CHAPTER 3

Project Initiation

In the mid eighties and early nineties, industries were jumping on the dot-com bandwagon. You may remember this as being a time where dot.com companies were popping up left and right, engineering salaries went through the roof, and the stock market was at an all time high. The dot.com era was making people millions. All you had to do was call yourself a dot.com startup company and you were worth a fortune. Most of those companies were not profitable, and many of them had no chance of ever being profitable. Their stock was hugely overpriced, and the companies often lacked the business leadership necessary for long-term success. Companies were launching projects haphazardly just to keep up with the competition and to show that they were being productive. Those projects were launched without clear business or technology goals and objectives, leaving the project without the strategic planning and direction needed for long-term business success. With this frameset, many companies were purchasing equipment and services that would work with existing systems—thinking that this would be conducive to faster time to market. This was not always the case. It took time to get projects up and running, it was often very difficult to integrate the old legacy systems with the new technology, and by the time the product was finally ready, a new product was introduced and the old one was already out of date.

In the midst of this dot.com madness, there was also the Y2K bug to contend with. Many of the old technologies were built many years ago, and no one was looking forward as far as the turn of the century and planning for how these systems would hold up. They probably never thought that the systems would be around that long. As a result, it became an engineering standard in many companies to write dates with a two-digit year. The default year (if the system could not recognize what year was entered) was usually 1984. Therefore, when we changed from 1999 to 2000, the system did not know whether it was the year 1900 or the default year—1984. There were thousands of projects being launched to address the Y2K bug, many of which were probably not needed. Nevertheless, there was a necessity to be seen doing something about this due to paranoia that companies could be sued if they did not show that they were addressing the year 2000 issues. This used up many of the market's engineering resources and created a technology boom. Many could not see that, at some point, the bubble had to burst.

It was a nice ride for a while, but the bubble did finally burst. Many dot.com's went out of business. Those investors who held onto their dot.com stocks for too long lost a fortune equivalent to the ones they thought they were going to make. Venture capitalists became much more savvy about technology businesses and were not throwing around anywhere near the many millions of dollars they had in the previous decade. The dot.com industry matured, and those companies that were left standing made serious changes to their business and technology strategies to enable them to compete in today's marketplace.

Today, companies are looking at the bottom line. They want to see the *return on investment* (*ROI*), and they want to know exactly how long it will take to achieve the ROI. Anything longer than a couple of years is a pretty hard sell these days. Proposals for senseless products are no longer getting into the boardroom. They are being nixed much earlier in the process. Processes are being put in place to ensure adequate due diligence on project proposals. Executive-level management is taking the time to review the proposed projects and their long-term goals and objectives. Initiatives are not being given the greenlight unless a justification for overall business success is prevalent. However, this does not mean that companies have stopped thinking outside the box. In fact, this kind of thinking is even more necessary in today's marketplace. Differentiating oneself from the competition by coming up with creative and unique concepts is the key to initiating successful projects.

Once a project concept has been presented to executive management and accepted as a feasible idea, the project *proposer* will be tasked with creating a formal *Project Proposal*.

The initial *Project Proposal* is a high-level plan that includes estimates for resources required to complete the project. Once the *Project Proposal* is complete, a decision will be made whether to go forward with the project, to change the scope of the project, or to terminate the project. If the project is greenlighted, the proposal will continue to the project definition stage.

There is a huge amount of work that needs to be accomplished in the planning stage of a project. So much so that it is often a wonder that the project got started at all! If you want your projects to be completed on time, within budget, and with high quality and if you want them to be successful, you need to invest the appropriate amount of time and effort in planning. It is an area that is often overlooked or rushed through very quickly. The eagerness to get started actually coding or building something can overwhelm common sense and cause companies to skimp on this phase. The more thorough the planning, the fewer problems you will encounter during the project creation. This will lead to higher productivity, higher team morale, and a higher-quality product.

The Planning phase of a project includes the following steps:

- Project concept
- Request for Proposal (RFP)
- Project proposal
- Project greenlight
- Project definition
- Project approval
- Project kick-off
- Project plan
- Project approach

In this chapter, we will cover the steps in the process from project concept through project greenlighting, the Project Initiation Process is shown in the Workflow diagram in Figure 3.1.

The project concept and proposal incorporate all the meetings, research, and reporting that is necessary to get the project greenlighted.

Project Concept

The *concept* is the idea or the reason that the project is deemed necessary. It can be a simple idea that can be implemented in a matter of weeks, or it can be a complex concept that will require a monumental engineering feat and a few years to accomplish. The concept describes either a problem that needs to be solved or a solution to a problem.

The concept should, ideally, contain the following elements:

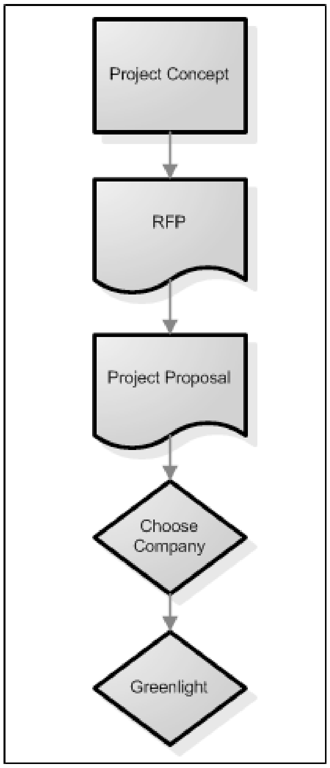


Figure 3.1: Project Initiation Process Workflow.

- Business Case—The business case describes the reason that the presenter believes the project is necessary. For example, the presenter could use facts about where the business is currently compared to where it needs to be in the future or financial statistics showing negative trends that can be reversed by implementing this project.

- Benefits—This is the value that the project will have to the organization. This should be described in terms of how it will help the company meet its business goals and objectives. If there is a company strategic plan in existence, the concept could be mapped to a specific goal. For example, increasing client satisfaction, reducing call center costs, and increasing development team productivity.

- Description (Problem or Solution)—A description of the problem that needs to be solved and the proposed solution (if there is one). At the concept stage, it is not always possible to know how a problem can be solved. If it is critical that the problem must be solved, it can be presented as a concept and the solution can be developed during the proposal stage.

- Initial Estimate—If the concept is still very vague, it may not be possible to include an estimate. If this is the case, it is good practice to decide on a "not to be exceeded" figure for what is considered a reasonable amount to spend to fix this specific problem. The estimates will be as vague as the description. Thus if the description is really well defined, the estimates will be much more accurate.

Companies often use brainstorming techniques to come up with concepts for projects. Brainstorming is an excellent way for a group of people to bounce ideas and thoughts off each other and work on them to come up with well-defined concepts for projects.

The Brainstorming Process

Brainstorming is the process of formulating ideas that can be refined into a concept for a potential project. Brainstorming can be done with small or large groups of people. You can have a successful brainstorming session with as few as 2 or as many as 100 people. Ideally, for the best results, limit brainstorming sessions to groups of 5–20 people. Meetings with more than 20 people can be difficult to facilitate because there will be too many people wanting to contribute ideas and comments at the same time. If people are waiting too long to contribute, the conversation may have gone in a different direction by the time it gets to their turn. They will either feel that it is too late for them to contribute to that particular topic so they pass or attempt to bring the conversation back to where it was previously, thereby losing the natural flow of the brainstorm for that topic.

There are many different techniques and tools for brainstorming. We describe one technique that works well. Experiment with the process, refine it, and customize it to fit your specific needs and the needs of the group that you are working with. You may decide to set the meeting up in a less formal way. If you have only two topics and they are very clearly defined, you may want to skip some of the early steps in the process and jump right into the discussion about them.

You should use a facilitator for your brainstorming process. It should be a person (or persons) who will not participate in the brainstorming. The role of the facilitator is to run the meeting, keep time, and ensure that the brainstorming process is followed. The facilitator will keep things moving along so that the team does not get stuck on one point or digress into areas that are not relevant to the brainstorming session.

Define the Purpose of the Meeting

The purpose should include the goals and objectives of the brainstorming and, very importantly, the scope of the brainstorming. If the scope is too broad, you will find it almost impossible to focus in on a limited number of specific topics! If you have a lot of topics that you need to brainstorm about, divide them into categories and schedule separate meetings for each category. Some examples of brainstorming topics may be:

- Product A—increase revenue
- Product B—reduce costs
- Dept X—decrease run-the-business (RTB) costs
- New product ideas

Communication and Preparation

Tell the attendees what to bring with them. If you want them to think about the topics in advance, schedule the brainstorming session a week or two in advance to give them time to think and to prepare notes. If you want the team to brainstorm cold and to come to the meeting with no preconceived ideas or notions, then do not tell them the purpose of the meeting in advance. Tell them just before you start the meeting.

Rules

It is vitally important that you have rules for your brainstorming process. The following are some typical rules for brainstorming sessions, but you should customize them for your particular session:

- Both problems and solutions will be discussed.
- You do not have to have a solution in mind in order to bring up a problem that you believe needs to be solved.
- You cannot name people or blame people for problems.
- You cannot defend yourself or others when problems are being discussed.
- Everyone’s opinions are valid.
- All comments are welcomed as long as they are relevant to the topics under discussion.
- Nothing discussed in this room should leave this room.

- The facilitator is running the meeting. If the facilitator asks the group to move on from a topic or discussion, they will do so. The facilitator will capture any ideas or suggestions that are important but not relevant to the session so they are not lost. Discussion around those topics will not continue in the brainstorming session.

Brainstorming Topics and Categories

This is where you start the actual brainstorming. You should have one or more brainstorming topics that you defined in the purpose. Use those topics to come up with a list of categories. You may already have a list of categories prepared, or you may wish to brainstorm with the team to come up with categories before you start. The brainstorming will not be limited to the categories you define, but these topics will help the group to get their creative juices flowing.

