Chapter 1

Managing Software Projects

Worldwide, some half a million project managers execute about a million software projects each year, producing software worth $600 billion. Many of these projects fail to fulfill customers' quality expectations or fail to deliver the software within budget and on schedule. One analysis suggests that about one-third of projects have cost and schedule overruns of more than 125%.[1]

Why do so many software projects fail? Although there are many reasons, one of the most important is improper management of the project. For example, the major reasons for runaways (projects that are out of control) are unclear objectives, bad planning, new technology, a lack of a project management methodology, and insufficient staff.[2] At least three of these five reasons clearly relate to project management. The other two—insufficient staff and new technology—can be considered as risks whose management is also a part of project management.

Clearly, by using effective project management techniques a project manager can improve the chances of success. But what are these effective techniques?

Let's consider an analogy. Suppose you want to develop a muscular, toned body. To reach your goal, you start looking at exercise routines described in magazines. One article describes how to develop arm strength, giving a set of 10 exercises to be done—not too many by any standard. But then another article, this one on developing thigh strength, also gives 10 exercises, and the evangelist for flat stomachs also feels that doing 10 exercises is not too much. If you want to develop your body overall by following each of these isolated exercise programs, you would find that you have a set of 50 to 100 exercises to do—a clear impossibility for most people, let alone a busy project manager. To achieve your objective, you need a comprehensive training program that is practical and effective.

Similarly, you'll find an abundance of suggestions for performing the various aspects of project management, including effort estimation, risk management, project monitoring, configuration management, and so on. Although each proposed technique solves the problem it is designed to solve, it is not clear how to combine these techniques into a practical and workable process. For effective project management, the need of the hour is a practical, manageable "exercise routine" that will deliver the result. In other words, what is needed is a balanced process that covers the management of the entire project from inception to completion. Unfortunately, there is a paucity of published approaches illustrating how to integrate techniques in this way.

This book fills this gap by describing the set of processes used in a world-class organization to effectively and efficiently manage software projects. The company is Infosys, a software development company that has an enviable track record of project execution; in 2000 alone, Infosys project managers used the processes described here to successfully execute about 500 projects for customers. This book discusses all aspects of Infosys project management—planning, execution, and closure. You'll learn how Infosys project managers estimate, plan for managing risks, collect metrics data, set quality goals, use measurements for monitoring a project, and so on. An interesting aspect of these processes, one that will appeal to busy project managers, is that they are neither complex nor cumbersome, and they use simple metrics.

Infosys has been assessed at level 5 (the highest level) of the Capability Maturity Model (CMM). By extracting project management processes from the set of processes at Infosys, this book also illustrates how projects are managed in a high-maturity organization. Through this illustration, I hope to bring the benefits of the CMM to project managers who have not studied it because of lack of time, because they regard it as being for "process folks" or because they have found it difficult to relate the CMM to project management practices.

This chapter introduces the two topics that form the background for the book: the CMM and Infosys. Because the focus of the book is project management and not the CMM, I restrict the discussion to the project management aspects of the CMM. This chapter also provides an overview of the project management process and the main case study; details of these are discussed in the remainder of the book. First, then, let's briefly discuss the role of processes in project management.

1.1 PROCESSES AND PROJECT MANAGEMENT

A software project has two main activity dimensions: engineering and project management. The engineering dimension deals with building the system and focuses on issues such as how to design, test, code, and so on. The project management dimension deals with properly planning and controlling the engineering activities to meet project goals for cost, schedule, and quality.

If a project is small (say, a team of one or two working for a few weeks), it can be executed somewhat informally. The project plan may be an e-mail specifying the delivery date and perhaps a few intermediate milestones. Requirements might be communicated in a note or even verbally, and intermediate work products, such as design documents, might be scribbles on personal note pads.

These informal techniques, however, do not scale up for larger projects in which many people may work for many months—the situation for most commercial software projects. In such projects, each engineering task must be done carefully by following well-tried methodologies, and the work products must be properly documented so that others can review them. The tasks in the project must be carefully planned and allocated to project personnel and then tracked as the project executes. In other words, to successfully execute larger projects, formality and rigor along these two dimensions must increase.

