Chapter 7: Defining the Goal and Scope of the Software Project

Up to this point, we have described processes and life cycles as a framework for performing software project management. This chapter describes the beginning of formal project creation. In this chapter, we describe how the project and product processes fit together to define a unique project instance. The five steps of any project are **Why**, **What**, **How**, **Do It**, and **Did It**. These steps are where the five PMI® project processes of *Initiating*, *Planning*, *Executing*, *Controlling*, and *Closing* are executed, regardless of the nature of the project. We saw how the five steps mapped into a business, product, and project life cycles.

We'll see what is required to define the goal and scope of a unique software development project. The techniques to be described are the SMART method for goal and objective completeness, and the Is / Is Not technique for goal and objective clarification. The goal and scope become the key content for the important documents that define the project. These include the Charter, Scope of Work statement, and Software Project management Plan.

Where We Are in the Product Development Life Cycle

Where are we in the Basic Software Life Cycle Model that serves as our map? The first three steps of Figure 7-1 are where the goal and scope of the project get defined. In Concept Exploration, the goal of the system will be identified at a high level in the statement of need. In the System Exploration and Requirements steps, the scope will be refined until it is understood and manageable.

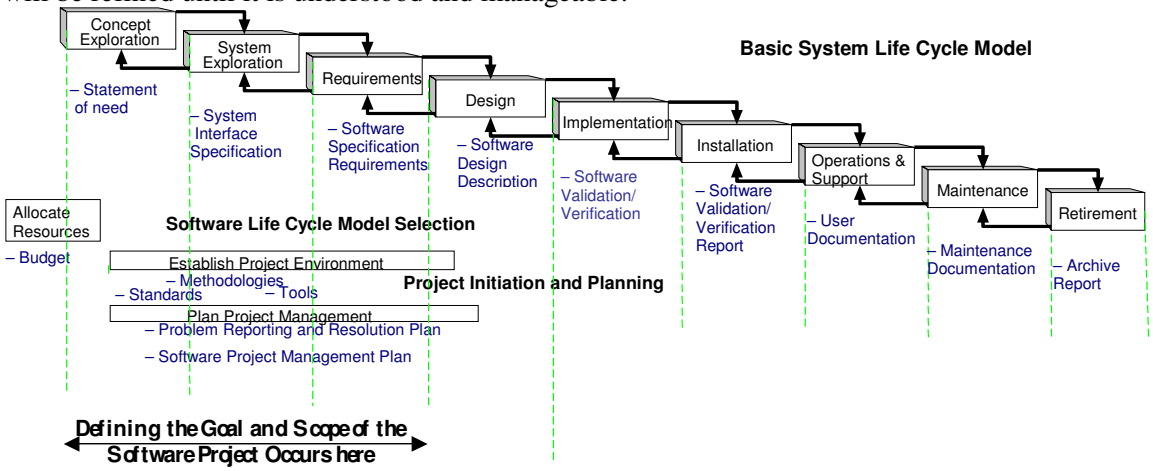
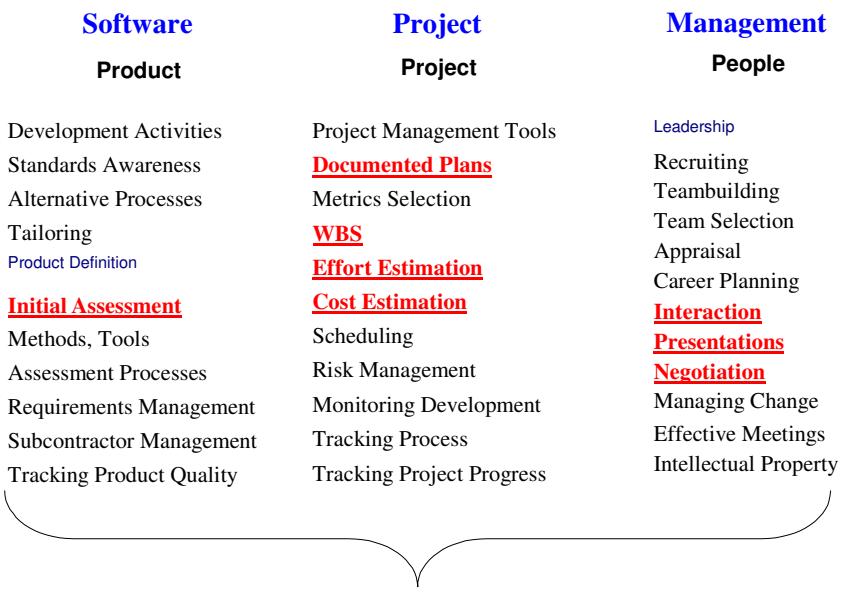


Figure 7-1. Where Goal and Scope Definition Occurs in the Life Cycle

Chapter 7 Relation to the 34 Competencies



Defining the Goal and Scope of the Software Project as it relates to the 34 Competencies

This chapter concerns the competencies shown in Figure 7-2. Defining the project goal and scope directly relate to the initial assessment and product definition product development techniques, and the project management skills related to estimating and documenting, using the people management competencies of leadership, negotiation, and communication and interaction.