Using the examples of "Increase Revenues" and "Reduce Costs," for instance, you may come up with the following categories:

- Increase per-customer revenue
- Attract more customers
- Provide incentives for existing clients to order more product
- Expand product range
- Offer accessories and related products (one-stop shopping)
- Reduce call center costs
- Reduce facility overhead

Using the categories as a general guideline, you can either use a method where the group calls out problems or ideas and the facilitator lists them on a whiteboard or flipchart or you can use the "sticky note process" (as described in Chapter 14). For this example we will use the flipchart process.

Capturing the Ideas

There are a couple of different approaches to this. The first is to have a free-flow discussion where ideas and suggestions are captured as the discussion continues. You can also go around the room and ask each person to contribute any ideas or suggestion they have, or you can just ask people to call out ideas as they think of them. Regardless of which process you use, the facilitator will capture the ideas on the flipcharts. As we mentioned before, the ideas can be problems or solutions. Both are valid in brainstorming sessions. If you have a large brainstorming group or the discussion is expected to be fast moving, it is a good idea for the facilitator to have one or more assistants to help capture the ideas in writing. If there is only one person scribing, either the discussion will have to stop to wait for the scribe to catch up or the ideas will get lost if the scribe cannot keep up with the pace of the group.

Categorizing the Ideas

Once all the ideas have been captured in writing, they need to be categorized. Some will fit into the categories that you defined at the beginning of the meeting, and some will require new categories. If there are ideas coming up that are clearly outside of the defined scope, capture them in an out-of-scope category. Those ideas may be very useful later for meetings or projects where they fall within the scope.

Summarizing and Prioritizing Categories

Each category should be summarized by the facilitator or a group member. The out-of-scope category does not need to be summarized.

Selecting Categories for Further Brainstorming

The team will need to prioritize the categories and ideas to decide which ones will move forward into the next round of brainstorming. Prioritizing can be done using a voting system, or it can be accomplished by group discussion. Depending on how much time you have available and how many ideas have been chosen for further discussion, you may need to schedule additional brainstorming sessions to discuss them. If the team is brainstormed out, you may decide to end the initial brainstorming session and continue the discussion of individual ideas in a subsequent meeting or meetings.

It is advisable not to try to discuss more than two or three ideas in one session. Brainstorming is mentally a very intense process, and it is easy to get burnt out and start to lose the creative edge if you try to push the group too far in one session. For some brainstorming sessions, one idea per meeting will be sufficient.

Brainstorming Individual Ideas

Brainstorming on the individual ideas (problems or solutions) can either be done by the whole group or it can be given to separate subgroups (brainstorming focus groups). It may be that to fully explore a specific idea, the group will need some specialized technical or subject-matter knowledge that was not available in the initial brainstorming session. It may also be that the initial brainstorming group was a large group (of, say, 20 people), and it is usually a good idea to have much smaller subgroups.

The brainstorming focus group takes one idea at a time and has an open discussion about the idea. If the idea is a problem, the group will first need to come up with some potential solution ideas before they can brainstorm on the potential solution or solutions. If the idea is a solution, the team should ensure that it fully understands the problem that the solution is designed to solve before brainstorming on the idea.

The group should be assigned a team lead and a note taker. As the team works its way through various scenarios and comes to conclusions about what will and what will not work, it is imperative that the note taker captures the key points and accurately summarizes the discussions. The brainstorming focus group should conclude their meeting(s) with a fully documented concept that is ready to be presented to the authorized approver and decision maker.

Prioritizing Concepts

Once all the ideas have been brainstormed and developed into project concepts, the concepts may need to be prioritized. The prioritization may be done by the original brainstorming group, or the concepts may go to a separate business or product management group for prioritization.

Finalizing Concepts

When the list of prioritized concepts is complete, the development team will most likely be asked to put some initial estimates on them. The business groups will then estimate ROI and other benefits before presenting them as concepts for approval to the senior management team.

Brainstorming can be used for more than just project and product ideas. It is a great tool for identifying process improvements, ideas for team events, and ideas on how to implement specific features or code for an approved project.

How Initial and High-Level Cost and Time Estimates are Generated

The initial and high-level cost and time estimates for projects are generated using historical data and information gathered both internally and externally from employees and outside vendors. Initial costs are generally required when submitting project concepts for approval to move forward to the proposal stage. High-level estimates are needed at the proposal stage of a project. For outsourced projects, many of the initial costs are estimated during the creation of *Request for Information* (*RFI*) and *Request for Proposal* (*RFP*) documents.

To estimate time and cost resources for a project prior to a project definition or project plan being created, you will need specific pieces of information and preferably access to people with some historical perspective on development time and costs. Initial costs are used when deciding whether to propose a project. The estimates are not expected to be 100% accurate at this stage. It is likely that you will be asked for ballpark estimates, but bear in mind that the estimates need to be based on some realistic foundation. Alternatively, you may be required to submit an estimate with a low and a high threshold ($120,000–$160,000 for example).

To estimate project costs you will need the following information:

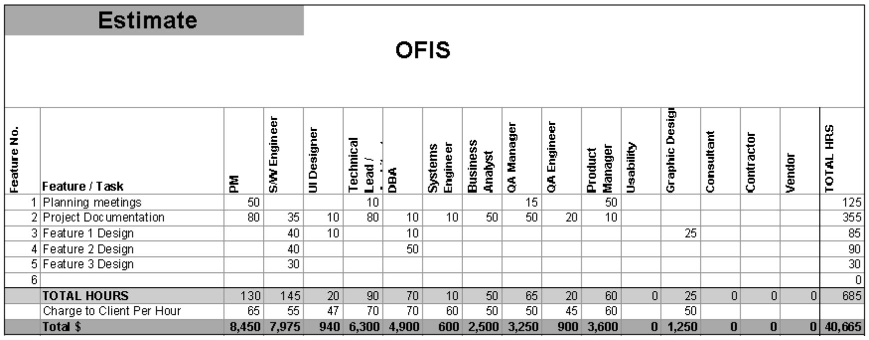
- Approximate development time (hours or weeks) for project and cost per hour per developer
- Management team required and costs

- Approximate size of project team required
- Approximate cost for hardware, software, development and testing equipment, and tools
- Approximate quality assurance testing time (hours or weeks) and cost per hour
- Support staff required (client support, technical support, operations team) and cost per hour
- Consulting or contractor time required and cost per hour
- Time needed for research, requirements, and design including costs for resources
- Deployment, delivery, and manufacturing costs
- Advertising and marketing costs
- Time and costs to produce estimates (initial, high-level, and detailed)
- Project kick-off and project completion parties!!

As a new project manager, you may have no idea where to start when trying to estimate these costs. With experience, you will find it becomes easier and easier to estimate the cost of a project. In fact, you can get to the point where you can do the calculations in your head and come up with a ballpark figure of say, $10,000 to $15,000 or $400,000 to $500,000. The closer you get to the project being approved, the more accurate the estimates need to be. You will continue refining the initial numbers as more information becomes available and as the project scope and definition become more developed.

If you have an experienced technical lead or senior engineer available to you, he or she will be able to furnish most of the above information. Your senior technical staff has in-depth knowledge of the technology, architecture, and design of the products as well as a great understanding of the skill sets of the other team members. Do not underestimate how useful they can be to you when you are trying to put together initial estimates for projects. They will often be hesitant to give you information if you have limited requirements (they are engineers, after all, and they like to have lots of details), but by presenting your questions in the right way (as hypothetical problems, for instance), you will be amazed at how much valuable information can be gleaned. Remember that no developer ever gets to spend 100% of his time on development tasks. Taking the estimate and adding a 20% buffer to allow for meetings, unit testing, technical issues, setting up equipment, support for other team members, design and code reviews, and so on, and ensuring that time has been allocated for writing requirements, specifications, and designs should give you a pretty good starting point for initial estimates. A sample

Estimate document is shown in Figure 3.2. An estimate template document is included on the CD-ROM that accompanies this book.



Estimate

OFIS

Feature No.	Feature / Task	PM	S/W Engineer	UI Designer	Technical Lead /	DBA	Systems Engineer	Business Analyst	QA Manager	QA Engineer	Product Manager	Usability	Graphic Desig	Consultant	Contractor	Vendor	TOTAL HRS
1	Planning meetings	50			10				15		50						125
2	Project Documentation	80	35	10	80	10	10	50	50	20	10						355
3	Feature 1 Design		40	10		10							25				85
4	Feature 2 Design		40			50											90
5	Feature 3 Design		30														30
6																	0
	TOTAL HOURS	130	145	20	90	70	10	50	65	20	60	0	25	0	0	0	685
	Charge to Client Per Hour	65	55	47	70	70	60	50	50	45	60		50				
	Total $	**8,450**	**7,975**	**940**	**6,300**	**4,900**	**600**	**2,500**	**3,250**	**900**	**3,600**	**0**	**1,250**	**0**	**0**	**0**	**40,665**

Figure 3.2: Estimate.

Business managers often create initial estimates. Sometimes they are realistic, and sometimes they are not. Do not panic if you see an initial estimate that you know is way off the mark. There will be opportunities to refine estimates before and after the project is formally proposed. It is possible that you will be asked to do a sanity check on an initial estimate. You may think that what you see is a complete joke and wonder how someone could seriously believe that they could develop the project with such a low budget. In this situation, it is not a good idea to be completely honest about your feelings. Remember to be tactful and respectful while being honest about the facts. It would be great if you could keep your job until the project is actually approved!!

Concept Approval

A concept is presented either in written form or verbally. Regardless of the method of presentation, it must contain sufficient information for the decision makers (typically, senior management and executives) to be able to determine whether the concept is viable and worth pursuing. The project concept will either be approved to move onto the proposal stage, declined, or postponed pending additional information being presented to support the request.