Formality requires that well-defined processes be used for performing the various tasks so that the outcome becomes more dependent on the capability of the processes. Formality is further enhanced if quantitative approaches are employed in the processes through the use of suitable metrics.

What is a process? Technically, a *process* for a task comprises a sequence of steps that should be followed to execute the task. For an organization, however, the processes it recommends for use by its engineers and project managers are much more than a sequence of steps; they encapsulate what the engineers and project managers have learned about successfully executing projects. Through the processes, the benefits of experience are conferred to everyone, including newcomers in the organization. These processes help managers and engineers emulate past successes and avoid the pitfalls that lead to failures.

For a project, the engineering processes generally specify how to perform engineering activities such as requirement specification, design, testing, and so on. The project management processes, on the other hand, specify how to set milestones, organize personnel, manage risks, monitor progress, and so on. This book focuses on the project management process.

When you consider project management processes, you must ask the question whether project managers will use them. I have often heard process designers complain that project managers don't follow the process and that they resist changes. My experience with project managers at Infosys and other organizations is that they actually want to use processes but only if they're reasonable and will help the project managers execute their projects better. Project managers do, however, resent processes that seem to be unnecessarily bureaucratic and add little value to their work. The trick, then, is to have *lightweight* processes—those that help project managers plan and control their projects better and that give them the flexibility to handle various situations.

In response to the question "Why should project managers follow processes?" S.D. Shibulal—founder, director, and the current head of customer delivery at Infosys—sums it up nicely in a few key points:

- Processes represent collective knowledge. Using them increases your chances of success.
- A process may have some extra steps, but you will not always know beforehand which ones are not needed, and hence you will increase your risks by taking shortcuts.
- Without processes, you cannot predict much about the outcome of your project.
- You and the organization cannot learn effectively without having defined processes. And learning and improvement are imperative in today's knowledge-based world.
- Processes lower your anxiety level. The checklists inevitably cover 80 percent of what needs to be done. Hence, your task reduces to working out the remaining 20 percent.

1.2 PROJECT MANAGEMENT AND THE CMM

Once it is accepted that use of effective processes can help in executing a project successfully, a question immediately arises: What are the desirable characteristics of these processes? The CMM for software is a framework that tries to answer this question.

The CMM for software is a framework that was developed by the Software Engineering Institute (SEI) at Carnegie Mellon University by observing the best practices in software and other organizations. Hence, the CMM reflects the collective process experience and expectations of many companies. It specifies desired characteristics of processes without prescribing specific processes. Thus, different processes can fulfill the requirements of the CMM. It can be used to evaluate the software process of an organization and to identify deficiencies.

The CMM is one of the most popular frameworks for software process improvement (the other commonly used framework is ISO 9001[3,4,5]). The foundations of the CMM were laid down in Watts Humphrey's *Managing the Software Process*,[6] and the framework itself is described completely in the SEI's *The Capability Maturity Model: Guidelines for Improving the Software Process*.[7] A "new edition" of the CMM, called CMM-I, has been released. But because the focus of this book is not on the models and because there is still little experience available with CMM-I, I discuss only the CMM for software and only the project management aspects, even though the CMM also covers organizational and process management issues. I do not discuss the assessment procedure, a brief description of which is given in my book *CMM in Practice*,[8] and a detailed description given in *CMM Based Appraisal for Internal Process Improvement,* by S. Masters.[9]

1.2.1 Overview of the CMM

One objective of the CMM is to distinguish mature processes from immature, or ad hoc, processes. Immature software processes imply that projects are executed without many guidelines, and the outcome of a project depends largely on the capability of the team and the project leader. On the other hand, with mature processes, a project is executed by following defined processes. In this case, the outcome of the project is less dependent on people and more on the processes. It follows, then, that the more mature the processes, the more predictable the results and the more well controlled the projects.

The range of results that can be expected in a project when it is executed using a process is its *process capability.* The actual result achieved in a project executed using the process is its *process performance.* Clearly, the process performance depends on the process capability. To consistently improve process performance on projects, you must enhance the process capability; the process itself must become more mature.

