OPENING CASE

Blue Sky Mutual Funds: A New Development Approach

Jim Williams, vice president of finance for Blue Sky Mutual Funds, spoke first. "There are some things I like about this new approach, but other things worry me," he told Gary Johnson, the company's director of information technology.

"This idea of 'growing' the system through several iterations makes a lot of sense to me. It is always hard for my people to know exactly what they need a new information system to do and what will work best for the company. So, if they can get their hands on the system early, they can begin acceptance testing and try it out to see whether it addresses their needs in the best way.

"Let me see if I understand the big picture, though. Your development team and my investment advisors will decide on a few core processes that the system needs to support and then your team will design and build a system to support those core processes. You will do that in a mini-project that will last about six weeks. Then, you will continue adding more functionality through several other mini-projects until the system is complete and functioning well. Is that right?"

Jim was becoming more enthusiastic about this new approach to system development.

"Yes, that's the basic idea," Gary said. "Your users need to understand that the first few versions of the system won't be complete and may not be completely robust either. But these early versions will give them something to work with and try out. We also need good feedback from their acceptance testing so the system will be thoroughly tested by the time we are through."

"I realize that," Jim said. "My people will like not having to think from the very beginning about everything they need the system to do. They'll like being able to try things out. As I said earlier, I like this approach. However, the part I don't like about this approach is that it will be more difficult for you to give me a firm time schedule and project cost. That worries me. In the past, those have been two of the major tools we used to monitor a project's progress. Are you saying that now we won't have a schedule at all? And you want an open budget?" Jim frowned.

"It's not as bad as it first sounds," Gary said. "This approach is an 'adaptive' approach, by which I mean that because the system is growing, the project is more open ended. The project manager will still create a schedule and estimate the project costs, but she won't even try to identify and lock in all the required functionality for several of the iterations. Because the system's scope is going to continually be refined over the first few iterations, there is the risk of 'scope creep.' That is one of the biggest risks with adaptive approaches. You and I should meet with the project manager fairly frequently to ensure that the scope is controlled and the project doesn't get out of control."

"Okay," Jim said. "You have convinced me to try this new approach. However, let's treat this project as a pilot and see how it works. If it's successful, we will consider using this iterative approach on our other projects." Jim and Gary agreed that a pilot was the best way to get started. Gary then headed off to meet with the project manager and get the project started.

Overview

Chapter 8 introduced you to the SDLC and the various alternatives for organizing software development activities. By now, you may be asking yourself such questions as:

- "How are all these activities coordinated?"
- "How do I know which tasks to do first?"
- "How is the work assigned to the different teams and team members?"
- "How do I know which parts of the new system should be developed first?"

The purpose of project planning and project management is to bring some order to all these (sometimes seemingly unrelated) tasks. As you will learn in this chapter, the success of any given project highly depends on the skills and abilities of those managing the project. You will also learn that project management skills aren't only for project managers—that all the project team members contribute to the management of the project and thus to its success.

This chapter first discusses the need for project management and the principles associated with it. The rest of this chapter discusses the detailed activities that are associated with the first two core processes of systems development, both of which are primarily project management processes. The purpose of this chapter is to teach you how to plan, organize, and direct a systems development project.

Principles of Project Management

Many of you may have built a Web page with HTML or written a computer program for yourself or a friend. In those cases, where it was just you working, you weren't too concerned about how to organize your work or how to manage the project. However, as soon as two or more developers are working together, the work must be partitioned and organized, with specific assignments for each developer. This is true whether the project uses a predictive approach or an adaptive approach. As discussed in the last chapter, the chosen methodology lays out a complex set of activities and tasks that must be carefully managed. Failing to organize usually causes wasted time and effort as well as confusion and may even cause the project to fail.

Even though every project team designates one person as the project manager, with primary responsibility for the way the team functions, all members contribute to the team's management. The project manager for the RMO CSMS project is Barbara Halifax, but she has a senior systems analyst helping her every step of the way. As the project proceeds, all team members are involved in aspects of managing the project.

As discussed in earlier chapters, a project is a planned undertaking with a beginning and an end, which produces a predetermined result and is usually constrained by a schedule and resources. The development of information systems fits this definition. In addition, it is usually a quite complex project, with many people and tasks that have to be organized and coordinated. Whatever its objective, each project is unique. Different products are produced, different activities are required with varying schedules, and different resources are used. This uniqueness makes information systems projects difficult to control.

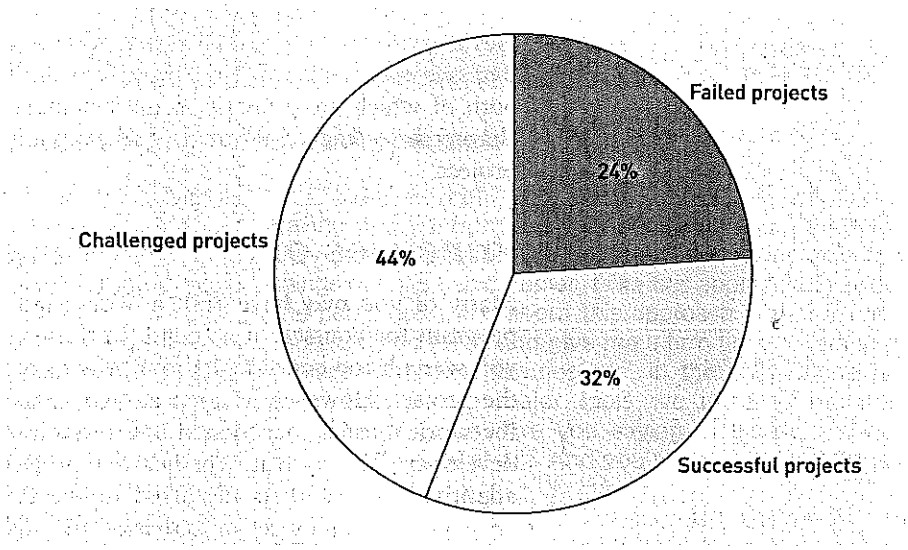
The Need for Project Management

Studies suggest that most IT projects are less than successful as measured by three criteria: finishing on time, finishing within budget, and effectively meeting the need as expressed by the original problem definition. Since 1994, the well-known Standish Group has produced an annual CHAOS report, which provides statistics on the outcome of IT development projects for the preceding year. The Standish Group categorizes projects in three ways: (1) successful projects, which are completed on time and within budget while meeting the users' requirements for functionality; (2) challenged projects, which have some combination of being late, over budget, or reduced in scope; and (3) failed projects, which are cancelled or result in the system never being used. The numbers vary somewhat year by year, with more recent years showing a slight improvement. In 2009, the results indicated that 32 percent were successful, 44 percent were challenged, and 24 percent were failed projects (see **Figure 9-1**). Billions of dollars are spent on projects that don't meet their objectives.

The Standish Group's report doesn't just indicate the rate of IT project failure or success; it also identifies the reasons for each. Here are some of the reasons for failure:

- Undefined project management practices
- Poor IT management and poor IT procedures

FIGURE 9-1
Project completion results as reported by the Standish Group



- Inadequate executive support for the project
- Inexperienced project managers
- Unclear business needs and project objectives
- Inadequate user involvement

It is notable that the primary reasons projects fail are a lack of executive involvement and a lack of management skills. The other major reason is lack of involvement by the user community. In other words, projects don't tend to fail for lack of programming skills or enthusiastic developers.

For an IT project to be successful, strong IT management and business direction need to be present. The other major element in all project success is sound project management procedures as well as experienced and competent project managers. In fact, good project managers always ensure that they have received clear directives from business executives and committed user involvement with the requirements for the new system.

The Role of the Project Manager

project management organizing and directing other people to achieve a planned result within a predetermined schedule and budget

Project management is organizing and directing other people to achieve a planned result within a predetermined schedule and budget. At the beginning of a project, a plan is developed that specifies the activities that must take place, the deliverables that must be produced, and the resources that are needed. Thus, project management can also be defined as the processes used to plan the project and then to monitor and control it.

One of the most exciting careers for IT-oriented people is being a project manager. As projects become more complex because of shorter time frames, distributed project teams (including off-shore and cross-cultural teams), rapidly changing technology, and more sophisticated requirements, highly qualified project managers are sought after and paid well. Many universities are adding project management courses to their curricula to respond to the needs of industry. There is a strong need and a high demand for people who are capable project managers. As your career progresses, you should develop your management skills. You may even want to become active in the Project Management Institute (PMI), which is the most well-known professional organization for project managers.

Overall, project managers must be effective internally (managing people and resources) and externally (conducting public relations). Internally, the project

manager serves as locus of control for the project team and all its activities. He or she establishes the team's structure so work can be accomplished. This list identifies a few of these internal responsibilities:

- Developing the project schedule
- Recruiting and training team members
- Assigning work to teams and team members
- Assessing project risks
- Monitoring and controlling project deliverables and milestones

Externally, the project manager is the main contact for the project. He or she must represent the team to the outside world and communicate the team members' needs. Major external responsibilities include:

- Reporting the project's status and progress
- Working directly with the client (the project's sponsor) and other stakeholders
- Identifying resource needs and obtaining resources

client the person or group that funds the project

oversight committee clients and key managers who review the progress and direct the project

users the person or group of people who will use the new system

A project manager works with several groups of people. First of all, there is the **client** (i.e., the customer), who pays for the development of the new system. Project approval and the release of funds come from the client. For in-house developments, the client may be an executive committee or a vice president. The client approves and oversees the project, along with its funding. For large, mission-critical projects, an **oversight committee** (sometimes called the steering committee) may be formed. This consists of clients and other key executives who have a vision of the organization's strategic direction and a strong interest in the project's success. On the other hand, the **users** are the people who will actually use the new system. The user typically provides information about the detailed functions and operations needed in the new system.

Communication with the client and oversight committee is an important part of the project manager's external responsibilities. Similarly, working with the team leaders, team members, and any subcontractors is an important part of a project manager's internal responsibilities. The project manager must ensure that all internal and external communication is flowing properly. **Figure 9-2** depicts the various groups of people involved in a development project.

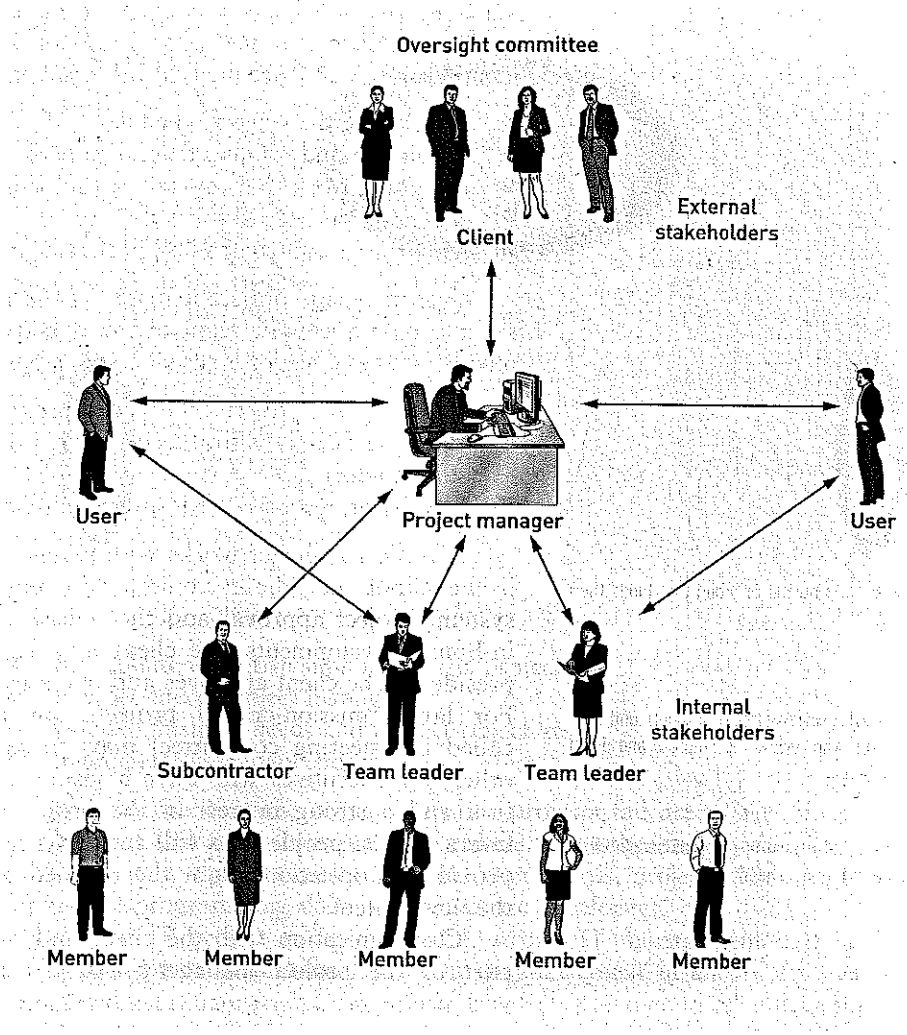
Project Management and Ceremony

ceremony the level of formality of a project; the rigor of holding meetings and producing documentation

Another dimension that has a heavy impact on project management is the level of formality, sometimes called ceremony, required for a given project. **Ceremony** is a measure of the amount of documentation generated, the traceability of specifications, and the formality of the project's decision-making processes. Some projects, particularly small ones, are conducted with very low ceremony. Meetings occur in the hallway or around the water cooler. Written documentation, formal specifications, and detailed models are kept to a minimum. Developers and users usually work closely together on a daily basis to define requirements and develop the system. Other projects, usually larger, more complex ones, are executed with high ceremony. Meetings are often held on a predefined schedule, with specific participants, agendas, minutes, and follow-through. Specifications are formally documented with an abundance of diagrams and documentation and are frequently verified through formal review meetings between developers and users.