Figure 7-2. Defining Goal and Scope of Project and 34 Competencies

Product Development Techniques

3. Defining the Product—Identifying customer environment and product requirements
7. Performing the Initial Assessment—Assessing difficulty, risks, costs and schedule

Project Management Skills

12. Building a WBS—Building a work breakdown structure for a project
13. Documenting Plans—Identifying key components
14. Estimating Costs—Estimating cost to complete the project
15. Estimating Effort—Estimating effort required to complete the project

People Management Skills

26. Interaction and Communication—Dealing with developers, upper management, and other teams
27. Leadership—Coaching project teams for optimal results
29. Negotiating Successfully—Resolving conflicts and negotiating successfully
31. Presenting Effectively—Using effective written and oral skills

Learning Objectives

The main point of this chapter is to define the goal and scope of the project so clearly that measuring progress is easy, and that there is no ambiguity that the project goals have been achieved.

There are several techniques and tools available to the project manager to define the goal and scope clearly. Among them is the **Is / Is Not** technique, and the **SMART** goal checklist.

Upon completion of this chapter, the reader should be able to:

- Explain why project planning is valuable
- Describe the value and contents of a project charter, scope of work statement, and software project management plan
- Create useable goal and objective statements
- Explain how to make crisp clean boundaries around the project's scope
- Describe the five project process steps for any project, and how they relate to the product processes of business and software development
- Show how the Project Charter, Statement of Work (SOW), Software Project Management Plan (SPMP), and Software Requirements Specification (SRS) relate to each other

Project Planning

Why plan? Won't the plans just change anyway? Why waste a lot of time and effort writing down what we think we're going to do, just to have blind luck change it all before we're done. Why not "just do it" and handle whatever happens whenever it happens? These are good questions in a world changing at Internet speed. Whatever life cycle roadmap you choose, uncertainties abound, and your team may end up achieving something different than what they originally envisioned.

Referring to the mountains of planning documents prepared for the historic D-Day invasion of Europe on June 6, 1944, Gen. Dwight Eisenhower said, *"Plans are nothing. But planning is everything."* He recognized that extremely well thought out and detailed plans are also extremely fragile in the face of fate and uncertainty. But he also knew that the depth of understanding of the problem space that came from the work to prepare the plans would enable his team to react intelligently to new situations. They would be better prepared to seize new opportunities and avoid serious mistakes. Good luck is where preparation meets opportunity.

Project planning is the process that lays the framework for how the project will be run. It includes the definition of the project's objectives, selection of an appropriate life cycle, and establishment of the policies, procedures, and processes necessary to achieve the objectives. How much framework you need to write down depends, in part, on how mature your organizational environment is. If your team is a freshly hired group of people from many different backgrounds who haven't worked together before, with you or your organization, then more structure is required to guide them. If most of your team has been working together for a while, then only a skeleton framework and simple checklist is required. The team probably already has enough cultural heritage to flesh out the plans well enough using established reporting structures, definitions, and work products.

The old jokes about planning are "Ready, Fire, Aim", and "You go ahead and get started coding while I go upstairs and find out what they want." Also, many projects have followed the tongue-in-cheek life cycle shown in Box 7-1. Let's hope your projects don't follow this life cycle!

Phase 1 - Project Initiation
Phase 2 - Wild Enthusiasm
Phase 3 - Disillusionment
Phase 4 - Chaos
Phase 5 - Search for the Guilty
Phase 6 - Punishment of the Innocent
Phase 7 - Promotion of the Non-participants
Phase 8 - Definition of the Requirements

Box 7-1. A Commonly Encountered Life Cycle in Low Maturity Organizations.

Regardless of the life cycle chosen for the project, there will be two distinct sets of processes followed:

- **Project processes** - describe how the project team will define and execute the product development processes (see Figures 7-3 and 7-4).
- **Product processes** - describe how the software product will be built (see Figure 7-1).

The product processes are introduced in chapters 3, 4, and 5, and are discussed in detail in Chapters 19 through 24. The project processes were introduced in Chapter 3 and will be discussed in detail here, as we look into how to execute a specific instance of a project plan.