The path to higher maturity includes some well-defined plateaus referred to as *maturity levels* by the CMM. Each maturity level specifies certain characteristics for processes, with higher maturity levels having more advanced characteristics that are found in more mature software processes. Hence, the CMM framework describes the key elements of software processes at different levels of maturity. Consequently, it also specifies the path that a software process follows in moving from immature processes to highly mature processes. This path includes five maturity levels, as shown in Figure 1.1.[7]

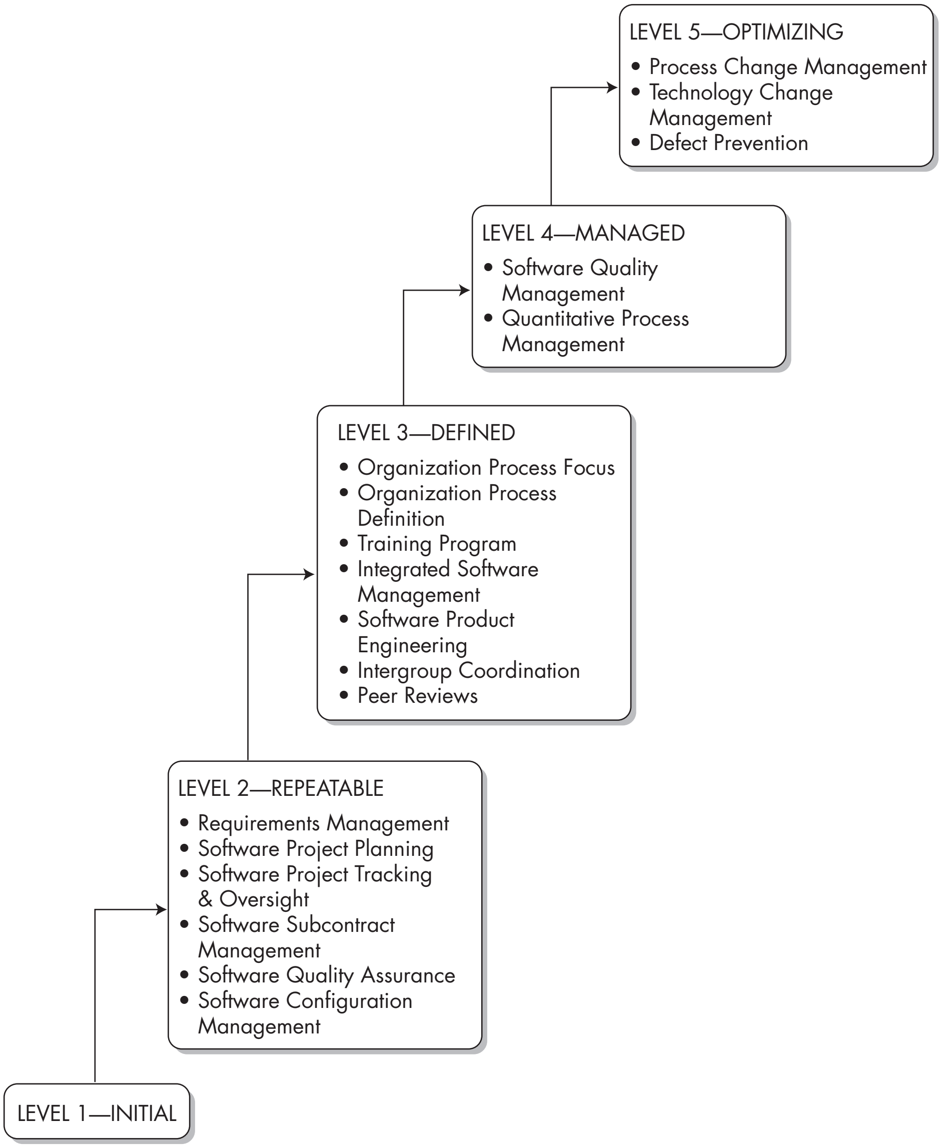
LEVEL 5—OPTIMIZING
• Process Change Management
• Technology Change Management
• Defect Prevention
LEVEL 4—MANAGED
• Software Quality Management
• Quantitative Process Management
LEVEL 3—DEFINED
• Organization Process Focus
• Organization Process Definition
• Training Program
• Integrated Software Management
• Software Product Engineering
• Intergroup Coordination
• Peer Reviews
LEVEL 2—REPEATABLE
• Requirements Management
• Software Project Planning
• Software Project Tracking & Oversight
• Software Subcontract Management
• Software Quality Assurance
• Software Configuration Management
LEVEL 1—INITIAL