A project's ceremony isn't the same as whether its approach is predictive or adaptive. However, even though the approach and ceremony are different, large predictive projects often tend to have high ceremony, with lots of meetings and documentation. Unfortunately, the extensive documentation tended to increase

FIGURE 9-2
Participants in a system development project



the length of the project and sometimes contributed to cost overruns. Techniques such as rapid application development (RAD) were utilized to help manage large predictive projects with less formality. This approach required less documentation and fewer status and review meetings. Of course, many smaller projects were often managed with less ceremony.

Adaptive projects can also be more or less formal in the way they are managed. The Unified Process, which will be explained in Chapter 14, is quite formal, with high ceremony. Each iteration is precisely defined, with such specific outcomes as specifications, diagrams, prototypes, and deliverables. However, adaptive, iterative approaches also lend themselves to being managed with much less formality. The inherent characteristics of an iterative approach, with its "just in time" project plans, easily adjust to less documentation, fewer diagrams for specifications, and less formal status reporting. The Agile approach, discussed in several chapters, is typically a low-ceremony approach.

Project Management Body of Knowledge (PMBOK)

The Project Management Institute (PMI) is a professional organization that promotes project management, primarily within the United States but also throughout the world. In addition, professional organizations in other countries promote project management. The PMI has a well-respected and rigorous

certification program, and many corporations encourage their project managers to become certified.

project management body of knowledge (PMBOK) a project management guide and standard of fundamental project management principles

As part of its mission, the PMI has defined a body of knowledge for project management. This body of knowledge, referred to as the **project management body of knowledge (PMBOK)**, is a widely accepted foundation of information that every project manager should know. The PMBOK is organized into these nine knowledge areas:

- Project Scope Management—Defining and controlling the functions that are to be included in the system as well as the scope of the work to be done by the project team
- Project Time Management—Creating a detailed schedule of all project tasks and then monitoring the progress of the project against defined milestones
- Project Cost Management—Calculating the initial cost/benefit analysis and its later updates and monitoring expenditures as the project progresses
- Project Quality Management—Establishing a comprehensive plan for ensuring quality, which includes quality control activities for every phase of a project
- Project Human Resource Management—Recruiting and hiring project team members; training, motivating, and team building; and implementing related activities to ensure a happy, productive team
- Project Communications Management—Identifying all stakeholders and the key communications to each; also establishing all communications mechanisms and schedules
- Project Risk Management—Identifying and reviewing throughout the project all potential risks for failure and developing plans to reduce these risks
- Project Procurement Management—Developing requests for proposals, evaluating bids, writing contracts, and then monitoring vendor performance
- Project Integration Management—Integrating all the other knowledge areas into one seamless whole

As you progress in your career, you would be wise to keep a record of the project management skills you observe in others as well as those you learn from your own experiences. One place to start is with the set of skills a systems analyst needs, as described in earlier chapters. A good project manager knows how to develop a plan, execute it, anticipate problems, and make adjustments. Project management skills *can* be learned.

Agile Project Management (APM)

In the last chapter, you learned about the Agile approach to developing systems and the four values of Agile development, which tended to prefer flexibility over plans and defined procedures. Obviously, these values have a large impact on the way a project is managed. However, one of the concerns with them is that they imply a working environment that has no controls or plans—one that can turn into pure chaos. In Chapter 8, we introduced a term, *chaordic*, that describes a project that expects and allows chaos while remaining controlled or ordered.

Agile project management is still a young discipline, and the IT industry is still learning how best to balance the flexibility and chaos of an Agile team with the order and control needed for a project. More than anything else, Agile project management is a way of balancing these two conflicting requirements: how to be agile and flexible while maintaining control of the project schedule, budget, and deliverables.

To help you understand Agile project management better, we will now go through five of the nine knowledge areas of the PMBOK and discuss the issues involved in implementing them by using Agile principles.

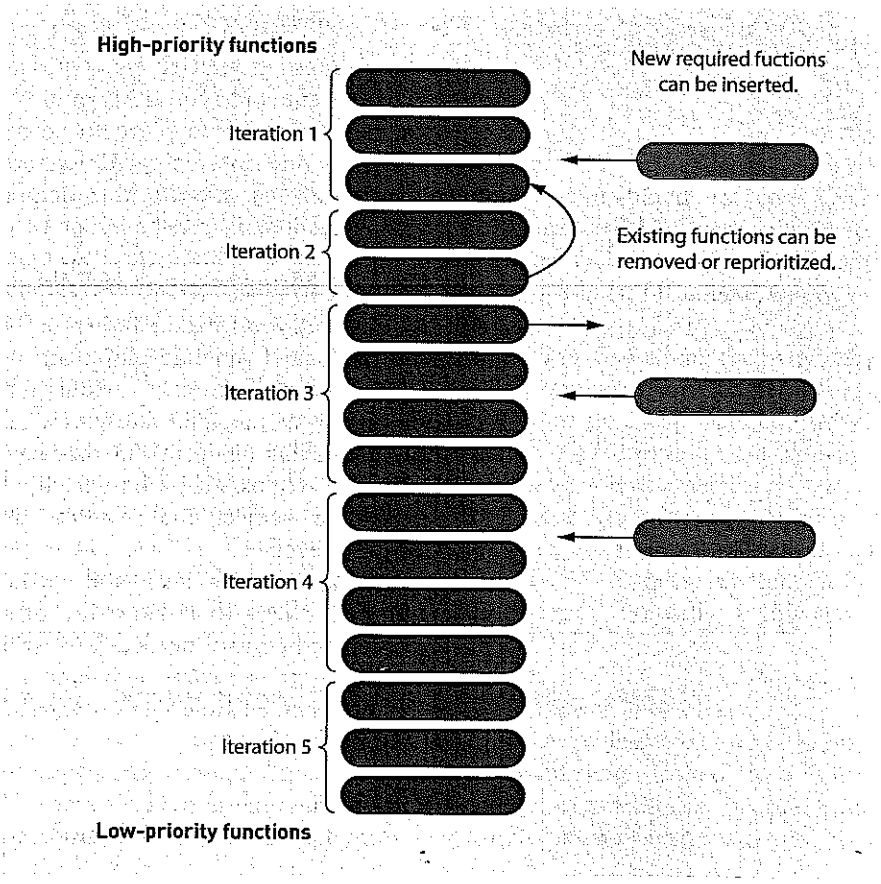
Agile Scope Management

Scope management refers to the scope of the new system and the scope of the project. In traditional predictive projects, the project manager and the team attempted to define the scope in both areas at the beginning of the project, during the planning phase. Unfortunately, for most new systems, there were so many unknowns that the scope was almost never defined accurately. The Agile philosophy accepts the fact that the scope isn't well understood and that there will be many changes, updates, and refinements to the requirements as the project progresses. However, uncontrolled scope can result in a project that never finishes, even if it is an Agile project. The project manager must have a process and mechanisms in place to control the scope of the project. How can this be done?

Let us assume that one of the major outcomes of the planning iteration was the decision to develop a prioritized list of business requirements that the new system needs to support. **Figure 9-3** represents this list, with the higher-priority items toward the top and the lower ones toward the bottom. These requirements can be prioritized by using several criteria, including importance to the business, risk, complexity, size, and other dependencies. In most projects, some combination of these criteria is used to prioritize the requirements. Figure 9-3 also indicates that the project team has made a preliminary assignment of these requirements to iterations. As new requirements are defined, they are prioritized, inserted into the stack, and assigned to an iteration.

Controlling the scope is a decision made by the client, with input provided by the project team and the users. With an iterative project, a deliverable is usually provided at the end of each iteration. Because the system is growing throughout the project, with the highest priority requirements implemented first, the client is able to shut down the project when he or she feels that the

FIGURE 9-3
Scope management with changing requirements



system is complete enough to satisfy the business need. Most projects usually require one or two more iterations to do final integration and testing to ensure that the system will scale for high volume and that it meets all the "hardening" requirements for security purposes.

Agile Time Management

Traditional time management is primarily concerned with scheduling tasks: creating the schedule, assigning work according to the schedule, and monitoring progress against the schedule. In predictive projects, the schedule is created during the initial planning phase and entered into a project scheduling system, such as Microsoft Project.

In an Agile project, because the requirements are always changing, it can be very difficult to create and maintain a meaningful project schedule. The initial planning effort will usually include the beginning set of requirements and divide the project into iterations, with a preliminary assignment of requirements to iterations. However, it is expected that the number of iterations and the assignments will change as new requirements are discovered and put on the prioritized stack.

Within an iteration, which often lasts from two to four weeks, a more detailed schedule can be developed. The Agile philosophy includes the idea that only for small work projects, in which the tasks are performed at nearly the same time (i.e., within one iteration), can a meaningful schedule be developed. In addition, the project team, not the project manager or team leader, will schedule its own work. Thus, for an Agile project, each iteration is usually planned as the first task within the iteration. The tasks are identified, estimates of the effort are developed, and work is assigned by the project team members. Because there are so many iterations in a project, the project team gets lots of practice and quickly becomes proficient at estimating and scheduling the work.

Agile Cost Management

It is normal for the client stakeholder to ask "How long will it take and how much will it cost for this new system to be developed?" These questions are hard to answer. For predictive projects, the project manager gives estimates, but as we saw earlier, these are usually incorrect. Agile project managers admit more readily that time and cost estimates are difficult to make, especially with a project in which the requirements are expected to change throughout. Hence, estimating the project's cost isn't as important as controlling the cost during the life of the project. The project manager's responsibility to control costs is just as important for an Agile project as it is for a traditional predictive project.

Agile Risk Management

In most adaptive, iterative projects, including Agile projects, close attention is given to project risks, particularly technical risks. Iterative projects are often risk-driven, meaning that early iterations focus specifically on addressing the most critical project risks. Although a similar emphasis on risk can be included in a predictive project, it is more difficult to integrate specific risk-reducing activities into the project schedule. The major difference between the two types of projects is that in predictive projects, separate prototypes are built, whereas in adaptive projects, the high-risk portions of the new system are built first.

Agile Quality Management

Usually, quality management has to do with the quality of the deliverable from the project. However, in an Agile project, we also consider the quality of the process. How well is the project working, and how well do the internal procedures promote project success?

