4

PLANS AND PLANNING

A plan in the mind of a man is not a plan.
—Richard H. Thayer

4.1 INTRODUCTION TO THE PLANNING PROCESS

By definition, every project of every kind is an endeavor of limited duration that uses resources to achieve stated objectives. A project plan specifies, among other things, the duration of the project, the resources needed, and how the resources will be applied to achieve the stated objectives. Software requirements (discussed in Chapter 3) provide the objectives for the product to be developed or modified. The planning process is concerned with developing the various elements of a project plan and documenting the plan in a specified format.

Your software project management plan must be a written document; otherwise, various stakeholders in the project will have differing interpretations of how the project will be conducted, and there will be no documentation of plans for effort, cost, schedule, resources, and supporting activities. The project plan also provides a vehicle for trade studies and for negotiating trade-offs among cost, schedule, and requirements, both initially and as changes occur. Baseline control of the written project plan supports systematic updating of the plan and communication of changes.

In the best case, your planning process will begin with tailoring of your organization's standard processes to fit the management, software development, and supporting processes of your project. In that case the information in this chapter can be used as a checklist against which you can compare your organization's planning processes and document templates.

Managing and Leading Software Projects, by Richard E. Fairley

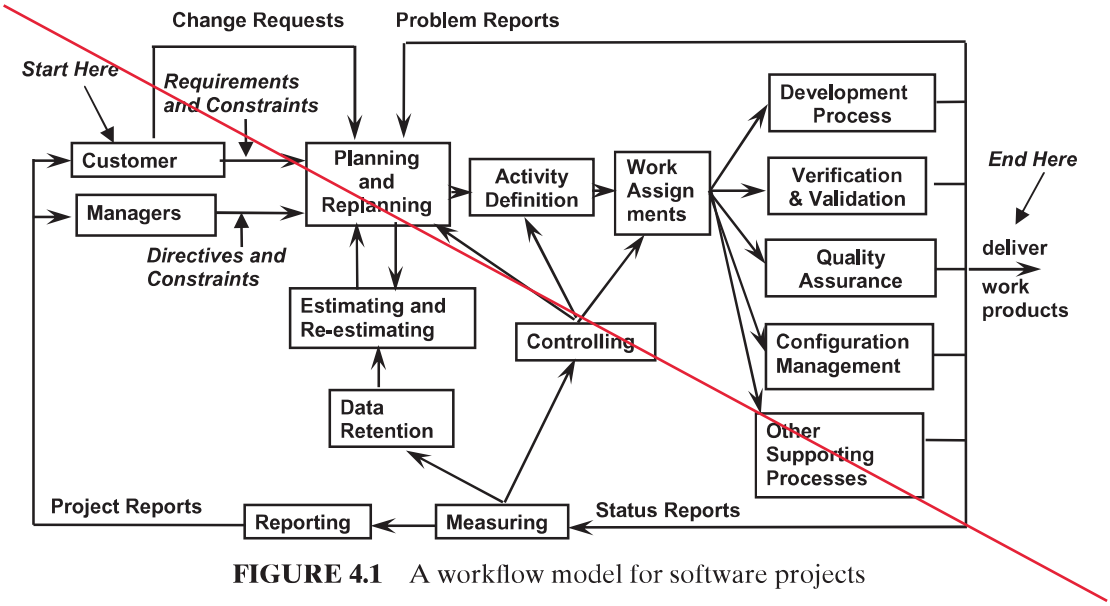

FIGURE 4.1 A workflow model for software projects

In the worst case, you will have to develop your project plan without any organizational structures or guidelines. In the absence of organizational structures and guidelines, the workflow model for managing software projects presented in Figure 1.1 (repeated here as Figure 4.1), the system engineering model presented in Figure 2.1*b*, and the development models and supporting processes described in Chapter 2, plus the information in this chapter can provide a tailorable framework for planning and executing software projects.

Project planning, like all elements of software development, is best accomplished in an iterative manner; details are added as understanding grows.