Figure 1.1 Maturity levels in the CMM

In level 1, the *initial* level, a project is executed in a manner that the team and project manager see fit. The *repeatable* level (level 2) applies when established project management practices are employed, although organization-wide processes may not exist. At the *defined* level (level 3), organization-wide processes have been defined and are regularly followed. At the *managed* level (level 4), quantitative understanding of the process capability makes it possible to quantitatively predict and control the process performance on a project. At the *optimizing* level (level 5), the process capability is improved in a controlled manner and the improvement is evaluated quantitatively.

Each maturity level (except level 1) is characterized by *key process areas* (KPAs), which specify the areas on which the organization should focus to elevate its processes to that maturity level. Figure 1.1 also shows the KPAs for the different levels. For an organization to achieve a maturity level, it must satisfy all the KPAs at that maturity level as well as the KPAs at all lower maturity levels.

Maintaining processes at higher levels of maturity is a challenging task requiring commitment from the organization and a proper work culture. Of the 900 assessments conducted between 1996 and June 2000 whose assessment results were provided to the SEI, only 3% of the organizations were at level 5, and another 5% were at level 4.[10] The rest were at level 3 or below, with 38% at level 2 and 18% at level 3.

1.2.2 KPAs for Project Management

Each KPA specifies goals that the processes of the organization must meet to satisfy that KPA. In addition, each KPA specifies a group of activities, called *key practices*, that collectively satisfy the goals of that KPA. In many senses, the goals for each KPA capture its essence. They specify the objectives that the CMM has set for the processes relating to the KPA. To illustrate the KPAs associated with project management, we briefly discuss here the goals of these KPAs. These goals are taken from the CMM,[7] with some minor changes in the wording of some goals.

Table 1.1 lists all the goals for KPAs at level 2, showing clearly that the level 2 focus is almost exclusively on project management. Under these goals, you create and document a project plan, evaluate the ongoing project performance against the plan, and take actions when the actual performance significantly deviates from the plan. Requirements are properly documented, and changes to requirements are properly managed. All work products are controlled, and changes to products are properly managed through a planned configuration management plan. Reviews and audits are performed to ensure that planned processes and standards are being followed. If some parts of the project are subcontracted to other vendors, the subcontracted work is also monitored properly.

Table 1.2 details the goals of three of the seven KPAs at level 3. The other KPAs focus on organizational and process management issues. A project in a

Table 1.1 Goals for KPAs at Level 2 (Repeatable)

KPA	Goals
Requirements Management (RM)	• Software requirements are controlled to establish a baseline for software engineering and management activities. • Software plans, products, and activities are kept consistent with requirements.
Software Project Planning (SPP)	• Estimates are documented for use in planning and tracking the project. • Project activities and commitments are planned and documented. • Affected groups and individuals agree to their commitments related to the project.
Software Project Tracking and Oversight (SPTO)	• Actual results and performances are tracked against the software plans. • Corrective actions are taken and managed to closure when actual results and performance deviate significantly from the software plans. • Changes to commitments are agreed to by the affected groups and individuals.
Software Subcontract Management (SSM)	• The prime contractor and the subcontractor agree to their commitments. • The prime contractor tracks the subcontractor's actual results against its commitments. • The prime contractor and the subcontractor maintain ongoing communication. • The prime contractor tracks the subcontractor's actual performance against its commitments.
Software Quality Assurance (SQA)	• Software quality assurance activities are planned. • Adherence of software products and activities to the applicable standards, procedures, and requirements is verified objectively. • Affected groups and individuals are informed of software quality assurance activities and results. • Noncompliance issues that cannot be resolved within the project are addressed by senior management.
Software Configuration Management (SCM)	• Software configuration management activities are planned. • Selected software work products are identified, controlled, and available. • Changes to identified software work products are controlled. • Affected groups and individuals are informed of the status and content of software baselines.

level 3 organization uses a tailored version of the standard process and reuses assets, data, and experience from past projects for planning. The various groups that contribute to the project cooperate smoothly through well-defined interfaces and mechanisms. Reviews are properly carried out to identify defects in work products, and sufficient support for conducting reviews and follow-up activities is provided.