Project Processes

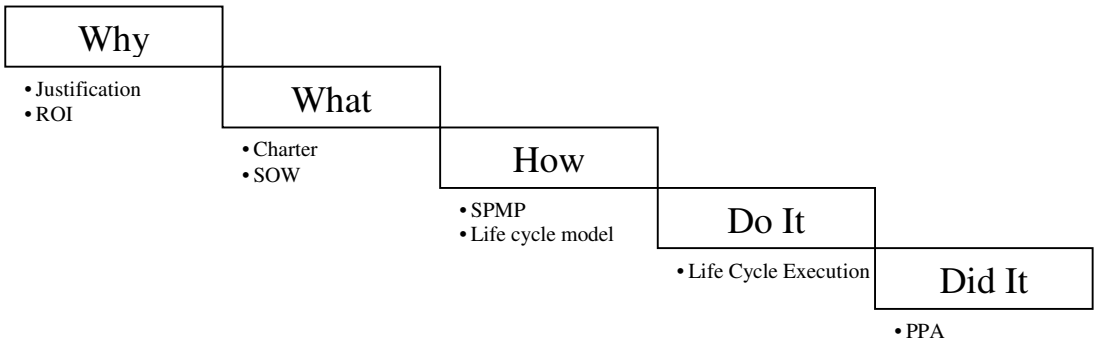


Figure 7-3. Project Process Framework

Why

Every project needs a good reason to exist, and this step ensures that there is at least one. Using business analysis suitable for the organization, the project manager should perform an opportunity analysis and prepare a return on investment (ROI) statement for the project. This raison d'etre may be just a statement indicating that the project has high strategic value and must be accomplished for other projects to exist, or it may be a complete ROI prepared by financial experts showing net present value (NPV), internal rate of return (IRR), payback period (PBP), or other acceptable calculations. Whatever the justification, it should be recorded, and usually becomes part of the project charter.

What

Once a good reason is identified for proceeding with the software project, the goal and scope of the project can be defined to distinguish it from on-going operations. Called a project charter, this is usually a simple outline statement of what the project will accomplish, including major deliverables, a rough high level schedule, initial estimate of resources needed, and the expected return to the organization for the effort invested. The charter has two purposes:

1. to formally document the existence of a software development project and separate the work of the project from on-going operations such as maintenance and support
2. to obtain management approval for the work to be done, and get commitment for the resources to do it

The charter may take the form of a legal contract and statement of work (SOW) for external work to be executed by a third party outside of the project manager's direct control. In this case, the details of the "**How**" may be left up to the contractor.

How

With the authority granted by charter approval, the software project manager can employ the many other software engineering and people skills needed to define the product and manage a team to develop and deliver it. This is the heart of software project planning. The major work product of this step is the Software Project Management Plan (SPMP). The SPMP explains (in an appropriate level of detail for the maturity of the performing organization) how the life cycle steps will be performed. These may vary for every project even though the basic life cycle used is the same. This is the roadmap that the project team will follow to prepare the software deliverables and meet customer expectations. Usually for large or long duration projects, iterative planning is used, with detail planning only done for the next immediate step of the life cycle as the project progresses.

Do It

When the Software Project Management Plan (SPMP) is completed and approved (by the customer or by management), then the team can execute the plan, following the life cycle steps chosen for this particular project instance. Even if a spiral model using rolling wave planning is chosen, the SPMP should be the roadmap to guide the team toward project completion. This step uses most of the information presented in this book for the 34 competencies.

Did It

Commensurate with good process management, an evaluation step should be performed before closing out the software development project. This is the Post Performance Analysis (PPA) step and results in a PPA report documenting lessons learned and recommendations for project or product process improvements for the next similar project that the organization attempts.

These five steps are required for every project, large or small, short duration or long, regardless of the project's intended output. The variables of size, scope, cost, schedule, complexity, and risk determine how much rigor and documentation is required in each step, and which development life cycle should be used. Even a small project such as assembling a child's bicycle for a birthday present is a project that requires thinking through these five steps. But it isn't likely that you would make an effort to document it for posterity.

The Project Management Institute (PMI®) cites five processes necessary for any project, or phases within a project: [1]

- Initiating
- Planning
- Executing
- Controlling
- Closing

Figure 7-4 shows that the *Initiating* processes are where the **Why** step is handled. It includes enough of the **What** step to describe the project at a high level. The rest of the **What** step and the entire **How** step is handled in the *Planning* processes. The **Do It** step is composed of the *Executing* and *Controlling* processes. And the **Did It** step is covered by the *Closing* processes.