In a predictive project, the final set of tasks consists of the system test, the integration test, and the user acceptance test. However, scheduling these

extensive tests at the end of the project renders it very difficult and expensive to make the necessary changes. An alternative is to deploy the system with minimal testing, which helps the budget but can cause many problems for the company.

In an Agile project, each iteration has a deliverable. Often, each iteration also integrates a new piece into the growing total system. Within each iteration, the new pieces are tested by themselves and as integrated with the rest of the system. The users also get involved in testing the system's ability to meet their business needs. Hence, testing and quality control are spread across the entire project and usually provide a better-tested and more robust system.

Another kind of quality control that should be done as part of an Agile project is a process evaluation at the end of each iteration. In other words, the project team does a self-evaluation to figure out how well it did and what could be done to improve the next iteration.

Activities of Core Process 1: Identify the Problem and Obtain Approval

The adaptive SDLC used in this text includes six core processes. Chapter 2 outlined the activities of Core Process 3 ("Discover and understand details"), and Chapter 6 outlined the activities of Core Process 4 ("Design system components"). In this chapter, we discuss the activities of Core Processes 1 and 2.

Core Process 1 is probably the most critical process for project success. As was noted in the Standish report, establishing such things as strong executive support, clear business case and direction, and effective planning is critical to project success. These important factors are identified and resolved during the activities of Core Process 1. **Figure 9-4** highlights the four activities associated with Core Process 1.

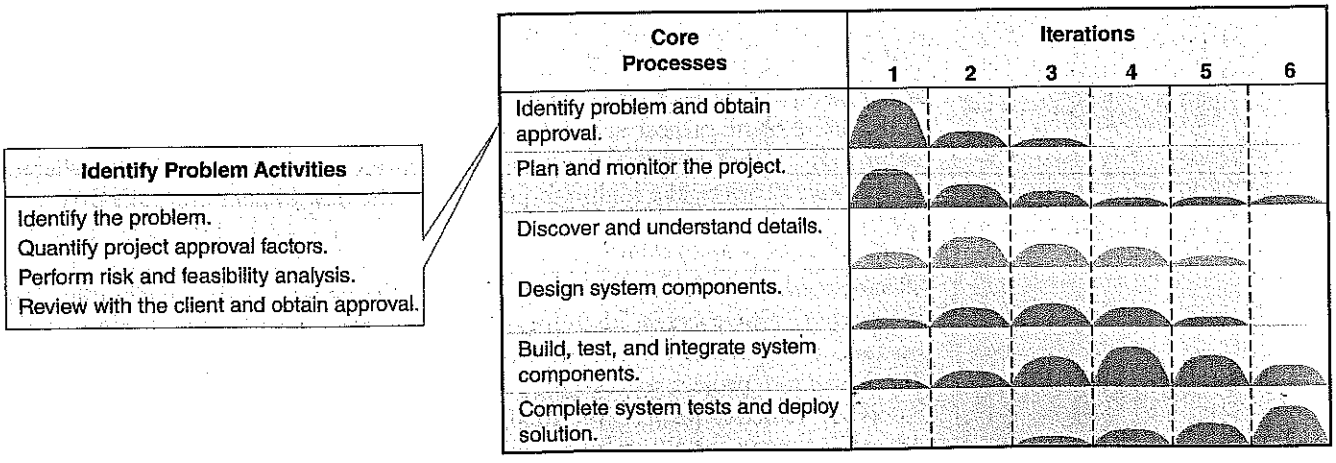
Identify the Problem

Information system development projects are initiated for various reasons, including: (1) to respond to an opportunity, (2) to resolve a problem, and (3) to respond to an external directive.

Most companies are continually looking for ways to increase their market shares or open up new markets. One way they create opportunities is with strategic plans—short term and long term. In many ways, planning is the optimal way to identify new projects. As the strategic plans are developed, projects are identified, prioritized, and scheduled.

Projects are also initiated to resolve immediate business problems. Such projects can be initiated as part of a strategic plan, but they are more commonly

FIGURE 9-4 *Activities of Core Process 1*



requested by middle managers who want to take care of some difficulty in the company's operations. Sometimes, these needs are so critical that they are brought to the attention of the strategic planning committee and integrated into the overall business strategy. At other times, an immediate need can't wait, such as a new sales commission schedule or a new report needed to assess productivity. In these cases, managers of business functions will request the initiation of individual development projects.

Finally, projects are initiated to respond to outside directives. One common version of this is legislative changes that require new information gathering and reporting—for example, changes in tax laws and labor laws. Legislative changes can also expand or contract the range of services and products that an organization can offer in a market. The regulatory changes in the telecommunications industry have opened doors for cable TV and telephone companies, which are vying for opportunities to provide cellular services, Internet access, and personalized entertainment.

Identifying and carefully defining the problem is a critical activity for a successful project. The objective is to ensure that the new system actually meets the business need. The purpose is to precisely define the business problem and determine the scope of the new system. This activity defines the target you want to hit. If the target is ill defined, all subsequent activities will lack focus. For example, a request might be made for a system that would "keep track of salesperson commissions." Without knowing more about the context surrounding this request, a system could be built that only records the commissions, ignoring the complexities of tax reporting, internal-versus-outside salespersons, deferred commissions, complex relationships, shared commissions, and so forth. Thus, even though all the specifications may not be defined in this initial activity, enough defining needs to be done to understand most of the implications of the required solution.

System Vision Document a document to help define the scope of a new system

An effective way to define the problem is to develop a **System Vision Document**, which was introduced in Chapter 1. There are three components to this document: the problem description, the anticipated business benefits, and the system capabilities.

The first task in developing a System Vision Document is to review the business needs that initiated the project. If the project was initiated as part of the strategic plan, then the planning documents need to be reviewed. If the project originated from departmental needs, then key users need to be consulted to help the project team understand the business need. From this task, a brief problem description is developed. As these needs are identified, the team also develops a detailed list of the expected business benefits. The list of **business benefits** contains the results that the organization anticipates it will accrue from a new system. Business benefits are normally described in terms of the specific results that can change the financial statements, either by decreasing costs or increasing revenues.

business benefits the benefits that accrue to the organization; usually measured in dollars

As the business benefits are being identified, the project team will identify the new system's specific capabilities to support the realization of these benefits. The objective of this task is to define the scope of the problem in terms of the requirements for the information system. This scoping statement, as defined by a list of **system capabilities**, helps identify the size and complexity of the new system and the project that will be required.

system capabilities the required capabilities of a new system; part of a System Vision Document

Members of the development team, working with the users and the client, combine these three components—the problem description, the business benefits, and the system capabilities—into a System Vision Document. **Figure 9-5** presents RMO's System Vision Document. Note the differences between the business benefits and the system capabilities. The business benefits focus on the financial benefit to the company. The system capabilities focus on the system itself. The benefits are achieved through the capabilities provided by the system.

FIGURE 9-5 *System Vision Document for RMO's CSMS*

Consolidated Sales and Marketing System
System Vision Document

RIDGELINE MOUNTAIN OUTFITTERS

Problem Description

Sales and marketing on the Web has changed drastically since the CSS system was built. Customers are more sophisticated, and they are used to catalog and sales systems that are easy to use and provide many services, such as one-click ordering, deferred-purchase tracking, simplified searches, and comparison shopping. In addition, research has shown that sales increase dramatically when social media marketing tools are combined with basic sales functionality. Hence, the new CSMS is needed not only to respond to today's competition but to launch RMO into today's world of social media and mobile computing. The longer RMO delays in starting this project, the more opportunities it misses.

System Capabilities

This document identifies the required system capabilities at a high level. Later documents will specify the detailed requirements. These capabilities are required:

- Provide a shopping cart capability.
  - Support customer sales with high automation (one-click, etc.).
  - Recommend related product purchases and comparison shopping.
  - Allow customer ratings and recommendations.
  - Include "friend" network capability.
- Include comprehensive order fulfillment.
  - Support multiple and split-order shipping and tracking.
  - Support back-ordering and tracking.
  - Allow customer comments and feedback.
- Provide customer account and billing capability.
  - Provide individualized customer accounting.
  - Support electronic billing and many electronic payment methods.
  - Accumulate customer "points" and allow transfer and sharing.
- Include marketing functions for promotions and specials.
  - Provide flexible promotions and sales.
  - Accumulate and track "points" from suppliers directly to customers.
  - Interface with social marketing media for advertising and social marketing activities.
  - Support mobile devices for social marketing and sales.

Business Benefits

The primary business benefit of these capabilities will be to increase sales by connecting with customers and improving the customer experience. The specific benefits include:

- Increasing the size of customer purchases
- Increasing the frequency of customer purchases
- Increasing customer satisfaction
- Increasing product recommendations from customers to friends
- Attracting new customers through recommendations and social marketing
- Building customer loyalty with recommendations and service
- Increasing speed of product availability
- Eliminating shipping delays and outages

RMO's existing CSS system was built under a tight deadline, and the company recognized that it would have a fairly short life. There were still many things to learn about Web marketing, but the existing CSS system will help the company define the requirements for its CSMS system.

Quantify Project Approval Factors

The first activity produced a high-level overview document that identified the need for a new system. However, that document alone may not be adequate to receive approval and funding. During this second activity, the project team, working with the users, will attempt to define more precisely the scope and impact of the project.

The objective is to provide sufficient justification so funds will be released and the project can start. Sometimes, the need is so great or so obvious that project approval is almost automatic. In other situations, it may be necessary to prepare a thorough cost-benefit analysis. These criteria must frequently be considered to obtain project approval:

- The estimated time for project completion
- The estimated cost for the project and system
- The anticipated benefits from the deployment of the new system

These are rough estimates. In the traditional predictive approach to systems development, estimates were often made with a considerable amount of detail. However, the estimates were usually far off the mark. The problem was, of course, that with most new systems, the team was venturing into unknown needs, requirements, and technologies. With the more adaptive approaches, the stakeholders recognize that the requirements are unknown and that it is more important to monitor and control scope, cost, and schedule than to try to make estimates.

The Estimated Time for Project Completion

During Core Process 2 ("Plan and monitor the project"), a more detailed project schedule is created. During project initiation, there usually isn't enough known about the project to create a schedule. But there is nevertheless a need to estimate the project's completion date, even though this is one of the hardest things to do.

Sometimes, there are business constraints that dictate the completion of the project. For example, new legislative requirements may affect the deployment date. A window of opportunity may also provide a powerful motivation to complete a project at a specific time. These considerations should be made manifest and considered in the project approval and project planning processes.

The major inputs toward estimating the project completion date are the scoping document and the amount of effort required to develop the listed requirements. As indicated earlier, it is difficult to make an estimate with any degree of accuracy. At this early point in the project, gross estimates of team size and time frame are usually the best that can be achieved. For a predictive approach, the list of requirements can serve as the starting point for estimating the effort required to define and develop a particular function. For an adaptive approach, the same information can be used to estimate the number of iterations required and the size and number of teams working on the various subsystems.

Figure 9-6 shows an example of a time estimate document for RMO.

For RMO, the development of the time estimate was a one-day exercise. Because the project didn't yet have approval or funding, neither a project manager nor any systems analysts had been assigned to the project. However, a project manager had been assigned to obtain approval, and two systems analysts were assigned to help him. These three experienced technical people

FIGURE 9-6
Project completion date estimate for the CSMS project

Time Estimate for the New CSMS Project			
Subsystem	**Functional requirements**	**Iterations required**	**Estimated time**
Sales subsystem*	15	5	20 weeks
Order Fulfillment subsystem*	12	5	20 weeks
Customer Account subsystem**	10	4	15 weeks
Marketing subsystem**	6	3	13 weeks
Reporting subsystem**	7	3	12 weeks
Total development time (2 teams)			40 weeks
Final hardening and acceptance testing		2	8 weeks
Total project time			48 weeks

*Assigned to Tiger team
**Assigned to Cougar team

met for four hours with the key users from the various RMO departments. The object of these meetings was to build a comprehensive list of all the functional requirements from each department. After the meetings, the group met again to organize this list of requirements into groupings that could be assigned to various iterations for the development of the software.

An assumption that was made by the director of new development was that there would be two subteams of four people each allocated to this project. As indicated in Figure 9-6, the time estimate for this project is 48 weeks from the date it begins.