The Project Proposal Process

At this point, you will need to enter the wacky world of the three-letter acronym (TLA). You will find a whole host of TLA's in general use in the technology industry and a boatload more that are unique to individual companies. No matter where you work, you are going to have to learn a certain amount of techspeak (if you have not done so already). If you are working with outside companies, either as the client or as the vendor, you will experience the unique acronyms and technical (and sometimes not so technical) terms that they commonly use. Initially you will wonder what on earth everyone is talking about, but you will gradually begin to understand this new language. Then one day, you will hear yourself talking and realize that you have finally become proficient in the TLA dialect of the company for which you are working. Don't be fooled into believing that everyone understands what all the TLA's mean. Some people are too embarrassed to ask so they just pretend that they understand. Don't be embarrassed about asking the meaning of terms used. If someone uses an acronym or a term you are not familiar with, ask them what it stands for or what it means. You may see a look of relief on more than one other face in the room when you get the answer. Some people just make up their own acronyms, and often they are the only person who understands its meaning. It is a rather amusing trick that engineers like to play on management. They make up a new TLA to confuse the other team members, and then they wait to see how long it takes until someone asks what it means. Who said that engineers do not have a sense of humor?

Companies who are looking either to outsource a project or to procure software or hardware from an outside vendor generally use Request for Information (RFI), Request for Proposal (RFP), and Request for Quotation (RFQ) documents. It is possible, though not very common, that some companies will also use these documents for internal projects.

As a project manager, you may never write an RFQ, RFI, or RFP. The chances are that at some point in your career you will be involved in some capacity in responding to, or supplying information to someone else in your company who is responding to, an RFI or an RFP. For internal projects, the Project Proposal will be written based on the Project Concept; therefore, an RFI, RFP, and RFQ may not be required at all.

Even if you never have to write any of these documents, it is important that you understand what they are and the importance that they have to the initiation and approval of projects.

Request for Quote (RFQ)

An RFQ is used for requesting an exact quote for a specific service or an item of software or hardware.

RFQs are commonly used for ordering development equipment and tools. In smaller companies, or those that are decentralized, you may be required to complete and submit the RFQs yourself. Alternatively, your company may have a procurement department or an administrative assistant who will take care of requesting quotes on your behalf. In either case, you will need to supply the necessary information to whoever is submitting the request to ensure that the quote you receive is to the correct specification.

An RFQ is a much more straightforward request than an RFI or RFP. It is used to request a quote for a specific standard product or service. For example, you may need to purchase 10 development systems for your project engineers for which you have a specific set of hardware and software requirements. The vendor, in this case, should easily be able to respond quickly with an exact quotation of the cost for each system. RFQs are used for getting quotations for off-the-shelf products such as consumer software products or desktop computer systems.

An RFQ template is included on the CD-ROM that accompanies this book.

Request for Information (RFI)

An RFI is used to garner information for a project prior to sending out an RFP. Some examples of the information requests that you may see in an RFI:

- Verify the assumption that the software can be designed to interface with the accounting system that your company uses
- Ascertain whether the software can be developed to run on multiple platforms
- Find out whether the company has a Unix development environment
- Find out whether their engineers have the necessary security clearance required for working on special government projects
- Discover how large the development team is
- Find out whether X component can interface directly with Y component
- Discover the cost of X part
- Ascertain the approximate time to develop X component

The information gathered from the RFI can help the submitter whittle down the vendor list to a more manageable number prior to sending out the RFP. For example, the RFI may be sent out to twenty-five companies, and the RFP may go out to only ten of those companies. Alternatively the RFI might be used to gather information such as input into a Project Concept.

The RFI is also commonly used as a sanity check for budget, timelines, and technical requirements that will enable the author to a creative a more realistic and comprehensive RFP.

An RFI template is included on the CD-ROM that accompanies this book.

Request for Proposal (RFP)

An RFP is a request sent out to a company (or companies) asking them to submit a proposal to develop a project as defined in the RFP.

The purpose of an RFP is to elicit high-quality and reasonable proposals. The proposals will be based on the information and requirements provided in the RFP so it is imperative that this document contains accurate and detailed requirements. In an ideal world, the RFP will serve as the foundation for building a successful and mutually beneficial working relationship between the client and the vendor(s). Projects can be challenging, problematic, and expensive so it is important that these early communications between the companies are an effective model for future communications.

It is very important not to underestimate the difficulty in creating a written document that expresses clearly what it is that the company is trying to achieve. Bear in mind that the person receiving the RFP may have no previous knowledge of the requester's company or products. Therefore, terminology needs to be explained and not left open to interpretation. The company sending out the RFP may not be clear on what exactly it is that they need, in which case the RFP may focus on the problem rather than on the desired solution. No matter whether the solution is predefined, the RFP must be clear, concise, and free of ambiguity. The must-haves must be clearly distinguishable from the nice-to-haves and the icing-on-the-cakes. The focus should be on what needs to be achieved rather than how it needs to be achieved. If the *how* is critical to the project, then it should be clearly stated. For example, if some or all areas of the project require that engineers have security clearance, if the solution must integrate seamlessly with existing systems, or if the solution must be written in a specific programming language. If a how is not critical to the success of the project but is desirable, then it should be listed as a nice-to-have, leaving the door open for the vendor to propose alternative solutions.

The information contained in an RFP will vary depending on the type of project. In addition to a detailed description of the company, the project, the market space, the end user, the project goal, detailed project requirements, and the budget range, it should also contain a high level schedule of events. For example:

- March 1st—Request for Information (RFI)'s sent out
- March 12th—RFI Information due
- March 15th—Short list of selected vendors
- March 23rd—Request for Proposal (RFP)'s sent to selected vendors
- April 20th—RFP Proposals due
- April 23rd—Project Contract awarded to selected vendor
- April 24th—Purchase order for project issued
- April 26th—Project Definition (Charter) meeting

- May 21st—Marketing Requirements Document due
- May 24th—Kick-off Meeting
- May 27th—Project Plan due
- June 4th—Development begins
- September 6th—Phase One complete
- October 4th—Phase Two complete
- December 15th—Phase Three complete
- January 20th—Project end date

The RFP will contain information about the project concept and will ask for specific information that will constitute the proposal. The information requested will differ depending on the type and size of the project. The vendors must respond to the RFP with a Project Proposal by the specified deadline to be considered for the project.

The most important thing to remember about RFPs is that they need to be realistic, as do the proposals submitted in response to them. You cannot expect to get a $100 million project developed for less than $500,000. Similarly, you cannot submit a proposal for $100 million for a project with a budget of only $500,000. Requirements and proposals need to be in line with the budget range. If your requirements are unrealistic, the vendors who receive the RFPs will not respond. This is why it is important to use an RFI to confirm and verify assumptions, facts, and figures before creating and sending out an RFP to multiple vendors.

Sometimes companies will pay vendors to write proposals for large projects. It can take a significant amount of time to respond to an RFP, and if many companies are bidding on the same project, the chances of winning the contract may be rather slim. The cost of responding to an RFP for a large project can be high. Hours or weeks of work may be required to complete it. These factors can discourage some vendors from submitting proposals, especially if they believe that the company is only using the proposals for comparison and already has a vendor of choice. To get around this problem, some companies will offer to pay a fee to help cover the costs of preparing the proposal.

The RFP will include, but is not limited to:

- An executive summary of the client's company and market space
- A description of the problem and proposed solution
- Key business and technical requirements
- Proposed project phases and milestones
- Quality assurance requirements
- Budget
- Schedule of events
- Proposal template

An RFP template is included on the CD-ROM that accompanies this book.

What do you do if you are asked to respond to an RFP? Imagine that you receive an RFP from a client and are unsure why a particular requirement is necessary. The requirement does not make sense to you, which leads you to the assumption that perhaps the person who wrote the RFP did not really know what they were asking for. The company who sent you the RFP may not take inquiries from vendors, especially if they sent it out to many companies. If the RFP is not clear and you believe there is a better solution, then offer it as a second alternative in the proposal that you submit. Do not automatically assume that the client is wrong and they do not need what they asked for. They may not have explained very clearly why a specific requirement is important, but that does not mean that you can ignore it. Respond to the requirements that they give and offer alternatives where it seems appropriate. This way you are covering all your bases. If you fail to respond to the specific requirements, they may put your proposal aside, assuming that you are unable to meet the requirements. They are not likely to contact you to ask why you proposed a different solution if other vendors are able to offer solutions that met the specified requirements. You need to get the proposal right the first time because it may be the only opportunity you will get. A good-faith attempt to offer alternative solutions, especially ones that will save the client money, will be appreciated even if the alternatives turn out to not meet the requirements. The client will feel that you have their best interests at heart and that you are focused on the most cost-effective solution rather than on making as much money as you can. You are demonstrating integrity and a commitment to the potential client to read between the lines and to make the best possible recommendation that you can.

The Project Proposal Document

The Proposal Document explains the problem together with a description of how the problem will be solved and includes a high-level estimate of what it will cost to develop and deploy the solution. For an external project, the proposal planning will begin, for the development company, after they receive a *Request for Information (RFI)* or a *Request for Proposal (RFP)* from a prospective client. The client company will have completed the project concept and will have been through an internal approval process before sending out the RFI or RFP documents to prospective vendors. For an internal project the *Project Proposal* will be based on the approved Project Concept.