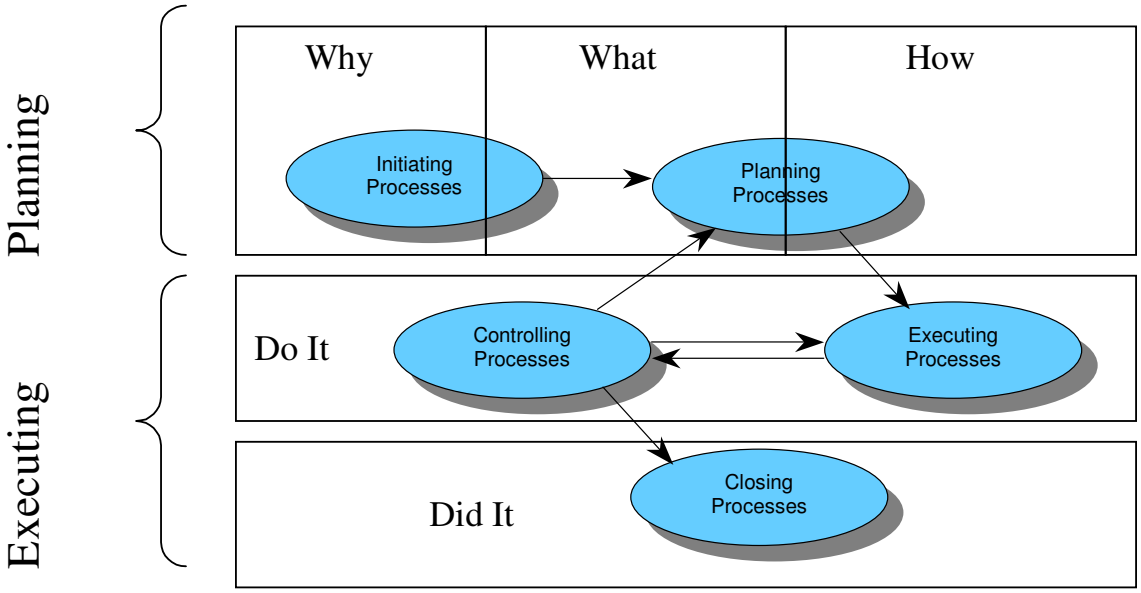


Figure 7-4. Project Process Integration Map

For the Product Processes shown in Figure 7-1, the project planning process starts in concept exploration by ensuring that you are pursuing a solution to the right problem for your customer, else much effort is wasted. This is where the "Why" of a project is explored.

In Chapters 16 and 17, we will look into the development of detailed requirements.

Planning is an iterative process, usually done repeatedly during the course of a project, as conditions change and new knowledge is gained. But the objectives set early should remain fairly static; else the project will wander aimlessly, and should be cancelled and then redefined. Objectives are usually interrelated. If some objectives conflict, then management must prioritize them, else the project will not be successful.

The general approach fits into the framework described in Figure 7-5.