4.2 OBJECTIVES OF THIS CHAPTER

After reading this chapter and completing the exercises, you should understand:

- the planning process for software projects
- the project planning process area of CMMI-DEV-v1.2
- an approach to planning Agile projects
- a template for software project management plans (SPMPs)
- tailoring the SPMP template
- techniques for preparing a SPMP

The planning process presented in this chapter is informed by the Project Planning process area of the CMMI-DEV-v1.2 process framework, the planning elements of ISO and IEEE Standards 12207, IEEE Standard 1058, and the PMI Body of Knowledge. These elements are described in Appendix 4A to this chapter. An annotated version of IEEE Standard 1058 is presented in Appendix 4B.

An electronic copy of the annotated version of IEEE Standard 1058, presentation slides for this chapter, and other supporting material are available at the URL listed

in the Preface to this text. Terms used in this chapter and throughout this text are defined in the Glossary at the end of the text. Mechanisms and techniques of project planning and estimation are presented in Chapters 5 and 6 of this text.

4.3 THE PLANNING PROCESS

As depicted in the workflow model of Figure 1.1 in Chapter 1 of this text, repeated here as Figure 4.1, inputs to planning include the customer's requirements and constraints as well management directives and constraints. The system requirements, system design, and software requirements may also be available or they are developed during project initiation. As discussed in Chapter 3, customer's requirements include operational features, quality attributes, and design constraints for the envisioned product. Constraints imposed by the customer may include both product and process constraints.

A product constraint might require that the system be developed using a specified version of an operating system or that the new or modified system provide an SQL interface to an existing database. A process constraint might require that the system be delivered in a staged sequence of increasing capabilities or that the source code for the deliverable software plus the requirements and design documentation be delivered to an independent agent for final verification and validation.

Management directives may include a policy statement that all projects must produce design documentation and verify it for completeness, correctness, and consistency using peer reviews. A management constraint might limit your project resources to a staffing level of 10 software developers.

Some of your first tasks as project manager are to establish a pattern of ongoing communication with the designated customer representative (your primary point of contact), and to clarify with him/her/them the operational requirements, development constraints, and success criteria for the project.

As stated in Chapter 3, each operational requirement must be prioritized as Essential, Desirable, or Optional to facilitate achievement of a balance among requirements, schedule, and budget. Sufficient time and resources must be provided to implement all of the Essential requirements and as many of the Desirable requirements as desired by the customer.

Depending on the nature and scope of your project, clarifying the operational requirements and developing the system requirements, system architecture, and software requirements may be your task. Alternatively, you may delegate it to one or more members of your planning team, or they may be provided as the starting point for your planning process.

Your understanding of the operational requirements and development constraints will influence your choice of the development model to be used and the procedures to be followed. Considerations include:

- Development of a user-intensive system may require prototyping to clarify the operational requirements and to provide information for design of the user interface.
- Development of the software for an embedded system may require the participation of you and your technical leader on the system engineering team.

- Development of staged delivery of system capabilities based on stable requirements and a stable architecture may indicate that an incremental build strategy is appropriate.
- Development of a first-of-its-kind system may require an evolutionary development strategy.
- An Agile process may be appropriate for development and ongoing enhancement of a Web-based application or in cases where the requirements are evolving or changing rapidly.

Development models are presented in Chapter 2.

An external customer (an acquirer) will typically specify the amount of money and the time available for the project, which may have to be negotiated to achieve a balance with the requirements. An internal customer may or may not provide money and/or resources for the project but will undoubtedly specify a schedule constraint for completion of the project, which may have to be negotiated. In any case, a contractual agreement in the form of a Statement of Work or a Memo of Understanding that contains items such as those listed in Section 3.3.2 of this text should be negotiated and accepted by you and your customer. Other planning activities include establishing an initial baseline of requirements, preparing estimates, and negotiating constraints to obtain a balance among requirements, cost, and schedule.