The *Project Proposal* document will include a high-level description of the project and the proposed solution. The proposal will be submitted to senior management (or to the client) for greenlighting to move on to the Project Definition stage. A proposal can be a very detailed document, or it can be a very high-level, conceptual document with a lot of open issues that will need to be addressed during the Definition phase. The accuracy of the estimates and costs are contingent upon the level of detail contained in the proposal. An external project will require a much more detailed proposal than an internal project. The client will be unable to approve the project to move ahead into the Definition phase without a pretty good idea of the time, cost, and effort involved in the project. If you are asked to submit estimates for an RFP document, be sure to document the accuracy of your estimates and any assumptions used to come up with the estimates. If a disclaimer is necessary, then include it. For instance, if you are not sure how you will be able to solve a specific problem due to the immaturity of the technology being proposed, then clearly state this on the proposal. If the client is not told that no one else has solved this problem yet, they may go into the project with unrealistic expectations! Don't try to tell them what they want to hear just to get the contract. It will come back to haunt you. Remember that the key to keeping your clients happy is to under-promise and over-deliver.

The following information will typically be included in a *Project Proposal* document:

- Executive summary
- Corporate and cultural information
- Previous projects and clients
- Client references
- Development methodologies and processes
- Quality assurance and testing processes and procedures
- Development environment
- Assumptions
- The Problem
- Proposed solution(s)
- Constraints, limitations, and risks
- Proposed project phases
- Milestones and deliverables
- The proposed project team
- Costs and payment details
- Terms and conditions
- Proposal submission and questions
- Proposal acceptance and approval

Executive Summary

For external projects where proposals are submitted in response to RFPs, an executive summary is very important. If the company requesting the proposal has never worked with your organization before, they will need some compelling data that compares you favorably with your competitors. The executive summary should include a brief history of the company together with some high-level financial information. The executive summary is a key input into the decision making process because many of the proposals are likely to be similar in solution and costs. If a company is investing a substantial amount of money in a project, they need to feel confident that they are working with a high-quality, reputable company.

Corporate and Cultural Information

This section will contain additional information about your company and the technical department(s) that will be involved in the project. It should include a corporate overview and some information about the company culture. For example, it could include operating values or mission statements for the engineering department. Explain whether the company has a casual, informal culture or a disciplined, formal one. This information could be very important to the client's selection of a vendor and will ensure that there are no misconceptions should the project be approved and awarded to your company.

Previous Projects and Clients

This section will contain descriptions of previous clients and projects that were similar to the project the client is proposing or that will demonstrate your organization's expertise and experience in project management and development.

For reasons of confidentiality, you may not be able to cite the names of some of your clients or too much detail about the projects you implemented for them. High-level information with a description of the market space that the client operates in, together with a synopsis of the success of the project(s), will be sufficient.

Client References

Here you should list some of your clients who have agreed to give references regarding projects you have implemented for them. Include contact names and titles as well as a brief description of the project(s) you worked on for each client.

Development Methodologies and Processes

This section will include an overview of the project Life Cycle and the documentation that is produced during each phase. It should also include information about processes and procedures for such things as configuration management, quality management, source control, security, and any other standard processes used during the development of projects.

Quality Assurance and Testing Processes and Procedures

Quality assurance processes, procedures, and methodologies should be outlined together with an overview of the documentation produced at each step of the Life Cycle. There may be testing outside of the quality assurance team that also needs to be accomplished, such as performance testing, security testing, or product verification and acceptance testing.

Development Environment

This section includes a brief description of the development environment. This will include details about the hardware platform for development, including operating system, design packages, programming software, unit test software, and so on.

Assumptions

This section should contain both organizational assumptions and technical assumptions. Organizational assumptions would include such things as the expectations the vendor has from the client as far as involvement in the project; client responsibilities, such as testing or documentation; and client representatives for the project (project manager, for example). Technical assumptions will include items such as the client's existing hardware, systems, and software that will be used for the project; specific technology that will be used in the development of the product; and consultants who will be engaged for specific areas of development requiring specialized knowledge or skill sets. A lot of assumptions will need to be made to create a Project Proposal document. It is important to document assumptions so that there are no misunderstandings later on.

The Problem

This section includes a brief description of the problem. This can be taken directly from the *RFP* or can be elicited from the client via phone conversations or meetings.

Proposed Solution(s)

This section contains details about the proposed solution or solutions. There may be more than one way to solve the specified problem and, if this is the case, give the client some alternatives. Different solutions should not be presented if they are just variations on the same theme. For example, do not present the same proposed solution with just one or two details changed. These specific options can be discussed and finalized later if the client decides to go ahead with the project. If you present 350 possible solutions, you will overwhelm the client with too much information, making it impossible for them to make a decision.

Constraints, Limitations, and Risks

Note any constraints, limitations, or risks that are currently known. This can include things such as the timeline for the starting the project. You may need to complete another project before you will be available to work on this one. Any limitations that your company has with regards to technology, knowledge, and skill sets should be identified together with a proposal for how you would manage them should you be awarded the project. Risks are things that should be obvious to you based on the information that the client has already furnished. It is important to ensure that you and the client are on the same page with what you consider to be constraints, limitations, or risks for the project.

Proposed Project Phases

The project may need to be developed and delivered in phases either because the client has requested it in the RFP or because you feel that the size or complexity of the project warrants it. For example, the project may consist of three distinct products that need to be developed consecutively because of dependencies among them. You may propose that the project be delivered in three phases, with one of the products being delivered in each.

Milestones and Deliverables

High-level project milestones and deliverables are detailed here. These would include the project phases, if relevant. At this early stage of the planning phase, the milestones may be confined to approximate dates for completion of specific documents for the planning phase and a high-level timeline for when each of the subsequent Life Cycle phases will be concluded. It should be obvious from the RFP and the solution section whether there are other key milestones or deliverables that should be specified.

The Proposed Project Team

The proposed project team will include the specific roles and positions needed together with some high-level qualifications and skill sets. Individuals do not need to be identified at this point. If it is known who the sponsor and the project manager will be, then name them. If you are not sure, you can specify that you have three project managers, all of whom have at least six years project management experience, and that one of those managers will be assigned as soon as the project contract is awarded and timelines are finalized.

Costs and Payment Details

This section will contain high-level estimates based on the initial estimates that were created during the concept phase or the budget information contained in the RFP.

Payment details will contain the payment terms. For example, payment for creating the Project Definition may be required prior to the project kick-off. Payment for development may be in installments. Some companies ask for 50% up front and 50% on completion of project.

Terms and Conditions

This will include some legal jargon that will most likely be supplied by the legal department. It will contain some disclaimers related the accuracy of the information contained herein, together with contractual requirements for the approval to move forward with the project.

Proposal Submission and Questions

This section will include the contact information of the proposal preparer, the date and time that the proposal was submitted, and the method of delivery. It will also include any questions the preparer has about anything in the RFP that was not clear. There may be some updates needed to the proposal before it can be finalized and approved if there are open issues or questions at the time the proposal is submitted.

Proposal Acceptance and Approval

By signing this document, everyone involved is in agreement that the proposal document is accurate and complete and is approved to move forward.

The client (or requester) will typically have a standard process for awarding the project contract to the chosen vendor.

Greenlighting

Greenlighting simply means that the project has been approved to continue. If the project is external, the greenlight will be given by the client. If it is internal, it will be given by a senior manager in your company. There is usually a standard process for greenlighting projects. In some companies, the CEO has to approve every project. In other companies the CIO has an annual budget and can spend it on whichever projects she sees fit. Product managers may be responsible for greenlighting in some organizations. The process is typically a little different for high-dollar projects. They will usually require approval from a more senior executive than the lower-cost projects. As a member of the management team, you may get to vote on which proposed projects you would like to see approved. The process is different in every company.

It is not unusual for there to be limited project management involvement in the initial planning stages of a project. Once the project has been greenlighted, however, the level of involvement is likely to increase significantly. The project may have gone through the concept and proposal stages, and the initial estimates may have been created, without any representation from project management or the development team. If this is the case, you will have quite a bit of catching up to do when the project is greenlighted and you are assigned as the project manager.

The Project Proposal may have been written in response to an RFP from an outside client or may have been written by someone working for your company who is proposing an internal project. The proposal may be as vague as a few ideas and a ballpark dollar figure that needs a lot of work to become a project description, or it may be a 50-page document that precisely details every aspect of the project and does not leave much to the imagination! Whatever the case may be, the project was greenlighted based on the information contained in the proposal. If you were not involved in its creation, it would be wise to get as familiar with it as possible without hesitation. Ask questions. Make sure that you understand the logic behind the problem, the description, the solution, and the initial estimates. If the project is for an outside client, the proposal may be seen as a template for a contractual agreement between the companies, so any discrepancies should be bought to the attention of the project sponsor as soon as possible.

It is important that you understand how projects are approved and greenlighted in your company. The more efficient the greenlighting process, the less likely that projects will be canceled part way through due to a change of heart, or change of direction, by senior executives. The more senior the managers that approve the projects, the less likely that someone will pull rank and cancel a project that is in progress. This can happen when someone on the management team "discovers" that projects are being worked on that do not fit within the company's business or strategic plan. If all projects go through a strict process for approval, this is less likely to happen. Many companies require that a Project Proposal clearly state how the project will move the company closer to its business and financial goals. A clear definition is required, specifying which company strategic objective the project is meeting and exactly how it is going to achieve it. It may seem at first sight that applying this kind of structured discipline to a creative process will result in inflexibility and the inability to get any projects off the ground. In fact, in the long term, more often than not, it results in the company being much more successful due to the fact that every department within the company is working towards the same company goals and objectives. This unified approach to strategic and tactical planning is the key to business success in today's competitive marketplace. However, if the company principals and employees are unable to make decisions, or are unable to stick to decisions once they are made, then things can get chaotic and unstable really quickly! Be aware that this will be the case whether you have a strict process or not. It is not the process that cannot make the decision: It is the people! Lack of effective decision-making is spreading like a virus in corporate America today. It is not the actual decisions that people struggle with; it is the reluctance of managers to take responsibility for a decision, so they refuse to make it. This is the reason why you must clearly identify and document all the decision-makers for your project during the planning phase!