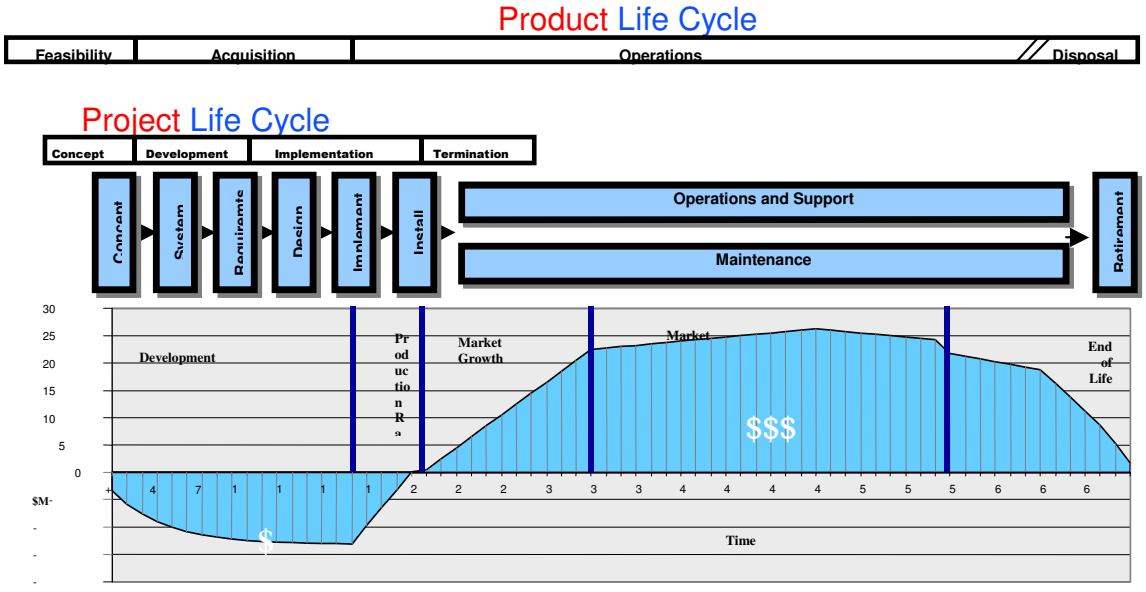


Figure 7-5. Relationship Of Project And Product Life Cycles For a Software Project

What is "The Goal"?

Every project needs at least one goal. Most have multiple goals. Sometimes these are called the project objectives, or are collectively referred to as the mission of the project. Actually, we believe that the mission of the project is to accomplish the goals and/or objectives. Whatever label is used, defining them clearly and crisply can be one of the simplest and most beneficial things done during the whole software development project. A fuzzy goal will most likely lead to fuzzy results. Managing software projects well is mostly about managing expectations and foreseeing risks.

The software project manager always has at least one goal: to finish the project. This comes from the definition of what a software project is: A **unique, temporary** endeavor with defined **start and end date**s to achieve one or more **objectives** within the **constraints** of cost, schedule, and quality performance.

Often, what seems like an obvious software project goal may not be seen the same way by everyone. This is why is important to write it down, and interpret it for the project team. For example, a software development project to build a web-based timesheet data entry system may be viewed as:

- an internal tool development effort by the engineering manager,
- a training exercise by the recently hired programmers,
- a requirement before starting a new software development project for an external customer by the general manager, or
- as a marketing tool to demonstrate the capabilities of the development team by the sales staff.

Each of these views implies a different level of robustness, ease of use, and maintainability for the final end product software deliverable. Unfortunately, many project managers would simply state that the project goal is to "Build a timesheet data entry system", and leave it at that, inviting misinterpretation later. To clarify the goal for all stakeholders, this project could have the goal and objectives defined as in Box 7-2.

Goal: Build and successfully deploy a web-based timesheet data application entry system for the engineering department's internal use before the beginning of the next major external software product development project.

Box 7-2. Example Goal Statement

This statement is what some call "the elevator speech". It is a clear but concise description of the project that could be conveyed to the CEO informally on the spur of the moment in a short elevator ride, without prior notice. In the example, it states that the project is for internal use (implying internal benefit but no revenue directly associated with it), it includes successful deployment (not just development), specifies who the customer is (the engineering department), and cites a time frame (before the beginning of the next major external software product development project).

It is very valuable for the overall goal of the project to be stated in terms that are easily understood and repeated. At any project meeting, anyone on the project team should be able to recite the project goal statement, if asked. This keeps the project focused, and may be used as the opening to the "elevator speech" about the project by anyone asked to explain what they are doing to an uninformed observer.

Setting Clear Objectives:

In the previous example, the objectives in the goal statement have certain characteristics that make it clear. Most objectives tell *what*. The best ones also imply *why*. Good project objectives are:

- Focused on deliverables, not just processes - the customer ultimately cares about the final end product (timesheet system), not the processes needed to get to the end product (SPMP, SRS, etc.)
- Measurable and testable ($, %, Mkt. Share, Dates) there is something quantifiable to measure and test, usually quality related ("successful" implies acceptance criteria are defined)
- Action-oriented the goal implies actions to achieve (the verbs "Build" and "deploy" compliment the noun "system")
- Conversational could be recited and explained in a few seconds (the elevator ride)
- Doable (within your authority) the goal is reasonable and does not imply trying to solve world hunger (many accounting applications like this have been done before)
- Communicated the team knows it, and it is published in the project charter (it's also the elevator speech)

Setting clear objectives goes a long way toward starting a successful software project. [2]

Once objectives have been set, measurable sub-objectives for some or all may be defined so that cost and performance may be tracked. Rather than the "big bang" approach of having one or just a few objectives, it is good project practice to set up a number of smaller measurement posts along the way. Many small milestones are better than one big one at the end of the project. The team should have a part in establishing their own objectives and sub-objectives.