Table 1.2 Goals of Three KPAs at Level 3 (Defined)

KPA	Goals
Integrated Software Management (ISM)	• The project's defined software process is a tailored version of the organization's standard software process. • The project is planned and managed according to the project's defined software process.
Intergroup Coordination (IC)	• All affected groups agree to the customer's requirements. • All groups agree to the commitments between different groups. • The groups identify, track, and resolve intergroup issues.
Peer Reviews (PR)	• Peer review activities are planned. • Defects in the software work products are identified and removed.

Table 1.3 shows the goals for the two KPAs at level 4. At level 4, the capability of the organization's process is understood in quantitative terms. The process capability is used to set quantitative goals for a project. Data on project performance are collected on an ongoing basis and are compared with data on past performance; if significant deviations are observed, proper corrective actions are applied to bring the project back in control. A key aspect of level 4 is the use of statistical process control techniques on an ongoing basis so that each activity can be evaluated and corrective action taken if needed.

The three KPAs at level 5 focus on improving the capability of the process. Of the three KPAs, the Defect Prevention KPA is the one that most directly affects project management. This KPA requires that defects be prevented proactively by systematically analyzing the causes of defects and then eliminating those causes. If

Table 1.3 Goals for KPAs at Level 4 (Managed)

KPA	Goals
Quantitative Process Management (QPM)	• The quantitative process management activities are planned. • The process performance of the project's defined software process is controlled quantitatively. • The process capability of the organization's standard software process is known in quantitative terms.
Software Quality Management (SQM)	• The project's software quality management activities are planned. • Measurable goals for software product quality and their priorities are defined. • Actual progress toward achieving the quality goals for the software products is quantified and managed.

defects can be prevented from entering the software, the effort spent in removing them can be reduced, thereby improving quality and productivity.

1.3 PROJECT MANAGEMENT AT INFOSYS

Infosys executes hundreds of projects each year. Full responsibility for executing a project rests with the project manager, who must make sure that the project team delivers high-quality software to the customer on time and within cost. To help the project manager fulfill this responsibility, support from the organization is necessary. This section provides a brief background on Infosys and its support for managing projects.

1.3.1 Background: Infosys

Infosys is a software house headquartered in Bangalore, India. Its stated mission is "to be a globally respected corporation that provides best-of-breed software solutions delivered by best-in-class people." It employs the global delivery model, in which the customer can be located anywhere in the world and customer fulfillment can be provided from anywhere. In this model, the customer is sought anywhere in the world where it provides the most value to the company. For customer fulfillment, a combination of processes, technology, and management is employed to segregate the work so that value can be added in the most optimum locations and then reaggregated for delivery to the customer.

Infosys currently employs about 10,000 people, with about 15 development centers in four countries and offices in more than a dozen countries. The company was founded in 1981 by seven software professionals with an equity base of only $300. Today, Infosys has a market capitalization of more than $8 billion (based on market rates in June 2001), and its revenue was more than $400 million in 2000 (revenue in 1994 was $9.5 million). Its customers are spread across the globe and include major corporations—more than 60 of them being Fortune 1000 companies—that are engaged in diverse businesses such as banking, retailing, manufacturing, telecommunications, financial services, insurance, and transportation.

Infosys is a highly respected company that has been rated as the best managed and most respected company in India and one of Asia's leading information technology (IT) companies. It has bagged many awards, including the Ramakrishna Bajaj award, which is modeled after the Malcolm Balridge award. It can be safely said that Infosys is one of the best software services corporations in the world.