The Different Kinds of Projects

Not all projects are created equal. There are large projects and small projects, complex projects and simple projects, short-term projects and long-term projects, top secret projects and open source projects, and on and on. There are many different perspectives on which projects belong in which categories. If a company usually works on projects with a duration of three to six months, they would consider a year-long project to be long term. An organization like NASA would likely consider a year-long project to be short term. A long-term project for them may be 20 or 30 years! Complexity is also subjective. What one company would consider complex another may consider simple. For these reasons, we are not going to try to specify what these terms mean. Each company will have their own definition for these terms and their meaning within their organization. If you are starting a new job as a project manager, you will need to ask for clarification on the terminology used within the organization. That way you can be sure that you are using the correct terminology and categorization for projects.

There are some categorizations of projects that are common across companies, such as internal and external projects, desktop and Internet projects, and software and hardware projects. These are explained in more detail in the following sections.

Internal versus External

An *internal project* is a project where the client is a person or department inside your organization. An *external project* is a project where the client is a person or company outside your organization. When we refer to the "client," we mean the person, department, or organization that is funding the development of the project. The end user of a product is not necessarily the project client or the customer.

Some companies almost exclusively develop internal projects. Technology, Internet, and software companies such as Microsoft, eBay, and Dell are primarily developing internal projects, where they are their own clients, and the end user is the general public. If your project is internal, it is important to remember that you do have a client and that you have the same responsibilities to that client as you would if you were working on an external project.

Let's assume that the company owns and develops an Internet product or service with product releases on a six-month project cycle. The client could be a product management group, a business development group or perhaps even the CEO. The client will provide requirements and will be the person or group who signs off on the deliverables. It can be trickier keeping the floodgates closed to midproject requests for additional features and enhancements when working on internal projects. If you do not have a contractual agreement, it can be harder to say "no" and to justify why you are unable to fulfill a request. Remember that a project manager has the same responsibility to her team members not to work them into the ground as she would if the client was an outside company. Project managers must not allow themselves to be bullied into taking on more work than the team can handle due to pressure from senior management. If there is a release due in six months, then there needs to be a project plan and schedule that can be accomplished in a six-month time frame. Believe us when we tell you that no project manager will enjoy working on a never-ending project. It can easily happen if the same discipline is not applied to internal projects as to external projects.

It is inevitable that if you keep adding more and more features and pushing out the delivery date, before you know it, your six-month project Life Cycle is going to be in its fifteenth month with no end in sight! If you see this trend starting to occur within your company and on your project, then you should meet with the project team and the sponsor to reevaluate the goals of the project. If the expected outcome of the project keeps changing, then you may be unable to deliver on it effectively. A disciplined approach to project management is a necessity for each and every project for them to be delivered on time, within budget, and with high quality. Having some flexibility with the completion date can also work to your advantage. If you do not have a legal responsibility to complete the project on time, both the internal client and the project team have the option of asking for more time to add or enhance features if it is going to add significant value to the project.

The quality of your product or service is a major success factor in both internal and external projects. The project manager of an internal project has the same responsibility to both define and achieve quality in the product as the project manager for an external project. Accountability may not be as formalized for an internal project. However, maintaining a high level of professionalism by ensuring that you are developing and delivering a high-quality project is of the utmost importance.

Another issue that can occur with internal projects is the lack of a standard budgeting process. If you do not know how much money is in your budget, how do you know whether the projects you are managing are cost efficient? Your internal client may assume that since you are not tied to a budget, you can just keep adding things to the schedule as you go along with little or no impact on the project outcome. This is not the case. The resources required for projects are not limited to money. People and time are also limiting factors. You may have the funds to pay for more people, but it is not always possible to divide a two-week task for one engineer into two one-week tasks for two engineers. There is a limit to how many people can be working on the same piece of code at the same time. This looseness around budgets can work to your advantage, however. If you want to use new development tools or environments, it will be easier to get funding than if an outside client were being asked to foot the bill. You also may have more flexibility in your estimates. If you have underestimated costs by 40%, it may not be a huge problem if the costs were not tied to a strict budget in the first place. Regardless of these advantages, we would advise that you try to stick to a standardized process for your projects as much as possible. It is a good discipline to learn early on, and it will serve you well in the future regardless of whether you are working on internal or external projects. Learning sloppy project management skills is not going to help you in the long term with your personal career development. It is very difficult to unlearn bad habits!

External projects have their own set of unique problems and advantages too. Working with outside clients where you are restricted by very specific contractual agreements can reduce the flexibility that you have in how you implement and manage your project. Your accountability is going to be to a wider group of people and departments. There will be a more formalized process for reporting the status of your project, and it is likely that you will be expected to follow a standard reporting process. You may find that you have less autonomy managing an external project than you would have managing an internal one. Your milestones and deliverables will have high visibility. Client interaction can be time consuming. The client's company likely uses some different terminology than your company does, so miscommunication and misunderstandings are common pitfalls. Every detail of the project should be documented, including the definitions for terms used. The client may be very demanding and not sensitive to the fact that you have other clients and other responsibilities.

The benefits of working with external clients are that you must have a clearly defined and documented budget for your project. The client has a project completion deadline and a contractual agreement detailing exactly what is included in the project. Changes must go through formal change control because it is the only way to ensure that costs are kept within budget. Though the client might be a pain in the butt to work with, you always have your "contract" as ammunition to back up your reasons for denying a request or referring it to change control. Your relationship with a client is very different from your relationship with someone that works for the same company as you. The relationship is much more formal with an outside client, and communication will be subject to specific standards and guidelines as defined by your company. It is good practice to ensure that you keep your client relationships as formal and professional as possible. It will help you avoid situations where the client employs the use of emotional blackmail tactics when requesting additions and changes. By "emotional blackmail," we mean using personal friendships or favors as collateral. It is hard enough when you have to deal with this internally; do not fall into the trap of letting it happen to you with external clients, too!

If your company works primarily on external projects, they are very likely to have some internal projects too. For example, they must have business or desktop systems that employees use for development and administrative purposes. They may have a company Web site and an Intranet site. Some companies develop their own proprietary tools and applications for use by their employees. In addition, system and software upgrades are managed by internal support groups such as IT, IS, or desktop service teams. If you are working on these internal projects, ensure that you are using project methodologies the same as you would if you were working on a product development project. All projects must be managed effectively if they are to be successful. It is easier to hide inefficiencies when working on projects for internal groups that do not affect a product or service that is destined for sale to an outside client. Do not allow yourself to fall into the trap of embarking on projects with no real plan, with unrealistic timelines, or without an approved budget. It is your responsibility to set the standard for how your projects will be set up and managed. Set the bar as high as you can, and expect others to follow your lead. Lack of accountability is a poor excuse for executing substandard projects.

Desktop versus Internet

There are quite a few differences between managing desktop software and Internet projects. Internet projects have added complexity due to the number of different operating system (OS) and browser combinations that need to be supported. In addition, there is significant system and load-testing required to support a Web product. This additional coding and testing needs to be taken into account at the project planning stage to be sure that you have allocated sufficient time for the deployment tasks.

One of the biggest problems with developing for the Internet is coding the graphical user interface (GUI) to look and feel good and to be usable on different platforms and browsers. The same code will display very differently on a PC than on a MAC and differently in Internet Explorer and Netscape browsers. There are differences between the operating systems on the same hardware platform and differences between different versions of the same browser. Then there is the issue of screen resolution, affecting how the site displays and scrolls. Decisions have to be made concerning which platform-OS-browser-connection speed combination the site should be optimized for use on. It is almost impossible to optimize the site for everyone. These issues create a huge amount of work for developers because they have to develop and unit test their code using many different system combinations. Quality assurance teams also have their work cut out testing the numerous combinations. Connection speeds can also be a problem. If a user is on a slow connection, they may time out on some CPU-intensive screens if the performance is too slow, which can make the product unusable for them. It is possible to get around these problems by supporting a very limited number of OS-browser-connection speed combinations, but this can sometimes backfire. If the majority of your prospective end users are using an unsupported system configuration, you risk losing a lot of business. The early adopters, who are generally also Internet users, are quick to upgrade to the latest OS and browser versions. As a result, if you release your product one month before a new OS or browser is released, you may find that your early adopter users are no longer able to use your product if the new browser is not supported. It could also be that the majority of your users are on slow connections. If this is the case, your site needs to be designed so that it will work for those users. Unless you specifically lock out users who are on unsupported systems from a Web site, you may risk bad publicity as well as losing customers if users are able to connect to the site but then find it unusable.

Load and stability tests are critical for Internet products and services. Desktop products are generally designed to run on one computer with one user at a time. Testing the performance of a system under those conditions is pretty straightforward. The performance of the product will be dependant upon the computer system that the user has the product installed on. With Internet products, this is true to some extent, but there are many other bottlenecks that can be encountered over which the user has no control. For instance, network problems either with the user's ISP or at the facility where the site is hosted, a high number of users on the site, and reduced Web site capacity due to technical problems and system crashes can all degrade performance. Load testing is an art, and if the application is complex it can be very hard to get the user models right for running the load tests. You need to know what percentage of your users are on which browsers, OS's, and connection speeds. If you have a large number of users on broadband connections, they will be moving through your site at a faster rate, which will increase load on the system. On the other hand, if they are getting through the site faster, the chances are that they will close their session sooner, freeing up bandwidth for more users. Therefore, with a large number of broadband users you may be able to support less concurrent users but more users per hour or per day. The more you get into the details of load testing, the more mind boggling it gets. Suffice to say, if you have an Internet product, make sure that you have adequate time for testing and that your load testers are experienced and knowledgeable in the tools and techniques for effective load testing.