A useful technique for defining clear objectives is the **S.M.A.R.T.** method. **S.M.A.R.T.** are the initials for:

- **S**pecific
- **M**easurable
- **A**chievable
- **R**ealistic
- **T**ime-bound

For any objective, determine the specific results and performance targets necessary to meet expectations. Are the objectives **Specific**? Do you and your sponsor/customer both agree on the results needed for each of your project's objectives? Are the objectives **Measurable**? How will you know you are achieving results? What does "meet expectations" mean to you and your customer for each of the objectives? Are the objectives **Achievable** and attainable? Why or why not? What support would you need to attain them? Are the objectives **Realistic** and relevant? Do they address the customer's requirements and real needs? Do they align with the key success factors for the organization, the business goals and strategies? Are your objectives **Time-bound**? Are there specific dates that the objectives should be achieved? Is there a clearly understood reason for each of the dates? What is the driver behind the dates (e.g. your customer's customer needs the product then)?

The following is an example of a (relatively) short duration objective that is part of a larger quality systems improvement project in a large multi-national corporation. Note that it is specific and measurable, achievable for the average corporate quality department, both realistic and relevant to the department and corporation long-term goals, and time-bound (in this case a specific due date, not a specific duration like "within 6 months from project start").

- Benchmark two other companies' automated line manufacturing processes according to ISO9000 standards. Complete the project by the end of the third quarter at a maximum budget of US$30,000.

What is the Scope of Work?

Often part of the Software Project Management Plan (SPMP), but sometimes a separate document (or several), is the **Scope of Work** (SOW), **Statement of Work** (SOW), or sometimes, **Statement of Requirements** (SOR), although this may get confused with the **Software Requirements Specification** (SRS), which contains the detailed requirements. The SOW contains just enough specific details and specifications of components packaged so that they can be given to a sub-contractor for execution. The SOW may be incorporated within, referenced as an appendix in the SPMP, or in a completely separate document if it needs to be separated from the main document for security or other reasons.

Setting Boundary Conditions:

It is usually easy for a project team to identify what the software project should include than what it should not include. The goal and objectives statements describe what is to be part of the final project scope. What is often harder to describe is what it will not include, but this is absolutely necessary to help define the edges of the project scope. Later, preparation of the Software Requirements Specification (SRS) will detail the contents of what is included more finely.

Use the **IS / IS NOT** technique to help draw crisp boundaries around the project scope, and its objectives individually. This technique is simple. For each goal or objective, your team uses brainstorming techniques to define what it is, and make a list for the team (see Figure 7-6). Next, brainstorm a list of what it is not and make that into a list. Both lists can then be used to make a list of assumptions about the project.

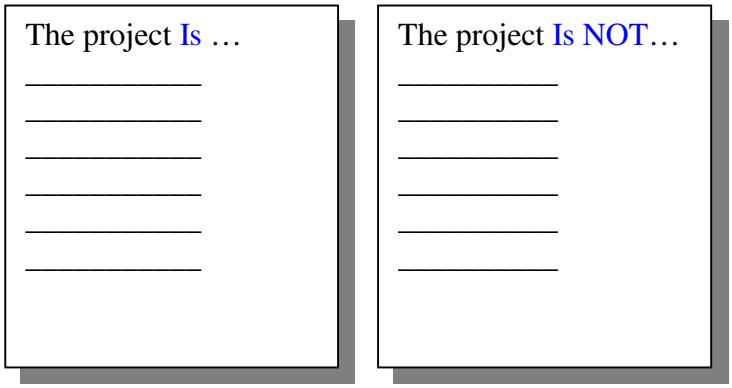


Figure 7-6. Is / Is Not Technique

This exercise pulls out hidden assumptions that team members had about the project scope or its work products. Better to find that out early, rather than get deep into the project and discover misunderstandings about the project objectives and scope For instance, using the example goal statement in box 7-2, we might say that the project:

IS:

- Internal
- A Timesheet Data Entry system
- Web-based
- For the Engineering Department
- To be deployed before we begin the next major external project
- To be successful per existing application deployment criteria
- To serve as a training project to familiarize the team with the new project management process

IS NOT:

- Intended for use by other departments or external customers
- A full blown labor accounting system
- Intended to be accessed by PDA's or wireless web cell phones
- Required to interface to existing earned value management programs such as MS Project
- To be in beta test mode when the next external project begins
- To use outside contractors for development

These characterizations become a way to crisply define the edges of the project's scope. They translate easily into assumptions to be recorded for the project charter and SPMP. The assumptions then become the first risks of the project, since if any of them are violated then the project scope is breached.