According to IEEE Standard 12207.1, every kind of plan, whether it is a project plan, a configuration management plan, a quality assurance plan, a training plan, or other kind of plan should contain the following information [12207]:

- needs to be satisfied
- success criteria
- work activities to be accomplished
- schedule, budget, and resources
- quality control measures
- change procedures and tracking of project history
- interfaces to relevant stakeholders
- roles to be played
- responsibilities and authorities
- resource acquisition plan
- skills acquisition plan, as needed

In addition every kind of plan must undergo a formal review and be accepted by the appropriate stakeholders, including the initial version of and subsequent changes to the plan.

In conjunction with preparing the generic information listed above, planning the specifics of a software project should include the activities contained in Tables 4.1 and 4.2.

Although the activities in Tables 4.1 and 4.2 are in sequential order, it should be understood that they, like most activities in software engineering, are best accomplished in an iterative manner. It should also be understood that the planning

TABLE 4.1 Preplanning activities for software projects

Preplanning activities
• Establish a working relationship with your customer/acquirer and other project stakeholders
• Develop and/or clarify operational requirements and development constraints
• Prioritize operational requirements
• Establish the initial baseline of operational requirements
• Develop system requirements and system architecture, as appropriate
• Develop technical specifications for the software requirements
• Establish traceability among operational requirements, system requirements, and software requirements
• Obtain commitment to an initial version of the requirements by customer/acquirer and other appropriate stakeholders
• Establish an initial baseline of operational requirements and technical specifications
• Identify the resources needed and a schedule for developing the initial version of the project plan

TABLE 4.2A Comprehensive scope of planning activities for software projects (part 1)

Planning activities
• Plan for ongoing interactions with the customer in reviews, demonstrations, approvals, and acceptance of the delivered product
• Plan for ongoing interactions with the user community in requirements elicitation, prototype demonstrations, and operational evaluations
• Prepare a preliminary estimate of effort, cost, and schedule to determine feasibility of the project within the constraints on those factors
• Refine the technical specifications for the system or product
• Specify a development process and supporting processes
• Develop an architecture decomposition view (ADV) of the product architecture and allocate requirements to the elements of the ADV
• Specify the interfaces among modules in the ADV and the interfaces between modules and the external environment
• Develop a work breakdown structure that includes work elements for the ADV modules
• Develop work packages for the tasks in the work breakdown structure (WBS)
• Define a schedule of objectively measurable milestones
• Prepare a schedule network and identify the critical path(s)
• Prepare a PERT (Program Evaluation and Review Technique) estimate of project duration
• Identify numbers and kinds of resources needed, when they will be needed, and for how long
• Prepare an estimate of optimal effort, cost, schedule, and resources
• Negotiate with the customer to obtain a balance among requirements, cost, resources, and project duration that satisfies the project constraints
• Finalize a contractual agreement with the customer that provides a balance among requirements, schedule, resources, and cost

TABLE 4.2B Comprehensive scope of planning activities for software projects (part 2)

Planning activities

- Define the organizational structure of the project team and specify roles, responsibilities, and authorities
- Establish the engineering environment to include standards, procedures, and tools for software development, verification, and validation
- Specify a version control process and a version control tool
- Establish a change control board for the project
- Identify work products to be placed under version control
- Establish a change control process that includes an impact analysis process
- Specify objective acceptance criteria for placing new and modified work products under version control
- Plan for verification and validation of work products
- Develop a measurement plan to measure and report quantity and quality of work products, effort, cost, progress, defects and other quality measures
- Develop a risk management plan to identify and confront risk factors on an on-going basis
- Develop plans, as appropriate, for the following kinds of activities:
  - management of subcontractors and vendors
  - coordination with associated projects and programs
  - coordination with the Independent (big "I") verification and validation organization
  - information security, including security clearances and access to information within various organizational entities
  - approvals as required by regulations, licensing agreements, and rights-in-data
  - installation, user training, and transition
  - ongoing maintenance activities
  - management of computing resources, facilities management, physical security
  - backup protection of product and process data
- Prepare a plan for updating the project plan on a periodic basis and as events dictate
- Document the project plan using the organization's standard format, a tailored format based on IEEE Standard 1058 [IEEE1058], or the format in Table 4.4 of this text
- Review the project plan with the customer, higher level managers, and other appropriate stakeholders; revise as needed
- Obtain commitment to the plan by the appropriate stakeholders
- Place the plan under version control, thus establishing the initial baseline of the plan

activities in Tables 4.1 and 4.2 may be more comprehensive than necessary for your project; they should be tailored to fit the needs of the project.