If an end user is playing around with the product and either intentionally or accidentally crashes the system on a desktop product, they are going to crash their own computer, and it will not affect anyone else. If they are using an Internet product and manage to crash the application, however, it is possible that the server the application is running on will also crash. There is a potential that hundreds or thousands of other users could be impacted by a situation like this. The Internet application needs to be designed to handle these types of problems in a graceful way with minimum impact on other users. Security for Internet products is another critical area that needs adequate attention and a lot of specialized testing. A denial of service (DoS) attack could bring down your whole system and perhaps even your business systems if they are running on the same network. None of these are issues for desktop products. There is a lot more to developing products and services for the Internet than just writing a few lines of HTML. Your product plan should reflect this.

Remember that it is much easier to track down the cause of a critical problem before your Web site is live than afterwards, so check and double check your test plans for both functional and performance testing before you sign off on the project plan!

Desktop products have their own unique challenges. One of the beauties of Internet development is the ability to fix problems with the product for every user with one update to one product (the Web site). With desktop products, once the CDs are burned and the customer has purchased one and installed it on his system, the product is going to stay exactly the way it was when it left the manufacturer unless there is a way for the user to upgrade the product later. Each fix requires that every user update his product individually. Most companies offer upgrades and bug-fix releases for desktop products via the Internet. The biggest problem with this is that there is no way of ensuring that all users download the upgrades. Many of them will not. It also means that, even for desktop developers, there is some Internet programming required in addition to some testing of the Internet systems used as the download site(s).

Programming code to run on different hardware and operating systems can also be challenging. Different versions of the product will need to be developed for different platforms. Backwards compatibility is also an issue. If a user upgrades to a new version of the product, there is a possibility that her files from the old version may not work with the new one. If a user upgrades from Windows 98 to Windows XP, will the product still work? Should the product work with the new OS, or will the user be asked to download an upgrade or purchase a new product?

Your engineering team may be developing on Windows XP machines with 2 gigabytes of RAM, but most of your end users may be on Windows 98 machines with 256 megabytes of RAM. Do you have sufficient test systems for your developers to ensure that their code runs optimally on the older OS and much less RAM? There are many intricacies involved in developing a product that go beyond making sure that the product runs on the system that the engineers are using for development. Your product plan must take all these things into consideration.

Software versus Hardware

Many projects contain a combination of both software and hardware, but they will likely be focused primarily on one or the other. For instance, a hardware project may use software to support the hardware development and vice versa.

Examples of software projects include desktop programs (one user), enterprise software (multiple users on a network), Internet products and services, telecommunications, and databases. Examples of hardware projects include desktop and laptop computers, servers, networking, backup systems (tape drives), storage systems (flatfile and databases), consumer electronics and telecommunications, and aerospace.

Most of the examples we use in this book relate to software projects. There are differences between managing software and hardware projects. However, the similarities far outweigh the differences. The following table shows a basic comparison of the Life Cycle phases for software and hardware projects.

Table 3.1: Hardware and Software Life Cycle Phases

Life Cycle Phase	Software	Hardware
Planning	Planning can be a long and drawn out process. It can be challenging defining the problem to be solved. There are often multiple options on how to solve a problem. It can be difficult defining the scope.	Planning generally can be completed in a much shorter time frame. The problem is more tangible and less elusive. There are limited options on how to solve a problem. Scope is often implied by the nature of the project. It is not open to debate or interpretation as much as it can be in software development.
Design	There can be any number of ways to solve one problem. Ten different engineers or designers will likely come up with 10 different designs based on the same requirements. The project team creates the software and system architecture. They also create a product design along with individual component or feature designs. This phase is much more complex in a software project than a hardware one. If a software project is going to be canceled, it is most likely to be at this stage when the true costs for the project are identified. As this phase ends, the very costly Development phase of the project begins.	Due to the use of standard components in hardware, there are usually only one or two ways to solve a specific problem. Ten different engineers or designers will likely come up with 1 or, at the most, 2 designs based on the same requirements. Different components may be designed by different teams or different companies. For instance, Power Supply Units (PSU)'s, Application-Specific Integrated Circuit (ASIC), Input/Output (I/O) Cards, and so on. For servers, the unit cases have to be a standard size to fit into standard racks. The standard rack sizes are used globally, and all rack-mounted electronic and computer equipment is built to standard rack size specifications. In the Design phase the team will produce blueprints for the circuit designs, and technical writers will create precise specifications for the fabricator.

Table 3.1: Hardware and Software Life Cycle Phases

Life Cycle Phase	Software	Hardware
Development	This is one of the most expensive phases of the project and takes a large percentage of the overall project time. In this phase the team uses the designs to create the technical specifications and to build the components and features that constitute the product. The features and components are tested with each build of the product. The entire product does not have to be completed for testing to begin. However, unlike hardware projects, software projects do not use standard components so there are no preexisting specifications. Each software component has to be tested individually as well as together. The complexity of this phase is much higher in software than in hardware projects.	This is a relatively inexpensive phase of the project. It takes up a small percentage of the overall project time. This phase is where the fabricator produces a working prototype from the circuit designs.
Integration (and testing)	This is the phase where the different components of the software are integrated and tested. The full QA test process is implemented and, for Internet products, stability and performance tests are run. The risk factor in this stage is fairly minimal. The product components will have been tested at regular intervals during the development of the product so the integration testing should identify integration-related bugs only. Issues that arise during testing at this stage are generally solvable. Usability and beta testing will be scheduled to occur during this phase.	This phase and the following one are the two most costly phases of a hardware project. In this phase the prototype is tested and refined to produce a blueprint for the final product. The testing for hardware can be more complex than for software projects. For instance, reliability testing, EMF testing, stability testing, and environmental testing are completed. Some of these tests are regulatory and need to follow a clearly defined process. The physical product needs to be tested for durability as well as the functional product testing. For instance, what happens if the server is dropped or if it gets too hot? If a hardware project is going to be canceled due to issues, this is the phase where it will often happen—right before the costs start to escalate at a rapid rate.

Table 3.1: Hardware and Software Life Cycle Phases

Life Cycle Phase	Software	Hardware
Deployment	This is where the final testing is done on the product. For desktop software, the master is sent to the CD manufacturer and the artwork is sent to the packaging manufacturer to produce the final product. Distribution deals are finalized. The major expense here is producing the master. If a problem is detected after the CDs have been burnt, the master can be recut and the CDs rerun. This is an additional cost but not one that is likely to lead to the project being canceled. Training for technical support and customer support groups occurs during this phase.	This phase is where the specifications and circuit designs are updated and sent to the manufacturer. The risk factor for this phase is very high for a hardware project. One mistake or undetected bug in the hardware can result in hundreds or thousands of defective units being manufactured at very high cost. Manufacture of spare parts for the product is also required for the actual hardware units. If the company is international, shipping and processing for spare parts can be a huge expense. Training for support (including repairs) is extensive and can be a global issue. If beta testing is required, it will happen in this phase—after the final product is released. Beta testers are usually the early purchasers of the product, and they often get discounted product rates for agreeing to be the guinea pigs for a new product line. It is not usual for them to get free use of the products for an extended period.

Table 3.1: Hardware and Software Life Cycle Phases

Life Cycle Phase	Software	Hardware
Post-Deploy ment	Post-deployment for software usually involves some scheduled bug fix releases and updates that are delivered either via the Internet (download) or by the customer buying an updated copy of the software. Product support for software is usually about three to five years. It is usual for an updated version(s) of the product to be released during that time frame and support for the older versions to be withdrawn after a few years.	This phase is much more complex and long lived for hardware. The term "sustaining engineering" is used often to describe the process and the teams involved in post manufacturer and post sales support. This support is more complex than for software because it involves getting hardware out to clients who have faulty units. It is a lot more time consuming and costly than asking a user to download a software update. Spare parts need to be available quickly along with trained personnel to install them. With global distribution this can be challenging. A customer does not want to be told that their part will arrive in 6 weeks because it has to be shipped from Germany to the USA and it "takes that long to get it." The shelf life for hardware is much longer than for software—on average about 5–10 years. Hardware is more costly to upgrade so clients usually keep the same systems for a number of years before upgrading. The sustaining engineering department is kept busy for years testing new software releases and hardware component upgrades.

Hardware projects are generally less complex than software projects due to the use of standard components. For example, a company that manufactures servers will usually purchase the disk drives, power supplies, and some circuit boards from a manufacturer who specializes in those specific components. The components are built to standard specifications and meet the compliance criteria. These components may be less complex, but mistakes can be much more costly. If you burn 400,000 CDs and find that there is a bug in the code and have to recut a master and reburn the CDs, it is going to eat into the bottom line a lot less than if you have 400,000 high-availability servers just off the production line and find a design flaw that makes them low availability and they have to be rebuilt with newly designed components. That kind of mistake could bankrupt a company overnight.

One anomaly that is interesting to note about hardware versus software is the apparent inability of these two technology disciplines to work together effectively. We are not sure whether this is due to a rivalry between these two distinct disciplines or a lack of understanding of how much each one impacts the other. We suspect the latter. It is often the case that we think whatever we personally are doing is the best, the most important, the most challenging, and so on when, in fact, we just do not understand enough about what others are doing to truly appreciate their challenges and their achievements.

In our observations, it seems that hardware developers tend to be totally hardware focused despite the fact that their products usually require the use of software to work. The same goes for software developers. They are so focused on software that they often forget to take into account the different kinds of hardware that the product needs to run on. Neither can work without the other, but there is not much evidence of integration between these two groups in today's technology companies. Some companies may have both hardware and software departments or teams, but it is often the case that they do not work together on one project team for a specific product. The hardware is kicked over the fence to the software department (or company) and vice versa. What great strides could be made in this area if the project managers working in each discipline had a greater understanding of the challenges that the other faced! We could see a lot less install and set-up time for products if they were truly designed to work together.

The Different Kinds of Stakeholders

You might be wondering what exactly we mean when we refer to a "stakeholder." If you asked 10 people that question you might get 10 different answers! It is not unusual for members of the same project team to maintain their own lists of stakeholders that bear no resemblance to the lists maintained by other team members. Therefore, when discussing stakeholders, project team members are not necessarily talking about the same set of people.