The Estimated Cost for the Project and System

The estimated costs of developing the new CSMS are shown in **Figure 9-7**. By far, the largest cost item in the project's budget is the salaries of the project team. Other cost elements include the cost of the new computers, training for the users, offices, facilities, and utilities for the project team, travel expenses for the project team to do site visits, and software licenses. As you can see, this estimate is a little over $1.5 million.

After the system has been put into production, there will be annual operating costs, as shown in **Figure 9-8**. The largest cost is for a hosting service to provide some of the equipment, the connection to the Internet, and server administration services. These estimated costs were based on RMO using a hosting service to provide the equipment, the connection to the Internet, and server

FIGURE 9-7
Summary of development costs for CSMS

Summary of Development Costs for CSMS	
Expense category	**Amount**
Salaries/wages (includes benefits costs) (1 PM, 8 analysts, 1 support)	$936,000.00
Equipment/installation	$308,000.00
Training	$78,000.00
Facilities	$57,000.00
Utilities	$97,000.00
Travel/miscellaneous	$87,000.00
Licenses	$18,000.00
Total	**$1,581,000.00**

FIGURE 9-8
Summary of estimated annual operating costs for CSMS

Summary of Estimated Annual Operating Costs for CSMS	
Recurring expense	**Amount**
Connectivity/hosting	$156,000.00
Programming	$75,000.00
Help desk	$90,000.00
Total	**$321,000.00**

administration. The project team estimated about $13,000 a month for those expenses, which is enough for 15 very large managed servers. This appeared to be more than adequate, depending on the traffic volume. Other costs were for one full-time programmer and two help desk personnel.

The Anticipated Benefits from the Deployment of the New System

The System Vision Document identifies the anticipated business benefits of the new system. In this task, we analyze those business benefits and provide an estimate of their value to the organization. This value becomes part of the total decision criteria. Obviously, the dollar amount associated with these savings or revenues must be estimated by the client. It isn't the project manager's job to predict the value of business benefits. However, the project manager can help the client identify categories of potential benefits. Typical areas of increased revenue or cost reduction benefits include:

- Opening up new markets with new services, products, or locations
- Increasing market share in existing markets
- Enhancing cross-sales capabilities with existing customers
- Reducing staff by automating manual functions or increasing efficiency
- Decreasing operating expenses, such as shipping charges for "emergency shipments"
- Reducing error rates through automated editing or validation
- Reducing bad accounts or bad credit losses
- Reducing inventory or merchandise losses through tighter controls
- Collecting receivables (accounts receivable) more rapidly

The project team at RMO worked with the vice president of sales and marketing to identify benefit areas and estimate a value for each one. This size of an investment and ongoing expense was going to require board approval within RMO. The board will want to know what the benefits of the new system will be and what the return on the investment will be. One of the difficulties for RMO is to determine how to assign a value to a benefit. A typical question might be "Do we assign the value of all our sales given that this system is needed to stay competitive in the marketplace? Or do we assign only the value of the increased sales we expect to get from marketing and higher volume?" If sales will drop because RMO becomes less competitive in the marketplace, the total sales value could be used. However, if the existing system is good enough to maintain a good client base, then only the increased sales should be used. These kinds of decisions are made by the client, not the project team. In this case, the vice president of sales and marketing at RMO decided to use a more conservative estimate. **Figure 9-9** summarizes the estimates he generated.

cost/benefit analysis process of comparing costs and benefits to see whether investing in a new system will be beneficial

net present value (NPV) the present value of dollar benefits and dollar costs of a particular investment

Many organizations like to compare the estimated costs with the anticipated benefits to calculate whether the benefits outweigh the costs. This process is called a **cost/benefit analysis**. Companies use a combination of methods to measure the overall benefit of the new system. One popular approach is to determine the **net present value (NPV)** of the new system. The two concepts behind net present value are (1) that all benefits and costs are calculated in terms of today's dollars (present value) and (2) that benefits and costs are

FIGURE 9-9
Estimated annual benefits for CSMS

Estimated Annual Benefits for CSMS	
Benefit or cost saving	**Amount**
Recapture/prevention of lost sales	$200,000.00
Increase sales to existing customers	$300,000.00
Sales to new customers	$350,000.00**
Increased efficiency in order processing	$50,000.00
Reduction of data center and equipment costs because of hosting	$146,000.00
Total	**$1,046,000.00**

**plus 8% annual growth

combined to give a net value. The future stream of benefits and costs are netted together and then discounted by a factor for each year in the future. The discount factor is the rate used to bring future values back to current values.

Figure 9-10 shows a copy of the NPV calculation done for RMO's new CSMS. There are various techniques for calculating the NPV of a given investment. In this example, Year 0 represents the development period prior to the deployment of the system. The annual benefits for each year are extended across the top row. The development costs are shown on the second row. Annual expenses are shown on the third. Those three rows are combined in the fourth row to give the net benefits and costs. The fifth row shows the discount value, given a 6 percent discount rate. The sixth row is the product of the fourth and fifth rows and represents the net value in terms of today's dollars (i.e., the NPV). The seventh row shows a cumulative total of annual NPVs.

break-even point the point in time at which dollar benefits offset dollar costs

payback period the time period after which the dollar benefits have offset the dollar costs

In Figure 9-10, the numbers in the seventh row eventually change from negative to positive. The point in time when that happens is called the **break-even point**. The length of time before the break-even point is reached is called the **payback period**. The payback period occurs in the year that the cumulative value goes positive. To calculate it, first take the last year that the cumulative value is negative—in this case, Year 2. Add to that year the number of days in the following year (in this case, Year 3) that it takes for the cumulative value to go positive. The method for doing that is to take absolute values of the ending value in Year 2 divided by the sum of the absolute values for the end of Year 2 and Year 3—in this case, 226,865 divided by (226,865 + 430,743). Here, that calculation indicates that the cumulative value goes positive after 35 percent of the year has passed. Multiply .35 times the 365 days in the year to get 128 days into Year 3. Many companies require a payback period of two to three years on new software.

tangible benefit a benefit that can be measured or estimated in terms of dollars

The previous cost/benefit calculation depends on an organization's ability to quantify the costs and benefits. If it can indeed estimate a dollar value for a benefit or a cost, the organization treats that value as a **tangible benefit** or

FIGURE 9-10 *Five-year cost/benefit analysis for CSMS*

	A	B	C	D	E	F	G	H
1				RMO Cost/Benefit Analysis for CSMS				
2		Category	Year 0	Year 1	Year 2	Year 3	Year 4	Year 5
3	1	Value of benefits		$1,046,000	$1,074,000	$1,104,240	$1,136,899	$1,172,171
4	2	Development costs	-$1,581,000					
5	3	Annual expenses		-$321,000	-$321,000	-$321,000	-$321,000	-$321,000
6	4	Net benefit/costs	-$1,581,000	$725,000	$753,000	$783,240	$815,899	$851,171
7	5	Discount factor	1.0000	0.9434	0.8900	0.8396	0.7921	0.7473
8	6	Net present value	-$1,581,000	$683,965	$670,170	$657,608	$646,274	$636,080
9	7	Cumulative NPV	-$1,581,000	-$897,035	-$226,865	$430,743	$1,077,017	$1,713,097
10	8	Payback period	2 years +	226865 / (226865+430743) = .35			or 2 years + 128 days (.35*365)	

cost. However, in many instances, an organization can't measure some of the costs and benefits to determine a value. Never discount the importance of ascertaining the "behind the scenes" reasons for a project. There may be political reasons for or against the project that override all other feasibility analyses. If there is no reliable method for estimating or measuring the value, it is considered an **intangible benefit.** In some instances, the importance of the intangible benefits far exceeds the tangible costs—at least in the opinion of the client, who pursues developing the system even though the dollar numbers don't indicate a good investment.

intangible benefit a benefit that accrues to an organization but that can't be measured quantitatively or estimated accurately

Examples of intangible benefits include:

- Increased levels of service (in ways that can't be measured in dollars)
- Increased customer satisfaction (not measurable in dollars)
- Survival
- Need to develop in-house expertise (such as a pilot program with new technology)

Examples of intangible costs include:

- Reduced employee morale
- Lost productivity (the organization may not be able to estimate it)
- Lost customers or sales (during some unknown period of time)

Determining Project Risk and Feasibility

Project risk and feasibility analysis verifies whether a project can be started and completed successfully. Because each project is a unique endeavor, every project has unique challenges that affect its potential success.

The objective of this activity is to identify and assess the potential risks to project success and to take steps to eliminate or at least ameliorate these risks. They should be identified during the project approval process so all stakeholders are aware of the potential for failure. The team can also establish plans and procedures to ensure that those risks don't interfere with the success of the project. Generally, the team assigns itself these tasks when confirming a project's feasibility:

- Determine the organizational risks and feasibility.
- Evaluate the technological risks and feasibility.
- Assess the resource risks and feasibility.
- Identify the schedule risks and feasibility.

Determine Organizational Risks and Feasibility

Each company has its own culture, and any new system must be accommodated to that culture. There is always the risk that a new system departs so dramatically from existing norms that it can't be successfully deployed. The analysts involved with feasibility analysis should evaluate organizational and cultural issues to identify potential risks for the new system. Such issues might include:

- Substantial computer phobia
- A perceived loss of control on the part of staff or management
- Potential shifting of political and organizational power due to the new system
- Fear of change of job responsibilities
- Fear of loss of employment due to increased automation
- Reversal of long-standing work procedures

It isn't possible to enumerate all the potential organizational and cultural risks that exist. The project management team needs to be very sensitive to the reluctance within the organization to identify and resolve these risks.

After identifying the risks, the project management team can take positive steps to counter them. For example, the team can hold additional training sessions to teach new procedures and provide increased computer skills. Higher levels of user involvement in developing the new system will tend to increase user enthusiasm and commitment.

Evaluate Technological Risks and Feasibility

Generally, a new system brings new technology into the company, even state-of-the-art technology. Other projects use existing technology but combine it into new, untested configurations. If an outside vendor is providing a capability in a certain area, the client organization usually assumes the vendor is expert in that area. However, even an outside vendor may find the requested level of technology too complicated.

The project management team needs to carefully assess the proposed technological requirements and available expertise. When these risks are identified, the solutions are usually straightforward. The solutions to technological risks include providing additional training, hiring consultants, or hiring more experienced employees. In some cases, the scope and approach of the project may need to be changed to ameliorate technological risk. The important point is that a realistic assessment will identify technological risks early, making it possible to implement corrective measures.

Assess Resource Risks and Feasibility

The project management team must also assess the availability of resources for the project. The primary resource consists of team members. Development projects require the involvement of systems analysts, system technicians, and users. Required people may not be available to the team at the necessary times. An additional risk is that people assigned to the team may not have the necessary skills for the project. After the team is functioning, members may have to leave the team. This threat can come either from staff who are transferred within the organization if other special projects arise or from qualified team members who are hired by other organizations. Although the project manager usually doesn't like to think about these possibilities, skilled people are in short supply and sometimes do leave projects.

The other resources required for a successful project include adequate computer resources, physical facilities, and support staff. Generally, these resources can be made available, but the schedule can be affected by delays in the availability of these resources.

Identify Schedule Risks and Feasibility

The development of a project schedule always involves high risk. Every schedule requires many assumptions and estimates without adequate information. For example, the needs (and, hence, the scope) of the new system aren't well known. Also, the time needed to research and finalize requirements has to be estimated. The availability and capability of team members aren't completely known.

Another frequent risk in developing the schedule occurs when upper management decides that the new system must be deployed within a certain time. Sometimes, there is an important business reason for setting a fixed deadline, such as RMO's need to complete the CSS in time for online ordering over the holidays. Similarly, universities require the completion of new systems before key dates in the university schedule. For example, if a new admissions system isn't completed before the admissions season, then it might as well wait another full year. In cases like these, schedule feasibility can be the most important feasibility factor to consider.