The items tagged "as appropriate" in Table 4.2 may not apply to your project; all other items in the list should be addressed at a level of detail appropriate for the nature and scope of your project and the criticality of the system or product to be developed. Items in Table 4.2 that are not included in your planning activities should be noted in the project plan, and brief justifications should be provided for not including them.

If you are fortunate to work in a well-managed organization most of the activities Tables 4.1 and 4.2 will have standard processes, procedures, and tools that will require little, if any, tailoring for your project. For example, the configuration management and independent testing processes may be standardized; there may be a set of development process models to choose from; there may be organizational

units that have trained personnel, procedures, and tools for most or all of the supporting processes; and there may be internal consultants to assist in tailoring a template and preparing the project plan.

If you are not so fortunate, the planning process can require a large amount of effort. In this case it is tempting to circumvent most of the planning activities listed above. The risk to project success must be assessed for the items that are not planned; for example:

- What risks are incurred if you don't have a process for managing changes to requirements?
- What risks are incurred if you don't have a process for assessing the impact of changes to requirements, cost, schedule, or technology?
- What risks are incurred if you don't have a schedule with objective milestones?
- What risks are incurred if you don't have a process for measuring effort and defects?
- What risks are incurred if you don't practice risk management?

These issues and other aspects of project risk are presented in Chapter 9.

Of course, the level of detail in your plan should be appropriate to the scope and criticality of your project. The plan may consist of a few pages for a small project or many pages for a large project. An additional consideration is that the planning activities listed above may not be accomplished in the order listed. For example, some elements of the contractual agreement may be specified in a legally binding contract before you, the project manager, become involved in the project.

Planning activities should occur in a manner that fits the needs of the situation; for example, some planning activities may occur in an evolutionary manner, as in situations where requirements evolve and schedule milestones indicate that a working version of the system is to be demonstrated to the customer on a weekly basis. In these situations planning what to do next will evolve as the situation evolves.

4.4 THE CMMI-DEV-V1.2 PROCESS AREA FOR PROJECT PLANNING

According to CMMI-DEV-v1.2, the purpose of the project planning process is to establish and maintain plans that define project activities. Other CMMI-DEV-v1.2 process areas related to project planning include:

- requirements development,
- requirements management,
- risk management, and
- the technical solution process areas.

Specific goals of project planning in CMMI-DEV-v1.2 include establishing estimates, developing a project plan, and obtaining commitment to the plan. Specific practices related to these specific goals are listed in Table 4.3. The nature of those

TABLE 4.3 CMMI-DEV-v1.2 specific goals and practices of project planning

	Specific Practices	Chapters of This Text
	SG 1 establish estimates	
SP 1.1-1	Estimate the scope of the project	Chapter 3
SP 1.2-1	Establish estimates of work product and task attributes	Chapter 5
SP 1.3-1	Define project life cycle	Chapter 2
SP 1.4-1	Determine estimates of effort and cost	Chapter 6
	SG 2 develop a project plan	
SP 2.1-1	Establish the budget and schedule	Chapter 6
SP 2.2-1	Identify project risks	Chapter 9
SP 2.3-1	Plan for data management	Chapters 7 & 8
SP 2.4-1	Plan for project resources	Chapter 5
SP 2.5-1	Plan for needed knowledge and skills	Chapter 5
SP 2.6-1	Plan stakeholder involvement	Chapter 2
SP 2.7-1	Establish the project plan	Chapter 4
	SG 3 obtain commitment to the plan	
SP 3.1-1	Review plans that affect the project	Chapter 4
SP 3.2-1	Reconcile work and resource levels	Chapter 6
SP 3.3-1	Obtain plan commitment	Chapter 4

specific practices is discussed here. Techniques for accomplishing these practices are presented in subsequent chapters, as indicated in the third column of Table 4.3.