Infosys provides a top-notch infrastructure so that its project managers can better serve the needs of its worldwide customers. The company has provided audio conferencing facilities to almost every group so that project managers can interact easily with customers and with group members located in different sites. Similarly, a state-of-the-art video conferencing facility is used for interaction among the company's various locations as well as for virtual meetings. Its main campus in Bangalore is now one of the largest software service facilities in the world, with work-related facilities such as a library, extensive computing and networking facilities, training facilities, discussion rooms, projection facilities, and so on, as well as recreational facilities such as an art gallery, a health club, and facilities for tennis, basketball, and so on.

Process orientation and improvement are a part of the Infosys work culture, and processes are defined for most tasks that are performed regularly. For process definition and improvement, Infosys first adopted the ISO 9000 framework and got its ISO certification in 1993. To further improve the software process, Infosys then adopted the CMM framework. It was first assessed at level 4 in December 1997, and then at level 5 in December 1999. In its pursuit of continuous improvement, Infosys now employs the Malcolm Balridge framework for all-around improvement and building leadership excellence in all areas of operation.

1.3.2 SEPG Support to Projects

The quality department at Infosys contains the software engineering process group (SEPG). The SEPG is responsible for coordinating all the process activities, including process definition, process improvement, and process deployment. It also manages all information and data related to the use of processes (such as the process database and the process capability baseline, which are discussed further in Chapter 2).

Although the responsibility for all aspects of delivery, including quality, belongs to the project team, the SEPG facilitates the project team in following the right processes. The SEPG also forms an independent channel for monitoring and reporting to senior management on process and quality issues. Because "processes won't stick by themselves,"[6] the SEPG helps to ensure that the defined processes are implemented and become standard practice.

To this end, in addition to offering training on processes, the SEPG provides a member who is associated with a project as a *software quality adviser*. The quality adviser assists in defining and following processes, ensures that the processes are followed, aids in analyzing the data, and provides any needed process training. Because the adviser is well versed on processes, guidelines, and so on, the adviser's main help comes during project planning. The adviser also reviews the project plan to ensure that it contains all the key elements.

In addition to providing consulting and help with processes and metrics, the Infosys SEPG schedules and manages regular independent audits (see Chapter 11) to ensure that the defined processes and standards are being followed.

1.3.3 Senior Management Involvement in Projects

Infosys prides itself in providing value to its customers through delivery excellence. Everything at Infosys, including its organizational structure, is driven by the aim of serving customers efficiently and effectively and quickly tapping new business opportunities.

For delivery of customer services, Infosys has many *business units*. Within a business unit, a *team,* headed by a *project manager*, executes a project. The project manager is responsible for all aspects of project execution, from determining the requirements to final installation of the software. The project manager reports to a *business manager*, who in turn generally reports to the *business unit head*.

To handle situations that cannot be resolved by the project manager, senior management involvement in projects is essential. At Infosys, the business manager regularly interacts with the project manager and monitors the project through status reports and milestone reports (discussed in Chapter 11). In addition to regular monitoring, the business manager also helps to resolve issues and problems that cannot be handled by the project team and are *escalated* to his level (escalation is discussed in Chapter 8). The business manager also interacts with customers to ensure that they are satisfied and that any issues are promptly raised and addressed.

In addition, other senior people also review projects periodically by regularly taking part in internal audits (discussed in Chapter 11). Through two systems—called PRISM (project review by senior management) and IPM (integrated project management)—milestone reports and project plans are available for senior management to review. All senior managers are expected to review some projects periodically through this system and to give feedback to the project leaders.

Overall, senior management maintains involvement in the project primarily by monitoring to ensure that the project objectives are met and that the customer is fully satisfied.

1.3.4 Training for Project Managers

Because project managers have the main responsibility for satisfying the customer, they need to master not only executing the technical aspects of a project but also interacting with customers, eliciting requirements, managing the team, and so on. Clearly no one is likely to possess all the skills needed, so it's crucial to train people to develop the necessary skills. Infosys has implemented a variety of programs to help people transition from being engineers to being project leaders.

All fresh entrants undergo a three- to four-month *induction training program*. In addition to training in engineering and technology, this program contains one- or two-day programs in business etiquette, written communication, public speaking, body language, and so on.