If the deadline appears arbitrary, the tendency is to create the schedule to show that it can be done. Unfortunately, this practice usually spells disaster. The project team should create the schedule without any preconceived notion of

required completion dates. After the schedule is completed, comparisons can be done to see whether timetables coincide. If not, the team can take corrective measures, such as reducing the scope of the project, to increase the probability of the project's on-time completion.

One objective of defining milestones and iterations during the project schedule is to permit the project manager to assess the ongoing risk of schedule slippage. If the team begins to miss milestones, the manager can possibly implement corrective measures early. Contingency plans can be developed and carried out to reduce the risk of further slippage.

Review with Client and Obtain Approval

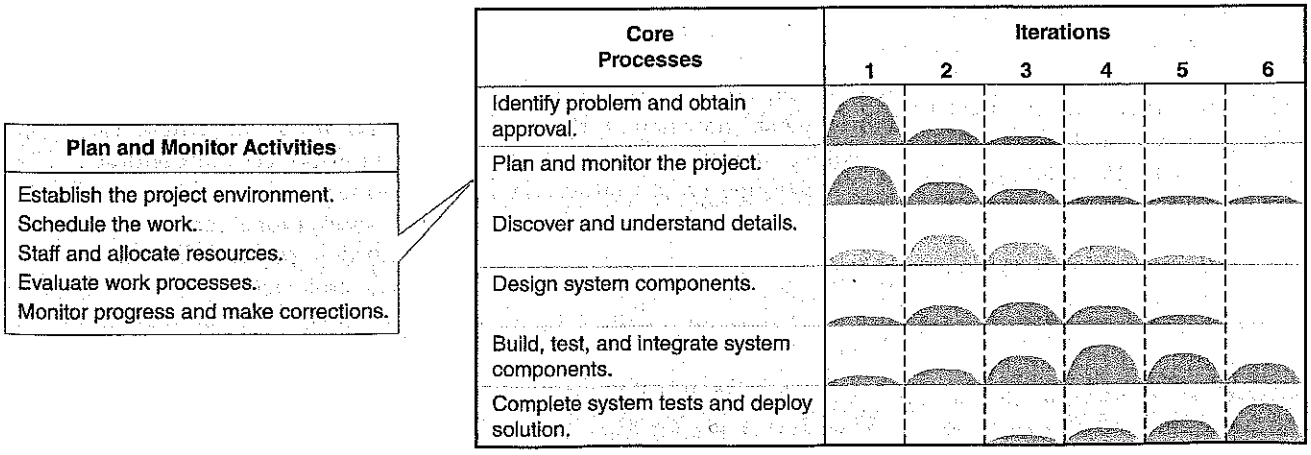
As mentioned earlier, the amount of expenditure for the RMO project required board approval. However, before a presentation could be given to the board, RMO's executive committee needed to understand and agree to the project. A project this size has major impacts on all areas of the company. The departments, such as sales and marketing, will be directly impacted. They will have to allocate staff and resources to help in defining the requirements, developing test cases, and testing the new system as it is developed. In other words, the people in this department will have extra duties for the next 12 months or so. Even departments not directly involved will need to support this heavy development activity, perhaps tightening their budgets. In any event, it is always good policy to get the approval and support of the entire company. This process starts by making presentations to the senior executives of RMO. Often, a project manager will be asked to make the presentation or at least be present to answer questions.

After the executive committee approves the project, it goes to the board. After board approval, the IT department begins to assign full-time resources to the project. It is also a good idea at this point to have a company-wide memo or meeting to mark the beginning of this major activity. If the entire company knows that all the executives are supporting it and requesting cooperation, the project will proceed much more smoothly.

Activities of Core Process 2: Plan and Monitor the Project

This core process lasts throughout the entire project. A major planning effort occurs immediately after the project is approved. Ongoing planning and project monitoring continue during all project iterations. Not only must each iteration be planned as it starts, but progress must continually be monitored and corrective actions may be required. **Figure 9-11** illustrates by the height of the effort

FIGURE 9-11 *Activities of Core Process 2*



curve in each iteration that planning and monitoring activities must be an integral part of every project iteration. The specific activities associated with this core process are also listed in Figure 9-11. We will discuss each of these activities individually.

Establish the Project Environment

So far in this text, we have discussed different types of projects, such as predictive and adaptive projects, as well as tools, techniques, and methodologies to use with these different types of projects. We have also discussed such concepts as ceremony, project reporting, stakeholders, user participation, and the project team work environment. All these elements must be put in place as the project gets under way. Some of these decisions will already have been made based on the organization's standard policies and procedures. Others will be decided during the approval process. In any case, the project manager must ensure that the project's parameters and the work environment are finalized so the work of the project can proceed without roadblocks or delays. There are important project structure considerations that must be addressed as the project gets under way. For example, what kind of communication processes will be needed to keep the team and external stakeholders informed about what is going on? In addition, the members of the project team all need computers and IDEs and other tools to do their work. Of course, specific procedures about how the project team meets with the users, how they write code, and how they submit code for acceptance also need to be finalized. We will discuss three important considerations:

- Recording and communicating—internal/external
- Work environment—support/facilities/tools
- Processes and procedures

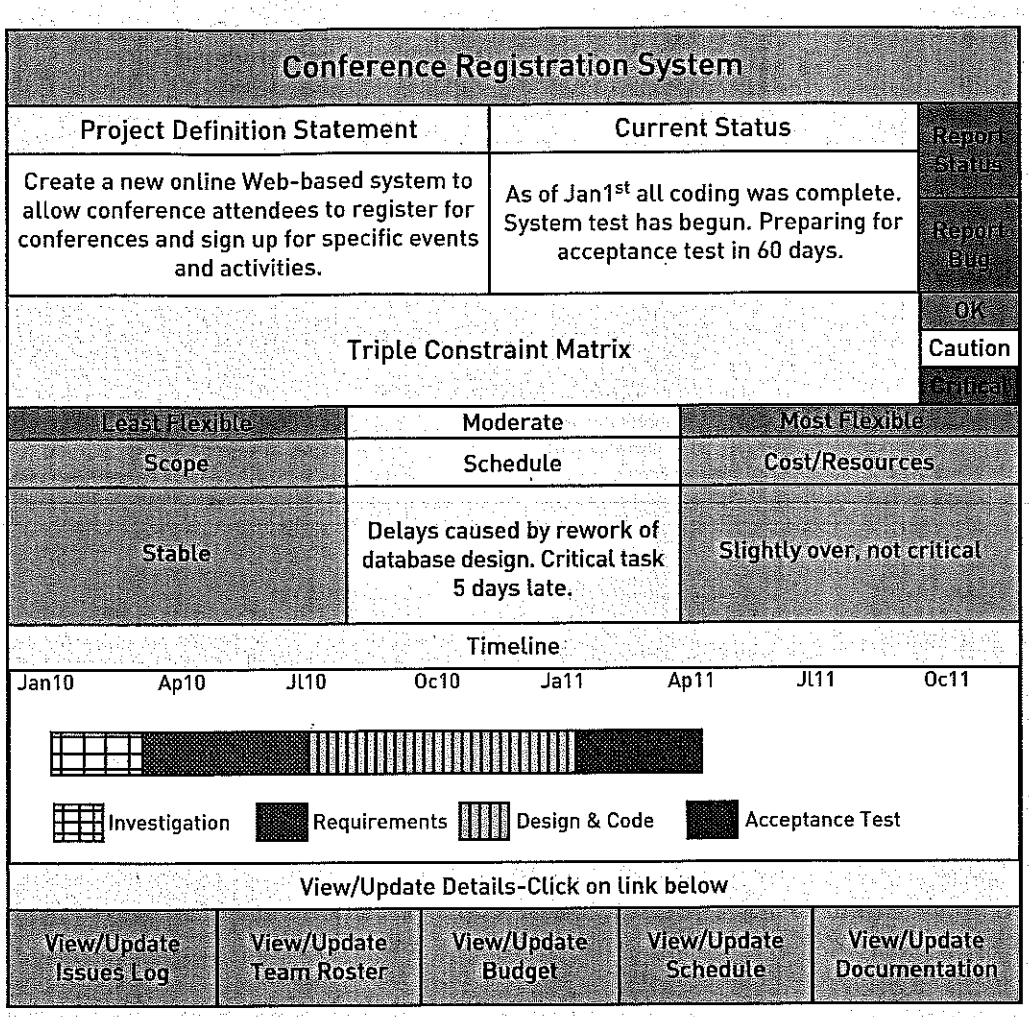
Recording and Communicating—Internal/External

The project manager and project team members will be involved in all types of meetings where decisions will be made and information developed. Determining what is important and how to record this information needs to be set out in specific project procedures. The other critical issue with information is what, how, how frequently, and to whom this information needs to be disseminated. One of the first tasks for a project manager on a new project is to establish the procedures and guidelines for how to handle the project's information.

A critical success factor for IT projects is to have the support of the organization's executives and other key stakeholders. A good project manager understands this need and structures his or her project so he or she communicates frequently, with the appropriate detail, to each of his or her stakeholders. Figure 9-2 identified the various stakeholders and participants in a project. Some of these stakeholders will be integrally involved with the project. Other stakeholders will be only marginally involved, receiving periodic status reports. The client stakeholders (the ones paying the project costs) will need to be kept aware of the project's status and of any difficulties or delays. A stakeholder analysis helps identify all those persons who have an interest in the project and defines what information they will want and need concerning the project. Generally, we refer to this as external reporting of project information.

Maintaining project information can be done via electronic means. Schedule information can be published to a Web site so everyone can view it. Another type of project-tracking tool, sometimes called a *project dashboard*, allows all types of project information to be posted and viewed by Web browsers. **Figure 9-12** is an example of a project dashboard system that allows easy access to project information. Spreadsheets, e-mails, newsletters, and list servers all provide ways to maintain, collect, and distribute information. Once the electronic systems are set up, they will often take care of themselves.

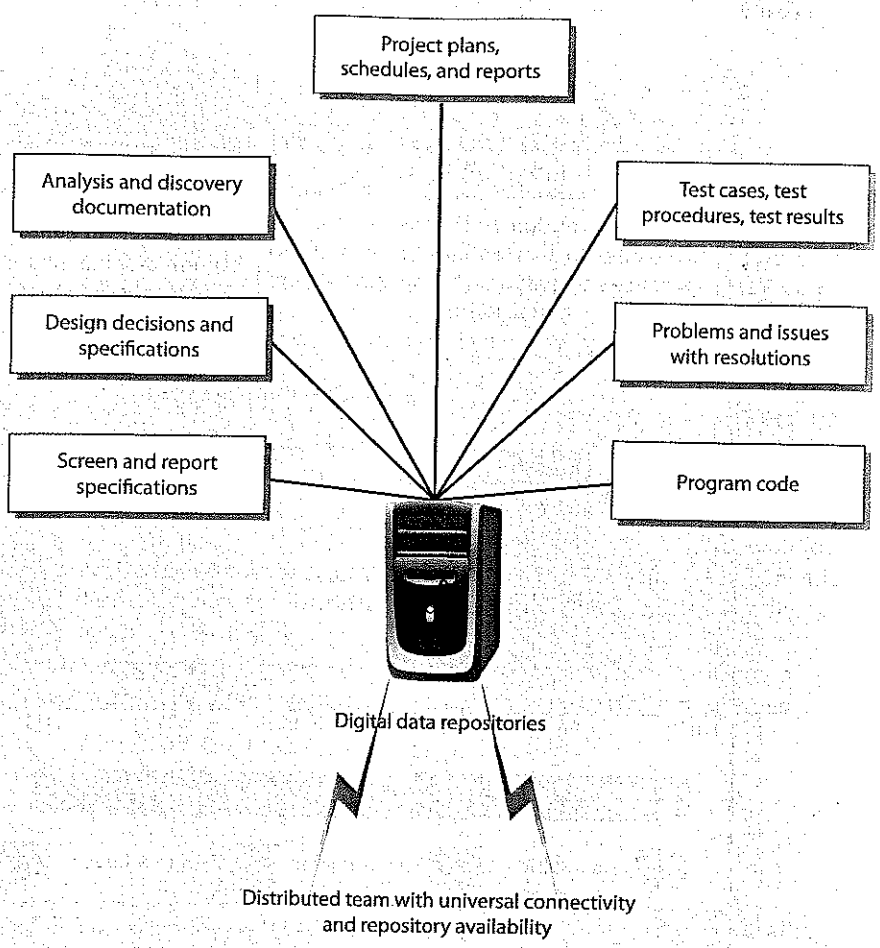
FIGURE 9-12 *Sample dashboard showing project information and status*



The members of the project team also need to have mechanisms in place to communicate among themselves and document project decisions. This is an entirely different type of information—information about the system under development. For example, during analysis activities, the project team documents the results of user meetings by using various means, such as writing use case descriptions. During design, information also needs to be recorded and distributed among members of the team as appropriate. During testing, when errors are found, they must be documented and assigned to programmers to be fixed. Finally, the entire recording and communication requirement is often made more critical by members of the project team (as well as the users) being located at various sites around the globe. **Figure 9-13** illustrates some of the information that may need to be captured and maintained. The data repository in the figure usually consists of many different types of data structures and storage techniques—from wikis to databases to issue-tracking systems.