Sometimes the scope of work (SOW) is large enough to warrant its own document, instead of appearing as statements in a charter. However, at this early stage in the life cycle, the SOW is just a rough planning item. Sometimes, this document is referred to as the Statement of Work (SOW), Statement of Requirements (SOR), or Software Requirements Specification (SRS). Often, the SOW/SOR/SRS is subdivided into pieces that fully describe what a subcontractor is expected to do, in relation to the overall requirements. In these cases, the SOW/SOR/SRS may contain non-spec items from the Project Management Plan, such as status reporting protocol, payment schedules, and legal statements. Details and templates for SOW construction vary with industry and domain, but generally carry the specifications of the product to be built in a format that can be separated from the other processes of the project.

Project Charter

> **charter** *n.* [**char**'ter] a document that formally recognizes the existence of a project [3]
> Includes:
>
> - business need
> - product description
> - major assumptions

Now that the individual objectives have been identified, and the overall project scope is understood and agreed upon. It is time to prepare the project charter. Basically, the project manager will capture the high level objectives and the scope, and then add other pertinent high-level information so that a sponsor and/or customer can approve it. Often, the charter is the basic document needed for project selection in the organization's project portfolio management process, where this project would be compared to others in a cost/benefit sense, and funded if selected. Project selection models and techniques are outside the scope of this book.. See the references section for more information.

Project Charter Contents

The charter may sometimes be called by other names, such as the "project initiation document" (PID), "scope baseline", or just "contract" (usually for external work). It may be produced in many forms, such as a narrative document (most common), fill-in-the-blank forms (paper or software application), or as spreadsheets for extensive financial justification.

The charter contains the "Why" and "What" of the project processes discussed at the beginning of this chapter. It should contain brief statements (very brief! - don't qualify every phrase!) about the following:

Objectives - what the desired outcomes are
Functions – major features and/or processes
Performance - generalized specifications
Constraints - limitations of the environment
Scope - boundaries of the project
Costs/Benefits - Rough order of magnitude estimates

Be sure to answer the usual questions about a project that uninformed observers tend to ask. Questions such as: What is the Reason for the Project? Is it to seize an opportunity, solve a problem, increase revenues, decrease costs, or a combination of these? Be sure to have an answer for the typical newspaper reporter's questions: Who? What? Where? Why? When?

Typically, the charter is a short 1-3 page letter, memo, or email document – just enough to secure management and/or customer approval for the project. Sometimes the charter is labeled as the Software Project Management Plan (SPMP) and uses its framework, but includes only the sections pertinent to the selection process at this early point. Sometimes it is better to make it a separate document and merge it later because executives don't like to see large documents come their way (like SPMP's with lots of extra sections, revision blocks and change control, specifications, appendices, etc.). Remember that the purpose of the charter is to concisely represent the project at a high level, to get management approval and support (and a signature!). From there, you can flesh out the rest of the project planning, because you will have the *authority* of approved sponsorship (and funding)! [4]

Software Project Management Plan

This is the most important document of a project. It defines how the project is supposed to be executed, and what it is going to produce. Next in importance would be the Software Requirements Specification (SRS). The SPMP should contain definitive project information that includes:

Charter - elements from the project charter that define the project, including deliverables
Organization - how the project will be organized and executed to produce the deliverables
Process - details of the managerial and technical processes that will be used during the project
Work Breakdown - work breakdown and work package details
Schedule - schedule, dependencies, and resources
Budget - Budgetary and Definitive estimates

All of these items are related, and the SPMP actually evolves over a period of time as the various items come together. The process usually starts, though, with incorporating the elements of the approved charter into the SPMP. Since the project is approved beyond the concept stage, it can proceed to the definition (What) and planning (How) stage, where the detailed planning takes place.

Major Elements of the Software Project Management Plan (SPMP)

Usually completed from a template such as that described by *IEEE Std 1058 Standard for Software Project Management Plans*, the Software Project management Plan (SPMP) describes the "How" of the project processes discussed earlier in this chapter [5]. It describes how the project team will implement the life cycle software development processes of the chosen life cycle.

The project charter information is integrated into appropriate sections of the **Software Project Management Plan** (SPMP). The rest of the document contains sections that include:

- Project Overview and Deliverables
- Project Organization
- Managerial Processes

- Technical Processes
- Work Packages, Schedule, and Budget

Other chapters in this book provide details about how to identify and represent the information necessary to complete an SPMP. The depth and breadth of the coverage for each section is a matter of judgment according the type and scope of the project. When completed and approved, the SPMP becomes the benchmark for *controlling* the project during execution (the "**Do It**" step).

How These Project Planning Documents Relate

As shown in Figure 7-7, the flow of information starts with defining the goal, objectives, and high-level scope of the proposed software project. These items are usually presented in a project charter document. For a large project, or one to be sublet through a contracting process, the charter may be supported by a Scope of Work (SOW) document containing more details outlining the work to be performed.