Later, when engineers are ready to become *module leaders* (those who manage the development of a system module, especially in larger projects) or project managers, they attend a series of technical and soft-skills training programs. Included in the former is a five-day project management course that focuses on all aspects of project management: planning, monitoring, controlling, and so on. A two-week course on requirements specification and management teaches how to elicit requirements, how to document them, how to verify them, and so on. The five-day residential soft-skills training program includes modules on appraisals and team management, customer focus and customer management, leadership, social and business etiquette for different countries, and so on.

Other regularly offered programs focus on various aspects of management; project leaders take these courses when their schedules permit. Also, team-building workshops are conducted by professionals.

1.3.5 The Project Management Process

For a project team to successfully execute a project, it must perform hundreds of tasks, many of them interdependent. Effectively managing this process is extremely important for success. At Infosys, the set of activities executed by a project manager is specified in the *project management* process. It is fairly standard, having three main stages:

- Project planning
- Project execution
- Project closure

In the *project planning* stage, the project manager reviews contractual commitments and creates a plan to meet them. Creating a project plan involves defining a life-cycle process to be followed, estimating the effort and schedule, preparing a detailed schedule of tasks, and so on. It also includes planning for quality and configuration management as well as risk management. In this phase, the major activities of the project manager are as follows:

- Perform startup and administrative tasks.
- Create a project plan and schedule.
  - Define the project objectives.
  - Identify a suitable standard process for project execution.
  - Tailor the standard process to meet project requirements.
  - Define a process for managing changes in requirements.
  - Estimate the effort.
  - Plan for human resources and team organization.
  - Define the project milestones and create a schedule.
  - Define the quality objectives and a quality plan to achieve them.
  - Make a defect prevention plan.
  - Identify risks and make plans to mitigate them.
  - Define a measurement plan for the project.
  - Define a training plan for the project.
  - Define project-tracking procedures.
- Perform a review of the project plan and schedule.
- Obtain authorization from senior management.
- Define and review the configuration management plan.
- Orient the project team to the project management plan.

In addition to the project manager, this phase involves the customer, an SEPG representative, and the business manager for the project. The entry criterion is that the contract or project authorization is available. The exit criterion is that the project plan has been documented and group reviewed (see Chapter 10).

The second phase, *project execution,* involves executing the project plan, tracking the status of the project, and making corrections whenever project performance strays from the path laid down in the project plan. In other words, it involves tracking and controlling the implementation of the project process. This phase is the longest in the project management process, incorporating periodic tasks such as monitoring project status and quality and taking any needed corrective steps. In this phase, the project manager performs these main activities:

- Execute the project as per the project plan.
- Track the project status.
- Review the project status with senior management.
- Monitor compliance with the defined project process.
- Analyze defects and perform defect prevention activities.
- Monitor performance at the program level.
- Conduct milestone reviews and replan if necessary.

Other members of the team also participate in this stage. The entry criterion is that the project plan is complete and approved, and the exit criterion is that all work products delivered are accepted by the customer.

The last stage of the project management process, *project closure,* involves a systematic wind-up of the project after customer acceptance. The main goal here is to learn from the experience so that the process can be improved. Post-project data analysis constitutes the main activity; metrics are analyzed, process assets (materials, such as templates and guidelines, used to aid in managing the process itself) are collected for future use, and lessons are recorded. Because learning from the project is the main goal, this is a group activity that involves the project manager, the SEPG, and other members of the team. The entry criterion is that the customer has accepted the work products. The exit criterion is that a postproject meeting has been conducted. The main outputs of this phase are the project closure report and the collected process assets.

The remainder of this book discusses the various elements of this management process. Part I includes separate chapters devoted to key planning activities, such as process definition and tailoring, risk management, effort and schedule estimation, quality planning, and configuration management planning. The other tasks in the planning phase (such as human resource planning, project organization, tools to be used, project tracking procedures, and so on) are discussed briefly in Chapters 7 and 8. Part II includes chapters on project monitoring and controlling and on project closure.