There is one caveat related to recording and communicating. With traditional predictive projects, the tendency was to create reams and reams of documentation. As you learned in the previous chapter, adaptive projects that use the Agile philosophy emphasize code over documentation. A novice project manager may interpret that to mean that no documentation is required. However, even with an Agile approach, the basic user definitions need to be

FIGURE 9-13
System information stored in data repositories



documented for later verification. It isn't uncommon during programming for a programmer to have to refer back to notes and models to remember the exact details and decisions of a particular requirement. An experienced project manager knows the right amount of documentation so the project isn't overloaded with overhead but critical decisions are recorded.

It should be obvious that a comprehensive recording and communication scheme needs to be put in place. Fortunately, in today's connected world, there are many tools available so external and internal communication can be done easily. With so many electronic tools, all project information should be available online and accessible to all stakeholders. In fact, with the use of wikis, it is now common to allow many team members and even users to assist in the recording and updating of critical project information.

The CSMS team wanted to maintain its project information in digital format and have it available to all stakeholders, including team members, users, the client, and the members of the steering committee. RMO is a very open shop guided by the philosophy that information should be widely distributed. **Figure 9-14** shows all the tools that the CSMS project team uses to communicate and capture information. The core team members had previously worked on several Agile projects, so they had learned that there is a correct balance of documentation—not too much but enough to be able to trace key decisions and requirements. Barbara Halifax, the project manager, wanted to ensure the tools were in place so it was easy to record information when it was prudent to do so.

User documents, such as sample invoices, were scanned and placed in a document repository. User functional definitions were recorded in a

FIGURE 9-14
Electronic digital repositories of information for CSMS

Electronic Digital Repositories		
Information captured	**Electronic tools**	**Who can update/view**
User definitions and functions User documents	Forum software Document server Scanners	Analysts, users/all
Screens and reports layouts	Web design tools Visio PowerPoint/Keynote	Analysts, users/all
Design specifications and diagrams	Wiki software Visio	Analysts/all
Issues and outstanding problems	Issue-tracking software	Analysts, users/all
Program code	Apache subversion (SVN)	Analysts/analysts
Project schedule	MS project	Analysts/all
Project status and information	Forum software	Analysts, users/all
Daily team coordination meeting	Video laptop conferencing	Project team
Distributed team communication	IM chat with video	Project team
Project update newsletter	Blog software	Project manager/all

forum system. Using a forum allowed team members and users to update it when key issues were discussed and needed to be remembered. Sample screen and report layouts were either sketched out or drawn with Visio or Keynote. Hand sketches were often scanned and saved. Most design decisions and specifications went right into the program code and weren't documented. However, some decisions were global, and those were captured in a wiki.

Each day, the project team had a "stand up" meeting—a short coordination meeting. Most of the team members were in the Park City Center, but some users were assigned to the team from other locations. Sometimes, team members were visiting user sites and therefore not available, and there were some team members who worked in the Salt Lake City office. Therefore, the daily meeting was conducted as a video conference call, with each person using his or her webcam and personal computer. The meeting normally lasted about 15 minutes.

Finally, there was some discussion of sending out a biweekly newsletter about the progress on the project. Barbara felt that it was important for the entire company to stay informed about the project in order to encourage their enthusiasm and support. However, instead of a printed newsletter, she opted to do it in the form of a blog. All users were invited to sign up with an RSS feed to keep informed about the project's progress.

Work Environment

Although the work environment may relate more to the work processes of the project team, the project manager must ensure that it is adequate to allow the project team to work productively. There are five major components of the work environment:

- Personal computer(s) and/or workstation(s)
- Personal development software and tools
- Development server with repositories, sandboxes, and communication tools

- Office space, conference rooms, and equipment, including printers, scanners, and projectors
- Support staff

Most importantly, of course, is the computer equipment and other hardware that the team will need. Obviously, each developer will need his or her own computing configuration, which may consist of multiple computers or monitors. Other important hardware includes the development servers, printers, and internal development network. If the team is distributed, video cameras and projectors may be necessary to conduct distributed team meetings. Along with the hardware, resources must be made available to administer things such as the development server.

Related to the hardware is the computer software and other tools. Software tools can get quite elaborate—from stand-alone Integrated Development Environment (IDE) tools to modeling software to code repository software. The development server, with its environment and software, must also be configured and deployed. The server may be set up as a virtual server or as a stand-alone computer. Applications include such things as code repository, issue-tracking application, testing system, and the project dashboard.

Along with the hardware and software, a work configuration must be provided for each developer, with log-on permissions, sandbox environments, repository access, and so forth. The final two components are the office space and other facilities that may be needed. This will include access to conference rooms, presentation equipment, and maybe even transportation vehicles. Finally, the productivity of the team members is always enhanced when adequate support staff is available to take care of myriad details that always accompany an active project.

Processes and Procedures

The final major set of decisions has to do with the project's internal processes and procedures. Earlier, we discussed a project's level of formality. Larger projects require more elaborate reporting processes and meeting schedules. When there are many people involved, coordination of activities becomes critical. Procedures include:

- Reporting and documentation—What is done? How is it done? Who does it?
- Programming—Single or pair programming? How is work assigned? By whom?
- Testing—Programmer tests or user tests? How to mark items ready for testing?
- Deliverables—What are they? How and when are they handed over to users? How are they accepted?
- Code and version control—How is the code controlled to prevent conflicts? How to coordinate bug fixing with new development? How and when are deliverables released?

Schedule the Work

Scheduling the work is necessary for any size or type of project. However, the techniques used can vary widely depending on the type of project. For predictive, highly controlled projects, a detailed and complete schedule that covers the entire project is usually built. Again, these kinds of schedules only work because the software to be built is well understood. However, even in those projects with detailed and comprehensive schedules, accommodation is required as things change during the life of the project. At the other end of the spectrum, small Agile projects sometimes don't even have a project schedule, with the team members being responsible for scheduling their own work. Coordination

is accomplished by talking and keeping each other informed of what each person is working on. This is what is meant by *chaordic*.

Scheduling the work for many of today's projects lies somewhere between these two extremes. Large projects may have several independent teams of developers working on various subsystems. Even though the work between the teams is fairly independent, coordination is still required. Adaptive projects also anticipate additional requests and changes to the original scheduled tasks.

For adaptive types of projects, creating the project schedule is done throughout the life of the project. During the initial planning phase, the initial list of use cases or user stores are developed for each subsystem. The use cases are divided up and tentatively assigned to the iterations. Let us call this the **project iteration schedule**. As each iteration is begun, a detailed schedule of tasks and work to be done is developed. You saw an example of creating an iteration schedule in Chapter 1. Let us call this schedule a **detailed work schedule**, meaning that it schedules the work within an iteration. Sometime during each iteration—often as one iteration is finishing and before the next iteration begins—the project manager, with assistance from the team leaders and key users, will review and rework the project iteration schedule. During this process, the changes and any new requirements are prioritized and placed on the schedule.

project iteration schedule the list of iterations and use cases or user stories assigned to each iteration

detailed work schedule the schedule that lists, organizes, and describes the dependencies of the detailed work tasks

Creating the project iteration schedule must take into account the total size and configuration of the solution system and the number of teams available to work on the project. Separate lists of requirements are made by subsystem, and a project iteration schedule can then be made for each subsystem. Some tasks, such as designing the database, may go across all subsystems and need to be scheduled separately or be included in every subsystem list. **Figure 9-15** shows a sample project iteration schedule for the CSMS Sales subsystem. As you can see, the length of each iteration is fairly constant at around four weeks. All the identified tasks, which represent the requirements, have been assigned to iterations. In this case, we have identified five iterations.

Developing a detailed work schedule for a single iteration is a three-step process:

- Develop a work breakdown structure.
- Estimate effort and identify dependencies.
- Create a schedule by using a Gantt chart.

FIGURE **9-15**
Project iteration schedule for the CSMS Sales subsystem

Project Iteration Schedule for the CSMS Sales Subsystem		
Iteration	**Time estimate**	**Use cases assigned to iteration**
1	4 weeks	1. Search for item. 2. View detailed descriptions. 3. View rotating (3-D) images. 4. Compare item characteristics.
2	4 weeks	5. View comments and ratings. 6. Search comments and ratings for friends. 7. View accessory combinations (images). 8. Save item + accessories as "combo."
3	5 weeks	9. Add item (or combo) to shopping cart. 10. Remove item (or combo) from shopping cart. 11. Add item (or combo) to "on reserve" cart. 12. Remove item (or combo) from "on reserve" cart.
4	4 weeks	13. Check out active cart. 14. Create and process store sale. 15. Create and process phone sale.
5	3 weeks	16. Clean up, final test, harden site, tune database, etc.
Total	20 weeks	

work breakdown structure (WBS) the list or hierarchy of activities and tasks of a project; used to estimate the work to be done and to create a detailed work schedule

A **work breakdown structure (WBS)** is a list of all the required individual activities and tasks for the project. There are two general approaches for creating a WBS: by deliverable or by a timeline. The first approach identifies all the deliverables that must be completed for a given iteration. Then, the WBS identifies every task that is necessary to create each deliverable. The second approach works through the normal sequence of activities that are required for the final deliverable. Experienced developers who have worked on Agile projects understand the steps and tasks that are required to create a particular deliverable. Of course, each iteration is slightly different depending on the particular functions and deliverables that are included.

Figure 9-16 is a sample handwritten WBS for the first iteration of the Sales subsystem. The tasks have been partitioned according to the core processes

FIGURE 9-16 *Work breakdown structure for first iteration*

Work Breakdown Structure
Iteration 1 of Sales Subsystem

I. Project planning
Develop WBS and build schedule and then plan the work – 1/2 day.

II. Analysis tasks
Meet with sales department – 1 day.
Meet with marketing department – 1 day.
Define required information and data elements (share with Cougar team) – 1 day.
Model user activities – 1 day.

III. Design tasks
Design database schema (work with Cougar team) – 1 day.
Design screen layouts and cross links – 2 days.
Identify program classes and methods – 1 day.

IV. Build tasks
Build database (coordinate with Cougar team) – 1/2 day.
Write program code – 4 days.
Integrate 3-D imaging code – 2 days.
Build test data – 2 days.
Set up user "simulated live environment" – 1/2 day.
Perform acceptance tests with users – 2 days.
Release accepted version – 1/2 day.
Perform team introspection – 1/2 day.

Note: The use cases in this iteration require data and a database schema that is also used in building the catalog. Building the catalog will be done in the first iteration of the Marketing subsystem. The two iterations will be done in parallel, and the two teams (Tiger team and Cougar team) will coordinate and share information.

Note: This is a four-person team. The work pattern is (1) all together for fact finding and design tasks and (2) pairs for programming and testing tasks.

Note: The four use cases that will be developed during this iteration are:
1. *Search for item.*
2. *View detailed information for an item.*
3. *View rotating (3-D) image of item.*
4. *Compare characteristics and prices across multiple items.*

Planning, Analysis, Design, and Building. In the figure, each task also has an estimate of the time required. Sometimes, two estimates are provided: the effort required and the expected duration. The effort required is given in person-days of work, and the duration is a measure of lapsed calendar time. Of course, these are related depending on the number of people working on the specific task. In Figure 9-16, only duration is shown; however, the time estimates assume a project team of four people.