When the charter containing the Why and What for the project has gained management approval, then the information in it is integrated into the Software Project Management Plan (SPMP) where the details of **How** the project will be executed are described at a level suitable for the size and scope of the project, and the maturity of the project team and organization.

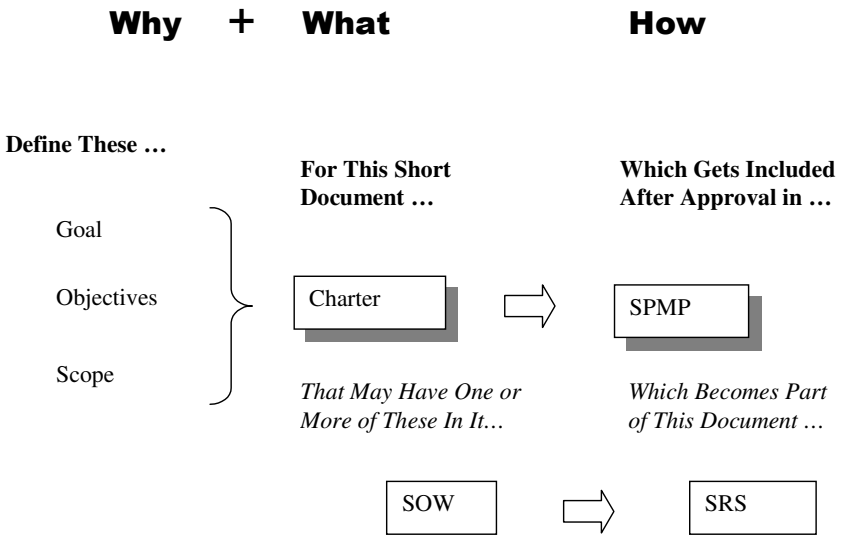


Figure 7-7. Relationship of Planning Documents

Summary

In this chapter, we have described how the project and product processes fit together to define a unique project instance. The five steps of any project are **Why**, **What**, **How**, **Do It**, and **Did It**. These steps are where the five PMI® project processes of *Initiating*, *Planning*, *Executing*, *Controlling*, and *Closing* are executed, regardless of the nature of the project. We saw how the five steps mapped into a business, product, and project life cycles.

We've seen what is required to define the goal and scope of a unique software development project. The techniques described were the SMART method for goal and objective completeness, and the Is / Is Not technique for goal and objective clarification. The goal and scope become the key content for the important documents that define the project. These include the Charter, Scope of Work statement, and Software Project management Plan.

Exercises for Review

Problem #1

What is (are) the goal(s) for the case project?

Problem #2

Define the SMART features of the goal for the case project.

Problem #3

What is the case project's scope?

Problem #4

Explain the difference between project scope and the project's goal.

Visit the Case Study

You have had the PMP approved by the management chain in your department. Mr. Lu, the CRM direct interface for your corporation wants a teleconference with you and the project leads. His management has approved the scope of the original prototype but wants a co-proposal for a second prototype to follow on within 60 days of the first prototype. This would be an extension to the ARRS that would include Beijing, Zhengzhou, Shanghai, and Tianjin. This is excellent news for your corporate management since you have a major manufacturing facility in Tianjin.

References

[1] Project Management Institute, *A Guide to the Project Management Body of Knowledge*, 1996

[2] Kerzner, Harold, *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*, 6th Ed., Wiley, 1998

[3] Dictionary.com http://dictionary.com/

[4] Lewis, James, *Project Planning, Scheduling, and Control*, Irwin, 1995

[5] IEEE Std 1058 Standard for Software Project Management Plans, *Software Engineering Standards Collection*, 1993

Suggested Readings

1. Adams, John, *Principles of Project Management*, Project Management Institute, 1997, 285 pages
2. Lewis, James P., Project Manager's Desk Reference : A Comprehensive Guide to Project Planning, Scheduling, Evaluation, and Systems, 2nd edition, McGraw-Hill, 2000, 546 pages

Web Pages for further information

1. http://dictionary.com/ Dictionary.com, online English Dictionary
2. http://www.pmi.org/ Project Management Institute (PMI®) This is the premier organization in the US representing PM interests. It is the certifying body for Project Management Professionals (PMP). Since its founding in 1969, Project Management Institute (PMI®) has grown to be the organization of choice for project management professionalism. With over 60,000 members worldwide, PMI is the leading nonprofit professional association in the area of Project Management. PMI establishes Project Management standards, provides seminars, educational programs and professional certification that more and more organizations desire for their project leaders.
3. http://www.sei.cmu.edu/activities/str/indexes/glossary/ SEI Glossary