We are going to give you a general definition of "stakeholder" that is designed to be used in conjunction with an identifier that is unique to a specific set of stakeholders.

A stakeholder is a person, group, department, or company who is involved in, or has an interest in, the project or the outcome of the project. For most projects, there will be different sets of stakeholders, each of whom will be interested in different aspects of the project. To really understand stakeholders and to manage communications with them effectively, you should analyze the list of all stakeholders that belong in the general stakeholder group and assign them to unique stakeholder subgroups according to their interest in the project. Many of the stakeholders will belong to more than one stakeholder subgroup.

It is very frustrating for the stakeholders, and the project manager, if there is only one list of stakeholders for a project. This means that everyone on the list has to either opt in or opt out for all communications about the project. What invariably happens in this situation is that stakeholders are bombarded with a lot of information that they have no use for. After a while, they will consider the communications to be junk mail and will delete or ignore everything that they receive concerning the project. The information that is really important to each person then gets lost in the noise. This could be disastrous for you and for your project.

Imagine that you have a stakeholder who works in the legal department at the client's headquarters. Additionally, you have a stakeholder who is a member of the data center operations team at your company. If you are sending out everything related to the project to every stakeholder, these two people are going to receive e-mails pertaining to the development environment for the engineering team; meeting minutes for your weekly development team meetings; team lunches and outings; decisions to cut features from the project; design review notes, and so on. As you can imagine none of these topics are going to be of interest to either of those stakeholders, and these e-mails could add up to many per week, especially if some of them turn into e-mail debates where "reply to all" is used on every e-mail. (You know this can happen!) You could have 50 people on your stakeholder list, and each one of them is going to be overwhelmed with too much information about details of the project that do not affect them in any way.

Therefore, split your stakeholders into subgroups and make sure that everyone on the project team is using the same subgroups to send out information and updates. You can group your stakeholders in many different ways and, of course, each stakeholder can belong to one or many of the stakeholder subgroups. Certain pieces of information will need to be sent to all stakeholders. For instance, if the project is canceled or postponed, or if there is a major change in the project team (a change of sponsor or project manager, for example), then this is information that all the stakeholders will need to be aware of.

Here are some examples of how you might subgroup stakeholders:

- Steering Committee—The steering committee comprises the key decision makers on the project. Typically this group will include (but is not limited to) the sponsor, the project managers, the product managers, and the client.

- Internal development—Use to share information that relates to the internal development groups and quality assurance such as development environment, changes to requirements or specifications, design reviews, code reviews, and coding standards.

- Security—Use to share information about security concerns, breaches, and standards with relevant members of development, senior management, security, client, and operations teams.
- Project plan changes—Use to share information about proposed changes to the plan including additions and deletions of features. Members may include client, product management, legal, quality assurance, and project sponsor.
-]Deployment—Use to communicate all deployment related information. Members may include operations team, client, and quality assurance.

Depending on the size and type of your project and how many stakeholders there are in total, you will need to decide how to subgroup the stakeholders based on who needs to know what and when. It would be wise to meet with the project team to discuss the process for defining stakeholder groups before finalizing the list of groups. Send the list of subgroups, including which stakeholders are included in each group and a description of the type of information that will be shared with each group, to all the stakeholders. This gives them an opportunity to ask to be added or removed from any of the lists. This validation of your communication plan for stakeholder groups ensures that everyone will receive the information that they need and that they have asked for.

The Different Kinds Of Resources

Two of the main factors involved in project planning are resources and scope. If you manage your resources effectively, you will also ensure that your project stays in scope. Resources are the inputs for each task, and deliverables are the outputs. Resources can be categorized into four main areas:

- People
- Equipment
- Time
- Money (budgets)

Resources may be acquired, rented, or provided from within the organization; may come from the client; or may be contracted or purchased from outside the organization. This includes your project team, equipment and technology, and money. Time is intangible so it cannot be acquired or increased without affecting the outcome of the project in some way. Managing project resources involves more than people management; it includes the management of equipment, money, and time. The number and combination of resources needed to execute a project is determined during the Planning phase of the project Life Cycle.

Costs are a major resource and they are determined and managed through budgets. The word “budget” often elicits groans and even feelings of near panic in some people. They are not as scary as you might think. You just need to learn how to create, track, and manage them. Your project schedule is your tool for tracking people and time resources; your budget is used to track your cost resources. Equipment resources are managed to some extent in both of these processes. For example, equipment purchases are tracked in your budget, but installing and setting up the equipment is tracked in your project schedule.

All four resource elements are interrelated and need to be balanced appropriately. Each of the four factors must be managed effectively and simultaneously for the project to be successful.

People

Your people resources include your project team, other company employees (operations staff, for instance), vendor staff, consultants, and contract labor. Assembling a project team, and identifying required support resources, with appropriate skills and availability is critical to your project. Resources must be identified in the planning phase of your project. You cannot commit to a project and a deadline if you have not yet found a vendor to supply necessary services.

When identifying your people resources you need to ensure that you have not forgotten anyone. Identifying and budgeting for your core project team should be fairly straightforward. Identifying other internal resources, such as quality assurance and testing, operations, technical support, marketing, and product management may not be as simple. If you identify these people or groups as resources, then you need a commitment of specified time that will be spent working on your project. Telling your operations team that you need some support and them telling you that is OK and to just let them know when you need them is not going to be sufficient. You need to have a plan, and you need them to sign off on the commitment of time to your project. This is also true of resources being supplied by the client. For instance, they may have agreed to take on some level of quality assurance or testing responsibility for the project. You need to ensure that they have adequate resources and that they will be available when you need them. You cannot afford to be waiting around for the client to start testing if your whole team is waiting for them to finish and sign on before proceeding. Contractor, consultant, and outside vendor resources need to be planned in the same way. For all of these different resources, you need costs. You need to ensure that you will stay within budget, so all these costs need to be documented and approved before you start.

It is vitally important that everyone understand his or her own roles and responsibilities, those of everyone else on the team, internal and client resources, and the roles and responsibilities of all contractors, consultants, and vendors who will be involved to any degree in the project. This information should be documented and shared with your team. Contractual agreements need to be in place with any external resources. Some kind of written agreement also needs to exist for any internal resources that are not part of your project team. The exact way the agreements work will vary from company to company, but be sure that you have them. External agreements will usually need review by a legal team. The company implementing the project will most likely have a process that you need to follow to ensure that this is done correctly.

The use of contract labor is becoming increasingly common in technology and IT projects. Some companies maintain a small staff of experienced people to serve as team leaders and employ staffing agencies to provide contract workers for the project teams. Other companies employ the majority of their staff full-time and use contract workers for specialized work. There are advantages and disadvantages to using contractors.

Some of the advantages are that the company doesn't have any obligation to the contractor after the completion of the project. When the project is finished, the contract staff member is bid farewell and thanked for a job well done. The company is paying a set hourly rate for the contractors and is not liable for benefits, bonuses, vacation and sick pay, or unemployment claims. Contractors are usually brought in to work on one specific project so the chance of someone stealing them from you midproject to work on something else is minimal. If you need a developer with specific specialized technical or domain knowledge for just this one project, then hiring a contractor is the most efficient way to fill that role. Because contractors are being paid hourly, they are not usually averse to working overtime when necessary.

On the other hand, there are also disadvantages to using contractors. Unless you have worked with the person before, you are really not sure what you may be getting or if their skills are really as advertised. Normally, when choosing team members, you have some previous knowledge of their work ethics and team interaction from observing or working with them on previous projects. With contractors, you have minimal information about their work ethics and personality types. When companies hire a new employee there is usually a rigorous hiring process that involves multiple interviews and background checks. With contractors, you are relying on the staffing agency to have done this on your behalf. One big problem with using contractors is that once the project is finished, they take all the knowledge and experience that they gained from it with them. If their code is not well documented or their project documentation is not accurate or updated, it can take a lot of time and effort for another team member to come up to speed sufficiently to resolve any post-project issues or bugs.

Most companies will tend to use the same one or two staffing agencies to supply all their contractors. Once a good professional relationship has been established with an agency, it becomes much easier to find the kind of contractors that you need. If the agency understands the company culture and the team dynamic, they will try to find you contractors that fit well into your environment and have the appropriate technical skills that you need. If you are using contract staff, you must remember that they are joining your team as bona fide team members, and you need to treat them with the same dignity and respect as the rest of your team. You also need to hold them to the same standards as the rest of your team. They need to be fully aware of their own roles and responsibilities, those of the other team members, and additional project participants. They need to be aware of the company and the team processes and procedures and understand the goals of the project in the same way that the rest of your project team does.

Consultants may be hired for the duration of the project or for one specific part of the project. Consultants are usually experts in a specific field or fields. They may work onsite or offsite. Depending on the nature of their involvement in your project, they may act more as advisors than developers. Whatever their role, they will need to work closely with the project team and with you, the project manager, to ensure that information is flowing effectively in both directions. Consultants sometimes work for an hourly fee and sometimes for a flat fee. If they are charging an hourly fee, you will need to closely monitor this expense to ensure that you are not going over budget. Make sure that the consultants are staying on track and that they are meeting deadlines. Ensure that you have a contract in place that specifies *exactly* what you are engaging them to do. Let them know that no additional work outside of the contract can be approved without a new contract being drawn up. You do not want your consultants doing additional work that they "assume" you need and then billing you for it! You may need to keep a close eye on the time spent working on your project to ensure that they are not neglecting your project to work for other clients. If your consultants are working offsite, a weekly status meeting is essential to make sure that they are on track for their deliverables.