When developing a WBS, new analysts frequently ask "How detailed should the individual tasks be?" A few guidelines can help answer that question:

- There should be a way to recognize when the task is complete.
- The definition of the task should be clear enough so one can estimate the amount of effort required.
- As a general rule for software projects, the effort should take one to five working days.

The second step in developing a detailed work schedule for a single iteration is to determine the dependencies between the tasks and the amount of effort required for each. The most common way to relate tasks is to consider the order in which they are completed; that is, as one task finishes, the next one starts. This is called a finish-start relationship. Other ways to relate tasks include start-start relationships, in which tasks start at the same time, and finish-finish relationships, in which tasks must finish at the same time. The effort required should be the actual amount of work required to complete the task. As with the identification of the tasks in the WBS, the dependencies and effort estimates should be done by the developers who are going to actually do the work.

The third step in developing a detailed work schedule is to actually create the iteration schedule. In Figure 1-7 of Chapter 1, we presented a graph of the tasks involved in the first iteration of the Tradeshow system, their sequence, and the estimated calendar time to complete them. The graph was, in actuality, a simplified PERT/CPM chart. We provide more information about PERT charts in Online Chapter C. The other form for presenting a schedule is a bar chart that shows the activities as bars on a horizontal time line; this is called a **Gantt chart**. A widely used tool for building Gantt charts is Microsoft Project. New versions of MS Project are network enabled and provide a powerful tool to not only create schedules but to also distribute schedule information across the organization by using the HTML protocol so it can be viewed in a browser. The benefit of using a tool such as MS Project is that the project manager can update progress easily and make that information widely available.

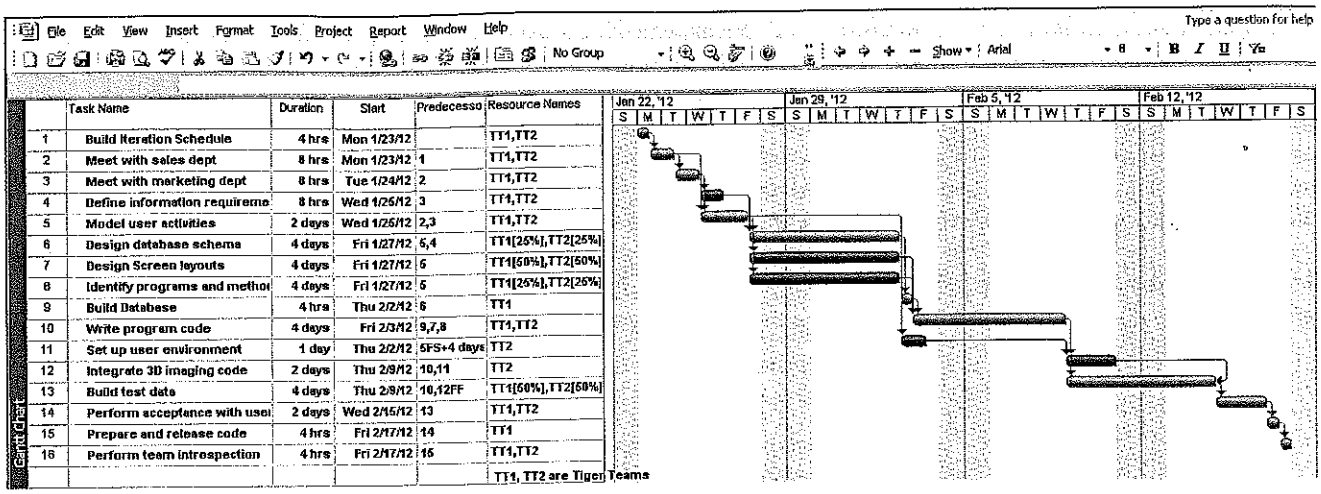
Gantt chart a bar chart that portrays the schedule by the length of horizontal bars superimposed on a calendar

Figure 9-17 shows an iteration schedule from the RMO CSMS project formatted as a bar chart. In the figure, the tasks from the work breakdown structure are listed in the Task Name column and the durations are listed in the Duration column. The Predecessor column identifies dependencies between tasks. As you can see, every task except the first has at least one predecessor task, and every task except the last is a predecessor to one or more other tasks. There are various ways to document dependencies. The most common way is to show the finish of one task occurring before the start of another (FS). Other common ways are finish-finish (FF), where both must finish at the same time, and start-start (SS), where both start at the same time. Any dependency can have a lag time, such as that shown on line 11 of Figure 9-17. The final column documents what resources have been assigned to each task. In this example, the Tiger Team is divided into two subteams of two people each: TT1 and TT2.

The bars in Figure 9-17 illustrate the duration of each task superimposed on a calendar. The red bars indicate a critical path on the schedule. The **critical path** is defined as those tasks that must stay on schedule. If any of the critical path tasks cause a schedule slip, then the entire project is delayed. The blue bars are those tasks that aren't on the critical path. Obviously,

critical path a sequence of tasks that can't be delayed without causing the entire project to be delayed

FIGURE 9-17 *An iteration schedule for the first iteration of the Shopping Cart subsystem*



	Task Name	Duration	Start	Predecesso	Resource Names
1	Build Iteration Schedule	4 hrs	Mon 1/23/12		TT1,TT2
2	Meet with sales dept	8 hrs	Mon 1/23/12	1	TT1,TT2
3	Meet with marketing dept	8 hrs	Tue 1/24/12	2	TT1,TT2
4	Define information requireme	8 hrs	Wed 1/25/12	3	TT1,TT2
5	Model user activities	2 days	Wed 1/25/12	2,3	TT1,TT2
6	Design database schema	4 days	Fri 1/27/12	5,4	TT1[25%],TT2[25%]
7	Design Screen layouts	4 days	Fri 1/27/12	5	TT1[50%],TT2[50%]
8	Identify programs and metho	4 days	Fri 1/27/12	5	TT1[25%],TT2[25%]
9	Build Database	4 hrs	Thu 2/2/12	6	TT1
10	Write program code	4 days	Fri 2/3/12	9,7,8	TT1,TT2
11	Set up user environment	1 day	Thu 2/2/12	5FS+4 days	TT2
12	Integrate 3D imaging code	2 days	Thu 2/9/12	10,11	TT2
13	Build test data	4 days	Thu 2/9/12	10,12FF	TT1[50%],TT2[50%]
14	Perform acceptance with user	2 days	Wed 2/15/12	13	TT1,TT2
15	Prepare and release code	4 hrs	Fri 2/17/12	14	TT1
16	Perform team introspection	4 hrs	Fri 2/17/12	15	TT1,TT2

a project manager will monitor critical path tasks quite closely. Online Chapter C gives more detailed explanations and instructions on how to use MS Project to create Gantt chart schedules.

Staff and Allocating Resources

In an Agile project, the various teams are self-organizing. They decide how they are going to work together and assign the tasks to be done among themselves. However, the job of identifying what expertise is needed for the project and getting those people assigned to the project falls on the shoulders of the project manager. This includes finding the right people with the correct skills and then organizing and managing them throughout the project. The staffing activity consists of five tasks:

- Developing a resource plan for the project
- Identifying and requesting specific technical staff
- Identifying and requesting specific user staff
- Organizing the project team into work groups
- Conducting preliminary training and team-building exercises

Based on the tasks identified in the project schedule, the project manager can develop a detailed resource plan. In fact, the schedule and the resource requirements are usually developed concurrently. In developing the plan, the project manager recognizes that (1) resources usually aren't available as soon as requested and (2) a period of time is needed for a person to become acquainted with the project. After developing the plan, the project manager can then identify specific people and request that they become part of the team.

On small projects, members of the project team may all work together. However, a project team that is larger than four or five members is usually divided into smaller work groups. Each group will have a group leader who coordinates the tasks assigned to the group. The project manager is responsible for dividing the team into groups and assigning group leaders.

Finally, training and team-building exercises are conducted. Training may be done for the project team as a whole when such new technology as a new database or a new programming language is used. In other cases, team

members who are unfamiliar with the tools and techniques being used may require individual training. The team should conduct appropriate training for technical people and users. Team-building exercises are especially important when members haven't worked together before. The integration of users with technical people is an important consideration in developing effective teams and workgroups.

Evaluate Work Processes (How Are We Doing?)

Although evaluating how the project team performed is sometimes done on predictive projects, it isn't a common practice. However, on iterative projects, many companies require an "end of iteration" review of how well the team performed and worked together. One of the advantages of an iterative project is that the same team often stays together for a number of iterations. After each iteration, team members can evaluate how well they worked together and how they can improve their effectiveness and performance as a team. In an Agile project, this is referred to as a **retrospective**. Here are the kinds of questions the team might want to ask:

retrospective a meeting held by the team at the end of an iteration to determine what was successful and what can be improved

- Are our communication procedures adequate? How can they be improved?
- Are our working relationships with the user effective?
- Did we hit our deadlines? Why or why not?
- Did we miss any major issues? How can we avoid this in the future?
- What things went especially well? How can we ensure it continues?
- What were the bottlenecks or problem areas? How can we eliminate them?

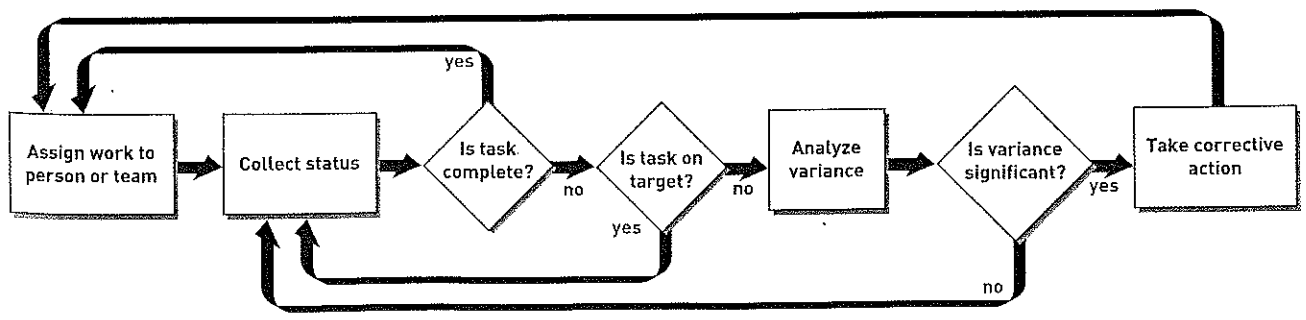
Monitor Project Progress and Make Corrections

In theory, executing and controlling the project plan sounds easy, but in fact, it is quite complicated. To execute any project, you need some type of project plan. How a team builds and executes project plans will vary depending on whether the project structure is based on a predictive approach or an adaptive approach. In the predictive approach, the project plans are quite large and complex. The adaptive approach is less daunting because the detailed project plan is done for each iteration. Because the piece of work is smaller and often better understood, these plans tend to be smaller and less complex.

Figure 9-18 is a high-level process chart that illustrates the basic process for monitoring and controlling the project. The first box—*Assign work to person or team*—refers to a task that is complex all by itself due to the fact that teams are made up of people with varying skill levels and experiences.

The task for the second box—*Collect status*—is less complex. When collecting status information, you should adhere to certain guidelines. First, providing

FIGURE 9-18 *Process to monitor and control project execution*



status information should be a standard process for all team members. Second, status information should be collected and posted electronically for all to see. Status information can be reported at milestones as complete or not complete.

The task for the third box—*Analyze variance*—requires the project manager to try to determine why the task isn't on target and how significant the delay is with regard to the impact on the total project.

The task for the fourth box—*Take corrective action*—can be complex. Experienced project managers have a whole set of tools they can use to try to correct the variance. Sometimes, the correction is as simple as reassigning team members, or maybe it just requires some extra hours of overtime. At other times, tasks may have to be rearranged. In more serious instances, the entire schedule may have to be reworked or more team members may need to be recruited for the team. The objective of corrective action is to get the project back to a known and predictable schedule.