An estimate of effort, schedule, and resources based on requirements and constraints is an essential element of a project plan; stated differently, it is not possible to develop a plan without estimates, and it is not possible to develop estimates without requirements. The CMMI-DEV-v1.2 process areas of requirements development and requirements management (presented in Chapter 3) are thus closely related to the project planning process area.

Estimating the scope of a project is concerned with identifying all of the work activities to be accomplished. A work breakdown structure is typically used to document the scope of a project; work breakdown structures are presented in Chapter 5. Product size and complexity are the primary factors typically used to determine the amount of effort that will be required to develop a software product. Other factors include required performance, reliability, safety, and security. Therefore establishing estimates of product attributes such as size and complexity is a specific practice of specific goal SG 1 (SP 1.2-1). Other factors that will impact effort and schedule should also be considered.

An appropriate development model for a software project depends on the scope of work to be accomplished, the attributes of the product, and the phases of development to be included. SP 1.3-1 is concerned with defining a software development model that includes a set of development phases appropriate for the project scope and product attributes. Based on the outcomes of specific practices 1.1-1, 1.2-1, and

1.3-1, an estimate of effort and cost is developed using historical data, expert judgment, and other techniques presented in Chapter 6 of this text.

Achieving specific goal SG 1 in Table 4.3 provides the foundation for achieving SG 2, Develop a Project Plan. A project plan that satisfies SG 2 will contain a schedule and a budget that satisfy the effort and cost estimates developed in satisfying SG 1; said another way, effort, schedule, and cost estimates provide constraints that cannot be exceeded in the project plan.

Because effort is the product of people and time, a schedule can be derived from an effort estimate; for example, 54 person-months of effort might be scheduled as 6 persons for 9 months. The schedule specifies the predecessor tasks that must be completed, and the work products produced by those tasks that must be available, before subsequent tasks can begin; the schedule also specifies the successor tasks that can be performed after each task is completed. Sequencing constraints among work activities are thereby specified, and opportunities for concurrent work activities are identified. The budget is then allocated to each of the tasks to be accomplished.

A risk is a potential problem that (should it become a problem) will adversely impact a successful outcome of delivering an acceptable product on time and within budget. Risk factors should be identified in the project plan and appropriate mitigation actions planned. Risk management for software projects is presented in Chapter 9.

SP 2.3-1 in Table 4.3 involves developing a plan for management of project data, which includes all of the data in all areas of the project (project management, development processes, and supporting processes). The plan should specify the project data to be collected, a schedule for collecting and validating the data, report formats and distribution lists, and any requirements for privacy and security of project data.

SP 2.4-1 and SP 2.5-1 are concerned with identifying and planning for the resources necessary to perform a project. Resources include quantities of people and required skill levels, software tools, computing hardware, facilities, travel budget, and all other resources needed to conduct a project (SP 2.4-1). Because people are typically the most important resource for a software project, it is important to identify the knowledge and skill levels needed to perform a project (SP 2.5-1).

Stakeholders are people whose involvement in a project is necessary or desirable to ensure a successful outcome. Different kinds of people may have different kinds and amounts of involvement during different phases of a project. For example, involvement of user representatives is more important during requirements definition and product acceptance than during detailed design and coding. SP 2.6-1 is concerned with planning for involvement of identified stakeholders.

Achievement of SG 2 in Table 3.1 culminates in SP 2.7-1 (Establish the Project Plan). Format and contents of a software project management plan are presented and discussed in the following section of this chapter.

Obtain Commitment to the Plan (SG 3) includes three specific practices. SP 3.1-1 involves reviewing all plans that affect the project to understand project commitments. For example, documentation of requirements, plans for some or all of the supporting processes, and plans for activities such as installing the delivered system and training of users are typically developed and documented separately, and referenced in the project plan.