Managing outside vendors can also be tricky. You will need to monitor progress closely to ensure that they are on track to deliver exactly what you ordered and that it is delivered on time. Vendors often over promise and under deliver. If you have a drop-dead date for what you need from the vendor, ask for it two to four weeks earlier than that. This will give you some wiggle room if they run into problems or if they turn out to be unreliable. As with the consultants, ensure that you have a contractual agreement and that they have a schedule from you detailing the deliverables that they have agreed to.

Internal resources outside of your project team will often be required for certain aspects of your project. Your IT group, QA department, data center, or operations team may be needed to set up systems, order hardware, implement some parts of the project, conduct security reviews, test the product, and so on. The commitments to deliver or participate in these tasks should be documented together with timelines and costs. (Work completed by other departments within your company is not free; those people have to be paid, and it is likely coming out of your budget!)

Equipment

The project may require hardware or software to run on. This is especially true for Internet products. You may need servers, databases, backup and restore systems, test systems, operating systems, application server software, Web server software, database software, an so on. The list is endless! The project team will also require development hardware, software, and tools in addition to various materials and supplies that will be needed to complete their tasks. This equipment includes the facilities, computers, printers, desks, communication equipment, development and staging servers, CD burners, software, hardware, licensing for software and hardware, books. This list can be endless, too! It is the project manager's responsibility to identify all the equipment needs for implementing, testing, and releasing the final project. Some of the equipment will likely already exist and will not need to be repurchased. You should not be trying to get your company or the client to buy everything that your heart desires. The object is to make sure that you identify what you need for the project. The key to managing equipment resources, much like managing people resources, is to make sure you have the right equipment in the right place at the right time. Any necessary supplies, licenses, or operating manuals should also be provided. There is no point in having a lot of hardware sitting around if it is not possible to operate it due to lack of supporting materials or equipment. You may also need the right people resources to support the equipment resources. If they are not available to set your equipment up when it arrives, then you are going to have a scheduling problem.

Creating a desirable environment in which the project team can be effective and successful is the responsibility of the project manager. A factor that you have to consider when building your team and your development environment is whether the team will be working on their home turf, at the client site, or at another remote site. If they will be traveling or working offsite, they may need laptop computers. If they are working at the client site, is the client providing a development environment? Is the client providing this service free of charge, or do they expect you to pay for it? If your team is working at another remote location, do you need to pay rent for the facilities and equipment? You also need to consider what specialized tools your team may need to implement the project. If you have the tools at your home site but your team will be working at the client site, will they have remote access to the tools? Do they have the necessary networking and Internet access that they need to connect to the home office? These are just a few of the funding questions that you will need to answer to accurately estimate your equipment resource needs. Equipment resource planning needs to be done early in the planning phase of your project to ensure availability when you need it. If you order equipment 1 week before you need it and discover that it will take 10 weeks to receive your order, then your project is going to be in trouble before you have even started!

Management of your equipment resources is a requirement for managing your people resources. Your team members rely on the equipment to be able to implement their project tasks. If you are not effectively managing your equipment resources, then your team could come to a standstill. You may need to get more creative when managing equipment resources than with other resources. If your project depends on it, then make sure you have a plan A, a plan B, and if possible, a plan C.

Consider this example: Your project plan specifies that the development team will be working on a new development server that was ordered during the planning phase of the project and is due to be delivered two weeks before development is due to begin. This task is on the critical path but is not considered to be a high risk because it is due to be delivered early enough to allow a week for installation and setup plus an additional week buffer zone before the team needs to start work. One week before the server is due to be delivered, however, you receive a phone call from the supplier telling you that the server is going to be two weeks late. So what do you do?

One option would be to try and push the start of development back a week and reschedule the setup of the server for two weeks later than it is currently scheduled. This is not a great option because you will start your project one week late and, if there are server issues, possibly even two weeks late.

Another possibility would be to see whether you can find a spare server in-house that you can borrow for a period of time until you receive your server and have the time to take the servers down and swap them out. This is a better option, but you still have some server downtime to deal with. You are also assuming a level of risk by making a midproject development environment change.

A much better solution would be to negotiate with the supplier to see whether they can offer you a substitute server that is comparable to the one you have on order. The chances are that the supplier will have a server either one step up or one step down that will still work for you. If you take one that is one step up, ensure that you are not charged any extra for it. However, if you take one that is one step down, negotiate a lower price. If neither of these options is available, you could ask the supplier for a loan server to use until your server is delivered.

As a very last resort, it might be possible for your team to start working on the project and saving all their work locally (on their desktop or laptop systems). They will need to be very meticulous about backing up the data every day, which could be tedious, and there is some level of risk involved, but it is better than not doing anything for a week or two.

As we said, be creative and think of as many options and plans as you can and be sure to have a couple of back up plans just in case your first one doesn't work! This is a good process to follow for anything that is outside of your direct control. By this, we mean any work or service being provided from outside of your core project team. You can prioritize and escalate issues on your own team, but you are relying on someone else to do that in other groups or companies. Always have a plan B just in case there is a problem. As you gain more experience managing projects, this kind of planning will become second nature to you. You will learn not to trust 100% that something being managed by someone else will definitely be completed on time. You should hope for the best and plan for the worst!

New technologies emerge so fast these days that you may find the original system you negotiated at the beginning of your project is no longer the latest or greatest solution available. Work closely with your suppliers and build strong professional relationships with them. If you are partnering with your vendors rather than just telling them what to do, they will be proactive in letting you know about changes or opportunities that may impact or benefit your project. They will work with you to get you the best price and proactively search for alternative or better solutions. Having said all this, you do need to be careful about using new technology on projects that cannot tolerate a high level of risk. If you are looking for stability and proven performance and reliability, then don't consider technology solutions that have been on the market for less than a year. No matter how long the technology or system has been on the market, ask for contacts at other companies that have implemented the same solution or are using the same technology and be sure to follow up with them. You can learn a lot by talking to the technical teams at those companies. Your supplier may tell you that ABC Company is using the same solution and it is working just great for them. When you call ABC Company you may find that, though they are using the same technology, they are using it for a completely different purpose and are not utilizing most of the features and functionality that you need for your project.

You will need time to work with your suppliers, and some of your technical staff may also need time to research technology solutions that they are recommending. Make sure that you allow time in your project schedule to accommodate this. You should also allow extra time for equipment installation, setup, and testing. You should never assume that your equipment is going to work without a bit of tweaking and testing. If you have ever received a new computer system, you are well aware of what we mean by this. There is always a lot more setup and troubleshooting than we anticipate! Remember, developing a high level of skill in managing equipment resources won't be of much use to you unless you can also stick to the project schedule. Schedule and time management is critical for successful project management.

Time

Time is intangible, but it still needs to be managed. You may think that as long as you have enough people to complete the work, the equipment you need, and the money to pay for it, where does time fit into this equation as a resource? You are assigned a set amount of time to implement your project. If you need more of it, you have to ask for it, and you will likely need more money to pay for it. So why can't you get more people and pay for them without asking for more time? That doesn't always work. For example, imagine that you are one week from your project completion deadline. You have 20 days of work that still needs to be completed and five full-time developers available. You cannot necessarily assign four engineers to work five days each or five engineers to work four days each to complete the task. The task may need to be completed by one person, sequentially, over the course of 20 days. This is true for all of your project scheduling. Some of your tasks may be able to be split between two people, but the majority of them will need to be implemented by one person. Throwing more people resources at the problem may not resolve it. If your development server arrives one week late and you need one week to set it up and test it, you cannot ask five engineers to work on it for one day each or ask the test team to work concurrently with the setup team to save time. You still need a week (or five days) to complete the task. Time is often the forgotten resource. If it is not managed appropriately, it can run out, and once that happens, it is inevitable that your project is going to be late or is going to be delivered with some functionality missing or compromised.

Money

Once you have your required equipment and people resources identified, you need to make sure that you have the money to pay for them. Managing money as a resource is not always an easy process, especially if you don't like finances. If the thought of managing money makes you break out in a cold sweat, you need to get over your fear and do so fast. Managing money and project budgets is a huge part of a project manager's responsibilities. Managing money includes all costs associated with the project. These costs include tools, people, equipment, testing, measurement, training, employee motivation, development environments, build systems, desktop environments, production environments, and travel. Depending on your project, you may need to budget for all or some of these. Depending on how the budgeting process is structured at the company implementing the project, some of these costs may be budgeted outside of your direct project costs. For example, if you are working on an internal project for a company that develops its own products, you may have separate budgets for all, or some, of your people- and equipment-related costs. The salaries, development environment, development tools, facilities, employee motivation, training, and training-related travel may all be included in your administrative or run-the-business budget, and only the costs uniquely and directly associated with your project may be in your actual project budget. Those costs would include special tools or technology required to support the project, contractors and consultant fees, vendor costs, travel related to the project, and project equipment. Different organizational and project structures organize budgets in different ways. You still have to manage all the costs, but you may be managing them independently of each other.

Each project feature or task will have a cost associated with it, whether it is for labor hours for your team members, or the purchase of your staging server. Within larger companies, there is usually a designated person (or department) who is responsible for making the purchases on your behalf. There are pros and cons associated with this method of procurement. On the plus side, it saves you the time because you don't have to do all the legwork of working with suppliers. It also allows the company to negotiate special rates and discounts and to manage those costs closely. This may mean that they are managing the equipment part of your budget for you, which will save you a lot of time tracking it. The negative side of this is that you may be limited in who you can order from. Your company may have purchase agreements with a limited number of suppliers, and if you want to order from elsewhere, there may be a long and painful process to go through to get that approved. If it is not approved, you may end up having to order equipment that is not really what you want or need and may require some work-arounds to make it suitable for your project. (This happens more often than you would think in larger companies. The process is running the company and project rather than the company and project running the process.) You also lose the personal contact with the supplier, which can make it harder for you to follow up to ensure that your order will arrive on time and that the procurement department did not forget to order any accessory items that you needed. In smaller companies, you may be the person who is responsible for all the negotiation for your equipment.