Every development project—whether it follows a predictive or adaptive approach—has lots of questions that need answers and many decisions that need to be made. In many cases, these issues are quickly resolved and the project moves rapidly forward. However, in other instances, the answer to a question or the resolution of an open issue will require additional research. For example, a set of rules for sales commissions might include when and how commissions are calculated, what happens to commissions on returned merchandise, when commissions are paid, how the commission schedule varies to encourage sales of high-margin items and sale items, and so on. If management is still making decisions about these rules, you will need to track these issues until they are resolved.

The monitoring and control of open issues and risks for a project is usually no more complex than building various tracking logs. These logs can be built in a simple spreadsheet and posted on the project's Web site or central repository. It is a good idea to make these logs available to all team members. **Figure 9-19** presents an example of a tracking log. The column headings will vary depending on the type of log you use. The tracking log in Figure 9-19 shows issues that need to be resolved by a certain date and the persons responsible for resolving those issues.

FIGURE **9-19** *Sample issue-tracking log*

	A	B	C	D	E	F	G	H	I
1	Issue log#	Issue date	Issue description	Priority	Issue impact	Person responsible	Target fix date	Resolution description	Actual fix date
2		1/18/2012	Commission structure for sales promotion is undefined	Urgent	Database structure may need to be modified	William Henry	2/1/2012		
3									
4									
5									

Chapter Summary

This chapter focused on the principles and activities related to planning and managing a systems development project. It covered three major themes: (1) the principles of project management, (2) the activities to get a project initiated and approved, and (3) the activities to plan the project and monitor its progress.

Project management is the organizing and directing of other people to achieve a planned result. Historically, software projects haven't had a very good track record. Strong project management is seen as one factor that improves success rates of software development projects. Other factors, such as the adaptive approach to the SDLC, can also contribute to project success.

In this chapter's first section, many important skills, techniques, and concepts that relate to project management were discussed. The project management body of knowledge (PMBOK) provides an extensive conceptual foundation for learning about project management. Agile project management requires the same foundation concepts and skills as the PMBOK, although many of the specific techniques may be different.

This chapter's second major section focused on the specific activities of Core Process 1, the objective of which is to identify the business need and get the project initiated. These activities include:

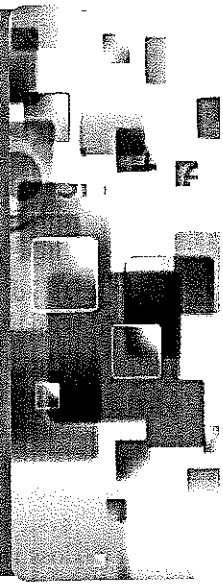
- Identifying the problem
- Extending the project approval factors
- Performing risk and feasibility analysis
- Reviewing with the client and obtaining approval

This chapter's third major section focused on those activities that are necessary to get the project planned, scheduled, and started. These activities include:

- Establishing the project environment
- Scheduling the work
- Staffing and allocating resources
- Evaluating work processes
- Monitoring progress and making corrections

Key Terms

break-even point 248
business benefits 243
ceremony 237
client 237
cost/benefit analysis 247
critical path 259
detailed work schedule 257
Gantt chart 259
intangible benefit 249
net present value (NPV) 247
oversight committee 237
payback period 248
project iteration schedule 257
project management 236
project management body of knowledge (PMBOK) 239
retrospective 261
system capabilities 243
System Vision Document 243
tangible benefit 248
users 237
work breakdown structure (WBS) 258



Review Questions

1. List the six major reasons that projects fail.
2. List six critical factors that contribute to project success.
3. Define project management.
4. List five internal responsibilities of a project manager.
5. What is the difference between the client and the user?
6. What is meant by an organic approach?
7. What is the importance of "ceremony"?
8. List the nine areas of the PMBOK.
9. What is meant by Agile project management?
10. How is scope management accomplished with Agile project management?
11. What are the four activities of Core Process 1?
12. What are three reasons that projects are initiated?
13. What is the difference between system capabilities and business benefits?

14. What factors are usually considered when approving a project?
15. List 10 types of benefits that may be considered when approving a project.
16. Explain how net present value (NPV) is calculated.
17. What is the difference between tangible benefits and intangible benefits?
18. What are some factors to consider when assessing organizational feasibility?
19. What are the five activities of Core Process 2?
20. List seven types of information that should be captured during a project.
21. What is the difference between the project iteration schedule and the detailed work schedule?
22. What is a work breakdown structure used for?
23. What is the benefit of an iteration review and retrospective?

Problems and Exercises

1. Read the following description and then make a list of expected business benefits that the company might derive from a new system:

   Especially for You Jewelers is a small jewelry company in a college town. Over the last couple of years, it has experienced a tremendous increase in its business. However, its financial performance hasn't kept pace with its growth. The current system, which is partly manual and partly automated, doesn't track accounts receivables sufficiently, and the company is finding it difficult to determine why the receivables are so high. It runs frequent specials to attract customers, but it has no idea whether these are profitable or if the benefit—if there is one—comes from associated sales. Especially for You wants to increase repeat sales to its existing customers, thus it needs to develop a customer database. It also wants to install a new direct sales and accounting system to help solve these problems.

2. Read this narrative and then make a list of system capabilities for the company:

   The new direct sales and accounting system for Especially for You Jewelers will be an important element in the growth and success of the jewelry company. The direct sales portion needs to track every sale and be able to link to the inventory system for cost data to provide a daily profit and loss report. The customer database needs to be able to produce purchase histories to assist management in preparing special mailings and special sales to existing customers. Detailed credit balances and Aged accounts for each customer would help solve the problem with the high balance of accounts receivables. Special notice letters and credit history reports would help management reduce accounts receivable.

3. Develop a System Vision Document for Especially for You Jewelers based on the work you did for Problem 1 and Problem 2.

4. Develop a work breakdown structure (WBS) based on the following narrative. It should cover all aspects of the move—from the beginning of the project (now) to the end, when all employees are moved into their new offices. Format your solution in tabular form with the following column headings: Task ID No, Task Description, Estimated Effort, Predecessor Task ID. For your solution, follow these guidelines:

   - Include dependencies.
   - Include effort (work) estimates.
   - Have 30 to 40 detailed tasks.
   - Cover a period of at least two months to a maximum of six months.

   You are an employee of a small company that has outgrown its facility. It is a Web development and hosting company, so you have technical network administrators, developers, and a couple people handling marketing and sales. There are 10 employees.

   The president of your company has purchased a nearby single-story building, and the company is going to move into it. The building will need some internal modifications to make it suitable. The president has asked you to take charge of the move. Your assignment is to (1) get the building ready, (2) arrange for the move, and (3) carry out the move.

   The building is nearly finished, so the job shouldn't be too difficult (no construction is necessary—just some refurbishing). The building has several offices as well as a larger area that needs to be set up with cubicles.

   You and the president are walking through the building, and she tells you what she wants:

   "Let's use the offices as they are," she says. "We will need a reception desk for visiting customers. The office in the back corner should be okay for our computer servers. Let's put the salespeople in these offices along the east wall. We are short a

few offices, so let's put up a few cubicles in the large room for our junior developers.

"Of course, we will need to get everybody connected to our system, and I think Ethernet would be faster than wireless for us. And we all need to have phones.

"Let's plan the move for a long weekend, like a Thursday, Friday, and Saturday. Of course, we need to be careful not to shut down the clients we are already hosting.

"Will you put together a schedule for the move for our employees and set up instructions for all the employees so they know how they are supposed to get ready for the move? Thanks."

5. Enter your WBS from Problem 4 into MS Project. First, enter the tasks, dependencies, and durations. Write a paragraph on your experience using MS Project.
6. Develop a six-year NPV spreadsheet similar to the one shown in Figure 9-10. Use the table of benefits, costs, and discount factors shown in **Figure 9-20**. The development costs for the system were $225,000.
7. Using MS Project, Build a Gantt chart based on the table shown in **Figure 9-21**. Enter the tasks, dependencies, and durations. Print out the PERT chart (Network chart) and the Gantt chart.

Figure 9-21 presents a list of tasks for a student who wants to have an international experience by attending a university abroad. Assume that all predecessor tasks must finish before the succeeding task can begin (the simplest version). Also, insert a few overview tasks, such as Application tasks, Preparation tasks, Travel tasks, and Arrival tasks. Be sure to state your assumption.

FIGURE **9-20**
Benefits, costs, and discount factors for calculating NPV

Year	Annual benefits	Annual operating costs	6% discount factor
1	$55,000	$5,000	0.9524
2	$60,000	$5,000	0.9070
3	$70,000	$5,500	0.8638
4	$75,000	$5,500	0.8227
5	$80,000	$7,000	0.7835
6	$80,000	$8,000	0.7462

FIGURE **9-21** *WBS task list for attending a university abroad*

Task Id	Description	Duration (days)	Predecessor task
1	Obtain forms from the international exchange office.	1	None
2	Fill out and send in the foreign university application.	3	1
3	Receive approval from the foreign university.	21	2
4	Apply for the scholarship.	3	2
5	Revive notice of approval for the scholarship.	30	4
6	Arrange financing.	5	3, 5
7	Arrange for housing in a dormitory.	25	6
8	Obtain a passport and the required visa.	35	6
9	Send preregistration forms to the university.	2	8
10	Make travel arrangements.	1	7, 9
11	Determine clothing requirements and go shopping.	10	10
12	Pack and make final arrangements to leave.	3	11
13	Travel.	1	12
14	Move into the dormitory.	1	13
15	Finalize registration for classes and other university paperwork.	2	14
16	Begin classes.	1	15

8. The state university wants to implement a better system to keep track of all the computer equipment it owns and needs to maintain. The university purchases a tremendous number of computers and software that are distributed throughout the campus and are used by faculty, staff, departments, and colleges. Currently, the university has very sparse records of its equipment and almost no records about maintenance or the software that has been purchased. A list of use cases has been defined; it will serve as the starting point to develop this system.

   Take the following list of use cases to create a project iteration schedule. You should try to arrange the use cases so that similar ones are developed together. Also, the most important use cases should be developed first. State your assumptions, and explain your reasons for your solution.

   Note: For brevity, we use the word *computer* to refer to any type of computing equipment, such as a desktop computer, laptop computer, server computer, printer, monitor, projector, wireless access point, and so forth.

   1. Buy a computer.
   2. Sell a computer.
   3. Put a computer in service.
   4. Take a computer out of service (surplus).
   5. Assign a computer to a person.
   6. Record the location of a computer.
   7. Repair a computer (in house).
   8. Return a computer for repair.
   9. Identify computers ready for replacement.
   10. Search for a specific computer by various options.
   11. Buy a software license.
   12. Renew a software license.
   13. Install software on a computer.
   14. Remove software from a computer.
   15. Record a warranty for a computer.
   16. Purchase a warranty for a computer.
   17. Search for multiple computers by various options.
   18. Search for software on computers by various options.
   19. Assign a computer to a department or college.

Further Resources

Scott W. Ambler, *Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process*. John Wiley and Sons, 2002.

Charles G. Cobb, *Making Sense of Agile Project Management: Balancing Control and Agility*. John Wiley and Sons, 2011.

Jim Highsmith, *Agile Project Management: Creating Innovative Products*. John Wiley and Sons, 2009.

Gopal K. Kapur, *Project Management for Information, Technology, Business, and Certification*. Prentice-Hall, 2005.

Craig Larman and Bas Vodde, *Scaling Lean and Agile Development: Thinking and Organizational Tools for Large-Scale Scrum*. Addison-Wesley, 2009.

Jack R. Meredith and Samuel J. Mantel Jr., *Project Management: A Managerial Approach* (6th ed.). John Wiley and Sons, 2004.

Project Management Institute, *A Guide to the Project Management Body of Knowledge* (4th ed.). Project Management Institute, 2008.

Kathy Schwalbe, *Information Technology Project Management*, (6th ed.). Course Technology, 2009.

Robert K. Wysocki, *Effective Project Management: Traditional, Agile, Extreme*. John Wiley and Sons, 2009.