1.4 OVERVIEW OF THE ACIC CASE STUDY

ACIC Corporation (name changed to protect confidentiality) is a multibillion-dollar financial institution. To keep up with the times, several years ago it started slowly

Web-enabling its applications, and it wanted to start an on-line service for opening and tracking accounts. Because Infosys had successfully built some e-services for ACIC earlier in a project called Synergy (name changed), ACIC employed Infosys to analyze the problem. This work was executed in *time and material* (T&M) mode—that is, the customer paid for the effort spent by Infosys in doing the analysis. Based on the analysis output, Infosys made a successful bid for the Web project, giving rise to the ACIC case study that runs throughout this book. The project successfully released the new service in time, and the software has been in operation without any problem. (This case study is different from the WAR project case study discussed in my earlier book.[6])

The ACIC project illustrates the various project planning and monitoring tasks undertaken in executing a project at Infosys. Many of the outputs related to management of the ACIC project are given in the relevant chapters. These include the following:

- The data from the Synergy project, which was used by the ACIC project manager during planning (Chapter 2)
- The project's process plan (Chapter 3)
- An analysis of the impact of a requirement change request (Chapter 3)
- Effort estimates and the high-level schedule, along with a description of how they were obtained (Chapter 4)
- The quality plan containing quality goals and plans for achieving them, including plans for defect prevention and reviews (Chapter 5)
- The risk management plan describing the major risks, their risk exposure and impact, their prioritization, and the risk mitigation plans for the high-priority risks (Chapter 6)
- The measurement and tracking plan (Chapter 7)
- The complete project management plan, including the team management plan and the customer communication and escalation plan (Chapter 8)
- The complete configuration management plan (Chapter 9)
- Project tracking documents, including the defect log, the issues log, the status report, and the milestone report (Chapter 11)
- Details of defect prevention, including defect analysis results and the impact on the project of the defect prevention plan (Chapter 11)
- The complete closure report, which includes the metrics data on quality, productivity, cost of quality, defect removal efficiency, and so on (Chapter 12)

1.5 SUMMARY

Software project management is perhaps the most important factor in the outcome of a project. Without proper project management, a project will almost certainly fail. Many organizations have evolved effective project management processes. This book describes these processes for one such organization, Infosys, which has been assessed at level 5 of the CMM and whose project managers have successfully executed hundreds of projects.

Here are the key takeaways from this chapter:

- Processes for the various aspects of project management should not be looked at in isolation. In a balanced process, the practices integrate smoothly.
- Processes of an organization should encapsulate its best practices so as to help new projects replicate past successes and avoid failures.
- At the top level, the project management process consists of three phases: planning, execution, and closure.
- For effective execution of projects, project managers should be supported through the help of an SEPG in executing processes; senior management monitoring and issue resolution; and good training.
- Many key process areas at all maturity levels of the CMM for software focus directly on project management.

1.6 REFERENCES

1. L.H. Putnam and W. Myers. *Industrial Strength Software: Effective Management Using Measurement.* IEEE Computer Society Press, 1997.
2. R.L. Glass. *Software Runaways: Lessons Learned from Massive Software Project Failures.* Prentice Hall PTR, 1998.
3. International Standards Organization. *ISO 9001, Quality Systems—Model for Quality Assurance in Design/Development, Production, Installation, and Services.* 1987.
4. International Standards Organization. *ISO 9000-3, Guidelines for the Application of ISO9001 to the Development, Supply and Maintenance of Software.* 1991.
5. U.K. Dept. of Trade and Industry and British Computer Society. *TickIT: A Guide to Software Quality Management System Construction and Certification Using EN29001.* 1992.

6. W. Humphrey. *Managing the Software Process.* Addison-Wesley, 1989.

7. Carnegie Mellon University/Software Engineering Institute. *The Capability Maturity Model: Guidelines for Improving the Software Process.* Addison-Wesley, 1995.

8. P. Jalote. *CMM in Practice: Processes for Executing Software Projects at Infosys*. Addison-Wesley, 2000.

9. S. Masters. *CMM Based Appraisal for Internal Process Improvement (CBA-IPI): Method Description. Technical Report, Software Engineering Institute, CMU/SEI-96-TR-007*, 1996.

10. Software Engineering Institute. Maturity Profile Report, http://www.sei.cmu.edu/activities/sema/profile.html.