SP 3.2-1 (Reconcile Work and Resource Levels) is concerned with reconciling differences between estimates and available resources. An estimate might, for example, indicate that 10 people having specified skills will be needed to complete a project in the required 12 months. Perhaps the budget will support only 7 people at the requisite skill level.

Acceptable options for reconciling differences between the work to be done, resources, and available time include:

- reducing the requirements (de-scoping),
- increasing the quantity of resources (and the corresponding budget),
- using more productive resources, and
- extending the schedule.

Unacceptable options for achieving a balance among the work to be done, resources, and available time (i.e., requirements, effort, and schedule) in the project plan include descoping plans for measurement and control, peer reviews, and verification and validation; and planning for overtime.

The final step in the CMMI® Project Planning process area is obtaining commitment to the plan by the stakeholders who are responsible for performing and supporting execution of the plan. Commitments to each work activity identified in SP 1.1-1 should be obtained from the relevant stakeholders internal to your project as well as any external stakeholders such as senior management, external customer, and associated projects. Organizational interfaces and technical interface specifications should be specified and commitments must be obtained from the appropriate stakeholders to participate in maintaining the interfaces. At minimum, stakeholders will include you, your software architect, the quality assurance group, your manager, and a customer representative (e.g., the marketing department, or an external customer).

The approach to project planning indicated in Tables 4.1 and 4.2 and by the Project Planning process area of CMMI-DEV-v1.2 is the basis of the so-called plan-driven approach to managing software projects. It must be emphasized that the comprehensive set of tasks in Tables 4.1 and 4.2 and the specific goals and practices of CMMI-DEV-v1.2 are sufficient for the largest and most complex software projects. They must be tailored and adapted to fit the needs of each project. Unfortunately, some people misinterpret the plan-driven approach and reject it as being too cumbersome and bureaucratic without understanding that a plan must be tailored and adapted to the needs of each situation. The plan-driven approach to project planning is appropriate in two situations:

1. when there is a formal contractual agreement between an acquirer and a supplier, and/or
2. for large, complex projects internal to an organization.

4.4.1 Planning Agile Projects

An Agile approach may be appropriate for small projects (e.g., 10 or fewer software developers) when formal contractual conditions do not apply and in cases where

the requirements are evolving or changing on a continuing basis and frequent delivery of evolving capabilities are to be delivered to users, for example, in a web-based application. The Agile development process is described in Section 2.5.3. As related there, planning for an Agile project involves:

- working with the customer to develop the product vision,
- determining project duration and level of effort to be applied,
- obtaining the commitment of a knowledgeable customer representative for ongoing involvement in the project,
- establishing the development environment,
- planning the frequency of iterations, and
- planning the frequency of delivery of evolving capabilities to users.

In addition a design metaphor must be established by the developers and the particular version of an Agile process to be used must be adopted and accepted by the project stakeholders; the Scrum version of Agile development is discussed in Section 2.5.3. A plan for ongoing reviews with customers, developers, and other stakeholders must be established, as must plans for periodically reviewing the planned and actual state of affairs and for reconciling differences. As with all software projects, initial assessment of risk factors and plans for ongoing risk management must be established. Planning an Agile project thus involves:

- developing the product vision;
- determining the project duration and level of effort;
- obtaining commitment of a knowledgeable customer representative;
- establishing the development environment;
- planning the frequency of iterations;
- planning the frequency of deliveries;
- establishing a design metaphor;
- adopting a version of Agile development;
- planning for ongoing reviews by the stakeholders;
- planning for periodic reviews of project status;
- conducting an initial risk assessment and risk mitigation; and
- planning for ongoing risk assessments and mitigation activities.

4.4.2 Balancing Agility and Discipline

As related in Chapter 2, the text *Balancing Agility and Discipline* by Boehm and Turner contrasts plan-driven and Agile approaches to software development and presents a middle-ground approach to achieving a balance that incorporates aspects of both approaches based on each particular situation [BOEHM04].

4.5 A MINIMAL PROJECT PLAN

At minimum, a plan for a software project, whether plan-driven or Agile, must include the following information: