 1995. *CHAOS.* West Yarmouth, MA: The Standish **Group.**